UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

BACHELOR IN COMPUTER SCIENCE

Decision support system for the automatic definition and assignment of personalised rehabilitation routines for stroke patients

Sergio Martínez Cid

July, 2022

DECISION SUPPORT SYSTEM FOR THE AUTOMATIC DEFINITION AND ASSIGNMENT OF PERSONALISED REHABILITATION ROUTINES FOR STROKE PATIENTS

UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

Department of Technology and Information Systems

BACHELOR IN COMPUTER SCIENCE

COMPUTING

Decision support system for the automatic definition and assignment of personalised rehabilitation routines for stroke patients

Author: Sergio Martínez Cid

Advisor: David Vallejo Fernández

Co-advisor: Cristian Gómez Portes

July, 2022

Sergio Martínez Cid

Ciudad Real – Spain

E-mail: Sergio.Martinez34@alu.uclm.es
Telephone: 608 39 62 79

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE	**VOCAL**	**SECRETARIO**
Fdo.:	Fdo.:	Fdo.:

Abstract

Stroke or cerebrovascular accident is one of the mayor causes of death and disability in the world. Stroke has a growing social and economic impact, and this impact is increased in developing countries, and in the low-income segment of the population. Stroke affects both cognitive and physical abilities of patients. In addition, patients need to perform a rehabilitation process, which usually takes multiple months.

Due to the impact of stroke, there are many IT systems that try to enhance the stroke prevention and rehabilitation processes. In particular, there are system designed to facilitate the execution of the physical routine from home. Home rehabilitation enables the reduction of costs incurred by transporting the patient to the clinic, and makes rehabilitation more accessible, but it creates challenges related to motivation and correct execution of exercises.

The project detailed in this document has been carried out in the context of the development of a commercial system in the company Furious Koalas, development in which the author has participated. The system is a web application whose goal is facilitating patient rehabilitation by guiding their movements, and supporting therapists in monitoring the progress of the patient.

The project consisted in the development of a decision support system in charge of automatically defining physical rehabilitation routines for treating stroke. The project also includes the integration in the system. The automatic definition of physical rehabilitation routines is advantageous for the therapist, since it reduces the time devoted to the definition of routines. Therefore, it is possible to increase the quality time that the therapist spends with patients. Additionally, the decision support system is also capable of generating explanations regarding the suggested rehabilitation routines. This functionality falls within the field of Explainable Artificial Intelligence (XAI), which is going to be key for overcoming the ethical obstacles that prevent the integration of artificial intelligence in areas such as healthcare.

Resumen

El ictus o accidente cerebrovascular es una de las mayores causas de fallecimientos y de discapacidad en el mundo. El ictus supone un impacto creciente tanto a nivel social como a nivel económico, y este impacto se acrecienta en los países en vías de desarrollo y en sección de la población con bajos ingresos. El ictus afecta tanto a las capacidades cognitivas como a las capacidades físicas de los pacientes. Además, los pacientes necesitan llevar a cabo un proceso de rehabilitación que suele durar varios meses.

Debido al impacto de los ictus, existe numerosos sistemas informáticos que intentan mejorar los procesos de prevención y rehabilitación del ictus. En concreto, existen proyectos dedicados a facilitar la ejecución de la rehabilitación física desde el hogar. La rehabilitación desde el hogar permite reducir costes relacionados con el transporte del paciente a la clínica y hace la rehabilitación más accesible, pero genera retos relacionados con la motivación y con la correcta ejecución de los ejercicios.

El proyecto detallado en este documento se encuadra en el contexto del desarrollo de un sistema comercial en la empresa Furious Koalas, desarrollo en el cual el autor del proyecto ha tomado parte. El sistema es una aplicación web que tiene como objetivo facilitar la rehabilitación de pacientes guiando sus movimientos y dando soporte al terapeuta para la supervisión del progreso del paciente.

El proyecto ha consistido en el desarrollo de un sistema de soporte a las decisiones, que se encarga de definir automáticamente rutinas de rehabilitación física para pacientes de ictus. El proyecto también incluye la integración en el sistema. La definición automática de rutinas de rehabilitación física supone una ventaja para el terapeuta, ya que reduce el tiempo dedicado a la definición de rutinas. Así, se puede aumentar el tiempo de calidad que el terapeuta pasa con los pacientes. Adicionalmente, el sistema de soporte a las decisiones incluye la capacidad de generar explicaciones sobre las rutinas de rehabilitación sugeridas. Esta funcionalidad se encuadra dentro del campo de la Explainable Artificial Intelligence (XAI), que va a ser clave para superar los obstáculos éticos que previenen la integración de la inteligencia artificial en áreas como la medicina.

Acknowledgement

Quisiera dar las gracias a aquellas personas que me han apoyado, que me han formado y me han hecho crecer como persona. En primer lugar, quiero agradecer a mi familia, por su amor sincero y por haberme guiado y acompañado hasta llegar a este punto. En particular, quiero agradecer a mis padres y a mi hermana, por ser tan pacientes, por haberme apoyado durante todo este tiempo y por formar una parte esencial de mi vida.

Quiero agradecer a mis amigos, con los que he pasado tantos buenos momentos y cuyo recuerdo llevo en mi corazón.

También quiero agradecer a los profesores que me han enseñado en la escuela, en el instituto y en la universidad. He tenido suerte de que muchos de ellos me han transmitido no solo su conocimiento, sino que también me han transmitido su pasión y han contribuido a mi formación como persona.

Finalmente, quiero agradecer a los tutores de este proyecto, David y Cristian, porque ha sido gracias a sus consejos que he sido capaz de llevar a cabo este proyecto.

Sergio

A mis padres y a mi hermana

Contents

List of Tables

List of Figures

List of code listings

List of acronyms

MVC	Model View Controller
AR	Augmented Reality
AI	Artificial Intelligence
XAI	Explainable Artificial Intelligence
DSS	Decision Support System
ANN	Artificial Neural Network
FIS	Fuzzy Inference System
MIS	Management Information System
KBS	Knowledge-Based System
EIS	Executive Information System
ODSS	Organizational Decision Support System
GDSS	Group Decision Support System
DBMS	Database Management System
ER	Entity–Relationship
FIRST	Fast, Isolated, Repeatable, Self-validating, Timely
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTML	HyperText Markup Language
SASS	Syntactically Awesome Style Sheets
CSS	Cascading Style Sheets
ORM	Object–Relational Mapping
UI	User Interface
ML	Machine Learning
ASD	Adaptive Software Development
OS	Operating System
IDE	Integrated Development Environment
UML	Unified Modeling Language

Chapter 1

Introduction

In this chapter the project detailed in this document is introduced. First, an overview of stroke an its socioeconomic impact is presented. Then, the context where the project has been executed, and the project itself are described. Finally, the remaining chapters of this document are listed.

1.1 Basic stroke foundations

Cerebrovascular accident or stroke is one of the leading causes of death and disability in the world. It particularly affects the elderly [FRN$^+$16]. The symptoms of stroke vary widely from patient to patient, but it can have an impact on their physical and mental abilities. Recovery from stroke may take several months, but in order to minimize the resulting disabilities, it is necessary to have constant supervision by healthcare professionals during this process [BHW$^+$17].

There are two main types of stroke: ischemic and hemorrhagic, where the former is the most common case. Ischemic strokes are also known as infarcts, and they are caused by a restricted blood flow to an area of the brain. This is normally a consequence of a blocked artery. Hemorrhagic strokes originate from damage to brain tissue caused by bleeding, and it constitutes up to 15% of all strokes. The major cause of hemorrhagic strokes is hypertension. Hemorrhagic strokes have a higher mortality than ischemic strokes. The impact of stroke in the patient not only depends on the type of stroke, but also on the size and the location of the stroke. A stroke may happen in either the left side of the brain or in the right side. In rare cases, it may happen in both sides of the brain, which is called a bilateral stroke [GW13].

Stroke has a significant impact on the physical condition of patients. It is common for stroke patients to have a feeling of weakness or an observable weakening (paresis) of the limbs and of the face. The limbs and the side of the face affected depends on the side of the stroke, where left side stroke patients will lose strength on their right side and vice versa [HD81]. Related with the weakened limbs, another relevant symptom is an abnormal gait. Additionally, some patients report a tingling sensation on the skin (paresthesia). It has also been observed that patients commonly report dysphasia, which consists in difficulty swallowing, and dysarthria, which consists in difficulty speaking. Symptoms affecting

Figure 1.1: Therapist guidance helps in the correct execution of physical rehabilitation

eyesight include eye movement abnormality and visual field defects. Stroke can also cause headaches and dizziness, but with less prevalence [YC09].

Stroke also causes cognitive impairment. The many stroke survivors suffer from dementia, but it is not the only effect on the cognitive ability of patients. A stroke on the left side of the hippocampus can impair verbal long-term memory, while a stroke on the right side of the hippocampus would impair nonverbal long-term memory. Stroke can also cause white matter lesions, which can lead to reduced cognitive ability and verbal fluency. Finally, cerebral micro-bleeds might can lower non-memory related cognitive abilities [STY14].

While exercise for increasing physical aspects such as strength or endurance are included in physical rehabilitation for stroke patients, functional exercises are of particular relevance in this context. Functional exercises are exercises where the patient performs day-to-day tasks. The main advantage of these exercises is that they simultaneously increase strength, speed, endurance, and precision, while providing the patient with a greater level of independence [Dob04]. Exercises such as climbing stairs, turning pages of a book, and using a spoon can have a significant impact on the quality of life of the patient.

1.2 Socioeconomic context

The beginning stages of the rehabilitation process are particularly costly, namely because of the cost of transporting patients from their homes to the hospital and the cost of the therapy itself. Additionally, patients are usually unable to take proper care of themselves. This leads to either higher costs or the need of another person, usually a family member, to help them. The latter solution may not have a direct socioeconomic impact, but it can lead to the *Caregiver Stress Syndrome*, which is characterized by physical, mental and emotional exhaustion and is highly associated with other health issues [CPANBM+19].

Since the rehabilitation process heavily relies in qualified healthcare professionals, stroke has an particularly high impact in developing countries, where treatment is not as readily available. The impact on the workforce is also quite substantial. 75% of the patients either are unable to work or have their productivity substantially lowered [CC00]. This creates a negative feedback loop in low-income countries, where the low availability of medical staff is further intensified by the impact of stroke in working professionals. Not only do low-income countries have a higher incidence of stroke, but the morbidity and the mortality rates of stroke patients is considerably higher than in high-income countries. In particular, mortality rates are more than 3 times higher in low-income countries [AAM+12]. Furthermore, the incidence of stroke is higher in low-income populations within high-income countries [MWC+15].

It is in this socioeconomic context where the system can help in reducing the impact of stroke, since it can make the rehabilitation process more effective and accessible. In particular, the DSS can offload a portion of the workload of the therapist, hence reducing costs and increasing the quality of the rehabilitation.

1.3 Project context

This project is executed in the context of a web system dedicated to home rehabilitation, called *Physio Galenus*. It is a commercial system that is currently being developed by the company Furious Koalas[1], which works in areas such as gamification, and serious games. The author of this project started in this company as an intern, and he is currently working in the development of *Physio Galenus*, especially in the server-side of the application.

In particular, *Physio Galenus* allows therapists to overview the physical rehabilitation of stroke patients telematically. This system helps patients perform their physical rehabilitation by guiding their movements through immersive technologies, namely Augmented Reality (AR). One of the key aspects of the system is accessibility. Rather than using additional sensors such as the Kinect™[2] for the immersion technologies, the system makes use of webcams, which are cheap, and ubiquitous. Additionally, webcams do not require much space for their usage, which makes the system more adoptable. Furthermore, the system has been implemented using web technology for facilitating the adoption process of therapists and patients, since no installation process is required. The system uses the gym metaphor for describing physical rehabilitation. The rehabilitation exercises are grouped into a routine, and each exercise in the routine has its own number of sets and repetitions. The system is structured into three different applications:

[1] `https://www.furiouskoalas.com/`
[2] `https://developer.microsoft.com/es-es/windows/kinect/`

1. A client application for the patient. Its main purpose its to execute the rehabilitation routine.
2. A client application for the therapist. It allows the therapist to manage patients.
3. A server application that receives requests from both client applications.

An overview of the system can be seen in Figure 1.2.

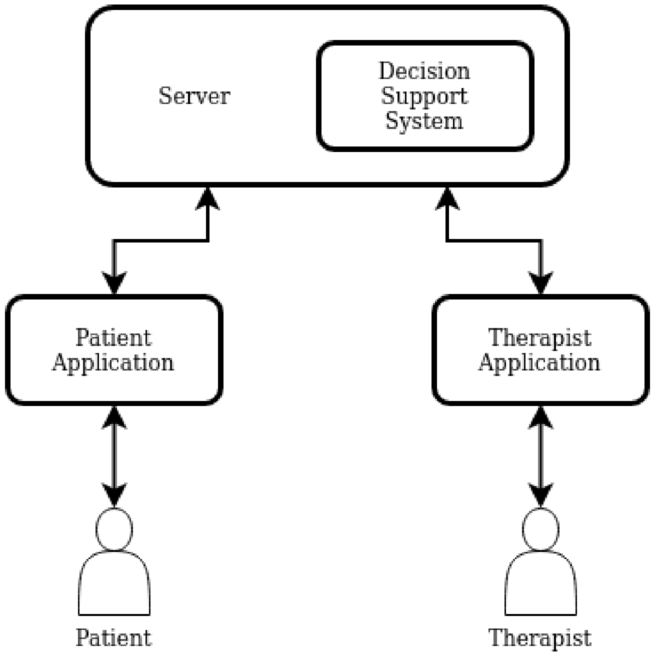


Figure 1.2: System overview

1.4 Project proposal

The project detailed in this document consists in the design and development of a DSS that automatically suggests a personalized routine for the patients. Having the possibility of receiving automatic recommendations reduces the time spent by the therapist reviewing each patient. The time saved can be allocated to more clients, which increases the effectiveness and reduces costs, or it can be dedicated to increasing the time spent with the patients during their rehabilitation. The latter option increases the level of quality, since guidance is needed for treatment to be effective [PCA07]. An important feature of the DSS is that, although recommendations are generated automatically, the therapist must ultimately review them before assigning the new routine to the patient. This workflow is enforced because a flawed suggestion could result in serious injuries.

For the development of DSS, multiple Artificial Intelligence (AI) techniques were evaluated, and a Fuzzy Inference System (FIS) was ultimately chosen. FIS was chosen because there is a level of uncertainty when defining the personalized routines, and fuzzy logic can handle uncertainty. The development of the FIS involved studying the strategy for adjusting the routine, defining the fuzzy variables, and defining the fuzzy rules. The definition of the fuzzy rules was an iterative process, since it was necessary to ensure that the rules generated an output for every combination of inputs, and that the output was realistic.

In addition, the DSS must be able to provide explanations of the decisions that it makes. The subfield of AI pertaining the explanation for the decisions of models is called Explainable Artificial Intelligence (XAI). XAI systems are more trustworthy than black-box AI models because users can see the reasoning behind the decisions of the system. XAI is key in the integration of AI in fields where clarifying the responsibility over the decisions of models is important. Healthcare in general, and physical rehabilitation in particular, is an example XAI could facilitate the integration of AI. A key feature of the generated explanations is that it is understandable by experts of the domain, in this case by the therapists. The explainable aspect of the DSS influenced the selection of the AI technique that was chosen, since fuzzy rules are highly interpretable. The first measure taken for generating an XAI was minimizing the number of fuzzy rules, and their complexity during the definition of the FIS. This maximizes the explainability of the rules. The explanation is provided by first selecting the most relevant rule for the output, and then generating an explanation from the most relevant rule. To facilitate the comprehension of the explanations, they are presented in a visual format rather than in a textual format.

1.5 Document structure

This document has been structured into the chapters listed below, in accordance to the standards for Final Degree Projects of the *School of Computer Science*, of the *Castilla-La Mancha University* (UCLM).

Chapter 2: Objectives

In this chapter the main objectives and the specific objectives of the project are presented.

Chapter 3: State of art

In this chapter, the theoretical foundations and technologies that were studied for the design and implementation of the project.

Chapter 4: Methodology

This chapter details the methodology followed during the project, as well as used resources.

Chapter 5: Architecture

This chapter describes the architecture of the implemented system. The problems that emerged during the project, and the pondered solutions are also explained.

Chapter 6: Results

This chapter presents the final results of the project.

Chapter 7: Conclusions

This chapter evaluates the achieved objectives, and lists possible improvements to the system.

Chapter 2

Objectives

In this chapter the objectives of the project will be discussed. First the general objective is presented, and then the specific objectives that support the main goal are specified in detail.

2.1 General objective

The main objective of this project is to design, develop and validate a decision support system for the automatic definition and assignment of personalized rehabilitation routines for stroke patients. The DSS will be integrated into a web system dedicated to home rehabilitation of stroke patients, and it will support the decision-making process of the therapist. The web system consists in a server, a client application for patients and a client application for therapists (see Chapter 5 for more details). The DSS will receive as input the execution data of the routines executed at home by the patients. The personalized routine consists in a set of exercises, each with their number of sets, repetition and time. The DSS will be responsible not only for adjusting the parameters of the exercises, but also for selecting the number of exercises and for choosing the particular exercises to be performed by the patient.

Another desirable feature of the DSS is that it belongs to the field of XAI, meaning that its decisions are explainable. If therapists understands the reasoning behind the decisions of the DSS, they will be able to rely on the system. Therefore, the DSS should be able to provide explanations about the personalized routine to the therapist.

2.2 Specific objectives

Once the general objective of the project has been established, it is necessary to define specific objectives that support the general objective.

- **Study the technologies, techniques and tools that optimize the development and deployment of the proposed DSS in a production system.** Since the DSS will be deployed in a production environment of a commercial system, it is important that the DSS is scalable and that the chosen tools for the development are optimal. Therefore, several tools and libraries will be evaluated, and factors such as the provided features and the level of maturity will be pondered. Additionally, the selected technology should facilitate the generation of explanations.

- **Study the required knowledge for the automatic definition of physical rehabilitation routines.** The definition of physical rehabilitation routines is not a trivial task, since it requires medical knowledge. Therefore, it is necessary to study the main factors that are considered by therapists when defining a rehabilitation routine. This knowledge will then be used for creating the model of the DSS, and for defining its input data. The data will be either manually introduced by the therapist or automatically generated by the system during the execution of the routines.
- **Design, development and integration of the DSS.** Once the technique has been chosen, it is necessary to design the particular strategy that the DSS will follow for suggesting the personalized routine. Then, the design will be implemented using the chosen technologies and tools. Finally, the DSS will be integrated in the web system. In particular, it will be necessary to integrate the DSS in the server and in the therapist application.
- **Integration of a component that explains to the therapist the reasoning of the DSS.** It is necessary that the DSS generates explanations for the decisions taken, as it is one of its distinctive features. The explanations will be shown to therapists when they review the personalized routine to inform them about the decisions the DSS took when adjusting the routine. The explanations should be intelligible by the therapist, ideally presented in a visual format. Providing explanations for decisions make by AI systems is a subfield of AI called Explainable Artificial Intelligence (XAI).
- **Testing of the DSS.** Testing the DSS is important to ensure that the output is within the expectations, and that the suggested personalized routines are realistic. Testing should be done frequently in order to correct errors as early as possible. Additionally, testing the DSS will be a task performed manually, since there is no simple automatic approach for validating the level of realism of a routine.

Chapter 3

State of art

The aim of this chapter is to present the technological advancements that have supported this development. Firstly, the technologies used in physical rehabilitation for neurological conditions are described, with emphasis on the AI techniques that have successfully been implemented in this context. Then, there is a general overview of Decision Support Systems (DSSs) and their applications. Additionally, the frameworks or categories in which DSSs can be divided are reviewed to understand the different approaches for supporting the decision-making process. Finally, techniques for incorporating explanations to decisions performed by AI are described, along with the resulting trade-offs.

3.1 Physical rehabilitation in neurological conditions

Neurological conditions have a global impact on the health of the patient, affecting not only their cognitive abilities but also their physical abilities [MSM11]. Stroke, cerebral palsy and brain tumors are examples of neurological conditions that require a considerable amount of physical rehabilitation. Several technologies have been applied to the rehabilitation process, including AI techniques.

3.1.1 Home rehabilitation and digital technologies

In recent years, home rehabilitation is becoming a viable alternative to traditional rehabilitation [PCP+14]. This is because of a series of technological improvements that allow therapist to offer treatment with similar quality than regular rehabilitation in a telematic way. One of the most relevant of these improvements is the internet, which enables therapists to perform video calls with the patients for guiding their exercises and for evaluating their progress [SZC+13]. The advancements made in terms of networking do not only refer to the bandwidth or the speed, which is essential for communicating with the patient via video conferences, but also to the widespread adoption of the internet. The internet adoption is no longer limited to developed countries or to urban areas. The regions where it is not viable or affordable to install wired connections can still connect to the internet by broadband cellular network or by other innovative projects with this goal in mind, such as Starlink[1]. As digitalization progresses, the demographic groups that used to have a low level of digital

[1] https://www.starlink.com/

skills are now actively using technologies such as the internet. Coincidentally, one of these demographic groups is the elderly, which make up the majority of stroke patients.

However, the internet is not the only technology that makes home rehabilitation viable. Only using the internet has a minimal effect on the required level supervision by the therapist. Ideally, supervision should be minimal to reduce the costs and to allow more flexibility in the schedule of both the therapist and the patient. To ensure that the quality of the rehabilitation does not decrease as the level of supervision is reduced, it is essential that countermeasures supported by technologies other than the internet are put in place. These include immersive technologies and computer vision, for guiding the movements of the patient in real time, and several Artificial Intelligence techniques to further improve its quality.

Computer vision consists in the analysis of images and video to extract information. In the particular case of home rehabilitation, it is capable of providing the pose of the patient from a video stream in real time (see Figure 3.1). To improve the precision of the analyzed pose, multiple sensors can be used to complement the video stream. Some of the most common sensors in home rehabilitation include accelerometers, gyroscopes, infrared sensors, the Wii Balance Board[2] and the Microsoft Kinect™. The most common sensor is the Microsoft Kinect™ which is one of the cheapest alternatives with a considerable level of precision. It was originally designed for playing video games, but it has made its way into multiple projects for home rehabilitation. It has a regular camera, an infrared projector and an infrared detector, producing an RGB-D image as a result. However, the use of excessive sensors rises the costs significantly and makes the setup of the home rehabilitation system more complex.

Immersive technologies is also highly beneficial to the rehabilitation process, whether virtual reality, augmented reality, or others. It allows the patient to visualize how the exercise should be performed by adding virtual elements such as spheres, paths or a human contour to the image. By using the data generated by computer vision, it is also possible to provide real-time visual feedback to patients about how their execution of the exercise compares to the ideal execution. This aspect is key for reducing the level of supervision.

Additionally, immersive technologies makes it possible to make exercises more appealing through gamification. Gamification is the process of improving activities by incorporating elements found in games [Ham19]. The term exergaming can be used to describe gamified physical activities. In particular, exergames are useful for motivating patients and make rehabilitation more effective. There are multiple techniques found in games that can be incorporated to rehabilitation systems [TWD+16]:

- Integrating some level of social interaction between users of the system.

[2] https://web.archive.org/web/20080321044858/http://www.consolewatcher.com/2007/07/stay-fit-with-nintendo-wii-balance-board/

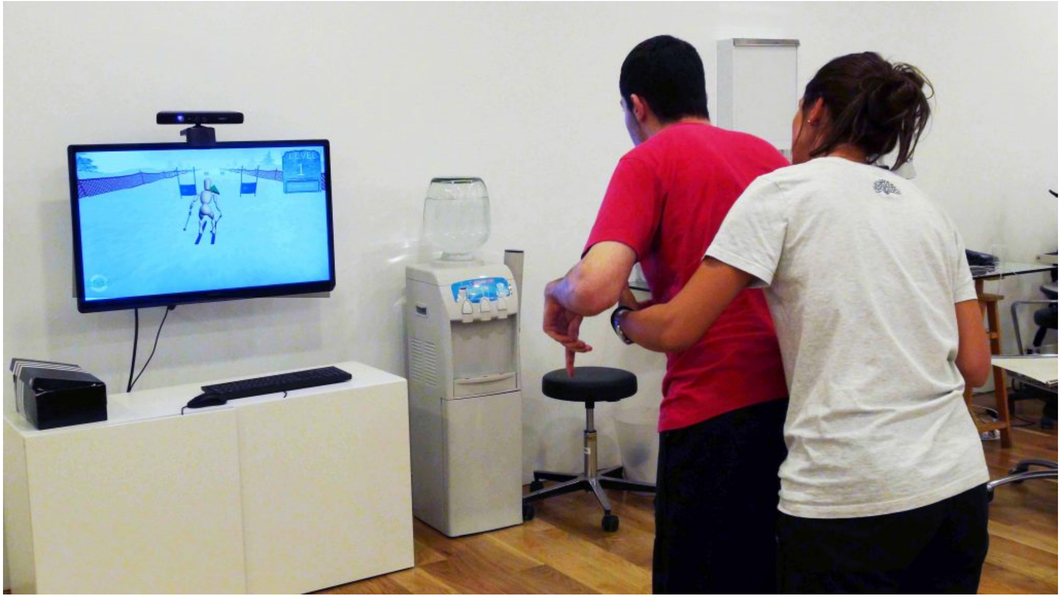


Figure 3.1: Example of computer vision being used for rehabilitation. Image obtained from: `https://rehametrics.com/videoconsola-rehabilitacion/`.

