Universitatea Tehnică a Moldovei

Utility Application for Hand Kinetotherapy

Aplicaţie utilitară pentru kinetoterapia mâinii

Student:

Barbaroş Nicolae

Conducător:

lector superior, Dumitru Ciorbă

Chişinău - 2016

Ministerul Educaţiei al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea de Calculatoare, Informatică şi Microelectronică
Filiera Anglofonă „Computer Science”

Admis la susţinere
Prof. dr. hab. Viorel Bostan,
Director Filieră Anglofonă

_____

„__”_____ 2016

Utility Application for Hand Kinetotherapy

Proiect de licenţă

Student: N.Barbaroş (_____)
Conducător: D.Ciorbă (_____)
Consultanți: G.Covdii (_____)
V.Bostan (_____)
M.Balan (_____)
E. Gogoi (_____)

Chişinău 2016

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică și Microelectronică

Filiera Anglofonă „Computer Science”

Aprob.
Prof. dr. hab. Viorel Bostan,
Director Filieră Anglofonă

_____

„__”_____ 2015

CAIET DE SARCINI

pentru proiectul de licență al studentului

Barbaroş Nicolae

(numele şi prenumele studentului)

1. Tema proiectului de licență: *Aplicație utilitară pentru kinetoterapia mâinii*

confirmată prin hotărârea Consiliului facultății de la „ *21* ” *octombrie 2015*

2. Termenul limită de prezentare a proiectului *31.05.2016*

3. Date inițiale pentru elaborarea proiectului *Sarcina pentru elaborarea proiectului de diplomă.*

4. Conținutul memoriului explicativ

Introducere
1. *Analiza domeniului de studiu*
2. *Analiza tehnologiilor disponibile și Implementarea*
3. *Arhitectura sistemului și experiența utilizatorului*
4. *Argumentarea economică*

Concluzii

5. Conținutul părții grafice a proiectului

Diagrama use-case generală a sistemului, Interfața principală a programului.

6. Lista consultanţilor:

Consultant	Capitol	Confirmarea realizării activităţii	
		Semnătura consultantului (data)	Semnătura studentului (data)
G.Covdii	Argumentarea economică		
E. Gogoi	Controlul calității		
M. Balan	Controlul calității		
V. Bostan	Standarde tehnologice		

7. Data înmânării caietului de sarcini *01.09.2015*

Conducător

semnătura

Sarcina a fost luată pentru a fi executată

de către studentul *01.09.2015*

semnătura, data

PLAN CALENDARISTIC

Nr. crt.	Denumirea etapelor de proiectare	Termenul de realizare a etapelor	Nota
1	*Elaborarea sarcinii, primirea datelor pentru sarcină*	*01.09.15– 30.09.15*	*10%*
2	*Studierea literaturii de domeniu*	*01.10.15– 30.11.15*	*20%*
3	*Alegerea și pregătirea de lucru a softului*	*01.12.15 – 25.12.15*	*20%*
4	*Realizarea programului*	*16.01.16 – 30.04.16*	*25%*
5	*Descrierea programului, diagramele UML*	*01.05.16 – 15.05.16*	*10%*
6	*Testarea aplicației*	*16.05.16– 28.05.16*	*10%*
7	*Finisarea proiectului*	*29.05.16– 31.05.16*	*5%*

Student *Barbaroş Nicolae ( )*

Conducător de proiect *Dumitru Ciorbă ( )*

UNIVERSITATEA TEHNICĂ A MOLDOVEI
FACULTATEA CALCULATOARE, INFORMATICĂ ŞI MICROELECTRONICĂ

AVIZ
la proiectul de licenţă cu tema
Utility Application for Hand Kinetotherapy
Aplicație utilitară pentru kinetoterapia mâinii
Studentul: ***Barbaroș Nicolae*** *gr. FAF-121.*

1. Actualitatea temei: *Creşterea considerabilă a numărului de pacienți cu atacuri cerebrale sau probleme de mână aduce la schimbări în modul în care se petrece un curs de reabilitare. Technologiile noi aduc noi posibilități și pot rezolva unele probleme pe care cu ceva timp în urmă erau imposibile, de exemplu exercițiile de reabilitare. În proiectul de licență se prezintă o aplicație care să prezinte unui pacient cu problemă la mână o metodă de reabilitare nouă, nemaiîntîlnită în Republica Moldova. Un instrument pentru exerciții la mână în care pacientul să combine folosul exercițiilor cu plăcutul, cu un system pus bun la punct de vizualizare și feedback realtime;*
2. Caracteristica tezei de licenţă: *Teza a fost realizată conform tututor cerințelor și standardelor în vigoare. Teza prezintă o analiză a soluțiilor existente pe piață, precum și definește caracteristicile de bază ale soluției create. Sunt prezentate statistici și sunt descrise tehnologiile moderne utilizate cu exemplificări de cod sursă și analiza arhitecturii sitemului prin prezentarea diagramelor UML și efectuarea unei analize minuțioasă a beneficiilor pentru utilizator. În final este prezentată analiza economică a sistemului prin calcularea tuturor cheltuielilor și a veniturilor;*
3. Analiza prototipului *a fost realizată în corespundere cu cerințele setate în timpul planificării aplicației. Prototipul este funcțional și cuprinde interacțiuni cu API de la Leap Motion.*
4. Estimarea rezultatelor obţinute: *Aplicația funcționează* ***corect*** *(în corespundere cu obiectivele tezei), iar metoda elaborată permite crearea unor seturi limitate de exerciții pentru mână în scopul reabilitării unui pacient ce a suferit în urma unui atac cerebral sau a avut alte motive de accidentare.*
5. Corectitudinea materialului expus: *Teza corespunde cerinţelor înaintate, iar corectitudinea materialului este argumentată de funcționarea corectă a aplicației;*
6. Calitatea materialului grafic: *Diagramele UML prezentate în teză corespund standardul UML 2.0. Figurile atașate sunt relevante tematicii și completează descrierea textuală a lucrării cu detaliile necesare. În teză sunt utilizate diagrame de componente, de clasă, de activitate și interacțiune, secvență, precum și alte elemente grafice, cu ajutorul cărora este posibil de analizat sistemul din mai multe puncte de vedere. Conținutul diagramelor este unul ce permite înțelegerea modului de funcționare a sistemului și componentele care interacționează pentru a crea sistemul funcțional în întregime.*
7. Valoarea practică a tezei *este bazată pe un studiu bine efectuat și este determinată de soluționarea unei probleme actuale pentru oamenii ce au problem cu mușchii și sistemul nervos al mâinii;*
8. Observaţii şi recomandări: *În perspectiva necesităților și a numărului de pacienți cu atacuri cerebrale sau accidentări la mâini în creștere se recomandă adăugarea a mai multor tipuri de exerciții și îmbunătățirea a sistemului prin gamificarea lui.*
9. Caracteristica studentului şi titlul conferit: *Studentul s-a prezentat la consultații conform orarului, a respectat indicațiile conducătorului, a studiat diverse articole și resurse bibliografice. A respectat termenii de realizare a proiectului de licență.*
Rezultatele obținute în cadrul tezei îmi permit să recomand admiterea tezei de licență spre susținere și să fie apreciată cu nota _____, iar dlui *Barbaroș Nicolae* de conferit titlul de inginer licențiat în Tehnologii Informaționale.

07 iunie 2015

Conducătorul tezei de licenţă
lector superior
Dumitru Ciorbă

Abstract

The **Utility Application for Hand Kinetotherapy** presented by student Barbaroș Nicolae as a Bachelor project and it was developed at the Technical University of Moldova. It is written in English and contains 66 pages, 19 figures, 22 listings and 16 references. The thesis consist of a list of figures, list of abbreviations, list of listings,introduction, four chapters, conclusions, and references list.

The thesis has the object of studying the existing solution in the market of a rehabilitation system of an patient that got a stroke, hand fracture or a surgery and followed hand pain or hand paralysis and eventually creation of an application that will aim to treat patients with hand pain, tendons injuries and neuro disorders.

With the help of Leap Motion technology, the application is capable to track the users hands in realtime. Using the tracked data offered by the controller about hand activities, the user is provided with a set of well defined exercises for a great rehabilitation experience and a feedback sistem so that every time the user will do an exercise, he will be notified about the correctness of the did exercise and the counted done exercises.

Hence the application does not require the use of extra complicated objects that will track users hand and only a small, easy, cheaper device that is connected only by an USB cable to the laptop, the user will be capable of having the device by his side and even in an airplain he can connect the device to the laptop, open the application, and start exercising.

The four chapter which compose the report are: the problem, domain analysis and the purposed solution chapter. The second one is the used technologies and implementation chapter, followed by the UML description of the system and user experience chapter, concluding with the chapter four about the economical analysis of the system. The first chapter describes the problem that was identified and defines the solution. It also has a deep analysis of the market and existing solutions. In the chapter two are presented the UML diagrams of the system and is listed the benefits of the user when using this application. In chapter three are presented the used technologies and some basic instructions on how to use them. Also there is described the implementation part with sample listings of code. In the last chapter is made a economical analysis in which are shown the total expenses, possible incomes and is analysed the profitability of the project. This document is for readers with technical background, engineers, IT students and programmers

Rezumat

Teza **Aplicație utilitară pentru kinetoterapia mâini** este prezentată de studentul Barbaroș Nicolae ca proiect de licență și a fost efectuată la Universitatea Tehnică din Moldova. Teza este scrisă in limba engleză și conține 66 pagini, 19 figuri, 22 de listări de cod și 16 referințe. Teza conține o listă de figuri, o listă de abrevieri, introducere, patru capitole, concluzie și o listă de referințe.

Teza are obiectivul de a cerceta solutiile existente pe piață în domeniul de reabilitate a unui pacient ce a avut un atac cerebral, o fractura la mână sau o interveție chirurgicală ce a adus la probleme de mobilitate a mâinilor și ulterior crearea unei aplicații ce va tinde să refacă durerile, leziunile de tendoane și dizabilitati neuromotorii ale mâinii.

Cu ajutorul tehnologiei noi Leap Motion, aplicația e capabilă să monitorizeze în realtime mâinile utilizatorilor. Utilizând datele oferite de Leap Motion despre activitățile mâinilor, le pot oferi pacienților un set de exerciții prin care își vor putea începe etapa de reabilitate și un sistem de feedback constructiv prin care să-i înștiințeze pe utilizatori despre cât de corect fac exercițiile și numarul de exerciții efectuate. Dat fiindcă aplicația nu necesită manuși de monitorizare, tehnologii cu piese mari si multe camere de monitorizare, ci un dispozitiv mic, ușor, mai ieftin (dintre toate opțiunile pe piață) și respectiv care se conectează printr-un singur cablu USB, utilizatorul va fi capabil să țină permanent cu el dispozitivul și va putea exersa chiar și în avion, având doar aplicația instalată pe laptop.

Cele patru capitole ce compun raportul tezei sunt: problema, analiza domeniului și a soluției propuse în capitolul unu. Al doilea capitol este despre diagramele UML care descriu sistemul și beneficiile utilizatorului, urmat de capitolul trei cu tehnologiile utilizate și implementarea codului sursă, finalizând cu capitolul despre analiz economică a sistemului. Primul capitol descrie problema care a fost identificată și descrie soluția care a fost găsită. De asemenea este făcută o analiză a pieței locale și a soluțiilor existente. În capitolul doi sunt prezentate tehnologiile și cîteva instrucțiuni de bază despre cum putem să le utilizăm. De asemenea este descrisă implementarea codului sursă cu exemple de cod listate. În capitolul trei sunt prezentate diagramele UML ale sistemului și beneficiile utilizatorului pentru a folosi acest sistem. În ultimul capitol este facută o analiză economică, sunt prezentate cheltuielile, veniturile posibile și este analizată profitabilitatea proiectului. Acest document este destinat pentru cititori cu cunoștințe în domeniul technic, ingineri, studenți TI și programatori.

Table of contents

List of Figures

Listings

Abbreviations

API	Application Programming Interface
CIMT	Constraint Induced Movement Therapy
IR	Infrared
OM	Online Marketing
NUI	Natural User Interface
LED	Light-Emitting Diode
SM	Social Media
SaaS	Software as a Service
UX	Experienta Utilizatorului
UML	Unified Modeling Language
UX	User Experience
UI	User Interface
VR	Virtual Reality

Introduction

Whether you are an athlete, or you had an undergone surgery, was paralyzed due a stroke or maybe you are just an usuall human being hoping to rid yourself of hand pain – kinetotherapy is a great recovery strategy to help get read or heal off your injury. The goal of kinetotherapy is getting read of injures as much as possible and promote muscular function and mobility, to work the muscles and the area with an injury, to strengthen the muscles and joints so that they perform better. Instead of depending on drug usage kinetotherapy recommends to rely on the body's ordinary resources of physical recovery through exercises.

Due to the expansions of medical-related technology new types of rehabilitation of patients are introduces. Here comes Kyno, a desktop application that helps users through rehabilitation by restoring movement and function of the hand which is affected by injury, illness or disability.

One of Kynos solutions, aims to treat patients with hand pain, tendons injuries and neuro disorders. With the help of a tracking device Kyno can manipulate the data and send the learning acitivity in the "virtual" environment, letting to patient's an open dor to focus more and pay attention to the details of their movements which might bring them closer to recovery.

Kyno is bringing a new wave of innovation and hope for patients everywhere. Among lots of benefits, it brings to the patients more fun while stretching and recovering their physical capacities. In my system the program with the help of Leap Motion device projects the patients hand onto a virtual environment shown on a screen. Then the patient by choosing one set of exercises will start performing it, which is aimed to improve their physical issue. While tradition kinetotherapy in most cases forces to use weights or other tools, kyno omits the usage of them.

More then that traditional old kinetotherapy is forced to work in strict physical places, while using Kyno the patiens will not only strengthen their muscles or recover their injuries, but also will improve their cognitive abilities due to exploring a brand new borned application. He will have to understand how the application works and then how he should act in it. For instance, choosing what exercise to take involves decision taking(which on to choose), motor skill (moving the mouse over a button and then click it), attention (sustaining concentration on the virtual hand while performing the exercise it is a brain challenge).

The program uses infrared light with a wavelength of 850 nanometers, which is outside the visible light spectrum to track and analyze the movement of objects that are withing the range of the Leap Motion controller that connects the real world with the computer. During therapy the patients stays in front of the screen with the hand above the controller, where he can watch his own hands generated in the virtual world among some UI. The patient's work is to choose one type of exercise provided by the application, to read the instructions and start the process of rehabilitation. Modern technology works in a fraction of a second, so thanks to that the application is able to provide a realtime feedback that indicates if the exercises were done correctly or not. The application is designed with 3 exercises that are intended to improve the motor skills and neuro capability of the hand and fingers.

This thesis is composed of 4 chapters, a list of abbreviations, a list of figures, a table of contents, conclusion and reference. The report has roughly 70 pages. Starting with the first chapter where its

been described the problem that needs to be solved, the solution that is purposed and the current market analysis.The second chapter describes the system architecture via UML diagrams and has a succint description of the UX. The third chapter is mostly about the technologies that are used and describes how the system is realized, with chunks of sample code listings. The fourth chapter analyses the project from the point of economics view, where the expenses of building the project, the marketing plan and the profit it can make can be seen.

