José Pedro da Silva Gouveia Proença

Licenciado em Ciências da Engenharia Eletrotécnica e de Computadores

Web-based Interface for Tailored Time Series Analysis and Visualization

Dissertação para obtenção do Grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Orientador:	Doutor José Manuel Fonseca, Professor Associado com Agregação, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa
Co-orientador:	Doutor Ricardo da Costa Branco Ribeiro Matias, Investigador, Fundação Champalimaud

Júri

Presidente:	Doutor Rodolfo Alexandre Duarte Oliveira
Arguente:	Doutor Rui Manuel Leitão Santos Tavares
Vogal:	Doutor José Manuel Fonseca

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2018

Web-based Interface for Tailored Time Series Analysis and Visualization

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

For my parents who have been the force that pushes me towards knowledge and personal improvement.
For my brother who's example and courage have shown me that achieving these goals is possible.
For Ana Catarina Rato who thought me a better way to be happy.
For Kinetikos that welcomed me so well.

Acknowledgements

Before anything else, I would like to acknowledge and thank my advisors PhD José Manuel Fonseca and PhD Ricardo da Costa Branco Ribeiro Matias, who made this project possible. To José Fonseca I would like to thank all the knowledge, sincerity and availability. His advices marked me not only during this project, but throughout my whole academic course. To Ricardo Matias I would like to thank his thrust, vision and never ending eager to perform better work and achieve bigger results. Their vision and advise are the reason why this work is relevant in the field of Electrical and Computer Engineering as well as that of motor skill disorders.

A strong word of acknowledgement to *Kinetikos* and all of its working force. They were my daily company throughout this project. I've learned so much from them.

I also want to thank FCT-UNL for all the efforts made for their students, not only in the multiple academic fields, but also at the human level.

Another acknowledgment to my great friend João Pedro Lopes, who accompanied me since the beginning of this journey in FCT-UNL, for his ambition and work habits made me realize that anything is possible if we give our very best.

To my family, for they shape their lives to make all of this possible.

To Ana Catarina Rato, for she was always the helping hand.

And lastly, to all my friends, for they have always something new to teach me.

Abstract

In the field of movement disorders there is a clear need for more accurate and insightful metrics to support clinicians in their decision-making process. These metrics should be easy to understand and get presented to the clinician in an intuitive way, to avoid creating a wall between clinicians and their patients. This could happen if the information system that presents these metrics requires too much interaction, if it doesn't show the relevant data for each situation or even by delaying the information collection and its processing. Clinical Decision Support Information Systems have been growing in usage and usefulness, but still, there are cases of miss-implementations, that don't put the user in the first place, or that don't take into consideration their workflows. This leads to less effective health care and poorly spent resources.

With this project we will produce a customizable web interface, to integrate such a system, that must have clinicians and their needs as the first priority. This interface has to be fast and intuitive, while, at the same time, present useful and clinically relevant metrics. Additional information will be presented as optional and non-intrusive, not requiring any input from the user unless he wants to interact with it.

Keywords: Clinical Information System, Clinical Metrics, Gait, Customizable Web-Interface

Resumo

No campo das disfunções do movimento existe uma necessidade clara de métricas relevantes e precisas que apoiem a decisão médica. Estas métricas devem ser apresentadas aos clínicos de forma intuitiva e fácil de entender, para que não se crie uma barreira entre o paciente e o seu médico. Tal poderá acontecer se o sistema que apresentar estas métricas necessitar de demasiada interacção do utilizador, se não mostrar as informações relevantes para cada situação, ou ainda por levar a atrasos na coleção e tratamento dos dados. Os Sistemas de Informação Médica têm vindo a crescer, tanto na sua utilização como na sua utilidade, no entanto, ainda existem casos de sistemas mal implementados, que não põem o utilizador em primeiro lugar, nem têm em consideração os seus ritmos e modo de trabalho. Isto resulta num cuidado médico menos eficaz levando ao desperdício de recursos.

Neste projeto iremos produzir uma interface *web* customizável integrada num Sistema de Informação Médica, que terá os clínicos e as suas necessidades como primeira prioridade. Esta interface será rápida e intuitiva, apresentando métricas e informação clinica relevante. Toda a informação adicional terá de ser apresentada como opcional e de forma não intrusiva, pelo que não poderá requerer nenhuma interacção do utilizador a não ser que o mesmo assim o deseje.

Palavras-chave: Sistemas de Informação Médica, Métricas Clínicas, Marcha, Interface Web Customizável

Contents

List of Figures

Glossary

Clinical Information System	Information System that enables the collection and manipulation of clinical data allowing the construction of a clinical data history for each patient. This, in conjunction with the easing of health information exchange, relevant charts and possibly some artificial intelligence built right into the system, is the foundation of a good clinician decision support, that this type of system aims to be.
Clinician Decision Support	Decision making support, given possibly from a CIS in the form of metrics and graphs, that helps the clinician validate his or her reasoning regarding one patient.
Web-App	Application that runs on a browser, making use of an Internet connection.

Acronyms

CDS	Clinical Decision Support.
CIS	Clinical Information System.
CSS	Cascading Style Sheet.
DOM	Document Object Model.
DST	Double Stance Time.
FCT	Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.
FF	Foot Flat.
HO	Hell Off.
HS	Heel Strike.
MS	Mid-Stance.
MSW	Mid-Swing.
OS	Operating System.
ROM	Range Of Motion.
SPT	Stance Phase Time.
SST	Single Support Time.
SVG	Scalable Vector Graphics.
SWT	Swing Time.
TO	Toe Off.

Chapter 1

Introduction

In this chapter, we will first take a look at what this work goals and motivations are. In other words, what is the problem that we are trying to solve and what solution are we aiming to build. Then we will talk about this dissertation's framework and see in what ways does it integrate into the fields of Clinical Information Systems and Electrical and Computer Engineering. Finally, we will explain how is this dissertation organized.

1.1 Goals

In the current work, we aim to build a web-based interface for tailored time series analysis and visualization. The main focus area for this web interface will be its integration on a Clinical Information System (CIS), aimed primarily at movement disorders. This interface needs to be completely customizable to meet the specific needs of every physician presenting only the relevant metrics and charts. This solution will be built for the web, using *HTML5*, *CSS* and *CSS-Grid*, *Django*, *JavaScript*, *D3.js* and a few other technologies.

It's important to note that this work is a partnership between Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCT) and *Kinetikos (Coimbra, Portugal)* and comes from the necessity to create a tailored user interface for the *Report* area of *Kinetikos'* own CIS, that puts in the hands of their users the ability to customize the reports of their patients.

1.2 Motivation

For the vast majority of diseases, there are tests and clear metrics to aid and support the decisions made by clinicians. As an example, in the case of diabetes we have the glycohemoglobin test, that gives information about the patient's average levels of blood glucose [22] to the clinician. That's not so much the case, however, for movement disorders, where there is the need for more effective and clear metrics to assess a patient's state and evolution. Even more, evidence-based medicine is still not easy for physicians to practice, [2] making its adoption smaller than desired [18], despite its broad support in health care.

It's from this necessity to support clinical decisions with evidence-based data, specially in the field of movement disorders, that comes the motivation for this work. We believe that by building an intuitive web based medical report interface, to be integrated on a CIS, we'll be able to simplify their practices, and support their clinical decisions. This support would come, naturally, from displaying useful and accurate data, alongside with movement-related scores that represent the patient state and evolution throughout the treatments. This work gains even more relevance given the fact that movement is an essential part of life, sometimes overlooked and not subjected to good medical care in an useful time frame. For example, movement plays a major role in our physical, cognitive and social development, specially in younger ages. [9] Hence the major role of good assessment and diagnosis tools in the field of physiotherapy.

During this work, as we've seen, we'll build a report page to display a patient's clinical information in an intuitive way, but it's important to understand how is this information getting collected. This is done with the help of sensors to assess lower limb 3D biomechanics, that are placed on the patients while they move. Then, with *Kinetikos* algorithms, the raw sensor data that got collected, is converted both into scores and time series.

Another aspect that motivates us, is that the utilization of CIS has been, not only growing, but also increasing its usefulness in the medical environment. This type of information system enables the continuous monitoring and manipulation of a patient's clinical data, storing all of his medical history, [3, 21]. By doing so, a CIS has the ability to aid clinicians in their work environment in a lot of ways, as diverse as charting, Clinical Decision Support (CDS), Computerized Physician Order Entry or even the exchange of medical information [21]. However, a CIS has to be built with the clinician in mind, taking into consideration his needs and the way he will be interacting with it. A CIS must never get in the way of a patient and his clinician.

We see then, that since in the field of movement disorders we are lacking data to support clinical decision and since CIS are gaining momentum and relevance in the clinicians way of work, there is a clear need for a highly customizable *Report* area, that presents clinicians the information they want about each patient in an intuitive way, when and how they want it. A type of CIS built with such abilities would mean a clear ease in the clinician interaction with the system and could lead to great improvements in the results regarding the patient's rehabilitation.

For all of these reasons, we hope that the present work comes as a great move forward in the field of movement disorders, and CIS in general, helping clinicians and their patients in the journey towards recovery and long-time health monitoring.

1.3 Dissertation Framework

1.3.1 Integration With Clinical Information Systems

As we'll see in 2.2.1, a well-conceived CIS can have a great positive impact in the workflow of a clinician [21], in the one hand, by the way it handles data collection and its management and, in the other hand, by the way that data is contextualized and presented, opening the door to the above-mentioned CDS techniques, 1.2. Non the less, there is still a great need for better and more intuitive information systems inside clinic facilities. The two main problems of the current CIS in use, are the amount of time the clinicians lose because of systems inefficiency, and the lack of clarity in the sense that the system doesn't reflect the workflow of the user.

With the construction of a report page that is customizable we think clinicians will be free to create a more friendly CIS, that they can tailor to better represent their workflow.

1.3.2 Integration With Electrical and Computer Engineering

This work gains its relevance in the field of Electrical an Computer Engineering, from the fact that it consists in the development of an information system, aimed at clinicians with the intent to solve some of the problems, discussed on Subsection 2.2.1, found in the currently used CIS.

1.4 Dissertation Organization

The current Dissertation is divided in 6 chapters: By now, in this first one, we discussed our motivation and how this work integrates the fields of CIS as well as Electrical and Computer Engineering.

Next, in the second chapter, we will explore what has already been done in the field of CIS, time-series analysis and visualization for motor disorders and, lastly, in the field of tailored and customizable web systems. Regarding CIS, their advantages and disadvantages will be addressed, as well as, what, according to the community, should be the main focus points and guidelines, about the usability and built of such a system. Some aspects of what is possible when it comes to presenting data in meaningful ways, especially in a Web-App with all the currently available technologies, will also be discussed.

Later, in the third chapter, the pre development phase is described. We start by clarifying the features and constraints of the solution that we built, to better understand our goals. The first ideas and solution concepts are also discussed, as well as all the design process.

With the forth part of this document, comes the development phase, where we talk about all the major aspects that came with the production of the report page. These aspects range from the database structure, to some Front-End concerns and decisions that needed to be made.

In Chapter 5, we get to the discussion about the results achieved with the construction of our solution. Such discussion is made with an eye on all this work features and constraints, presented in Chapter 3, and all the information gathered throughout the State of the Art, 2.

By the end of this Dissertation, we present all the conclusions that can be taken from what was created during this process, alongside some possible future work.

It is also very important to note that, this being a partnership between FCT and *Kinetikos*™, the execution of this work took place at *Kinetikos'* offices.

Chapter 2

State of the Art

2.1 Introduction

In the present chapter, we will look at the current state of CIS and the technology that we'll use to reach the goals set for this dissertation work. We will also take some time to understand what are the standard tests and representations of movement in the medical field regarding gait, since those metrics and time series will be the ones present on the final medical report produced by the end of this dissertation.

2.2 Clinical Information Systems

As stated in *The electronic medical record, safety, and critical care* [5], since the year 2000 the interest in Electronic Medical Recording has been growing. With the promise of reducing deaths related to medical error, the implementation of systems that enabled Computerized Physician Order Entry in Intensive Care Units has given way to the creation and growth of what we perceive today as being a CIS.

As we said in 1.2 a CIS is a computer program that enables the collection, manipulation and visualization of clinical data. This data and medical records are what makes a CIS something more than just a patient management tool, because they are what enables CDS. The concept of CDS is defined as being the process to enhance the decisions and actions made in the health environment, thus improving healthcare delivery, by using pertinent and organized clinical knowledge, alongside the patients information. [6, 24] This support given to clinicians by their use of a CIS can come in four forms [6, 24], namely:

- **Data Entry** - Which encapsulates all of the patient's medical data as well as family history. It is important in this phase to make this process easy and quick, but simply, with data entry the clinician can collect all the data in a single place;

- **Data Review** - This support area comes in the form of the visualization of the data in an organized form. The data can be, as an example, the real time vital signs of a patient. [6] In the case of this work, the data review aspect will come in the form of the created report area, that allow clinicians to see the clinical data and metadata about each patient;
- **Assessment and Understanding** - We can look at the assessment of the entered data as the process of transforming it into clinical insights and information, to support clinicians as they discuss treatment options [6]. This goes a little further than just presenting the raw data to clinicians, but rather giving then real information upon which they can act. This type of assessment will also be present on the report area created during this work in the form of the displayed metrics calculated using *Kinetikos* expertise and intelligence;
- **User Triggered Support** - Such support comes in the form of alerts to draw the clinician's attention to something specific like an abnormal test result about a patient, or an examination that is due today. [6] In the current work some of this aspects were also implemented, like a small calendar where a clinician can see his appointments.

These systems should also process this data, retrieving useful information, like scores and charts presenting them in meaningful ways. Some of these metrics and time-series will be covered in section 2.3. In this manner, these systems can be quite useful aiding clinicians in their decision-making process. [18]

By applying these techniques to a CIS aimed at motion skill disorders, we hope to improve the effectiveness and reduce the errors in health care delivery.

2.2.1 Advantages and Disadvantages of a CIS

As with all solutions, a CIS can have great advantages, but, when poorly implemented, can also have a few downsides. Let's look at both advantages and disadvantages, taking into consideration what is said in [21], so that we have a better understanding of these systems and what areas should be taken into heavy consideration while undergoing the construction of this project.

2.2.1.1 Advantages

Regarding the advantages we have:

- **Data Collection** [6] - Given it's capabilities, CIS can take in and structure large amounts of data, that can be used to provide better medical care or to study new ways of treatment. In the field of movement disorders, a CIS can allow clinicians to collect data about the patient's motion, possible with the help of sensors, or to register values such as step length, or range of motion, and then use os of this data to retrieve clinical insights.

- **Data Availability** [21] - The physician has access to all the relevant data collected about it's patients.
- **Data Sharing** [21] - Despite not being consensual in which manners data sharing in CIS should be done, with structured data, sharing of knowledge and useful information is made very easy. This can be extremely helpful for studies [33] and also, for a better communication between physicians, which can help in CDS.
- **Data Security** [21] - Unlike what happens with paper logs, data collected through a CIS, if well implemented, is stored with a great amount of security measures, including encryption of all patient records. Further, data stored in servers is usually stored with redundancy, having copies in several places around the world, so that it can always be retrieved despite server failures or natural catastrophes. In other cases, the stored files are divided into many pieces, storing each piece in different serves, making it impossible to retrieve all of the information stored, in the remote event of one of them being hacked.
- **CDS** [21] - When presented in meaningful ways, data about a patient's medical history and current health state can be of great value to validate or aid a clinical decision. This data can tell if a chosen method is being successful towards patient recovery and at what rate is that recovery taking place. As discussed in the beginning of this chapter this support can be given in four formats, Data Entry, Data Availability, Assessment and Understanding and lastly User Triggered Support.
- **Implementation of Best-Practice Guidelines** - A CIS can have the best-practice guidelines integrated into the user's workflow, thereby improving care given to the patient. [21]. By using these workflows there is less room for human error, as an example, from skipping some essential step in an examination.

