Case-based reasoning in postoperative pain treatment

Thesis Report By:
Arash Kianifar (870427-0951)

Supervised By:
Mobyen Uddin Ahmed

Examiner:
Peter Funk

MÄLARDALEN UNIVERSITY
SWEDEN

School of Innovation, Design and Engineering (IDT)
Mälardalen University
Västerås, Sweden
Abstract

Even today, with modern medicine and technology, post-operative pain still exists as an major issue in modern treatment. A lot of research efforts have been made, in order to improve pain outcome for patients that has undergone surgery[18][15].

Even though physician's and doctors are well educated, the success rate is about approximately 70 %, still there are patients that experience severe pain, after they have undergone surgery. There could be several reasons to this, for example, lack of methods or support should be amongst other things, factors to consider[18].

The problem has been to initiate a case-library and eventually create a tool, that could aid physicians or doctors in their decision making, which hopefully would help in improving pain outcome. The chosen method to do this, is a modified version of the CBR-algorithm, which is an artificial intelligence algorithm. The CBR-algorithm makes use of features, solution and outcome, and is implemented with a simple prototype, as a similarity function.

The are several reasons for why this method was chosen, but using this method makes it possible to easily create a web-based tool, so it can easily be accessed from anywhere, but still be effective and work as a support tool.

The algorithm works as a self learning mechanism, and is easy to implement, and the interface has been constructed, allowing the physician or doctor to retrieve information about patients and run CBR. The desired results are as expected, it's possible to run the CBR, retrieve and compare cases, and get suggestion of solution or action that should be performed.

The conclusion that can be made, is that, although this is a very basic working medical application, still an overall improvement is needed in order to be used as a medical application. It's anyhow a start. For more details and information, check the appendices please.
Figures

Figure 1: CBR cycle.........................................................................................................................................10
Figure 2: The work cycle..................................................................................................................................11
Figure 3: The CBR work flow........................................................................................................................13
Figure 4: The new case entered, in main GUI..................................................................................................15
Figure 5: new case added in database............................................................................................................16
Figure 6: CBR result and feature comparison..................................................................................................16
Figure 7: table architecture .............................................................................................................................18
Figure 8: CBR case and Case 7 comparison....................................................................................................28
Figure 9: CBR case and Case 4 comparison.....................................................................................................29
Figure 10: the main GUI...................................................................................................................................30
Figure 11: the user GUI Implementation..........................................................................................................31
Figure 12: The basic database structure........................................................................................................32
Figure 13: The User Guide..............................................................................................................................33
Figure 14: Table Screening.............................................................................................................................37
Figure 15: Demographic Information.............................................................................................................37
Figure 16: Medical History...............................................................................................................................38
Figure 17: Surgical Procedures.......................................................................................................................38
Figure 18: pre-medication.................................................................................................................................39
Figure 19: intra-operative treatment..............................................................................................................40
Figure 20: recovery room1.............................................................................................................................41
Figure 21: recovery room2.............................................................................................................................41
Figure 22: recovery room and ward.................................................................................................................42
Figure 23: ward................................................................................................................................................43
Figure 24: ward and questionnaire..................................................................................................................44
Figure 25: questions p1-p6...............................................................................................................................45
Figure 26: questions p7-p11.............................................................................................................................46
Figure 27: questions p12-p17...........................................................................................................................46
Figure 28: ranked order list...............................................................................................................................47
Figure 29: comparement table start.................................................................................................................48
Figure 30: comparement table end..................................................................................................................49

Tables

Table 1: First Case base..................................................................................................................................20
Table 2: The Second case base.......................................................................................................................27
Table 3: The Similarity Matrix........................................................................................................................35
Contents

1. Introduction.............................................................................................................................................6
   1.1 Objective of the Thesis.........................................................................................................................7
   1.2 Problem Formulation.............................................................................................................................7
2. Background and Related Work..................................................................................................................8
   2.1 PHP, HTML and PHP MySQL..............................................................................................................8
      2.1.1 PHP..............................................................................................................................................8
      2.1.2 HTML.........................................................................................................................................8
      2.1.3 PHP MySQL.................................................................................................................................8
   2.2 CBR..................................................................................................................................................8
   2.3 Post-Operative Pain.............................................................................................................................9
   2.4 Pain and Operations.............................................................................................................................9
   2.5 The Pain-Out-Project...........................................................................................................................9
   2.6 Related Work..................................................................................................................................10
3. The Work Cycle........................................................................................................................................11
   3.1 The Main Work Process......................................................................................................................11
   3.2 Outcome.........................................................................................................................................12
      3.2.1 Outcome 24 hr..............................................................................................................................12
      3.2.2 Outcome Week.............................................................................................................................12
   3.3 Features.........................................................................................................................................12
4. Approach and Method..............................................................................................................................13
   4.1 Choice of Method...............................................................................................................................13
   4.2 CBR and A Scenario...........................................................................................................................14
      4.2.1 CBR Functionality.........................................................................................................................14
      4.2.2 A Scenario..................................................................................................................................15
      4.2.3 The Local Similarity Function......................................................................................................17
5. Implementation.......................................................................................................................................18
   5.1 Database Table Architecture.............................................................................................................18
   5.2 CBR Implementation...........................................................................................................................19
6. Evaluation...............................................................................................................................................20
7. Discussion and Future Work....................................................................................................................21
   7.1 Problems Faced................................................................................................................................21
   7.2 Discussing The Solved Problems........................................................................................................22
      7.2.1 Importing Data.............................................................................................................................22
      7.2.2 Choosing Correct Data Types......................................................................................................22
      7.2.3 Web Browser Support...............................................................................................................23
      7.2.4 Implementation of the CBR............................................................................................23
      7.2.5 Future Work................................................................................................................................24
8. Conclusion and Summary.........................................................................................................................25
9. References............................................................................................................................................26

Appendices

Appendix A : Another CBR Example............................................................................................................27
Appendix B : Screenshots and the Large Scale Work..................................................................................37
Appendix C : Introduction to PHP MySQL..................................................................................................50
Appendix D : Other Links..........................................................................................................................64
Acknowledgement

With all respect, many thanks to Mobyen Uddin Ahmed and Peter Funk for their support and help. During this work, they have been very helpful and supporting. Without them, it would not have been possible to make it this far and learn so much, thank you for your great help and guidance throughout the work of this thesis. Also special thanks to all members of the Pain-Out project. This thesis work is a part of the pain-project!
1. Introduction

Pain is still a major issue in modern treatment. A lot of research efforts have been made, in order to improve pain outcome for patients that has undergone surgery. Even though there are physician's and doctors that are well educated, the success rate is not fully as expected[18][15].

Still there are patients that experience severe pain[18], after they have undergone surgery. There could be several reasons to this, for example, lack of methods or support for clinical decision making are factors to consider, as well as how well educated and how much knowledge a phycisian or doctor have, and the mount of doses given to the patients. And amongst other factors, it's inevitable to ignore the fact that, even though there are experts and very well educated phycisians and doctors, still, it's a difficulty to try adapt solutions, so that they are well suited for the specific patient, since all patients have their own unique needs.

The problem has been to initate case-library for a case-based system in post-operative pain management, to make it possible to run different functionality of the existing case-based system. Another problem is also to make it possible to support physicians in their decision about which solution to give to a certain patient, based on previously known cases, or in other words, based on the experience that previously known patients have gone through. So, with the help of a special type of algorithm, it's possible to achieve this functionality. The method or algorithm that has been used, is called Case-based reasoning. Suppose that the physician make use of this decision support tool, and she/he gets informed about that previous patients have gone through a bad experience, for a certain anesthetic or solution, then the physician is informed about that this solution is maybe not ideal.

Another example can be also be that, the decision support tool, informs the physician, that a certain mix or combination of specific solutions or anesthetics, is not recommended, since a combination might be dangerous, in some cases. However, although this decision support system is intelligent, it's not relaible to let it make the final decision, so the final decision is made by the physician. How effective the algorithm is, depends partly on how good and dependable the chosen features are, which are parameters that affects the algorithm. The features shall usually be collected by the physician, doctor or nurse, before the operation. Eventually, the solution will be carefully filled in by the physician, and if the patient is willing, he/she will be able to answer questions regarding the pain he/she experienced after the operation.

To make it easy and user friendly for the physician, a web-based tool that supports all common web-browers today, has been constructed. This tool allows the physician, to set weights for each feature and get information about how similar the current case is with the other cases, shown in a list with ranked order, with the most similar case top of the list. Additional information is also presented together with the ranked list, such as experienced pain and surgical procedures, since these are very useful information to the phycisian.

And if the physician wants to, he/she can even compare the current case features with other cases features, making it possible for the phycian to get additional detailed information, and also information about solution given of course. The results has been as desired, the phycisians can now retrieve information and make use of the CBR tool, in order to get
detailed information and make comparisons, and decide which solution to give. Although, conclusions that has been done so far, is that, the tool is not yet fully ready to be used as a medical application yet, an overall improvement is still needed, in both the GUI and the CBR tool. Improvements that still can be made, is for example, improving the CBR similarity function. Making it possible to easily change the weights for the features, additional information in the ranked-order table, and also some other things. However, it's a good start at least, building a medical application. With improvements, this could maybe be a helpful and effective medical application that would satisfy a physician's all needs, and perhaps, help improving pain outcome.