1 System analysis

1.1 Motivation and the problem description

A stroke[1] is a brain attack. It happens when the blood supply to part of your brain is cut off. Can be a devastating experience, leaving the patient with serious physical impairments and beset by concerns for the future. Today, that future is much brighter, as stroke rehabilitation has made enormous strides. First, and most importantly, researchers are working to improve patients' compliance with their rehabilitation regimen, since up to 65 % [2] of patients fail to adhere fully—or at all—with their programs. In addition, they are addressing the lack of accessibility and the high cost associated with rehabilitation. If you have just had a stroke, even getting to the clinic is a challenge, and the cost of hiring a private physical therapist to come to your home is too high for most people.

1.1.1 Physical challenges after stroke

All strokes are different so for some people the effects may be relatively minor and may not last long, while others may be left with more serious long term problems.

A stroke can affect the way your body functions Although all strokes are different, there are some common physical problems that many people experience:

- **problems with movement and balance**, many people experience muscle weakness or paralysis after a stroke, which can affect your mobility and balance. This usually happens on one side of your body and can also cause a lot of pain and discomfort;
- Problems with your vision;
- Problems with swallowing;
- Excessive tiredness.

But there are other effects that can not been seen. Some of the 'hidden' effects of stroke include:

- **Problem with communication**, many people have difficulty with speech and language after their stroke. A common communication problems, which affects around one third of stroke survivors, is aphasia. People with aphasia find it difficult to speak and understand what other people are saying to them, as well as reading and writing;
- **Problems with memory and thinking**, it is very common to find that their short-term memory and concentration is affected by stroke, but it can also affect other thinking processes as well, such as problem-solving, planning and finding your way around;
- **Changes to your emotions**, a stroke has an emotional impact, which can lead to problems like depression and anxiety. It can also make it more difficult to control your emotions;
- Changes to your behaviour.

Given the statistics from figure 2.6, it is important to note that more and more humans everyday die of stroke related diseases which is a cause of concern. More than in Republic of Moldova, in 2008 the number of patients that had a stroke started from 13 000 in 2014 the number got more then 5 times bigger with over 70 000 stroke patients [4].

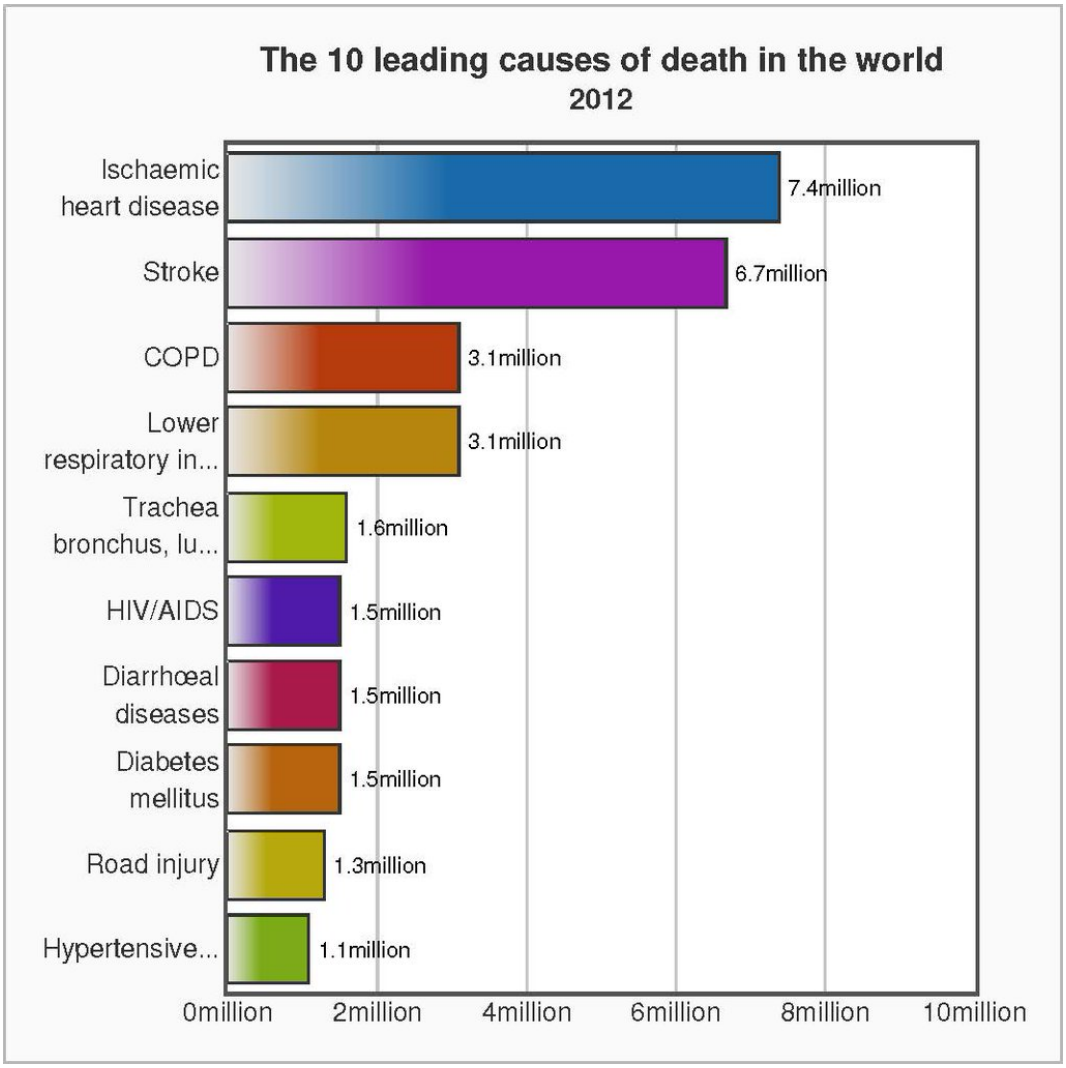

Figure 1.1– Top 10 leading causes of death in the world 2012 [3]

1.1.2 Physical therapy after stroke

Movement problems affect each person differently.Different therapies may include:

- Practicing tasks/activities that you have difficulty doing. This may include rolling over in bed, sitting or standing up. It can also include walking and using your hand or arm;
- Exercising to improve your strength, sensation (ability to sense or feel things), coordination, balance or fitness. Often this can be done as you practice normal activities such as standing or walking. Exercises that use electrical stimulation and other equipment (for example treadmills) may also be used as part of your therapy. This will help improve your ability to move;
- Joining a fitness centre, club in the community, or exercise program at your local community health care centre. This can help to keep you fit. Often after a stroke, fitness levels drop.

Therefore it is important to keep yourself as active as possible in the long-term. Talk to your therapist about how best to keep fit;

- Learning how to walk safely. This may include the help of an aid like a frame or walking stick;
- Limiting the use of your good arm to encourage use of the affected arm. This is called CIMT [13]. Research has found that 'forcing' you to use your affected arm can improve recovery of your affected arm. It is important to talk to your therapist first.

1.2 Stroke Case Study

Now Consider Maria, a 56-year-old patient. After experiencing a stroke 5 months ago, she now has difficulty on controlling the left hand of her body. Like most stroke victims, Maria faces one to two weekly therapy sessions for up to 1 year. Unable to work, she worries about the fee per visit, as she has exhausted her insurance coverage. Maria also have to exercise hours daily to maintain her mobility. Unfortunately, the doctor gave her boring repetitive exercises, and Maria finds it diffucult to motivate herself to do them.

What is the solution? Kyno offers a significant advance to help stroke patients restore their physical functions: an affordable motion-capture system for physical rehabilitation that uses Leap Motion technology. Kyno tries to solves all of these issues by providing patients with "gamified" exercises that accelerate recovery and increase adherence. In addition, Kyno gives patients immediate feedback, which ensures that they perform their movements correctly. This is critical when the patient is exercising at home.

1.3 Existing solutions and their drawbacks

Before getting to implement the idea is better to do a market research for finding similar solutions to the problems that are being used by those applications.
After analysing these solutions Kyno application will become better, stronger, different and adapted tool for local market and attractive for patient's and customers. It is important to know what the user wants the most. What need to be done to make him happy, to make him have a great experience while using the software. Thats why more than that it is important to know the background for solution, the countless features that are counted as advantages and the ones that are not crucial. It is important to know what features will offer him confort or excitement.

There was created a list of solutions after a quick research by searching the web for similar solutions and after to select only the best one, only those who have something common with Kyno, others where taken out, because the purpose of those application's wasn't as the purpose of Kyno. The best existing solutions at the moment for patients kinetotherapy comes from big companies and these are:

- VirtualRehab;
- Jintronix;

– SeeMe.

What I have seen, is that there are quite a lot of application of kinetoteraphy, each of them uses Microsoft Kinect to detect the motion, each of them is great amazing aplications but what is the problem in them is that each of them detects the full body of the patients and is provided most of the times with exercises/list of mini games where you have to move. However that is bad for patients that are paralyzed, that can't walk, that can't even move their legs.

1.3.1 VirtualRehab application

The first solution to kinetotherapy rehabilitation is *www.virtualrehab.info*. In my opinion it is one of the most powerful rehabilitation application on the market.

This tool provides functional training so as to improve equilibrium, coordination, weakness, fatigue and spasticity. The exercises can be adapted to a patient's disability levels so that the programme can be used with a wide range of ability levels.

The strong advantage of VirtualRehab is that it has 2 type of application VirtualRehab Body and VirtualRehab Hands. VirtualRehab Body is a suite of therapeutic games designed to help retrain upper and lower limb motor functions. Through the use of a variety of highly motivating games, the system makes it possible to retrain abilities such as balance (while sitting and standing), thrust inhibition, load transfer and changing between sitting and standing positions. VirtualRehab Hands works the mobility and strengthening of the muscles used in flexion, joining, separation and extension of the fingers. This possibility offers to cover a full controll of body rehabilitation.

A second great advantage is that the rehabilitation process in VirtualRehab is made through games. It has 9 games that help treat various physical symptoms.

Another great feature is the feedback system of this applications. With the help of Kinect they are capable to keep track of progress and give feedback in real time to the patient about the correct or incorrect movement performes. This feature is going to be implemented in Kyno.

Even more it has a simple patient management panel. It ncludes an easy to use therapy editor that allows therapists to program customized therapy sessions taking into account each patient's particular needs. All the information from the sessions is stored in the data server immediately making it possible to track and monitor each patient´s progress in the prescribed therapy.

1.3.2 Jintronix application

Jintronix is transforming rehabilitation by providing an innovative, accessible and value-driven model for the delivery of physical and occupational therapy.

Combining kinect motion tracking, virtual gaming and remote clinical monitoring, Jintronix offers patients a fun and effective tool for their rehabilitation through games. The games were developed through researching which exercises and sports best fitted with conventional therapy. The rehab modules are adaptable to the level of a patient's functional and cognitive abilities and are designed to train balance and mobility, muscle strengthening and endurance, flexibility and range of motion, fall prevention, postural control, motor control and relearning and bilateral coordination.

Jintronix is capable of tracking a players' movements to see whether they are performing the activities correctly and relays this to the therapist who can then make adjustments. So far, there are seven games that have been created and work smoothly.

1.3.3 SeeMe application

A third solution to Kyno is SeeMe at *www.virtual-reality-rehabilitation.info* SeeMe provides active training in the form of games – what makes patients more motivated to participate in their rehabilitation process. SeeMe creates a feedback loop between a patient performing rehabilitation exercises and a physical therapist. In real time the physical therapist can monitor the patient's performance and adjust parameters of current "gamified" exercise to match the patient's individual recovery needs.

These are the great key features of SeeMe at this moment:

- **Deep Customization**, each exercise can be personally customized to meed the specific requirements of the patient. All the tasks customizations can be done in real time while patient is playing;
- **Many Applicaitons**, SeeMe uses a wide variety of therapeutic tasks to enable training in all rehabilitation domains;
- **Engaging Activities**, all the therapeutic tasks included in SeeMe offer plety of paratemters and leverls. By having those options - therapists are able to prepare trainings that let patients experience positive emotions, keep motivation, become more self-confident and in the same time remain challenged;
- **Powerful Reports**, enables detailed insight into the course of each training and long-term progress as well. Therapists can collect objective results of treatment progress.

SeeMe it is a very comprehnsive tool which makes the patient/user happy, it brings to the patient a new way of rehabilitation, home rehabilitation with lots of interactive fun to play games.

1.4 Proposed solution

Kyno tries to solves all of these issues by providing patients with "gamified" exercises that accelerate recovery and increase adherence. In addition, Kyno gives patients immediate feedback, which ensures that they perform their movements correctly. This is critical when the patient is exercising at home. However to create a really useful application, it is important to know what the market really needs.

Kyno should be capable of giving the following solutions :

- Helping getting read of injuries, promote muscular function and mobility, to work muscles and the area with injuries, strengthen the muscles and joints so that they perform better;
- During the exercise the tools that will help on rehabilitation will be the users hands only(not other tools, weights);

- The UI should be intuitive, easy to use, clean and not very complicated. The UX must be great with not so many pop ups, additional panels. The application should provide to the user at least 3 options of exercises which are most used and most useful for injuries recovery;
- Have a price accordingly to the local and external market. Current solutions that involves technologies are way way more expensive then what it is proprosed. Thus, the price range must be affordable for the local and external market;
- The application will be developed and build for desktop's with later support for Virtual Reality which will give a brand new exited experience for patients.

There are numerous rehabilitation hand exercises to which Kyno could work, but there should be chosed a couple of them for initial market test they are written in the itemized list further and for launching the project, after which there can be added other exercises as well. Also due to the Leap Motion device tracking limitation some of the exercises can not be included because Leap Motion is not able to "see through the fingers" - for example, when one finger covers the other. Fingers right next to each other also pose a problem for the cameras and might not be recognized individually. This is not a device which incorporates science fiction hardware with x-ray vision and magical recognition properties - even though some buyers on the bleeding edge might have expected exactly that. That's why some exercises cannot be done.

- **Grab**.
- **Pinch**.
- **Roll**.

Some key features/benefits of the application:

- **Gains**.Kyno will use a wide range of kinetotherapeutic exercises to train the following rehabilitation domains:

  Musculo-skeletal

  a) Range of motion;

  b) Strength;

  c) Endurance;

  d) Fitness and cardiovascular training.

  Balance and Equilibrum

  a) Self control;

  b) Anticipatory postural responses;

  c) Adequate reactions to stimuli and distractors placed in preplanned positions or random.

  Neurological

a) Hand movement quality;

b) Hand movement awareness and proprioception;

c) Bilateral movements in response to bilateral stimuli.

Cognition

a) Memory;

b) Perception;

c) Planning and executive functions.

– **Engaging activities**. There is no need to wear, hold or be attached to any equipment – patients can almost forget it is still a real rehabilitation.