2.2.1.2 Disadvantages

Looking now at the possible disadvantages regarding CIS, we have:

- **Implementation Errors** - Being an information system, a CIS is always prone to implementation errors, such as miscalculated metrics or confusing and not user friendly interfaces, that don't respect their natural work flow. [21, 29]
- **Efficiency Decrease** [21] - A system like this, if badly implemented may increase the steps that a physician has to take to accomplish a certain goal. The learning curve to specialize in the CIS work flow can be quite big, specially if it was built

without any consideration for the physicians natural work flow. Lastly the system could be really slow or depend on the Internet's availability or velocity.

- **System Dependency** - Being a computer system, a CIS needs some resources to function, such as an Internet connection or certain Hardware. If any of the components of the system fail, it might turn completely useless, making the flow of data impossible. [21].

2.2.2 What Needs to be Addressed When Creating a CIS

After the presentation of CIS's advantages and disadvantages, it is time to look into what the community thinks are the main focus points when starting the implementation of such a system. A lot of the points that we are about to present are intrinsically tied to the Report page that is the central part of this project.

First of all, it is very important to never forget that a CIS is a system that will be used by clinicians, a lot of times during their appointments with the patients. This means that their workflows need to be addressed [2, 6] and that the team building the system needs to know very well what are the main problems that need to be tackled down. In order for a CIS to be good, it has to have efficient workflows and require little training to learn. [21]

Secondly, this type of system needs to operate really fast. As said by Shabot, M Michael, the most important parameter for clinical users is the response time of the systems they use during their day-to-day utilization, namely subsecond response times [2, 29]. So during this project speed must be a key factor, prioritized well above all the others, even that of having a clean and beautiful design.

One point of real importance is that all the information provided to the clinician, specially that that comes from the assessment of clinical data should always be evidence based [6].

Another important attribute of a CIS is the delivery of only the relevant information where and when it's needed. [6] This is one of the main focus points of this project, where the Report page can be tailored to fit each physician preferences. It is very important to implement all of the features in a nonintrusive way. Additional capabilities or information should appear as options that don't require the user's dismiss. They should make them self-present when the user needs them, [2] but not require any action unless the he wants to use them [29]. A good example, as stated in by Michael Shabot [29] is the CIS used by Brigham and Women's Hospital, named BICS. This system had an interesting feature, that consisted of showing the patient's last measured potassium level alongside the dose level choices, when the physician ordered digoxin. This meant that the relevant information was present when needed, but didn't require any interaction unless the physician choose to act upon it.

One thing that is relevant and should not be overseen in a CIS is its usability [2]. The development team has to make it really easy for the clinician to act as intended with as little effort as possible. This means that the design aspect of the system must reflect the action that is intended in each state. As we said previously, if something is just additional information, it should be nonintrusive [29], but, in the case that something should not be ignored, it must be hard to ignore it [2]. To enable a better CDS, the information must be presented in the right way [6]. This will also be a major concern in our work.

As any computer system, a CIS will most likely be subjected to iterations and adjustments. This need to implement new features or change some of its core aspects, one of the main ones being the physician workflow, has to be forethought so that it may be an easy process in the future. Also, the databases upon which the CDS is created, need to be constantly revised and updated. [2]

Concluding, after looking at all of these considerations we can say that the main concern for the creation of a CIS is its speed, and a great understating of the real needs of clinicians prior to its construction. A CIS built with user-interfaces that have the clinician requirements in mind will show better results and be quite more effective. Ahmed A *et al.* have tested just that, and concluded that user interfaces within a standard electronic medical record that are developed with an understanding of the healthcare provider's needs, turnout to have more advantages than the currently used interfaces, that didn't had those concerns as a priority. [1]

2.3 Time-Series Analysis and Visualization

In what concerns our project, given that it will be designed to integrate a CIS aimed at motor skill disorders, it is important to have a clear understanding of what type of time-series we'll encounter. These time-series will represent certain movements of the human body, such as gait.

2.3.1 Gait

Gait can be described as being a person's manner of ambulation or locomotion [14, 16] and it involves the whole body. The standard gait cycle at a normal walking speed involves the lower limbs as well as the trunk and arms that are used to provide stability and balance. [16] Also during gait, the three major joints of the lower body and pelvis, are working in conjunction with the muscles to propel the body forward. [16] As for the cycle itself for one leg it is described as a repetitive pattern, composed of two major phases called stance phase and swing phase, the first one making for 60% of the cycle and the second, the remaining 40% [14, 19]. During the stance phase, the foot is always in contact with the floor, while the swing phase goes from the initial moment where the foot spots touching the floor until it makes contact again. [23]

As we can see from figure 2.1, these two phases can then be subdivided into six more detailed phases, which are: [14, 16, 19, 23]

1. Heel Strike (HS):
    - Happens when the heel makes contact with the ground;
    - The hip is flexed at 30° and the knee is fully extended; [14]
    - Marks the start of the gait cycle;
    - The leg begins to receive the weight of the body;
2. Foot Flat (FF):
    - At this point, we are at 10% of the full gait cycle;
    - The foot lays flat on the ground;
    - All the body weight is placed on this leg, and we enter single limb support;
    - The second leg starts swing phase;
3. Mid-Stance (MS):
    - Goes from 10% to 30% of the gait cycle;
    - The body stops absorbing the impact on the floor, and transitions to forward propulsion; [14, 16]
    - Right after this point, Hip-extension begins;
4. Hell Off (HO):
    - Occurs at 50% of the cycle;
    - Contact between the heel and the floor ends;
    - Knee flexion between 0° to 5°; [14, 19]
5. Toe Off (TO):
    - Happens at 60% of the gait cycle;
    - Marks the end of the stance phase;
    - The knee flexes between 35° and 40°;
    - The toes no longer touch the ground. [14, 19];
6. Mid-Swing (MSW):
    - Marks the middle of the swing phase and occurs at 87% of the cycle;

- The tibia is vertical and perpendicular to the ground;

After all of these phases, the cycle ends when the heel makes new contact with the ground.

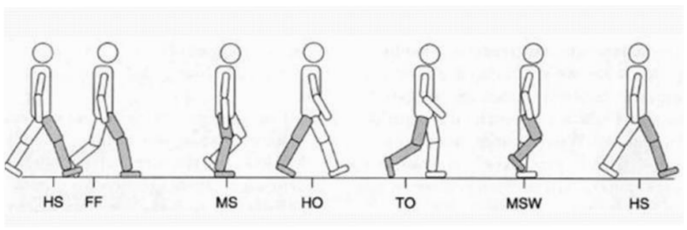


Figure 2.1: Six phases of a gait cycle. Image taken from [32].

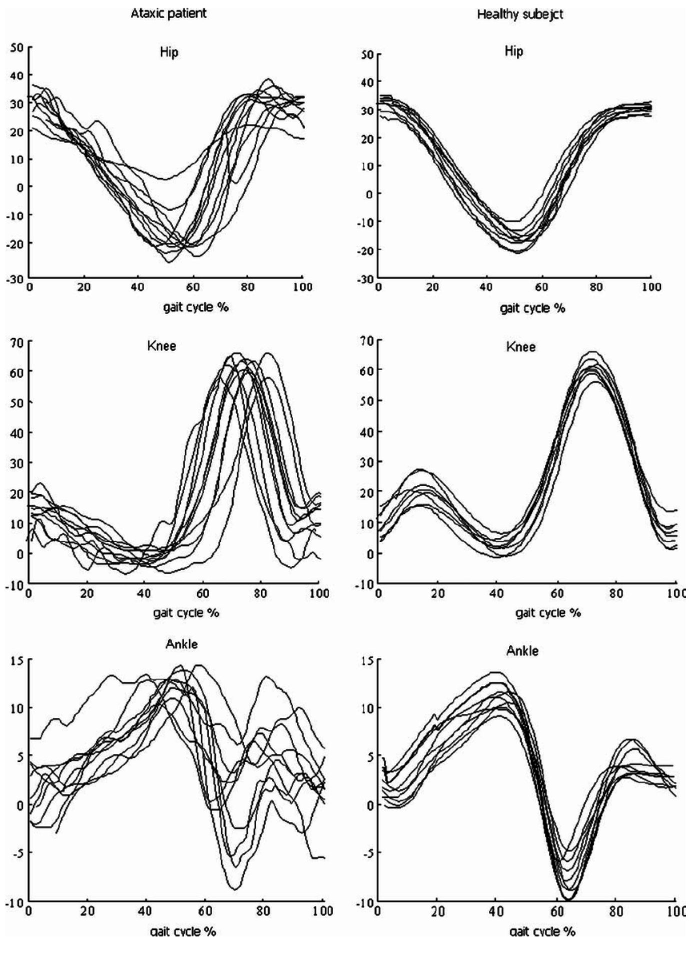


Figure 2.2: Multiple examples of gait cycles representing the hip, knee and ankle angle(deg) in order of the cycle's percentage of completion. Image taken from [28].

When it comes to the representation of these events, a usual representation uses the angles of the Ankle, Knee or Hip in order of the percentage of gait cycle or time spent as showed in Figure 2.2. Doing such representations, that have these large amounts of data and variables, can evidently result in a hard to read plot. So, giving into consideration what was discussed in 2.2.2 about the intuitiveness of a CIS, it is important that the representations presented in the final solution of this work have the ability to be zoomed or panned, and even the ability to dismiss individual lines in a plot. This concerns were present in the development phase, as can be seen in 3.1 or 4.4.3

Another possibility is the representation of angle-angle between two of these variables, for example, ankle-knee, as we can see from figure 2.3.

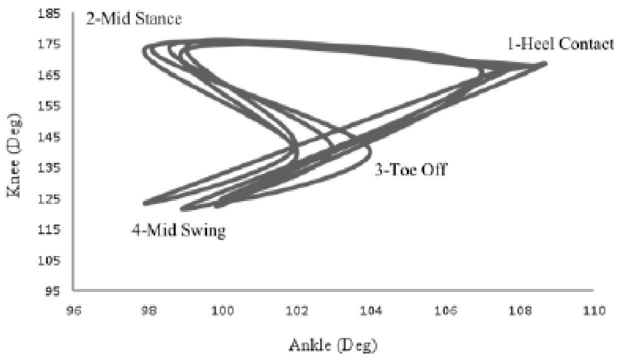


Figure 2.3: Example of an angle-angle plot representing knee and ankle joints during a gait cycle for a healthy population. Image taken from [30].

It is then important to understand how this type of event is found from a continuous recording of a person's gait. The most commonly used segmentation algorithms are peak detection algorithms based on the Z-score rule. [17] As an example, the first method takes advantage of the fact that the signal recorded has semi-periodic characteristics, namely, peaks and vales, that are maximums and minimums of the signal. New and more robust methods have been presented, due to the need for algorithms less prone to signal noise. This need comes from the fact that a patient with a pathological gait usually has multiple noisy peaks, in opposition to the theoretical gait cycle, [15] thus making it harder to segment the cycles. One of these robust algorithms is the one proposed by Jiang, Shuo *et al.* where the algorithm makes use of interpolation in the time domain to eliminate false peaks and detect the ones that are missing, since the cycle lengths, despite the pathologies, tend to be uniformly distributed. [17]

After having these gait events detected, another necessity is to calculate objective scores for a patient's gait, which will be one of the main focus points of this project. A really intuitive way to do this, for example, is to normalize the obtained signal and compare with the one of normal gait accounting for the standard-deviation, calculating the percentage of time that the signal was not between the upper and lower bounds, like shown in the figure 2.4.

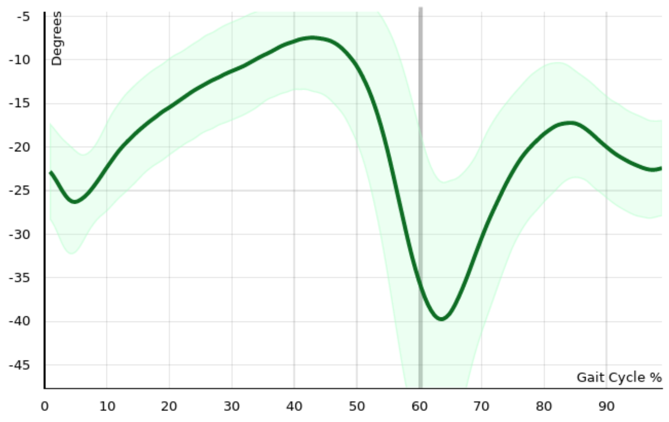


Figure 2.4: Plot of the Ankle's angle(deg) in a normal gait cycle and upper and lower bounds. Image adapted with permission from a Kinetikos' gait report.

From the detected events we can also extrapolate some temporal features, namely Swing Time (SWT), Single Support Time (SST), Stance Phase Time (SPT) or Double Stance Time (DST). [7] These features will then help us compare the subject to the normal range of values, attributing a score to the obtained gait cycle signal that can help the clinician make his decision or accessing the subjects evolution throughout the treatment.

SWT will be the time passed from TO until HS. Related to this metric, is the SST, because"*in fact, one leg's SST is exactly the same as the SWT of the other leg*". [7] As for the SPT, this will be the time passed when the foot is touching the ground, so from HS to TO, while in DST we calculate the time where both feet are making contact with the ground. [7] This time goes from HS of the first leg until TO of the second one.

In short, the main calculations that we'll be made during this project are: [7]

$$SWT = TO - HS \tag{2.1}$$

$$\begin{cases} SST_{RightLeg} = SWT_{LeftLeg} \\ SST_{LeftLeg} = SWT_{RightLeg} \end{cases} \tag{2.2}$$

$$SPT = HS - TO \tag{2.3}$$

$$\begin{cases} DST = SPT_{LeftLeg} - SWT_{RightLeg} \\ DST = SPT_{RightLeg} - SWT_{LeftLeg} \end{cases} \tag{2.4}$$

2.4 Web Interfaces

In this section, we'll cover what the best options are for building our web interface and its core features. But first of all let's dive into the reason why, at the beginning of this project, building a Web-App was the chosen option.

2.4.1 Web-App

The web browser is one of the main content consuming software, existing since the early 1990's. It has been evolving ever since, with a lot of breakthrough moments in the most recent years. A lot of sources declared web-apps the way to go, stating that it will be the major way to interact with content in the future.

This type of application has a lot of advantages over the more traditional approach of creating a local program [13, 34] that runs either on Windows, Macintosh, Linux, or any of the mobile platforms' software, such as Android or iOS. The main advantages have to do with the fact that despite some layout differences, all web browser will show the same app, with the same features, leaving out the need to build a different app in a different language for every type of Operating System (OS). This also makes updating the app really fast, given the fact that the whole team is working on the exact same application, and, as we said in 2.2.2, being able to be quickly updated is one important aspect of a CIS. Also, since there is no need to download or install any part of these type of applications, all users are always on the most up to date version.

However, there are at least two major downsides in this type of approach. The first one is that a Web-App relies on the existence of an Internet connection and, in order for the application to run smoothly, it has to be a strong one. In the recent years, with the evolution of web technologies and the introduction of WiFi and 4G LTE connections, this is starting to be less and less of a problem. In the case of clinics and hospitals, the need for this type of connection is known in advance, so the contracted bandwidth should have this type of use in mind, hence making it less likely for the system to suffer from slow Internet. The second major downside, however, is precisely the need to have such a contract, that, to fill the needs of these systems, will have to be considerably more expensive. [13, 31, 34]

2.4.2 Browser Technologies

The Web Browser as said in 2.4.1 has been evolving, and became one of the primary means to transmit and display information. The browser is the way into the web, and thus, the technologies behind it have been getting better and better at displaying information in intelligent and significant ways. Some of these technologies come in the form of libraries or frameworks that enable a more high-level approach to the page flow or allow the construction of really precise interactive charts, videos or 3D models. In this section, we will take a quick look at the major technologies that will help us achieve our goals during this project.

2.4.2.1 D3.js

D3 stands for "Data-Driven Documents"and is an open-source JavaScript framework that allows the “efficient manipulation of documents based on data” [4]. This means that with D3.js it is possible to create Document Object Model (DOM) elements to represent data in interactive ways, using only current Web standards like HTML, SVG, and Cascading Style Sheet (CSS) [4]. This is extremely powerful because it also allows the manipulation of these elements in an easy and usual way. The usage of D3.js has been growing a lot, so there is a lot of feedback and help forums maintained by the Internet community. During this project, this framework will be used to produce SVG’s with various types of interactive charts.

