CMPE 491 – Physiomate Analysis Report

by

Hilal KÖKTÜRK
Ecem ERDOLU
Nur Bengisu KIRKDEVELİ
Deniz Merve GÜNDÜZ

Physiomate

Web site: physiomate.wordpress.com

06.11.2017
# Table of Contents

1. Introduction ......................................................................................................................... 1
2. Current System ...................................................................................................................... 1
3. Proposed System .................................................................................................................. 1
   3.1. Overview ......................................................................................................................... 2
   3.2. Functional Requirements ................................................................................................. 2
       3.2.1. User Class 1 – Patience ......................................................................................... 2
       3.2.2. User Class 2 – Doctor ......................................................................................... 4
       3.2.3. User Class – Administrator ............................................................................... 6
   3.3. Nonfunctional Requirements .......................................................................................... 7
       3.3.1. User Interface and Human Factors ...................................................................... 7
       3.3.2. Documentation ...................................................................................................... 8
       3.3.3. Hardware Considerations ..................................................................................... 8
       3.3.4. Performance Considerations .............................................................................. 8
       3.3.5. Error Handling and Extreme Conditions ............................................................. 8
       3.3.6. System Interfacing ............................................................................................... 8
       3.3.7. Quality Issues ....................................................................................................... 9
       3.3.8. System Modification ............................................................................................. 9
       3.3.9. Physical Environment .......................................................................................... 9
       3.3.10. Security Issues ..................................................................................................... 9
       3.3.11. Resources and Management Issues .................................................................. 10
   3.4. System Models ............................................................................................................... 10
       3.4.1. Scenarios ............................................................................................................... 10
       3.4.2. Use Case Model .................................................................................................... 12
       3.4.3. Object and Class Model ....................................................................................... 13
       3.4.4. Dynamic Models ................................................................................................ 15
       3.4.5. User Interface ....................................................................................................... 19
4. Glossary ............................................................................................................................... 21
Table of Figures

Figure 1: Use Case Model To Patient and Doctor ................................................................. 12
Figure 2: Object and Class Model (for database) ................................................................. 13
Figure 3: Object and Class Model (for game environment) ...................................................... 14
Figure 4: Sequence Diagram for Database ............................................................................ 15
Figure 5: Doctor Web Application Interface State Diagram .................................................... 16
Figure 6: Patient Game Application Interface State Diagram .................................................. 17
Figure 7: Patient Web Application Interface State Diagram ................................................... 18
Figure 8: Doctor Interface ...................................................................................................... 19
Figure 9: Patient Interface ..................................................................................................... 20
Figure 10: Game Interface ..................................................................................................... 20
1. Introduction

As mentioned before in our reports, we are developing a program for physiotherapy patients. We are planning to make our program like a game for the user to do those movements while having fun. We also planned to have a doctor interface which is a web application. Doctor will access everything about his/her patients and assign new exercises to patients. Patient will have a web-application too for accessing feedbacks from doctor and see the improvement of him/her.

Thanks to the improvements in the technology, we can supply a lot more to the patients in medical area. From the inspiration of these improvements we wanted to be one of those people whom supply patients a lot more. Some of our close relatives are having physiotherapy and not completing the exercises that are given from their doctors and this forced us to have the feeling for getting them better while making them not feeling like they are doing an obligation.

Right now, we have one game for our patient to do the exercise and we will explain a lot more about our project in the following parts of the report.

2. Current System

We have no current system for now in our project but we will develop it in future.

3. Proposed System

In Physio Mate project, there will be some users who are patients, doctors and admins. In the project, there will be a web system and also a game system. In web system, there will be 3 different interfaces which are for patients, doctors and admins. All users will use web system for log in the system but patients will also use the game system for doing physiotherapy movements.

There is some permission for these 3 groups of people in web system. Admins will use web system for adding/deleting doctors and movements. Doctors will use web system for adding/deleting patients, searching movements, adding/deleting movements to patient and giving feedback to patients. Patients will use web system for changing personal information, getting feedback from the doctor. And patients will use game system for doing the movements which are decided by the doctor.
3.1. Overview

Physio Mate system provides patients to do their physiotherapy movements in an entertaining way by using Kinect and see their improvements in terms of their physiotherapy. It also provides doctors to see their patients’ improvements easily. Patients will have movement cart and they will see the movements he/she should do and they will do their movements by the game environment and also all users will use web environment to see their information.

3.2. Functional Requirements

3.2.1. User Class 1 – Patience

3.2.1.1. Functional Requirement 1.1

ID: FR1

TITLE: Patience registrations.

DESC: In registration step, a patient needs to have an id which is given by doctor. After, patient enter the id he/she can create username and password.

3.2.1.2. Functional Requirement 1.2

ID: FR2

TITLE: Logon the system.