1.1 Objective of the Thesis

The main objective of the thesis has been to initiate case-library for a case-based system in post-operative pain management, to make it possible to run different functionality of the existing case-based system, and eventually creating a tool, that aids physicians and doctors in their decision making. This is a step toward improving pain outcome, since the tool aids physician and doctors in making a more accurate and better decision. An important task has been in this thesis, to understand the overall data structure, datafields and datatypes, the frame layout, and also amongst other things, fixing the main problems when loading the data from the database-files into the corresponding tables and fields. In order to finish the main objective, there has been many obstacles to face, and careful analysis and testing has been done, to finish the objective of the thesis.

1.2 Problem Formulation

Before working with the retrieval process, creating the graphical user interface and implementing CBR, it was necessary to first study the features, datastructures, most sql basics, the framework, and the libraries from which the data was going to be imported from. Careful analysis and understanding of everything as mentioned above was necessary in order to complete the tasks. However, all tasks have been completed, but still, problems that problems that was faced, was also solved during the work.

Some of the problems faced, are as following:

• How is the importing of data going to work?
• How to confirm that the data was imported correctly.
• How to make sure all web-browsers are supported.
• How shall the datatypes for certain fields look like, when importing data from the libraries, since the database files the data is loaded from, are Oracle and not Sql?
• How shall the CBR tool be implemented?

How the problems were solved, will be discussed further on in the thesis report.
2. Background and Related Work

2.1 PHP, HTML AND PHP MySQL

2.1.1 PHP

php is a widely used general purpose scripting programing language. Learning php is not difficult, especially not difficult if one have programing experience since earlier. PHP is used generally in webservers, and can also perform desired operating system operations. It's somehow similar to the C language.

2.1.2 HTML

Html is the language for webpages. When writing an html page, usually html elements is used, which consist of so called "tags" surrounded by "<" and ">", within the page content, a html page usually start with the <html> tag, and ends with </html>. Html also supports javascript and css. Html is not difficult to learn, and there are also programs to facilitate development of webpages, such as Dreamweaver.

2.1.3 PHP MySQL

MySQL is a database system. The data is stored in tables, which consists of data entries. A table consists of rows and columns. Getting data from a database table is easily made by simple queries.

2.2. CBR

Case-based reasoning(CBR) is a method for solving new problems, based on solutions for previously solved problems, similar to the current problem. Case-based-reasoning is also a well known method or self-learning algorithm[16][1][14][3], that can be used for building medical applications, providing medical AI support, to aid clinicians in decision making. CBR is not only used to build medical applications, it's widely used in other areas as well, to help solve problems, based on earlier knowledge.

It's important to note that, even though the medical applications are effective, it's not ideal to fully rely on the medical AI applications, in other words, letting the medical application to have the responsibility. But having the physician making the final decision with the aid of the intelligent medical cbr based system, makes it an excellent combination!

Another area where CBR can be used, could be, as an auto mechanic, that helps repair engines of helicopters based on solutions(action made) for earlier cases(helicopters that was repaired) most similar to this helicopters engine problems. Well, CBR can be used anywhere, where some auto-mechanic might be needed, or decision support also. CBR is a powerful self learning mechanism, and is very useful in many areas and not only when creating medical applications. It's something widely used, in medical, mechanical, economical, and in dustrial areas as well, and in many other areas.
2.3. Post-Operative Pain

Pain, can be seen as a defense mechanism from our body, telling us to withdraw from harmful situations, avoiding things that could seriously damage the human body. A situation where the feeling of pain is intense, could be for example, when seriously burning a body part. In most medical conditions, pain is seen as a major symptom, the reason for this, is partly because pain can badly affect our lifestyle[20][19].

However, even though it works as a defense mechanism, it can in many cases be a bad thing. According to a research, pain more or less prevents recovery, causes suffering and badly affects our health[19]. Depending on the situation, pain could for example cause us to sleep less, make us less willing to eat food, and also make us less willing and able to move. Except for these things, depending on the condition, pain can amongst other things, also cause depression, further aggravated pain, and seriously affect factors that are essential for maintaining a healthy lifestyle. With today's technology, different types of anesthesia that help us to reduce pain has been created, especially to help reduce pain after operations, but also during and before, though this, pain still mostly remains as consequence of operations[19].

2.4. Pain and Operations

There might be several reasons for why most patients that have undergone surgery, experience severe pain. It can for example, depend on the physician's decision that was made. An improper decision could lead to serious consequences, and can even be life threatening. The physician's knowledge, education and experience are important factors to consider, but the amount of doses given are also factors that needs attention.

So, one might wonder, is it enough with educated physicians that have a vast amount of knowledge and experience? How are physicians able to adapt the treatment, so it's well suited for a specific patient? According to a research made, there is lack of support for clinical decision making and improvement of pain outcomes, across the European continent.

According to the same research, lack of methods to measure and compare outcome quality, is a big problem for clinicians to assess[17], which in turn, as a consequence, affects how well clinicians make their treatment decisions. According to the same research again, it is ideal to improve clinician decision making in the treatment of patients with post-operative pain, which in turn will result in an improvement of treatment outcome for citizens across the European continent.

2.5. The Pain-Out Project

The pain-out-project is a European funded project, that is aimed at improving the treatment of post-operative pain for European citizens. There are several reasons for why this research projects was started, but two main purposes of this projects is to first, construct a huge database that consist of a large amount of cases, and the second one is to create tools and medical applications, that make use of the information stored in the huge database, to help increasing knowledge, reducing pain, and thereby also improving pain outcome[18].
2.6 Related Work

A CBR method can very much work like human reasoning, e.g., solves new problems by making use of previously known experience (past experience). This is basically how experts increase their knowledge and make use of it in new situations. Here an expert, makes use of his/her knowledge and experience in order to give the correct and well-suited solution or anaesthetics for the specific patient. In the same way, the CBR system can make use of past experience which is actually stored cases, to determine which solution or anaesthetics that is best suited for the specific patient[14].

![CBR cycle](image)

Figure 1: CBR cycle [6]

CBR is made up of four steps, see below please.

1. **Retrieval:**
   This is the process where the retrieval of the cases is made, within the case base. Not any cases though, but the cases that are most similar to the current case.

2. **Reuse/Adaptation:**
   In this case, the solution of the previous case is mapped to the problem. In this process, usually the solution might be needed to be adapted to be able to be as a solution for the current problem.

3. **Revise:**
   In this process, when the previous solution have been mapped to the problem, the new solution shall be tested, and if it's not good enough, eventually revised and tested.

4. **Retain:**
   After a good and well-working solution as been found and adapted to the target problem, the resulting experience shall be saved as a new case in memory, for further use.
The CBR cycle is given in Figure 1 which is discussed in [6], and seen from above, CBR consists of four parts. The retrieve process can be considered as taking the most similar case from case library, with respect to a new case. If new case is solved by making use of the previously known solved case, then it is considered as reuse. Through the revise cycle, that's where the solution is checked, to see whether it was good enough. If it's all a success and the solution was good, the new case and the solution will be considered for future use, this experience is considered as learning, and corresponds basically to phase 4, retain. The CBR and retrieval phase is implemented and evaluated in this thesis.

3. The Work Cycle

3.1 The Main Work Process

The actual main process (the entire routine), starts with the physician or doctor collecting the features in the ward, before the operation. And then, the solution (anesthesia), that is adapted/suited for the specific patient will be used. 24 hours after the surgery is complete, the patient will be answering some questions regarding the experienced pain, that is, the "outcome after 24 hours".

One week later, after the surgery, the patient will again, answering another questionnaire, but this time, questions regarding the pain (if any), that the patient experienced one week after the surgery, which is the "outcome after one week". Note, to be able to save and store the current case, information about the patient's solution, outcome after 24, and outcome after one week, must be collected.

The main process cycle can be described in four steps:

1. Firstly, the collection of the features in the ward, before the actual operation.
2. The suited/adapted solution (anaesthesia) will be used for the specific patient.
3. 24 hours after the surgery, the patient will be answering questions about the experienced pain.
4. Once again, but this time, one week after the surgery, the patient will be answering questions regarding the experienced pain.

The four routines in the main process, can be illustrated visually, as shown below.

![Figure 2: The work cycle.](image)
3.2 Outcome
The outcome could be seen, as the result of the actions(solutions) that was performed. So in this case, the outcome is basically the pain that the patient experienced. Hopefully, in this case, the outcome is improved by the actions that was made, which in this case was the solutions or anesthetics that was given to the patient, in order to improve pain outcome.