- Enabling social competition between players, through either real time competitions or the use of leaderboards.
- Adding unlockable content to the system, commonly in the form of customization elements.
- Introducing goals, levels, scores, achievements, and badges. These elements generate a feeling of progression, which motivates patients to adhere to the rehabilitation.

Gamification has been proven to provide better results in stroke rehabilitation. In particular, balance, functional mobility and functional independence improvements are greater when gamification is integrated [CJC$^+$21]. Additionally, gamification increases patient adherance to the physical routine. This is specially important, since adherence to the routine is generally low [KJS14].

These digital technologies allow patients to perform their rehabilitation exercises with minimal supervision by their therapist. This has an impact on the costs as well as on the quality of life of the patients and their families. Regular rehabilitation requires the patient to go to the hospital, which usually involves the help of a family member. Home rehabilitation enables patients with some digital skills to be more independent, and reduces the responsibilities of the family members of patients who are not able to manage a home rehabilitation system by themselves.

However, there are challenges in home rehabilitation. Namely, few systems evaluate the execution of the patients, which is why therapist supervision is still required. Still, it is not always possible for the therapist to overview the execution of the exercises by the patient in real time. For this reason it is useful to rehabilitation systems to integrate intelligent processes that facilitate the decision-making process of the therapist. The motivation of this project is to integrate one such intelligent process into a rehabilitation system.

3.1.2 AI-assisted rehabilitation technology

Artificial intelligence is a powerful technology that can be integrated in the rehabilitation process in a multitude of ways. In the case of on-site rehabilitation it enables us to enhance the effectiveness of the performed exercises, while in the case of home rehabilitation is is a key piece for compensating the reduced involvement of the therapist. The most important uses for AI in the rehabilitation process are adjusting the difficulty of the exercises, detecting dangerous positions and analyzing the collected data.

Adjusting the difficulty is a task performed by therapists that is particularly complex, and therefore it is not straightforward to substitute with technology. A large number of factors must be accounted for in order to select a suitable workload for the patient. A balanced difficulty motivates the patient and it is essential for an effective therapy. One of the main obstacles for home rehabilitation is the need to ensure that exercises are being performed safely at all times by the patient. This need can be overcome by AI technology able to detect dangerous positions and warn the patient accordingly [FCSO+15]. It would also be of use in a hospital or a rehabilitation center to help the therapist and to reduce the number of accidents. Such a technology not only requires accurate pose estimation, but it would also need to analyze how dangerous the position is in real time. A more indirect method of incorporating AI in rehabilitation would be collecting and analyzing data from the patients performing the exercises. A data mining process supported by AI techniques would be able to extract additional data from the patients, such as when a patient is lagging behind. Furthermore, it would be possible to detect general flaws and lacking areas in the rehabilitation processes at a clinic level.

While there is a wide range of AI techniques available, one of the most prominent technique in recent years is Artificial Neural Network (ANN) [WLZ+20] [HBATA20] [SPS+20]. ANNs, similarly to other AI techniques like Particle Swarm Optimization and Ant Colony Optimization, are inspired by natural phenomenon. In particular, their design is based on the neural networks in the brains of animals. In 1943, Warren McCulloch and Walter Pitts initially proposed the idea of imitating neurons in the human brain. The first ANN was the perceptron, designed by Frank Rosenblatt in 1958 [Hay08]. Nowadays, ANNs are a very popular technique that is being tested in multiple fields with varying degrees of success.

ANNs work by the combined result of the neurons, organized in layers, that constitute the network. Each neuron has several input and an output. A weight is applied to every input, and the net input is determined by a function applied to the sum of the weighted inputs. Whether a neuron is activated or not is determined by the net input, an activation function and a threshold (see Figure 3.2). As ANNs are a supervised learning technique, it is necessary to train them using examples from which the output is known in advance. These examples are used to adjust the weights and thus minimize the error and refine the output.

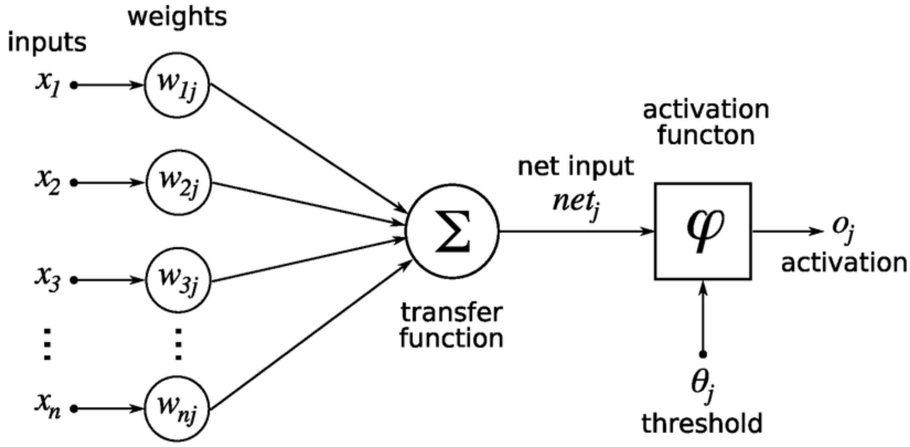


Figure 3.2: Architecture of a single neuron in a neural network. Image obtained from: [Kei18].

The limiting factor when using ANNs is the computing power and the availability of diverse, labeled data. The issue of computing power used to be a major concern in the origins of ANNs, but currently it is becoming less relevant as hardware improvements continue. The training data is still prevailing, as data is labeled manually most of the time. A partial solution for the lack of data is the automatic generation of synthetic data, which can provide good results in certain scenarios [CCP+19]. However, it is not a general solution for the lack of labeled data, as it is necessary that the data is realistic and that it covers all of the expected instances. While the number of application where ANNs can be used is very large, one of its strong points is dealing with images and video. It can be used for identifying elements in an image or for extracting information from a video source. In the context of rehabilitation, ANNs can play a significant role in rehabilitation enhanced by AR and computer vision, as well as detecting dangerous poses [GRBS20]. One of the drawbacks of ANNs is that it is a black-box model, meaning that it is not interpretable by humans.

Another relevant AI technique to be integrated in the rehabilitation process is fuzzy logic in the form of FIS, even if they are usually associated with control systems rather than AI. Fuzzy logic is a particular case of infinite-valued logic, where truth variables may take values in the interval [0, 1] [Pel00]. It was created in 1965 by Lofti Zadeh, and it aims to achieve *computing with words*, to deal with uncertainty [MZT+10]. The most common approach for creating a FIS is the Mamdani approach [Mam74]. The first step consists in the *fuzzification* of input values. Fuzzy sets are defined in every universe of discourse, each with their own fuzzy membership function whose range is the interval [0, 1]. Therefore, for each input value, the degree of membership of each fuzzy set in the aforementioned universe of discourse is computed. Next, inference is performed on the fuzzified values with a set of fuzzy rules. Fuzzy rules are of the form *if VARIABLE1 is FUZZY_SET1 then VARIABLE2* *is FUZZY_SET2*. Simple clauses can be combined by using operators analogous to those of classical logic, namely AND, OR, NOT. Additionally, fuzzy logic incorporates a specific set of operators called hedges. These operators are not found in classical logic, and they allow for further expressiveness when defining fuzzy rules. Some examples of hedges are *very*, *extremely*, *a little*. Once inference has been performed, a fuzzy set is obtained as an output. It is then necessary to *defuzzify* the fuzzy set to obtain a number as the final result. Popular approaches include the center of gravity and the mean of maxima. As a final note on fuzzy logic, it is worth noting that there are alternative approaches to Mamdani, most notably the Takagi-Sugeno-Kang (TSK) approach is also popular.

One of the most relevant application for fuzzy logic is medical decision making [DDS+16]. The inherent fuzziness of making a diagnosis can be dealt with by incorporating fuzzy logic in the process. This makes FISs a great tool for enhancing certain aspects of physical rehabilitation, such as adjusting parameters of the rehabilitation exercises for an optimal recovery. Additionally, FISs are ideal for XAI since the inner structure of the system is not a black box, and the decisions made are more trustworthy. However, creating a FIS in a medical context is not a streamlined process. Specifying the rules that guide the decisions of the system is not a trivial task, as it depends on knowledge elicited from experts.

Another AI technique that deals with uncertainty is Bayesian networks. Bayesian networks are built with expert knowledge, and they consist in directed acyclic graphs, where variables are nodes and the edges represent the conditional dependencies between any two variables. Inference consists in calculation of the probabilities of the variables that are not observed from the variables that have been observed and the conditional dependencies between variables. There is not an unique algorithm for performing inference [Ste00]. This AI technique has been used in some physical rehabilitation systems [SSG+10].

Markov models are also occasionally used in the context of rehabilitation systems. They are stochastic models for randomly changing systems where the Markov property applies: future states depend exclusively on the current state [KLC98]. The system can be defined as an agent interacting with its environment synchronously. Markov models are stochastic because the resulting state of an action in a concrete initial state is not deterministic. There are multiple types of Markov models that are specific for different situations:

- Markov chain, where the system is autonomous and the state of the system is fully observable
- Hidden Markov model, where the system is autonomous and the state of the system is partially observable
- Markov decision processes, where the system is controlled and the state of the system is fully observable
- Partially observable Markov decision processes, where the system is controlled and

the state of the system is partially observable

While Markov models are not as versatile as other AI techniques, they can be implemented for aiding rehabilitation in certain situations. Some examples include guiding the patients movement when performing the exercises and evaluating the performance of the patient [KHH+11] [OSB20].

3.1.3 AI projects for rehabilitation

Several projects, similar to the system in which the subsystem is going to be integrated have been reviewed. The following is a summary of the surveyed projects.

- **Serious games for rehabilitation: Gestural interaction in personalized gamified exercises through a recommender system** [GGTDMCTC19]
  - It consists in a rehabilitation system using augmented reality with an exergame player and a designer tool. There is also a recommender system that selects gamified exercises depending on the user interactions and history.
  - Advantages: It has extensive validation, it supports physical, cognitive and hybrid exercises, and there are single-player and multiplayer, collaborative and competitive modes.
  - Disadvantages: A Kinect™ sensor is required.
- **Serious games: Rehabilitation fuzzy grammar for exercise and therapy compliance** [FCSO+15]
  - It consists in a remote rehabilitation system for physical injuries based on the Kinect sensor. The system has a grammar that enables therapists to specify rehabilitation exercises. Additionally, there is a component based on fuzzy logic that determines whether the routine is executed correctly.
  - Advantages: The therapist can specify precisely the exergames to be executed, and the system considers dangerous postures for the patient.
  - Disadvantages: A Kinect™ sensor is required, there is no automatic adjustment of any of the parameters of the routines, and there is a lack of feedback.
- **Self-adaptive games for rehabilitation at home** [PMBB+12]
  - It consists in a rehabilitation system integrated with a rehabilitation station. The system checks the constraints defined by the therapist using fuzzy logic, and it adapts the games' difficulty using the Quest Bayesian adaptive approach. Games are played either with a Nintendo Wii Balance Board™ or a Microsoft Kinect™ according to the exercise goals.
  - Advantages: The game difficulty is adapted automatically according to a set of parameters for each game, and there is a fine level of granularity in the monitoring of the execution of exergames.

  - Disadvantages: Only one game has been designed, and the specification of additional exercises is not trivial.

- **A Telerehabilitation System for the Selection, Evaluation and Remote Management of Therapies** [ABB+18]
  - It consists in a remote rehabilitation system using the Kinect™ technology. It provides useful information to adapt the therapy of the patient. It can extract knowledge from the physical therapy record, and it can also provide real-time multimedia communication between the therapist and the patient when performing the therapy.
  - Advantages: It can provide information about the performance of the patient, and it can automatically recommend therapies.
  - Disadvantages: A Kinect™ sensor is required.

- **A Fuzzy-Based Adaptive Rehabilitation Framework for Home-Based Wrist Training** [KEA+14]
  - It consists in a web-based wrist rehabilitation system. The system can evaluate the performance and adapt the intensity of the tasks using fuzzy logic.
  - Advantages: Since it is a web-based system, it is available in all platforms that support web browsers and that have the required sensors, and the system can handle varying ranges of motion of the patient's wrist, and different speed and steadiness.
  - Disadvantages: It requires several sensors, such as accelerometers, gyroscopes and tracking cameras.

- **A personalized limb rehabilitation training system for stroke patients** [WWWL16]
  - It consists in a rehabilitation system that makes use of virtual reality for the visualization of the exercise and sensors such as the 9-dof posture sensor and the ultrasonic sensor. It is also able to generate a report of the user's performance. Limb stretching and limb strength exercises are supported.
  - Advantages: The accuracy of the detected posture of the patients is high.
  - Disadvantages: It requires a virtual reality device, and 9-dof sensors, which may not be affordable for all patients.

- **Opportunities of a Machine Learning-based Decision Support System for Stroke Rehabilitation Assessment** [LSS+20]
  - It consists in a system for assessing the rehabilitation of stroke patients. Machine learning is used to quantify the quality of the rehabilitation exercises from the data collected by the Kinect™ sensor. The assessment is provided by a web-based interface.

  - Advantages: The system was developed by making extensive use of expert knowledge, and several supervised learning algorithms were tested.
  - Disadvantages: A Kinect™ sensor is required.

- **Rehabilitation Exergames: Use of Motion Sensing and Machine Learning to Quantify Exercise Performance in Healthy Volunteers** [OSB20]
  - It consists in a rehabilitation platform called MIRA that automatically assesses the performance of the patients. There is support for exercises as well as exergames. It is based on the Unity engine. Two machine learning techniques have been used, namely dynamic time warping and hidden Markov model.
  - Advantages: There is a high variety of exergames, which enhances the level of gamification.
  - Disadvantages: A Kinect™ sensor is required, and the validation is limited by using healthy participants.

- **Winning compensations: Adaptable gaming approach for upper limb rehabilitation sessions based on compensatory movements** [ACL20]
  - It consists in a rehabilitation system for upper limb rehabilitation. It adapts the difficulty of the exergames by considering the compensatory movements of the patient. Additionally, it provides to the therapist quantitative measures of the number and type of compensatory movements.
  - Advantages: The system has multiple original exergames.
  - Disadvantages: A Kinect™ sensor is required.

- **Adaptation in serious games for upper-limb rehabilitation: an approach to improve training outcomes** [HGC+15]
  - It consists in a game adaptation technique for upper-limb rehabilitation. The rehabilitation consists of games controlled with a graphic tablet. The difficulty of the game is adjusted by using the ability zone of the user: a matrix containing all reachable areas of the graphic tablet.
  - Advantages: High levels of usability and gamification by involving the therapists and the users in the development process, and it does not require the use of a camera or an open space for performing the exercises.
  - Disadvantages: This system requires the use of a graphic tablet of a sufficient size to cover most of the arm's reach, which may increase the costs.

- **Adaptive gameplay and difficulty adjustment in a gamified upper-limb rehabilitation** [PCC+18]
  - It consists in a gamified upper-limb rehabilitation system that includes adaptative gameplay and difficulty to enhance the motivation of the patients. The system integrates several mini games which simultaneously motivate the patients and boost

their rehabilitation. A state machine is used to transition between the difficulties of the mini games.

 - Advantages: Highly gamified system, with adaptability of the difficulty of the mini games.
 - Disadvantages: A Kinect™ sensor is required.

- **The development of an adaptive upper-limb stroke rehabilitation robotic system** [KHH+11]
 - It consists in an automated system for a rehabilitation robotic device that guides stroke patients through an upper limb reaching task. It combines many sensors with a robotic arm to guide the patient. The decisions of the system are guided by a Partially observable Markov decision process.
 - Advantages: High level of monitoring of the patient's movements.
 - Disadvantages: Low accuracy on the results: The system was correct 65% of the time. Additionally, the expensive initial investment makes it less affordable.
- **Supporting rehabilitation training of COPD patients through multivariate sensor-based monitoring and autonomous control using a Bayesian network** [SSG+10]
 - It consists in a system for autonomous physical rehabilitation of chronic obstructive pulmonary disease. Sensors are used for monitoring the physical rehabilitation of the patient, and the captured data is analyzed with a Bayesian network.
 - Advantages: It can deal with uncertainty.
 - Disadvantages: It has insufficient accuracy and additional sensors cannot easily be integrated.
- **SWORD Health**[3]
 - A commercial remote rehabilitation AR system that guides the patients' movements when performing the exercises for musculoskeletal pain. It provides live feedback and it integrates gamification components.
 - Advantages: It is approved by the Food and Drug Administration (FDA), and it allows the therapist to review the progress of each patient.
 - Disadvantages: Makes use of a proprietary device for the rehabilitation.

3.2 Decision-support systems

Decision support systems are a broad category including many varied subcategories. Since the only common factor within DSSs is the ultimate goal they accomplish, a wide range of techniques are implemented for the purpose of supporting decisions. When developing a DSS, knowing about the techniques that have been applied successfully in the past is a valuable insight.

[3] https://swordhealth.com/

3.2.1 Basic foundations

Decision-support systems are information systems that supports the decision-making process. Using computers for aiding the decision-making process was first proposed in 1963, and the term was coined in 1971. DSSs can be developed for a wide range of knowledge domains, where the most common knowledge domains are healthcare and and businesses [EK06].

While there is not an unique definition of DSSs, there are 4 distinct features commonly present in DSSs [Spr80]:

1. DSSs are aimed at unstructured and semi-structured problems usually (but not always) faced by middle and upper management.
2. DSSs integrate models and the retrieval of data.
3. DSSs are designed with users with a low level of computer literacy in mind.
4. DSSs adapt to changes in the environment of the system.

DSSs are similar to Management Information Systems (MISs), but there are key different between the two types of systems. MISs are focused on querying and modifying information, and they are aimed at middle managers, whereas DSSs are decision focused, and they are commonly aimed at upper management. MISs may play a role in the decision-making process, but it is not their ultimate purpose. Additionally, MISs have a more structured information flow while DSSs can work with unstructured problems.

DSSs is a broad category that includes multiple specialized systems, one of the most relevant being Knowledge-Based System (KBS). KBS are a type of DSSs that heavily uses knowledge bases for making the decision. One of the main characteristics of KBS is that expert knowledge is used for the development of the KBS and that the knowledge is represented explicitly [SI94]. Another type of DSSs is Executive Information Systems (EISs). The term EIS refers to DSSs that are aimed at executives and other upper managers with a focus on the user interaction and and on data processing for providing relevant information [LE93]. Another popular specialization of DSSs is Group Decision Support Systems (GDSSs). GDSSs focus on supporting the solution of unstructured problems whose solution involves groups of people. The most relevant feature of GDSSs is providing adequate solutions for issues that arise in group dynamics [SC02]. Organizational Decision Support Systems (ODSSs) also derive from DSSs, but aimed for organizational tasks that involve multiple organization units [Geo91].

3.2.2 Architecture and components

The main components of a DSSs architecture are the knowledge base, the model and the user interface [Spr80]. In some cases, a communication component is also included (see Figure 3.3). The knowledge base, also known as the data base is responsible for storing data to be used by the model. The knowledge base contains internal data originated in transactional processes, but it may also contain external data from public data sources. In the particular case of knowledge-based systems, it may also include expert knowledge. A rich set of data is essential for effective decision making. Additionally, the unstructured nature of DSSs requires more flexibility than a MIS in the data capture and extraction process.

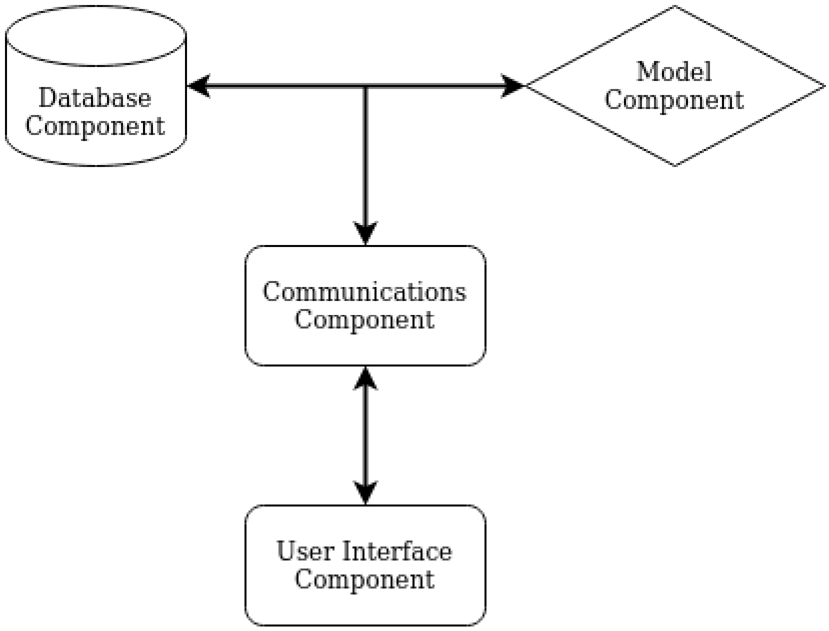


Figure 3.3: General DSS architecture.

Models are the core component of a DSSs. There are a multitude of types of models, including optimization models, econometric models, and fuzzy models. Models may also be categorized according to the layers of management; there are strategic models, tactical models and operational models. The chosen model is highly dependent on the concrete problem to be solved. Decision models in a DSSs are embedded in an information system to make use of the knowledge base. Additionally, the information system should should allow for the creation of new models and the interconnection between models.

The user interface is a major component in a DSSs, since a lot of the flexibility of the system stems from the user interface, and users are a key component of the DSSs architecture. The interface experience in a DSSs can be divided into three parts:

- The action language. It describes the actions at the disposal of the user for interacting with the system.
- The display or presentation language. It describes how the system presents information to the user.
- The knowledge required from the user. This is the knowledge that enables the user to effectively use the system. It may be memorized or it might be supported by a manual or by help commands in the system.

3.2.3 Applications and frameworks

There are many applications for DSSs, since there are many fields that greatly benefit from support in decision-making process. Most DSSs applications are for the corporate functional management. In particular, Production/Operation Management is the corporate area for which the largest number of DSSs have been designed. Production/Operations Management is the process of an organization that is in charge of the transformation of inputs into products and services while maintaining the required level of quality [KS08]. Within the corporate area, another popular application of DSSs is supporting business decisions. The focus of DSSs in this field include summarizing data to allow for better decisions, management of resources and identifying trends [SY94]. These systems fall under the term of *business intelligence*. Other relevant corporate areas for DSSs include marketing, transportation and finance.

One major application of DSSs in non corporate areas is the support for medical diagnosis [EK06]. The diagnosis of an illness is a complex task that requires a deep medical knowledge and it involves managing many factors, such as the symptoms or the patients history, and their interactions (see Figure 3.4). A DSSs for medical diagnosis is able to serve as a second opinion to a doctor, to offload some workload to increase the accuracy and precision of the diagnosis, and it can also enable less qualified professionals to perform a preliminary diagnosis. Other non corporate applications of interest for DSSs include ensuring structural safety, education, and managing natural resources such as forests and reservoirs [SCL96] [VFW00] [Kot11].

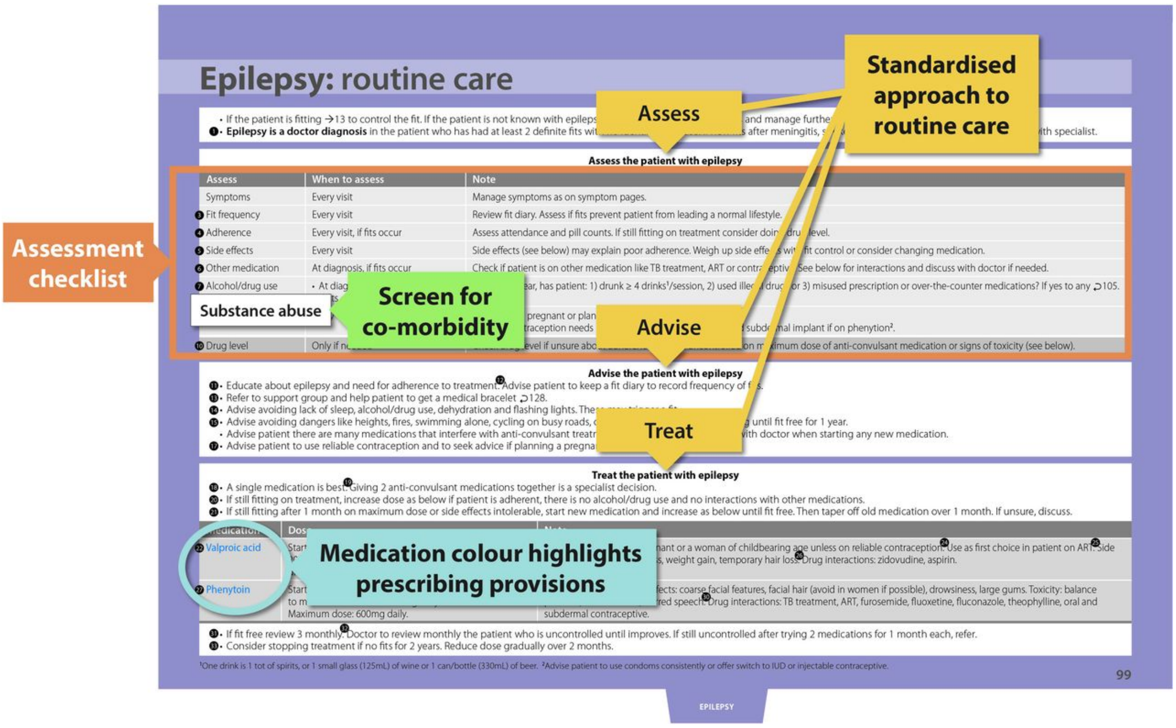


Figure 3.4: Example of a DSS. Image obtained from: [CPW+18].