2.4.2.2 CSS-Grid

CSS-Grid is a new and powerful set of CSS rules that allows developers to layout any page element inside a grid. This is extremely useful and practical, finally allowing developers to think in a way that is closer to that of the designers, opening the possibility to build more interesting layouts and in a more timely fashion. Developers can now create a grid layout, including sub-grids, and then place the elements inside the desired cells or even leave some of them empty. [8, 10] These rules were also created with various screen sizes or even resizable screens in mind, so adaptations and rearrangements of the layout are totally supported. This allows the developer to think and build two different grid layouts, one for desktop and one for mobile, and the correct one will be applied in the correct scenario. It is also very important to notice that this technology is already present in a large amount of browsers, including in all major ones.

2.4.2.3 Django

“Django is a high-level Python Web framework” [12] that focuses on fast and secure web development. With this framework, it is possible to use the powerful data analysis features of python, while building web platforms. This is also a really easy to understand language, making code readable and fast to produce. [35] This framework comes with a lot of built-in features, like authentication or session frameworks, making it really easy to implement them on web platforms. However Django doesn’t support real-time Web-App, so there is the need to use it in conjunction with other Python engines, such as Jinja2. [27, 35]. Looking directly at this work, it is relevant to note that Django is based on the Model-View-Controller, [11] thus it separates back and front end. Not only that but “Django takes this separation one step further by separating code from the static media—images, files, CSS and JavaScript”, [11] making for a perfect integration with D3.js creating the right conditions to manipulate and present data to clinicians in an intuitive, fast and beautiful way.

Chapter 3

Pre Development

To start this project it is first necessary to understand all the features that we need to develop to succeed on our project. In Chapter 2 we could already understand what are the main aspects that compose a good CIS and what should be avoided. These concepts were already a great basis for us to drive the design phase upon. However, with *Kinetikos* being a company that has direct contact with clinicians, we had access to an even clearer and more relevant view of what should be present in our final design. In the next sections we will present the constraints and features of this project, as well as explain all the main concepts that are necessary to understand the development of this solution. By the end of the Chapter we will also go through the design phase, with the presentation of some mockups and drafts.

3.1 Features and Constraints

The goal of this project is, first of all, to create a report area inside of a CIS that presents clinicians with meaningful information about their patients in a intuitive and tailored way. This goal is then composed of some features and requests that are listed bellow, and came from *Kinetikos* expertise and feedback, as well as the research presented in Chapter 2:

- **Customizable** - The report area must have the ability to be tailored by each clinician, in order to display only the information that they want and in the way that they feel most comfortable with.
- **Quick and Responsive** - The report area, being full of data and plots, is an area that tends to be slow to load. This, however, should not be the case, given the timely fashion at which clinicians have to work. So, in this project, we should have in mind

that the code and the loading of all elements needs to be efficient. The same applies to all interactions inside the page, that should be quick to respond.

- **Intuitive** - The report needs to be intuitive and easy to understand and navigate. The learning curve should be low so that our solution is ready to be used just after a few moments of exploration by the clinician.
- **Interactive** - When displaying data, like plots, it's important that the users can interact with it. This interactions could range from simply zooming and dragging, to viewing only sub selections of what's being displayed.
- **Mobile Friendly** - Some clinicians voiced their frustration when trying to use their phones to navigate the systems that they use during work. So, in this project, we'll need to prepare the report to be used on mobile, to allow the users to quickly check the information about their patients on the go.

Now that we have an even better understanding of all the objectives of this project, we can present the phases that occurred before the beginning of the report's development.

3.2 Template Layout

One of the main ideas behind this project was to make the report page tailored for the clinician. This is important because we saw, in Subsection 2.2.2, that a CIS must respect the clinician's workflow. By being able to customize the way the report page looks and which information it presents, we think the user will be able to reflect his way of working into the report itself, thereby fulfilling this concern. In order to achieve this type of user customization, we decided to implement the concept of templates. In this case, a template will be a different tab in the report page, that will hold different types of data and is completely independent from the other tabs. The way the information is presented in these templates, as we will see in next Section, 3.3, is through small cards. But, by now, the most important aspect is that each user can create and edit their own templates. In other words, clinicians will be able to create different tabs with only the data that they want to view and organize them in the way they feel more logical.

Templates are organized by the type of exam that they were created for, but they are not limited to only one type. In this manner one template can be used for multiple types of exams. Templates can then be created and edited at any time, and these changes will be stored into the database. Additionally, it is possible to have templates that are of general access and so, available to all users. These will be created by *Kinetikos* and are not editable by the users. A similar functionality will be given to the health institutions at which the clinicians work. They too will be able to create general access templates for all their clinicians, that can be edited only by accounts with the right clearance.

We think this will be a very successful way of enabling a more tailored and meaningful report, that doesn't impose a specific workflow or methodology, but radder is open to user customization. In the Figure 3.1 you can see an example of a template.

Regarding the technologies used to create the templates' layout, we decided to use the new CSS-Grid standards, discussed in the Subsection 2.4.2.2. Hence, templates will be composed of a grid with a defined number of columns and an undefined number of rows, though they all have the same dimensions. The amount of rows present in a given template is thereby defined by the number of cards present on it.

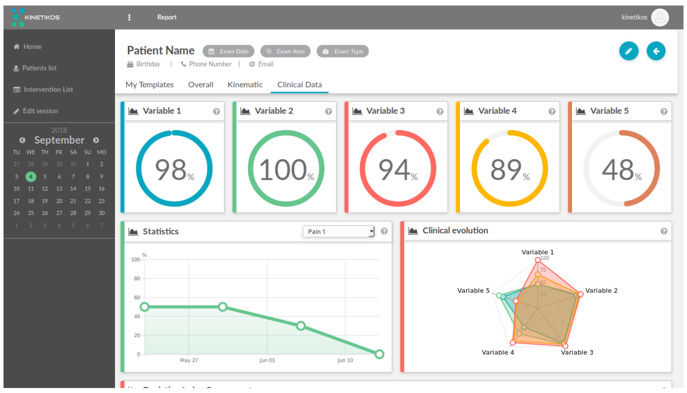


Figure 3.1: Example of one report template created during this project.

3.3 Informative Cards

To support the idea of templates, that we just presented, we needed to find a way to individualize every type of clinical data. We intended to individualize the data, not only to give clinicians the ability to add or remove information to and from the report, but also, to make it more intuitive and simple to read, which was also a concern brought up in the Section 2.2.

The solution found was the creation of cards, with different purposes and data. These cards were then divided into types, that represent a singular way of displaying information, since the data can be represented in metrics or single numeric values, scores and percentages, plots, videos plain text and so many other ways. As an example we can go back to Figure 3.1, where we find a first row made up of similar cards with a circular way of representing percentages. This first row is made of cards which are all of the same type, but that are completely independent from each other, meaning that they can be added or removed from the template individually. All the other cards in the figure are independent from each other, and are also of different types, since they don't share the same look.

After understanding this way of showing information, it's really easy to come to the conclusion that the templates introduced in Section 3.2 are nothing more than an ordered collection of cards. This order, as we'll see in Section 4.3, is also subjected to user customization. When editing a template, users will be allowed to choose all the cards that they want from a list of all available cards, and then drag and drop them around the canvas of the template to reorganize them.

We said before, in Section 2.2.2, that a CIS should be able to be quickly updated, hence that was another big concern that we had during the pre-development phase. To make our report quick and easy to fix or update, this card interface was of major importance. This is because we only need to worry about creating or changing one card and not the whole report. Building the report in this way, revealed itself as a great bet, enabling a faster and more seamless way to develop the report page.

3.4 Design

In this section we'll discuss the design aspects of the current project, from the initial drafts to the end result.

3.4.1 Report

Let's begin by addressing the outer part of the report, the header and the left side bar, both found on Figure 3.1. In the header we will find all the metadata about the patient and the current exam, as well as some action buttons and the templates available for this type of exam in the form of tabs. The Figure bellow presents an header section explaining where each type of information goes.

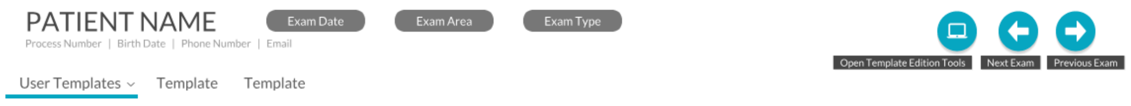


Figure 3.2: Detailed Report Header.

We can observe that this is a really rich and dense area, filled with useful information. By giving it a quick look, the clinician can find all the major information about the patient and the exam that he did with him. Regarding the exam information we have the date of the exam, the *Exam Type*, that can be *Admission*, *Follow Up* or *Discharge* and the *Exam Area* that is the area of the body where the examination took place. This layout, however, was an iteration from another draft made for this project, which, as found in Figure 3.3, had way less functionality and information, leaving a lot of unused space. This lack of functionality and wasted space were the reasons why this design got rejected.

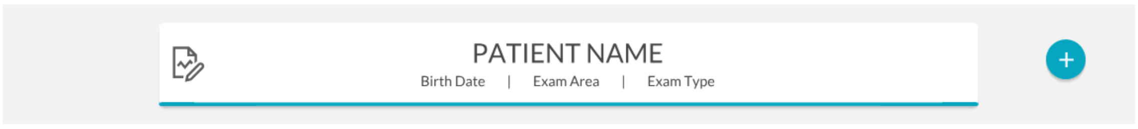


Figure 3.3: Detailed Report Header.

There is also a different way to present the header, that we'll call collapsed. This collapsed view occurs automatically, after the user scrolls beyond a certain limit, and has less information, which allows us to largely reduce its size, while still keeping the tabs and the exam information at hand. An example of this view is shown in Figure 3.4.

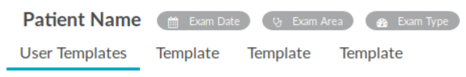


Figure 3.4: Collapsed Report Header.

Regarding the left side bar, found of Figures 3.1 and 3.5, clinicians will find there some quick links. This area is a good place for the type of CDS triggered by user, as presented in Section 2.2. This is also an area prone to updates, in the sense that these links can be easily exchanged as functionalities are added to the platform. One of the ways to integrate CDS in this bar came in the for of an interactive calendar, from which clinicians can navigate through their appointments, Figure 3.5. This area is also collapsible, disappearing completely, so that the informative cards fill the entire width of the screen.

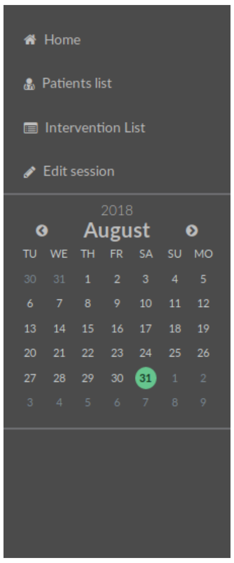


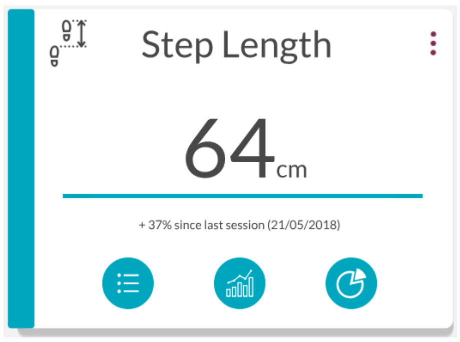
Figure 3.5: Example of the side bar with a calendar.

3.4.2 Cards

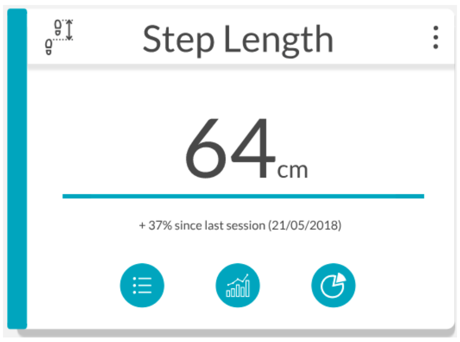
Looking now at the cards that compose the templates that we talked about previously, we can see that despite all the differences between them, there are some aspects that remain the same in every card. By maintaining a consistent design language, that keeps cards looking similar to each other, we aim to make everything more predictable and intuitive.

In Figure 3.6 to the right, we can see one of the draftings that were the base from which we created the cards that are used in the report. In this card we can observe some elements that became common on all cards, the first one being the header. In this part we have an icon representative of the information presented and the title of the card. We have also a three dots menu that by the end of this project was replaced with a button to open a pop-up with some help information about the card itself.

Next we can see the body of the card. Is inside of here that the data and useful information is found and it can be presented in really diverse ways. However, in all cards, when there is the possibility to take some action, like zoom in on plots, or show more information, these actions are presented as blue rounded buttons on the lower part of the card.



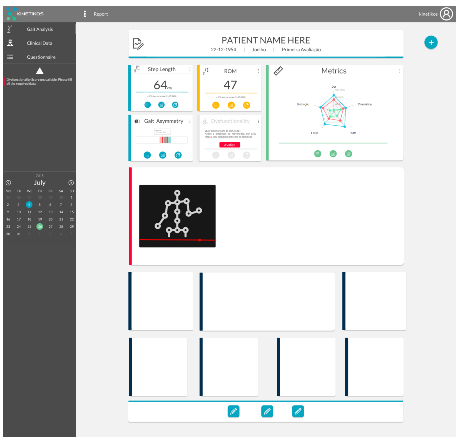
(a) First draft..

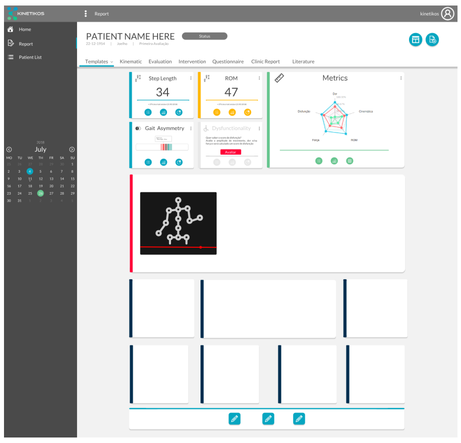


(b) Second draft with some small adjustments..

Figure 3.6: Initial draftings of one of the Card Types.

In the next Figures we can see some other drafts of the template compared with what ended up being implemented:


(a) First draft of the Report..


(b) Final draft of the Report..

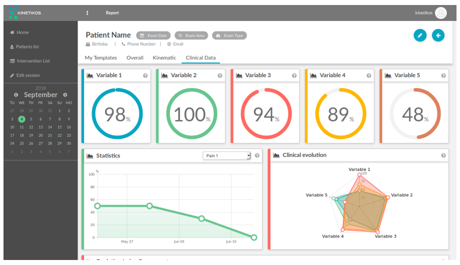
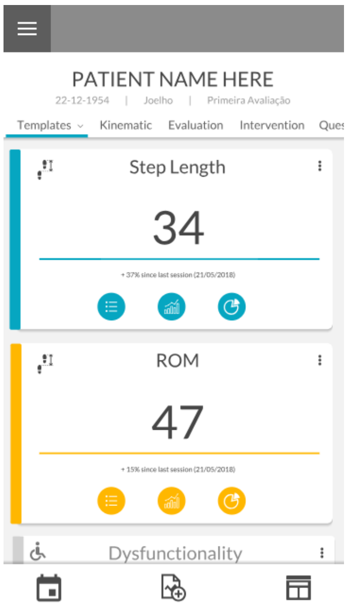

(c) Report after implementation..

Figure 3.7: Report drafts (on the top) and an example of a report by the end of this project (on the bottom).