3.2.1 Outcome 24 hr
The information about the outcome(experienced pain) after 24 hours, regarding the patient, is retrieved when the first questionarie is answered 24 hours after the surgery or operation has been completed. The questions, mostly consists of a pain sub-scale, that ranges between 1 – 10, where a higher value means worst pain that is possible, and lesser value means, no pain at all(a value of 0) or less pain. Since the pain, is based normally on the patients own experience, it makes it partly a little complex and difficult to compare pain. In other words, its partly something more abstract, rather than concrete.

3.2.2 Outcome Week
As it says, outcome after one week, is the pain the patient experienced one week after the surgery or operation. Again, the information about the outcome(experienced pain after one week) is retrieved when the patients answers the second questionarie, one week after the operation.

3.3. Features
Features are parameters that affects the CBR algorithm. Features regarding a patient, could be age, gender and weight. It's not always easy to choose good features, and can in some cases be difficult. Since features affects the CBR algorithm, it's important to have "good" features. Good features, can be seen as data or parameters, that affects the CBR to perform well in making its decision accurately.

If CBR shows a list with the most similar cases to the current case, and if there are good features, the CBR should hopefully, be able to perform well in making a decision that influences or affects the resulting pain, in a way that, it makes a good and accurate suggestion for the physician, about which solution to give for the specific patient. Although, it's the doctor making the final decision. The CBR only supports the physician as a decision support tool.

Depending on the situation, bad features could cause the CBR to make bad decisions, for the physician, making it a dangerous tool, rather than a helpful one. So, selecting good features as mentioned, are very important. In this project, the importance or weight factor has been created, for each feature, making it possible for the physician, to further make use of his/her expertise. The weights, range between 1 to 10. A high value means important, and a low value means less important.
4. Approach and Method

To better understand how the entire process works (the CBR work flow), this can be visualized by a simple flow diagram as shown below, which will also further down be explained.

![CBR Work Flow Diagram]

**Figure 3:** The CBR work flow.

4.1 Choice of Method

Since it was necessary to design a tool that can easily be accessed, and be available from nearly anywhere, why not let it be web-based? Having the tool being web-based, there is no need of installation. Everything including the user-guide are web-based. The web-based tool is based on the algorithm called Case-based reasoning (CBR).

Some advantages of the web-based tool are as follows:

- Since it's web-based, it's easy to access, from nearly anywhere.
- User guides can also be web-based.
- No installation required.
- The user is also able to make detailed comparison of the features for the current case, with the other cases that are shown the ranked list, presented by the CBR.
Some advantages of the CBR algorithm:

- CBR effectively shows the most similar cases to the user, allowing the user to further analyze and more accurately decide, which solution to give to a certain patient.
- CBR is said to be easy to understand and use.
- CBR learns by itself, thereby increasing knowledge[14].
- CBR is easy to implement.
- CBR is also said to be intuitive, seems to match our experience[14].

Some of the disadvantages with CBR:

- Cases may not cover domain well.
- Most appropriate cases may not be retrieved.

4.2 CBR and A Scenario

4.2.1 CBR Functionality

CBR (Case Based Reasoning) is an Artificial Intelligence algorithm, that is based on previous knowledge and works as a self-learning mechanism. CBR makes use of features and outcome, to give suggestion of which solution (action recommended) that might be good for the specific case. Here is a brief description of how the CBR mechanism works[13]. For a more detailed example of how the algorithm works, see Appendix A please.

1. CBR makes use of a long list of previously known cases. And for each case, there is a solution (action) that has been performed to solve the specific problem. There is also an feature part for each case as well. So, this means that, the features (also called circumstances) and action (solution) performed is known for the previously known cases, as well as the outcome (the result). So, action (solution) performed, the circumstances (features) and the outcome (result), is known for the previously known cases.

2. When a new case shows up, only the features are known, that was collected by the phycisian.

3. So, the CBR now compares the new case with all previously known cases, one at a time. The comparison is made by comparing the corresponding feature part of each case, one feature at a time. The weight and similarity of each features is what determines the similarity (weighted similarity) value. And eventually, after all features between the current case, and the target case has been compared, a global similarity value is calculated, as a result of similarity between the current case and the target case. The global similarity is basically the sum of all similarity (weighted similarity) that was calculated for all the features between the current case and the target case.

4. Now, after that all calculation of similarity has been done, the CBR will show the most similar cases, usually in a ranked order list. This aids the phycisian to make his/her choice of the action to perform, to solve the new case. Probably the most similar case shown by the CBR is to prefer, but still, the decision is up to the phycisian.
4.2.2 A Scenario

Even though a brief description of the CBR functionality has been explained, let's consider a more realistic example. Since this is an example, imaginary data is used. Consider a situation, where a patient named Simon, is very ill and have a bad state of health. Besides, he is in urgent need of medication for his dangerous disease he have, which is: Histoplasmosis. This is a disease that primarily affect the lungs.

This man, named Simon, arrives at the hospital, and meets with the physician on the ward. Together with the physician, Simon answers some questions on a computer screen, which is the features actually, such as: "Age", "Gender", "Weight", "Health State", that is directly afterwards, stored in the database. Look at the picture of the tables below please.

![Figure 4: The new case entered, in main GUI.](image-url)
This is how it looks like in the database, after the features has been entered, as shown in the figure below.

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Age</th>
<th>Weight</th>
<th>Health_State</th>
<th>Name</th>
<th>Solution</th>
<th>Solution_Cost</th>
<th>Solution_Description</th>
<th>Outcome_Direct</th>
<th>Outcome_Week</th>
<th>Outcome_Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>male</td>
<td>48</td>
<td>90</td>
<td>normal</td>
<td>Tommy</td>
<td>x2</td>
<td>4000</td>
<td>Tommy is ok</td>
<td>low</td>
<td>medium</td>
<td>none</td>
</tr>
<tr>
<td>2</td>
<td>female</td>
<td>25</td>
<td>71</td>
<td>normal</td>
<td>Nina</td>
<td>x3</td>
<td>5600</td>
<td>solution is ok</td>
<td>high</td>
<td>medium</td>
<td>none so far</td>
</tr>
</tbody>
</table>
| 3  | male   | 44  | 68     | good         | Anders| x1     | 5000          | potent!             | low            | low         | good results!
| 4  | female | 38  | 50     | bad          | Tina | x2     | 4000          | Good solution for  | high           | high        | good results!
| 5  | male   | 49  | 64     | bad          | Jack | x3     | 5600          | strong solution for | low            | medium      | good pain relief for Jack! |
| 6  | female | 45  | 85     | normal       | Emily| x1     | 5000          | potent!             | high           | low         | none so far  |
| 7  | male   | 48  | 90     | good         | stefan| x2     | 4000          | somewhat potent!    | medium         | medium      | none so far  |
| 8  | female | 25  | 70     | normal       | alexandra | x3    | 5000          | well working solution | medium        | medium      | nothing special yet! |

**Figure 5:** new case added in database.

So, the phycisian decides to make use of the CBR tool. He uses the CBR tool, and gets the result shown in the following figure below.

The phycisian gets a list of the top most similar cases, and decides to press on the top most similar case, "5", which will show comparison between the current case 9(Simon) and the case "5" and also the case below "5", which is "4". As you can see, the phycisian gets a good comparison. But even though Case 5 is the most similar one, as it can be seen, the phycisian decides, according to his own experience as a phycisian, to give the patient the solution that the patient for Case 4 got, which is: x2, and the solution cost is 4000. Note that it is the doctor that makes the final decision, and not the CBR system. The CBR only helps the phycisian to come up with a good solution.

The doctor maybe for some reason of his own experience, decides that the solution the patient for case "4" got, is more suited for this patient, rather than the solution for the most similar case "5". The solution maybe even need to be adapted somehow, to be suited for the patient named Simon. However, the cost factor might also be something that affects the phycisians's final decision. Well, Simon gets his medication, and lets say in this case that it worked very well.

Arash Kianifar

-16-

Bachelor Thesis – B Level
The physician and the patient also discuss about how the patient felt, directly after taking the medicine, when he was with the physician (the physician notes this in the database, directly after he got the answers about how the patient feels, since the medicine has a, let's say "direct effect").

The physician and the patient meet again, after one week, to see how the patient is feeling. The physician notes that the patient is much better (after one week), and the solution given to the patient seems to have been effective enough to prevent the dangerous disease called: Histoplasmosis. The doctor then, after the meeting, fills in the outcome for the patient, and notes (in the outcome description), that this solution for the patient was effective, and the patient has much less pain than he had when he was very ill because of this disease, which very much affected his lungs.

So, the doctor fills in the outcome for the patient, and the data about this patient named Simon is saved into the database, thereby increasing knowledge and reducing pain for this patient. Thanks to the physician and the medical AI support the physician was supported by, this patient feels much better than earlier. A combination of a physician's expertise and medical AI support, make it a powerful tool.

### 4.2.3 The Local Similarity Function

This function is responsible for calculating similarity values between the CBR case and the other cases. It compares corresponding features, and calculates the similarity value accordingly. However, earlier it was mentioned that some problems were encountered when defining the similarities. Some of the problems are as follows:

- How should similarity between symbolic (strings) features be calculated?
- How should the function handle similarity calculation between numerical features?
- How should the similarities according to specific distances be defined?
- How should the function calculate similarity between floats?
- How about missing data?