One way of classifying DSSs was proposed by Alter in 1979. In this classification schema, the category of a DSSs is determined by the degree to which the output of the system is able to influence the final decision [Alt79]. The schema defines the following characteristics:

- File drawer systems. These systems provide access to data items, such as a data mart, which is a storage of data targeted to a particular area.
- Data analysis systems. These systems support data manipulation and summarization tools.
- Analysis information systems. These systems provide small models to generate information from the available data, whether internal or external. A major example of analysis information systems would be Business Intelligence systems.
- Accounting model-based DSSs. These systems are focused on supporting financial decisions. They incorporate financial models for tasks such as generating income statements and balance sheets, and estimating profits. Accounting models are usually based on formulas.
- Representation model-based DSSs. These systems estimate the consequences of actions that are generally based on simulation models.
- Optimization model-based DSSs. These models receive multiple constraints and generate an optimal solution while complying with those constraints. Notable examples include scheduling systems and resource allocation systems.
- Suggestion model-based DSSs. These systems are designed for structured tasks that are well understood, and where there is low variability on the solutions.

The most common categories are optimization model-based DSSs, suggestion model-based DSSs, and representation model-based DSSs.

The main issue with Alter's taxonomy is that as DSSs evolved, they became increasingly diverse. This means that the proposed categories no longer cover the majority of the DSSs. To solve this issue another framework was developed by Power [Pow02]. Alter's taxonomy was taken as a starting point, and the seven categories were reduced to three: data- driven , model-driven, and knowledge-driven DSSs. Additionally, two new categories were introduced for encompassing those DSSs that did not fit in any of the previous categories. The categories defined by Power are the following:

- Data-driven DSSs. These DSSs focus on the analysis, querying and manipulation of structured data, whether internal or external. The databases used are generally large. The included categories from Alter's taxonomy are file drawer systems and data analysis systems.
- Model-driven DSSs. These DSSs emphasize the use of models for supporting the decision process. Since only light use of data is needed for these systems, large database

are not necessary. The included categories from Alter's taxonomy are analysis information systems, accounting model-based DSSs, representation model-based DSSs and optimization model-based DSSs.

- Knowledge-driven DSSs. These DSSs make use of business rules and knowledge bases for decision-making. These systems are tightly related with the concept of data mining. It is an analogous category to Alter's suggestion model-based DSSs.
- Document-driven DSSs. These DSSs focus on managing and retrieving unstructured documents, including images, text files, audio, video and Web pages. A search engine is an essential component of a document-driven DSSs, especially when the number of documents involved is high.
- Communication-driven DSSs. These DSSs are systems incorporate technologies for communication, for collaboration and for decision support. This category includes GDSSs and groupware.

Additionally, there are several DSSs that do not fit neatly in any particular category, but they incorporate characteristics from multiple categories. These are called *compound* or *hybrid* DSSs, and they are the most common type of DSSs [HW01].

3.3 Explainable AI

There has been a surge in AI, with many developments in the techniques available to developers. Consequently, systems incorporating AI are flooding the market. This means that AI is increasingly in charge of more aspects of our lives. Given that this trend, many ethical and legal concerns have emerged. A proposed solution for these issues is XAI [CCWB20].

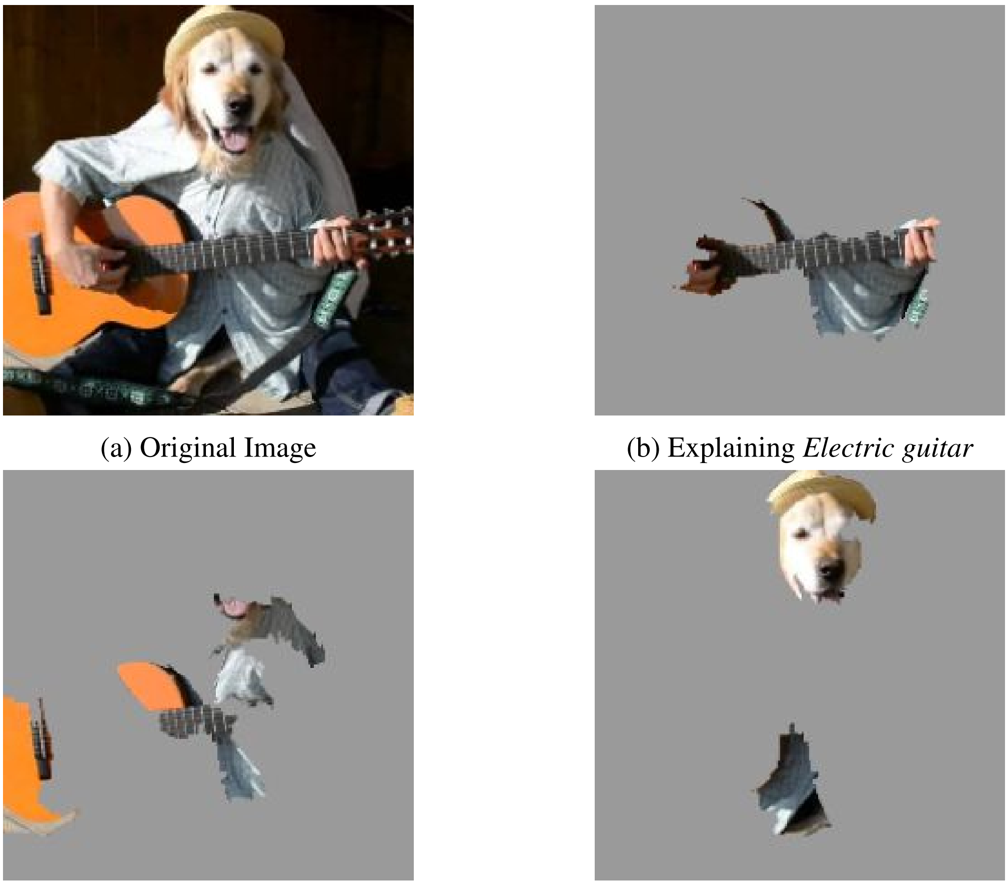
3.3.1 Basic principles

XAI is a subfield of AI where the provided results can be understood by a human with relative ease [GSC$^+$19]. XAI models are also called glass box models, in contrast to the opacity of black box models. The explanations provided by XAI allows us to trust the suggested solutions. This in turn means that AI will be able to be used in a wider range of fields. In particular, implementing AI systems in fields like healthcare, defense and law creates notable issues. In the case of healthcare, a misdiagnosis or a wrong treatment may cause serious disabilities or even the death of the patient. When it comes to defense, it is not clear who is responsible for the decisions taken in the battlefield. This issue is particularly present in the case of autonomous weapons, which are weapons that do not need human intervention for their operation (such as autonomous drones). Finally, the level of responsibility of AI agents is a heated debate. The degree at which an AI system or their creators are liable for the output of the system would have serious consequences in criminal trials. Another legal area where the position of AI is unclear is the area of intellectual property; it is not known whether a work created by an AI is own by their creators or whether it belongs to the public domain.

While explainability is a desirable aspect in AI systems, creating an XAI system is not an effortless task. First and foremost, extracting an explanation from the output of the model not trivial. It its costly and, since the field of XAI is still growing, it is not straightforward either. Therefore, it has been argued that explainability should only be incorporated in the AI applications where explainability provides a significant benefit. Another obstacle in the way of XAI is the trade off between explainability and performance. The recent developments in AI are partly due to the appearance of new Machine Learning (ML) techniques that automatically construct an internal model. These techniques provide a high level of performance that could not be achieved previously. However, it is considerably hard to provide an explanation from the model created by these techniques. A paradigmatic example of this challenge is deep learning; the neural network iteratively builds a model by adjusting the weight of the inputs of the neurons. The performance of the resulting model is generally high, but at the cost of explainability. The neural network and its parameters are chaotic, and generating an explanation from the result is extremely complex. Fortunately, there are ML models which are relatively adequate for XAI. For example, ensemble ML methods and decision trees create a model from which an explanation can be generated. Additionally, there are some AI techniques that do not belong to ML that are very explainable, the most notable example being fuzzy inference systems. In sum, there is a dichotomy between high-performance opaque models that incorporate the latest technology or explainable models that have less accuracy.

There are multiple techniques for creating an XAI, depending on the chosen AI model. One of the most adequate models for XAI are FISs, also known as Fuzzy Logic Systems. This is because their internal structure based on if-then rules with linguistic labels facilitates providing an explanation, since the executed rules are the explanation itself. However, not all FISs provide the same level of explainability. The larger the number of inputs and the number of rules, the more a FIS behaves like a black box model. This phenomenon is called the curse of dimensionality, and it leads to yet another trade-off between accuracy with many complex rules and explainability with few simple rules. One of the few tools created for explainable FISs is FisPro, a toobox that supports the process of generating FISs with supervised learning and data-driven approaches [GC11]. The resulting systems comply with the general guidelines mentioned above that enhance the explainability of a FIS: few rules with few inputs.

While the inner models of ML techniques are not easily interpretable, there have been multiple attempts at generating explanations from ML models, with varying degrees of success. This is usually an attempt of providing some explainability to highly effective models where having some explainability is very beneficial. The explanations provided may be model-based i.e. the simplicity of the model is enough to derive an explanation or it may be post hoc i.e. the explanation is generated after the fact [SML+21]. The explanation might be model-independent or it might be specific to the chosen model, which is more constraining. Additionally, the explanations can be global i.e. they apply to the complete model, or they might be local i.e. they only apply to a single input for the model. One of the earliest XAI techniques in ML is backpropagation and deconvolution. In the particular case of ML techniques that work with images, the areas of the image that were relevant for the output can be highlighted [vdVKGV22] (see Figure 3.5). This approach is quite comprehensible by humans, but the explanation is not complete and in the case of scientific fields it may require expert knowledge. Another similar approach is occlusion analysis, which differs on the way the relevant areas of the image are determined. By occluding sections of the input image and analyzing the effect on the score of the model. This is an example of a post-hoc model-agnostic XAI technique.



(a) Original Image (b) Explaining *Electric guitar*

(c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 3.5: Example of XAI applied to a classification ANN. Images obtained from: [RSG16].

3.3.2 Legal framework

AI, as well as many other bleeding edge technologies, has made an impact on society in a short time span, and it can be expected that the influence of these technologies increases. Such is the brevity of the irruption of AI that governments and institutions are still creating the legal framework where AI techniques will operate. The main issue of AI in the legal system is that determining who is accountable for the decisions of an AI system is unclear and difficult task. Circumstances where determining a culprit is not only difficult but also particularly important include car accidents caused by autonomous cars, deaths caused by autonomous weapons and misdiagnosis performed either partially or completely by AI. The individual that is deemed accountable depends on factors such as the area where the AI operates, the type of decisions that the AI is able to make, the circumstances of the particular incident, and the subjective beliefs of the judge or administrative authority in charge. The creator of the AI system, and the user of the AI system are the candidate culprits, but depending on the particular scenario it might also be considered an accident where no party is held accountable. While current legal systems in modern countries have already dealt with controversial cases regarding AI, in order to fully integrate AI in society, it is essential that most of the subjective and ambiguous aspects of the legal framework are properly managed.

One of the most relevant concerns about AI in the legal system is that many models are black-box models, where the inner logic is not intelligible. This means that it is not usually possible to determine whether there was a fault in the model or whether there was a fault in order component of the system where the AI is integrated e.g. sensors and actuators. It is for this reason that XAI is particularly beneficial, since it is designed to provide explanations for the decisions that are taken [Dee19]. However, another aspects that must be properly established is the requirements that determine whether an AI is XAI and the form that explanations must take to be considered as valid. The form would also depend on some factors of the case, such as whether there was a disability or a death caused. Since XAI is a growing field, there is no legal framework that clarifies these aspects yet.

While the main concerns where XAI is the solution involve criminal trials with serious harm, there are many civil cases where XAI can also be helpful. One of the most relevant examples is regulations related with data protection, and in particular regulations safeguarding citizens from arbitrary automatic decisions. The most important in this regard is the *General Data Protection Regulation*[4] of the European Union. Articles 13 to 15 define a *right to explanation*, where a meaningful information about the logic involved in automated decisions is required [SP17]. Although the particular requirements are not concrete, as the technology and the jurisdiction evolves the requirements for the necessary XAI will be specified. Another aspect where AI has a significant influence is automatic profiling. This can lead to unfair situations due to social biases in many use cases, be it for racial, religious or sex. For example, when automatically accepting or declining an employee in a job offer, or when promoting user content in social media. Automatic profiling is already restricted by the GDPR when there are legal implications and, it can also be subject to the right to explanation.

[4]`https://gdpr-info.eu/art-22-gdpr/`

3.3.3 Understanding explanations in physical rehabilitation

In the context of physical rehabilitation, XAI is relevant for the same reasons as it is relevant in other medical areas: errors can lead to the death or injury of the patient. The approach for providing an explanation depends on the task being supported by XAI. When detecting dangerous positions by the patient, deep learning neural networks are an appropriate alternative [FCSO$^+$15]. Since the input of the model is generally a stream of images of the patient performing the exercise, an explanation can be provided by using deconvolution, occlusion analysis, or similar techniques. There is not such an straightforward solution for providing an explanation when making suggestions about the physical rehabilitation routine followed by the patient. This suggestion might refer to the exercises being performed, to parameters regarding those exercises, or a combination of both tasks. However, the latter is not very common. Since these problems are not image-based, the best AI techniques that can solve them while providing an explanation are based on rules. This is because rules following an if-then structure can easily be understood by humans, particularly those that have not been automatically generated but instead have been created by a human. An additional factor to keep in mind when creating a model for physical rehabilitation is that there may be patients that deviate from the average performance i.e. either the overperform or they underperform. It is important to consider both of these possibilities when designing the model so as to not exclude potential patients from using the system [BM21].

Chapter 4

Methodology

In this chapter, the methodology followed during the project will be described. First, the development methodology used is described. Then, the work schedule that has been followed is presented. Finally, the hardware and software resources that have been used during the execution of the project will be listed.

4.1 Development methodology

The development methodology is a key aspect of all software development projects. The main constraint that the development methodology must comply with is the large amount of uncertainty of the project. The uncertainty of the project originates mostly from the AI technique to be used in the DSS. The chosen AI technique should allow the automatic suggestion of a personalized routine, and the generation of explanations for said suggestions. This means that it is necessary to perform a research for selecting the AI technique, and the requirements may change as a result of the research, or during the development process. It is for this reasons that an agile methodology is preferable to a traditional methodology.

Agile methodologies are methodologies focused on adaptability rather than in predictability. The guiding principles of agile methodologies are defined in the *Manifesto for Agile Software Development*[1]. The four main principles of the *Agile Manifesto* are as follows:

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan*

Agile methodologies achieve a high level of adaptability by prioritizing the elements on the left over the elements on the right.

The most popular agile methodologies include scrum, extreme programming, feature-driven development and Adaptive Software Development (ASD) [Awa05]. Firstly, scrum was discarded due to the high reliance on meetings and roles; many changes would be necessary to adapt scrum to a team with a single developer. Extreme programming was discarded for

[1] `https://agilemanifesto.org/`

similar reasons. ASD was ultimately chosen as the development methodology for the project because it is oriented towards high-changing and unpredictable projects. In addition, it is particularly lightweight, and it can easily be tailored to a development with a single developer.

4.1.1 Adaptive Software Development (ASD)

ASD was created by James A. Highsmith [Hig00]. Planning is seen as a paradox in adaptive contexts, since it is not possible to create successful plan in a fast-changing environment. Therefore, in ASD planning is substituted by a speculate-collaborate-learn life cycle (see Figure 4.1). The three phases are non-linear and overlapping, and they are as follows:

- **Speculate.** This phase replaces traditional planning phases. During the speculate phase, the project mission is defined. The project mission encompasses the goals and requirements of the project.
- **Collaborate.** In order to deal with the unpredictable aspects of a project, it is necessary to create a collaborative environment for emergent behavior. In the particular case of this project, collaboration happens through weekly meetings, where the state of the project, and the possible solutions to problems are discussed.
- **Learn.** In ASD it is essential to admit and react to mistakes. In this phase all stakeholders, including developers and customers, examine their assumptions about the project. This may lead to changes in the requirements.

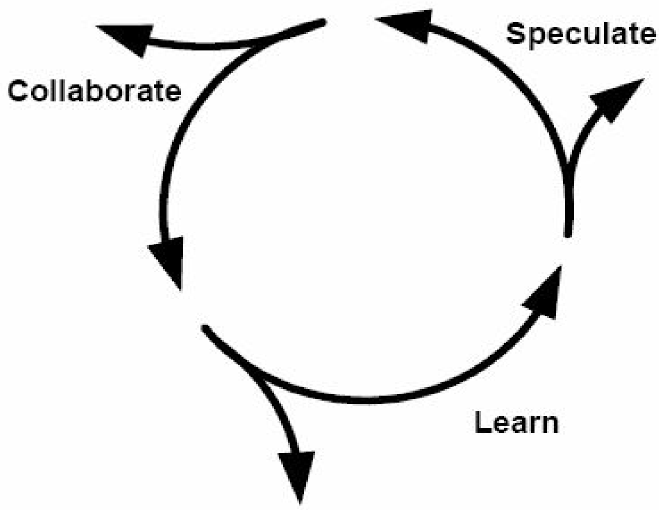


Figure 4.1: ASD life cycle. Image obtained from: [Awa05].

4.2 Work distribution

In this section, the work distribution of the project is presented. In particular, the objectives of the project were initially divided into multiple work packages. However, no fixed planning was defined, in accordance with the development methodology. As the project was executed, work packages were added, removed, and modified in the speculate phase of ASD (see Section 4.1.1 for more details). First the final work packages are described below, along with the time interval where they were executed. Then, a Gantt chart is presented for illustrating graphically the time distribution of the work packages (see Figure 4.2).

1. **Study the technologies, techniques and tools for implementing the DSS.** Given the limited knowledge about AI techniques applied to the difficulty adjustment of routines, and to routine definition, it was necessary to study the alternatives available for the implementation of the DSS. This study started on 24/01/22 and ended on 22/03/22.
2. **Extract knowledge for the definition of physical rehabilitation routines.** While knowledge regarding the definition of physical rehabilitation routines is not essential for every AI technique, some research was performed. Once FIS was chosen as the AI technique, the need for knowledge increased. This work package was executed from 03/02/22 to 28/03/22.
3. **Define the fuzzy rules and implement the FIS.** The most relevant task of the project consists in the definition of the fuzzy variables, and the fuzzy rules of the FIS. Additionally, the rules were progressively improved and refined after the first version of the FIS. The definition of the fuzzy rules ranged from 15/03/22 to 16/05/22.
4. **Integrate the FIS into the server side of the system.** The FIS was integrated in the server side of the system to enable the personalized routine adjustment. This task was started on 10/02/22, when the essential data for adjusting the routine was decided, and it was finished on 03/06/22.
5. **Add the personalized routine adjustment functionality to the therapist application.** The functionality for suggesting personalized routines needed to be accessible from the therapist application. This work package was executed alongside the later stages of the integration of the DSS into the backend, starting on 28/04/22 and ending on 08/06/22.
6. **Study possible techniques for generating explanations.** Generating explanations of the personalized routine is not a trivial task. Several approaches from the field of XAI were studied from 30/03/22 to 26/04/22 before defining the approach for generating the explanations.
7. **Integrate the explanations into the system.** Once the approach for generating explanations was chosen, then the DSS was modified to generate the explanations. This involved modifying the FIS, the backend and the therapist application. This work package was executed from 29/04/22 to 08/06/22.
8. **Testing of the DSS.** As the FIS was refined, it was also manually tested to ensure that it generated an output for all the input combinations, and that the suggestions were

realistic. The testing started on 21/03/22, and it ended on 27/05/22.

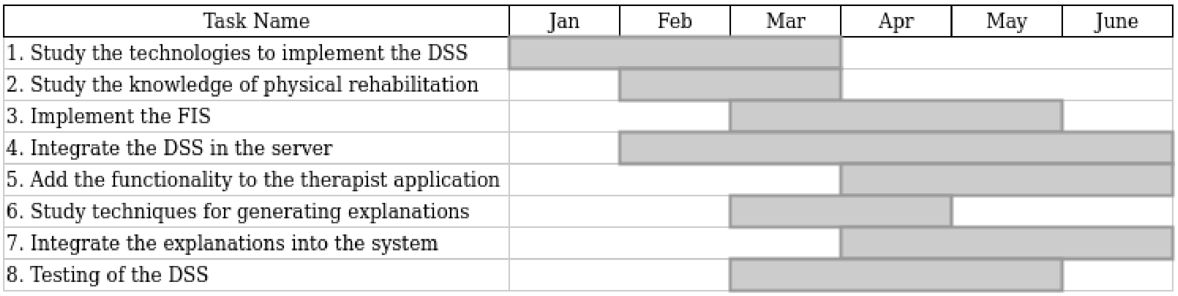


Figure 4.2: Gantt chart of the work packages.

4.3 Hardware and software resources

In this section, the resources that have been employed for the execution of the project are detail, including hardware and software resources.

4.3.1 Hardware resources

In this section, the hardware resources used during the project are discussed. The main hardware resource was the computer used during the development of the project. It is the same laptop used during the development of the system, meaning it was not purchased expressly for the project. While other systems for home rehabilitation require special sensor, such as the Kinect™, this system only needs a webcam for its usage.

4.3.2 Software resources

In this section, the software resources that have been employed for the execution of the project are detailed. The software resources have been categorized based on the role they have played in the project. The usage of many software resource was mandatory because they are used in the system, and it was used during the integration process.

Operating System (OS)

- **Ubuntu 21.10.** This OS was used during the project because of its support for many development tools. In addition, it is the preferential OS for the development and execution of the system. In the latter stages of the project, it was updated to version 22.04.

Programming languages

- **R.** R was used for developing the FIS. It was chosen because the most popular library supporting fuzzy logic in this language was mature, and because many tools for R have good support for visualization.

- **TypeScript.** Typescript was used for integrating the FIS into the system. It is a superset of JavaScript that incorporates type annotations. It is the language used in the development of the server, and of the therapist application.
- **HTML.** HTML is a standard in web development, used for defining the structure of a webpage. In particular, the HTML used is an extension provided by Angular for defining templates for the components in the application fo the therapist.
- **Syntactically Awesome Style Sheets (SASS).** SASS is used for describing the presentation of the template of components in the therapist application. It is an extension of Cascading Style Sheets (CSS) that introduces variables.

Frameworks and software libraries

- **Angular[2].** Angular is a frontend web framework. It was used for integrating the DSS in the therapist application. It helped the development by modularizing sections of the application into components, consisting in a HTML and SASS template, and a TypeScript class for defining the logic.
- **Angular Material[3].** Angular Material is a software library that provides ready-to-use components for the creation of the UI. Since they are Angular components, they can easily be integrated in the therapist application. They have been chosen for their high degree of usability and for consistency along the application.
- **NestJS[4].** NestJS is a framework for developing server side applications with TypeScript. It facilitates the definition of endpoints in a modular way.
- **TypeORM[5].** TypeORM is an ORM for Typescript. As an ORM, its role is to simplify the process of accessing and modifying the data in the database by mapping data from the database to objects.
- **sets[6].** *sets* is a R library that supports operations with general sets, and in particular, fuzzy sets. It has been used for implementing the FIS.
- **RJSONIO[7].** In order to easily integrate the FIS in the system, the input and output of the FIS were serialized and deserialized using the JSON format. While TypeScript supports JSON natively, in R the library *RJSONIO* was used for this purpose.
- **r-integration[8].** In order to integrate the FIS implemented in R into the system, the library *r-integration* was used. It enables the execution of R methods from JavaScript.

[2] https://angular.io/
[3] https://material.angular.io/
[4] https://nestjs.com/
[5] https://typeorm.io/
[6] https://cran.r-project.org/web/packages/sets/
[7] https://cran.r-project.org/web/packages/RJSONIO/
[8] https://www.npmjs.com/package/r-integration

Development tools

- **Docker[9].** Docker is a tool that allows the virtualization at a OS level into units called containers. This minimizes issues related to the environment where the system is executed. In the system, it is used for virtualizing the therapist application, the server, and the database.
- **Git[10].** Git is a version control system used for tracking changes in source code files, and for coordinating work between developers.
- **Bitbucket[11].** Bitbucket is a service for hosting Git repositories. It is used for saving the progress of the project, since it is the hosting provider for the source code of the system.
- **Visual Studio Code[12].** Visual Studio Code is the Integrated Development Environment (IDE) used for the integration of the FIS in the system. It supports many programming languages through extensions, such as TypeScript, SASS and the Angular extension of HTML.
- **RStudio[13].** RStudio is the IDE used for the development of the FIS. It was chosen for its ease of use and for its support of visualizing the fuzzy variables.
- **PostgreSQL[14].** PostgreSQL is an open-source relational Database Management System (DBMS). It is the same DBMS as the one used by the system, and it is virtualized using Docker.

Documentation tools

- **Overleaf[15].** Overleaf is a web-based LaTeXeditor which has been used for writing and saving the documentation of the project.
- **draw.io[16].** *draw.io* is a web-based tool for the creation of diagrams. It was chosen for its ease of use and for its support for Unified Modeling Language (UML) diagrams. It has been used in the creation of many diagrams in the documentation.

