Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea de Calculatoare, Informatica și Microelectronică
Filiera Anglofonă "Computer Science"

Admis la susținere
Șef de catedră: prof. dr. hab. Viorel Bostan

"___" _____ 2016

Utility application for hand Kinetotherapy

Proiect de licență

Student: _____(N.Barbaroș)
Conducator: _____(S.Vîrtosu)
Consultanți: _____(I.Tyrell)
_____(G.Covdii)
_____(V.Bostan)
_____(M.Balan)

Chisinau 2016

Abstract

Thesis **Utility application for hand Kinetotherapy** presented by Barbaroș Nicolae was written in English. It has 25 figures, 11 listings, 8 tables, and 15 references. The report consists of introduction, 4 chapters, and conclusions.

The thesis aims to research the potential of data journalism in Republic of Moldova. For that purpose, OpenMedia project was developed. It is a platform that aggregates online media channels for offering means of visualization of word frequency requested by a user.

The application was build using C-Sharp programming language. The platform consists from two components. The first one is the data gathering and preprocessing. It can be divided in four smaller parts. Crawling the media channels an fetching the article pages, parsing the articles and storing to database, executing natural language processing operations over articles and preparing data for client side visualization. The second part of the platform is the client side application. It aims to provide an interactive way for querying words and visualizing the data. The data representation is done using enhanced line plots that denote the word frequency for different media sources.

The technologies used are Sinatra framework for building the web interface, MongoDB for storing the data and running native mapreduce tasks, Sidekiq for launching asynchronous jobs and Racai SOAP service for NLP operations.

OpenMedia represents an attempt to offer a tool for data journalism targeted for Republic of Moldova. It does not have many features but it has the right infrastructure for future development. It also proved that a data journalism software has a lot of potential on inexistent local market.

Rezumat

Teza **Modele matematice şi metode de eficientizare a conversiei energiilor regenerabile în baza efectelor aero-hidrodinamice**, prezentată de către Viorel Bostan pentru conferirea gradului ştiinţific de doctor habilitat în tehnică, a fost elaborată la Universitatea Tehnică a Moldovei, Chişinău, este scrisă în limba română şi conţine 342 pagini, 90 de figuri, 38 tabele, şi 250 de titluri bibliografice. Structura tezei include: introducerea, 6 capitole, concluzii şi anexe. Anexele conţin 145 de pagini cu 52 de figuri şi 48 tabele.

Teza este consacrată studiului fenomenelor aero-hidrodinamice în rotoarele turbinelor eoliene (TE) şi microhidrocentralelor de flux (MCHF) de mică putere ($P < 20$ kW), cu aplicarea modelelor matematice de descriere a fizicii curgerii fluidelor şi a metodelor moderne de simulare numerică a turbulenţei din cadrul dinamicii fluidelor CFD.

Scopul lucrării constă în sporirea eficienţei conversiei şi a capacităţilor funcţionale ale turbinelor eoliene şi microhidrocentralelor de flux de mică putere.

Au fost identificate modelele şi metodele matematice moderne de descriere a curgerii turbulente a fluidului, specifică rotoarelor de mică putere, cu evidenţierea efectelor aero-hidrodinamice tranzitorii şi în vecinătatea palelor. A fost argumentate profilurile aero-hidrodinamice ale palelor eficiente din punct de vedere al randamentului conversiei energiei şi în baza lor au fost elaborate concepte originale de rotoare aero-hidrodinamice.

În baza modelelor CAD ale rotoarelor propuse: au fost efectuate simulări CFD complexe ale curgerii tranzitorii a fluidului prin rotoare şi în vecinătatea palelor, cu determinarea gradului de influenţă a parametrilor constructiv-cinematici asupra caracteristicilor de putere şi factorilor de performanţă aero-hidrodinamică a rotoarelor TE şi MHCF; a fost efectuată analiza fenomenului de curgere a fluidului în stratul limită şi identificate soluţii tehnice de control şi minimizare a impactului negativ al acestuia asupra eficienţei conversiei energiei.