For handling string-type features, a simple case-insensitive comparison has been done so far, to see whether they are equal or not. However, that is not the best way to compare string types, but might work when comparing very simple string-feature values. The implementation of the function is very simple. For example, if there would be strings such as "string" and "String", it is most likely, basically the same word, but would be recognized as different.

Of course, it's possible to convert the strings to lower-case, before making a comparison, but that would only be a temporary solution, since in some cases, it might be needed to make case-sensitive comparison. However, the integer-types, are handled using Euclidean "multi-dimensional" distance formula. When defining the similarities for distances, it was not that easy to come up with the definitions of how the similarities should look like, depending on the range.
But after some testing and careful planning, the definitions of the similarities was created, as shown in the similarity matrix. How about floats? That has not been handled yet, and that's something that actually needs some attention. So far, so good, but still, there are improvements needed.

However, as it is a good start, it can also be altered or changed, for improvement and better functionality. An improvement could for example be, making it handle string-type features more effectively and also handling of floats could be a good additional improvement. Missing data is also something that should be considered to be handled. So, generally, this function has a basic start, but needs to be improved, for better functionality. There is always something that could be improved.

5. Implementation

To see screenshots about how the GUI looks like, see the appendix please. Here, focus will be on the system table architecture and implementation of the CBR function. First of, lets take a look at how the system architecture looks like, in other words, the structure of the tables, containing data, and how they are related to each other.

5.1 Database table architecture

![Figure 7: table architecture.](image)

There is something important that shall be noted here. Since real relations was not used when the SQL tables was created, this is more to illustrate how the connections or in other words, "INNER JOIN" SQL matching is done, when selecting data, as if there was any kind of relation between the tables. Now that there are some basic information about how the system architecture looks like, lets take a closer look at the CBR implementation.
5.2 CBR Implementation

To understand how the CBR algorithm works, you must know how the similarity function works, the formulas and all other functions. These will be explained below.

This is how the CBR basically functions:

1. Read current case from database.
2. Read all other cases from database.
3. Calculate local similarity between the current case features, and all other cases features, one at a time, using the local sim function:

\[
\text{sim}(f_1,f_2) = 1 - \frac{\text{abs}(f_1-f_2)}{\text{max} - \text{min}}
\]

**Formula 1:** Calculate similarity between two features, without the need of normalization of the value, where \(f_1\) and \(f_2\) are the feature values, and abs is used to get the absolute value of \(f_1-f_2\).

\[
\text{Similarity(case1,case2)} = \frac{\sum_{i=1}^{n} W_i \times \text{sim}(F_i_{CASEi},F_i_{CASEcbr})}{\sum_{i=1}^{n} W_i}
\]

**Formula 2:** The global similarity function calculation where \(CASEcbr\) is the CBR case and \(CASEi\) is one of the cases in caselist, \(W_i\) is the weight for \(CASEi\), \(F_i\) is the feature \(i\) and \(\text{sim}\) is the local similarity function. Not necessary to divide by weights, if the weights are already normalized.

4. Calculate global similarity using the global "similarity" function:

\[
W_f = \frac{\sum_{i=1}^{n} lW_f}{\sum_{i=1}^{n} lW_f}
\]

**Formula 3:** Calculating the normalized local weight \(W_f\), where \(lW_f\) = local weight for the feature \(f\).

5. Show the most similar cases in a ranked order list, to the user.
6. Evaluation

To evaluate, a simple example will be shown, about how the local similarity function calculates the similarity between features. So, consider that there are 5 features as show in the table below, and follow the example to see how calculation is done.

Case base:

<table>
<thead>
<tr>
<th>Case</th>
<th>D8_RESP</th>
<th>D1_GENDER</th>
<th>D3_WGHT</th>
<th>D8_GASTRO</th>
<th>BA_OPIOID</th>
</tr>
</thead>
<tbody>
<tr>
<td>382</td>
<td>1</td>
<td>1</td>
<td>102</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>626</td>
<td>0</td>
<td>1</td>
<td>71</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>617</td>
<td>0</td>
<td>0</td>
<td>68</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: First Case base

So, as seen from the table above, let's now take a look at how the calculation is done.

Local sim between Cases: 382(current case) and Case 626.

Feature: D8_RESP

<table>
<thead>
<tr>
<th>Values</th>
<th>Sim:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 382: 1</td>
<td>0.5</td>
</tr>
<tr>
<td>Case 626: 0</td>
<td></td>
</tr>
</tbody>
</table>

NLW: 0.066

Weighted Sim:

Weighted sim is calculated by multiplying the normalized local weight of the feature with the similarity value. In this case, the weighted similarity value between case 382 and 626 for the feature D8_RESP is as following: Sim * NLW = 0.5 * 0.066 = 0.033 where NLW is the normalized local weight of the feature. See the algorithm section for further details please, about how to calculate the normalized local weight.

Feature: D1_GENDER

<table>
<thead>
<tr>
<th>Values</th>
<th>Sim:</th>
<th>Weighted Sim:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 382: 1</td>
<td>1</td>
<td>Sim * NLW = 1*0.049 = 0.049</td>
</tr>
<tr>
<td>Case 626: 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Feature: D3_WGHT

<table>
<thead>
<tr>
<th>Values</th>
<th>Sim:</th>
<th>Weighted Sim:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 382: 102</td>
<td>0.6</td>
<td>Sim * NLW = 0.6*0.082 = 0.049</td>
</tr>
<tr>
<td>Case 626: 71</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Feature: D8_GASTRO

<table>
<thead>
<tr>
<th>Values</th>
<th>Sim:</th>
<th>Weighted Sim:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 382: 0</td>
<td>1</td>
<td>Sim * NLW = 1*0.066 = 0.066</td>
</tr>
<tr>
<td>Case 626: 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Feature: BA_OPIOID

<table>
<thead>
<tr>
<th>Values</th>
<th>Sim:</th>
<th>Weighted Sim:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 382: 0</td>
<td>1</td>
<td>Sim * NLW = 1*0.033 = 0.033</td>
</tr>
<tr>
<td>Case 626: 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Local sim between Cases: 382(current case) and Case 617.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Values</th>
<th>Sim</th>
<th>Weighted Sim</th>
</tr>
</thead>
<tbody>
<tr>
<td>D8_RESP</td>
<td>Case 382: 1, Case 617: 0</td>
<td>0.5</td>
<td>Sim * NLW = 0.5*0.066 = 0.033</td>
</tr>
<tr>
<td>NLW</td>
<td>0.066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1_GENDER</td>
<td>Case 382: 1, Case 617: 0</td>
<td>0.5</td>
<td>Sim * NLW = 0.5*0.049 = 0.025</td>
</tr>
<tr>
<td>NLW</td>
<td>0.049</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D3_WGHT</td>
<td>Case 382: 102, Case 617: 68</td>
<td>0.6</td>
<td>Sim * NLW = 0.6*0.082 = 0.049</td>
</tr>
<tr>
<td>NLW</td>
<td>0.082</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D8_GASTRO</td>
<td>Case 382: 0, Case 617: 0</td>
<td>1</td>
<td>Sim * NLW = 1*0.066 = 0.066</td>
</tr>
<tr>
<td>NLW</td>
<td>0.066</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BA_OPIOID</td>
<td>Case 382: 0, Case 617: 0</td>
<td>1</td>
<td>Sim * NLW = 1*0.033 = 0.033</td>
</tr>
<tr>
<td>NLW</td>
<td>0.033</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is how the calculation is done. The global similarity value can easily be achieved by plusing all the weighted similarity values together. For a more detailed example, see the appendix please. There are also actually more than what has been seen here to evaluate, such as the graphical user interface, but to get more detailed information, see the appendix please, here, only the most important things are evaluated.

7. Discussion and Future Work

7.1 Problems Faced

Before working with the retrieval process, creating the graphical user interface and implementing CBR, it was necessary to first study the features, datastructures, most sql basics, the framework, the libraries from which the data will be imported from, and most other things also. Careful analysis and understanding of everything as mentioned above was necessary in order to complete the tasks. However, all tasks have been completed, but still, problems was faced but also solved during the work.

Some of the problems faced, are as following:

- How is the importing of data going to work?
- How to confirm that the data was imported correctly, when importing data?
- How to make sure all or at least, most known web-browers are supported?
- How shall the datatypes for certain fields look like, when importing data from the libraries, since the database files the data is loaded from, are Oracle and not Sql?
- How shall the implementation the CBR tool be done?
7.2 Discussing the Solved Problems

7.2.1 Importing Data

When importing the data, very careful analysis of the database files was necessary, from which the data was going to be imported from. Everything was compared in this file, to the database tables that was created. However, one main problem, which was actually, very difficult to discover, but very easy to fix, was that, the order of the columns was not in the correct order in the database tables relative to the database files. As so far known, there are at least two ways of fixing the problem. The first one is that, the order of the columns in the database tables can be changed, that was created according to the information given at the pain out dictionary webpage, so that the order of the columns will match each order, both in the database files and the database tables. The reason for why it was difficult to find the problem, is because, there are alot of columns, maybe between 100 to 200 for each created table, and there was only two columns that did not match in being positioned in correct order.