DESC: A patient should be able to log on the system to make their movement.

3.2.1.3. Functional Requirement 1.3

ID: FR3

TITLE: Logout the system

DESC: A patient should be able to log out the system after he/ she complete his/her movements.

3.2.1.4. Functional Requirement 1.4

ID: FR4
TITLE: Filling personal informations

DESC: After first login, a patient needs to fill his/her personal information to information form. This personal informations will be include his weight, his height, etc.

### 3.2.1.5.  
**Functional Requirement 1.5**

**ID: FR5**

TITLE: Updating informations

DESC: A patient should be able to update his / her personal informations. This informations will include username, password, weight, height, etc.

### 3.2.1.6.  
**Functional Requirement 1.6**

**ID: FR6**

TITLE: Doing movements

DESC: After a patient starts game application, patient should be able to do movements which are decided by doctor. When patient is doing his/ her movements should be able to feel like she is playing a game.

### 3.2.1.7.  
**Functional Requirement 1.7**

**ID: FR7**

TITLE: Accessing results

DESC: A patient should be able to access his/ her results about their movements from web application and game application. This results will be include time, success rate, etc.

### 3.2.1.8.  
**Functional Requirement 1.8**

**ID: FR8**

TITLE: Accessing feedbacks

DESC: A patient should be able to access his/ her feedback about their movement’s success from web application. This feedback will be given by doctor.
3.2.2. User Class 2 – Doctor

3.2.2.1. Functional Requirement 2.1

ID: FR8

TITLE: Logon the system.

DESC: A doctor needs to be able to logon the system with username and password. A system which will be used by doctor will be web application.

3.2.2.2. Functional Requirement 2.2

ID: FR9

TITLE: Updating informations.

DESC: A doctor needs to be able to update his/ her informations. These informations will include phone number, office address, username, password, etc.

3.2.2.3. Functional Requirement 2.3

ID: FR10

TITLE: Adding patients.

DESC: A doctor needs to be able to add a patient to the system with patient name, surname and id.

3.2.2.4. Functional Requirement 2.4

ID: FR11

TITLE: Searching patients.

DESC: A doctor needs to be able to search a patient who is using the system.

3.2.2.5. Functional Requirement 2.5

ID: FR12

TITLE: Deleting patients.

DESC: A doctor needs to be able to delete a patient from the system.
3.2.2.6. Functional Requirement 2.6

ID: FR13
TITLE: Listing patients.
DESC: A doctor needs to be able to list all patients in the system.

3.2.2.7. Functional Requirement 2.7

ID: FR14
TITLE: Searching movement.
DESC: A doctor needs to be able to search a movement which is contained by the system.

3.2.2.8. Functional Requirement 2.8

ID: FR15
TITLE: Adding movement to patient
DESC: A doctor needs to be able to add a movement to a patient movement list.

3.2.2.9. Functional Requirement 2.9

ID: FR16
TITLE: Deleting movement to patient
DESC: A doctor needs to be able to delete a movement to a patient movement list.

3.2.2.10. Functional Requirement 2.10

ID: FR17
TITLE: Analyzing patient process
DESC: A doctor needs to be able to analyze a patient process using patient data.
3.2.2.11. Functional Requirement 2.11

ID: FR18

TITLE: Giving feedback

DESC: A doctor needs to be able to give feedback to a patient.

3.2.2.12. Functional Requirement 2.12

ID: FR19

TITLE: Logout from the system.

DESC: A doctor needs to be able to logout from the system after he/she completed his/her job.

3.2.3. User Class – Administrator

3.2.3.1. Functional Requirement 3.1

ID: FR20

TITLE: Login from the system.

DESC: An admin should be able to login the system.

3.2.3.2. Functional Requirement 3.2

ID: FR21

TITLE: Adding doctor.

DESC: An admin should be able to add doctor to the system.

3.2.3.3. Functional Requirement 3.3

ID: FR22

TITLE: Deleting doctor.

DESC: An admin should be able to delete doctor from the system.
3.2.3.4. **Functional Requirement 3.4**

**ID: FR23**

**TITLE:** Adding movement

**DESC:** An admin should be able to add movement to the system.

3.2.3.5. **Functional Requirement 3.5**

**ID: FR24**

**TITLE:** Deleting movement

**DESC:** An admin should be able to delete movement from the system.

3.2.3.6. **Functional Requirement 3.6**

**ID: FR25**

**TITLE:** Logout from the system.

**DESC:** An admin should be able to logout from the system.

3.3. **Nonfunctional Requirements**

3.3.1. **User Interface and Human Factors**

We think that every patient and doctor should use our program without confusion of a complex interface, so we will create a basic and simple one for both. Since we have at least three interfaces (web application for patient and doctor, game application), we will consider this usability issue. We do not want to annoy users with many nested layouts because we consider that the patients may not be able to use an application or they may not use any kind of it before and the doctors are so busy people and they have limited time. Then, they will prefer simple applications to use.