În baza rezultatelor cercetărilor, au fost elaborate şi fabricate modele noi de TE şi MHCF pentru diverse aplicaţii, inclusiv conceptul TE cu rotor basculant şi orientare la direcţia curenţilor de aer cu windrose cu profil aerodinamic al palelor. Soluţiile tehnice elaborate au fost protejate cu 17 brevete de invenţie şi apreciate la saloanele internaţionale de inovaţii, cercetare şi transfer tehnologic cu 43 medalii de aur, 13 de argint şi 2 de bronz.

Cuvinte-cheie: modele matematice; simulare numerică CFD; strat limită; curgere turbulentă, rotor aero-hidrodinamic, turbină eoliană; microhidrocentrală.

Table of contents

List of Figures

Listings

Abbreviations

VR	Realitatea Virtuala
UX	Experienta Utilizatorului
UML	Unified Modeling Language
UX	User experience
UI	User interface
OM	Online Marketing
SM	Social Media

Introduction

Most programming languages, on first glance, "look like Greek" to the untrained eye, an amalgam of English and unusual syntax. Consider, for instance, the program below, written in a language called Java, see listing 1

```
class Hello
{
    public static void main(String [] args)
    {
        System.out.println("hello, world!");
    }
}
```

Listing 1– Java example, code is listed

0.1 What problem is being solved

All the program above does, when executed, is display "hello, world!" on the user's screen. You might have guessed as much just by looking over the code—and ignoring anything that didn't make sense! But what's with all the curly braces (and)? What's System.out? What does class Hello mean? And public static void main(String [] args)? Let's not even go there.

Suffice it to say that, when it comes to learning to program, there's quite a learning curve with languages like Java. Before you can begin to solve problems, you must first learn to read and write a new language, even if the task at hand is relatively simple (e.g., "hello, world!"). And whereas you might still understand a foreigner who mispronounces some English word, computers aren't so forgiving when it comes to mistakes. Leave out a semicolon, and the program above won't even work!

Learning to program is ultimately about learning to think logically and to approach problems methodically. The building blocks out of which a programmer constructs solutions, meanwhile, are relatively simple. Common in programming, for instance, are "loops" (whereby a program does something multiple times) and "conditions" (whereby a program only does something under certain circumstances. Also common are "variables" (so that a program, like a mathematician, can remember certain values).

For many students, the seemingly cryptic syntax of languages like Java tends to get in the way of mastery of such relatively simple constructs as these. There we turn our attention to the application I developed before we tackle a language like Java, a "new programming language" that lets you create your own animations and interactive art. This desktop application is just as useful (and fun) for budding computer scientists. By representing programs' building blocks with color-coded blocks (cubes and other 3d polygons), this desktop application "lowers the bar" to programming, allowing budding computer scientists to focus on problems rather than syntax, to master programmatic constructs rather than syntax. But, for now, we focus on programming itself. It just so happens that programming, for now, will be more like putting together a 3d puzzle construction in VR than writing Greek.

0.2 Why this problem is important

The term 'UX ' is used to refer to the approaches and methods employed to make sure that an application is entirely tailored and customized for its target market. If a platform does not appeal to a certain type of audience, it is likely to be quickly forgotten. That's why there should be a change in the way we are programming. The UX of my application will bring to the user changes in understanding the concept of programming and the world we live in by encouraging him to develope the following skills.

- Information Skill

  "By programming in this desktop application, users learn to select, create and manage different forms of media, animation and arts. This way they become more critical into analyzing the world they created around them."
- Communication Skills

  "Nowdays it's not all about just the ability to read and write text (code). This VR enages users in choosing, interacting with a large variety of media to express their creativity.
- Critical and System Thinking

  "As they learn and interact with the programmable blocks, user become commited in critical and system thinking. During process of creation, users need to coordinate the timing and interaction between the programmable moving objects and themselves."
- Creativity Curiosity

  "It encourages amazing creative thinking, an increasing skill in today's exponantially chaning world, in seeking different solutions to unexpected problems, not just learning how to solve a well given problem, but being always prepared to come up with new ideas, new solutions as new challengies occur."
- Interpersonal Skills

  "Because of the ability to program and literally interact through VR hands with programmable blocks, the "programming code" is more readable then other programming languages."