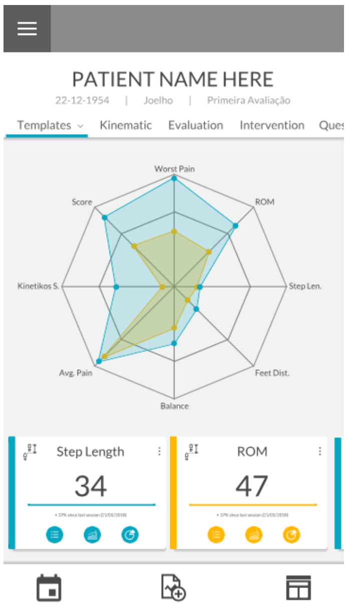
3.4.3 Mobile

Lastly, we did some design concepts for the way the report should look on mobile phones. They are all visible in the next Figure:

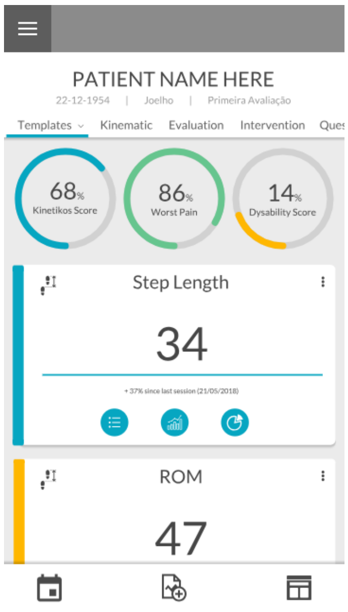
Despite some of the concepts being really visually interesting, like the full screen plot in the Figures 3.8e and 3.8f, we ended up following the first draft, in Figure 3.8a, since this one respected the design language used in the rest of the report, namely in the non-mobile version. This decision was a step in the direction of intuitiveness and simplicity, over beautiful design, like discussed in Section 2.2.2.


PATIENT NAME HERE
22-12-1954 | Joelho | Primeira Avaliação
Templates Kinematic Evaluation Intervention Ques
Step Length
34
ROM
47
Dysfunctionality

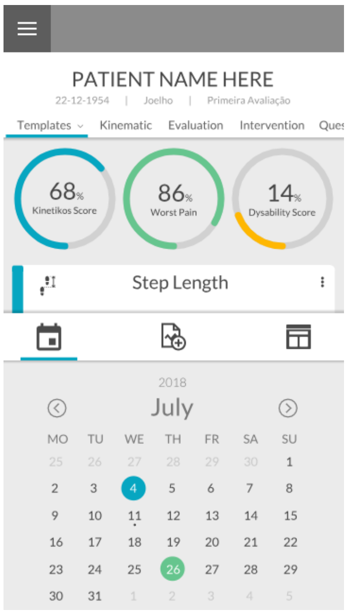
(a) Draft for the report on a mobile browser.



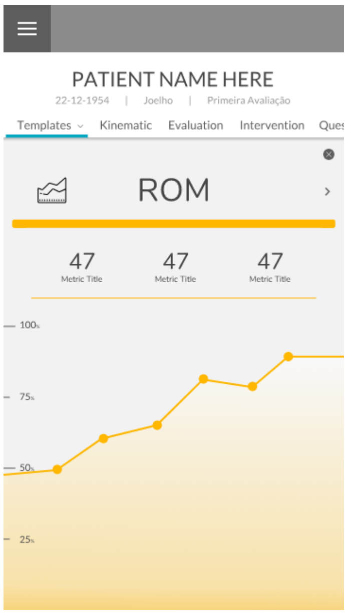
(b) Draft for the report on a mobile browser.


PATIENT NAME HERE
22-12-1954 | Joelho | Primeira Avaliação
Templates Kinematic Evaluation Intervention Ques
68% Kinetikos Score
86% Worst Pain
14% Dysability Score
Step Length
34
ROM
47

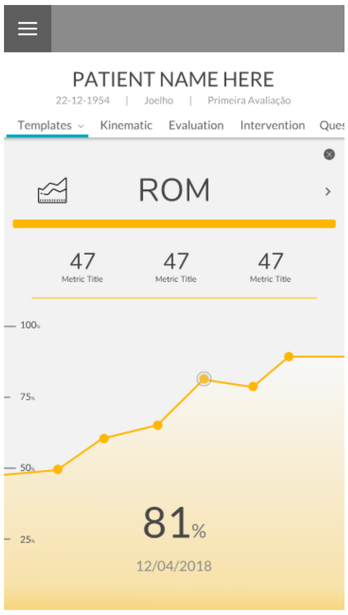
(c) Draft for the report on a mobile browser.



(d) Draft for the report on a mobile browser.


PATIENT NAME HERE
22-12-1954 | Joelho | Primeira Avaliação
Templates Kinematic Evaluation Intervention Ques
ROM
47 Metric Title
47 Metric Title
47 Metric Title
100%
75%
50%
25%

(e) Draft for the report on a mobile browser.



(f) Draft for the report on a mobile browser.

Figure 3.8: Set of drafts for the report as viewed on a mobile browser.

Chapter 4

Development

Starting the development phase it became a priority to start building the database structure. Later, some essential BackEnd functions were built, to enable the creation and management of the report templates. After this two phases we started to build the functions for each specific card and card type. At the same time the connection with the FrontEnd started to be created. Lastly, we built all the FrontEnd structure. These phases will be discussed in the next sections.

4.1 Database Design

Regarding the database structure, tables and respective relationships we'll start by presenting all the tables and their purpose. This explanations should be seen with attention to Figure 4.1, that represents the database that was created for this project.

4.1.1 Template

This table will represent the templates. It will hold the IDs of all the cards that compose the template, in the form of a list in the order that they should appear in the FrontEnd. This list could have been done using a cross reference table, like we did in all the other many to many relations throughout this project, however, doing it in this way presented a lot of advantages which led us to decide to use this strategy. They were: reducing the complexity of setting the order in which the cards should appear in a template, since the order is now just the order in which they appear in the list and not an integer value set in the relations table; enabled a quick and intuitive edition of the cards present in a template (mainly to easily update the templates created by *Kinetikos* itself, like discussed in Section 3.2); making saving the editions made by the user to the templates, a really simple process, given the fact that the only thing that needs to be done in the FrontEnd, is to send a JSON array of card id's to the BackEnd that stores it as is.

In this table we can also find a name that is there just to make the database more readable and is never used on the report itself. Lastly we can see three more values, *inst_id*, *user_id* and *type*. These are responsible to define who created the template. If it was a clinician the user and the institution ID will be filed and the *type* set to 1. In the case that it was an institution, the *user_id* will be set to *Null* and the *inst_id* will be filled. The type is then set to 2. Finally, if the template is of general access, in which case it was created by *Kinetikos*, neither the user nor the institution IDs are filled and the type is set to 3.

4.1.2 Exam_id_Template

In this project, like seen on Section 3.2, templates are connected to exam types. This means that an exam type can have many templates and that a template can be used in many exam types. Given this, we need a cross reference table, which is the exam_id_template table. In this table, besides the IDs of the exam type and the template, we have an integer that represents the order in which this specific template tab should appear in the header for the reports of this exam type. This is the opposite strategy to the one taken for the cards order stored in the template table. This happened because there was no need to edit this order directly on the database.

4.1.3 Template_text

This is a very common type of table throughout this entire project. These tables are responsible to manage the text values of their parent entries. They will have the values in all available languages in the platform, by relating each entry with the corresponding language ID. For the case of templates we need a title, that will appear in the tabs, and a description that could be used in some help menu in the future.

4.1.4 Card

Within this table we will organize all the information about the cards that compose the templates. The Cards table will have a column for the card type ID, that is a Foreign Key; an icon that is a text field representing a <*i*> tag in HTML, to use on the card header; a name for better database readability; and, lastly, a color, that will style the card itself and, if necessary, the elements inside of it. The color can be in whichever CSS readable format, including CSS variables, but naturally, in this last case, the variable value needs to be declared in the stylesheet.

4.1.5 Card_text

Has we stated before in the Subsection 4.1.3, this is a very common type of table. In the current case, this will store the card title and description in all languages present in the platform. The title value is the one presented in the header of the card, and the description is used to explain what the card does when the user is choosing new cards for a template or even as an help text when browsing the report.

4.1.6 Card_type

Card types agglomerate all cards that have a similar way of displaying information. This means, looking back at the example given in Section 3.3, that all cards in the fist row of the Figure 3.1 are of the same type, because they all carry information in the same way. Card types, then, store a name and a description used for database readability, and the dimensions that a card of this type should occupy in the template canvas. The dimensions are represented by two integer values that set the number of rows and columns of the card. The ID of this card types is very important, for it is used on the FrontEnd to render cards in the right way.

4.1.7 Card_type_text

Like the ones before, this table will store the text values associated with card types in all available languages. These values are a title and a description, that the user will see when on the menu for choosing new cards.

4.1.8 Exam_id_card

Like the table we discussed in 4.1.2, the exam_id_card will connect cards to exam_types. In this manner, we can know what cards are available for a given exam type. This will be useful when presenting the user with card choices for him to populate a template.

4.1.9 Option and Option_text

These tables are not in use in the current state of the work, but exist to enable future improvement. The idea behind these is to represent options in a three dot menu that could be situated on the right upper corner of a card. The options would have an icon, like the ones presented in 4.1.4 and a link, to be triggered on click. As for the Option_text, it would store the text of the option displayed on the menu.

4.1.10 Action and Action_text

The action table is another one that ended up not being used in the current project. They should represent the blue button actions that populate the inferior part of the cards, but, during the development we came to the conclusion that it didn’t made sense to store the actions in this way. This table should store a link value, however, there are a lot more types of actions, like enabling zoom, showing labels, or opening another card in full-screen. Because of this, we understood that this table wasn’t versatile enough and that it was better to create the actions based on the *card_id* or the *card_type_id*. Finding a better approach to this problem could be a focus point for future work. In the Action_text we could store a title that appears when the user hovers the mouse on the action button.

4.1.11 Link and Link_text

Like the previous Option and Action tables, the Link table did not get any use in the final work of this project. The initial intention was for this table to store a link that would be launched when clinicians clicked the card itself, however, not only was this feature not found to be useful for almost all cards, but also, since links are dynamic, it made way more sense to call the links using Django URLs. The Link_text would store a title that became visible when the user hovered the mouse on the card itself.

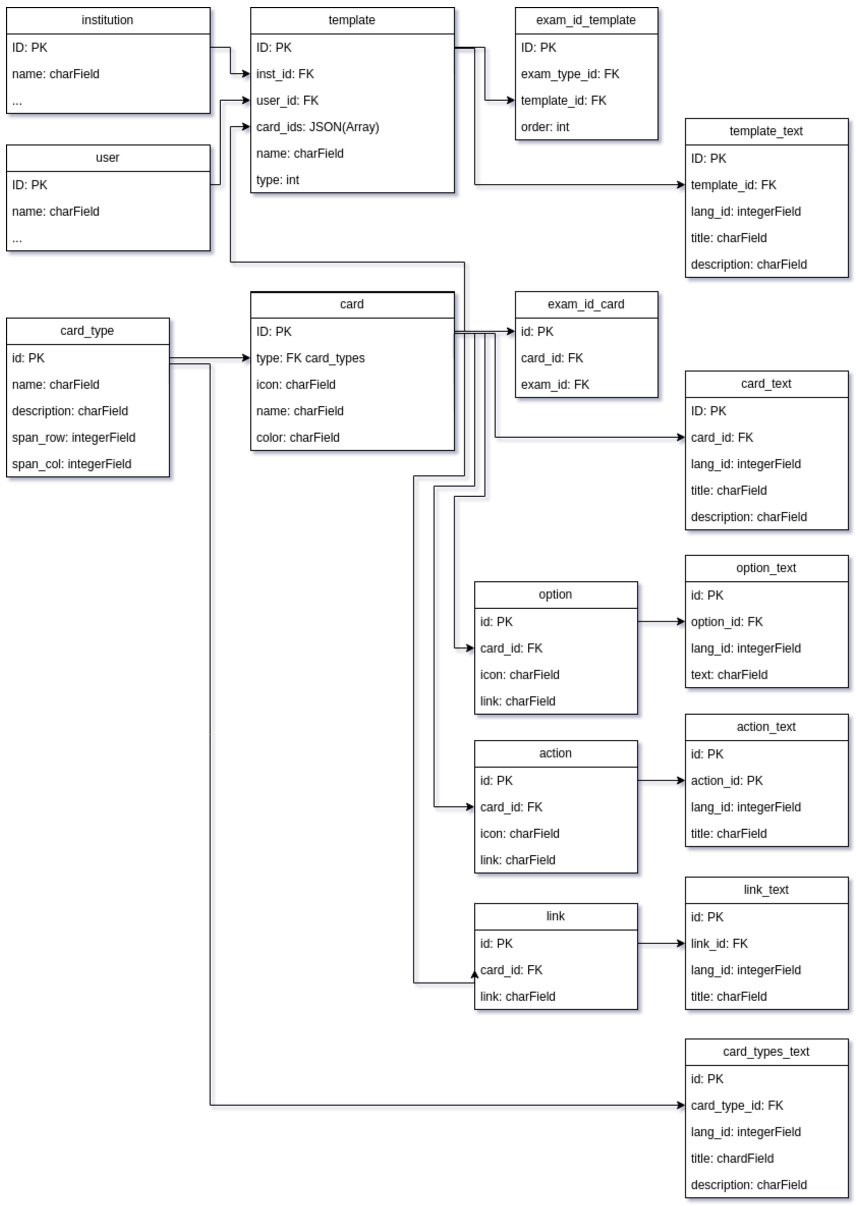


Figure 4.1: Data model used for the current project.

4.2 Template Construction

In the current section we'll cover the construction process of a template when a user requests it. This process, as expected, starts with the URL that leads us to a patient's exam report page. This URL can be in one of two formats, *.../report/{exam_id}* or *.../report/{exam_id}/{template_id}*. In the first case, when no template ID is given, we'll look into the database and get the first template that the user has access to and that is available for the current exam type. The search is firstly made for user created templates, then institution created templates, and only then will we search for the general access ones. The user is then redirected to the found template with a link constructed in the second format.

In the case that the template_id was given on the URL we need to check if the user has access to that specific template and if it is available for the current exam type. If one of these two conditions fails, an exception is raised and a *Permission Denied* page is shown.

After having the ID of the template we want to present, we need the list of cards that compose this template. For each one of the cards we also need to get its metadata, namely the title, icon, color and grid dimensions.

The next step is to collect all data related to the tabs shown in the header of the report, like their text and link. These tabs will be arranged in accordance to the correct order, just like we discussed in 4.1.2. We also need some variables to better control the look of the tabs and emphasize which is the current one being displayed.

Lastly, the only thing left to do is getting all the remaining metadata about the patient and the exam itself, to present in the report header. All of these data is sent to the FrontEnd and we use Jinja2, like stated in the Subsection 2.4.2.3, to make the variables available in the HTML and JavaScript.

4.2.1 Asynchronous Calls

At this point we know how the template and cards metadata is collected and gets sent to the FrontEnd. Is time now to understand how does the clinical data that needs to be displayed on the card body gets to the FrontEnd. This is made via Ajax posts, that perform asynchronous calls to the BackEnd. This is a very important notion in this project, because of the quick response times that we need to achieve, 2.2.2. The clinical data could be fetch while loading the template, alongside all the metadata, however, this would slow down the loading of the page. This happens because all the calculations related to the kinematic data and metrics had to be made before rendering the entire page, creating a really substantial time gap between the user clicking the link that leads him to the report and its rendering. Using one Ajax post per card to get its clinical data, on the other end, enables us to completely render the page before any calculation is made. This will achieve a better user experience, by constructing the new page in phases, starting to populate the template as soon as the user launches the report. The first construction phase is responsible to build the skeleton of the report, without any data inside of the cards, as found in Figure 4.2. During the second phase the cards' body will be rendered. This will happen after the response of each Ajax post, which means that cards will get populated individually and as soon as possible, rather than only after all the calculations are done for every other card.