[9] https://www.docker.com/
[10] https://git-scm.com/
[11] https://bitbucket.org/
[12] https://code.visualstudio.com/
[13] https://www.rstudio.com/
[14] https://www.postgresql.org/
[15] https://www.overleaf.com/
[16] https://app.diagrams.net/

Chapter 5

Architecture

This chapter describes the architecture of the system consisting of modules which are interrelated with each other. Firstly, a general description of the DSS and of the system the DSS is going to be integrated into is provided. Then, each one of the modules, which can be seen as subsystems, are described in detail, along with the decisions taken during their development. The decisions were taken after studying the problems that appeared, and pondering the advantages and disadvantages of the possible solutions. In addition, the design pattern used in the project are discussed.

5.1 General overview

The DSS is integrated into a commercial system in development with a broader scope. The purpose of this system is supporting the physical rehabilitation process of stroke patients. The author of the project has taken part in the development of the system, with a particular focus in the development of the backend. The architecture of the DSS emulates the architecture of the complete system, which is a web-based client-server architecture. In particular, there are two client applications, and there is a single server application. One of the clients is designed for therapists while the other is for patients. The purpose of the therapist application is to manage the patients, and their rehabilitation routines, and to overview their progress. The purpose of the patient application is mainly to guide the movements of the patient while they perform the rehabilitation routine. To this end, AR and computer vision are used, as well as gamification techniques. To execute the client application, no sensor other than a webcam is required.

Since it is a client-server architecture, each side of the system can be considered as its own isolated system with an independent architecture. The architectures of both sides are layered architectures [Sch96]. This type of architecture was chosen because it is the same as the architecture of the complete system, meaning integration is easier, and because it is low in coupling and high in cohesion. The three main layers that constitute the complete system are the presentation layer, the business layer and the persistence layer. Strictly speaking, it is also possible to include in the architecture an extra layer whose responsibility would be handling networking tasks. However, given that only minimal modifications have taken place in this layer no in-depth description is provided. A graphical description of the system architecture can be seen in Figure 5.1.

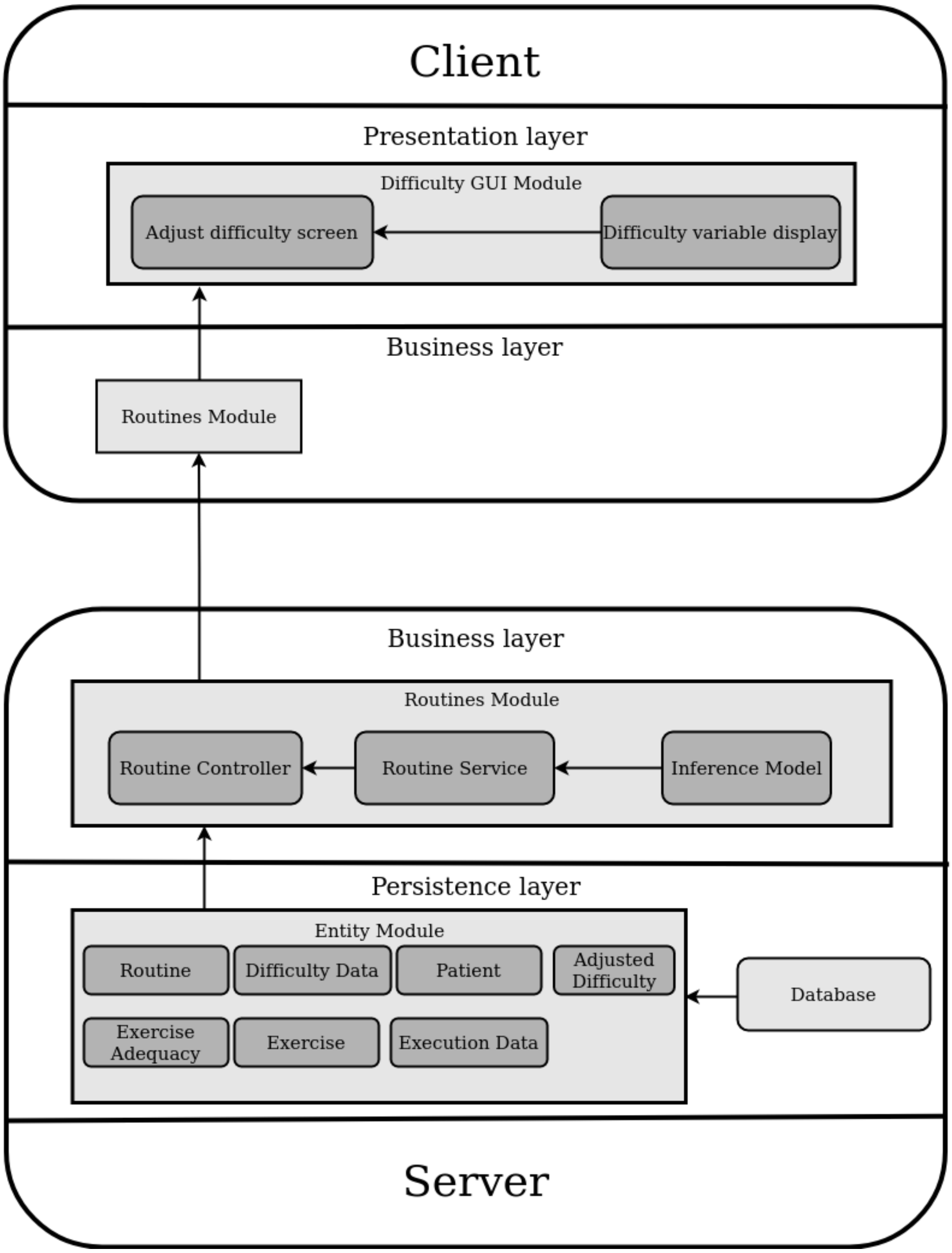


Figure 5.1: DSS architecture diagram.

The responsibilities of the layers are as follows:

- **Presentation layer**. This layer is exclusively included in the client-side architecture. It is in charge allowing the user to interact with the system. Most of the work in this layer has consisted in creating a section in the therapist web application where the therapist can request, review and validate suggestions for adapting the difficulty of a routine. Additionally, some disperse changes have been made in the Graphical User Interface (GUI) to adapt to the changes in the database schema, such as including new fields in the form for registering patients.

- **Business layer**. This layer is implemented by both client and server applications. It is in charge of encoding the domain logic and communicating the other two layers. It is the most relevant layer of the DSS, since it includes the inference model used for making the recommendations. Additionally, a service was implemented to serve as an intermediary between the model and the system, as well as for transactional tasks.
- **Persistence layer**. This layer is exclusively included in the server-side architecture. It is in charge of storing the data generated during the execution of the system in a non-volatile format, as well as retrieving said data. The tasks performed here mainly consisted in adapting the original database schema of the system to store the necessary data for the DSS.

To deploy the system, Amazon Web Services[1]. Using their services greatly simplifies the deployment process, since the hardware architecture is abstracted. In addition, their services substitute the initial investment and the costs of maintaining a server infrastructure by a pay-as-you-go model. Furthermore, this approach ensures that the system is fault-tolerant, and that the system availability is over 98%. The main service used for this project is AWS Elastic Beanstalk[2].

5.2 Persistence layer

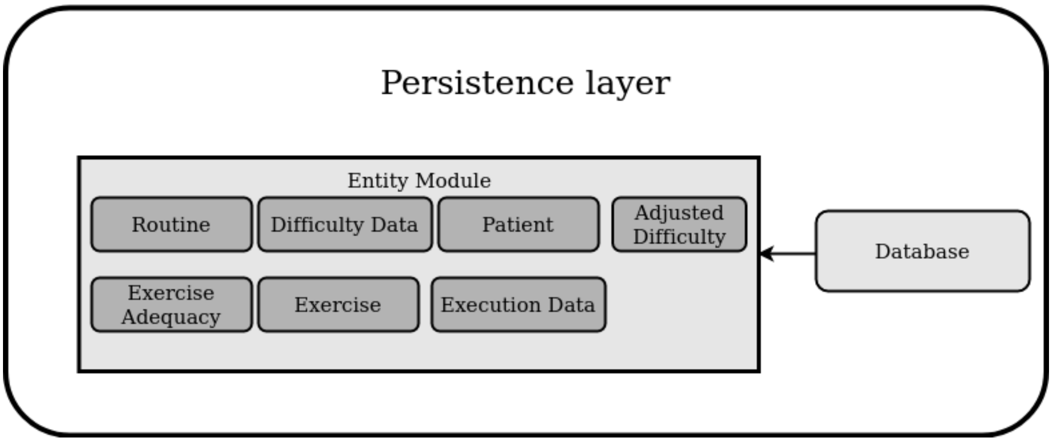


Figure 5.2: Persistence layer.

The persistence layer is in charge of storing the data generated during the execution of the system in a non-volatile format. Additionally, the persistence layer also supports querying the stored data, as well as performing update or delete operations. In particular, the data

[1] `https://aws.amazon.com/`
[2] `https://aws.amazon.com/ec2/`

stored and queried by the DSS includes the routines, their execution data and the suggested adjustments for the routines.

Since the DSS is integrated in another system, the chosen technologies for persistence were the technologies used in the global system, for the sake of simplicity. Firstly, a relational database was used for storing the data. PostgreSQL[3] was used as the DBMS. In order to access the database from the server, a Object–Relational Mapping (ORM) was used. An ORM is a tool that enables the manipulation of the database through an Object-Oriented paradigm. An ORM is used because it abstracts the details of the chosen database technology. Additionally, it simplifies common tasks that are necessary when using a database, such as preventing SQL injections, or managing SQL connections. The ORM used in the DSS is TypeORM[4], which is the same as the ORM used in the system.

There are two main design patterns that can be used for implementing an ORM: *Data mapper* and *Active record* (see 5.5.3 for more details). Both are supported by TypeORM. The *Data Mapper* alternative was chosen in order to be consistent with the global system. Another feature supported by TypeORM is migrating from one database schema to another. Although the system has been deployed for testing purposes, it is not currently available to the public. Therefore, no migration was implemented for changing the schema, since the schema could be recreated from scratch without losing data. The only migrations that were implemented were for adding the difficulty of the exercises and for seeding sample data.

To integrate the DSS, it was necessary to make use of some use some tables in the original schema and expand it with new tables. The following are the relevant entities for the DSS:

- **Routine**. This entity represents the routines of the patients. The relevant data to the DSS is the the start date of the routine, the end date of the routine and the weekdays in which the routine is supposed to be executed. Although currently a routine may be assigned to multiple patients, that is not the final behavior and it will be changed in the future as other features are added to the global system.
- **Patient**. The relevant data of the patients is essentially the side where the patient had the stroke.
- **Exercise**. The relevant data of the exercises is the stroke side that the exercise is aimed at, the waypoints, which in the DSS are used for determining the difficulty, and the difficulty itself.
- **Routine to Exercise**. This entity represents a join table between routines and exercises in the database, and it is used to store exercise data which is specific to the routine. These are the available time to execute a set, the rest between sets, the number of sets and the number of repetitions per set.

[3] `https://www.postgresql.org/`
[4] `https://typeorm.io/`

- **Routine Execution Data**. This entity represents the execution of one set of one exercise of the routine assigned to a patient. It includes the time spent by the patient executing the set, the number of repetitions actually executed, and the completion i.e. the ratio between the executed repetitions and the total number of repetitions. Additionally, the timestamp when the patient hits each waypoint was also initially stored in order to capture all the generated data. However, it is not used in the model.

- **Patient Difficulty Data**. This entity represents the patient data that is specific to the DSS. Since it is expected that the patient entity will grow over time, this data was extracted into a new entity. The most important data for the inference model is the performance and the mobility. The date of the last inference is also stored to avoid performing inference twice over the same execution data.

- **Exercise Adequacy**. This entity represents the adequacy of an exercise for a patient. Since that data depends on both the exercise and the patient, a new entity was created for this purpose.

- **Routine Adjusted Difficulty**. This entity represents the adjusted difficulty of a routine, and it is stored in the database until the therapist either accepts it or discards it. The updated performance of the patient and adequacy of the exercises is stored in this entity. The exercise configurations of the new routine are also stored i.e. the number of repetitions, sets and the time. Finally, the explanations provided by the model are also stored. In this entity, many of the columns are not properly atomic columns, but instead they are of type *jsonb*. This type is a binary variant of the *JavaScript Object Notation*[5], that allows many properties in a single cell of the table. While choosing this data type makes this table non normal, it was chosen for its flexibility. The alternative would have required the creation of multiple join tables for storing all the inference data.

5.3 Business layer

This layer is mainly responsible for adjusting the difficulty of the routine. Additionally, various transactional tasks are necessary to support the complete workflow of the adjustment. Both on the client side and on the server side of the architecture there is a *Routines Module* that has been expanded to include the DSS. The module in the client is a small service for connecting to the server, while the module in the backend has three parts: a controller for handling the endpoints, a new service for the adjustment of the routines and the inference model.

[5] `https://www.json.org/json-en.html`

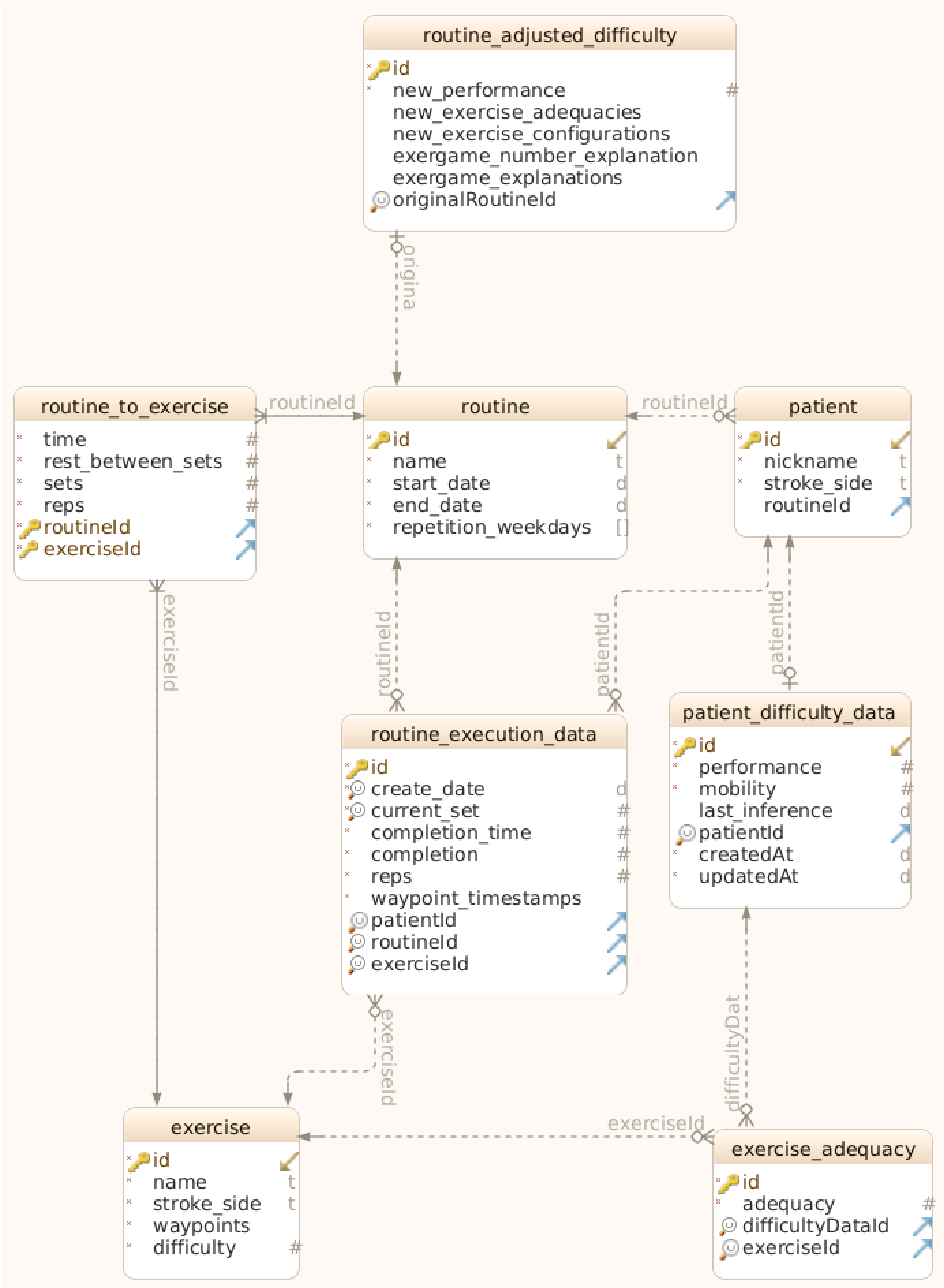


Figure 5.3: Entity–Relationship (ER) diagram of the entities and columns relevant for the DSS.

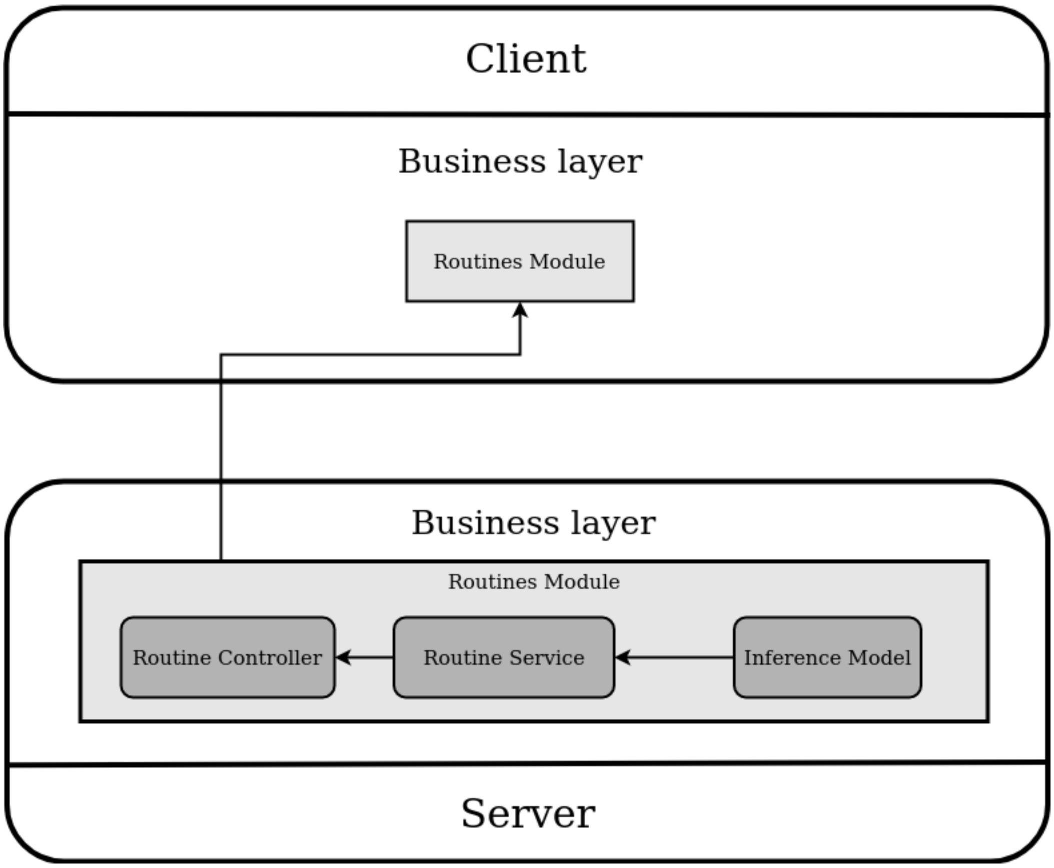


Figure 5.4: Business layer.

5.3.1 Recommender Module

The inference model is a key piece in the DSS, as it is in charge of adjusting the routine according to the execution data, namely the number of repetitions, and the time spent performing exercises. There were two candidate techniques for implementing the inference model. The first technique that was considered was the use of ANNs or another supervised learning techniques. The main benefits of ANNs is that they are highly effective. However, it requires labeled data in order to train the model. Since the model requires data specific to this particular system and the system has not been used by real patients, there is no dataset for training the model. Generating synthetic data by introducing small variations to real data was considered as well, but with no real execution data available it was not a viable approach. Additionally, implementing an explainable ANN is a complex task, and explainability is a relevant aspect of the DSS. The alternative technique for implementing the inference model was using a FIS. This technique was considered because fuzzy logic has successfully been used in the medical field for tasks such as diagnosis [FCSO+15]. Fuzzy logic was ultimately chosen as the technique used for the model. One of the main reasons is the capacity of FISs for dealing with uncertainty. Dealing with uncertainty is common in healthcare; it is not possible to know all the data about the patient with precision before making a diagnosis or recommending a treatment. For this reason, the chosen AI technique needs to deal with a lack of information. Additionally, FISs are highly interpretable, meaning it is easy understand the reasons for the output they generate. This is because FISs are based on fuzzy rules, which have a if-then structure. This structure is similar to how humans reason. This last factor is of particular relevance because the DSS needs to be explainable, and the approach for generating explanations is highly dependent on the degree of interpretability of the AI technique. Finally, ANNs were not ultimately a viable alternative, since not enough labeled data was available.

It is not trivial to define the fuzzy rules for the adaptation of the routine. The approach for the definition of the FIS is heavily inspired by similar projects using fuzzy logic in the context of physical rehabilitation, and in particular by [GPCSA+21]. This task relies on expert knowledge in physical rehabilitation for stroke patients. However, the expert knowledge available was very limited. In particular, the indications followed in other DSS for physical rehabilitation were used as the basis of the model. The most important consideration was the need to maintain the motivation of the patient i.e. the resulting routine must not be neither too easy nor too difficult for the patient. Additionally, it has been decided that the model should be more lenient with the suggested available time, since a strict time limit can limit the number of repetitions that the patient performs, rather than incentivizing that the patient performs the exercises at a certain speed. Another factor that affects the definition of the rules and the fuzzy variables and categories is the period of the inference. Initially, inference was automatically performed daily, but the need of allowing the therapist to review the inference changed this approach. In the end, inference is performed when the therapist requests it, considering the execution data of the complete routine. To handle the variable periods, the daily average of the days when the patient performs the exercise is made before passing the data to the model.

The first step for defining the FIS is to define the fuzzy variables, along with the values the variable may take. The following are the input variables of the FIS, and their descriptions.

- **time.** This variable expresses the daily time spent executing the exercise in question. It ranges from 0 to 300 seconds.
- **completion.** This variable expresses the rate between the executed exercises and the scheduled exercises of the routine being adjusted. It is expressed as a percentage, ranging from 0 to 100.
- **reps.** This variable represents the daily number of executed repetitions of a particular exercise. The maximum daily value considered was of 50 repetitions.
- **sets.** This variable represents the daily number of executed sets of an exercise, with 10 being the maximum number of sets.
- **performance.** This variable represents the general level of performance of the patient, and it was introduced to summarize the performance of previous executions. Since this

variable is not related to any physical metric, unlike the time or the number of repetitions, it was defined from 0 to 100 for simplicity. The initial value of the performance is 50, and subsequent recommendations modify this value.

- **mobility.** This variable represents the general level of mobility of the patient as perceived by the therapist. For the same reasons as the performance, it is defined from 0 to 100. At first the possibility of using the range of motion of all the joints involved in the exercise was considered, but that data might not always be available so that approach was discarded.
- **exergame_diff.** This variable is used to complement the execution data when analyzing the parameters of exergames. For the same reasons as the performance, it is defined from 0 to 100. Initially, it was going to be a manual input from either the therapist or the developers of the general system, like the mobility. However, a minimal set of fuzzy rules were created for determining the difficulty according to the waypoints. The waypoints are points located in a unitary square that are used for guiding the movements of the patient in the system.
- **waypoint_number.** This variable represents the number of waypoints an exercise has for each repetition. It ranges from 2 to 10.
- **waypoint_distance.** This variable represents the average distance between consecutive waypoints for a given exercise. It ranges from 0 to 1, since the waypoints are located in a unitary square.

The following are the output variables of the FIS, and their descriptions.

- **rep_incr.** This variable represents the adjustment to the number of scheduled repetitions of a particular exercise, which is an increment when the value is positive or a decrement when the value is negative. An increment has been chosen as the output variable for many aspects of the routine because generating absolute values might lead to abrupt changes if the execution was an outlier. It ranges from -40 to 40.
- **set_incr.** This variable represents the adjustment to the number of scheduled sets of a particular exercise. It ranges from -3 to 3.
- **time_incr.** This variable represents the adjustment to the allocated time for the execution of an exercise. It ranges from -120 to 120.
- **performance_incr.** This variable represents the contribution of an exercise to the adjustment of the performance. The total adjustment of the performance is the average of the contributions by all the exercises. It ranges from -20 to 20.
- **adequacy_incr.** The adequacy represents how adequate an exercise is for the patient, and it ranges from 0 to 100. It is not an input to the FIS, but it is used when the personalized routine is created. This variable represents an adjustment to the adequacy of an exercise for a patient, ranging from -20 to 20.

- **exergame_number.** This variable represents the number of exercises that will be included in the personalized routine, and it ranges from 1 to 10.