0.3 Other products on the market

Scratch [1] is an online program that kids as young as 5 years can use to express their online creations artistically while collaborating and sharing with other online users. Scratch has an intuitive drag-and-drop interface as shown in figure I.1 where the users will choose blocks of instructions to perform tasks.

Scratch is the work of MIT media lab and has a large number of Users creating and sharing their Projects online. Scratch is highly touted as the next generation tool to help kids prepare for the 21st century.

Various types of Projects can be created using Scratch like Stories, Animations, Games and even interaction with the Physical World

By creating projects in Scratch, kids will learn to think about designing stuff before building, will learn to solve problems creatively, express the ideas that are close to them and work collaboratively with others. Kids will also learn to share their work, present their work to others and get

feedback and also comment on others projects. These are some of the most important skill sets needed for the 21st Century. We turn our attention first to statements.

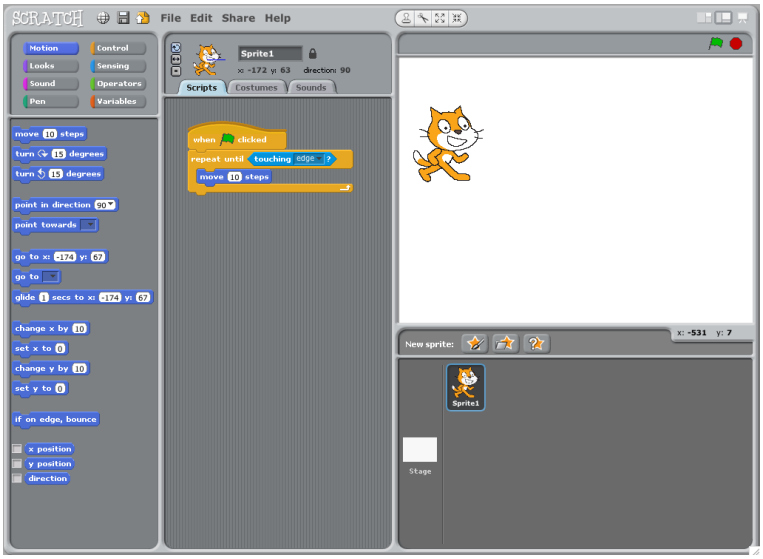

Figure I.1– The Scratch user interface

0.3.1 Advantages

This are some of the advatages of Scratch while a explored it.

– Simple to use.
– Suitable for children of all ages.
– Most if not all of the coding is done for you, so all you have to worry about is how to put blocks
– together.
– Cros-Curricular.
– Friendly user interface.

0.3.2 Disadvantages

This are some of the disadvantages of Scratch while a explored it.

– Hard to use at first.
– Simple sprites and graphics (you cannot use your own graphics).
– If there is a bug in the engine, unless it is open source you can't fix it.
– It's sprite based (only 2D)
– Not suitable experienced programmers

0.4 Short thesis description

Entering the application, the user will be in a wide room where he will be provided with a menu of programmable 3d objects such as cubes and other 3d primitives. With the help of Leap Motion he will be able to take those objects and to create blocks where after a push of a button the object that was programmed will behave accordingly to the block definition that user created

previously. The programmable objects will have functionalities such as variables, methods, loops, structures. The user will enhance his VR experience even more by wearing Oculus Rift. It is going to be written in C-sharp programming language using Unity Game Engine. Also for this I will need Oculus Rift SDK for VR and Leap Motion SDK for virtual hands. The first prototype should be able to do handle minimal functionality from the system like rotating an object, moving it.

How will this blocks be created? Through sticking different 3d objects together (like tetris game).