On the other hand, if a user (patient or doctor) doesn’t know how to use our system, it will not be a useful and usable anymore. This is because the program cannot fulfill the user’s needs in these kinds of situations. At this point, our aim is creating a system that will be learned easily so the learning process will not force the users and will not waste their time by this way.
3.3.2. Documentation

Currently, there will be 3 documentations; one for doctors, one for patients and one for the engineers who interested in implementation part. We are planning to write a guideline for the doctors and patients for increase the usability of the program. Also, we will add some descriptions/explanations about how to write and read the classes to the coders beside every class.

3.3.3. Hardware Considerations

First of all, we are using Kinect as our camera so every person who wants to use our system, should have the Kinect and a computer/laptop that has Windows operating system. So, having the Kinect is the main hardware requirement for the project.

3.3.4. Performance Considerations

In our application, response time should be minimized since this will be a kind of interactive application. When a doctor sends a movement or want to contact with any patient, it should be done at the same time, not later on. When the patient completes his/her exercises, the system should give feedbacks rapidly. Similarly, if there is a problem about any issue (adding patient, giving id and password, forgetting password, changing password or other information, etc.) patients and doctors should be able to contact the admin as fast as the system can do.

Also, since patients make their exercises with the application, there cannot be any time waste for waiting the system response because it makes the system unusable.

3.3.5. Error Handling and Extreme Conditions

First of all, since the doctor id will be given by admin and the patient id will be given by doctor, there will not be any confusion. There are some situations these will cause errors like the user will enter wrong password, doctor will try to give same id to more than one patients unconsciously, there will some network connection problems, etc. since these are not very critical or insolvable, the errors can be solved easily by giving some warning messages.

3.3.6. System Interfacing

As we mentioned before, we are planning to create a simple and easy to use interfaces for our web and game applications. The inputs like patient and doctor id will be come from inside of the system (we will give doctors to id, doctors will give id to patients), the users will not create their own profile,
they can just edit it. For instance, editing the height and weight information will be an addition for the patients or adding patients will be a role for the doctors. Also, this is a restriction for patients and doctors that they cannot create their own profile.

3.3.7. Quality Issues

We want to have our system to be a reliable as much as we can do. This is because we want to create a virtual environment for patients to have fun when they do their exercises. We cannot reach our aim if we couldn’t make a system to make patients feel like they are in a real environment. So, the system should be stable and reliable by keeping up with patients. We are trying to minimize the time for restarting the system after a failure for this situation because sometimes we have these kinds of failures and we already started to fix them.

This is not a very portable system since we use Kinect camera. The camera should be placed in a wide area and considering that the patients cannot carry it every time they use, the Kinect camera should be placed to permanent location.

3.3.8. System Modification

We can add some new exercises and games after we receive some feedback from the patients and doctors as a modification. Also, changing the interfaces depends on the users’ thoughts can be a modification for the project.

Additionally, we can modify the patient’s web application with adding a button for installing their exercise history as a pdf. Also, we can add this button to doctor’s web application for getting the patients’ exercise state history.

3.3.9. Physical Environment

As a first and must requirement, there should be at least 2 meters by 2 meters space for the camera to used and for people to be seen in the camera with their whole body. Also, there should be an empty space for Kinect to detect accurately the patient and to provide them a movement area. We think that this equipment will be used in some physical therapy center at first by the reason that is everyone cannot have this equipment in their home or company, etc.

3.3.10. Security Issues

The system should be controlled because we will keep the patients’ and doctors’ personal and password information in the database. We have to salt the passwords, then encrypt and get a
password hash. This is the most common and safe way to keep password information in any database. Also, we need to create two web applications besides the game application.

3.3.11. Resources and Management Issues

This system should be backed up regularly and short time period because we will keep the data for every patient of the doctors and that information include how many time the patient can repeat the exercise, which exercise the patient completed, what percent of movements were done correctly, improvement rate in completing exercises, etc.

Administrator is the only authorized person for backing up the system, installing and maintaining it.

3.4. System Models

3.4.1. Scenarios

3.4.1.1. Patient Scenario

3.4.1.1.1. Game Environment Scenario

I. A patient opens the web application.
II. If patient enters the system for the first time:
   i. Patient enters the id which were given by the doctor earlier, then user choose a user name and password.
   ii. Patient can change the password if s/he wants.
   iii. Patient fills the form about his weight, height.
III. If the patient enters the system before, sign in the system with his/her credentials.
IV. If patient credentials are false, patient try to login with correct credential.
V. If the patient forgets his/her password or id, s/he sends a request for getting his/her credentials to his/her mailbox.
VI. If patient credentials are true, patient passes to the next page.
   i. Patient checks on the feedbacks about his/her success rate on exercises that sent by the doctor.
   ii. Patient can update his/her information.
VII. Patient sign out from the system.
3.4.1.2. Web Environment Scenario

I. A patient opens the web application.

II. If patient enters the system for the first time:
   i. Patient enters the id which were given by the doctor earlier, then user choose a user name and password.
   ii. Patient can change the password if s/he wants.
   iii. Patient fills the form about his weight, height.