In addition to the minimum and maximum values that each fuzzy variable may take, it is also necessary to define the fuzzy sets within each variable. To this end, a systematic approach was taken for the definition of the fuzzy sets, called fuzzy partitioning. For each fuzzy variable, five fuzzy sets were defined: VL (very low), L (low), M (medium), H (high), and VH (very high). The fuzzy membership function of all of the fuzzy sets is a cone, meaning there is a value where the membership is 1, and the membership decreases linearly at the same in both directions until it reaches a level of membership of 0. The fuzzy sets are distributed uniformly, with VL centered at the lowest value possible for the variable and VH centered at the highest value possible. The radius of the cones are large enough to have intersections between cones, since this reduces the total number of rules necessary to generate an output for every input combination. An example of a fuzzy variable can be seen in Figure 5.5. It is worth noting that initially, variables that were increments were modeled with three fuzzy sets, two trapezoids at the extremes and a triangle at the middle. However, this approach limited how expressive rules could be because there were only three fuzzy sets, and it did not provide any practical advantage, so it was discarded.

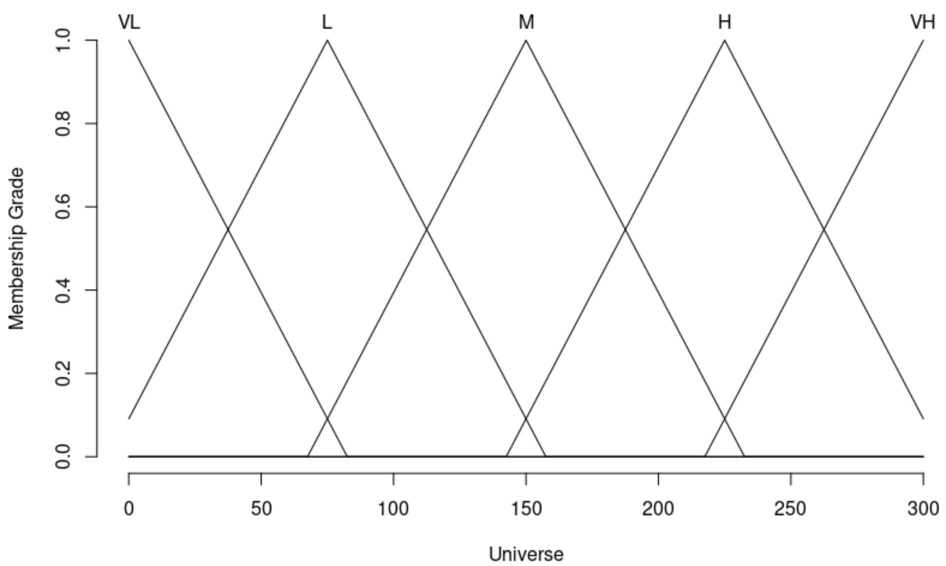


Figure 5.5: Sample fuzzy variable with a fuzzy partition.

The major challenge faced when defining fuzzy rules was that the FIS deals with two problems simultaneously: determining which exercises should comprise the routine and adjusting the parameters of the exercises that have been executed. To this end, rules were defined with one of the following goals in mind:

- **Adjusting a parameter of an individual exercise** according to the previous execution. The parameters include the *number of repetitions*, the *available time for performing the exercise*, the *number of sets* and the *performance* on that exercise.
- **Determining how adequate is an exercise for a patient.** For this goal another variable similar to the performance was introduced, called the *adequacy*. This variable has a lower value when the exercise is too difficult or too easy for the patient.
- **Selecting the number of exercises of the routine.** The number of exercises, along with the adequacy will be used outside of the model for determining the exercises that will comprise the new routine i.e. the exercises that have a higher adequacy will be chosen.
- **Determining the difficulty of the exercise.** While the difficulty of exercises was initially an input to the FIS, it is now calculated using fuzzy rules. These fuzzy rules have *waypoint_number* and *waypoint_distance* in the antecedents.

The algorithm used for generating the personalized routine is described in Algorithm 1. Another alternative where the output of the fuzzy rules is an action for modifying either the routine or one of the exercises was also considered, but it was ultimately discarded due to the complexity in defining the fuzzy rules. Some sample rules are shown below. For the complete list of fuzzy rules, see Code Listing A.1.

- fuzzy_rule(time %is% VL && completion %is% VH, rep_incr %is% VH)
- fuzzy_rule(performance %is% M && mobility %is% VL && exergame_diff %is% H, rep_incr %is% L)
- fuzzy_rule(time %is% VL && completion %is% M, performance_incr %is% H)

In addition to the theoretical definition of the FIS, it is also necessary to decide the technology used for the implementation. There are several libraries that can operate with fuzzy logic and with fuzzy inference, but some of them were troublesome. Libraries were considered in three different programming languages: *Python*, *R* and *Typescript*. *Python* and *R* were considered because many relevant libraries for fuzzy logic were available in this languages. *Typescript*[6] was considered because the backend architecture of the complete system is implemented in *Typescript* with *Node.js*[7], therefore there would be no issues due to the integration of different languages. Another factor that was considered was whether the library provided a way of accessing the execution of the fuzzy inference for generating the explanation, but none of the reviewed libraries provided such functionality. In the end, the *R* library *sets*[8] was chosen because it was considered more mature than the other libraries.

[6] `https://www.typescriptlang.org/`
[7] `https://nodejs.org/en/`
[8] `https://cran.r-project.org/web/packages/sets/index.html`

Algorithm 1 Generation of the personalized routine.

Input exercises history, patient data, exergames parameters
Output routine, final performance

$performance\ accumulator \leftarrow 0$
$exergame\ number \leftarrow$ fuzzy inference of exergame number(patient data)
for each exercise history in exercises history, exergame parameters in exergames parameters **do**
  $exergame\ increments \leftarrow$ fuzzy inference of exergame data(exercise history)
  $exergame\ parameters \leftarrow exergame\ parameters + exergame\ increments$ ▷ The increments are applied to the parameters of the exercise
  $performance\ increment \leftarrow$ fuzzy inference of performance increment(exercise history)
  $performance\ accumulator \leftarrow performance\ accumulator + performance\ increment$
end for
$sorted\ exergames \leftarrow$ Sort exergames according to their new adequacy
$routine \leftarrow$ Pick the top *exergame number* exergames from *sorted exergames*
$final\ performance \leftarrow performance\ accumulator/length(exercises\ history)$

This library uses the mamdani system for performing inference. Additionally, tools such as RStudio[9] were used during the development of the model for their visualization tools.

Generating explanations

To ensure that the FIS was explainable, it was necessary to minimize the number of rules and their complexity. To minimize the complexity of rules, they were limited to 3 variables in the antecedent joint by t-norms. This allows the presentation layer to represent the rules intelligibly. To minimize the number of rules in the system, rules were added incrementally until the system provided an output for every input and that the output was reasonably realistic. Limiting the number of rules leads to a lower effectiveness, but the relevance of explainability in this DSS means that it is a valid trade-off. For checking that for every input the system provided an output, a small *R* script was created. This script was later integrated into the system as a new test. While the combinatorial explosion leads to high execution times, which violates one of the Fast, Isolated, Repeatable, Self-validating, Timely (FIRST) principles, it is a valuable tool to check the lack of errors when modifying the model.

The first approach for enabling the model to provide an explanation was to manually add an explanation to each rule, and then select the associated explanation to the most relevant rule. When performing fuzzy inference, a fuzzy set is produced as the output for the activation of each rule, and the total output is the result of applying the t-conorm to all the individual outputs. The individual output of each rule is used for determining the most relevant rule. There is no individual membership value for each fuzzy set, but a mapping from values to membership values. In theory, the relevance of any given rule is determined by

[9] `https://www.rstudio.com/`

the defined integral of the membership function of the fuzzy set over the entire domain of the variable. In practice however, fuzzy sets are implemented in a discrete way, meaning the integral becomes the addition of all of the membership values. This strategy is described in Algorithm 2. Additionally, since the library does not provide information about the individual output of each rule, this process is performed separate from the inference process. Although the selection of a rule for an explanation worked well, the manual description of the rules involved a high workload. For this reason, the explanations are generated from the rule itself. The model is in charge of providing a string representation of the rule, while the parsing of the rule is performed outside of the model. Due to the simplicity of the language describing the rules, the parsing could be performed only with regular expressions. The language is described below.

rule ::= **list(antecedent=** clause {{ clause }}**,consequent=**clause**)**
clause ::= variable**%is%**value
variable ::= **time** | **completion** | **reps** | **sets** | **performance** | **mobility** | **exergame_diff** | **rep_incr** | **set_incr** | **time_incr** | **performance_incr** | **adequacy_incr** | **exergame_number** | **waypoint_number** | **waypoint_distance**
value ::= **VL** | **L** | **M** | **H** | **VH**

Algorithm 2 Generation of the explanation for the decisions of the DSS.

Input rules, inference input
Output relevant rule

$max\ relevance \leftarrow -\infty$
$relevant\ rule \leftarrow null$
for each rule in rules **do**
  $fuzzy\ set \leftarrow fuzzy\ inference(rule, inference\ input)$
  $relevance \leftarrow reduce(fuzzy\ set, +)$ ▷ All the membership values are added to obtain the relevance
  if $relevance > max\ relevance$ **then**
    $max\ relevance \leftarrow relevance$
    $relevant\ rule \leftarrow rule$
  end if
end for

5.3.2 Support for routine recommendation

The support for the routine recommendation in the system required from development work in both the server, and in the therapist application. The work consisted in the creation of endpoints in the server that expose the functionality of recommending the routine, and in accessing said said functionality by the therapist application. For the implementation of the endpoints, NestJS[10] has been used. NestJS is a Node.js framework, and this technology was chosen because it is mature, reliable, and scalable. The following four endpoints were

[10] https://nestjs.com/

defined for the integration of the DSS:

- *getDifficultyData*. It fetches the adjustment of a routine if it has already been created.
- *adjustRoutineDifficult*. It creates the suggestion from the execution data previously stored in the database. The task performed in this endpoint can be divided into three steps:

  1. Fetch the execution data from the database and preprocess it.
  2. Invoke the FIS.
  3. Store the inference in the database.

  Initially, the routine was updated when the suggestion was created, but that approach offers little control to the therapist so it was changed. Additionally, tasks that cannot easily be performed within the model are performed in this step. Some tasks that were extracted from the FIS include, only allowing a patient to perform exercises according to the stroke side, or parsing the rules for the explanations.
- *acceptDifficultyAdjustment*. It applies the adjustment of a routine that was previously executed, and is stored in the database. After updating the routine with the adjustiment, the adjustment is deleted.
- *cancelDifficultyAdjustment*. It discards the adjustment of a routine if it had previously been executed

Since the model is written in the *R language* and the server is implemented in *Typescript*, connecting them is not as simple as importing a library. Two solutions were evaluated: creating a socket in R and using network protocols for communication, or executing R from the server and using the *stdin*, *stdout* and *stderr* for communication. The downside of creating a socket is that it complicates the deployment process, but it is a more versatile option. Executing R from the server is simpler but less flexible. Additionally, this solution is more dependent on the environment in which the system is deployed i.e. the OS may change and that affects the way paths and commands are written. The second solution was chosen because the environment is controlled by using Docker[11] and there is a library called *r-integration*[12] that makes the process of calling R methods from Node.js straightforward. *r-integration* was chosen over *r-script*[13] and other alternative libraries with the same purpose because of its ease of use, and of the frequency at which the library was being updated.

5.4 Presentation layer

The presentation layer is responsible for the interactions of the user with the system, by properly handling the input of the user and by providing relevant information. Besides the

[11] https://www.docker.com/
[12] https://www.npmjs.com/package/r-integration
[13] https://www.npmjs.com/package/r-script

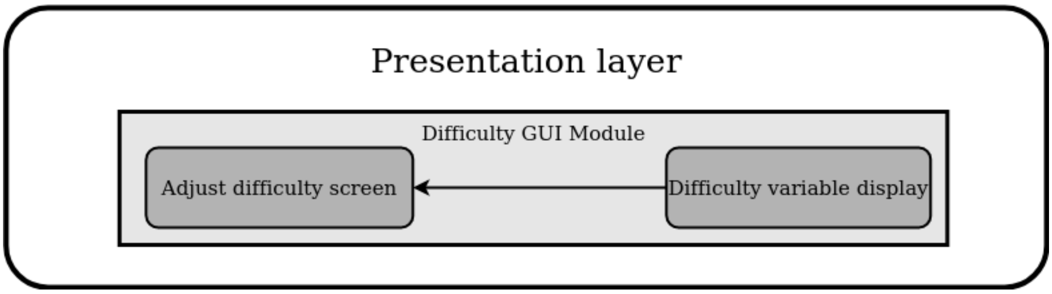


Figure 5.6: Presentation layer.

advantages of making code more modular, having a presentation layer is benefitial because it decouples business logic from the User Interface (UI), and it allows changing the UI without affecting the code for the business logic, or for the persistance. In the particular case of the DSS, it is limited to a section in the therapist web application. In this section, the therapist can request the difficulty adjustment of a routine in case no adjustment has been performed. Otherwise, the therapist can evaluate the suggested adjustment and accept or decline it.

When developing a webpage, it is possible to use a frontend framework to simplify the development process. The global system in which the DSS is going to be integrated makes use of the Angular[14] framework developed by Google. Therefore, this framework was used for the development of the presentation layer of the DSS. Other popular alternatives in the web industry include React[15] and Vue.js[16]. All three frameworks could have been chosen for the development of the presentation layer, since they share many features. Angular was chosen because it is a stable framework that integrates many features needed for the development of a web application out of the box. Additionally, it is well documented, it has good community support, and it is a mature framework.

The most important concept in the Angular framework are components. Components are classes representing a GUI element that have a HyperText Markup Language (HTML) template. They vary in size, from small widgets to complete pages. Additionally, another relevant tool in the development of the presentation layer is the Angular Material[17] UI component library. The components provided by this library are used throughout the system, so they have also been used in this development for consistency. Angular components are based on the Model View Controller (MVC) design pattern (see Subsection 5.5.1 for more details).

For the development of this project, two Angular components have been created. The most relevant one is the *Adjust difficulty screen*, which allows the patient to use the functionality for adjusting the difficulty of a routine. Since the time it takes for the system to perform

[14] `https://angular.io/`
[15] `https://reactjs.org/`
[16] `https://vuejs.org/`
[17] `https://material.angular.io/`

the inference is not negligible, this component has two different states. The first state is when no inference has been performed. In this case, the component only contains a brief explanation and a button for requesting the adjustment of the difficulty of the routine. The therapist may then wait until the inference is completed in the same screen or return at a later time once the inference has been completed. The second state is when an inference has been performed, and in this state the screen contains the suggested adjustments to the routine. If the parameters of an exercise have been modified, that is the exercise was already present in the previous routine, then the explanation for the adjustments can be revealed by expanding the *expansion panel* of the exercise. Otherwise, if an exercise was not previously found in the routine, then the explanation is replaced by a brief text detailing why no explanation is available.

The first approach for generating explanations was to generate them in natural language. This approach caused few problems and its implementation was straightforward. However, having to continuously read the explanation could make the workflow of the therapist slower, specially if the number of patients is high. Thus, it was decided that it was best to present the explanation in a visual way because it would be easier and faster to read. The antecedent of a rule are statements about the value of a variable e.g. *MOBILITY is LOW*, joined by the logic *AND* operator. The consequent of a rule is similarly the resulting value of the output variable. To represent rules visually, the value of a variable is represented with a slider, where the leftmost position represent the minimal value and the rightmost position represent the maximum value. The name of the variable is displayed with a label below. Since this representation is common to the antecedent and to the consequent, it was extracted to its own component, *Difficulty variable display*, for reusability. Since explicitly displaying the *AND* operator would needlessly make the interface more cluttered, the statements that make up the antecedent are stacked vertically i.e. the *AND* operator is implicit. Finally, the antecedent and the consequent are separated by an arrow going from the antecedent to the consequent. Since there are many explanations for each exercise, rules are delimited using UI cards, which group together the elements of the rule by adding a border. This is an example of an application of the principle of common region [Pal92].

Besides the creation of the presentation layer for the DSS, changes were made to the presentation layer of the global system to incorporate the DSS. Firstly, the *Adjust difficulty screen* was added to the routing system of the therapist web application. It was also made accessible from the list of the routines of the therapist. Additionally, some changes were required due to the new database schema. Most notably, the forms in charge of inserting a patient and modifying a patient were changed. These forms now include fields for the side where the patient had the stroke and for the mobility of the patient. The complete workflow for adjusting a routine is described in a high-level sequence diagram (see Figure 5.7).

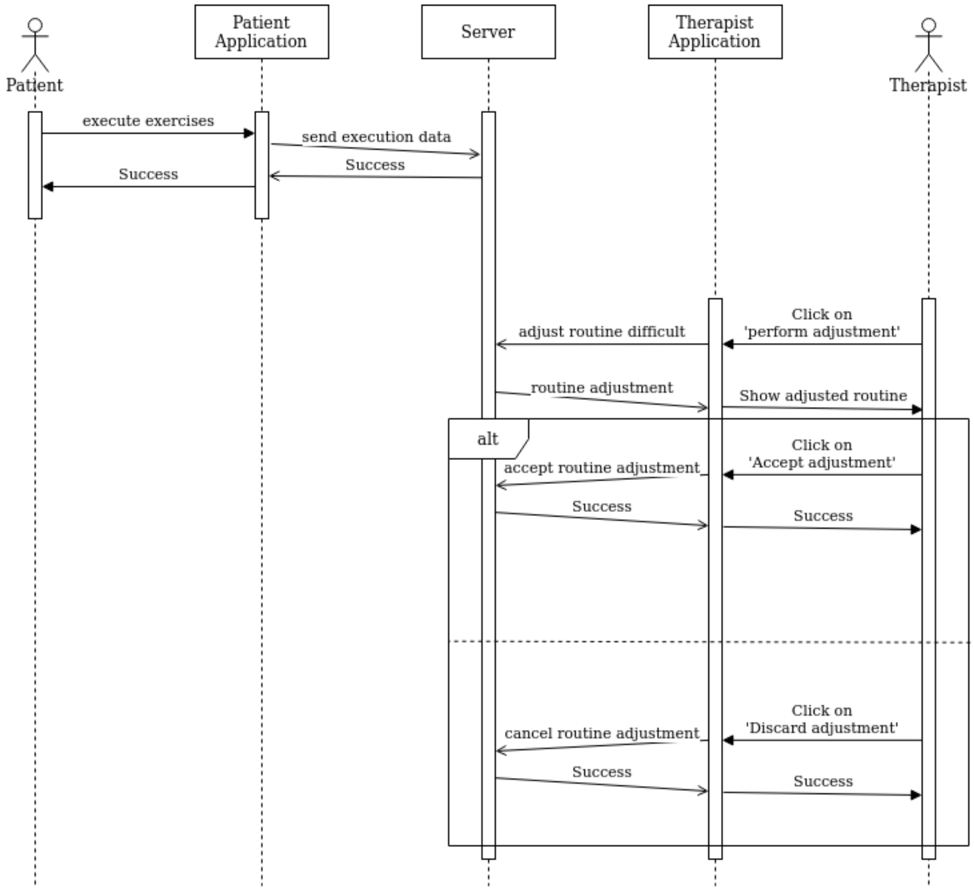


Figure 5.7: High-level Sequence diagram of the adjustment process.

5.5 Design Patterns

A design pattern is a reusable solution to a commonly occurring problem in the field of software development [GHJ98]. It is a general solution, meaning the solution is not bound to a particular programming language or technology, but instead it is a general description. In the development of the DSS, the most relevant design patterns that have been used are MVC, *dependency injection* and *data mapper*. This section consists in a description of these design patterns.

5.5.1 Model View Controller (MVC)

One of the most popular design patterns for designing and implementing interfaces is the MVC design pattern. In Angular, the MVC pattern is used for implementing components. MVC stands for model–view–controller, which are the three elements that comprise MVC. The model is the part that controls the data and the business logic, independent of the GUI. The view is a part of the user interface, displaying information from the model and inputs for modifying the model. The controller acts as a mediator between the model and the view. It can update the model according to the inputs of the view and it can modify the views in order to display updated data. The main benefit of MVC is decoupling the user interface from the data.

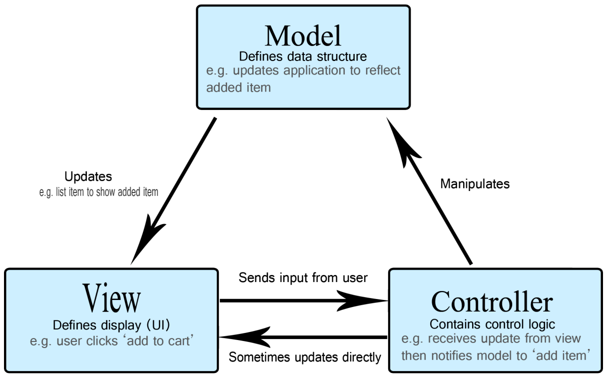


Figure 5.8: Diagram describing the interactions in the MVC design pattern.

5.5.2 Dependency injection

Dependency injection is an strategy for complying with the dependency inversion principle, which is one of the SOLID principles [Mar00]. The dependency inversion principle states every dependency in the design should target an interface, or an abstract class, to the extent that it is feasible to do so. Dependency injection complies with said principle by injecting dependencies to objects, separating the creation of dependencies with their usages. Dependencies are injected by an additional element, the injector. The general structure of dependency injection can be seen in Figure 5.9. Dependency injection can be performed through constructors, through setters and through interfaces. The most common way of injecting dependencies, and the way used in this project is through constructors. There are many frameworks that implement dependency injection, either as its main goal or as an additional feature of the framework. Both frameworks used in the development, Angular and NestJS, include dependency injection.

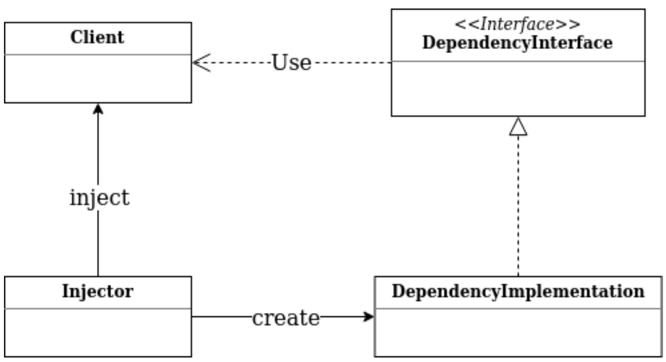


Figure 5.9: Dependency injection.

5.5.3 Data mapper

There are several patterns for mapping persistent data and objects. Two patterns are supported by TypeORM: *data mapper* and *active record Data mapper* consists in using an object (that in TypeORM is called a *Repository*) that implements the functions create, read, update, and delete. These functions operate on objects that represent entities in the database. *Active record* consists in using an object that implements the read, update and delete functions, as well as having as properties the columns of the entity being represented. This objects represents the records of the database. The main difference between the two patterns is that in the *active record* pattern the classes are coupled with the database, since each class represents a table in the database. That issue is not as relevant when using data mapper. Since the conversion is performed by the repository, the mapping between tables and classes is more flexible.

5.5.4 Adapter

The adapter pattern is a structural pattern that converts the interface of a class into another interface, which allows collaboration between classes happen that would otherwise not take place [GHJ98]. The most prominent example of this pattern in the project is the ORM. One of the advantages of using an ORM is that it abstracts the DBMS technology used. To achieve this, it is necessary to adapt the client interfaces, that is, the interfaces that are offered by the ORM, to the specific interfaces that are defined for each supported DBMS.

5.5.5 Object pool

The object pool pattern is a creational pattern that consists in keeping a set of objects in reserve, rather than creating them when they are needed, and destroying them when they are not needed. This pattern emerged to optimize the creation and destruction of objects when its performance impact is not negligible. This pattern has been used in the system with the ORM, and in particular for managing database connections. The creation and destruction of database connections requires sending messages through the network, which takes a considerable amount of time. Since connections are reusable, a pool of connections is maintained by the ORM for minimizing wait time.

5.5.6 Observer

The observer pattern is a behavioural pattern whose goal is ensuring that when an object changes its state, other dependent objects are notified of the change [GHJ98]. This pattern ensures consistency between related objects. Dependent objects subscribe to the observable object, and when the observable object modifies its state, it notifies every subscriber. In this project, this pattern is implemented by the RxJS[18] library. This library is used by the Angular module for handling Hypertext Transfer Protocol (HTTP) requests. In particular, when a HTTP request is started, an observable object is returned, and when the request is finished, the objects that subscribed to the object received the response.

[18] `https://rxjs.dev/`

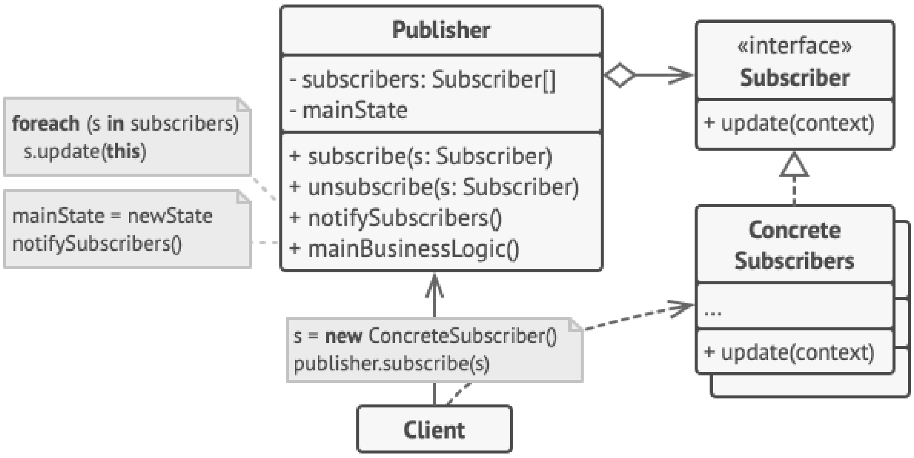


Figure 5.10: Observer diagram. Image obtained from: `https://refactoring.guru/design-patterns/observer`

5.5.7 Decorator

The decorator pattern is a structural pattern that attaches additional responsabilities to an object dynamically [GHJ98]. It is an alternative to statically adding responsibilities through inheritance. Its advantage over inheritance is that it avoids the combinatorial explosion when there are many additional responsabilities. This pattern has been used extensively in the project. Firstly, it is implemented in the classes provided by TypeORM for defining database entities. It is also used by NestJS for the definition of endpoints. Finally, it is used by the libraries *class-validator*[19] and *class-transformer*[20] for the serialization and deserialization of the requests and responses. For example, a restriction on the type of a attribute, or on the values it might take can dynamically be added with decorators.