The second way of fixing the problem, was actually, specifying in which order to read the columns(consisting of data), so that it matches the order of the columns in the database tables created. Anyhow, the first option was prefered. When importing the data, this could also be done in at least two ways:

1. Using the built-in import tool.
2. Using the "LOAD" query.

Since the second option was used, the "LOAD" query, lets see how it was done. After studying how the "LOAD" query works, a query such as the one shown below was used, to load everything from the database, into the tables, without any problems at all. To make sure the data was loaded correctly, random cases was selected, from the database file that was opened, and compared all the data of the selected case, to the same case data that was imported, to se whether everything was imported correctly or not. Here is the query that was used, to load data from the files, into the tables.

```
LOAD DATA INFILE 'C:/pain_out_qn.csv' INTO TABLE pain_out_qn
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '' STARTING BY '
'
IGNORE 0 LINES
```

In the same way, the data was loaded from the other tables. In the following section, the chosen datatypes will be shortly discussed.

7.2.2 Choosing Correct Datatypes

So, there is not much to say here. But choosing correct datatypes for the fields in the database tables created, was important, to make sure the data is in the correct format, when retrieved. The most effective datatypes chosen is basically "varchar", which is string actually. This type worked very well, for all the data in the database files that was imported, including the "date" types.
7.2.3 Web Browser Support

An important task was to make it work for all web browsers, since the tool is web-based, it's important that it supports most web-browsers. To make sure it well works for all web-browsers, everything was mostly done in php, and tried to not use javascript. Javascript works well of course, but there is a risk that one might experience bugs, depending on which web-browser that is being used.

Since php is much more reliable, it was ideal to stick to it as much as possible. Bugs with the javascript was encountered of course, that caused IE7, to not work properly when all other browsers that was tested, did fully work, including IE8. However, this problem was also solved, the problem was the javascript, that was a little buggy, however, it was fixed. After the problems has been fixed, IE 7 was tested again, and it worked properly. But still, as mentioned, php is the one to prefer, since it is much more stable and reliable.

7.2.4 Implementation of the CBR

The implementation of the CBR, was very much similar to the one that was implemented in the other work, see the appendix please. Some of the differences are as follows:

- The distance formula is now slightly changed, to use the "one-dimensional" euclidean formula, which is actually the absolute value of "number" minus "number"
- Similarities varies.
- The functionality to be able to change the weights for the features has unfortunately not been implemented yet, even though there are weights for the features.
- And possibly some other things.
7.2.5 Future Work

Well, there are a few things that would be ideal to improve. There have not been anyone to get feedback from, which would have been valuable. For example, reaching any physician that could comment and give feedback, would have been nice. Anyway, fake data was used in this project. There are indeed a lot of things that could be improved, some of them are as following:

- Improve the CBR interface, by adding extra information and functionality. Another table below the ranked list, providing the user much more information, could be added for example.
- Much more improve the user-guide.
- Improve the CBR interface, by adding extra information and functionality.
- Make it possible, to change similarity values, for everything, such as distance and comparison between symbolic features. This could be done, by having a similarity matrix, as has been seen earlier in this report, that is web-based, and can easily be configured via the web.
- Make it possible to change the weights of the features, which are more likely hardcoded at the moment. But this can also easily fixed whenever needed, which will allow the physician or doctor to make further use of his/her expertise by setting the importance factor for respective feature accordingly.
- The user is able to see some basic information about the cases listed, but however, more information to show is still needed.
- Realistic evaluation with real physician and doctors, saying their opinion, which has not been done yet, unfortunately.
8. Conclusion and Summary

The work in this thesis, has been tough, and lots of time has been spent on testing. Careful analysis has been done in order to make sure that everything was correctly done and working. When working with this thesis, a lot of reading and careful analysis was required, as well as some problem solving. A smaller work of CBR system was made, before doing the work in a larger scale. The functionality was very much the same for both of the works, except from that, the similarity varied, and there was many more features in the larger scale work.

The local similarity function needs improvements and the overall functionality is also something that needs improvements. The works, are both, very well good as a start, but not yet really ready to be used as a medical application. With maybe some improvements, of the overall functionality, perhaps it could be a effective medical application. The conclusion that can be made, is that, its a good beginning for a medical application, and with improvements, this could do a good medical program, and hopefully, completely fulfill a phycisian's needs. For more details, see amongst other things, the appendices please.

A lot of new knowledge has also been acquired or learnt, specially how to code in Javascript, PHP and MySQL, but also how CBR works and can possibly be effectively implemented.
9. References


Arash Kianifar -26- Bachelor Thesis – B Level
Appendix A

Another CBR Example

When working with this thesis or project, two different works of CBR systems has been made, they are mostly very similar to each other, the only biggest difference is that, the other one is made in a much larger scale. Below, you will be given an overview of how the first work of the CBR system has been done, which later on, was made in a much larger scale. The larger scale work, is what has been focus in the thesis report. To see figure and other detailed information, please see appendix B.

There are three important things to be evaluated:

1. The CBR
2. The GUI
3. The User-guide/help-guide

There are several ways the CBR could be tested, to see how well it functions. One way of testing could be show how calculation is made, and prove that it works, as will be shown below.

Case base:

<table>
<thead>
<tr>
<th>Case</th>
<th>Age</th>
<th>Weight</th>
<th>Gender</th>
<th>Health State</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>38</td>
<td>59</td>
<td>Female</td>
<td>Bad</td>
</tr>
<tr>
<td>7</td>
<td>48</td>
<td>90</td>
<td>Male</td>
<td>Good</td>
</tr>
<tr>
<td>CBR Case 1</td>
<td>48</td>
<td>90</td>
<td>Male</td>
<td>Normal</td>
</tr>
<tr>
<td>MAX AGE</td>
<td>MIN AGE</td>
<td>MAX WEIGHT</td>
<td>MIN WEIGHT</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>25</td>
<td>90</td>
<td>59</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: The Second case base

1. The algorithm reads the information about the current CBR case.
2. It reads all other cases in the case base.
3. It calculates similarity between the current CBR case and the other cases in the case base, one at a time. The calculation below shows how it is done. For the numerical features(Age, Weight) it finds the maximum and minimum. The Euclidean "multi-dimensional" distance(Formula 1)formula is used for numerical features, below is shown how the calculation is done.
For the Age feature, between CBR Case and case 4:

\[ 1 - \frac{\text{abs}(48-38)}{49-25} = 1 - \frac{10}{24} \approx 0.58 \text{(distance)}. \]

According to the expert defined similarity matrix, a distance of age( >0.5 && <=0.6), gives a similarity value of 0.6*0.16=0.096(multiplied with the normalized weight of the feature).

For the Weight feature, between CBR Case and case 4:

\[ 1 - \frac{\text{abs}(90-59)}{(90-59)} = 1 - \frac{31}{31} = 0 \text{(distance)}. \]

According to the expert defined similarity matrix, a distance of weight( >=0 && <=0.1), gives a similarity value of 0.1*0.24=0.024(multiplied with the normalized weight of the feature).

For the Gender feature, between CBR Case and case 4:

According to the expert defined similarity matrix, the same Gender give a value of 1 else 0.5. So in this case, the CBR case is a Male and case 4 Female, which gives a value of 0.5*0.28 = 0.14(multiplied with the normalized weight of the feature).

For the Health State feature, between CBR Case and case 4:

According to the expert defined similarity matrix, the similarity for Health State is: 0.7*0.32 = 0.224(multiplied with the normalized weight of the feature).

The global similarity between the CBR case and case 4, is the sum of all weighted similarity values of all features that was recently compared, which is:

0.096+0.024+0.14+0.224 = 0.484 * 100 = 48.4 %.

In the same way, it calculates the similarity between the current CBR case and case 7, which is: 90.4 %.

As seen from this figure, since case 7 was pressed on, it will also show the next case below it, which is 3. However, lets take a look at another calculation, where the calculation is done between the CBR case 1, and case 4. See the figure below please.
As you can see, the comparison and calculation is correct. The resulting similarity is as expected, precisely the same resulting similarity that was calculated, in the example above. Since case 8 was pressed on, it will show the comparison between the CBR case 1 and case 8 but also comparement between case 1 and 4.

However, this was one way to show that the CBR functions well. As seen, one good thing is that, the physician can change the weights for the features, if he/she wants to further make use of his/her expertise. But it's also important to note here that, the physician is the one that makes the final decision, and not the CBR. Another way to possibly test it, is trying to change the weights, and see how it would behave, but however, since the weights are already included in the examples and calculations above, simply the test was not made.
The GUI
This is how the user interface looks like, which has been also shown earlier in this thesis report, in the example.