III. If the patient enters the system before, sign in the system with his/her credentials.

IV. If patient credentials are false, patient try to login with correct credential.

V. If the patient forgets his/her password or id, s/he sends a request for getting his/her credentials to his/her mailbox.

VI. If patient credentials are true, patient passes to the next page.
   i. Patient checks on the feedbacks about his/her success rate on exercises that sent by the doctor.
   ii. Patient can update his/her information.

VII. Patient sign out from the system.

3.4.1.2. Doctor Scenario

Web Environment Scenario

I. A doctor opens the web application.

II. Doctor logon the system.

III. If doctor credentials are true, doctor passes to the next page.
   i. Doctor selects a patient.
   ii. Doctor sees and analyzes a patient’s progress.
   iii. Doctor adds or deletes movements to a patient.
   iv. Doctor can update his/her information.
   v. Doctor log out from the system.

IV. If doctor credentials are false, patient tries to login with correct credential.

V. If doctor forgets his/her credential, he/she opens to web application and he/she wants his/her credentials to his/her mailbox.
3.4.2. Use Case Model

Figure 1: Use Case Model To Patient and Doctor
3.4.3. Object and Class Model

3.4.3.1. Database Model

Figure 2: Object and Class Model (for database)
3.4.3.2. Game Environment Model

Figure 3: Object and Class Model (for game environment)
3.4.4. Dynamic Models

3.4.4.1. Database Sequence Diagram

Figure 4: Sequence Diagram for Database
3.4.4.2. Doctor’s Web Environment State Diagram

Figure 5: Doctor Web Application Interface State Diagram
3.4.4.3. Patient’s Game Environment State Diagram

Patient Game Application Interface

- Forget Password
- Want an email to mail address
- Enter password

Figure 6: Patient Game Application Interface State Diagram
3.4.4.4. Patient's Web Application State Diagram

Patient Web Application Interface

- Register
- Change password
- Add personal information

Forget Password
- Want an email to mail address
- Enter password

Login
  If the password is wrong
  Else

Check the feedbacks given from the doctor

Check the exercise history

See the exercise progress

Figure 7: Patient Web Application Interface State Diagram
3.4.5. User Interface

We planned to have three different interfaces first one is for doctor web-application interface, second one is for patient interface and last one is game interface.

In doctor’s interface, we planned to have some buttons for different causes. First of all, the doctor can contact us for problems in the system and for more information. Next, we planned to give exercises we put in game and how they are done correctly with videos to exercises button. And finally and most importantly, we have a choose patient button which is used for adding and deleting patients, assigning exercises to patients and see the success rate of the patients.

![The Physio Mate](image)

*Figure 8: Doctor Interface*

In patient’s interface, we planned to have patient’s about my improvement button which shows the patient the movements he made and made with how much accuracy and how much he improved himself from the beginning. Also, feedbacks from his doctor will be seen in this page. And has a start button which leads patient to the game.
Finally, in the game interface patient will select a level of the game which is an exercise he must do and play the game. Also, he has to log in to the system for us to hold the data for him.
### 4. Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>Information</td>
</tr>
<tr>
<td>Admin</td>
<td>Administrator</td>
</tr>
<tr>
<td>Uname</td>
<td>User name</td>
</tr>
<tr>
<td>FR</td>
<td>Functional Requirement</td>
</tr>
<tr>
<td>R-ID</td>
<td>Register-id which will be given by doctor to patient</td>
</tr>
<tr>
<td>Feedback</td>
<td>The comment that is given to patient by the doctor about the patient’s improvement</td>
</tr>
<tr>
<td>Patient Program</td>
<td>The game environment that is used by patient for doing the physiotherapy movements</td>
</tr>
<tr>
<td>Movement</td>
<td>The movement that patient will do for his/her physiotherapy</td>
</tr>
<tr>
<td>Movement List/ Movement Cart</td>
<td>The list which will include the physiotherapy movements which will be done by the patient.</td>
</tr>
<tr>
<td>Movement Area</td>
<td>The area which is used by the patient for doing the movements</td>
</tr>
<tr>
<td>Password Hashing</td>
<td>Hashed version of the password</td>
</tr>
<tr>
<td>Password Salt</td>
<td>The data which is used for hashing the password</td>
</tr>
<tr>
<td>Kinect</td>
<td>The tool we will use for game environment in our project</td>
</tr>
</tbody>
</table>