[19] `https://www.npmjs.com/package/class-validator`

[20] `https://www.npmjs.com/package/class-transformer`

Chapter 6

Results

After discussing the technical aspects of the project, and the design decisions that were taken, the results of the development are presented, with an emphasis on the graphical appearance of the DSS and the user workflow. The system where the DSS is going to be integrated into is also shown to provide context. Examples of results of the DSS are also included. In addition, several statistics related to the project are discussed. Finally, a cost estimation of the project is also provided. The source code of the project is available on the following Bitbucket repositories:

- `https://bitbucket.org/furious_koalas/physio-galenus-therapist/`, in the branch *feature/221-expand-the-execution-data-stored-in-the-database*
- `https://bitbucket.org/furious_koalas/physio-galenus-shared/`, in the branch *feature/221-expand-the-execution-data-stored-in-the-database*

6.1 Project Context

Since this project has been executed in the context of a larger system, it is valuable to show an overview of the system. The system consists in an application for the therapist, an application for the patient and a HTTP server. The purpose of the patient application is to guide patients when they perform the rehabilitation routine using AR (see Figure 6.1). The purpose of the therapist application is overviewing the progress of patients. Using the application, the therapist can create patients, assign routines to patients, and adjust the parameters of the routine. The changes performed in the therapist application are then reflected in the patient application (see Figure 6.2).

6.2 Final Result

The starting point of the DSS workflow is the *My Routines* page (see Figure 6.3). In this page, all the routines created by the user are listed. Initially, the DSS was accessed through the page for editing routines, but it was moved here because there was already a section for actions affecting the routine. Also, there are many elements in the page for editing a routine, therefore adding another element would only make it more cluttered. In order to access the DSS, it is necessary to click the newly created button for that purpose.

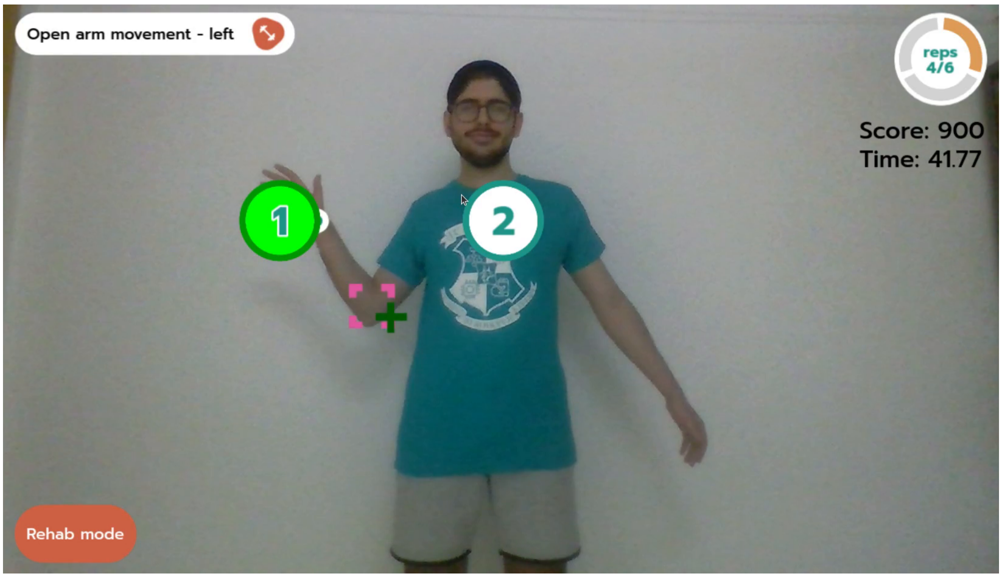
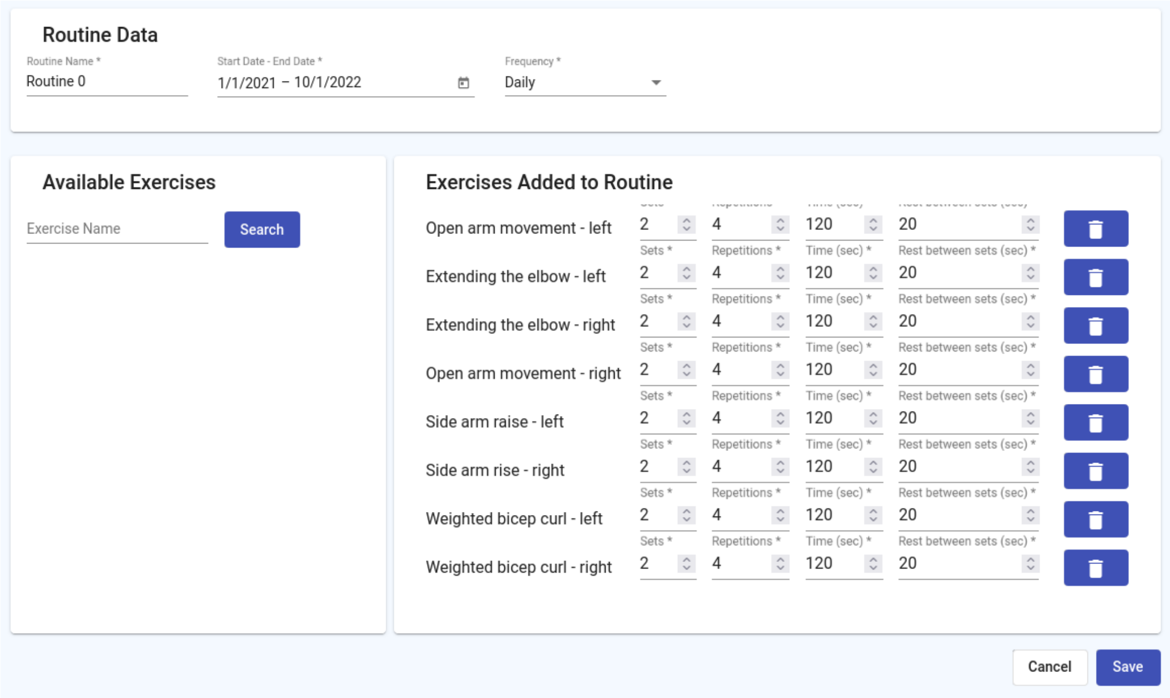


Figure 6.1: Patient application guiding the movements of the patient.

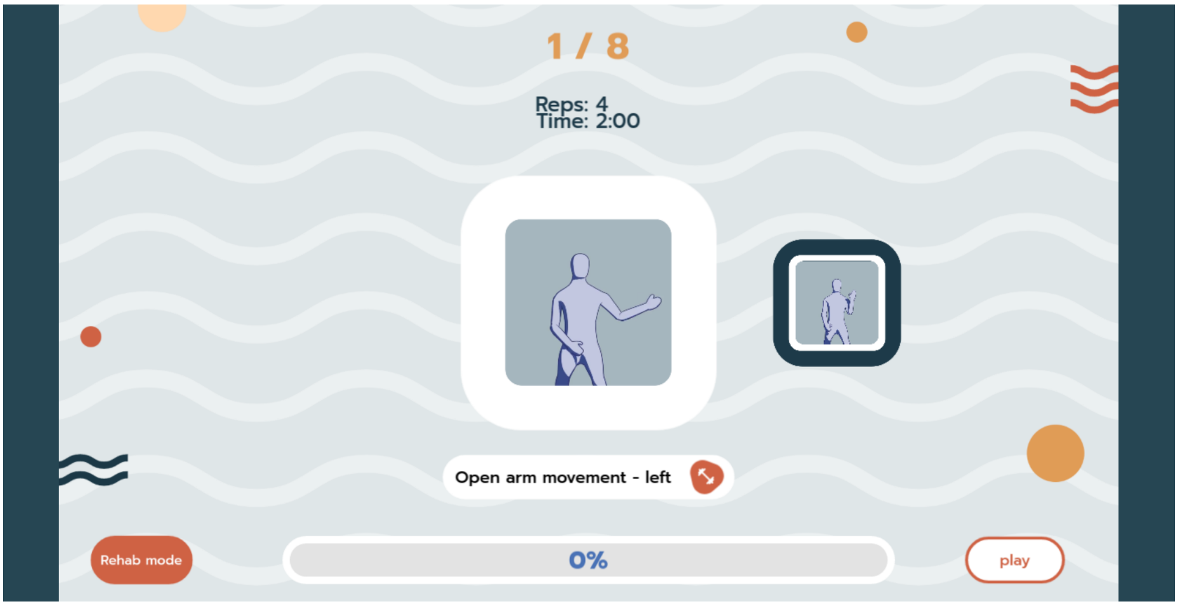
After clicking the button, the *Adjust Difficulty* page is accessed. The page for requesting an adjustment appears initially if no adjustment is pending, that is, it has not been accepted nor declined by the therapist. This page consists in a small explanatory text with a button for requesting the inference (see Figure 6.4). Once the button is pushed, the adjustment process begins. Since the adjustment may take several minutes, the user interface is not blocked by a loading spinner. Instead, the button is disabled to prevent the user from unintentionally requesting an adjustment again (see Figure 6.5). Additionally, a toast notification is created to reassure the user that the request is being processed and that the system is not stuck.

Once the adjustment has been performed, the *Adjust Difficulty* page shows the routine adjustment (see Figure 6.6). If the user left the page while the page was being adjusted, the adjusted routine will be shown when the user returns to the page. The information presented to the user about the routine adjustment is comprised by the selected exercises for the routine as well as their configurations, that is, the number of sets and repetitions, the available time for completing the exercises and the resting time between sets. At this point, the user must evaluate the suggested routine to ensure that it is correct. This is because the goal of the DSS is to support the work of therapists, not to replace them.

Since the explanations related to the work on XAI are relevant when evaluating the adjustment, they are integrated in this screen. However, the explanations are not directly shown to the user because they take up a considerable amount of screen space, especially because they are visual explanations rather than text. For this reason, exercises are listed as expansion panels. The user may reveal the explanations for a particular exercise by clicking on the expansion panel of said exercise (see Figure 6.7). There is one explanation for each adjusted parameter of the exercise. The values of the variables are represented using a slider, and the implication between the antecedent and the consequent is represented with an arrow. Exer-



(a) Therapist editing the routine of a patient.



(b) The changes in the routine are reflected in the patient application.

Figure 6.2: Description of the connection between the two client applications.

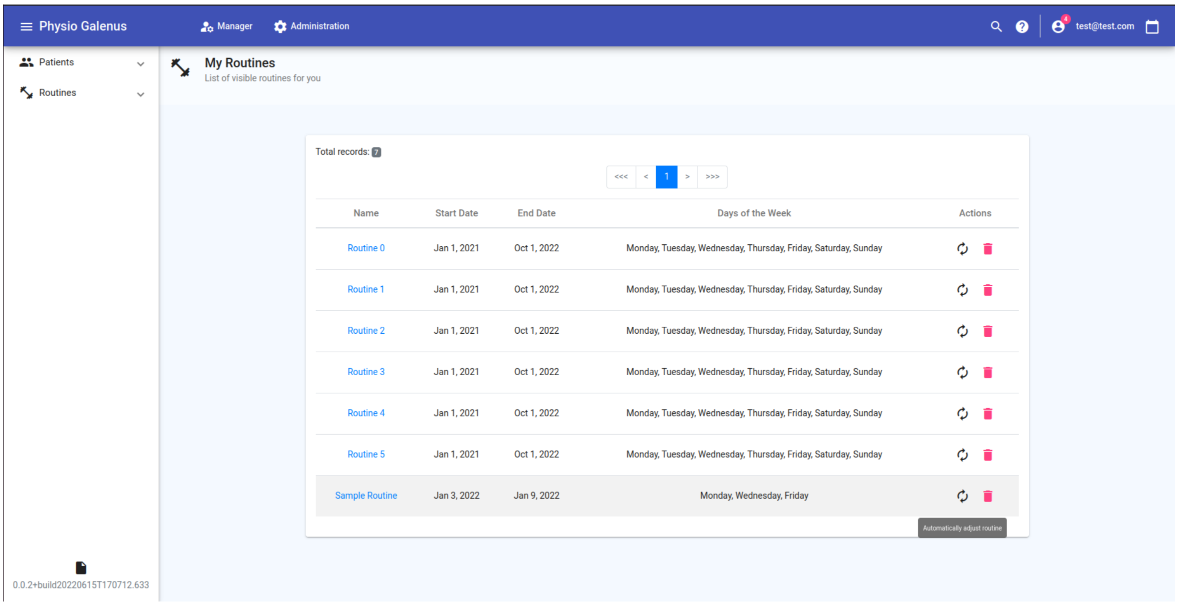


Figure 6.3: *My Routines* Page.

Figure 6.4: Initial *Adjust Difficulty* Page.

cises that were not previously found in the routine cannot be adjusted, since there is no initial value to adjust nor there is any execution data for that exercises. In this cases, the expansion panel reveals an explanatory text detailing that the exercises configuration has been generated using default data (see Figure 6.8). Once the user has reviewed the suggested routine adjustment, he may apply it, and update the routine, or discard the suggestion.

In addition to the *Adjust Difficulty* page, other minor changes were made to the system in order to fully integrate the DSS. Namely, two fields were added to the forms for inserting a patient, for showing the data of the patient and for editing the data of the patient (see Figure 6.9).

6.3 Inference Results

In this section the result of the inference process from three sample inputs are included to better illustrate the model of the DSS. All examples refer to the execution data of the same routine: a routine starting on 03-01-2022 and ending on 09-01-2022. The routine is executed every Monday, Wednesday and Friday, and the daily goal is of 2 sets of 20 repetitions for every exercise in the system (8 in total). The execution data is identical on all three days for

Figure 6.5: Adjustment being processed.

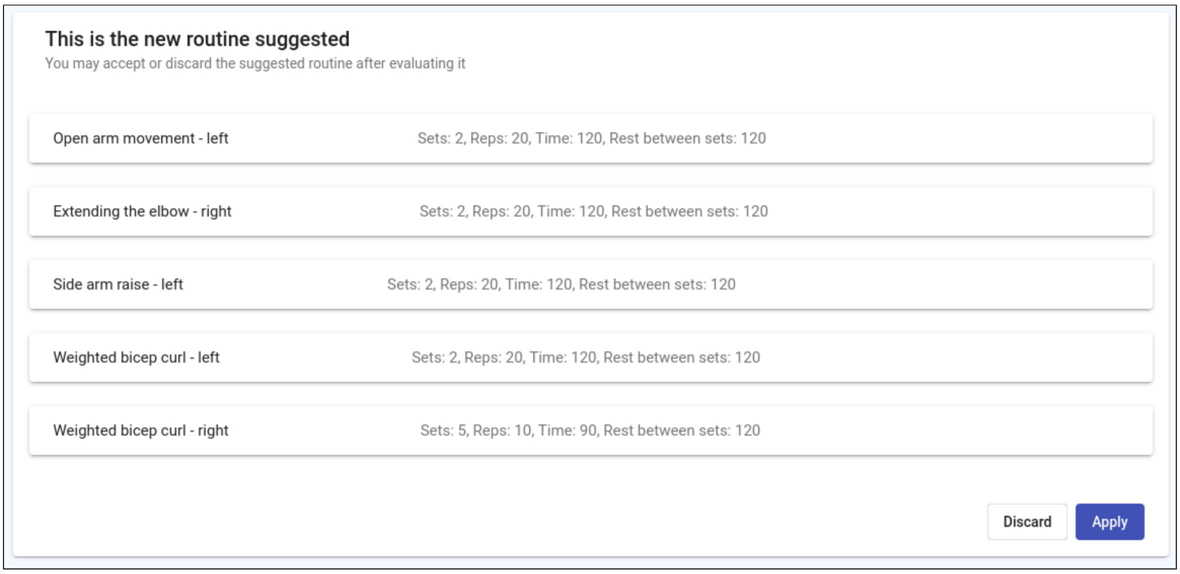


Figure 6.6: *Adjust Difficulty* Page after performing the adjustment.

the sake of simplicity. The sample execution data can be seen in Table 6.1, while the output can be seen in Figures 6.10, 6.11, and 6.12.

Number	Mobility	Performance	Sets	Repetitions (total)	Time
1	50	50	2	20	120
2	90	90	2	40	60
3	10	10	1	5	120

Table 6.1: Sample execution data.

6.4 Project Statistics

In this section, the statistics generated from the work in the two repositories will be presented. The main repository is the one containing the server-side of the application and the client application for the therapist. The other repository is a repository with shared data between the server and the client application for the patient, which is held in a different repository that has not been modified during the development of the DSS. The first alternative for extracting statistics from the repositories was using the Awesome Graphs[1] extension. This is because

[1] https://marketplace.atlassian.com/apps/1210934/awesome-graphs-for-bitbucket

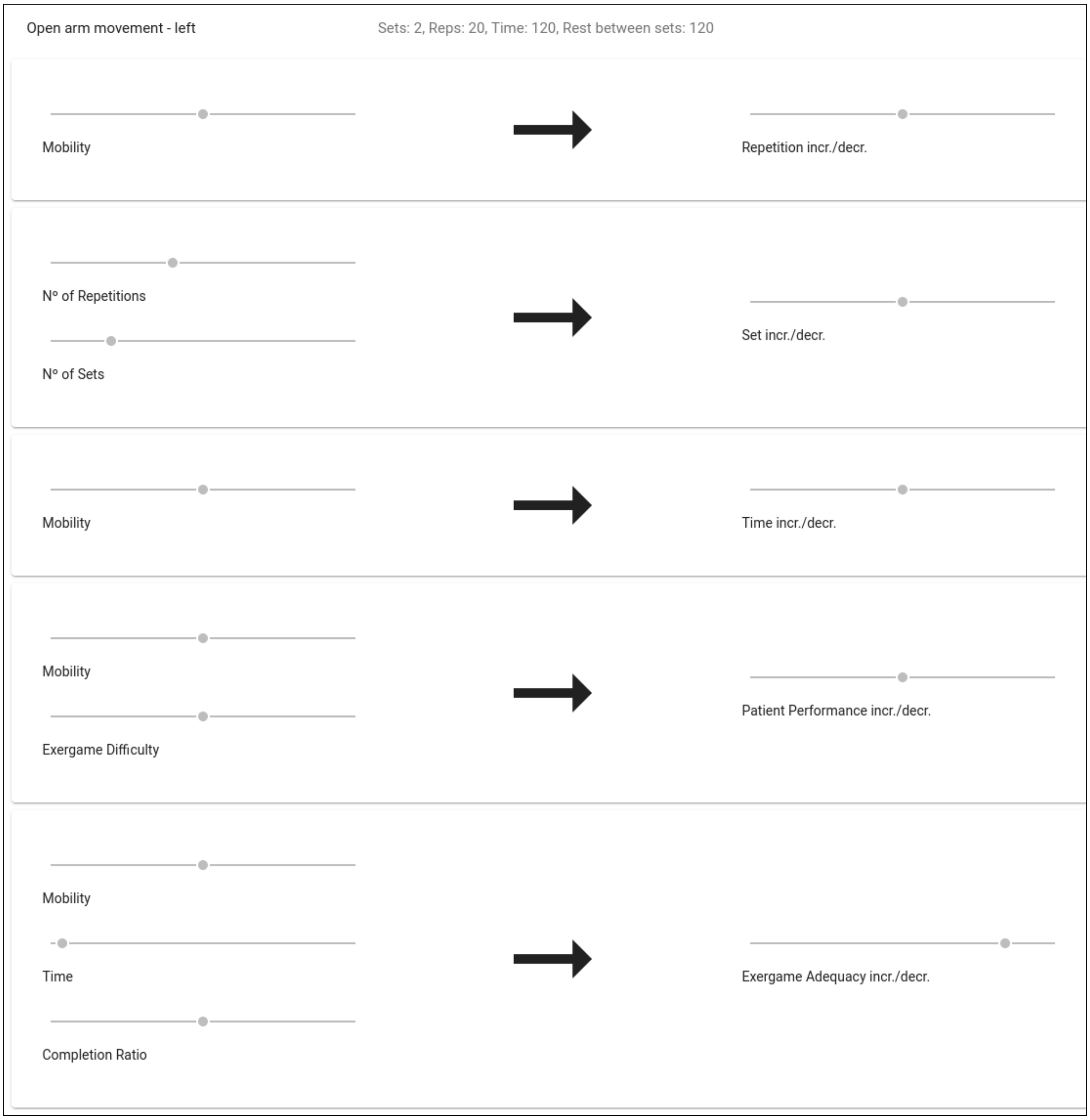


Figure 6.7: Adjustment Explanation.

both repositories are hosted in the Bitbucket[2] repository hosting service. However, the statistics obtained from said extension were not exclusive to this project because it has been executed alongside other development processes of the system. A more limited number of statistics were extracted using the GitStats[3] tool, which can work with local Git repositories. The generated statistics are a summary table of the changes in the repositories (Table 6.2), and graphs representing the number of commits over time (Figures 6.13a and 6.13b). It is worth noting that the late integration of the model itself in the system partially skews the statistics.

[2]https://bitbucket.org/
[3]http://gitstats.sourceforge.net/

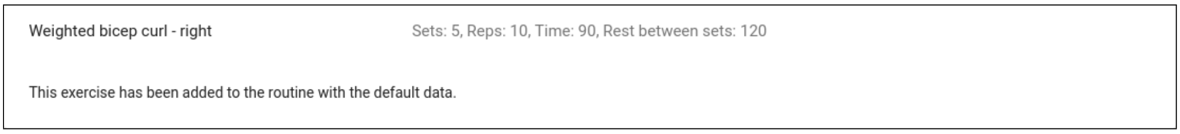


Figure 6.8: Expansion panel when no explanation is available.

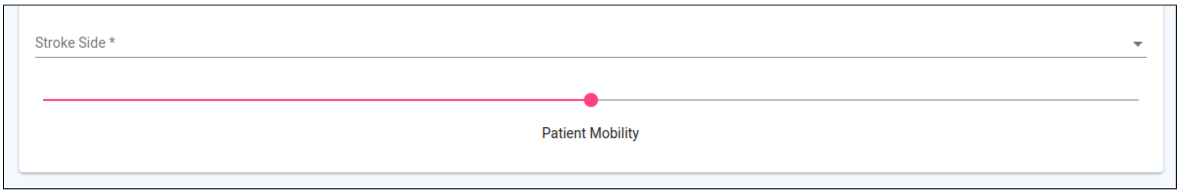


Figure 6.9: New patient fields.

6.5 Project Costs and Resources

In this section, an estimation of the costs of the project is performed. Firstly, there are no costs besides the salary of the developer. Since this project does not require the use of Kinect™ and the software used is freely available, there are no costs derived from the acquisition of hardware or software. The personal laptop used during the project was not acquired for the purpose of this project, therefore its cost is not considered.

For estimating the costs of the developer, it is necessary to analyze the time dedication of the development. The project was executed from January 2022 to June 2022. Although the workload was not consistent during this time frame, it can be estimated that the total workload was of 250 hours, excluding the time dedicated to other development tasks in the system, and to the creation of this document. Then, an estimation of the salary was done using the employment platform InfoJobs[4]. The hourly costs for the company were approximated to be of 35€. Therefore, the total cost of the project is 8750€.

[4]`https://www.infojobs.net`

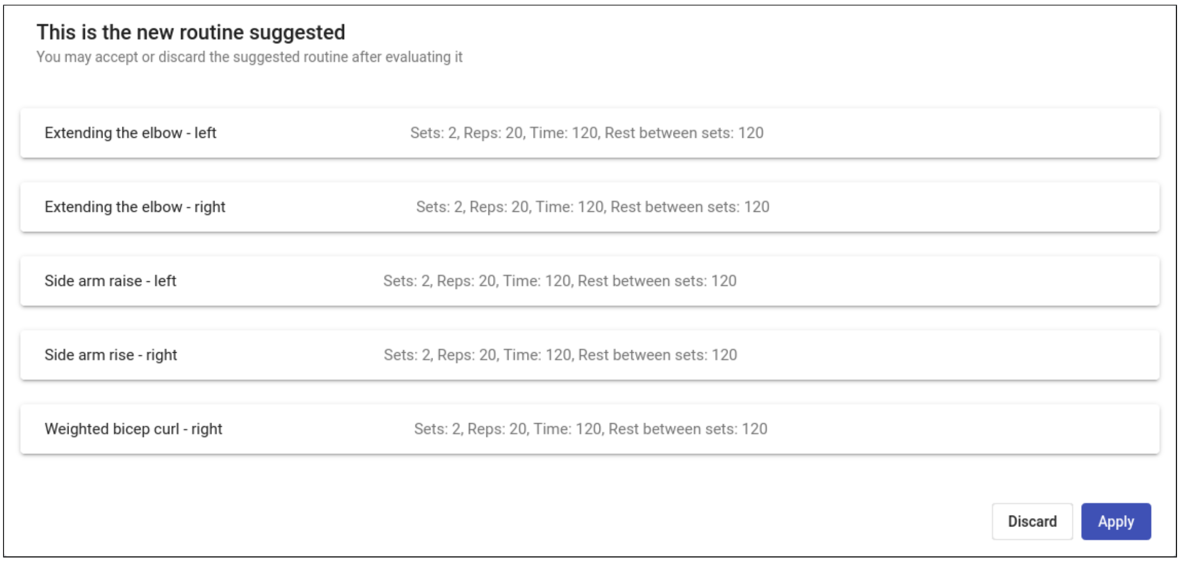


Figure 6.10: Sample Execution 1.