0.5 Chapters description

- System Analysis
  This approach breaks systems analysis into 5 [2] phases:
  Scope Definition: denoting an instrument for observing, viewing, or examining
  Problem analysis: analyzing the problem that arises
  Requirements analysis: determining the conditions that need to be met
  Logical design: looking at the logical relationship among the objects
  Decision analysis: making a final decision
- Implementation and Used Technologies
- Architecture of the System
- Economic Analysis

1 System analysis

1.1 Short thesis description

Entering the application, the user will be in a wide room where he will be provided with a menu of programmable 3d objects such as cubes and other 3d primitives. With the help of Leap Motion he will be able to take those objects and to create blocks where after a push of a button the object that was programmed will behave accordingly to the block that user created previously. The user will enhance his VR experience even more by wearing Oculus Rift. It is going to be written in C-sharp programming language using Unity Game Engine. Also for this I will need Oculus Rift SDK for VR and Leap Motion SDK for virtual hands.

2 Implementation and Used Technologies

3 Architecture of the System

3.1 UML Diagrams

The Unified Modeling Language (UML)[3] 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.

3.1.1 Use Case Diagram

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 3.1 , 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 3.2 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.

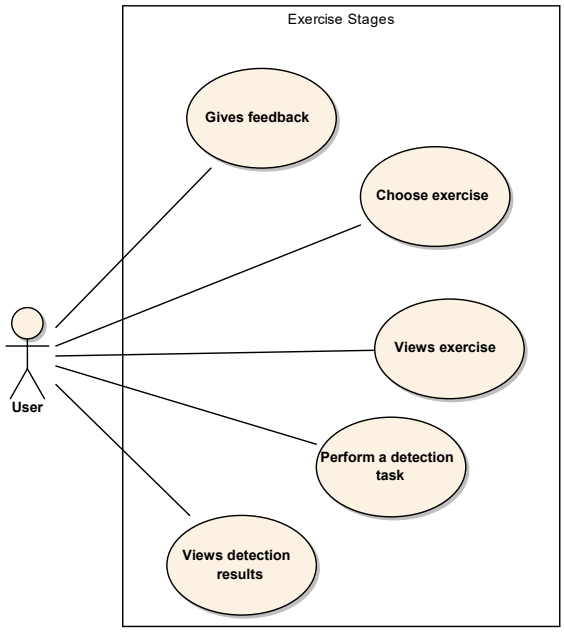

Figure 3.1– General Use Case diagram of the system.

Now back to the figure 3.2 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.

3.1.2 Deployment Diagram

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:

– Describe runtime processing nodes.
– Describe the hardware components used to deploy software components.
– Visualize hardware topology of a system.

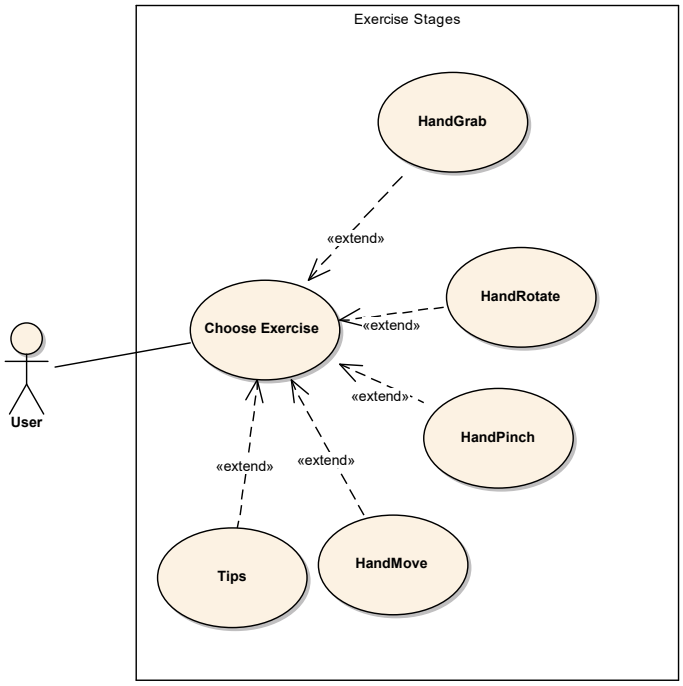

Figure 3.2– Use Case diagram of exercise action.