![Figure 10: the main GUI](image)

In this project or thesis, a button has been created for activating the CBR. Once the user have an case-id specified(or added a new case, with the features specified), the user is able to run the CBR tool.

When the CBR is activated, it will automatically calculate and show the user the results in a new window. In the new windows, there are two tables. The first one to the left is the ranked-order list and the other one to the right is the comparison table. To the see this two tables, look at figure 3 please.
Some of the data that could be helpful for the physician in making his/her decision could be as follows:

- Global Weighted Similarity – Similarity with respect to weights
- Weights - Allowing expertise.
- Outcome – The intensity of pain experienced, both direct (24 hours) and week.
- Outcome Notes - Extra information about the outcome for the specific case.
- Solution – Action that was performed, or anaesthetics that was given.
- Description of the solution(s) that was given.
- Features and other useful information.

All these additional information that is provided to the user, is to help the user in making a good decision. Even though a specific case might have a high similarity value, it doesn't always mean that is the best solution. However, the physician might consider first taking a closer look at the outcome notes and description of the solutions, before making any hasty decision. If the physician also thinks it's ideal, he/she is able to also make further use of his/her expertise by setting the weights to the features accordingly, in order to aid him/her making a more accurate decision.

**The GUI Implementation**

How the GUI is structured and communication is made with the database, can more easily be understood visually as shown below:

![Diagram of GUI Implementation](image)

**Figure 11: the user GUI Implementation**

The blue arrows, basically indicates that these tables (database tables) are being connected together, via one or another method. The green arrow indicates the data flow (inserting, editing, retrieving or updating). This is a very simple structure though, and easy to understand. It's possible to use several strategies when planning how the communication should work, and how the data shall flow.
For example, insertion of the data could have been made separately into each table, without connecting them together, but somehow, it was thought that, it's maybe more efficient to combine all tables together in this case, and then, insert the data. It's the same principle when retrieving data from the database.

Suppose there is a need to retrieve information about a specific case with a specific id, then, during the retrieval process, a connection will be established to the database and all the database tables will be connected together, making it easy to issue one SQL command and retrieve all the desired information from all the tables, with only one SQL command, rather than doing an query three times separately.

However, the information that was retrieved is then presented to the physician in the graphical user interface. As mentioned, the physician is able to insert, edit, update or add cases, with the help of the graphical user interface. Everything happens in the background, the user won't be able to see the query commands and everything else, except the data that was retrieved. The information that is retrieved from the database, is amongst other information, the feature, solution and outcome data relative to the selected case id.

### The Database Structure

This is how the database structure looks like, in which the data is stored:

![Database Structure Diagram](image)

**Figure 12:** The basic database structure

The arrows indicate that the following information below is additional information. In this project, it was not a requirement to have "relations", instead, the tables was "inner joined" partly achieving the same functionality, as if there was relations.

Usually, it's ideal to think of the relational structure, and make sure to avoid redundant data, since redundant data can cause different information to be inserted into the same place, where it actually should be the same. There are also other factors to think of as well, but these factors won't be covered in details for now. In the following section, let's focus on how the graphical interface for the CBR is structured.
The User Guide

Some of the main reasons for why an user-guide is important to have is as following:

- Describe and give the users, and introduction to the project, and how they can benefit from making use of it.
- Describe the functionality of the user interface, to make sure that even an first-time user easily understands how to use the tool.
- Describe and make sure that, the user understands that the tool can only help the user, and the final decision is made by the user and not the tool, so the user knows that they are not basically taking any risks by using the tool.

The figure below, is a simple picture of how the user guide looks like, giving sample examples and explanations, although its not fully yet finished, improvements are still needed, but not only in the user-guide, but also to the overall CBR functionality.

![Figure 13: The User Guide](image-url)
As you can see, it has a very simple structure. The user guide, is a window that pops up, whenever the user press on the help link in the user interface. The fields and checkboxes seen in the user-guide, is basically there together with some text, to explain what certain fields are used for.

Still, information and background about the tool have not been added, and also other related information, which is actually important, this needs still to be added, except for that, possibly the user-guide should also be improved. However, the goal was not creating an very professional guide, only a very basic to start with, and later on be improved.
**The Similarity Matrix**

For the numeric features, the distance between the two features is calculated using Euclidean "multi-dimensional" distance formula. After calculating the distance, depending on what the distance is between the two features, an similarity value is retrieved accordingly.

For symbolic features, similarity is evaluated without the help of the distance formula. For example, calculating similarity between two "gender" features, can be easily done by checking if these two features are both "male" or "female". If the two features are equal, the similarity is 1, otherwise, if they are not equal, having "female" and "male" or "male" and "female", the similarity is 0.5. An similarity matrix could be defined by an expert, having the similarity values of both symbolic and non-symbolic features. See below please.

<table>
<thead>
<tr>
<th>Similarity for Age</th>
<th>Similarity for Weight</th>
<th>Similarity for Gender</th>
<th>Similarity for Health State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>Sim</td>
<td>Distance</td>
<td>Sim</td>
</tr>
<tr>
<td>==0 and &lt;=0.1</td>
<td>0.1</td>
<td>==0 and &lt;=0.1</td>
<td>male</td>
</tr>
<tr>
<td>&gt;0.1 and &lt;=0.2</td>
<td>0.2</td>
<td>&gt;0.1 and &lt;=0.2</td>
<td>male</td>
</tr>
<tr>
<td>&gt;0.2 and &lt;=0.3</td>
<td>0.3</td>
<td>&gt;0.2 and &lt;=0.3</td>
<td>female</td>
</tr>
<tr>
<td>&gt;0.3 and &lt;=0.4</td>
<td>0.4</td>
<td>&gt;0.3 and &lt;=0.4</td>
<td>female</td>
</tr>
<tr>
<td>&gt;0.4 and &lt;=0.5</td>
<td>0.5</td>
<td>&gt;0.4 and &lt;=0.5</td>
<td>male</td>
</tr>
<tr>
<td>&gt;0.5 and &lt;=0.6</td>
<td>0.6</td>
<td>&gt;0.5 and &lt;=0.6</td>
<td>male</td>
</tr>
<tr>
<td>&gt;0.6 and &lt;=0.7</td>
<td>0.7</td>
<td>&gt;0.6 and &lt;=0.7</td>
<td>male</td>
</tr>
<tr>
<td>&gt;0.7 and &lt;=0.8</td>
<td>0.8</td>
<td>&gt;0.7 and &lt;=0.8</td>
<td>male</td>
</tr>
<tr>
<td>&gt;0.8 and &lt;=0.9</td>
<td>0.9</td>
<td>&gt;0.8 and &lt;=0.9</td>
<td>male</td>
</tr>
<tr>
<td>&gt;0.9 and &lt;=1</td>
<td>1</td>
<td>&gt;0.9 and &lt;=1</td>
<td>male</td>
</tr>
<tr>
<td>&gt;1</td>
<td>1</td>
<td>&gt;1</td>
<td>male</td>
</tr>
</tbody>
</table>

*Table 3: The Similarity Matrix*
Future Work

Well, there are a few things that would be ideal to improve. There have not been anyone to get feedback from, which would have been valuable to us. For example, reaching any phycisian that could comment and give us feedback, would have been nice. Anyway, fake data was used in this thesis or project. Ther are indeed a lot of things that could be improved, some of them are as following:

- Improve the user interface.
- Much more improve the user-guide.
- Improve the CBR interface, by adding extra information and functionality.
- Make it possible, to change similarity values, for everthing, such as distance and comparement between symbolic features. This could be done, by having a similarity matrix, as has been seen earlier in this project, that is web-based, and can easily be configured via the web.
- So far, it's possible to change the weights of the features, but still it's ideal to make it possible to have a direct effect, so the user don't have to press the update button, every time, and restart the application.
- The user is able to see some basic information about the cases listed, but however, more information is still needed to be shown.
- Realistic evaluation with real phycisian and doctors, saying their opinion.
Appendix B

Screenshots and the Large Scale Work

Appendix B, consists of many figures of how the GUI looks like, for the larger scale work that has been made. There will be several pictures, see below please. Sorry for the bad quality, zooming in might help a little.