Making this process work in this fashion, is a more intuitive way to load the report, than to load it all at once, because the user understands at which state he's in, since there is not an idle state, where no feedback is given. That idle time is prone to errors, because it could also led the user to perform multiple clicks on the report link possibly generating unexpected behaviors. In this way, even if the user can't use the report immediately, he can see it getting populated and use the information as it gets rendered on the FrontEnd.

It is interesting to note that in the begging of this project we made this process in three phases, rather than two just presented. The first phase consisted of the loading of the metadata about the patient and the exam and the rendering of the report header and side bar. Then the metadata about each card was fetch with an Ajax post, and only after getting the response was the data of the card fetch, with another Ajax post. This strategy proved to be flawed, since this would generate the cards inside of the template, also in an asynchronous way. That meant that for the cards to remain in the correct order inside of the template, we had to move cards around, which led to a terrible user experience.

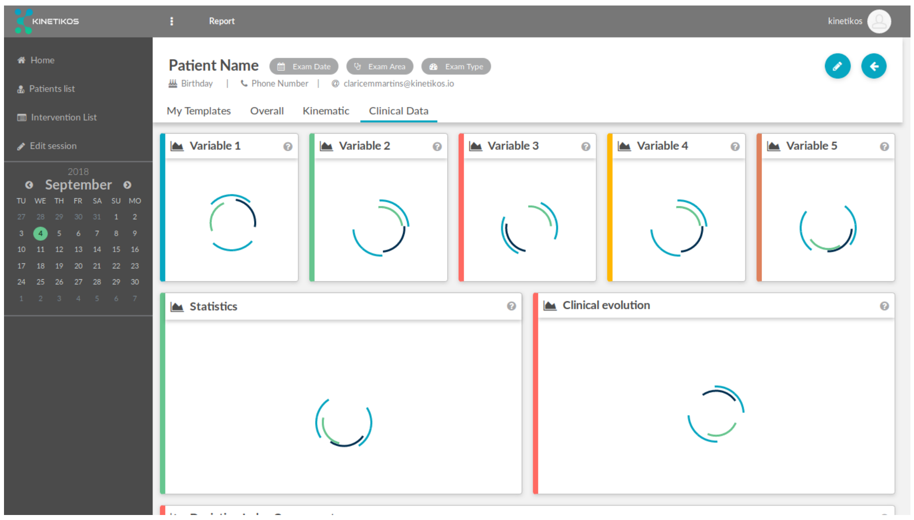


Figure 4.2: Template while waiting for the response of the asynchronous Ajax posts made to get the clinical data.

4.3 Implementation of the Drag and Drop Feature

Since the begging, with the idea of a tailored report, we envisioned some type of drag and drop functionality. The materialization of that feature came in the form of being able to move cards around the template. We know from 3.2 that the template canvas has rows and columns creating a grid, like we can see in Figure 4.3, hence, when moving cards they will span to this grid. In the end, when the user has moved all the cards to his likings he can save the reordered template. The only thing that is needed in the FrontEnd to save the new template is to read all the card IDs into a list in the order that they appear. This list will then be sent to the BackEnd and stored in the database, like described in the Subsection 4.1.1.

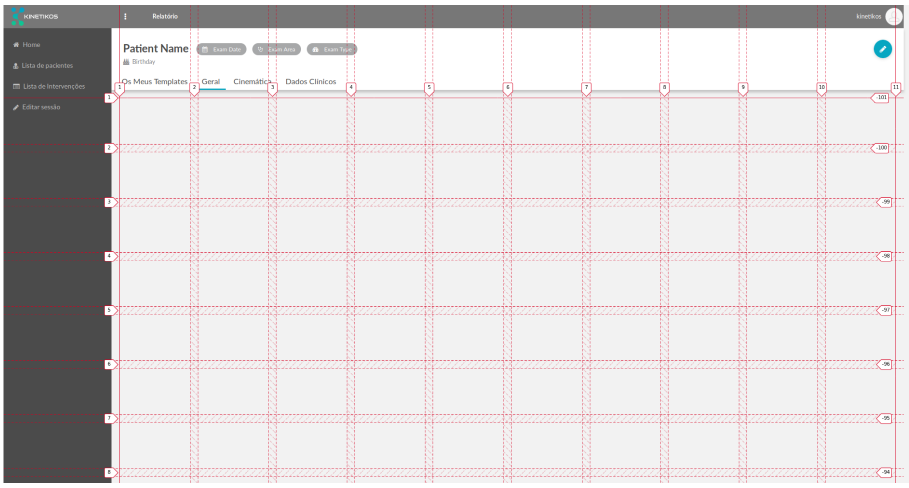


Figure 4.3: Template canvas grid.

The implementation of this feature takes advantage of the HTML5 Drag and Drop API. We start by making all card elements draggable by setting that HTML property to True, and giving them event listeners, for drag start, drag over and drag end. On the drag start phase we do some variable attributions and give a CSS class to the element being dragged, so that it looks more transparent. Next, on the drag over phase, we start by checking if the target of the event is a card, in other words, if the element that is being hovered with the mouse is of the class *card*. If not, we change the target to the closest parent that is of that class, as an example, if we were hovering the header of a card, we change the target value from being the header, to being the card that contains the header. At last, after a few other verifications we insert in the DOM, right before the target, the element that is being dragged. At the drag end we simply remove the class added on drag start to the dragged element to make it look opaque again. It is important to understand that cards are always aligned from left to right and from top to bottom, given the specifications we set on the CSS-Grid and that this will influence the behavior of the moving cards while dragging. In the Figure 4.4 we present an example of a template while one card is being dragged:

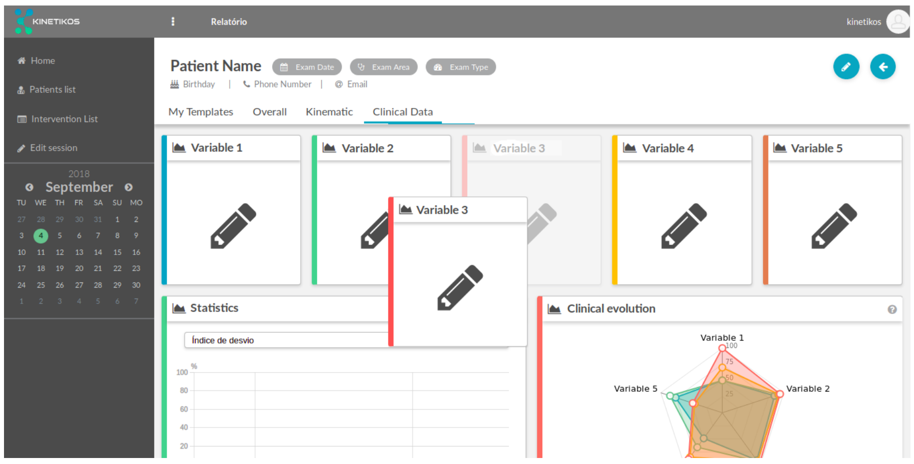


Figure 4.4: Template while card is being dragged.

4.4 Card Construction

Taking into account what we described in 4.2 we can infer that the construction of the cards inside the templates is done in two phases. The first one places the cards in the right order onto the template canvas, makes them occupy the right amount of rows and columns according to the database values, and gives them an header populated with the respective icon and title. By the end of this phase the card body is empty displaying only a loading circle. Next, we enter the phase where the necessary clinical information is fetch from the database and displayed to the end user. In the current section we'll talk a little bit about this last phase.

It all begins with the Ajax post that will try to get from the server the right data for this specific card, taking into account the current exam and patient. It is important to remember that a different Ajax post is made for every single card in the template, like described in the Section 4.2.1. Upon receiving the response, that comes in the form of an object with no universal format, meaning that its structure can change from card type to card type, we call the JavaScript function responsible to populate the cards of the current card type. If the Ajax post, however, returned an error, we call a function that will populate the card's body with an error message. An interesting aspect about the way we managed the function calling for every different type of card is that, instead of using a switch statement, we used an object lookup. As we can see from [25, 26], the switch syntax is one that is harder to read and more prone to errors, so, in order to make more readable and modern code, throughout this whole project we used object lookups instead for this type of situation.

4.4.1 SVG and D3.js Considerations

One of the core aspects of a CIS, as we said in 2.2, was to display clinical data in meaningful and intuitive ways. For this project, most of the displayed data took the form of plots, that were built using the D3.js Framework, 2.4.2.1. By doing so, we were able to manipulate large amounts of data, and create plots in a SVG format. With D3.js, concepts like axis and scales are built into the Framework itself and so have corresponding JavaScript objects. So in order to create an axis for a plot we just need to create the D3.js element with the right arguments and then append it to the SVG, which will translate the axis object, into actual SVG components, like lines. After that, if desired, we can also add some behaviors like zooming or hovering effects to the plots.

One big concern that appeared during this project, non the less, came from the utilization of the SVG standard, since we wanted the plots to be able to render on any screen size and still look good. To achieve this we first needed to get the width and height of the card area where the plot would reside. This, despite being easily achieved, presented itself as a problem due to a small animation present on the cards when the page loads, that expands the card's height from 0% to 100%. This got fixed by using an *animationend* event listener. Having now the actual size of the plot area, we could create the plots with these values as a basis. Hence, in this project we create the plots in a way so that they occupy as much of the card space as possible, meaning that if the screen is taller, the plot will also be taller, not having a fixed size or aspect ratio. By doing this method for the plot construction, we prepared ourselves also for the utilization of the platform on mobile phones, something that was required for this project, like seen on Section 3.1.

Lastly, regarding this concern, we needed only to understand what the behavior of the plot should be if the browser window size changes, when, for instance, it is dragged to a secondary screen. We decided that in this situation the rendered plot should keep its aspect ratio, scaling up or down, in accordance to the new card real estate. This solution is far from perfect in the sense that it doesn't guarantee that the plot will look perfect after scaling, leaving possibly some empty space inside the card, however, this was the only way to keep the interface fast and computationally friendly. The legibility of the card is kept and no time is wasted re-rendering the card, that was our second option. Re-rendering the cards would also completely reset them, making us lose any changes already made by the clinician, like having the zoom set to a certain area or some lines deselected. Non the less, if the page is reloaded in this new window, the plot will be created with the new area in mind, and so, will fill the largest space possible and assume the best aspect ratio for the current conditions.

4.4.2 D3.js Plot Implementation

Looking back at the D3.js framework, we found, however, that for more elaborate types of plots, not everything is built in. A case like this is that of radar plots, like the one displayed in the Figure 4.5. Cards like these were created for this project to be able to display a patient's evolution in various areas of concern in a very simple an easy to read way. We'll now cover the construction of this plot.

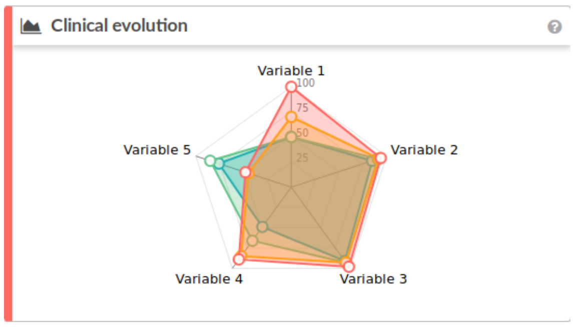


Figure 4.5: Example of a radar plot card type.

As we can see, the radar plot is composed of multiple categories, that are created dynamically, depending on the values coming in the data object fetched in the Ajax post. For each category we need to create a line that will converge with all the others in the center of the plot. We also want to clearly see some divisions to represent the areas between 0%, 25%, 50%, 75% and 100%, so we need to add a line for every step and every category. Looking at the example in the Figure 4.5, we can see that, because we have five categories and four steps, we'll need $5 \times 4 = 20$ lines, plus one line to represent each category, we get 25 lines. These lines will need to be created in D3.js giving them a start and end point, that we'll need to calculate, using the SVG coordinate system. This plot is also composed of polygons, that are the colored areas, and circles, that overlap the polygons on their vertices. These circles are necessary to visually emphasize the relevant values and to allow the plot to have some mouse related behaviors, like hovering over them. Naturally both the circles and the polygon vertices have the same coordinates, that also need to be calculated.

We have just seen that a few points need to be calculated within the coordinate system of a SVG. This system starts in the top left corner of the SVG, having there the point (0,0). From there, the coordinates increase going down and to the right. This contrasts with the coordinate system used in the unitary circle, which will be the base for our calculations. In order to make this conversion we start by creating a group element inside the SVG. This element has its own coordinate system, that will be the one we'll use for all the calculations. Then we make the group have the desired diameter of the plot and translate it so that it gets centered on the canvas. After that we just need to apply the following relation: $x_{svg} = (x_{unity\_circle} + 1) \times radius$ and $y_{svg} = (1 - y_{unity\_circle}) \times radius$. This will allow us to get the coordinates of any point in the group coordinate system, by applying the corresponding relation to any point calculated in the unitary circle coordinate system. Figure 4.6 aims to clarify how we came to this conclusions:

After finding the relations just presented we went on to calculate all the necessary points.

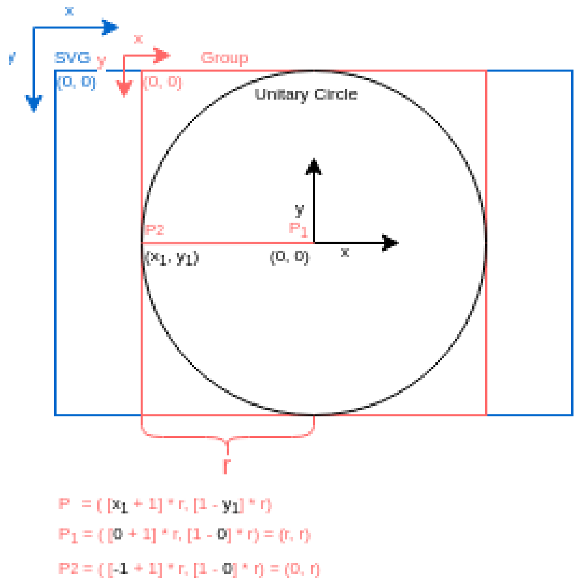

x
y
SVG
(0, 0)
x
y
Group
(0, 0)
Unitary Circle
y
P2
P1
(x1, y1)
(0, 0)
x
r
P = ([x1 + 1] * r, [1 - y1] * r)
P1 = ([0 + 1] * r, [1 - 0] * r) = (r, r)
P2 = ([-1 + 1] * r, [1 - 0] * r) = (0, r)

Figure 4.6: Radar Plot coordinate system.

Regarding the category lines, that start at the center of the plot, their formulas will be:

$$x_1 = y_1 = radius \tag{4.1}$$

$$x_2 = \left[\sin\left(2 \times \pi \times \frac{category\_index}{number\_of\_categories}\right) + 1\right] \times radius \tag{4.2}$$

$$y_2 = \left[1 - \cos\left(2 \times \pi \times \frac{category\_index}{number\_of\_categories}\right)\right] \times radius \tag{4.3}$$

From this formulas we can retrieve some more useful information, because we are using sine where we would expect cosine. This happens because the unitary circle angles go in a counterclockwise way starting in the x axis, but what we want is for the categories to start in the y axis and go around the plot in a clockwise manner. Regarding the category index it is important to state that it must start at zero.
We'll present now the formulas for the step lines, that connect the category lines. We need to create one for each level and for each category.

$$x_1 = \left[\sin\left(2 \times \pi \times \frac{category\_index}{number\_of\_categories}\right) \times \frac{step}{number\_of\_steps} + 1\right] \times radius \tag{4.4}$$

$$y_1 = \left[1 - \cos\left(2 \times \pi \times \frac{category\_index}{number\_of\_categories}\right) \times \frac{step}{number\_of\_steps}\right] \times radius \tag{4.5}$$

$$x_2 = \left[\sin\left(2 \times \pi \times \frac{category\_index + 1}{number\_of\_categories}\right) \times \frac{step}{number\_of\_steps} + 1\right] \times radius \tag{4.6}$$

$$y_2 = \left[1 - \cos\left(2 \times \pi \times \frac{category\_index + 1}{number\_of\_categories}\right) \times \frac{step}{number\_of\_steps}\right] \times radius \tag{4.7}$$

The new element that is present in these formulas is the fraction between the step, that starts at one because we don't need any lines at 0%, and the number of steps. This division will give us a value as a percentage that corresponds to the index of the step. The number of steps, in the case of Figure 4.5 will be four. We can also notice that the end of any line will be the start of the next one.
Lastly, we only need the formulas to know the coordinates of a point from which we have the category and the correspondent value as a percentage. The formulas are as follow:

$$x_1 = \left[\sin\left(2 \times \pi \times \frac{category\_index}{number\_of\_categories}\right) \times \frac{value}{max\_value} + 1\right] \times radius \tag{4.8}$$

$$y_1 = \left[1 - \cos\left(2 \times \pi \times \frac{category\_index}{number\_of\_categories}\right) \times \frac{value}{max\_value}\right] \times radius \tag{4.9}$$

This time, we see that the division between the step and number of steps, as found in 4.4, was replaced by the division of the current value by the maximum possible value. In the case of percentages the maximum possible value will be one hundred. These two formulas will then be used for both the polygons vertices and the correspondent circles. To build this card we only need now to call the correct D3.js functions, in order to create every single element needed, with respect to the formulas just presented.