Repository	Commits (%)	+ lines	- lines	First commit	Last commit
Therapist	24 (43.64%)	3781	884	2022-01-11	2022-06-24
Shared	13 (76.47%)	194	48	2022-01-09	2022-06-24

Table 6.2: Summary statistics of the changes in the repository.

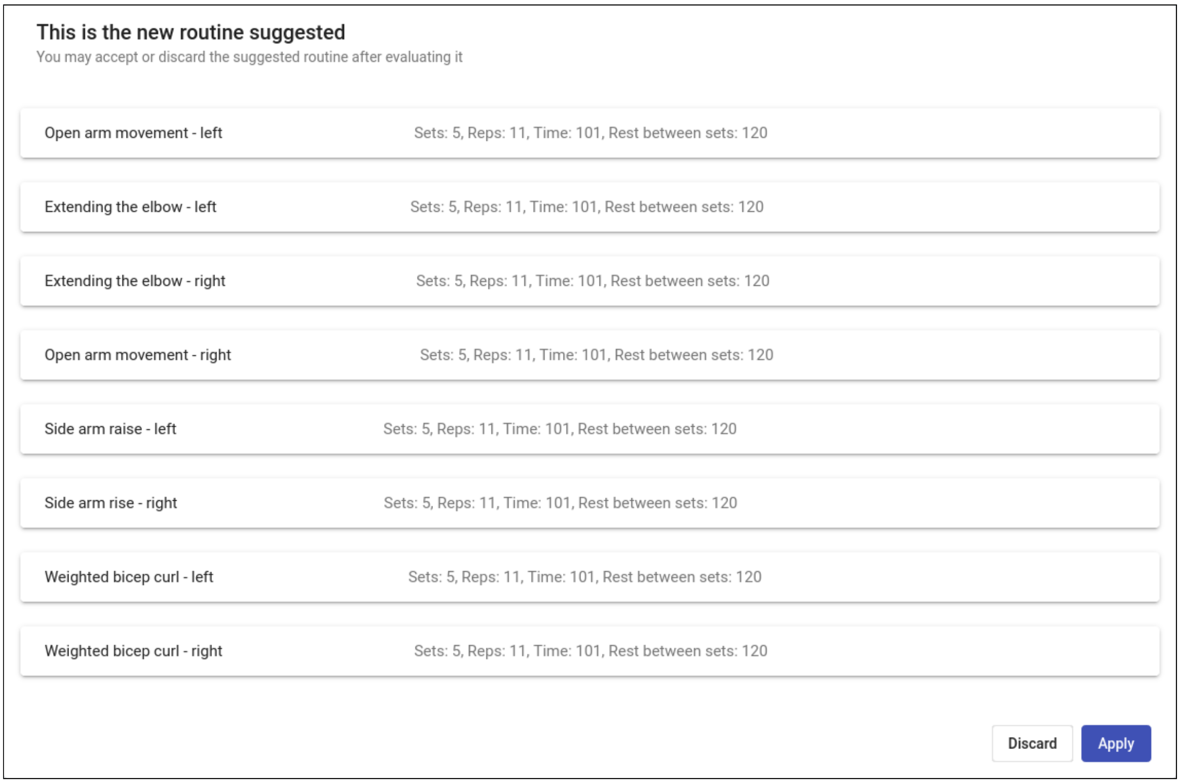
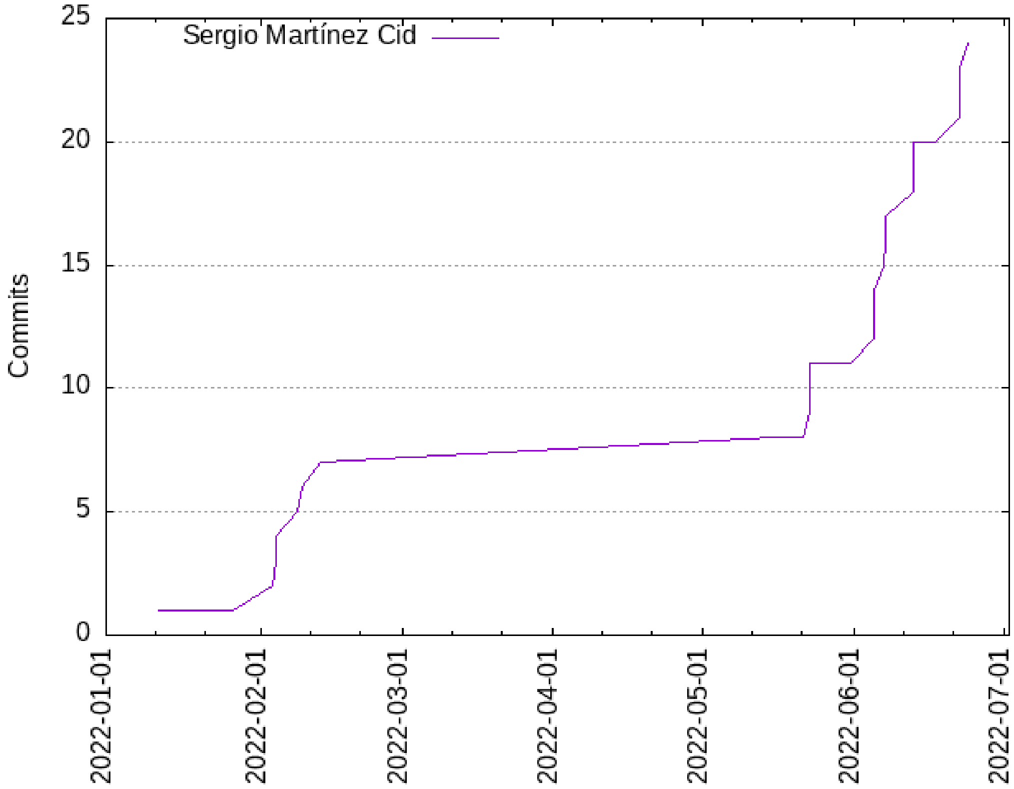
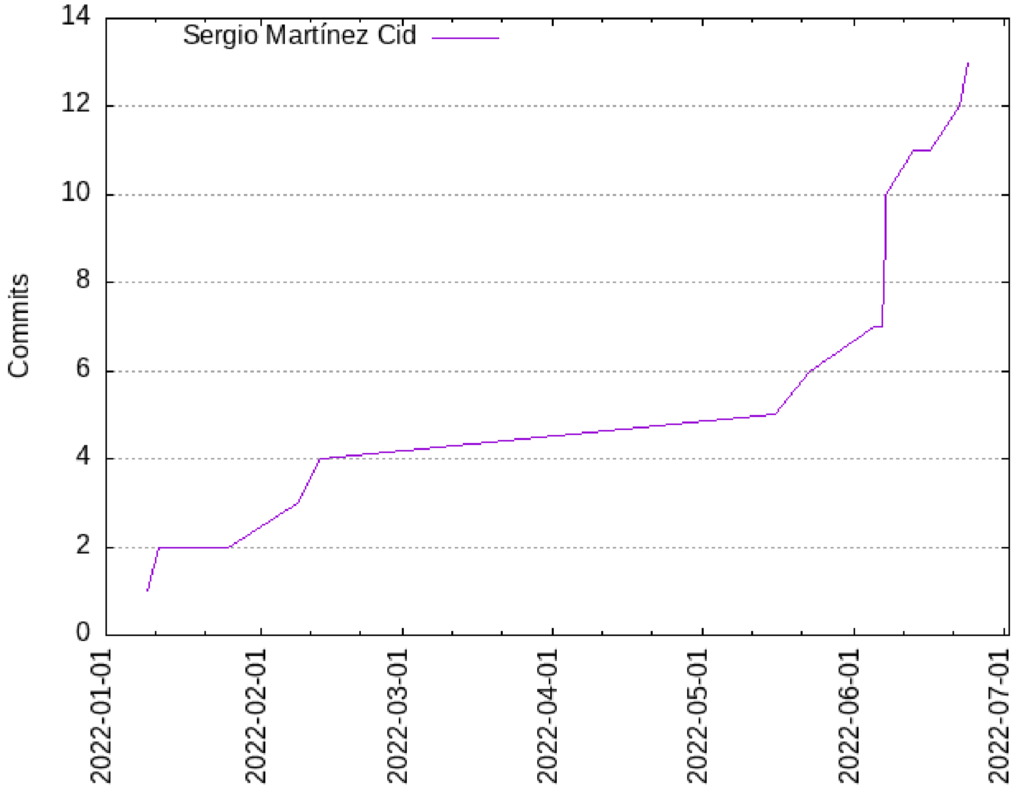


Figure 6.11: Sample Execution 2.

Figure 6.12: Sample Execution 3.



(a) Commits in the therapist repository.



(b) Commits in the shared repository.

Figure 6.13: Graphs representing repository statistics.

Chapter 7

Conclusions

In this chapter the reached objectives of the project will be discussed. Then, the addressed competences will be reviewed. Finally, possible future work and improvements on the DSS will be presented.

7.1 Reached objectives

Firstly, the general objective can be considered complete. The DSS described in this document has been successfully implemented, it has been integrated into the system, and it generates explanations for its decisions. Below, the degree to which the partial objectives have been achieved will be discussed.

- **Study the technologies, techniques and tools that optimize the development and deployment of the proposed DSS in a production system.** This objective can be considered completed, as a thorough study was performed for selecting the optimal alternatives. Firstly, several AI techniques were evaluated, considering their scalability and their aptitude for generating explanations. Ultimately, FISs were chosen as the AI technique for adjusting the routines and generating the subsequent explanation. Once the technique was chosen, concrete libraries supporting fuzzy logic were evaluated.
- **Study the required knowledge for the automatic definition of physical rehabilitation routines.** This objective can be considered complete because research was conducted on the main factors that are considered by therapists when defining a rehabilitation routine. Although sources detailing guidelines for the definition of physical rehabilitation routines were scarce, the indications discovered during the research were used for the definition of the fuzzy rules and for the definition of the fuzzy variables. Consequently, the chosen fuzzy variables had an effect on the data that was captured by the system during the executions of the routines, and stored in the database. This is because the execution data of the routines is used as input for the FIS.
- **Design, development and integration of the DSS.** The design of the DSS focused on the strategy followed to select the exercises that will constitute the routine and to adjust the difficulty of the exercises. The design is highly dependent on the selected AI technique. Since FISs were selected, the design consisted in specifying the rules for

achieving the aforementioned goals. The rules were implemented using the selected technologies to create the inference model. Finally, the inference model was integrated in the server side of the system and a user interface was added to make the functionality accessible to the user.

- **Integration of a component that explains to the therapist the reasoning of the DSS.** The model was augmented with the possibility of generating explanations of its output. Since the model is a FIS, explanations are provided in a *if-then* form which is highly intelligible. Initially, explanations were provided in a textual format, but they were replaced with a visual representation. Providing explanations visually is preferable since they are easier to understand.

- **Testing of the DSS.** The testing of the DSS was accomplished. The first step of the testing process was ensuring that the model always provides an output. This is not guaranteed in the case of FISs, since there may not be enough rules and for some combinations, no rule is fired. Since this step could be automatized, it was incorporated in the system as a test for validating future versions of the model. Then, it was manually checked that the output of the DSS was within the expectations. This was an iterative process that improved the model over time.

7.2 Addressed competences

In this section the addressed competences will be discussed.

- **[CM4] Ability to design and create systems that utilize techniques from intelligent systems.** The DSS developed in the project is fundamentally an intelligent system. Fuzzy logic was applied as an AI technique for the development of the DSS, meaning knowledge regarding techniques from intelligent systems was necessary for the design and implementation of the DSS. Additionally, Explainable Artificial Intelligence (XAI) was also a relevant aspect of the DSS, which is a subfield of AI and is therefore related to the creation of intelligent systems.

- **[CM5] Ability to acquire and formalize human knowledge in a computable format.** This competence is address by the design and implementation of the FIS. Firstly, the definition of the fuzzy rules involved acquiring expert knowledge about physical rehabilitation for stroke patients. Secondly, the acquired human knowledge was represented and formalized using fuzzy rules. Through the use of fuzzy logic, the knowledge can be used by a computer for solving problems, in this particular case for recommending personalized routines.

- **[CM7] Ability to know and develop machine learning techniques.** While machine learning techniques were not ultimately implemented in the DSS, many of the alternative AI techniques considered for the model were machine learning techniques. In particular, many supervised learning techniques were contemplated, and knowledge about these techniques, their advantages and disadvantages was essential in the design phase.

7.3 Personal conclusion

This project marks the end of my bachelor's degree, and it has been possible thanks to the knowledge and the skills that I have acquired during the last 4 years. This project has been an opportunity to deepen my knowledge about AI techniques, and to put into practice one of the techniques I was already familiar with, which is fuzzy logic. In addition, I have learned about the field of Explainable Artificial Intelligence (XAI), which I consider promising for the integration of AI in new areas.

Although I have had previous experience in the development of software systems professionally, including the system where my project has been executed, this development has been a nurturing experience. This is because I am still learning from the professional development world, and this project in particular has been another source of knowledge and experience.

Additionally, this project has allowed me to apply AI techniques in a real system. The intelligent techniques studied throughout the degree have been presented in an isolated fashion. This approach is necessary for an initial contact, but it hides the challenges that emerge when it is necessary to integrate AI techniques in a working system.

Finally, I highly value that this project has been executed in a commercial system. I am aware that there is still room for improvement, but it would be very fulfilling if the result of this project were able to relieve therapists from part of their work, and therapist were able to spend more quality time with their patients.

7.4 Future work

In this section, possible improvements to the DSS are presented.

- **Improve the generation of explanations.** While the current approach for generating explanations is effective, it does not handle equally all cases. When there is only one rule that contributes significantly to the output of the system, it is selected as the explanation. However, when there are many rules that contribute to the output in a significant way, only one of those rules is selected. This could be improved by generating an explanation as the combination of similar rules, or by selecting multiple rules as the explanation.
- **Handle therapist indications.** The current version of the DSS does not accept indications from the therapist besides the level of mobility of the patient. If therapists had a greater level of influence, they could handle edge cases better. For example, if the

therapist considers that the patient has to perform a particular exercise, then the DSS would include that exercise in the routine despite the level of adequacy of that exercise. Furthermore, if a patient had obesity, arthritis or any other condition where excessive exercise could harm the joints, the therapist should be allowed to limit the workload.

- **Detect trends in the progress of patients.** The current version of the DSS is passive; a suggestion is only provided when the therapist requests it. This is because altering the rehabilitation routine of the patient without the supervision of the therapist is not without risk. However, there are other tasks in the supervision of the progress of patients where the system could have a more proactive approach. For example, if the patient failed execute the routine or did not execute enough repetitions, the system could notify the therapist of this situation.
- **Provide insight about the patient progress.** There are many ways of supporting the decision-making progress, as seen in Chapter 3. It could be useful to provide support to the decisions of the therapists in ways other than adjusting the difficulty. In particular, the current version of the system does not provide any way to visualize the execution data of the routines. In addition to displaying this data to the therapist, it would also be possible to extract useful information from said data.
- **Validate the DSS with an expert.** Although the output of the DSS has been extensively reviewed during the execution of the project, it would be beneficial to validate the DSS with an expert on physical rehabilitation. There are subtle details that can only be detected by experts with experience in the field. Additionally, an expert could also help in determining other ways of improving the DSS.
- **Deploy the application in production.** The system has not been deployed yet to a production environment with the DSS. Even though Docker[1] is used to control the environment, there is no guarantee that no challenges will emerge during the deployment process.

[1] `https://www.docker.com/`

Appendices

Appendix A

Appendix A

A.1 Rules from the FIS, implemented in R

```r
# Repetitions increment
rep_incr_rules <- list(
  fuzzy_rule(time %is% VL && completion %is% VH,
             rep_incr %is% VH),
  fuzzy_rule(time %is% VL && completion %is% M,
             rep_incr %is% H),
  fuzzy_rule(time %is% VL && completion %is% VL,
             rep_incr %is% M),

  fuzzy_rule(time %is% M && completion %is% VH,
             rep_incr %is% VH),
  fuzzy_rule(time %is% M && completion %is% M,
             rep_incr %is% M),
  fuzzy_rule(time %is% M && completion %is% VL,
             rep_incr %is% VL),

  fuzzy_rule(time %is% VH && completion %is% VH,
             rep_incr %is% M),
  fuzzy_rule(time %is% VH && completion %is% M,
             rep_incr %is% L),
  fuzzy_rule(time %is% VH && completion %is% VL,
             rep_incr %is% VL),


  fuzzy_rule(mobility %is% M, rep_incr %is% M),

  fuzzy_rule(performance %is% M && mobility %is% VL && exergame_diff %is% L,
             rep_incr %is% M),
  fuzzy_rule(performance %is% M && mobility %is% VL && exergame_diff %is% M,
```

```
            rep_incr %is% M),
    fuzzy_rule(performance %is% M && mobility %is% VL && exergame_diff %is% H,
            rep_incr %is% L),

    fuzzy_rule(performance %is% M && mobility %is% VH && exergame_diff %is% L,
            rep_incr %is% H),
    fuzzy_rule(performance %is% M && mobility %is% VH && exergame_diff %is% M,
            rep_incr %is% M),
    fuzzy_rule(performance %is% M && mobility %is% VH && exergame_diff %is% H,
            rep_incr %is% M),

    fuzzy_rule(performance %is% VL && mobility %is% VL && exergame_diff %is% L,
            rep_incr %is% M),
    fuzzy_rule(performance %is% VL && mobility %is% VL && exergame_diff %is% M,
            rep_incr %is% L),
    fuzzy_rule(performance %is% VL && mobility %is% VL && exergame_diff %is% H,
            rep_incr %is% L),

    fuzzy_rule(performance %is% VL && mobility %is% VH && exergame_diff %is% L,
            rep_incr %is% H),
    fuzzy_rule(performance %is% VL && mobility %is% VH && exergame_diff %is% M,
            rep_incr %is% M),
    fuzzy_rule(performance %is% VL && mobility %is% VH && exergame_diff %is% H,
            rep_incr %is% M),

    fuzzy_rule(performance %is% VH && mobility %is% VL && exergame_diff %is% L,
            rep_incr %is% M),
    fuzzy_rule(performance %is% VH && mobility %is% VL && exergame_diff %is% M,
            rep_incr %is% M),
```

```
  fuzzy_rule(performance %is% VH && mobility %is% VL && exergame_diff %is% H,
             rep_incr %is% L),

  fuzzy_rule(performance %is% VH && mobility %is% VH && exergame_diff %is% L,
             rep_incr %is% H),
  fuzzy_rule(performance %is% VH && mobility %is% VH && exergame_diff %is% M,
             rep_incr %is% H),
  fuzzy_rule(performance %is% VH && mobility %is% VH && exergame_diff %is% H,
             rep_incr %is% M)
)

# Set increment
set_incr_rules <- list(
  fuzzy_rule(reps %is% VL && sets %is% VL,
             set_incr %is% M),
  fuzzy_rule(reps %is% M && sets %is% M,
             set_incr %is% M),
  fuzzy_rule(reps %is% H && sets %is% H,
             set_incr %is% M),

  fuzzy_rule(reps %is% L && sets %is% M,
             set_incr %is% L),
  fuzzy_rule(reps %is% VH && sets %is% M,
             set_incr %is% VH),
  fuzzy_rule(reps %is% H && sets %is% VL,
             set_incr %is% VH),
  fuzzy_rule(reps %is% L && sets %is% VH,
             set_incr %is% VL)
)

# Time increment
time_incr_rules <- list(
  fuzzy_rule(time %is% VL && completion %is% VH,
             time_incr %is% L),
  fuzzy_rule(time %is% VL && completion %is% M,
             time_incr %is% L),
  fuzzy_rule(time %is% VL && completion %is% VL,
```

```
                 time_incr %is% M),

    fuzzy_rule(time %is% M && completion %is% VH,
                 time_incr %is% L),
    fuzzy_rule(time %is% M && completion %is% M,
                 time_incr %is% M),
    fuzzy_rule(time %is% M && completion %is% VL,
                 time_incr %is% H),

    fuzzy_rule(time %is% VH && completion %is% VH,
                 time_incr %is% M),
    fuzzy_rule(time %is% VH && completion %is% M,
                 time_incr %is% H),
    fuzzy_rule(time %is% VH && completion %is% VL,
                 time_incr %is% VH),

    fuzzy_rule(mobility %is% M, time_incr %is% M),

    fuzzy_rule(performance %is% M && mobility %is% VL && exergame_diff %is% L,
                 time_incr %is% M),
    fuzzy_rule(performance %is% M && mobility %is% VL && exergame_diff %is% M,
                 time_incr %is% H),
    fuzzy_rule(performance %is% M && mobility %is% VL && exergame_diff %is% H,
                 time_incr %is% H),

    fuzzy_rule(performance %is% M && mobility %is% VH && exergame_diff %is% L,
                 time_incr %is% L),
    fuzzy_rule(performance %is% M && mobility %is% VH && exergame_diff %is% M,
                 time_incr %is% M),
    fuzzy_rule(performance %is% M && mobility %is% VH && exergame_diff %is% H,
                 time_incr %is% M),

    fuzzy_rule(performance %is% VL && mobility %is% VL && exergame_diff %is% L,
                 time_incr %is% M),
```

```
  fuzzy_rule(performance %is% VL && mobility %is% VL && exergame_diff %is% M,
             time_incr %is% H),
  fuzzy_rule(performance %is% VL && mobility %is% VL && exergame_diff %is% H,
             time_incr %is% H),

  fuzzy_rule(performance %is% VL && mobility %is% VH && exergame_diff %is% L,
             time_incr %is% L),
  fuzzy_rule(performance %is% VL && mobility %is% VH && exergame_diff %is% M,
             time_incr %is% M),
  fuzzy_rule(performance %is% VL && mobility %is% VH && exergame_diff %is% H,
             time_incr %is% M),

  fuzzy_rule(performance %is% VH && mobility %is% VL && exergame_diff %is% L,
             time_incr %is% M),
  fuzzy_rule(performance %is% VH && mobility %is% VL && exergame_diff %is% M,
             time_incr %is% M),
  fuzzy_rule(performance %is% VH && mobility %is% VL && exergame_diff %is% H,
             time_incr %is% H),

  fuzzy_rule(performance %is% VH && mobility %is% VH && exergame_diff %is% L,
             time_incr %is% L),
  fuzzy_rule(performance %is% VH && mobility %is% VH && exergame_diff %is% M,
             time_incr %is% M),
  fuzzy_rule(performance %is% VH && mobility %is% VH && exergame_diff %is% H,
             time_incr %is% M)
)

# Performance increment
performance_incr_rules <- list(
  fuzzy_rule(time %is% VL && completion %is% VH,
```

```
              performance_incr %is% VH),
    fuzzy_rule(time %is% VL && completion %is% M,
              performance_incr %is% H),
    fuzzy_rule(time %is% VL && completion %is% VL,
              performance_incr %is% M),

    fuzzy_rule(time %is% M && completion %is% VH,
              performance_incr %is% H),
    fuzzy_rule(time %is% M && completion %is% M,
              performance_incr %is% M),
    fuzzy_rule(time %is% M && completion %is% VL,
              performance_incr %is% L),

    fuzzy_rule(time %is% VH && completion %is% VH,
              performance_incr %is% M),
    fuzzy_rule(time %is% VH && completion %is% M,
              performance_incr %is% L),
    fuzzy_rule(time %is% VH && completion %is% VL,
              performance_incr %is% VL),

    fuzzy_rule(mobility %is% VL && exergame_diff %is% VL,
              performance_incr %is% M),
    fuzzy_rule(mobility %is% VL && exergame_diff %is% M,
              performance_incr %is% M),
    fuzzy_rule(mobility %is% VL && exergame_diff %is% VH,
              performance_incr %is% H),

    fuzzy_rule(mobility %is% M && exergame_diff %is% VL,
              performance_incr %is% M),
    fuzzy_rule(mobility %is% M && exergame_diff %is% M,
              performance_incr %is% M),
    fuzzy_rule(mobility %is% M && exergame_diff %is% VH,
              performance_incr %is% H),

    fuzzy_rule(mobility %is% VH && exergame_diff %is% VL,
              performance_incr %is% M),
    fuzzy_rule(mobility %is% VH && exergame_diff %is% M,
              performance_incr %is% M),
    fuzzy_rule(mobility %is% VH && exergame_diff %is% VH,
              performance_incr %is% H)
  )
```

```
# Adequacy increment
adequacy_incr_rules <- list(
  fuzzy_rule(mobility %is% L && time %is% VL && completion %is% VH,
             adequacy_incr %is% L),
  fuzzy_rule(mobility %is% L && time %is% M && completion %is% H,
             adequacy_incr %is% VH),
  fuzzy_rule(mobility %is% L && time %is% VH && completion %is% L,
             adequacy_incr %is% L),

  fuzzy_rule(mobility %is% L && time %is% VL && completion %is% L,
             adequacy_incr %is% M),
  fuzzy_rule(mobility %is% L && time %is% VH && completion %is% VH,
             adequacy_incr %is% L),

  fuzzy_rule(mobility %is% M && time %is% VL && completion %is% VH,
             adequacy_incr %is% L),
  fuzzy_rule(mobility %is% M && time %is% M && completion %is% H,
             adequacy_incr %is% VH),
  fuzzy_rule(mobility %is% M && time %is% VH && completion %is% L,
             adequacy_incr %is% L),

  fuzzy_rule(mobility %is% M && time %is% VL && completion %is% L,
             adequacy_incr %is% M),
  fuzzy_rule(mobility %is% M && time %is% VH && completion %is% VH,
             adequacy_incr %is% M),

  fuzzy_rule(mobility %is% H && time %is% VL && completion %is% VH,
             adequacy_incr %is% VL),
```

```
  fuzzy_rule(mobility %is% H && time %is% M && completion %is%
     H,
             adequacy_incr %is% VH),
  fuzzy_rule(mobility %is% H && time %is% VH && completion %is%
      L,
             adequacy_incr %is% VL),

  fuzzy_rule(mobility %is% H && time %is% VL && completion %is%
      L,
             adequacy_incr %is% L),
  fuzzy_rule(mobility %is% H && time %is% VH && completion %is%
      VH,
             adequacy_incr %is% VL),

  fuzzy_rule(mobility %is% VL && time %is% M && completion %is%
      VL,
             adequacy_incr %is% L),
  fuzzy_rule(mobility %is% M && time %is% M && completion %is%
     VL,
             adequacy_incr %is% L),
  fuzzy_rule(mobility %is% VH && time %is% M && completion %is%
      VL,
             adequacy_incr %is% VL),
  fuzzy_rule(mobility %is% M && time %is% VH && completion %is%
      M,
             adequacy_incr %is% L)
)

# Exergame number
exergame_number_rules <- list(
  fuzzy_rule(mobility %is% M && performance %is% VH,
             exergame_number %is% VH),
  fuzzy_rule(mobility %is% M && performance %is% M,
             exergame_number %is% M),
  fuzzy_rule(mobility %is% M && performance %is% VL,
             exergame_number %is% VL),

  fuzzy_rule(mobility %is% H && performance %is% VH,
             exergame_number %is% H),
  fuzzy_rule(mobility %is% H && performance %is% M,
             exergame_number %is% M),
```

```
  fuzzy_rule(mobility %is% H && performance %is% VL,
             exergame_number %is% VL),

  fuzzy_rule(mobility %is% L && performance %is% VH,
             exergame_number %is% VH),
  fuzzy_rule(mobility %is% L && performance %is% M,
             exergame_number %is% M),
  fuzzy_rule(mobility %is% L && performance %is% VL,
             exergame_number %is% L)
)

# Exergame difficulty
exergame_diff_rules <- list(
  fuzzy_rule(waypoint_number %is% VL && waypoint_distance %is% VL,
             exergame_diff %is% L),
  fuzzy_rule(waypoint_number %is% VL && waypoint_distance %is% M,
             exergame_diff %is% L),
  fuzzy_rule(waypoint_number %is% VL && waypoint_distance %is% VH,
             exergame_diff %is% M),

  fuzzy_rule(waypoint_number %is% M && waypoint_distance %is% VL,
             exergame_diff %is% M),
  fuzzy_rule(waypoint_number %is% M && waypoint_distance %is% M,
             exergame_diff %is% M),
  fuzzy_rule(waypoint_number %is% M && waypoint_distance %is% VH,
             exergame_diff %is% H),

  fuzzy_rule(waypoint_number %is% VH && waypoint_distance %is% VL,
             exergame_diff %is% H),
  fuzzy_rule(waypoint_number %is% VH && waypoint_distance %is% M,
             exergame_diff %is% H),
  fuzzy_rule(waypoint_number %is% VH && waypoint_distance %is% VH,
```

```
                     exergame_diff %is% VH)
)
```