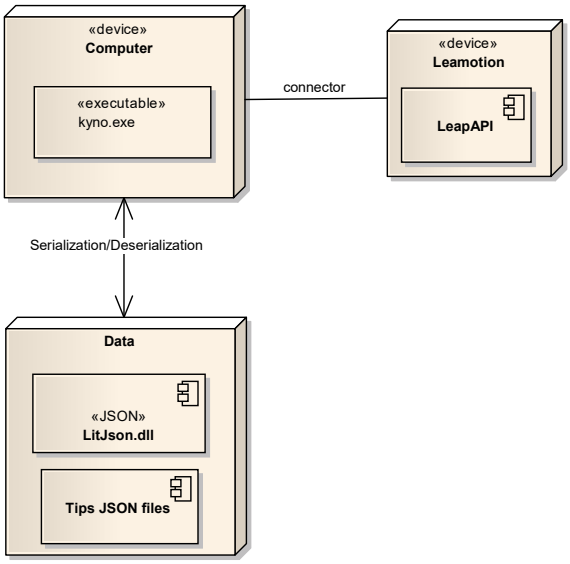

Figure 3.3– Deploy diagram of the system.

3.1.3 Class Diagram

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.
– Forward and reverse engineering.

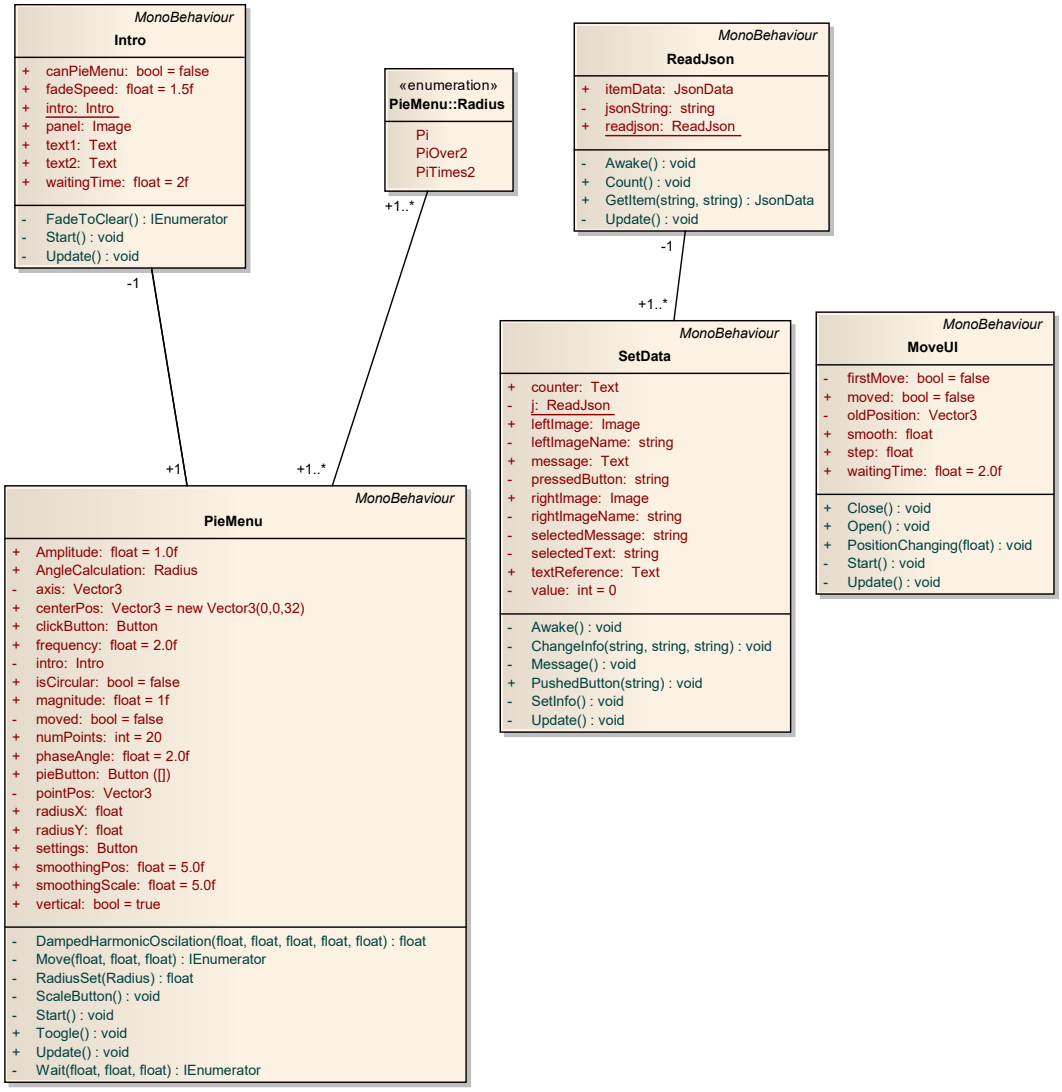

Figure 3.4 – Class Diagram of User Interface and User Interaction.

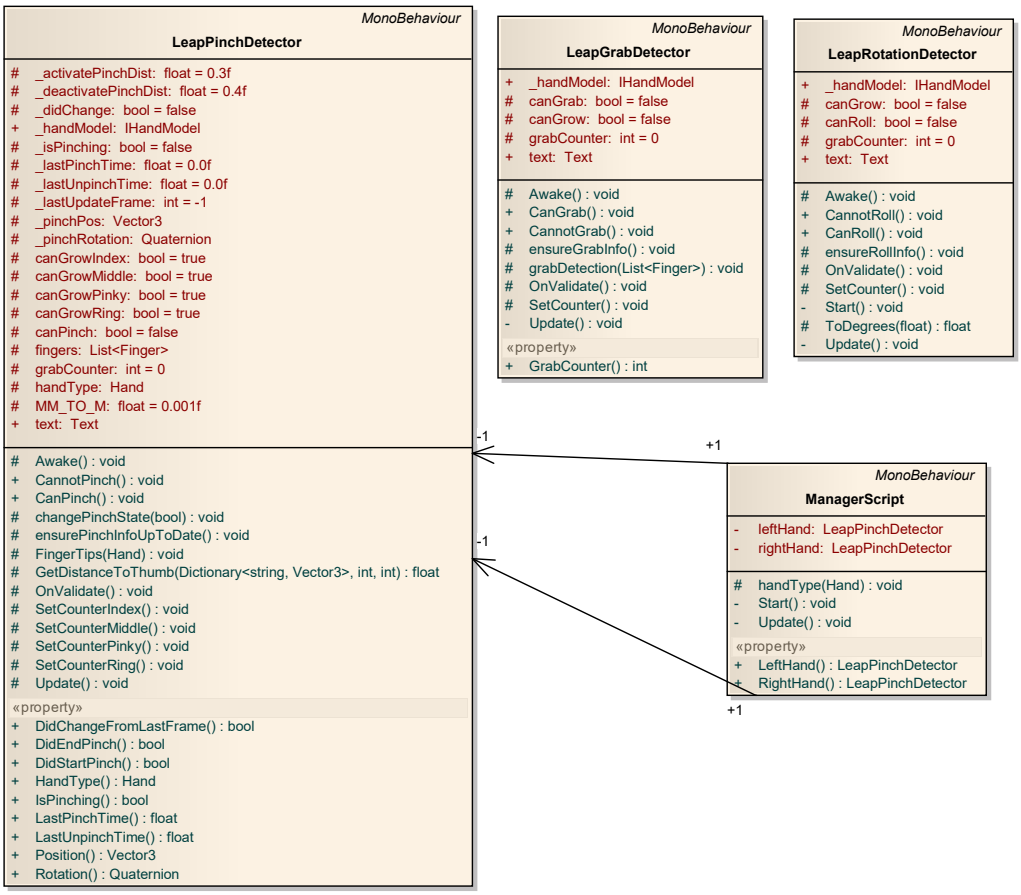

Figure 3.5– Class Diagram of Leap Motion gesture tracking.