4.4.3 Developed Cards

In this section we'll show and discuss briefly all the card types that were created during this project.

- Interactive Line Plot - This type of card presents a multiple line plot, that is interactive in the sense that: the lines can be hidden; the user can zoom in and out as well as drag the visible area around; there is the possibility to see with detail the values of each line through time; and even add a new multiple line plot on top of the current one to compare the two. These features were discussed in Subsection 2.3.1, where we explored the commonly used way to display gait cycles.

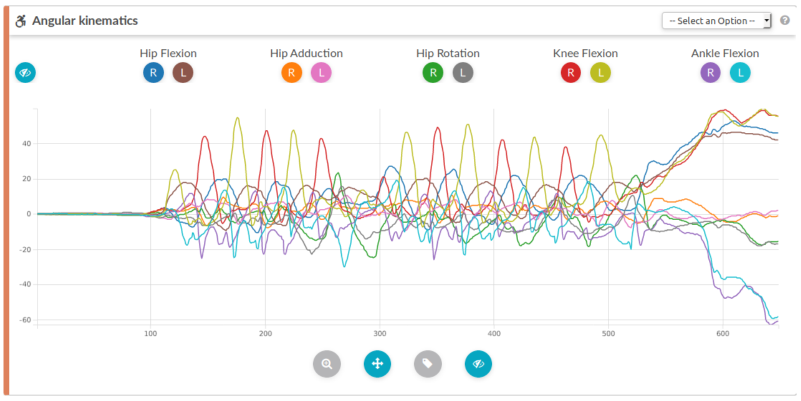


Figure 4.7: Example of an Interactive Line Plot type of Card.

- Events Plot - This is a very important card. In the displayed plot clinicians will see the kinematic events that Kinetikos' algorithms detected from a recorded patient movement. In the Figure 4.8 we can find gait events, both for the left and right leg. Clinicians can then edit these events by zooming and dragging the lines, in the case that they disagree with the detected limits.

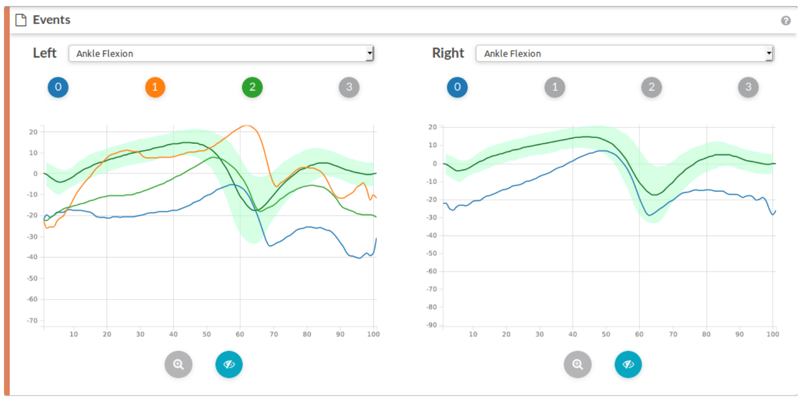


Figure 4.8: Example of an Events Plot type of Card.

- Horizontal Bar Chart - Cards of this type display an horizontal bar chart of percentage values. These are very visually interesting cards, condensing a lot of information and presenting it in a truly simple and intuitive way. The first line, which has a different color, is a score, that was calculated from all the other values presented.

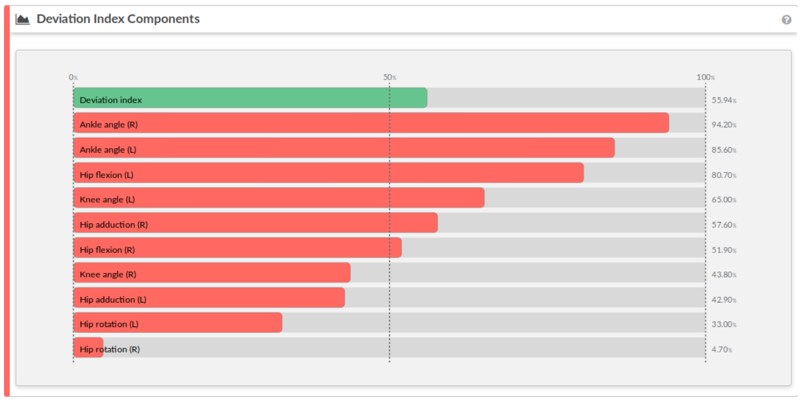
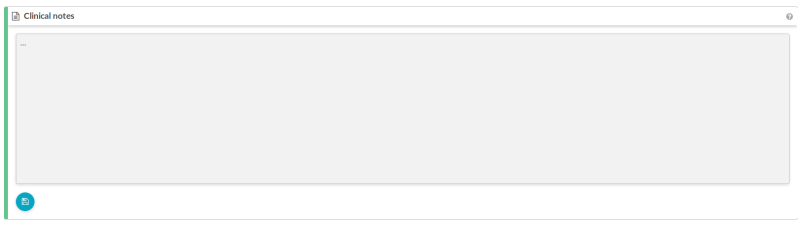


Figure 4.9: Example of an Horizontal Bar Chart type of Card.

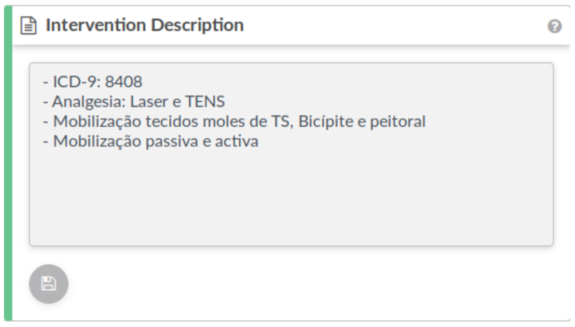
- Editable Text Area - With this type of card the user can write, store and edit text notes. This card is useful, because clinicians are more prone to fill free text fields than structured input methods [2], despite the fact that it's easier, in an information system, to assess clinical insights from a structured input method.

- Small Editable Text Area - Just like the previous card, this one enables the clinician to write, store and edit text notes.


Clinical notes

...

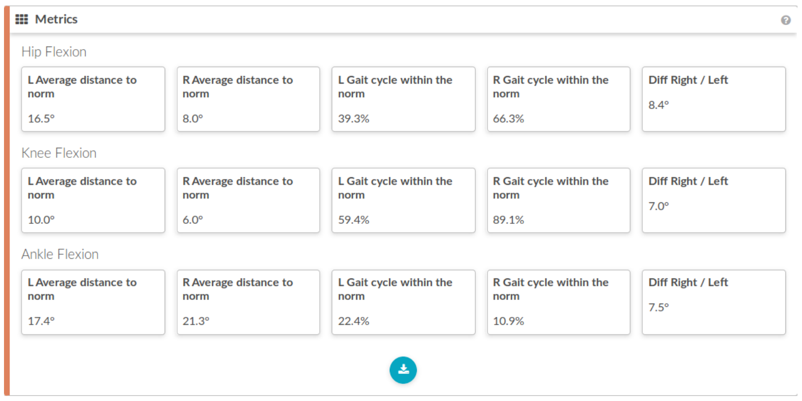
Figure 4.10: Example of an Large Editable Text Area type of Card.


Intervention Description

- ICD-9: 8408
- Analgesia: Laser e TENS
- Mobilização tecidos moles de TS, Bicípite e peitoral
- Mobilização passiva e activa

Figure 4.11: Example of an Small Editable Text Area type of Card.

- Metrics - Within this card, the users will find an overall collection of metrics and data. It can be a good way to find a single value of interest or have a general idea about the patient's state.

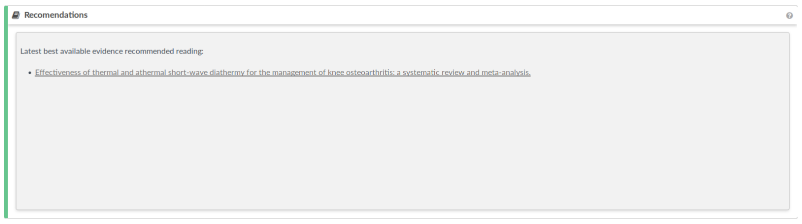

Metrics

	L Average distance to norm	R Average distance to norm	L Gait cycle within the norm	R Gait cycle within the norm	Diff Right / Left
Hip Flexion	16.5°	8.0°	39.3%	66.3%	8.4°
Knee Flexion	10.0°	6.0°	59.4%	89.1%	7.0°
Ankle Flexion	17.4°	21.3°	22.4%	10.9%	7.5°

Figure 4.12: Example of a Metrics type of Card.

- Text Output - Like the card on Figure 4.13 this is a text output card, so it displays text that cannot be edited. In the current example it displays some literature recommendations for the clinician.

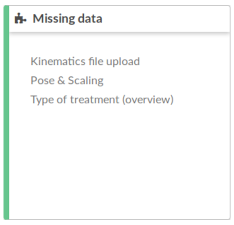
- Small Text Output - This type of card is very similar to that found on Figure 4.14, but in a smaller form factor. The example displayed gives the user some information and links about which patient's data is yet to be filled. This is a good way to support the clinical decision by pointing out what crucial information is still missing for a good clinical judgment.


Recomendations

Latest best available evidence recommended reading:

- Effectiveness of thermal and athermal short-wave diathermy for the management of knee osteoarthritis: a systematic review and meta-analysis.

Figure 4.13: Example of a Text Output type of Card.


Missing data

Kinematics file upload
Pose & Scaling
Type of treatment (overview)

Figure 4.14: Example of a Small Text Output type of Card.

- MPLD3 - This card takes advantage of a very powerful framework built upon D3.js, called mpld3 that enables the creation of a D3.js plot from the server side, using python's Matplotlib library. Having this type of card is a great advantage, because it makes possible the creation of a completely new card in terms of information and the way it is displayed, in a quick way, hence it frees the developers from writing code in D3.js that, despite being a powerful framework, is pretty extensive and produces code that is not easily readable. Another advantage, is that, since all the metrics are calculated using python, given that that's the language of the BackEnd of the platform, the plots in Matplotlib, in most cases, were already being created in the research phase.

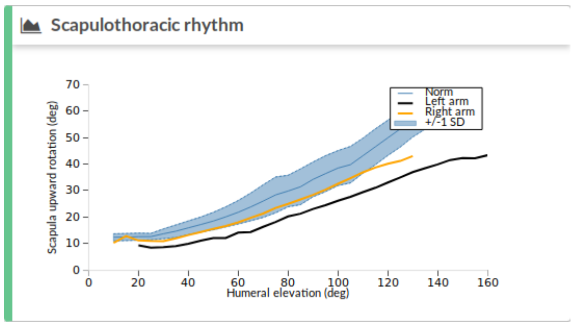


Figure 4.15: Example of a MPLD3 type of Card.

- Radar Plot - Cards like this display a radar plot. This type of plot was already an idea in the initial drafts, as seen in Figure 3.8b, because of its familiar visuals and intuitive way to retrieve data with just a quick look. With it, a clinician can already understand if a patient has been improving, if the improvements were big or small, and what areas are still a concern.

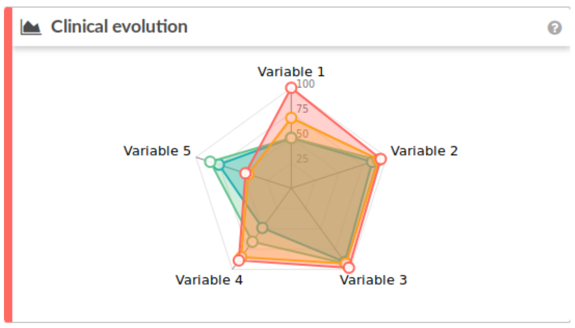


Figure 4.16: Example of a Radar Plot type of Card.

- Percentage Circle - This type of card is one of the most used, since it is really simple and visually interesting. It displays a percentage with a colored circle, which size is proportional to the actual value.

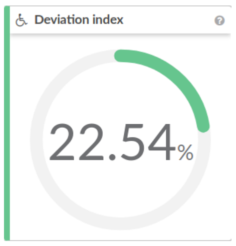


Figure 4.17: Example of a Percentage Circle type of Card.

- Single Value - With this type of card we can display any single value, like age, angles, percentages, or any other thing that might be useful. The card will always display just the value, without any other type of visual indicator.

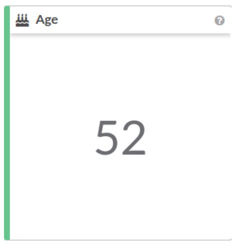


Figure 4.18: Example of a Single Value type of Card.

- Single Line Plot - This plot displays a temporal line of any metric. We can get an idea of the patient evolution throughout the treatments. When there is no data about the displayed metric from one of the exams an *X* is placed.

- Single Line Plot With Dropdown Menu - Like the card just presented, this type of cards will present a single line plot, with the temporal evolution of a specific metric. However, in this case, the user can also change the displayed metric, by selecting a new one from the dropdown menu.

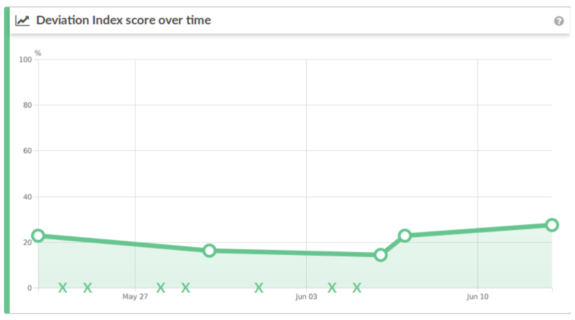


Figure 4.19: Example of a Single Line Plot type of Card.

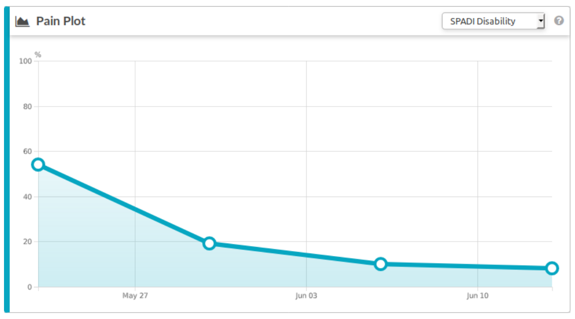


Figure 4.20: Example of a Single Line Plot With Dropdown Menu type of Card.

- Video - In this card we can see a video representation of the patient movement, that was recorded during the exam using *Kinetikos'* proprietary algorithms for human motion reconstruction and visualization.

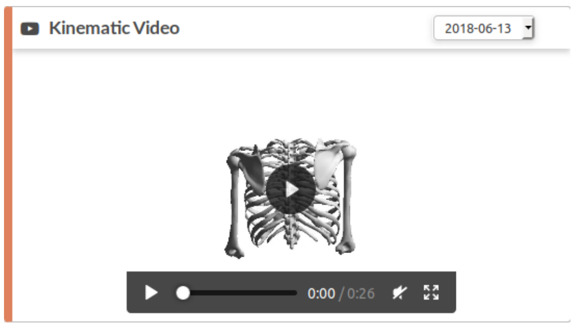


Figure 4.21: Example of a Video type of Card.