Code listing A.1: Rules from the FIS, implemented in R

References

[AAM+12] Juliet Addo, Luis Ayerbe, Keerthi M. Mohan, Siobhan Crichton, Anita Sheldenkar, Ruoling Chen, Charles D.A. Wolfe, y Christopher McKevitt. Socioeconomic Status and Stroke. *Stroke*, 43(4):1186–1191, Abril 2012. url: `https://doi.org/10.1161/strokeaha.111.639732`.

[ABB+18] David Anton, Idoia Berges, Jesús Bermúdez, Alfredo Goñi, y Arantza Illarramendi. A Telerehabilitation System for the Selection, Evaluation and Remote Management of Therapies. *Sensors*, 18(5):1459, Mayo 2018. url: `https://doi.org/10.3390/s18051459`.

[ACL20] Tomás Alves, Henrique Carvalho, y Daniel Simões Lopes. Winning compensations: Adaptable gaming approach for upper limb rehabilitation sessions based on compensatory movements. *Journal of Biomedical Informatics*, 108:103501, Agosto 2020. url: `https://doi.org/10.1016/j.jbi.2020.103501`.

[Alt79] Steven Alter. *Decision support systems: Current practice and continuing challenges*. Addison-Wesley, 1979.

[Awa05] MA Awad. A comparison between agile and traditional software development methodologies. *University of Western Australia*, 30:1–69, 2005.

[BHW+17] Lara A Boyd, Kathryn S Hayward, Nick S Ward, Cathy M Stinear, Charlotte Rosso, Rebecca J Fisher, Alexandre R Carter, Alex P Leff, David A Copland, Leeanne M Carey, Leonardo G Cohen, D Michele Basso, Jane M Maguire, y Steven C Cramer. Biomarkers of stroke recovery: Consensus-based core recommendations from the Stroke Recovery and Rehabilitation Roundtable. *International Journal of Stroke*, 12(5):480–493, Julio 2017. url: `https://doi.org/10.1177/1747493017714176`.

[BM21] Florian Berger y Wolfgang Müller. Back to Basics: Explainable AI for Adaptive Serious Games. En *Serious Games*, páginas 67–81. Springer

International Publishing, 2021. url: https://doi.org/10.1007/978-3-030-88272-3_6.

[CC00] C. Edward Coffey y Jeffrey L. Cummings. *27. Stroke*, página 626–630. American Psychiatric Press, 2000.

[CCP+19] Junqiao Chen, David Chun, Milesh Patel, Epson Chiang, y Jesse James. The validity of synthetic clinical data: a validation study of a leading synthetic data generator (Synthea) using clinical quality measures. *BMC Medical Informatics and Decision Making*, 19(1), Marzo 2019. url: https://doi.org/10.1186/s12911-019-0793-0.

[CCWB20] Roberto Confalonieri, Ludovik Coba, Benedikt Wagner, y Tarek R. Besold. A historical perspective of explainable Artificial Intelligence. *WIREs Data Mining and Knowledge Discovery*, 11(1), Octubre 2020. url: https://doi.org/10.1002/widm.1391.

[CJC+21] Kendy Gui Fang Chan, Ying Jiang, Wen Ting Choo, Hadassah Joann Ramachandran, Yanjuan Lin, y Wenru Wang. Effects of exergaming on functional outcomes in people with chronic stroke: A systematic review and meta-analysis. *Journal of Advanced Nursing*, 78(4):929–946, Diciembre 2021. url: https://doi.org/10.1111/jan.15125.

[CPANBM+19] Campos-Puente, Avargues-Navarro, Borda-Mas, Sánchez-Martín, Aguilar-Parra, y Trigueros. Emotional Exhaustion in Housewives and Alzheimer Patients' Caregivers: Its Effects on Chronic Diseases, Somatic Symptoms and Social Dysfunction. *International Journal of Environmental Research and Public Health*, 16(18):3250, Septiembre 2019. url: https://doi.org/10.3390/ijerph16183250.

[CPW+18] Ruth Cornick, Sandy Picken, Camilla Wattrus, Ajibola Awotiwon, Emma Carkeek, Juliet Hannington, Pearl Spiller, Eric Bateman, Tanya Doherty, Merrick Zwarenstein, y Lara Fairall. The Practical Approach to Care Kit (PACK) guide: developing a clinical decision support tool to simplify, standardise and strengthen primary healthcare delivery. *BMJ Global Health*, 3(Suppl 5):e000962, Octubre 2018. url: https://doi.org/10.1136/bmjgh-2018-000962.

[DDS+16] He Deng, Wankai Deng, Xianping Sun, Chaohui Ye, y Xin Zhou. Adaptive Intuitionistic Fuzzy Enhancement of Brain Tumor MR Images. *Scientific Reports*, 6(1), Octubre 2016. url: https://doi.org/10.1038/srep35760.

[Dee19] Ashley Deeks. THE JUDICIAL DEMAND FOR EXPLAINABLE ARTIFICIAL INTELLIGENCE. *JSTOR*, 119(7), Nov 2019.

[Dob04] Bruce H Dobkin. Strategies for stroke rehabilitation. *The Lancet Neurology*, 3(9):528–536, Septiembre 2004. url: `https://doi.org/10.1016/s1474-4422(04)00851-8`.

[EK06] S Eom y E Kim. A survey of decision support system applications (1995–2001). *Journal of the Operational Research Society*, 57(11):1264–1278, Noviembre 2006. url: `https://doi.org/10.1057/palgrave.jors.2602140`.

[FCSO+15] Victor Fernandez-Cervantes, Eleni Stroulia, Luis E. Oliva, Francisco Gonzalez, y Claudio Castillo. Serious games: Rehabilitation fuzzy grammar for exercise and therapy compliance. En *2015 IEEE Games Entertainment Media Conference (GEM)*. IEEE, Octubre 2015. url: `https://doi.org/10.1109/gem.2015.7377229`.

[FRN+16] Valery L Feigin, Gregory A Roth, Mohsen Naghavi, Priya Parmar, Rita Krishnamurthi, Sumeet Chugh, George A Mensah, Bo Norrving, Ivy Shiue, Marie Ng, Kara Estep, Kelly Cercy, Christopher J L Murray, y Mohammad H Forouzanfar. Global burden of stroke and risk factors in 188 countries, during 1990–2013: a systematic analysis for the Global Burden of Disease Study 2013. *The Lancet Neurology*, 15(9):913–924, Agosto 2016. url: `https://doi.org/10.1016/s1474-4422(16)30073-4`.

[GC11] Serge Guillaume y Brigitte Charnomordic. Learning interpretable fuzzy inference systems with FisPro. *Information Sciences*, 181(20):4409–4427, Octubre 2011. url: `https://doi.org/10.1016/j.ins.2011.03.025`.

[Geo91] Joey F. George. The Conceptualization and Development of Organizational Decision Support Systems. *Journal of Management Information Systems*, 8(3):109–125, Diciembre 1991. url: `https://doi.org/10.1080/07421222.1991.11517932`.

[GGTDMCTC19] Carina S. González-González, Pedro A. Toledo-Delgado, Vanesa Muñoz-Cruz, y Pablo V. Torres-Carrion. Serious games for rehabilitation: Gestural interaction in personalized gamified exercises through a recommender system. *Journal of Biomedical Informatics*, 97:103266, Septiembre 2019. url: `https://doi.org/10.1016/j.jbi.2019.103266`.

[GHJ98] Erich Gamma, Richard Helm, y Ralph Johnson. *Design patterns elements of Reusable Object Oriented Software*. Addison Wesley, 1998.

[GPCSA+21] Cristian Gmez-Portes, José Jesús Castro-Schez, Javier Albusac, Dorothy N. Monekosso, y David Vallejo. A Fuzzy Recommendation System for the Automatic Personalization of Physical Rehabilitation Exercises in Stroke Patients. *Mathematics*, 9(12):1427, Junio 2021. url: `https://doi.org/10.3390/math9121427`.

[GRBS20] Bruna Maria Vittoria Guerra, Stefano Ramat, Giorgio Beltrami, y Micaela Schmid. Automatic Pose Recognition for Monitoring Dangerous Situations in Ambient-Assisted Living. *Frontiers in Bioengineering and Biotechnology*, 8, Mayo 2020. url: `https://doi.org/10.3389/fbioe.2020.00415`.

[GSC+19] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, y Guang-Zhong Yang. XAI—Explainable artificial intelligence. *Science Robotics*, 4(37), Diciembre 2019. url: `https://doi.org/10.1126/scirobotics.aay7120`.

[GW13] Joao Gomes y Ari Marc Wachsman. Types of Strokes. En *Handbook of Clinical Nutrition and Stroke*, páginas 15–31. Humana Press, 2013. url: `https://doi.org/10.1007/978-1-62703-380-0_2`.

[Ham19] Juho Hamari. Gamification, Noviembre 2019. url: `https://doi.org/10.1002/9781405165518.wbeos1321`.

[Hay08] Simon Haykin. *1. Rosenblatt's Perceptron*, capítulo 1. Rosenblatt's Perceptron, páginas 47–67. Pearson, 2008.

[HBATA20] Adedoyin Ahmed Hussain, Ouns Bouachir, Fadi Al-Turjman, y Moayad Aloqaily. Notice of Retraction: AI Techniques for COVID-19. *IEEE Access*, 8:128776–128795, 2020. url: `https://doi.org/10.1109/access.2020.3007939`.

[HD81] K. York Haaland y Harold D. Delaney. Motor deficits after left or right hemisphere damage due to stroke or tumor. *Neuropsychologia*, 19(1):17–27, Enero 1981. url: `https://doi.org/10.1016/0028-3932(81)90040-3`.

[HGC+15] Nadia Hocine, Abdelkader Gouaïch, Stefano A. Cerri, Denis Mottet, Jérome Froger, y Isabelle Laffont. Adaptation in serious games for upper-limb rehabilitation: an approach to improve training outcomes. *User*

Modeling and User-Adapted Interaction, 25(1):65–98, Enero 2015. url: `https://doi.org/10.1007/s11257-015-9154-6`.

[Hig00] J. Highsmith. *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Pearson Education, 2000.

[HW01] Clyde W. Holsapple y Andrew B. Whinston. *Decision Support Systems: A knowledge-based approach*. Thomson Learning Custom Pub., 2001.

[KEA+14] Ali Karime, Mohamad Eid, Jihad Mohamad Alja'am, Abdulmotaleb El Saddik, y Wail Gueaieb. A Fuzzy-Based Adaptive Rehabilitation Framework for Home-Based Wrist Training. *IEEE Transactions on Instrumentation and Measurement*, 63(1):135–144, Enero 2014. url: `https://doi.org/10.1109/tim.2013.2277536`.

[Kei18] Georg Keilbar. Modelling Systemic Risk using Neural Network Quantile Regression. *University of Vienna*, 2018. url: `https://edoc.hu-berlin.de/handle/18452/20079`.

[KHH+11] Patricia Kan, Rajibul Huq, Jesse Hoey, Robby Goetschalckx, y Alex Mihailidis. The development of an adaptive upper-limb stroke rehabilitation robotic system. *Journal of NeuroEngineering and Rehabilitation*, 8(1), Junio 2011. url: `https://doi.org/10.1186/1743-0003-8-33`.

[KJS14] Leonie Klompstra, Tiny Jaarsma, y Anna Strömberg. Exergaming to increase the exercise capacity and daily physical activity in heart failure patients: a pilot study. *BMC Geriatrics*, 14(1), Noviembre 2014. url: `https://doi.org/10.1186/1471-2318-14-119`.

[KLC98] Leslie Pack Kaelbling, Michael L. Littman, y Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, Mayo 1998. url: `https://doi.org/10.1016/s0004-3702(98)00023-x`.

[Kot11] S. B. Kotsiantis. Use of machine learning techniques for educational proposes: a decision support system for forecasting students' grades. *Artificial Intelligence Review*, 37(4):331–344, Mayo 2011. url: `https://doi.org/10.1007/s10462-011-9234-x`.

[KS08] S. Anil Kumar y N. Suresh. *Production and Operations Management (with Skill Development, Caselets and cases)*. New Age International (P) Ltd., Publishers, 2008.

[LE93] Dorothy E. Leidner y Joyce J. Elam. Executive Information Systems: Their Impact on Executive Decision Making. *Journal of Management Information Systems*, 10(3):139–155, Diciembre 1993. url: https://doi.org/10.1080/07421222.1993.11518014.

[LSS+20] Min Hun Lee, Daniel P. Siewiorek, Asim Smailagic, Alexandre Bernardino, y Sergi Bermúdez i Badia. Opportunities of a Machine Learning-based Decision Support System for Stroke Rehabilitation Assessment, 2020. url: https://arxiv.org/abs/2002.12261.

[Mam74] E.H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121(12):1585, 1974. url: https://doi.org/10.1049/piee.1974.0328.

[Mar00] Robert C Martin. Design principles and design patterns. *Object Mentor*, 1(34):597, 2000.

[MSM11] Michelle N. McDonnell, Ashleigh E. Smith, y Shylie F. Mackintosh. Aerobic Exercise to Improve Cognitive Function in Adults With Neurological Disorders: A Systematic Review. *Archives of Physical Medicine and Rehabilitation*, 92(7):1044, Julio 2011. url: https://doi.org/10.1016/j.apmr.2011.01.021.

[MWC+15] Iain J Marshall, Yanzhong Wang, Siobhan Crichton, Christopher McKevitt, Anthony G Rudd, y Charles D A Wolfe. The effects of socioeconomic status on stroke risk and outcomes. *The Lancet Neurology*, 14(12):1206–1218, Diciembre 2015. url: https://doi.org/10.1016/s1474-4422(15)00200-8.

[MZT+10] Jerry Mendel, Lotfi Zadeh, Enric Trillas, Ronald Yager, Jonathan Lawry, Hani Hagras, y Sergio Guadarrama. What Computing with Words Means to Me [Discussion Forum. *IEEE Computational Intelligence Magazine*, 5(1):20–26, Febrero 2010. url: https://doi.org/10.1109/mci.2009.934561.

[OSB20] Reza Haghighi Osgouei, David Soulsby, y Fernando Bello. Rehabilitation Exergames: Use of Motion Sensing and Machine Learning to Quantify Exercise Performance in Healthy Volunteers. *JMIR Rehabilitation and Assistive Technologies*, 7(2):e17289, Agosto 2020. url: https://doi.org/10.2196/17289.

[Pal92] Stephen E Palmer. Common region: A new principle of perceptual grouping. *Cognitive Psychology*, 24(3):436–447, Julio 1992. url: `https://doi.org/10.1016/0010-0285(92)90014-s`.

[PCA07] Björn Palmqvist, Per Carlbring, y Gerhard Andersson. Internet-delivered treatments with or without therapist input: does the therapist factor have implications for efficacy and cost? *Expert Review of Pharmacoeconomics & Outcomes Research*, 7(3):291–297, Junio 2007. url: `https://doi.org/10.1586/14737167.7.3.291`.

[PCC^{+}18] Joana F. Pinto, Henrique R. Carvalho, Goncalo R. R. Chambel, Joao Ramiro, y Afonso Goncalves. Adaptive gameplay and difficulty adjustment in a gamified upper-limb rehabilitation. En *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*. IEEE, Mayo 2018. url: `https://doi.org/10.1109/segah.2018.8401363`.

[PCP^{+}14] Mara Paneroni, Fausto Colombo, Antonella Papalia, Arcangela Colitta, Gabriella Borghi, Manuela Saleri, Antonella Cabiaglia, Elena Azzalini, y Michele Vitacca. Is Telerehabilitation a Safe and Viable Option for Patients with COPD? A Feasibility Study. *COPD: Journal of Chronic Obstructive Pulmonary Disease*, 12(2):217–225, Agosto 2014. url: `https://doi.org/10.3109/15412555.2014.933794`.

[Pel00] Francis Jeffry Pelletier. Petr Hájek. Metamathematics of fuzzy logic. Trends in logic, vol. 4. Kluwer Academic Publishers, Dordrecht, Boston, and London, 1998, viii + 297 pp. *Bulletin of Symbolic Logic*, 6(3):342–346, Septiembre 2000. url: `https://doi.org/10.2307/421060`.

[PMBB^{+}12] Michele Pirovano, Renato Mainetti, Gabriel Baud-Bovy, Pier Luca Lanzi, y Nunzio Alberto Borghese. Self-adaptive games for rehabilitation at home. En *2012 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, Septiembre 2012. url: `https://doi.org/10.1109/cig.2012.6374154`.

[Pow02] Daniel J. Power. *Decision Support Systems: Concepts and resources for managers*. Quorum Books, 2002.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, y Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, 2016. url: `https://arxiv.org/abs/1602.04938`.

[SC02] Qiping Shen y Jacky K.H. Chung. A group decision support system for value management studies in the construction industry. *International Journal of Project Management*, 20(3):247–252, Abril 2002. url: `https://doi.org/10.1016/s0263-7863(01)00076-x`.

[Sch96] Douglas C. Schmidt. *Architectural Patterns*, capítulo 2, páginas 44–45. Wiley, 1996.

[SCL96] P. Salvaneschi, M. Cedei, y M. Lazzari. Applying AI to structural safety monitoring and evaluation. *IEEE Expert*, 11(4):24–34, Agosto 1996. url: `https://doi.org/10.1109/64.511774`.

[SI94] H. Senay y E. Ignatius. A knowledge-based system for visualization design. *IEEE Computer Graphics and Applications*, 14(6):36–47, 1994.

[SML+21] Wojciech Samek, Gregoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, y Klaus-Robert Muller. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proceedings of the IEEE*, 109(3):247–278, Marzo 2021. url: `https://doi.org/10.1109/jproc.2021.3060483`.

[SP17] Andrew D Selbst y Julia Powles. Meaningful information and the right to explanation. *International Data Privacy Law*, 7(4):233–242, Noviembre 2017. url: `https://doi.org/10.1093/idpl/ipx022`.

[Spr80] Ralph H. Sprague. A Framework for the Development of Decision Support Systems. *MIS Quarterly*, 4(4):1, Diciembre 1980. url: `https://doi.org/10.2307/248957`.

[SPS+20] Karan Sheth, Keyur Patel, Het Shah, Sudeep Tanwar, Rajesh Gupta, y Neeraj Kumar. A taxonomy of AI techniques for 6G communication networks. *Computer Communications*, 161:279–303, Septiembre 2020. url: `https://doi.org/10.1016/j.comcom.2020.07.035`.

[SSG+10] Mareike Schulze, Bianying Song, Matthias Gietzelt, Klaus-Hendrik Wolf, Riana Kayser, Uwe Tegtbur, y Michael Marschollek. Supporting rehabilitation training of COPD patients through multivariate sensor-based monitoring and autonomous control using a Bayesian network: prototype and results of a feasibility study. *Informatics for Health and Social Care*, 35(3-4):144–156, Septiembre 2010. url: `https://doi.org/10.3109/17538157.2010.528659`.

[Ste00] Todd Andrew Stephenson. An Introduction to Bayesian Network Theory and Usage. *IDIAP*, 2000. url: `http://infoscience.epfl.ch/record/82584`.

[STY14] Jia-Hao Sun, Lan Tan, y Jin-Tai Yu. Post-stroke cognitive impairment: epidemiology, mechanisms and management. *Annals of Translational Medicine*, 2(8), 2014.

[SY94] Amir A. Sadrian y Yong S. Yoon. A Procurement Decision Support System in Business Volume Discount Environments. *Operations Research*, 42(1):14–23, Febrero 1994. url: `https://doi.org/10.1287/opre.42.1.14`.

[SZC+13] Simonetta Scalvini, Emanuela Zanelli, Laura Comini, Margherita Dalla Tomba, Giovanni Troise, Oreste Febo, y Amerigo Giordano. Home-Based Versus In-Hospital Cardiac Rehabilitation After Cardiac Surgery: A Nonrandomized Controlled Study. *Physical Therapy*, 93(8):1073–1083, Agosto 2013. url: `https://doi.org/10.2522/ptj.20120212`.

[TWD+16] Gustavo F. Tondello, Rina R. Wehbe, Lisa Diamond, Marc Busch, Andrzej Marczewski, y Lennart E. Nacke. The Gamification User Types Hexad Scale. En *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. ACM, Octubre 2016. url: `https://doi.org/10.1145/2967934.2968082`.

[vdVKGV22] Bas H.M. van der Velden, Hugo J. Kuijf, Kenneth G.A. Gilhuijs, y Max A. Viergever. Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis*, 79:102470, Julio 2022. url: `https://doi.org/10.1016/j.media.2022.102470`.

[VFW00] Vivek K. Varma, Ian Ferguson, y Ian Wild. Decision support system for the sustainable forest management. *Forest Ecology and Management*, 128(1-2):49–55, Marzo 2000. url: `https://doi.org/10.1016/s0378-1127(99)00271-6`.

[WLZ+20] Huaizhi Wang, Yangyang Liu, Bin Zhou, Canbing Li, Guangzhong Cao, Nikolai Voropai, y Evgeny Barakhtenko. Taxonomy research of artificial intelligence for deterministic solar power forecasting. *Energy Conversion and Management*, 214:112909, Junio 2020. url: `https://doi.org/10.1016/j.enconman.2020.112909`.

[WWWL16] Weibin Wu, Deli Wang, Tianyunyang Wang, y Ming Liu. A personalized limb rehabilitation training system for stroke patients. En *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, Diciembre 2016. url: `https://doi.org/10.1109/robio.2016.7866610`.

[YC09] Kenneth S Yew y Eric Cheng. Acute Stroke Diagnosis. *Am. Fam. Physician*, 80(1):33–40, Julio 2009. url: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2722757/`.

Este documento fue editado y tipografiado con LaTeX empleando
la clase **esi-tfg** (versión 0.20181017) que se puede encontrar en:
`https://bitbucket.org/esi_atc/esi-tfg`