3.1.4 Sequence Diagram

In the figure 3.6 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

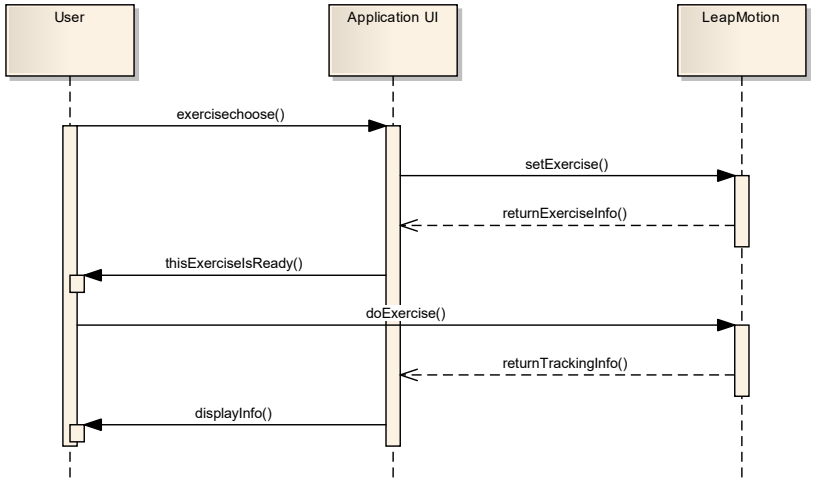

Figure 3.6– Sequence diagram of doing an exercise set.

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.

3.1.5 Activity Diagram

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 3.7, represents the list of actions done on the exercise part of the application. Every executed step in the chain depends on the previous one.