- Link - These cards are merely a link to some other area of the platform. In Figure 4.22 we can see a card that links to the page responsible for adding new sessions with the current patient.

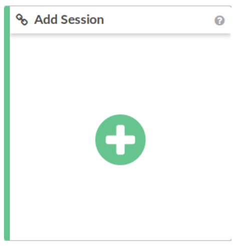


Figure 4.22: Example of a Link type of Card.

- Files Upload - This cards allow the user to upload some files to the platform. In the example of Figure 4.23, the user can upload the files that the sensors generated, where the user movement is stored.

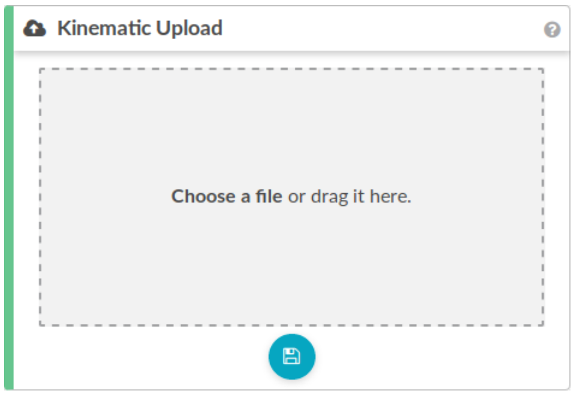


Figure 4.23: Example of a Files Upload type of Card.

- Metrics Plot - Cards like this will display metric values, in the form of deviation indexes or percentages. Each metric can have a normal area and either a single value or the left and right side values.

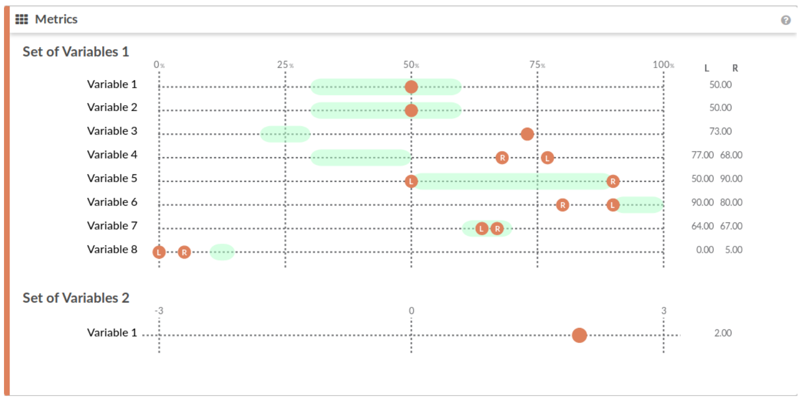


Figure 4.24: Example of a Metrics Plot type of Card.

Chapter 5

Discussion and Results

In this chapter we'll take an overview of the produced work and try to evaluate its success, discussing all of its main aspects and features, as well as their relevance.

5.1 Report Overview

During Chapter 4 we've looked at the pieces that, when put together, compose the report. This report by itself is the ultimate result of this work. However, we're still missing an overhaul look at the whole page and how a clinician can make use of it during his practice. Thereby, in this first section we'll present a possible use case for the developed web page. It's important to understand that the names and data presented are simulated and don't correspond to a real patient. The used report templates are also hypothetical, as well as their names and organization. This example is presented merely for the purpose of displaying the result of this work in action, and how each of the created cards can be used to convey knowledge and information about a patient and by that influence a clinician's workflow.

5.1.1 Patient Admission

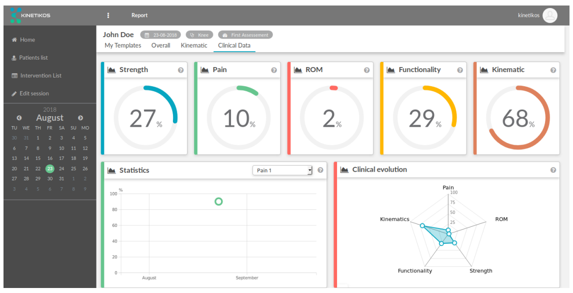
In a session with a patient, clinicians will do some examinations and ask a few questions to the patient regarding his condition. With this data and *Kinetikos* insights, some scores are calculated and they start to populate the report. We can see in Figure 5.1a a first set of clinical data, accessible by clicking on the *Clinical Data* tab. Since this is the first session, there is no data to compare this session against, so we'll focus for now on the first line of circular plots. From these plots, we can infer, for example, that the patient has low Range Of Motion (ROM), has lost more than 50% of his functionality, which is related to day to day activities, and has a kinematic score, calculated with *Kinetikos*' technology, bellow

70%. This can indicate that the patient has lost some of his ability to move. With this type of scores, in the next sessions we'll be able to know, with a greater degree of accuracy, if the patient is improving and by which amount.

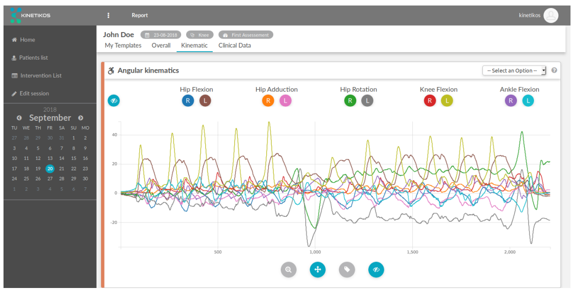
Following, we have Figure 5.1b, representing the *Kinematic* tab, where we can see the angular kinematic data collected with the sensors. At first glance this plot is a little bit too crowded, because it encompasses all the collected data, displaying multiple gait cycles, however by zooming and deselecting some of the lines, displayed in Figure 5.1c, the clinician can take a look at some specific area of interest.

By scrolling down on this tab we reach the events plots, 5.1d. This is one of the best ways to understand where the patient movement is abnormal. In these plots we see the normal area, marked with green, and the multiple gait cycles performed by the patient. These cycles are retrieved from the complete sensor data with an algorithm built by *Kinetikos*. In this case we can clearly see that our patient has an abnormal right knee flexion, being almost unable to bent his leg.

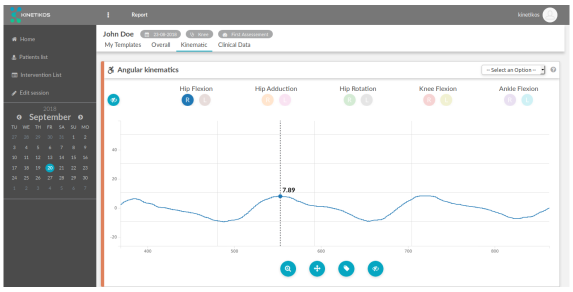
The clinician could also need to store some notes about the treatments performed, his recommendations to the patient or some other thing. This can be done in the *Overall* tab, Figure 5.2, where there is an editable text area card for writing clinical notes.



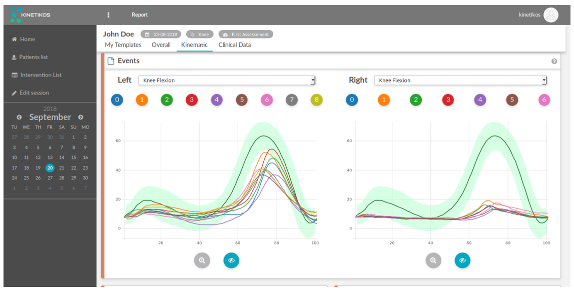
(a) Example of an hypothetical *Clinical Data* report tab.



(b) Example of an hypothetical *Kinematic* report tab.

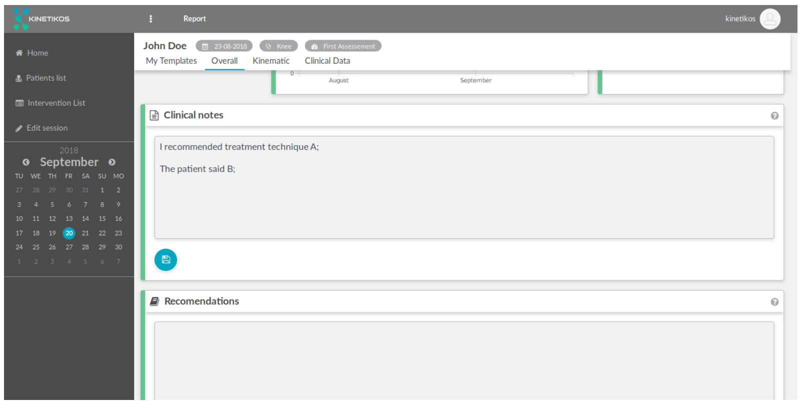


(c) Example of an hypothetical *Kinematic* report tab, with the Hip Flexion line highlighted.



(d) Example of an hypothetical *Kinematic* report tab.

Figure 5.1: Multiple report views.



I recommended treatment technique A;

The patient said B;

Figure 5.2: Example of an hypothetical *Overall* report tab.

5.1.2 Follow Up Sessions

Follow up sessions are all the treatment sessions until the patient is dismissed. For the purposes of this simulation, there will be only one follow up session.

Since we now have more than one session, we can compare data and see the evolution of the patient. One of the ways to see this evolution is through a radar plot, like we presented in Subsection 4.4.3. We can find this plot, entitled *Clinical evolution*, in the *Clinical Data* tab of the report, as displayed in Figure 5.3. From this plot we can see, for example that the Pain score suffered the biggest improvement but that the ROM is almost the same as in the previous session, still needing much improvement.

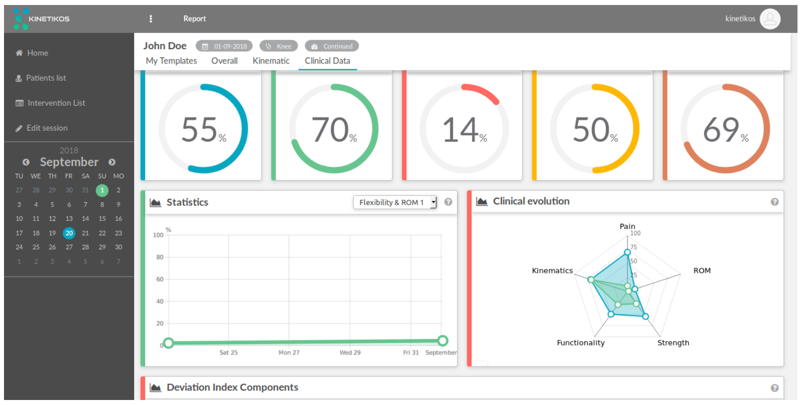


Figure 5.3: Example of an hypothetical *Clinical Data* report tab.

On the same idea of comparing the two sessions, we can go to the *Kinematic* tab and find two video cards, Figure 5.4. Since we have a video for each session and these cards have the ability to change between them, we can have both the video for current session and for the previous one side by side for comparison. This way of comparing the sessions is what Figure 5.4 tries to demonstrate. In this Figure we can see that in the admission session (on the left video card, dated for 2018-08-23) the patient almost didn't bent the right leg, but now, in the current session (on the right video card, dated for 2018-09-01), he can. This improvement can also be inferred by looking at the current session's *Events* card, Figure 5.5, which shows that the right knee flexion of the patient is now much closer to the norm.

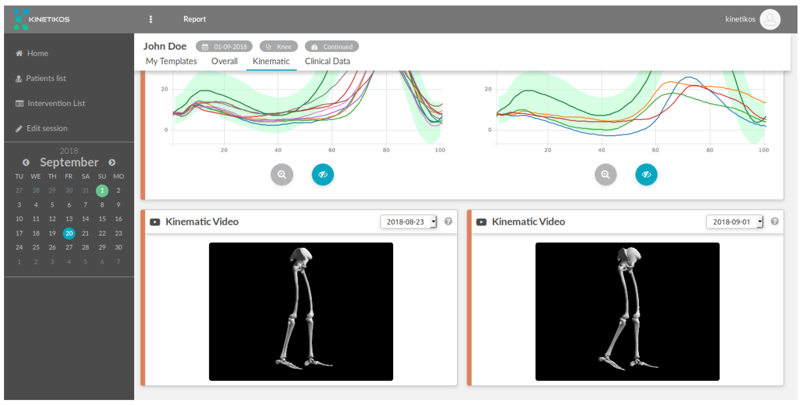


Figure 5.4: Example of an hypothetical *Kinematic* report tab.

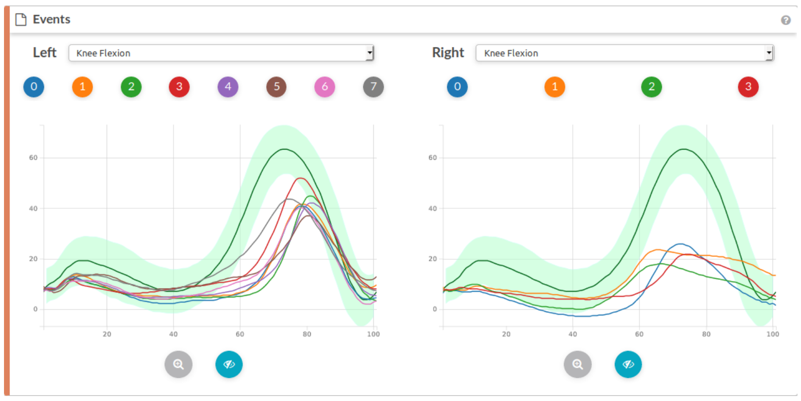


Figure 5.5: Example an *Events* card.

5.1.3 Discharge

By using this report page, a clinician can support his decision to discharge a patient based on the patient's scores and evolution. In Figure 5.6 we can see once again the circular plots as well as the radar one. Based on them we can assess that the patient is showing real improvements and is, probably, in condition to be discharged, since almost all of his scores are above 90%.

It might be of interest also, to show to the patient himself is progression throughout the treatments. This can be done, for example, through the *Angular kinematics* plot, displayed in Figure 5.7, using it's ability to display the data of two sessions. In the Figure we can see the knee flexion angles throughout multiple gait cycles for both the admission and the dismiss sessions. It is clearly visible that the admission session line, represented in orange, displays way less flexion than the line of the discharge session, represented in red. By just looking at this the patient can get a real sense of how much he improved and possibly feel reassured about the treatment process.

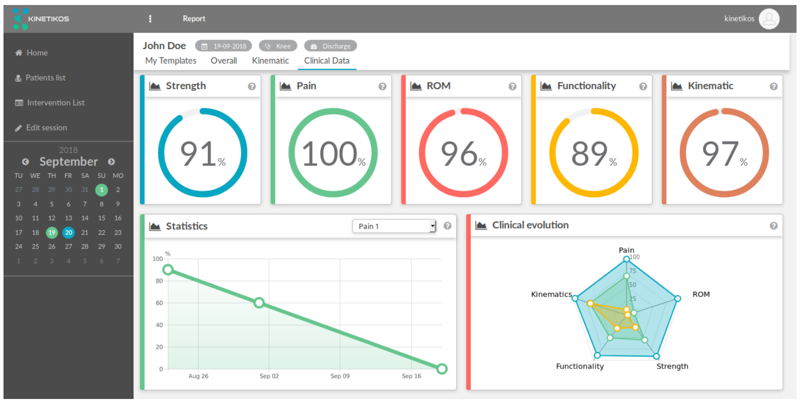


Figure 5.6: Example of an hypothetical *Clinical Data* report tab.