2 Architecture of the System

2.1 Design visualization of the system

The Unified Modeling Language (UML)[5] is a development and all use modeling language in the field of software engineering. Is indended to assure a standard way of visualizing the design of the system that it was made for.

In the current chapter is represented and described the architecture of Kyno application. It contains a set of relevant diagrams modeled in UML language. The diagrams provide a fundamental documentation an description of the system structure and behavior.

- Use Case Diagram;
- Deployment Diagram;
- Class Diagram;
- Sequence Diagram;
- Activity Diagram;
- State Diagram.

These diagrams will illustrate the users possibilities, the system architecture and will also illustrate the procedures of interaction between the modules.

2.1.1 Functionality of the system

To model a system the most important aspect is to capture the dynamic behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

So to model the entire system n numbers of use case diagrams are used.

Purpose of Use Case diagram is:

- Used to gather requirements of a system;
- Identify external and internal factors influencing the system;
- Show the interacting among the requirements are actors;
- Used to get an outside view of a system.

In the figure 2.1 , is shown the process of the user that interacts with the application. Therefore 2 use case diagrams were modeled to show the set of avaiable actiones offered to the user's. The client part of the application represents an executable. As we can see the most important stages of the project is around user's exercises, later in figure 2.2 we will see the type of exercises the user can do. The possibility of giving feedback to the application is offered. Also, the user can view the results after the end of the exercise or at the moment of doing the exercise.

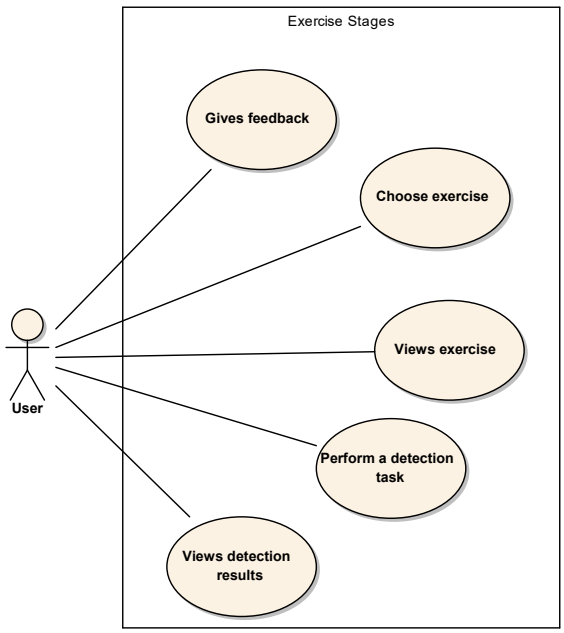

Figure 2.1– General Use Case diagram of the system.

Now back to the figure 2.2 where the operations can be seen, there are 4 main actions an user can perform. When the user opens the application he will be provided with a menu he can choose one of this 4 actions from there. By choosing the first action which is grab the user will have to open and close slowly his hand n times. The second action is pinch, this action is a little bit more complicated since it will make the user to touch every finger, one after one, with the thumb. Next action and the third one is rotate. Rotate is rather simple, the user will have to rotate his hand horizontally untill its reaching the position of completing one rotation. The forth and the last action is movement, its not that complicated, just moving the hand from left to right and right to left will be counted as 1 movement. A text panel with results and a second panel with tips will be displayed on the screen for the user on each active action.

2.1.2 Topology of the physical components

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. The purpose of deployment diagrams can be described as:

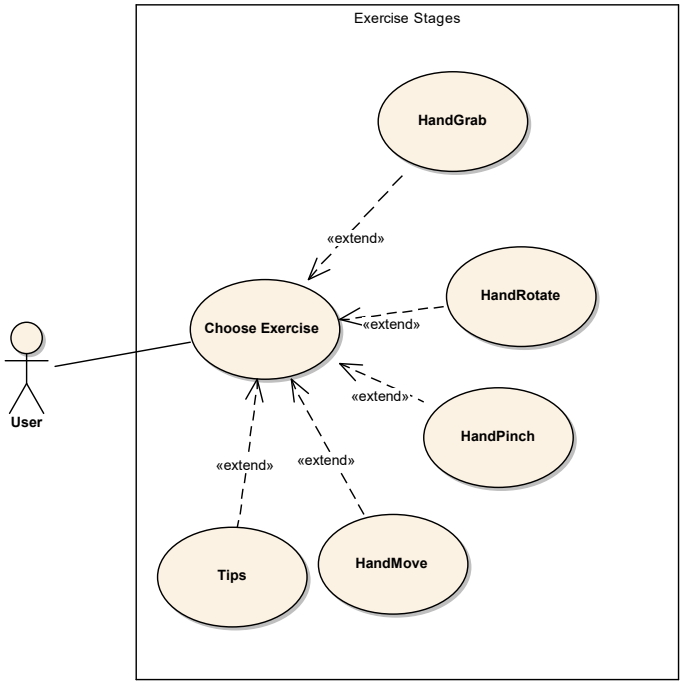

Figure 2.2– Use Case diagram of exercise action.

– Describe runtime processing nodes;

– Describe the hardware components used to deploy software components;

– Visualize hardware topology of a system.

In the figure 2.3, is described how the system is setup on the physical level. On the left there is the executable application used by the user in order to interact with the application. The user creates a connection with the Leap Motion device through a connector (USB), by pluggin the device into an USB port. The request is processed by the application and it checks whether the leapmotion is connected or not. Also at the runtime a JSON library runs and it loads from a JSON file text information into a dictionary where eventually will be displayed in the application.

2.1.3 Modelling of the object oriented system

The class diagram is a static diagram and itrepresents the static view of an application. Describes the attributes and operations of a class and more than that, also the constraints appointed on the system. Because class diagrams are the only UML diagrams that can be mapped directly with object oriented languages it is widely used in the modelling of the object oriented system and at the time of system construction.

Purpose of the class diagram can be summarized in:

– Analysis and design of the static view of an application;

– Describe responsibilities of a system;

– Base for component and deployment diagrams;

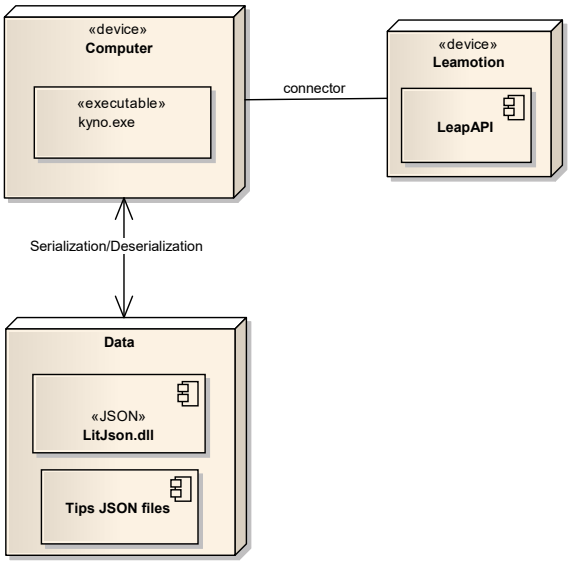

Figure 2.3– Deploy diagram of the system.

– Forward and reverse engineering.

In the first class diagram which is figure 2.4 is described the UI of the application. Animation and how the menu is generated. How the pannels move, from left to right and right to left.

One of the classes, illustrated in figure 2.4, depicts the parsing objects. The conceptual model is done in an analogical way. The defined class has a set of predefined messages that denotes the possible actions needed to successfully push a button, get info from json file and finally set info to the corresponding text object.

The last step of Kyno application is gesture tracking part. The class structure is much simpler, see figure 2.5. Manager Script class has association relationships with LeapPinchDetector. It is used to set tracking data to transform it into a pinching gesture.

2.1.4 Component interaction of the system

In the figure 2.6 is represented the action of doing an exercise. In order to do that, the user must press on one of the buttons shown in the menu, after pressing one of the button, it will make a function call to the Leap Motion, by asking him to grant access on that action. The Leap Motion will send back to the user info about what button was pressed and what action the user can do now. After the user pressed on of these buttons he can do now the action itself. Also after every set of exercise in that action, the user is able to retrieve the status of his action from Leap Motion

From the first glance each action behavior looks similar, having the same interface, however each has unique characteristics worth checking out. The entire process is straight and monotonous. Anyway there has to be some conditions which is the LeapMotion device is required to be connected to the users device.

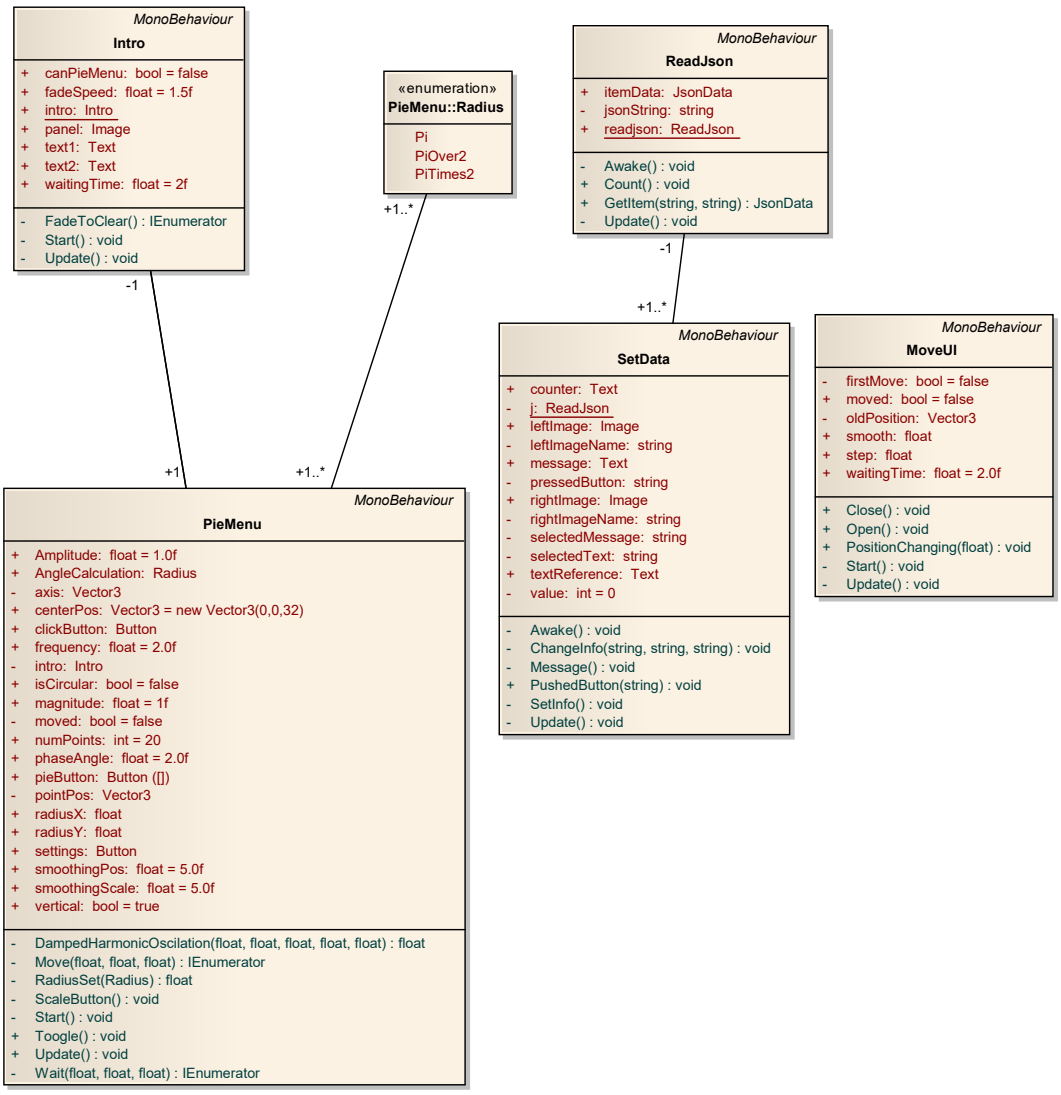

Figure 2.4– Class Diagram of User Interface and User Interaction.

2.1.5 Dynamics aspects of the system

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. This diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system.

So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

Purpose of activity diagram can be:

– Draw the activity flow of a system;

– Describe the sequence from one activity to another;

– Describe the parallel, branched and concurrent flow of the system.

By now every step of exercising was particularly described. The activity diagram from figure 2.7, represents the list of actions done on the exercise part of the application. Every executed step

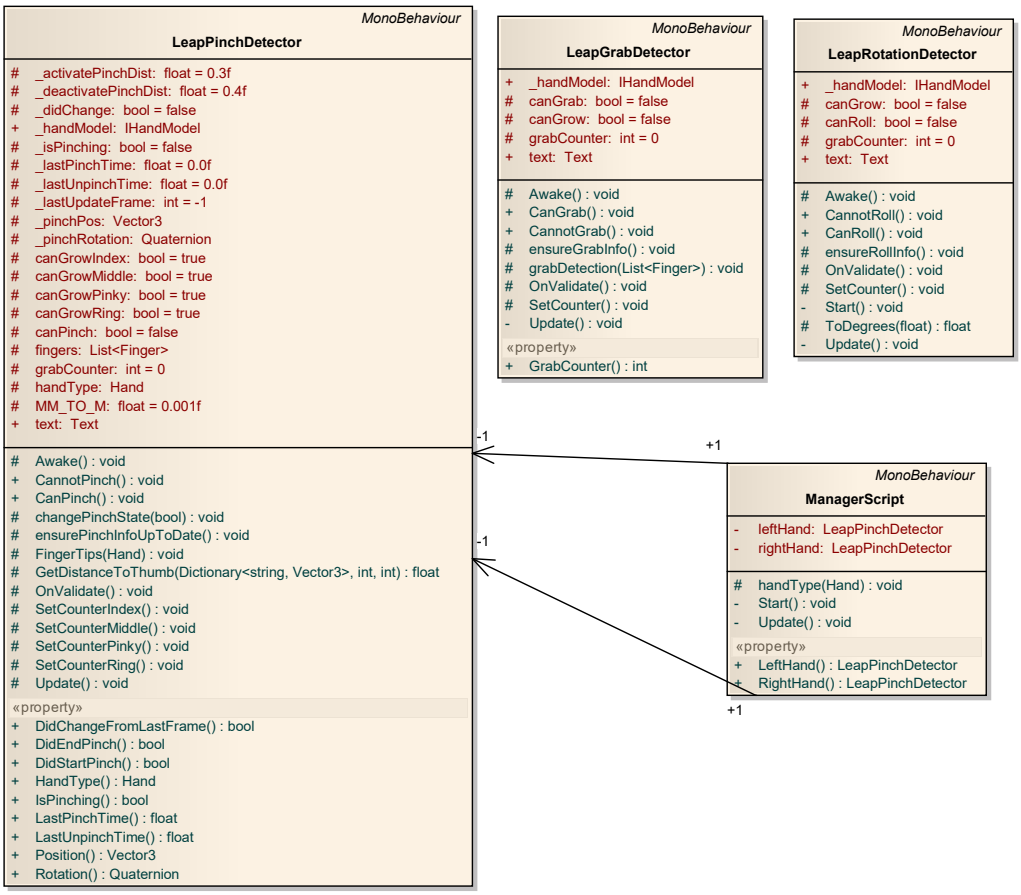

Figure 2.5– Class Diagram of Leap Motion gesture tracking.