---

### Figure 14: Table Screening

<table>
<thead>
<tr>
<th>S1 INCLUSION CRITERA</th>
<th>YES</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of data collection is POD1 AND patient is 6 hrs (minimum) in the ward.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>End of surgery: Date: 10-03-02 Time: 10:06:21 POD1?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time back in ward:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patient is conscious age or over</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patient has not undergone repeat surgery (same organ) during current hospitalization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patient is in ward, available for interview</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patient has given his consent (or consent) to participate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patient is able to fill in questionnaire on his/her own, unaided (include also patients who need help filling in the questionnaire for technical reasons).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason why patient is unable to fill in the questionnaire on his/her own (select):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. too ill</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. cannot read/write language of questionnaire</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. cognitively impaired</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. other (describe)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Figure 15: Demographic Information

<table>
<thead>
<tr>
<th>DEMOGRAPHIC INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1 Gender</td>
</tr>
<tr>
<td>Male</td>
</tr>
<tr>
<td>Female</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D3 Weight</th>
<th>D4 Country of birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>102 Kg</td>
<td>207</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D5 Nationality</th>
<th>MEDICAL HISTORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>176</td>
<td>D8 Comorbidities (multiple answers possible):</td>
</tr>
<tr>
<td></td>
<td>- Psychiatric</td>
</tr>
<tr>
<td></td>
<td>- Cardiovacular disease</td>
</tr>
<tr>
<td></td>
<td>- Hepatic disease</td>
</tr>
<tr>
<td></td>
<td>- Renal disease</td>
</tr>
<tr>
<td></td>
<td>- Substance Use Disorder</td>
</tr>
<tr>
<td></td>
<td>- Gastrointestinal</td>
</tr>
<tr>
<td></td>
<td>- Cancer</td>
</tr>
<tr>
<td></td>
<td>- Neurovascular</td>
</tr>
<tr>
<td></td>
<td>- Central nervous system (inc cognitive impairment)</td>
</tr>
<tr>
<td></td>
<td>- Respiratory (incl obstructive sleep apnea)</td>
</tr>
<tr>
<td></td>
<td>- None recorded from this list</td>
</tr>
<tr>
<td></td>
<td>- Not possible to obtain information</td>
</tr>
</tbody>
</table>
**Figure 16: Medical History**

<table>
<thead>
<tr>
<th>Medical History</th>
<th>Immediate release (PO &amp; others)</th>
<th>Controlled release (PO &amp; others)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tramadol</strong></td>
<td></td>
<td>mg/day</td>
</tr>
<tr>
<td><strong>Fentanyl</strong></td>
<td></td>
<td>mg/day</td>
</tr>
<tr>
<td><strong>Codeine</strong></td>
<td></td>
<td>mg/day</td>
</tr>
<tr>
<td><strong>Hydromorphone</strong></td>
<td></td>
<td>mg/day</td>
</tr>
<tr>
<td><strong>Morphine</strong></td>
<td></td>
<td>mg/day</td>
</tr>
<tr>
<td><strong>Oxycodone</strong></td>
<td></td>
<td>mg/day</td>
</tr>
<tr>
<td><strong>Hydromorphone</strong></td>
<td></td>
<td>mg/day</td>
</tr>
<tr>
<td><strong>Buprenorphine</strong></td>
<td></td>
<td>mg/day</td>
</tr>
</tbody>
</table>

**Figure 17: Surgical Procedures**

<table>
<thead>
<tr>
<th>Surgical Procedure(s)</th>
<th>Code</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1434</td>
<td>3</td>
<td>Partial gastrectomy with bypass gastrostomy</td>
</tr>
<tr>
<td>34-41</td>
<td>5</td>
<td>Flexible endoscopy</td>
</tr>
</tbody>
</table>
## Figure 18: pre-medication

### M1 Sedatives (pre-medication)
- **Midodrine**: 5 mg/day
- **Diazepam**: mg/day
- **Promethazine**: mg/day
- **Haloperidol**: mg/day
- **Diklakamchlorzepat**: mg/day
- **Lorazepam**: mg/day

**Other specify**: [ ]

**Other specify**: [ ]

### M2 Non-opioids (pre-medication)
<table>
<thead>
<tr>
<th>Drug</th>
<th>p.o</th>
<th>i.v.</th>
<th>i.m</th>
<th>supp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paracetamol/Acetaminophen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metamizol</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ibuprofen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diclofenac</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ketoprofen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ketorolac</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Naproxen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Celecoxib</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parecoxib</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salmisicib</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gabapentin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pregabalin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neofam</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### M3 Opioids (pre-medication)
- **Imidacodol**
- **Tetradosodol**
- **Codeine**
- **Dialin (very Naloxon)**
- **Fenfluramine (meporeidone)**
- **Pentazocine**
- **Morphine**
- **Peridox**
- **Oxycodone**
- **Oxycodone with Naloxon**
- **Fentanyl**: µg/hr tran-infusion

**Other specify**: [ ]

**Other specify**: [ ]

### Immediate Release (PO & Other)
<table>
<thead>
<tr>
<th>Drug</th>
<th>p.o</th>
<th>i.v.</th>
<th>i.m</th>
<th>supp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tramadol</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tapentadol</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oxycodone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oxycodone (with Naloxon)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fentanyl (transdermal)</td>
<td>µg/hr</td>
<td>µg/hr</td>
<td>µg/hr</td>
<td>µg/hr</td>
</tr>
</tbody>
</table>

**Other specify**: [ ]

**Other specify**: [ ]
**Figure 19:** intra-operative treatment
Figure 22: recovery room and ward
Figure 23: ward
Figure 24: ward and questionarie
**Figure 25: questions p1-p6**

<table>
<thead>
<tr>
<th>Questions</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1. Do you think your pain during the last 24 hours was severe?</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Q2. Do you think your pain during the last 24 hours was moderate?</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Q3. Do you think your pain during the last 24 hours was mild?</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Q4. Do you think your pain during the last 24 hours was minimal?</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Q5. Do you think your pain during the last 24 hours was none?</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Q6. What was your average pain during the last 24 hours?</td>
<td>1-10 scale</td>
</tr>
<tr>
<td>Q7. What was your average pain during the last 24 hours?</td>
<td>1-10 scale</td>
</tr>
<tr>
<td>Q8. What was your average pain during the last 24 hours?</td>
<td>1-10 scale</td>
</tr>
<tr>
<td>Q9. What was your average pain during the last 24 hours?</td>
<td>1-10 scale</td>
</tr>
<tr>
<td>Q10. What was your average pain during the last 24 hours?</td>
<td>1-10 scale</td>
</tr>
</tbody>
</table>

*Note: Questions p1-p6 are related to pain assessment.*
**Figure 26:** questions p7-p11

**Figure 27:** questions p12-p17
Well, there are many pictures, but however, it was thought that it was necessary to show how the entire User Interface looks like. Sorry for the bad quality of the pictures, but at least, you can see how the tables look like, which is more important rather than reading the text. Now, let's take a look at how the CBR interface looks like, in the following section. Please, note that, the entire comparement table will not be shown, since it's very long, so it's ideal here to show how the top of the table looks like and the end of the table, where the similarity values are shown. See below please.

**THE CBR INTERFACE**

**CBR MOST SIMILAR CASES**

<table>
<thead>
<tr>
<th>Case</th>
<th>Percent</th>
<th>Proced.1</th>
<th>Proced.2</th>
<th>Proced.3</th>
<th>Proced.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>626</td>
<td>67.5%</td>
<td>other proctopexy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>617</td>
<td>67.2%</td>
<td>laparoscopic cholecys</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1166</td>
<td>67.1%</td>
<td>ACHILIAN TENOPLAST</td>
<td>ATONIC ULCER EXCIS</td>
<td>other and unspecified</td>
<td>other and open repair</td>
</tr>
<tr>
<td>612</td>
<td>67.1%</td>
<td>exploratory laparotom</td>
<td>other and unspecified</td>
<td>other and open repair</td>
<td></td>
</tr>
<tr>
<td>625</td>
<td>66.9%</td>
<td>open and other resect</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>622</td>
<td>66.8%</td>
<td>radical excision of the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>624</td>
<td>66.7%</td>
<td>Ligation and stripping</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>620</td>
<td>66.2%</td>
<td>Excision of aneurysm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>615</td>
<td>66.0%</td>
<td>Laparoscopic gastropl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1165</td>
<td>62.4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>614</td>
<td>53.4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>613</td>
<td>53.4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>611</td>
<td>53.3%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>607</td>
<td>53.2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>605</td>
<td>53.2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>608</td>
<td>53.0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>610</td>
<td>53.0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>606</td>
<td>52.9%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>609</td>
<td>52.8%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>619</td>
<td>50.6%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 28: ranked order list*
<table>
<thead>
<tr>
<th>Attributes</th>
<th>Local Weight</th>
<th>Normalized Local Weight</th>
<th>Current Case Sim</th>
<th>Stored Case Sim</th>
<th>Sim</th>
<th>Weighted Sim</th>
<th>Stored Case Sim</th>
<th>Weighted Sim</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1.ROC1.UHR.1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S1.OCNSNT.AGE.1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S1.RNSAMESUR.1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S1_AVAIL</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S1_ASGENT</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S1_FILL_UNAD.1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S1_FILL_UNAD.11</td>
<td>0.016</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>S1_FILL_UNAD.1</td>
<td>0.016</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>S2.SLP.SED.2</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D1_GENDER.3</td>
<td>0.049</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.049</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>D2_BIRTHYR.4</td>
<td>0.056</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.056</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D2_BIRTHYR.5</td>
<td>0.062</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.062</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D4_AIRTH.COU1.0</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D4_NATIONAL_L.2</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D4_PRTZ.LANG.3</td>
<td>0.049</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.049</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_PSY.4</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_RESP.4</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_MOTLIST.4</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_NONF.4</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_CARDIO.4</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_Renal.4</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_GASTRO.4</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_NEUROMUS.4</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_PREGANNCE.5</td>
<td>0.022</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.022</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D8_LACRAT.5</td>
<td>0.022</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.022</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D10_ACRONIC.1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D10_ACRONIC.11</td>
<td>0.016</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.016</td>
<td>-1</td>
<td>0.1</td>
</tr>
<tr>
<td>D24_OPOD.2</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D24_TRMRL.2</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D24_TRMRL.22</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 29**: Comparison table start
**Figure 30:** comparement table end

| LOPR_LANA | 0.098 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.068 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.000 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | 1 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.068 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.098 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.098 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | 1 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.068 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |
| LOPR_LANA | 0.066 | D | 0.000 | D | 0.000 |

**GLOBAL SIM:** 67.5% **GLOBAL SIM:** 67.2%
Appendix C

Introduction To PHP MySQL

Here, some of the very basic knowledge will be covered, that is required to get started with the coding part of the thesis. More focus will be on PHP MySQL rather than html. First of, some SQL basics.