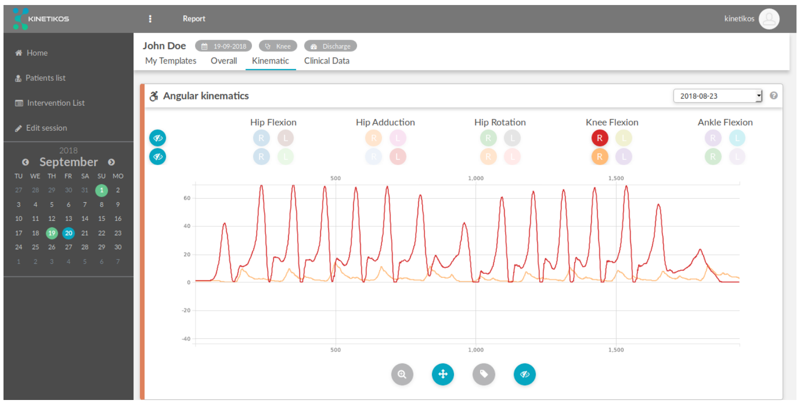


Figure 5.7: Example of an hypothetical *Kinematic* report tab, with knee flexion highlighted for both the admission(orange) and discharged(red) sessions.

5.2 Discussion

After the conclusion of this work we need to look back at what we set up to do in the beginning, and what factors would be used to judge the achieved results. Let's walk through them.

First of all we needed to respect the clinicians workflow. As described in the Section 3.2, we achieved this by enabling clinicians to create their own report area, with the information that they feel is useful and respectful of their workflow for each situation. They need only to create templates that can also be reused for different types of exams. We believe that with templates, this CIS will have an advantage over the ones that have restrict reports, which may or may not reflect the desired workflow and logical thinking behind the clinicians decision making.

Right after this concern, we needed to make sure that only the right information was presented to the clinicians, and that that information was shown in an intuitive way. Regarding the aspect of showing only the right information, we can also say that we succeeded because of the concept of a customizable report, since users can dismiss or add to a template, only the cards that they want. A better way to achieve this concern would be to add or remove cards automatically from the template, with the help of an intelligence responsible to find which cards are relevant for the patient's current state. We can say though, that this work is ready to implement such a feature, since the displayed cards on a given template come from a list of *card_id*'s that is passed to the FrontEnd as a JavaScript array. Because of this, upon having the intelligence to create the list of relevant cards, the only thing needed to do, is to call the construction of the report with that list.
Regarding the intuitiveness and clear understanding of the displayed information, we can't say for sure if the project succeeded. This happens because design and intuitiveness are subjective matters and, even though we had these concerns in mind, there is most certainly room for improvement. However, thanks to the division of the information into cards, updating them or creating new ones is possible, without having to change the entire report, making it a more simple and quick process. Even more, the fact that information is divided into cards, means that we can even have different cards to display the exact same information but in a different way, giving clinicians the power to choose which one they feel more comfortable with.

In the Subsection 2.2.2 we also said that a CIS should be easy to update. We think we succeeded in this concern, again, and like said in the previous paragraph, because of the division of the report into small cards, and because of the customizable templates. If new types of data or metrics are found in the future, *Kinetikos* only needs to create new cards to hold them, without compromising any part of what has already been developed. Also, if the user or *Kinetikos* intends to change the order of the cards in a template, or which cards appear, they can do so without having to write any code.

Lastly, we talked a lot about the need to have small response times while using a CIS. The successful implementation of this concern is hard to assess, since it would require a lot of testing and resources that were not available for this dissertation. Also, since this is a web-based solution, the results will always be dependent of the quality of the Internet connection used during the tests. However, this concern was definitely present in our development phase and we took some measures, not only to make the loading of the page faster and the information on each individual card available as soon as possible, but also, we made sure that the user always received feedback from the system, so that this process occurred in the most frustration free and seamless way possible. This discussion can be found on Subsection 4.2.1.

By all of this, we can say, with some confidence, that this work was a success. This is not only because we were able to build a solution that respected the technological concerns we had, like the ability to display time-series and to customize the look and information present on the report of a CIS, but also, because we believe that the solution created respected all of the major concerns and needs presented on Section 2.2, as we discussed throughout this chapter.

A better way to assess the quality of the produced work would be to inquire some clinicians on their thoughts about the report page after they'd spend some time using it. As a future work, we'd like to present the System Usability Scale inquiry to some clinicians, because this scale is a recognized international standard for system usability, which already has an European Portuguese version. [20] By doing so we would be able to understand the impact of this work and which areas are well implemented or deserve improvements.

Chapter 6

Conclusions and Future Work

By the end of this work we can say that we were able to achieve our main goals, actually being able to build a tailored web-based interface to display and analyze time series, that is integrated in a CIS targeted at motion skill disorders. As shown in Chapter 5, we built a report area that respected, at least to some degree, all the major areas of concern, discussed on chapter 2, more specifically in section 2.2.2.

We believe that the developed solution will present itself as a valuable asset for clinicians, since that, by using it, they can now create a report that respects their workflow and that can be updated at their own will. The fact that information is divided into smaller parts, that we named cards, means that clinicians can add or remove information from the report. We also think that this division of information into smaller parts will make the report easier to read and understand in a timely fashion.

Regarding Kinetikos' expectations, they've been meet and this work is expected to be integrated into the company's platform by the end of September, 2018. One big technological request that they had was that the report needed to work on a mobile browser and that also has been achieved.

However some work can still be done to make this an even more valuable solution. First of all we could think of a way to integrate the card actions, represented by the blue buttons on the bottom of some cards, into the data base, like we said in Subsection 4.1.10. By doing so, the project complexity would get smaller, because this information would not be on the code itself and also because it would then be possible to edit it directly in the database.

In the future, users should be able to create their own cards, possibly by choosing a type of plot and the metric they want to display. This cards could be made using the mpld3 framework, already integrated into the report. We think this would represent a considerable leap forward in the direction of a completely free and editable report.

We said also in Chapter 5 that in the future the CIS could also have the capability to choose which cards to show on a template based on the patient's state. We explained that this would be possible to integrate with the report as it is by the end of this work, since each template is just a list of *card_id*'s that, in this case, would only need to come from the intelligence responsible for choosing the cards, instead of the database.

Lastly, the report page could also be made slightly more intuitive. One aspect that could improve this is the use of custom icons, that better represent the functionalities and information present in each card or button. Also we could make some small adjustments in the user interactions with the report in order for them to be more obvious. This changes would imply, however, some further investigation regarding user experience.

Bibliography

[1] A. Ahmed, S. Chandra, V. Herasevich, O. Gajic, and B. W. Pickering. "The effect of two different electronic health record user interfaces on intensive care provider task load, errors of cognition, and performance*." In: *Critical Care Medicine* 39.7 (2011), pp. 1626–1634. issn: 0090-3493. doi: 10.1097/CCM.0b013e31821858a0. url: http://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage{\&}an=00003246-201107000-00004.

[2] D. W. Bates, G. J. Kuperman, S. Wang, T. Gandhi, A. Kittler, L. Volk, C. Spurr, R. Khorasani, M. Tanasijevic, and B. Middleton. "Ten Commandments for Effective Clinical Decision Support: Making the Practice of Evidence-based Medicine a Reality." In: *Journal of the American Medical Informatics Association* 10.6 (2003), pp. 523–530. issn: 10675027. doi: 10.1197/jamia.M1370. url: http://www.ncbi.nlm.nih.gov/pubmed/12925543http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC264429.

[3] *Biohealthmatics.com - Clinical Information Systems.* 2006. url: http://www.biohealthmatics.com/technologies/his/cis.aspx (visited on 01/08/2018).

[4] M. Bostock, V. Ogievetsky, and J. Heer. "D3 data-driven documents." In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2301–2309. issn: 10772626. doi: 10.1109/TVCG.2011.185. arXiv: 1408.4626.

[5] W. F. Bria and M. M. Shabot. *The electronic medical record, safety, and critical care.* 2005. doi: 10.1016/j.ccc.2004.08.001.

[6] Campbell and R. James. "The Five Rights of Clinical Decision Support: CDS Tools Helpful for Meeting Meaningful Use." In: *Journal of AHIMA* 84.10 (2013), 42–47 (web version updated February 2016). issn: 10605487. url: http://library.ahima.org/doc?oid=300027{\#}.W4E0SS2ZN24.

[7] S. Chen, J. Lach, B. Lo, and G. Z. Yang. *Toward Pervasive Gait Analysis With Wearable Sensors: A Systematic Review.* 2016. doi: 10.1109/JBHI.2016.2608720. url: http://ieeexplore.ieee.org/document/7574303/.

[8] J. Clark. *A list apart ALA : for people who make websites.* 2006. url: http://alistapart.com/article/the-story-of-css-grid-from-its-creatorshttps://alistapart.com/article/tohellwithwcag2 (visited on 02/07/2018).

[9] W. Cools, K. D. Martelaer, C. Samaey, and C. Andries. "Movement skill assessment of typically developing preschool children: a review of seven movement skill assessment tools." In: *Journal of sports science & medicine* 8.2 (2009), pp. 154–68. ISSN: 1303-2968. DOI: 10.1016/S0031-9406(05)66164-0. URL: http://www.ncbi.nlm.nih.gov/pubmed/24149522http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3761481http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3761481{\&}tool=pmcentrez{\&}rendertype=abstract.

[10] CSS Working Group. *CSS Grid Layout Module Level 1*. 2013. URL: https://www.w3.org/TR/2017/CR-css-grid-1-20171214/http://www.w3.org/TR/css3-grid-layout/.

[11] *Django Overview - The Django Book*. URL: https://djangobook.com/tutorials/why-django/https://djangobook.com/tutorials/django-overview/ (visited on 02/12/2018).

[12] Django Software Foundation. *The Web Framework for perfectionists with deadline*. 2014. URL: https://www.djangoproject.com/ (visited on 02/12/2018).

[13] L. Elizabeth. *Should You Develop a Desktop or Web App?* 2015. URL: https://www.sitepoint.com/web-desktop-apps/ (visited on 02/06/2018).

[14] *Gait - Physiopedia*. URL: https://www.physio-pedia.com/Gait (visited on 02/09/2018).

[15] C. Glackin, C. Salge, D. Polani, M. Tuttemann, C. Vogel, C. R. Guerrero, V. Grosu, S. Grosu, A. Olensek, M. Zadravec, I. Cikajlo, Z. Matjacic, A. Leu, and D. Ristic-Durrant. "Learning gait by therapist demonstration for natural-like walking with the CORBYS powered orthosis." In: *IEEE International Conference on Intelligent Robots and Systems* 2015-Decem.September 2015 (2015), pp. 5605–5610. ISSN: 21530866. DOI: 10.1109/IROS.2015.7354172.

[16] S. J. Shultz, P. A. Houglum, and D. H. Perrin. *Examination of Musculoskeletal Injuries*. 2005. URL: http://books.google.com/books?id=-2RL0p3wClkC{\&}printsec=frontcover{\%}5Cnpapers://c25f1780-76f5-45ee-9877-1c6029fe81a5/Paper/p3158.

[17] S. Jiang, X. Wang, M. Kyrarini, and A. Gräser. "A Robust Algorithm for Gait Cycle Segmentation." Bremen, Germany, 2017.

[18] K. Kawamoto. "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success." In: *BMJ* 330.7494 (2005), pp. 765–0. ISSN: 0959-8138. DOI: 10.1136/bmj.38398.500764.8F. URL: http://www.ncbi.nlm.nih.gov/pubmed/15767266http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC555881http://www.bmj.com/cgi/doi/10.1136/bmj.38398.500764.8F.

[19] J. K. J. K. Loudon, M. Swift, and S. Bell. *The clinical orthopedic assessment guide.* 2008. URL: http://search.ebscohost.com/login.aspx?direct=true{\&}db=congale{\&}AN=edsgcl.193189151{\&}site=eds-live.

[20] A. I. Martins, A. F. Rosa, A. Queirós, A. Silva, and N. P. Rocha. "European Portuguese Validation of the System Usability Scale (SUS)." In: *Procedia Computer Science*. Vol. 67. Elsevier, 2015, pp. 293–300. ISBN: 978-9-8998-4345-5. DOI: 10.1016/j.procs.2015.09.273. URL: https://www.sciencedirect.com/science/article/pii/S1877050915031191?via{\%}3Dihub.

[21] C. Mason and T. Leong. "Clinical information systems in the intensive care unit." In: *Anaesthesia and Intensive Care Medicine* 17.1 (2016), pp. 13–16. ISSN: 18787584. DOI: 10.1016/j.mpaic.2015.10.011. URL: http://dx.doi.org/10.1016/j.mpaic.2015.10.011.

[22] NIDDKD. *The A1C Test & Diabetes*. 2017. URL: https://www.niddk.nih.gov/health-information/diabetes/overview/tests-diagnosis/a1c-test (visited on 02/05/2018).

[23] Physiotutors. *Gait Cycle & Gait Analysis - YouTube*. 2015. URL: https://www.youtube.com/watch?v=1u6d1CX7o9c (visited on 02/09/2018).

[24] E. A. Pifer and D. F. Sittig. *Improving Outcomes with Clinical Decision Support: An Implementer's Guide*. HIMSS, 2005, p. 323. ISBN: 9780984457731. URL: https://www.crcpress.com/Improving-Outcomes-with-Clinical-Decision-Support-An-Implementers-Guide/Osheroff-Teich-Levick-Saldana-Velasco-Sittig-Rogers-Jenders/p/book/9780984457731.

[25] *Replacing switch statements with Object literals*. URL: https://toddmotto.com/deprecating-the-switch-statement-for-object-literals/ (visited on 08/24/2018).

[26] *Rewriting Javascript : Replacing the Switch Statement*. URL: https://medium.com/chrisburgin/rewriting-javascript-replacing-the-switch-statement-cfff707cf045 (visited on 08/24/2018).

[27] A. Ronacher. *Jinja2 (The Python Template Engine)*. 2014. URL: http://jinja.pocoo.org/ (visited on 02/12/2018).

[28] M. Serrao, F. Pierelli, A. Ranavolo, F. Draicchio, C. Conte, R. Don, R. Di Fabio, M. Lerose, L. Padua, G. Sandrini, and C. Casali. "Gait pattern in inherited cerebellar ataxias." In: *Cerebellum* 11.1 (2012), pp. 194–211. ISSN: 14734222. DOI: 10.1007/s12311-011-0296-8.

[29] M. M. Shabot. "Ten commandments for implementing clinical information systems." In: *Proceedings (Baylor University. Medical Center)* 17.3 (2004), pp. 265–269. ISSN: 0899-8280. URL: http://www.ncbi.nlm.nih.gov/pubmed/16200110http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1200662.

[30] M. Shafizadeh, P. Watson, and B. Mohammadi. "Intra-limb Coordination in Gait Pattern in Healthy People and Multiple Sclerosis Patients." In: *Clinical Kinesiology* 67.3 (2013), pp. 32–38. URL: http://www.researchgate.net/profile/Mohsen{_}Shafizadeh2/publication/256595772{_}Intra-limb{_}Coordination{_}in{_}Gait{_}Pattern{_}in{_}Healthy{_}People{_}and{_}Multiple{_}Sclerosis{_}Patients/links/02e7e52374460a6c70000000.pdf.

[31] J. Smith. *Desktop Applications Vs. Web Applications*. URL: https://www.avestagroup.net/DetailsEN.aspx?PostID=1006{\&}CataType=5{\&}CataID=1006http://www.streetdirectory.com/travel{_}guide/114448/programming/desktop{_}applications{_}vs{_}web{_}applications.html (visited on 02/06/2018).

[32] *Sporttechnologie - Bergwandelen - Onderzoek*. URL: https://eduweb.hhs.nl/{~}bergwandelen/onderzoek.htm (visited on 02/09/2018).

[33] J. P. Vandenbroucke. "Increasing the accessibility of data." In: *BMJ (Clinical research ed.)* 309.6958 (1994), p. 879. ISSN: 0959-8138. DOI: 10.1136/BMJ.309.6958.879. URL: http://www.ncbi.nlm.nih.gov/pubmed/7950640http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2541047.

[34] *Why desktop apps are making a comeback – Mathilde Collin – Medium*. URL: https://medium.com/@collinmathilde/why-desktop-apps-are-making-a-comeback-5b4eb0427647 (visited on 02/12/2018).

[35] *Why Django is the Best Web Framework for Your Project*. URL: https://steelkiwi.com/blog/why-django-best-web-framework-your-project/ (visited on 02/12/2018).