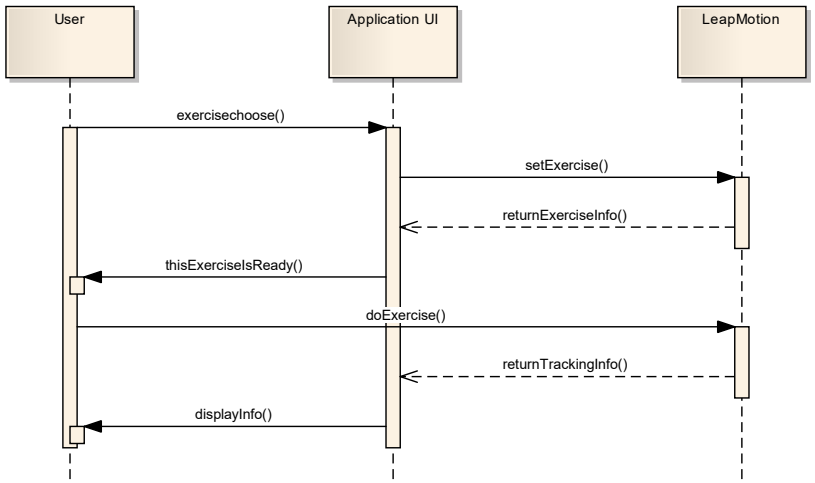

Figure 2.6– Sequence diagram of doing an exercise set.

in the chain depends on the previous one.

Before selecting an action from menu the user must to connect the Leap Motion device. When the status of Leap Motion device is connected the user can choose an exercise. At the exercise choosing the right tips are displayed in a panel placed in the middle most right side of the screen. While the user keeps waving his hand and performing the exercises, the Leap Motion API tracks the hands and sends the results to UI where eventually the user can see it. After done with one set of exercising the user has the option to choose another one or to exit. And there ends the process of doing a kinetoteraphy exercise.

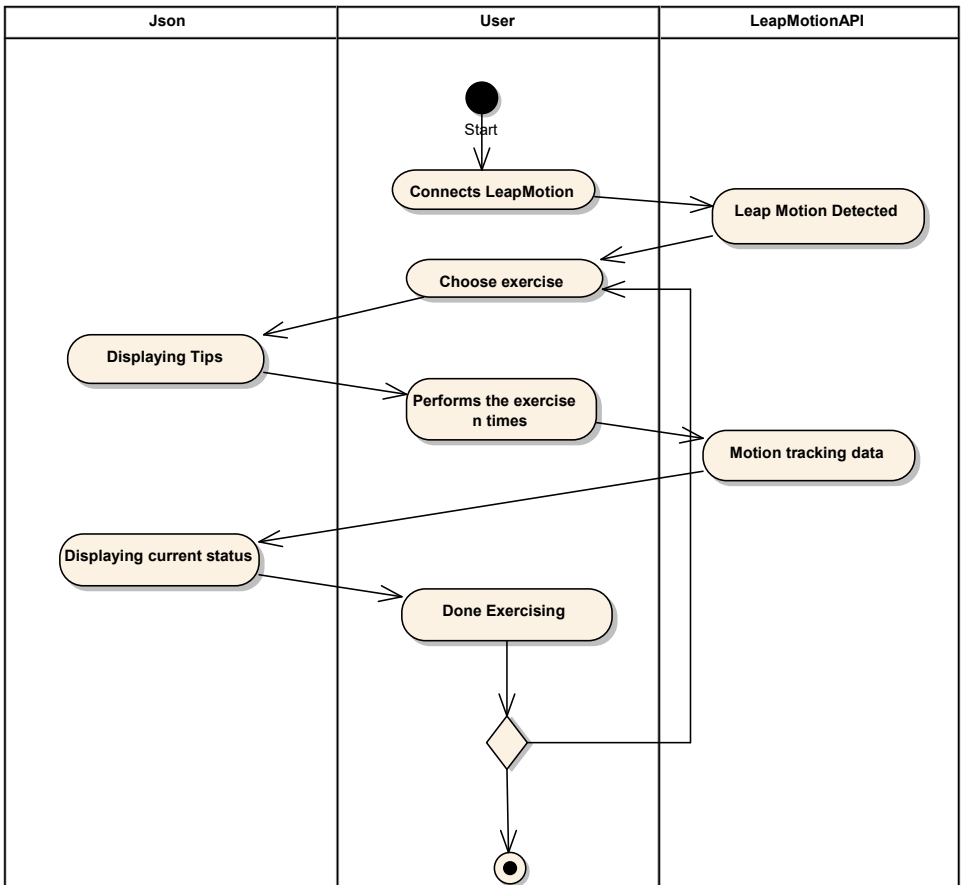

Figure 2.7– Exercise performing activity diagram.

2.1.6 Model state definition

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A Statechart diagram describes a state machine. Now to clarify it state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

Statechart diagrams are also used for forward and reverse engineering of a system. But the main purpose is to model reactive system.

Following are the main purposes of using Statechart diagrams:

– Define a state machine to model states of an object;

– To model life time of a reactive system;

– To model dynamic aspect of a system;

– To describe different states of an object during its life time.

In the figure 2.8 and 2.9 is represented the application state diagram. However due to the fact that Kyno application offers a limited amount of operations, imply that there is a small amount of states that an application user can be. All the application states are mapped to the executable.

After running the application the first step is to connect the Leap Motion device. Where in the next step the Leap Motion API detects if there is or there is not connected an Leap Motion device. On the left bottom corner of the window is rendered a pie menu that can get the user into any state of the application. The remaining states are related only to visualizing the information and the exercise action itself.

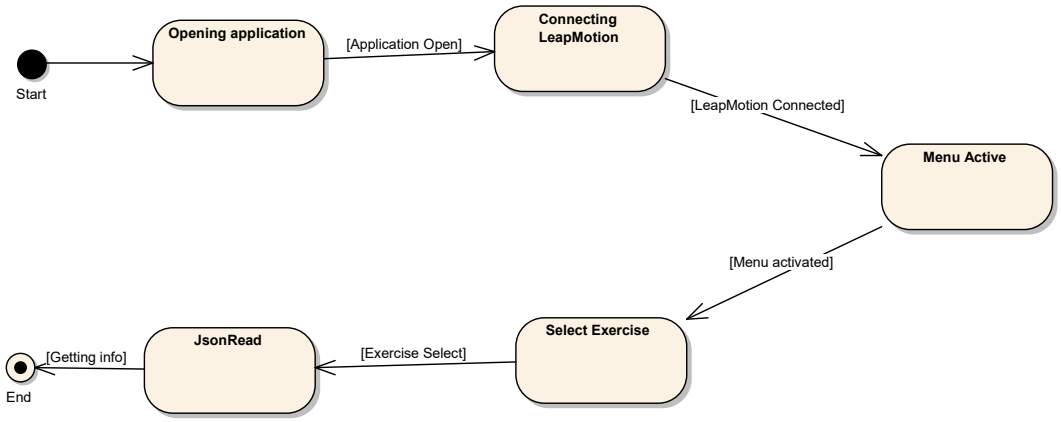

Figure 2.8– Application state diagram.

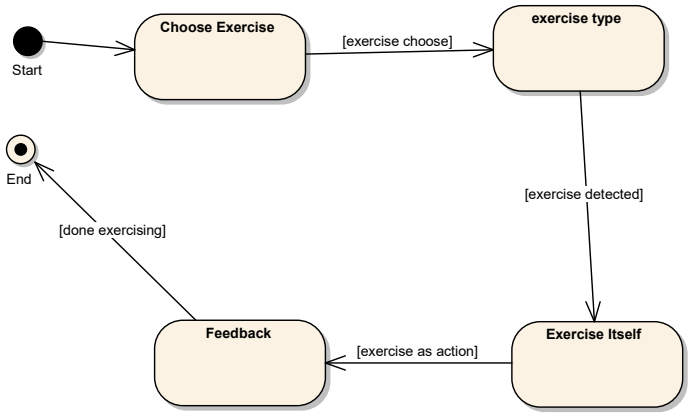

Figure 2.9– Exercising state diagram

The implementation chapter focuses on the provided UML design, implements the classes and follows the use cases and sequence diagrams to define the flow of the application. The application components and dependencies will probably extend to a bigger degree. This concludes the UML description of the project, that aimed at presenting the most relevant aspect of the system and covering the general architecture of the application. The UML methodology offered a good documentation basis and a clear view on requirements of the software.

2.2 Benefits for the user

By using Kyno, the user is provided with a list of benefits that can make their rehabilitation work better, easier, cheaper and with more results.

– **Free Hands** Kyno is not required of full rooms with sensors, gloves or other device that will be attached to user's hands. It's using Leap Motion a device that tracks hands by positioning

them above the controller as shown in figure **??**.

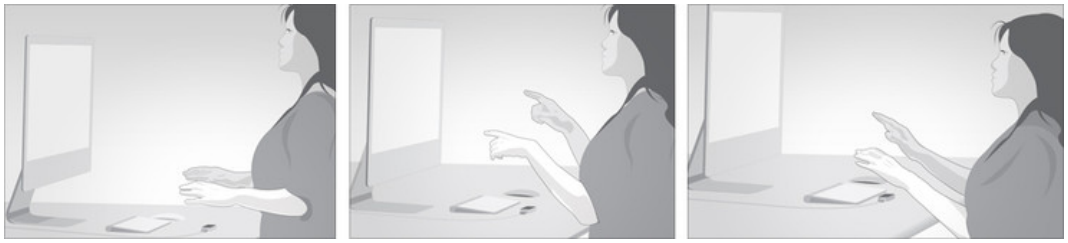

Figure 2.10– The PC in your hands [6]

– **Simple design**. Being simple is key for making the work easy and comfortable. This is why Kyno has a little number of buttons, a pleasant for for displaying information to the user, one type of font and nothing else that could distract the user from being productive;

– **Future improvements**. After the first iteration and feedback from it's users, the team that works on the project will implement and it will constanly improve the UX of the software;

– **The market**. Kyno is mostly developed and designed for the local market and this is a great advantage, because in Republic of Moldova the home rehabilitation after a stroke or some kind of other injury is bad and there are no tools like Kyno in our rehabilitation centers that will take a patient through a whole new adventure and experience. This is why, if there will be future investments in the project, it can actually grow at a high scale.

3 Implementation and Used Technologies

For a better understanding of the application, on how it works, how it behaves for the patient and in general for those who will use it, it is better and I would say important to describe the technologies that were used to build Kyno. In this chapter will be analysed the implementation of the features that presents the Kyno as a software, will be shown some code samples and analysed the solutions that were choosed to develop the application.

3.1 System Requirements

In order to run the application and to have a nice user experience, there is needed for the client to have 2 componenets.

- A computer, laptop or any system/device where you can install a desktop application
- A Leap Motion tracking device with again the ability to install software on your device.

As we can see the requirements are not that simple and easy to get since the Leap Motion can be bought only online from USA and cost almost 100 euros. However the application will come with the Leap Motion controller in the set.

3.2 Used technologies

3.2.1 Unity

Unity is a powerful cross-platform 3D engine and a user friendly development environment. Easy enough for the beginner and powerful enough for the expert.

Why Unity over others?

The main "pro" of Unty is that it's crazy fast. I'm not talking about performance here, but about development speed. It has:

- **Unified asset pipeline**. No need to spend time on resource subsystem at all, no buggy import routines to write and fix: just drop a file into folder, and it works.
- **Integrated level editor**. No need to spend time on level tools: just get straight to business.
- **Great tweaking and debugging support**.All your scripting variables are shown in the editor right as you play, and can be changed on the fly too - and all this without writing a single line of code. Pause the application anytime, or step through code one statement at a time.
- **Quite comprehesive library of ready-made components**. Rendering, sound, physics, controls - a lot of "boilerplate" code is already written so that you can focus on the application and not on how to create an engine.

One aspect is that Unity is a game engine and editor that publishes almost anywhere. What does it mean "everywhere"? One can divide the game world in four continents: consoles, smartphones, desktop (installed), browser. Unity runs in most "countries" from all these continents – as shown in figure 3.1.

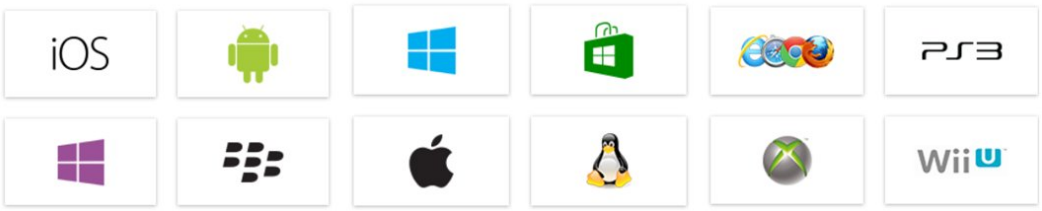

Figure 3.1 – Exercising state diagram [7]

Application created in Unity can be deployed into multiple platforms simply by downloading and installing support of these platform. After that offcourse by changing your code so that it will work on other platform too.

So let's see 2 examples in of how Unity handles a mouse input in a desktop application and fingers input in a mobile application.

```
1  void Update() {
2          if (Input.GetMouseButtonDown(0))
3              Debug.Log("Pressed left click.");
4
5          if (Input.GetMouseButtonDown(1))
6              Debug.Log("Pressed right click.");
7
8          if (Input.GetMouseButtonDown(2))
9              Debug.Log("Pressed middle click.");
10
11     }
```

Listing 1 – Mouse input for desktop application in Unity [8].

As we can see Listing 1 returns true during the frame the user pressed the given mouse button. Either will be left click, middle or right click.

```
1  void Update ()
2      {
3          Touch myTouch = Input.GetTouch(0);
4          Touch[] myTouches = Input.touches;
5          for(int i = 0; i < Input.touchCount; i++)
6          {
7              //Do something with the touches
8          }
9      }
```

Listing 2 – Multiple touch input for mobile application in Unity [9].

In Listing 2 Unity handles multi-touch by giving you the number of touches on the screen during a given frame, and/or gives you an array of all touches during a frame. While myTouch will

be the first touch the user did and myTouches will be the total amount of touches the user does.

This way Unity does for every platform, it is a console, it is a desktop computer, a mobile phone or web browser. Unity make life easire.

3.2.2 Leap Motion

The Leap Motion controller is a small USB peripheral device which is designed to be placed on a physical desktop, facing upward. It can also be mounted onto a virtual reality headset. Using two monochromatic IR cameras and three infrared LEDs, the device observes a roughly hemispherical area , to a distance of about 80 cm figure 3.2. This range is limited by LED light propagation through space, since it becomes much harder to infer your hand's position in 3D beyond a certain distance. LED light intensity is ultimately limited by the maximum current that can be drawn over the USB connection. The LEDs generate pattern-less IR light and the cameras generate almost 200 frames per second of reflected data. This is then sent through a USB cable to the host computer, where it is analyzed by the Leap Motion software using "complex maths" in a way that has not been disclosed by the company, in some way synthesizing 3D position data by comparing the 2D grayscale stereo images generated by the two cameras, separated into the left and right cameras.
Next, the tracking layer matches the data to extract tracking information such as fingers and tools. It can track the movement of both hands and all 10 fingers with up to 1/100th millimeter accuracy [10] and no visible latency.