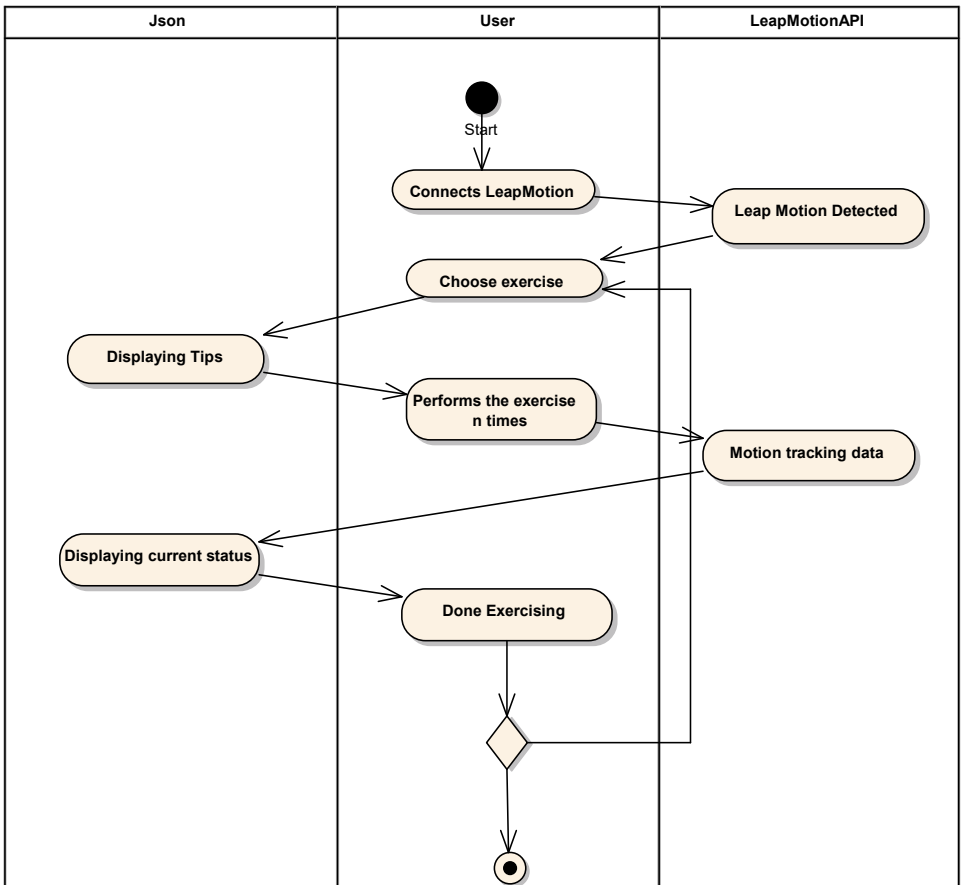

Figure 3.7– Exercise performing activity diagram.

3.1.6 State Diagram

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 3.8 and figure 3.9 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.

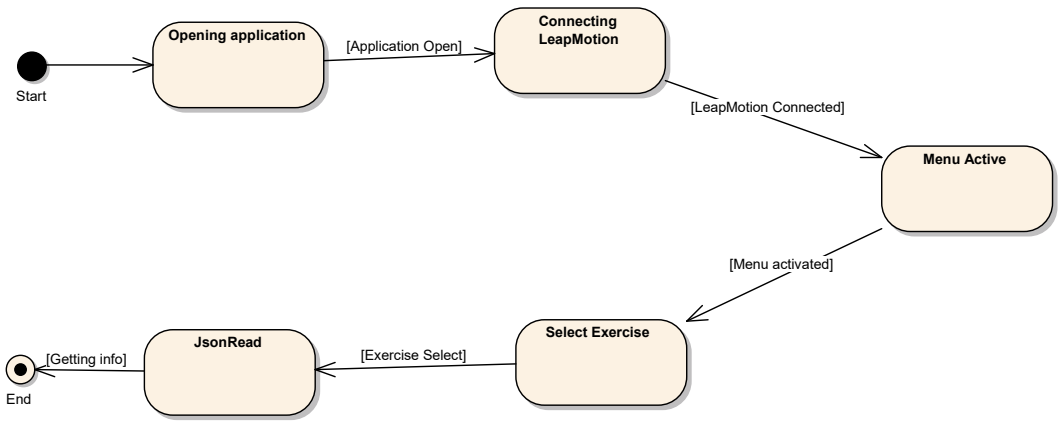

Figure 3.8– Application state diagram.

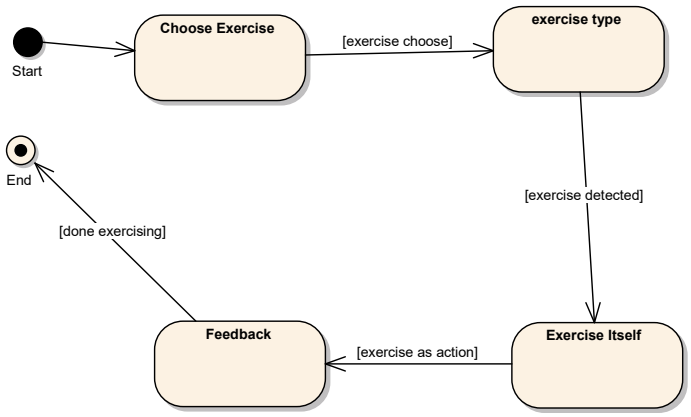

Figure 3.9– Exercising state diagram

3.2 User Experience

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.

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

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 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– 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		

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

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

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.

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

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 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, I went with 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.

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.

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**

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

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

This is the conclusions section to the thesis. It should contain at least 2-3 pages. You can end conclusions using a description of the future development of your product/work.

References

1 MIT https://scratch.mit.edu

2 Wikipedia http://en.wikipedia.org/wiki/Systems_analysis

3 Unified Modeling Language www.tutorialspoint.com/uml/