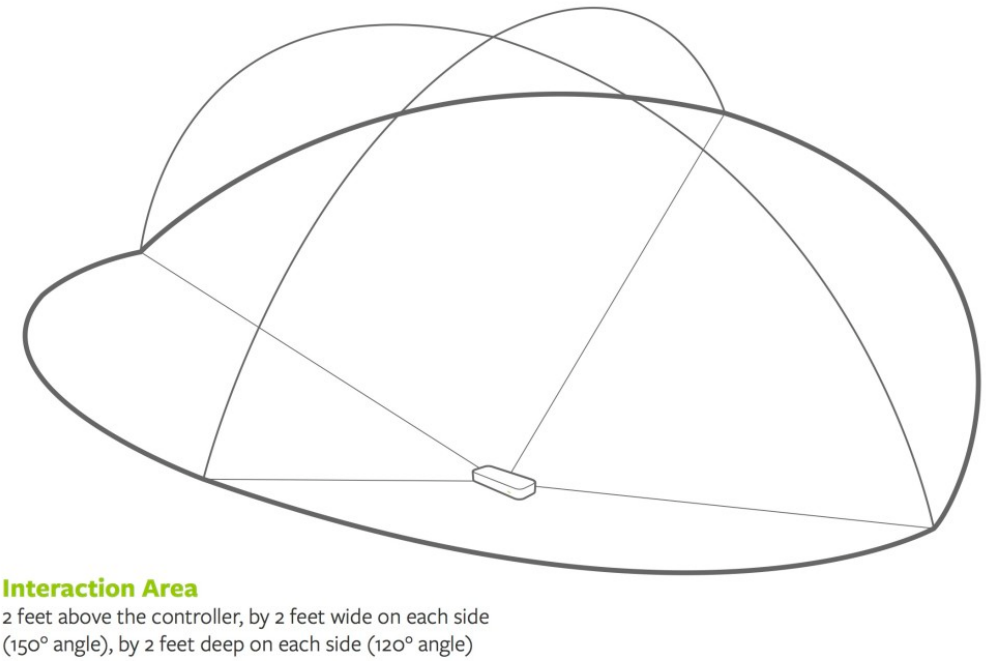

Figure 3.2– Interaction area of Leap Motion device. [11]

Next will be displayed some advantages and disadvantages of Leap Motion controller that I think I have found.

Advantages of Leap Motion controller

– **Inexpensive.** As for other tracking devise out there Leap Motion costs arround 80 $ without

shippment and texes;

- **Extremly precise motion control**.It can track the movement of both hands and all 10 fingers with up to 1/100th millimeter accuracy [10] and no visible latency;
- **Easy to set up**. To make it work, Leap Motion only needs the software to be installed on the machine and to be properlly plugged in an USB port;
- **It works with Windows and Mac and any size display**;
- **It is the most accurate sensor that is current available on the market**;
- **It's size is very small**.

Now let's see what are the disadvantages of Leap Motion.

Disadvantages of Leap Motion controller

- **Range of detection is limited**. After the last update of Leap Motion Orion, the range of detection got from 60 cm to 80 cm. However sometimes is not enough;
- **Limited as a practical controller**. Is designed only to detect hands and finger tracking;
- **Sensors does not always register hand position properly**;
- **Limited support**. There is still no support for other platforms such as Mobile, Linux;
- **Facing upwards**. You have to put it facing upwards and interact in the abover region which might get a little awkward. People this days prefer sensors to be placed in front e.q. in the place of a webcam.

Why I did choose to use Leap Motion over Microsoft Kinekt?

Eventhough I have a Microsoft Kinekt also, I choosed to use Leap Motion because is a total new devise which may bring new possibilities of solving problems. It is rather simple to use and install and the price is much much less than to a Microsoft Kinekt. The documentation on how to integrate it in Unity is more. It is more accurate sensor than Kinekt and it's size is very small which is an advantage. I thought that users, patients, hospitals, rehabilitation centers will appriciate it's advantages over Microsoft Kinekt.

3.2.3 C# over JavaScript scripting

Firstly will be talked about JavaScript. In Unity, JavaScript is not the same as writing it for browsers(which is why it is popularly nicknamed UnityScript in the Unity's community). UnityScript has so many shortcuts that are just invitations to make mistakes, sometimes programmer will lose the time just tracking bugs or mistakes. One of the biggest, most awful thing is implicit variable declaration. Following example in

```
1
2 var myCounter;
3
4 void Update ()
5     {
6    if (mycounter > 0)
7       {
8           // Does something
9       }
10    }
```

Listing 3 – Implicit variable declaration in UnityScript.

Can you spot the error in there?. The variable of misspelled from myCounter to mycounter in Update, the compiler will create a new variable with a default value to 0, so if statement will never happen, despite of what value myCounter get's in the editor.

There won't be warnings from the compiler, since implicit declaration is legal. This first reason is a must use C# case.

The number 2 reasons is Visual Studio. As from 2016 Visual Studio now support Unity . With Intellisense and inline help, the API of Unity will be more easy to learn as you type. There is simply no better editor and debugger on the market, at least for .Net development.

And lastly, with C# the developer has access to .Net meachanisms that have not been ported to UnityScript, which is Mono as a script host. While one can argue about merits of C# as a language, Mono's base class library offers a wealth of functions. Collections, I/O, multithreading, and insanely expressive LINQ all speed up development considerably.

3.2.4 LitJSON

JSON is a simple, yet powerful notation to specify data. It defines simple scalar types such as boolean, number (integers and reals) and string, and a couple of data structures: arrays (lists) and objects (dictionaries). For more information on the JSON format, visit JSON.org.

LitJSON is written in C#, and it's intended to be small, fast and easy to use. It was developed on a GNU/Linux environment, using the Mono framework.

In order to consume data in JSON format inside .Net programs, the natural approach that comes to mind is to use JSON text to populate a new instance of a particular class; either a custom one, built to match the structure of the input JSON text, or a more general one which acts as a dictionary.

Conversely, in order to build new JSON strings from data stored in objects, a simple export operation sounds like a good idea.

For this purpose, LitJSON includes the JsonMapper class, which provides two main methods used to do JSON-to-object and object-to-JSON conversions. These methods are JsonMapper.ToObject Listing 4 and JsonMapper.ToJson.

```
1 using LitJson;
2 using System;
```

```
3
4  public class JsonSample
5  {
6      public static void Main()
7      {
8          string json = @"
9            {
10             ""album"" : {
11               ""name""   : ""The Dark Side of the Moon"",
12               ""artist"" : ""Pink Floyd"",
13               ""year""   : 1973,
14               ""tracks"" : [
15                 ""Speak To Me"",
16                 ""Breathe"",
17                 ""On The Run""
18               ]
19             }
20           }
21         ";
22
23         LoadAlbumData(json);
24     }
25
26     public static void LoadAlbumData(string json_text)
27     {
28         Console.WriteLine("Reading data from the following JSON string: {0}",
29                           json_text);
30
31         JsonData data = JsonMapper.ToObject(json_text);
32
33         // Dictionaries are accessed like a hash-table
34         Console.WriteLine("Album's name: {0}", data["album"]["name"]);
35
36         // Scalar elements stored in a JsonData instance can be cast to
37         // their natural types
38         string artist = (string) data["album"]["artist"];
39         int    year   = (int) data["album"]["year"];
40
41         Console.WriteLine("Recorded by {0} in {1}", artist, year);
42
43         // Arrays are accessed like regular lists as well
44         Console.WriteLine("First track: {0}", data["album"]["tracks"][0]);
45     }
46 }
```

Listing 4 – LitJson usage example in C# [12].

In order to use LitJSON is needed to download the .dll file from their github page, add it to your project and then include the following code in every C# script file, as shown in Listing 5

```
using LitJson;
```

Listing 5 – Using LitJSON as a json reader for your project.

For Kyno it was used only JsonMapper.ToObject which helped it to easily read text from a text file. See for how it was implemented in subsection **Implementation**.

3.2.5 Application Programming Interface

For the purpose of getting tracked data from the Leap Motion it is of a great use to use the API, which is a set of tools, routes, protocols used in order to build an applicaiton.

An API offers SaaS. This is a great benefit and opportunity for the programmers because they do not have to start programming from scratch every time they need to create a new software. APIs are very useful and they will be used in Kyno in order to get frame, hand, fingers tracking data from leapmotion and reading text from a file without having the programmers to write code from scratch, but focusing their attention on code that does what the APIs or framework can not do.

An API is basically a method of software to speak with another software. So everytime the programmer wants to access a set of data he has to call the API. But the amount of data that can be accessed by him is limited so he has to communicate to the API in a very specific language.

In figure 3.3 is visualized the concept of API as an middleman between a programmer and application. The one that accepts request and informs programmers about everything is middleman. However if the requested is allowed the middleman returns the data.

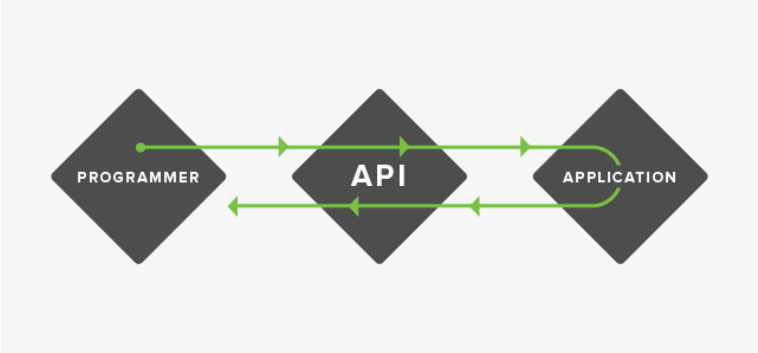

Figure 3.3 – Communication between programmer and application through API. [14]

For the implementation of having tracked data and then using it there are used the API of Leap Motion device.

3.2.6 Leap Motion API

At the very bottom level, the leap motion API returns the tracking data in the form of frames. Each Frame object contains lists of tracked entities, such as hands, fingers and tools, as well as objects representing recognized gestures and factors describing the overall motion of hands in the scene.

Motions are continuous hand movements – estimates of how the position of tracked objects (hands, fingers, and tools) change over time. These consist of translation, rotation, and scale.

Comparing any two frames containing the same hand allows you to compute the change in motion through time.

The API is able to provide a wide range of additional tracking data, including left vs. right hands, tracking confidence values, as well as grab and pinch strength. Finger tracking is now persistent (so that each hand always has five fingers), digit types are identified (thumb, index, middle, ring, and pinky), and individual bones and joints are tracked.

In figure 3.4 is shown how the tracked entity in the Leap Motion is placed withing a hierarchy that starts with the hand.

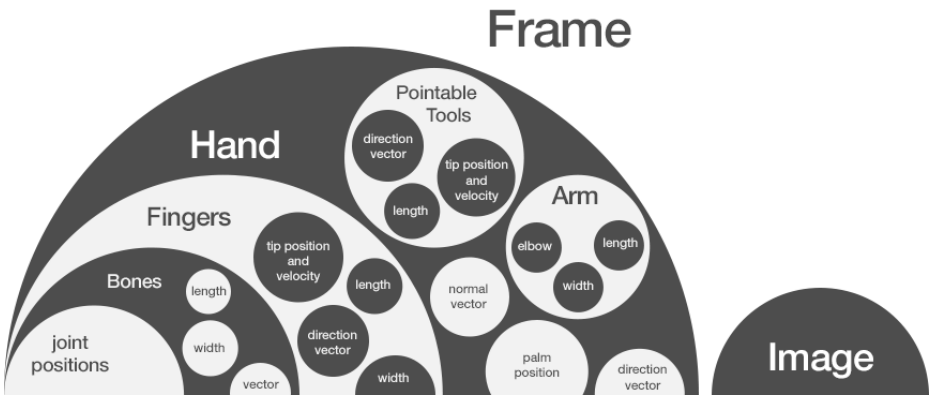

Figure 3.4– Communication between programmer and application through API. [15]

A hand object includes:

- palm position and velocity;
- direction and normal vectors;
- orthonormal basis.
- **Fingers**.
  a) tip position and velocity;
  b) direction vector;
  c) orhonormal basis;
  d) length and width.
- **Pointable Tools**
  a) tip position and velocity;
  b) direction vector;
  c) length and width.

And other components that are shown in figure 9.

In order to understand how it works this API we need to check the main components it provides to us:

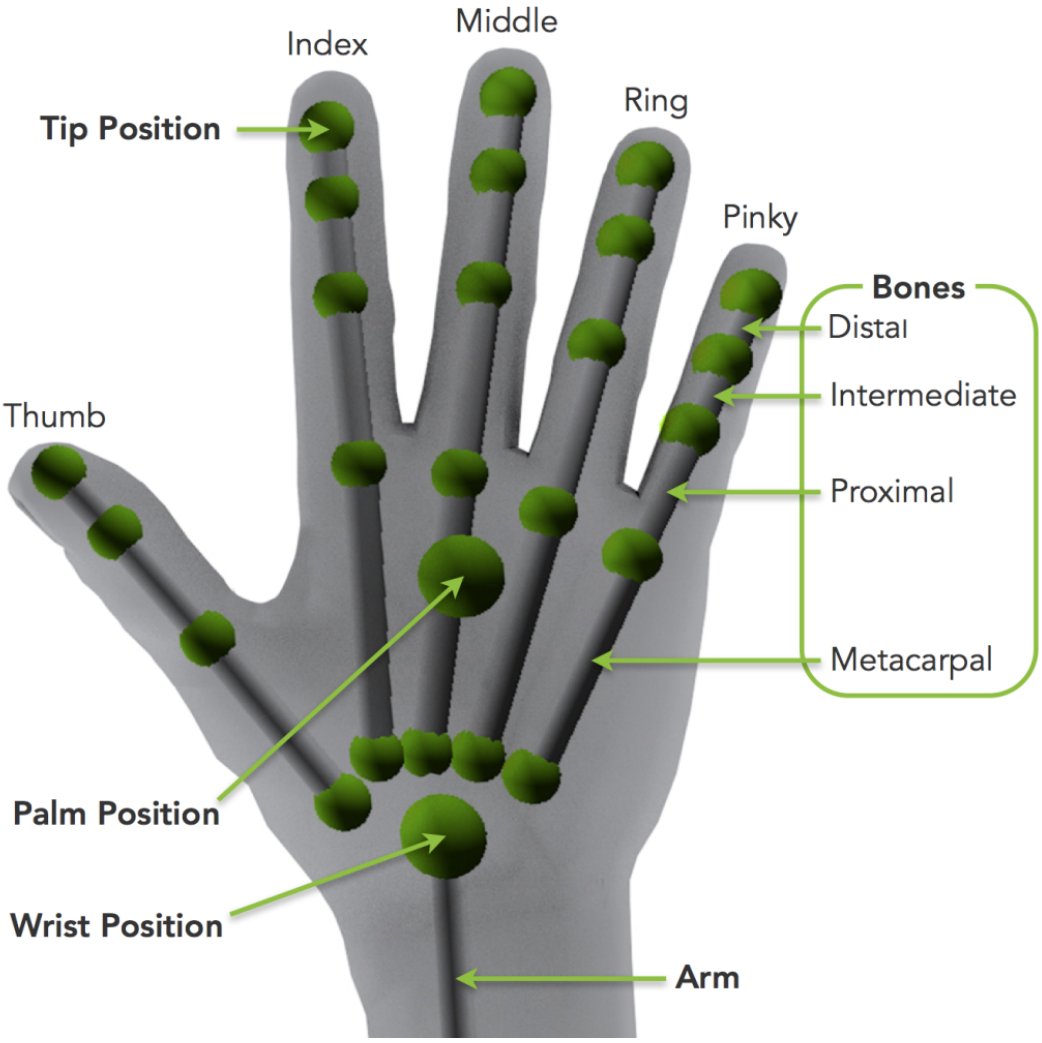

Figure 3.5– Components of Hand object in Leap Motion. [15]

– **Connecting to the controller**. Connecting to the Leap Motion controller.

To connect to the Leap Motion device, is needed to be created a Controller object Listing 6 . The Controller object establishes a connection automatically to the Leap Motion daemon which then passes tracking data in the form of Frame object to the application.

```
1 Controller controller = new Controller ();
```

Listing 6– Controller object creation in C#

Using controller object can be used to get information about the state of the connection and connected hardware and to set connection options to the application.

– **Frames**. Getting tracking data from the API.

The Leap Motion API presents motion tracking data as a series of snapshots called frames. Each frame of tracking data contains the measured positions and other information about each entity detected in the snapshot.

Each Frame object contains snapshot of the scene recorded by the Leap Motion controller. Hands, fingers are the basic physical entities tracked.

A Frame contains tracked data from a connected Controller Listing 7object.

```
if (controller.IsConnected) { //controller is a Controller object
    Frame lastFrame = controller.Frame ();
    Frame previousFrame = controller.Frame (1);
}
```

Listing 7– Accesing last and previous frame from a controller object.

Then getting data from a frame as the basic objects tracked by the Leap Motion system is shown in Listing 8:

```

Frame frame = controller.Frame ();
List<Hand> hands = frame.Hands;
```

Listing 8– Getting data from a frame

The returned object by the Frame object are all read-only. They can be stored and used later in the future.

– **Hands**. Tracking Hands. Hands are the main entity tracked by the Leap Motion controller. The controller maintains an inner model of the human hand and validates the data from its sensors against this model. This allows the controller to track finger positions even when a finger is not completely visible. Note that it is possible for movement or changes in position to be lost when a finger is behind or directly in front of the hand (from the point of view of the controller). The Leap Motion software matches the internal model against the existing data. In some cases, the software can make an incorrect match – for example, identifying a right hand as a left hand.

  The Hand class represents a physical hand detected by the Leap. A Hand object provides access to lists of its pointables as well as attributes describing the hand position, orientation, and movement.

  Get Hand objects from a Frame:

```
Frame frame = controller.Frame ();
if(frame.Hands.Count > 0){
    List<Hand> hands = frame.Hands;
    Hand firstHand = hands [0];
}
```

Listing 9– Getting Hand from a Frame

– **Fingers**. Tracking Fingers. Fingers are represented by Pointable objects. In addition, a separate Finger class specializes the Pointable class to provide specific finger information. The fingers can be get as a list or using an ID obtained in the previous frame. The next example is shown how to get the fingers by list:

```
List<Finger> fingers = hand.Fingers;
```

Listing 10 – Getting fingers from a Hand by list

– **Coordinate Systems**. Converting from Leap Motion to application coordinates. A fundamental task when using the Leap Motion controller in an application is mapping the coordinate values received from the controller to the appropriate application-defined coordinate system.

The Leap Motion Controller provides coordinates in units of real world millimeters within the Leap Motion frame of reference. That is, if a finger tip's position is given as (x, y, z) = [100, 100, -100], those numbers are millimeters – or, x = +10cm, y = 10cm, z = -10cm.

The Leap Controller hardware itself is the center of this frame of reference. The origin is located at the top, center of the hardware. That is if you touch the middle of the Leap Motion controller (and were able to get data) the coordinates of your finger tip would be [0, 0, 0].

In its normal position, that is on a desk with the user on one side and the computer monitor on the other, the user is "in front" (+z) of the controller and the monitor screen is "behind"(-z) the controller figure 3.6.

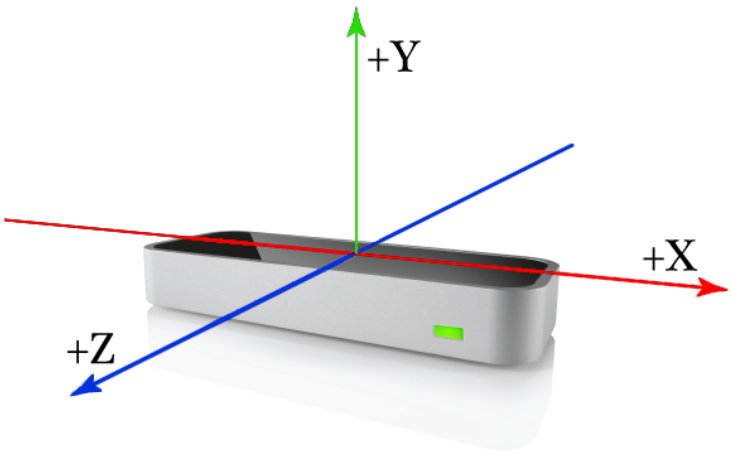

Figure 3.6 – The Leap Motion controller uses a right-handed coordinate system. [16]

The Unity environment uses the left-hand rule, meaning the z axis points away from the user when the x axis is to the right and y is up. So, it needs to multiply Leap Motion coordinates by -1 Listing 11 for z axis to be map correctly in Unity3D.

```
Leap.Vector leapToWorld(Leap.Vector leapPoint, InteractionBox iBox)
{
    leapPoint.z *= -1.0f; //right-hand to left-hand rule
    Leap.Vector normalized = iBox.NormalizePoint(leapPoint, false);
    normalized += new Leap.Vector(0.5f, 0f, 0.5f); //recenter origin
    return normalized * 100.0f; //scale
}
```

Listing 11 – Mapping right hand to left hand rule.

3.3 System implementation

3.3.1 Software license

Kyno will be offered for a period of 1 month for free. In this period, the software will be tested and feedback will be prompted from users. It is very important at this stage to see how the application is doing, where are the problems, what features need to be added.

After 1 month of free using, a monthly or yearly subscription plan will be applied with different prices for individual patients or hospitals and rehabilitation centers.

3.3.2 Software backend

In order for the Leap Motion to work with the application it needs to be created a controller object Listing 6. The Controller object establishes a connection automatically to the Leap Motion daemon which then passes tracking data in the form of Frame object to the application Listing 13.

```
protected virtual void Start() {
      createController();
      _untransformedUpdateFrame = new Frame();
    }
        protected void createController() {
          if (leap_controller_ != null) {
            destroyController();
          }
          leap_controller_ = new Controller();
        }
            protected void destroyController() {
              if (leap_controller_ != null) {
                if (leap_controller_.IsConnected) {
                  leap_controller_.ClearPolicy(Controller.PolicyFlag.
  POLICY_OPTIMIZE_HMD);
                }
                leap_controller_.StopConnection();
                leap_controller_ = null;
              }
            }
```

Listing 12 – Controller connection and frame setting

Now in the following listing it is extracted a transform matrix containing translation, rotation, and scale from a Unity Transform object and returns a Leap Motion LeapTransform object. Using this matrix to transform Leap Motion tracking data to the Unity world relative to the specified transform.

In addition to applying the translation, rotation, and scale from the Transform object, the returned transformation changes the coordinate system from right- to left-handed and converts units from millimeters to meters by scaling. It returns A Leap.LeapTransform object representing the specified transform from Leap Motion into Unity space.

```
protected void updateIfTransformMoved(Frame source, ref Frame toUpdate) {
      if (transform.hasChanged) {
        _transformedUpdateFrame = null;
        transform.hasChanged = false;
      }

      if (toUpdate == null) {
        toUpdate = source.TransformedCopy(transform.GetLeapMatrix());
      }
    }
```

Listing 13– Leap Motion tracking matrix data.

Next in Listing 14 is how to update the graphical part of hand representation by taking the current frame each time the update function is executed. And then passing it to a hand factory which contains the current hands displayed on the screen and from there it will set the updating data each time a frame is generated.

```
  protected LeapProvider provider;
  protected HandFactory factory;

    protected virtual void Start() {
        provider = requireComponent<LeapProvider>();
        factory = requireComponent<HandFactory>();
      }

       protected virtual void Update() {
            Frame frame = provider.CurrentFrame;

            if (frame != null && graphicsEnabled) {
              UpdateHandRepresentations(graphicsReps, ModelType.Graphics, frame);
            }
          }
```

Listing 14– Updating hand representations

From here using the factory object, each hand's data such as position, scale, rotation of the hand can be get and mapped to a virtual hand in Unity.

A lot of back magic was made by Leap Motion they provided a package for Unity where there are scripts and prefabs ready to be used in development and these prefabs will provide the developer with a set of virtual hand. As a developer, it is more comfortable to focus more on the practical part of the application by implementing the gestures it was planned to have in the application.

Pinch

Leap Motion's API provide pinching only with the index finger. In order to get a pinch with other finger it need to be calculated the distance between Thumb and rest of the fingers. For that in Listing 15 at line 19 is provided a functiont that will return the distance between 2 fingers or better say tip position of the fingers. So every time GetDistanceToThumb will be called it will calculate that distance. By default the API returns the distance in mm so it had somehow to convert it to a better form for the purpose of later usage in the code.

```
protected void FingerTips(Hand h)
    {
      Dictionary<string, Vector3> fingerInfo = new Dictionary<string, Vector3>();
      List<Finger> fingers = h.Fingers;

      foreach (Finger finger in fingers)
      {
        fingerInfo.Add(finger.Type.ToString(), finger.TipPosition.ToVector3());
      }


      var toIndexDist = GetDistanceToThumb(fingerInfo, 0, 1);
      var toMiddleDist = GetDistanceToThumb(fingerInfo, 0, 2);
      var toRingDist = GetDistanceToThumb(fingerInfo, 0, 3);
      var toPinkyDist = GetDistanceToThumb(fingerInfo, 0, 4);

}
 protected float GetDistanceToThumb(Dictionary<string, Vector3> fingers, int
   firstPosition, int secondPosition)
    {
      float distance = (float)Math.Round(Vector3.Distance(fingers.ElementAt(
   firstPosition).Value, fingers.ElementAt(secondPosition).Value), 4) * 10.0f;
      return distance;
    }
```

Listing 15– Pinching detection in Kyno

After getting the distance from Thumb to other tip of the fingers it can be checked if there would be a pinch or not.

Grab

A finger is considered extended if it is extended straight from the hand as if pointing. A finger is not extended when it is bent down and curled towards the palm.

So here it just needs to be checked if all of the fingers are not extended and if not that it will be a grab otherwise it won't be.

```
void Update()
    {
      if (canGrab)
      {
        Debug.Log("Can Grab");
        ensureGrabInfo();
      }
    }
 protected virtual void ensureGrabInfo()
    {
      Hand hand = _handModel.GetLeapHand();
      var fingers = hand.Fingers;
      grabDetection(fingers);
```

```
14      }
15
16       protected virtual void grabDetection(List<Finger> fingers)
17          {
18            if (!fingers.Any(o => o.IsExtended))
19            {
20              //StartCoroutine(DoTheDance());
21              if(canGrow)
22                SetCounter();
23              Debug.Log("Its a full finger grab!!!");
24            }
25            else
26            {
27              Debug.Log("Its not a full finger grab");
28              canGrow = true;
29            }
30          }
```

Listing 16 – Grab detection in Kyno

Rotate

Rotation is quite tricky but not possible to detect tanks to Leap Motion. Using normal vector to the palm.

If your hand is flat, this vector will point downward, or "out" of the front surface of your palm.

The direction is expressed as a unit vector pointing in the same direction as the palm normal (that is, a vector orthogonal to the palm).

It can be used the palms normal vector to compute the roll angle of the palm with respect to the horizontal plane as it shown in line 12 of the following code snippet:

```
1  void Update()
2      {
3        if (canRoll)
4        {
5          ensureRollInfo();
6        }
7      }
8
9      protected virtual void ensureRollInfo()
10     {
11       Hand hand = _handModel.GetLeapHand();
12       float roll = -hand.PalmNormal.Roll;
13       float rollDegrees = ToDegrees(roll);
14       Debug.Log(rollDegrees);
15       if((rollDegrees > 85 || rollDegrees < -85))
16       {
17         if(canGrow)
18           SetCounter();
19       }
20
```

```
      else
      {
        canGrow = true;
      }
    }

    protected float ToDegrees(float Radian)
    {
      float Degrees;
      Degrees = Radian * 180 / Mathf.PI;
      return Degrees;
    }
```

Listing 17– Rotation detection in Kyno

The angle on which the hand was rotated will be returned in radians so in order to transform it in angle ToDegree method was implemented.

Rotate, grab and pinch is executed in Update function which means that it will be called once per frame.

3.3.3 UI of Kyno application

In this section is described how the UI in Kyno was implemented, from the pie menu that is at the bottom left corner of the screen to the tips pannel on the right side of the screen. In Listing 18 is shown how the text file is loaded from an json file and saved to an object of JsonData type. Than by with the help of readJson variable the data that was saved in the itemData variable will be accessed from another script. Also Awake function is called at the moment when the application is started so the reading will be made at this stage.

```
using LitJson;

  private string jsonString;
  public JsonData itemData;
  public static ReadJson readjson;
  void Awake()
  {
    TextAsset text = Resources.Load("Items") as TextAsset;
    jsonString = text.text;
     itemData = JsonMapper.ToObject(jsonString);
  }
  public JsonData GetItem(string value, string type)
  {
    for (int i = 0; i < itemData[type].Count; i++)
    {
      if (itemData[type][i]["value"].ToString() == value)
        return itemData[type][i];
    }
    return null;
  }
```

```
21
```

Listing 18 – Reading JSON from a file.

One way to not overpopulate the screen with different graphics that will give distraction to the user is to close the panels/menu as much as possible. One solution is shown in Listing 19 where the panel with the tips for user is displayed/hidden at a button press.

```
1   void Start()
2  {
3    oldPosition = transform.position;
4  }
5
6  // Update is called once per frame
7  void Update ()
8  {
9     if (moved)
10      PositionChanging(step);
11    else
12      PositionChanging(0);
13   }
14   //position change in relation to the new distance
15   public void PositionChanging(float distance)
16   {
17     if (!firstMove) {
18       Vector3 newPosition = new Vector3(oldPosition.x + distance, oldPosition.y,
19                                         oldPosition.z);
20       transform.position = Vector3.Lerp(transform.position, newPosition, Time.
    deltaTime * smooth);
21     }
22   }
23   //toggle the moved boolean
24   public void Open()
25  {
26    moved = true;
27  }
28
29   public void Close()
30   {
31     moved = false;
32   }
33
```

Listing 19 – Moving tips panel

Next is described in the Listing 20 is described how the menu is placed in a circular way. For a better vizualizing experience at the application launch the user is prompt with a welcome screen. During that time Move function must wait some seconds in order to start, for this is used coroutines which will delay the running of Wait with required time.

```
    private IEnumerator Wait(float smoothScale, float smoothingPos, float angle
)
    {
      yield return new WaitForSeconds(intro.waitingTime  + 2);
      StartCoroutine(Move(smoothingScale, smoothingPos, angle));
    }

```

Listing 20 – Coroutines

Coroutines are computer program components that generalize subroutines for nonpreemptive multitasking, by allowing multiple entry points for suspending and resuming execution at certain locations. Coroutines are well-suited for implementing more familiar program components such as cooperative tasks, exceptions, event loop, iterators, infinite lists and pipes.

In Listing 20 is described the itself action on generating a pie menu. First of all in order to place some graphics in a circular path we need to find the points of that circular path. For this the parapetric equation of a circle come in hand. A circle can be defined as the locus of all points that satisfy the equations $x = rcos(t)$ and $y = rsin(t)$

where x,y are the coordinates of any point on the circle, r is the radius of the circle and t is the parameter - the angle subtended by the point at the circle's center. Hence the pie menu will be displayed only on 1/4 of a full circle the angle will be only $\pi/2$

```
  private IEnumerator Move(float smoothScale, float smoothingPos, float a)
   {
     while (true) {
       for (int i = 0; i < numPoints; i++)
       {
         //multiply 'i' by '1.0f' to ensure the result is a fraction
         float pointNum = (i * 1.0f) / numPoints;
         //angle along the unit circle for placing points
         float angle = pointNum * a;

         float x = Mathf.Sin(angle) * radiusX;
         float y = Mathf.Cos(angle) * radiusY;

         //position for the point prefab
         if (vertical)
           pointPos = new Vector3(x, y) + centerPos;
         else if (!vertical)
         {
           pointPos = new Vector3(x, 0, y) + centerPos;
         }
         //place the prefab at given position
         axis = pieButton[i].GetComponent<RectTransform>().localPosition -
  pointPos;
        axis.Normalize();
        pieButton[i].GetComponent<RectTransform>().localPosition = Vector3.Lerp
  (pieButton[i].GetComponent<RectTransform>().localPosition, pointPos * axis,
```

```
    smoothingPos * Time.deltaTime);
25          pieButton[i].GetComponent<RectTransform>().localScale = Vector3.Slerp(
    pieButton[i].GetComponent<RectTransform>().localScale, Vector3.one,
    smoothingScale * Time.deltaTime);
26        }
27        yield return null;
28      }
29    }
30
```

Listing 21– Creation of a Pie Menu

Now the placing on the path is rather simple. In order to display evenly the graphics on line 9 it is calculated a new angle for each of the graphic where it will be placed. Then again for each graphic on line 11 and 12 it will be calculted 2 float variables with info of the position of the current graphic. On line 16 or 19 it will be created a new Vector3 with coordinates of the point on the circle calculated earlier. Where on the 24 line the current graphic component will take this new Vector and it will Lerp through current position and the new position. In result the following figure 3.7 is displayed to the user

Figure 3.7– Panel with tips for user.

The next Listing 22 is recieving information about the action it was made in the application and then it spreads the required data to the right text and image game objects of the application. With the help of the j parameter that is of ReadJson at Awake time it get's the ReadJson script component. With this parameter it will be possible to be accessed the JSON object that was saved when the JSON file was readen.

By specifying in the Unity editor which button it was pressed through accesing the parameter on the pushedButton function it can then be easily set the corresponding data as it can be seen in SetInfo function.

```
1
2   private static ReadJson j;
3
4   // Update is called once per frame
5   void Update () {
6     Message();
7   }
8   void Awake()
9   {
```

```
    j = this.GetComponent<ReadJson>();
    textReference.text = "";
  }
  private void SetInfo()
  {
    switch (pressedButton)
    {
      case "Grab":
        selectedText = j.GetItem(pressedButton, "tips")["text"].ToString();
        leftImageName = j.GetItem(pressedButton, "tips")["img1"].ToString();
        rightImageName = j.GetItem(pressedButton, "tips")["img2"].ToString();
        break;

      case "Roll":
        selectedText = j.GetItem(pressedButton, "tips")["text"].ToString();
        leftImageName = j.GetItem(pressedButton, "tips")["img1"].ToString();
        rightImageName = j.GetItem(pressedButton, "tips")["img2"].ToString();
        break;
      case "Pinch":
        selectedText = j.GetItem(pressedButton, "tips")["text"].ToString();
        leftImageName = j.GetItem(pressedButton, "tips")["img1"].ToString();
        rightImageName = j.GetItem(pressedButton, "tips")["img2"].ToString();
        break;
      default:
        Debug.Log("Default case");
        break;
    }

    ChangeInfo(selectedText, leftImageName, rightImageName);
  }
  private void ChangeInfo(string s, string image1, string image2)
  {
    textReference.text = s;
    leftImage.sprite = Resources.Load<Sprite>("Images/" + image1);
    rightImage.sprite = Resources.Load<Sprite>("Images/" + image2);

  }
  public void PushedButton(string s)
  {
    pressedButton = s;
    SetInfo();
  }
```

Listing 22– Data popularization

Figure 3.8 is after data popularization when the user pressed the pinch button. This panel will provide user with some tips and images on how to do an exercise.

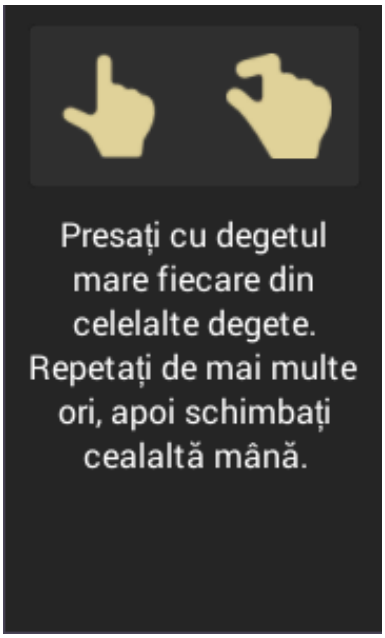

Figure 3.8– In application pie menu.

4 Economic Analysis

4.1 Project description

Kyno project is a treatment application for patients who have suffered from physical injury or illness on hands. This application will be used to improve a person's endurance, mobility and strength in hand. The rehabilitation techniques used by kinetoteraphies are often prescribed to help individuals enhance their overall physical conditioning. A patient may see a kinetotherapist after receiving a prescription from a physician, physician assistant or nurse practitioner. Kinetotherapists primarily work in public and private hospitals, sports medicine facilities, rehabilitation centers and academic institutions, as well as in private practice and as consultants. Which puts them as my main marketing targets. The success of this application will dramatically increase if we cross the countries borders, since there private hospitals, sports medicine have more money to invest and are willing to have better system of pacient treatment.

There are multiple solutions which provide kinetoteraphy application, but there is no other similar product in Moldova. The main advantage is that it's simple to use, it has a nice UX and UI, it gives feedback in a pleasant way and it has a system that tells you if the exercise was done well. It's perfect for everybody as soon as it has the required tools. That's why it is expected to be a promising product, with other possibilities which are going to be implemented in future.

4.2 SWOT analysis

It is necessary to make an analysis of strong and weak points for the given application, in order to have a brief overview about expectations or about possible problems that can appear. In 4.1 it is represented the strategic planning method, called SWOT, used to evaluate Strength, Weaknesses, Opportunities and Threads involved in the project.

After elaboration of SWOT Analysis, it was taken in consideration the objective of the business venture of project and there were identified the internal and external factors that are favorable and unfavorable to achieve the goal. There will always be concurrency, this factor having an important role in market development and increase of systems' quality.

4.3 Project time schedule

For the accomplishment of a project it is necessary to establish a schedule. For the development of the Kyno application, agile project management is applied to offer flexible and iterative method of designing the application. It goes in 5 stages: planning, research, development, testing and deployment. The process flows in repetitive and incremental way.

4.3.1 Objective determination

The main objective of the following project is to provide a complete and functioning application for it's users. Otherwise without a finished product there is no profit. More to that, it is important to market the application and get exposed to a large audience in need. This can be done by targeting

Table 4.1 – SWOT analysis

Strengths	**Weaknesses**
– it is a new product on the market – easy to use – price, value, quality	– You need LeapMotion devise in order to use it – client application available only on Windows platform – lack of funding – location and geography
Opportunities	**Threads**
– it save time and money to the client – extendable to more regions – outsourced labor for development – not yet mature – time to market	– won't be bought by hospitals – similar application can be developed, so the popularity of this system may decrease – integration with existing systems – technical challenges

first private hospitals. Since it is not a common piece of software, it creates a very specific audience of users.

To keep up with the latest trends and researches, it is also an essential objective to keep updated and provide enhancements to the software. The lifecycle of the application will require bugfixes, interface changes, feature implementations. All of that will help the system still be trendy and up-to-date on the market.

4.3.2 Time schedule establishment

As it was said above the project will iterate over 5 steps: planning, research, development, testing and deployment. Naturally as most of the IT projects, it is subdivided into smaller parts. Planning step isn't supposed take up a lot of time, since the requirements are flexible. Moreover due to the research part the design solutions can change over time and open up new perspectives. The process of development is being split up in smaller tasks that can be accomplished within a 2-5 day period. Total duration of the project is computed using (4.1).

$$D_T = D_F - D_S + T_R, \tag{4.1}$$

where D_T is the duration, D_F – the finish date, D_S – the start date and T_R – reserve time. In table 4.2 is presented the first iteration of the project schedule. It uses the following notations: PM – project manager, SM – sales manager, D – developer/designer, T – Tester.

Table 4.2 shows the activities that will occur during project development, who is involved into

Table 4.2– Time schedule

Nr	Activity Name	Duration (days)	People involved	Notes
1	Define the project concept and objectives	10	PM, SM, D	
2	Perform market analysis	10	PM, SM	Market analasys document
3	Analysis of the domain	10	D	Research of algorithms and technologies
4	Requirements and specifications	5	PM, D	Write them down
5	System design	10	PM, D	UML
6	Preprocessing and learning part of the implementation	25	PM, D	
7	End-user application development	30	PM, T, D, SM	This includes UX and UI design
8	Validation of results	5	PM, T, D, SM	
9	Documentation	5	D	
10	Building and testing the entire project	15	PM, T, D	Real users for testing
11	Active marketing	15	SM	OM on SM and private hospitals
12	Total time	140		

each process and how much time does it take to accomplish a task. Total amount of time spent on this project is 140 days or 20 weeks, which means almost 5 months for a strong beta version. For each individual, it is indicated below the number of spent days:

- PM: 110 days;
- SM: 70 days;
- D: 115 days;
- T: 50 days

4.4 Economic motivation

The following section describes the evaluation of the project from the economic point of view. That includes the total profit, number of potential clients, salaries that have to be paid to employees, revenues that the company gets by commercializing the product. All the costs and prices are given in MDL (Moldavian lei) currency. Tangible and intangible assets, indirect expenses will also be taken into account. Wear and depression in regard to final product will also be computed.The entire economical part is done on the presumption that the software will have payed licenses. Either way it is a curios approach to compute all the necessary resources and indexes for developing a project. It opens managerial insights over entrepreneurial ideas.

4.4.1 Tangible and intangible asset expenses

The budget for the required tangible and intangible assets is shown in Table 4.3, Table 4.4. Direct expenses are presented in Table 4.5.

Table 4.3– Tangible asset expenses

Material	**Specification**	**Measurement unit**	**Price per unit (MDL)**	**Quantity**	**Sum (MDL)**
Mac Book pro	retina display i7	Unit	25000	2	50000
Apple Display	27 inch	Unit	20000	2	20000
Asus laptop	K55VD, i5	Unit	5000	1	5000
Leap Motion	hand controller 5	Unit	1600	2	3200
				Total	78200

The total amount of expenses in MDL is about this much.

$$T_e = 78200 + 18200 = 96400 \tag{4.2}$$

4.4.2 Salary expenses

This section is concerned about the salaries to employees and various funds that should be paid. The distribution of salaries per day is the following: project manager - 500MDL, tester - 350 MDL, sales manager - 400 MDL, developer - 480 MDL.

Now by having computed all the salaries for the employees, it is time to compute how much to be paid to social services fund, medical insurance fund and the total work expenses by summing up all previous expenses.

Salary expenses are introduces in Table 4.6.

This year the social service fund is approved to be 23%, therefore the salary expenses are

Table 4.4 – Intangible asset expenses

Material	**Specification**	**Measurement unit**	**Price per unit (MDL)**	**Quantity**	**Sum (MDL)**
Unity Pro	Subscription	Unit	1500	1	1500
VS Professional 2015	License	Unit	10000	1	10000
Enterprise Architect	Home	Unit	1900	1	1900
Windows 10	License	Unit	2400	1	2400
MS Word 2016	License	Unit	1400	1	1400
Adobe Illustrator	Subscription	Unit	1000	1	1000
				Total	18200

Table 4.5 – Direct expenses

Material	**Specification**	**Measurement unit**	**Price per unit (MDL)**	**Quantity**	**Sum (MDL)**
Whiteboard	Universal Dry Erase Board	Unit	400	1	400
Post-it note	Stickers	Unit	20	10	200
Paper	A4	500 sheets	60	1	60
Marker	Whiteboard marker	Unit	15	10	150
Pen	Blue pen	Unit	5	20	100
				Total	910

Table 4.6– Salary expenses

Employee	**Work fund (days)**	**Salary per day (MDL)**	**Salary fund (MDL)**
Project Manager	110	500	55000
Tester	50	350	17500
Sales Manager	70	400	28000
Developer	115	480	55200
		Total	155700

computed according to the relation (4.3).

$$\begin{aligned} FS &= F_{re} \cdot T_{fs} \\ &= 155700 \cdot 0.23 \\ &= 35811 \end{aligned} \tag{4.3}$$

where FS is the salary expense, F_{re} is the salary expense fund and T_{fs} is the social service tax approved each year. The medical insurance fund is computed as

$$\begin{aligned} MI &= F_{re} \cdot T_{mi} \\ &= 155700 \cdot 0.045 \\ &= 7006.5 \end{aligned} \tag{4.4}$$

where T_{mi} is the mandatory medical insurance tax approved each year by law of medical insurance and this year it is 4.5%.

So now having computed social service tax and medical insurance tax, it is possible to compute total work expense fund as follows

$$\begin{aligned} WEF &= F_{re} + FS + MI \\ &= 155700 + 35811 + 7007 \\ &= 198518 \end{aligned} \tag{4.5}$$

where WEF is the work expense fund, FS is the social fund and MI is the medical insurance fund. In that way the total work expense fund was computed.

4.5 Individual person salary

Along with total work expense fund, it is necessary to compute the annual salary for the project manager. Considering that the project manager has a salary of 500 MDL per day and there

are approximately 256 working days in the year, so the gross salary that the project manager get is

$$GS = 400 \cdot 256 = 102400 \tag{4.6}$$

where GS is the gross salary computed in MDL.

Social fund tax this year represents 6%, so the amount that should be tax paid in MDL represents

$$SF = 102400 \cdot 0.06 = 6144 \tag{4.7}$$

Medical insurance tax represents 4.5% and gives the following result

$$MIF = 102400 \cdot 0.045 = 4608 \tag{4.8}$$

In order to proceed with income tax computations, it is necessary to calculate the amount of taxed salary.

$$\begin{aligned} TS &= GS - SF - MIF - PE \\ &= 102400 - 6144 - 4608 - 10128 \\ &= 81520 \end{aligned} \tag{4.9}$$

where TS is the taxed salary, GS – gross salary, SF – social fund, PE – personal exemption, which this year is approved to be 10128.

The last but not the least thing to be computed is the total income tax, which is 7% for income under 29640 MDL and 18% for income over 29640 MDL.

$$\begin{aligned} IT &= TS - ST \\ &= 29640 \cdot 0.07 + 51880 \cdot 0.18 \\ &= 2074.8 + 9338.4 = 11413.2 \end{aligned} \tag{4.10}$$

where IT is the income tax, TS – the taxed salary and ST – the salary tax. With all this now it is possible to find out what's going to be the net income.

$$\begin{aligned} NS &= GS - IT - SF - MIF \\ &= 102400 - 11413.2 - 6144 - 4608 \\ &= 80.234.8 \end{aligned} \tag{4.11}$$

where NS is the net salary, GS – gross salary, IT – income tax, SF – social fund, MIF – medical insurance fund.

4.5.1 Indirect expenses

The project is having 140 full working days, one working day has 6 h of work and the total amount of h is 840. Laptop consumes 60W/hour, a bulb light consumes arround 100W/hour and the IMac screen consumes from 54.1 W in idle state to 86W at 50 % brightness and 145W at 100

% brigthness, the value is somewhere in the middle, because the IMacs where not always during the developemt of the project working. Now the total power consumed withing 140 days is calculated bellow.

$$\begin{aligned} PU &= 3 \cdot L \cdot 840h + 2 \cdot S \cdot 840h + 3 * B \cdot 840h \\ &= 151200W + 134400W + 252000W \\ &= 537,6kW \end{aligned} \tag{4.12}$$

where PU is total power usage, L – laptop power usage, S – external monitors power usage, B – bulbs power usage.
The indirect expenses are things like electricity, Internet traffic, water, etc. Those will be presented in Table 4.7.

Table 4.7– Indirect expenses

Material	**Specification**	**Measurement unit**	**Price per unit (MDL)**	**Quantity**	**Sum (MDL)**
Internet	Moldtelecom	Pack	200.00	5	1000
Transport	Trip	Units	2.00	150	300
Electricity	Union Fenosa	KWh	2.16	537,6	1161.2
				Total	2461.3

4.5.2 Wear and depreciation

Another important part of economic analysis is the computation of wear and depreciation. It is a well known fact that any product decreases its value with time. Depression will be computed uniformly for the whole project duration, so that there are no accountancy issues. In other words, if a product is planned for 3 years, it should be divided into 3 uniform parts according to each year.

Straight line depreciation will be applied. Normally wear is computed regarding to the type of asset. The notebook and single-board computer are usable for a period of 3 years. Licenses will last for a single year. At first tangible and intangible assets are summed up and then the salvage costs

of each of the items at the end of their period of use has to be subtracted:

$$\begin{aligned} TAV &= \sum n(AC - SV) \\ &= 2 * (25000 - 1000) + 2 * (20000 - 1000) + (5000 - 1000) + 2 * (1600 - 1000) \\ &+ (1500 - 1000) + (10000 - 1000) + (1900 - 1000) + (2400 - 1000) \\ &+ (1400 - 1000) + (1000 - 1000) \\ &= 102400 \end{aligned} \tag{4.13}$$

where TAV is the total assets value, AC – assets cost, SV – salvage value, n – number of items. In order to get the yearly wear, divide total asset value by the period of use of assets, being 3 years.

$$\begin{aligned} W_y &= TAV/T_{use} \\ &= 102400/3 \\ &= 34133 \end{aligned} \tag{4.14}$$

where W_y is the wear per year, TAV – total assets value, T_{use} – period of use. Relation (4.14) included tangible assets which will last for 3 years and intangible assets which last only one year. The initial value of assets in MDL was

$$\begin{aligned} W &= W_y/D_y \cdot T_p \\ &= 34133/365 \cdot 140 \\ &= 13092 \end{aligned} \tag{4.15}$$

4.5.3 Product cost

With all the project expenses computed, it is easy to compute the product cost which includes the cost used to create this product. 4.8.

Table 4.8– Total Product Cost

Expense type	**Sum (MDL)**	**Percentage (%)**
Indirect expenses	2461.3	1.15
Direct expenses	910	0.42
Salary expenses	198518	92.4
Asset wear expenses	13092	6.03
Total product cost	**214981.3**	**100**

4.5.4 Economic indicators and results

At this point it is crucial to sell the product to clients from mediacal sphere. The total product cost is very high, consequently there are 2 strategies that can be applied – whether sell less with a high price or sell more with a lower price. It is not possible to add a percentage to the product cost that will represent the profit. It is assumed that the expected profit represents 20% of the total product cost and the expected number of sold copies to be 300.

$$\begin{aligned} GP &= C_{total}/N_{cs} + P_p \\ &= 214981.3/300 + (214981.3/300) \cdot 0.2 \\ &= 859.9 \end{aligned} \tag{4.16}$$

where GP is the gross price, C_{total} – total product cost, N_{cs} – number of copies sold, P_p – chosen profit percentage. This is not the price of the end product, since it is necessary to add sales tax (VAT), which represents 20% and is added to the gross price.

$$\begin{aligned} P_{sale} &= GP + TX_{sales} \\ &= 859.9 + 859.9 \cdot 0.2 \\ &= 1031.88 \end{aligned} \tag{4.17}$$

where P_{sale} is the sale prices including VAT, GP – gross price, TX_{sales} – sales tax. The net income is computed by multiplying gross price and the number of expected copies to be sold, which will be

$$\begin{aligned} I_{net} &= GP \cdot N_{cs} \\ &= 1031.88 \cdot 300 \\ &= 309564 \end{aligned} \tag{4.18}$$

where I_{net} is the net income, GP – gross price, N_{cs} – number of copies sold. Moreover it is necessary to compute the gross and net profit. The indicators are GPr – gross profit and NPr – net profit.

$$\begin{aligned} GPr &= I_{net} - C_{production} \\ &= 309564 - 214981 \\ &= 94583 \\ NPr &= GPr - 12\% \\ &= 94583 - 94583 \cdot 0.12 \\ &= 83233.04 \end{aligned} \tag{4.19}$$

where I_{net} is the net income, $C_{production}$ – cost of production. The profitability indicators are C_{profit}

– cost profitability, S_{profit} – sales profitability computed in MDL.

$$\begin{aligned} C_{profit} &= GPr/C_{production} \\ &= 94583/214981 \\ &= 0.44 \\ S_{profit} &= GPr/I_{net} \\ &= 94583/309564 \\ &= 0.3 \end{aligned} \tag{4.20}$$

4.6 Marketing Plan

Concept of Marketing derived from the word market. Marketing - economical activities that guide flow of goods and services from producer to consumer. Marketing is a system of economical activities about price setting, promotion and distribution of products and services to satisfy current and potential consumers requests. Marketing is the science and art of exploring, creating, and delivering value to satisfy the needs of a target market at a profit.

Functions of Marketing:

- Analyzing of external environment;
- Analyzing consumers behavior;
- Development of product;
- Development of distribution;
- Development of promotion;
- Price setting;
- Social responsibility;
- Management marketing.

This application will be spread between private/public hospitals and people at home. To make people use a new application is not so easy because it needs time and investment to make it popular and well known.However the application will be easy to use so that an ordinary application user will be able to intuitively use it.

Market research stages:

- Identifying the problem;
- Developing program of research and gathering; information
- Establishing specific information (internal, external);

– Establishing methods for collecting data;

– Performance of research;

– Information analysis, drawing conclusions.

Introduction stage This stage of the cycle could be the most expensive for a company launching a new product. The size of the market for the product is small, although they will be increasing. On the other hand, the cost of things like research and development, consumer testing, and the marketing needed to launch the product can be very high, especially if it's a competitive sector.

Strategy - Screaming, massive penetration The growth stage is typically characterized by a strong growth in sales and profits, and because the company can start to benefit from economies of scale in production, the profit margins, as well as the overall amount of profit, will increase. This makes it possible for businesses to invest more money in the promotional activity to maximize the potential of this growth stage.

Maturity Stage - During the maturity stage, the product is established and the aim for the manufacturer is now to maintain the market share they have built up. This is probably the most competitive time for most products and businesses need to invest wisely in any marketing they undertake. They also need to consider any product modifications or improvements to the production process which might give them a competitive advantage.

Declining stage - the market for a product will start to shrink, and this is what's known as the decline stage. This shrinkage could be due to the market becoming saturated (i.e. all the customers who will buy the product have already purchased it), or because the consumers are switching to a different type of product or even a new/better product.

4.7 Economic conclusions

Kyno project was analyzed from the economic point of view. It was computed the production cost, different profit and profitability indicators, various types of expenses involved, including direct, indirect, salary and taxes. The whole analysis is worth to understand if the product will be successful and if it's worth investing money in it. The biggest expense represents the intellectual equity, since it is critical to have a reliable product, which is based on extensive research and professional development techniques. The price of the application can become a blocker, therefore it's price might be dropped. In such scenario other means of profit can exist.

The commercialization of the product is not an easy task. High-quality service and customer support can be provided only to institutions and users that bought the product. The success of the product highly depends on financial strategy and solid economic analysis, which was presented in this chapter.

Conclusions

Kyno project is the result of the thesis work. It is an application that gives to the user exercises for kinetotherapy rehabilitation. The uniqueness of the application is that the targeted users and the proposed solution is unusual for the moldavian market. There are still no tools available that does a similar thing on the local level. However, developing a software that doesn't yet exist on a market offers a lot of freedom, but at the same time is problematic because the needs of the potential customers are yet blurry. A joint work with a the Physiotherapist representative would have resulted into a more specific and useful product.

There are other rehab systems out there that use motion capture, but they often require sensor gloves or other proprietary hardware that take a lot of training and supervision to use, or they depend on rigging an entire room with expensive cameras or placing lots of sensors on the body. Thanks to Leap Motion, Kyno doesn't require any extra hardware, cameras, or body sensors, which keeps the price affordable,. That low price point is extremely important.

Kyno is working to remove all the major barriers to physical rehabilitation by making a system that is fun, simple to use, and affordable. Kyno demonstrates the potential of NUI to make technology simpler and more effective—and the ability of Leap Motion technology to help high tech meet essential human needs.

Building the application required many multi-step planning of the infrastructure. The first problem encountered with choosing what gestures to be detected in the app. Due to the Leap Motion device tracking limitation some of the exercises can not be included because Leap Motion is not able to "see through the fingers" - for example, when one finger covers the other. Fingers right next to each other also pose a problem for the cameras and might not be recognized individually. That's why some exercises cannot be done.

During first stage of marketing it is really important to collect feedback about the application from the clients and then releasing new versions with the implemented feedback. Also if the marketing want's to cross the border and wants to grow on international levels, the more functionalities should be implemented in the sistem.

One of the functionality to be implemented is a system that is monitoring the progress of patients from anywhere in the world. Patients can perform complex rehabilitation programs using entertaining therapies either in a Rehabilitation clinic as well as in their own homes.

Each session will be registered using Microsoft® Azure, a cloud-based platform that will allow patients to complete therapy sessions under the supervision of their specialist, whether at a medical centre or at home.

An important part for the future development of Kyno is to add mini artificial inteligent component that will take the user through an amazing experience on how to use the application, how to put the hands over Leap Motion controller, how to perform certain exercises. So basically the component will tell the user to press one button so the user will have to do that, then it will say to the user to put his hands over Leap Motion and again user will have to act as the component shows. It will be runned every time the application was launched for the first time on a system after that it will just ask the user if he want to go through a tutorial.

Another future implementation is adding more exercises, which means detecting more gestures, then gamify them to create a better user experience. Gamified content makes users to establish a special excited relation between them and the application.

As a generalization, working on this project was an extreme enjoyable challenge for me. Since the goal was to write a full documentation of it in the form of a report, to devide into time periods the process of development and work constanly on the application.

References

1 What is a stroke, http://www.stroke.org.uk/what-stroke/what-stroke

2 Sandra Frances Basset, *The assessment of patient adherence to physiotherapy rehabilitation*, http://www.physiotherapy.org.nz/assets/Professional-dev/Journal/2003-July/July03commentary.pdf

3 Althea Group *Top 10 cause of death among poor, developing, and developed countries*, http://www.altheadistributor.com/philippines-heart-stroke-cancer

4 Liber TV *În Republica Moldova, numărul pacienților cu accident vascular cerebral a ajuns la 70000*, http://www.libertv.md

5 Unified Modeling Language http://www.tutorialspoint.com/uml/

6 Dennis Ziesecke *Review Leap Motion Motion Control Technology*http://www.notebookcheck.net/Review-Leap-Motion-Motion-Control-Technology.98821.0.html

7 Build once deploy everywhere http://www.unity3d.com/unity/multiplatform

8 Detecting mouse button down documentation http://www.docs.unity3d.com/ScriptReference/Input.GetMouseButtonDown.html

9 Unity Technologies *Multiple touch input* http://www.unity3d.com/learn/tutorials/topics/mobile-touch/multi-touch-input

10 Leap Motion launches world's most accurate 3D motion control technology for computing. http://www.leapmotion.com/news/leap-motion-launches-world-s-most-accurate-3-d-motion-control-\technology-for-computing

11 Interaction Area of Leap Motion Device http://www.blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/

12 LitJSON Quickstart Guide http://www.lbv.github.io/litjson/docs/quickstart.html

13 Constraint-Induced Movement Therapy http://www.strokeassociation.org/STROKEORG/LifeAfterStroke/RegainingIndependence/PhysicalChallenges/Constraint-Induced-Movement-Therapy_UCM_309798_Article.jsp#.V1STCJF97Dc

14 Michael Patterson *What is and API, and Why Does It Matter?* http://www.sproutsocial.com/insights/what-is-an-api/

15 Alan Davis *Getting Started with the Leap Motion SDK* http://www.blog.leapmotion.com/getting-started-leap-motion-sdk/

16 Coordinate Systems http://www.developer.leapmotion.com/documentation/csharp/devguide/Leap_Coordinate_Mapping.html#map3d