First of, the tables was created. When opening the phpMyAdmin page, it's possible to use the "SQL prompt" from which sql commands can be run. From the beginning, nothing is selected, the database is empty, so firstly, it's necessary to creat the database, via the "SQL prompt". This is how its done.

Step1:
**command:** CREATE DATABASE hospital;

```
CREATE DATABASE hospital
```

Now to step 2, creating the tables.
Before creating the tables, the created database needs to be selected, by the command:

**command:** USE hospital

```
USE hospital
```

Now to step 2, creating the tables. This is done by the commands:

Step2:
**command:**
CREATE TABLE patients (P_ID int NOT NULL AUTO_INCREMENT PRIMARY KEY, Gender char(1) NOT NULL, Age int NOT NULL, Weight int NOT NULL, Health_State int NOT NULL, Handicap char(1) NOT NULL)

First table patients created. Now the same with the other two.

**command:**
CREATE TABLE patients_solution (P_ID int NOT NULL AUTO_INCREMENT PRIMARY KEY, Name varchar(255) NOT NULL, Solution varchar(255) NOT NULL, Solution_Descr varchar(255), Solution_Cost int)

Second table created, now to the last one.
command:
CREATE TABLE patients_outcome (P_ID int NOT NULL AUTO_INCREMENT PRIMARY KEY, Outc_Direct int, Outc_Week int, Outc_Notes varchar(255))

And tada, last table created.

Now to get descriptions of the tables that was recently created. So, this is how it's done.

command: DESCRIBE PATIENTS
As you can see, good information about the specified table is given.
Now for the other two tables.
Next, the description of table: patients_solution.

**command:** DESCRIBE patients_solution

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_ID</td>
<td>int(11)</td>
<td>NO</td>
<td>PRI</td>
<td>NULL</td>
<td>auto_increment</td>
</tr>
<tr>
<td>Name</td>
<td>varchar(255)</td>
<td>NO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solution</td>
<td>varchar(255)</td>
<td>NO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solution_Descr</td>
<td>varchar(255)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>Solution_Cost</td>
<td>int(11)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>

And here is the description for patients_solution. And now for the last one, The table: patients_outcome.

**command:** DESCRIBE patients_outcome

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_ID</td>
<td>int(11)</td>
<td>NO</td>
<td>PRI</td>
<td>NULL</td>
<td>auto_increment</td>
</tr>
<tr>
<td>Outc_Direct</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outc_Week</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outc_Notes</td>
<td>varchar(255)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
And, this is how table description works. Next, will be how the basic database operations work, which is also essential for the coding part. So, how does it work when inserting records into the database? Let's try to insert a record into the patients table.

**command:**

```
INSERT INTO patients (Gender, Age, Weight, Health_State, Handicap)
VALUES ('F', 23, 56, 2, 'N')
```

Now that a record has been added, let's try to retrieve all possible data from the table patients.

**command:** SELECT * FROM patients

```
<table>
<thead>
<tr>
<th>P_ID</th>
<th>Gender</th>
<th>Age</th>
<th>Weight</th>
<th>Health_State</th>
<th>Handicap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F</td>
<td>23</td>
<td>56</td>
<td>2</td>
<td>N</td>
</tr>
</tbody>
</table>
```
Everything from the table patients is now given. And it's now possible to see that, the id is currently '1' for the first record. Note that the P_ID is automatically increased for each record being added. Two more records has been added to the table patients. This is how the table now looks like, and then, lets see how to select specific records from the table. This is how the current table looks like, after inserting two more records.

**command:** SELECT * FROM patients

<table>
<thead>
<tr>
<th>P_ID</th>
<th>Gender</th>
<th>Age</th>
<th>Weight</th>
<th>Health_State</th>
<th>Handicap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F</td>
<td>23</td>
<td>56</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>M</td>
<td>35</td>
<td>67</td>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>F</td>
<td>56</td>
<td>55</td>
<td>1</td>
<td>N</td>
</tr>
</tbody>
</table>

Ok, lets only select those records where the handicap entry is 'Y'. This is how this can be done.

**command:** SELECT * FROM patients WHERE Handicap = 'Y'

<table>
<thead>
<tr>
<th>P_ID</th>
<th>Gender</th>
<th>Age</th>
<th>Weight</th>
<th>Health_State</th>
<th>Handicap</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>M</td>
<td>35</td>
<td>67</td>
<td>3</td>
<td>Y</td>
</tr>
</tbody>
</table>
As you can see, it's possible to select specific data. It's possible to combine it as you want, selecting specific P_ID, or age and so on. Another example, let's select everything where age is larger than 30.

**command:**

SELECT * FROM patients WHERE Age > 30

One more example, somewhat more complex can be, selecting everything from where Age is larger than 30 and where Gender is Male(M) and where Handicap is Yes(Y).

**command:**

SELECT * FROM patients WHERE Age > 30 AND Gender='M' AND Handicap='Y'
As you see, if there are any records according to the criteria, it will show you that. It's possible to do many things, including the OR command and so on. This is just a few examples. Next, let's see how the "INNER JOIN" can be used, to combine or connect different tables. But before going further to the "INNER JOIN" command, let's take a look at how to update or edit a table. Before updating or changing anything in the table, let's take a look at how the table currently looks like.

**command:** SELECT * FROM patients

Ok, let's select those records where the handicap entry is 'Y'. This is how this can be made.

**command:** SELECT * FROM patients WHERE Handicap = 'Y'

---

Arash Kianifar - Bachelor Thesis – B Level
As you can see, it's possible to select specific data. It's possible to combine it as you want, selecting specific P_ID, or age and so on. Another example, lets select everything where age is larger than 30.

**command:** SELECT * FROM patients WHERE Age > 30

One more example, somewhat more complex can be, selecting everything from where Age is larger than 30 and where Gender is Male(M) and where Handicap is Yes(Y).

**command:**
SELECT * FROM patients WHERE Age > 30 AND Gender='M' AND Handicap='Y'
As you see, if there are any records according to the criteria, it will show you that. It's possible to do many things, including the OR command and so on. This is just a few examples. In the next section, let's see how the "INNER JOIN" can be used, to combine different tables. But before going to the next section, let's take a look at how to update or edit a table.

Before updating or changing anything in the table, let's take a look at how the table currently looks like.

**command:** SELECT * FROM patients

Now, let's see how to change the age for which the P_ID is 1, from 23 to 20, and also change the weight from 56 to 60 for which the P_ID is 2.

**command:**
UPDATE patients SET Age=20, Weight=60 WHERE P_ID=1; UPDATE patients SET Weight=60 WHERE P_ID =2

Note that there was two SQL statements separated by ";", to achieve the desired result. Also, this is another useful command, it can be used to change/edit a simple value, instead of re-entering all the data again. For now, some useful commands have been covered, which has been used. Now, let's move on to the "INNER JOIN".

How about retrieving data from three different tables, and then output the data according to the desired criteria? One useful command to do this, is using the "INNER JOIN" command, to combine one or several tables. Let's see an example of doing that.
First of, lets try to combine two tables, "patients" and "patients_solution", and then, retrieve all possible data from them. First, lets take a look at them.

**command:** SELECT * FROM patients_solution

```
Visar rader 0 - 2 (3 totalt, Frågan tog 0.0002 sek)
```

**command:** SELECT * FROM patients

```
Visar rader 0 - 2 (3 totalt, Frågan tog 0.0002 sek)
```
Ok, now that the tables have been seen, lets combine these two, and then read all data possible from it when combined. But before doing that, it's good to consider how they should "match", because the "INNER JOIN" command, or statement requires you to specify the matching criteria. Below is an example of how to combine these two tables.

**command:**
SELECT * FROM patients INNER JOIN patients_solution ON patients.P_ID=patients_solution.P_ID

So, the tables are matched using the P_ID column, so all rows and so on, are matched, as you can see above. So basically, two tables has been combined, and all data has been retrieved from them, but it is also possible, to retrieve data from specific fields, as shown earlier. Note that the "matching criteria" comes after the ON statement. How is then three tables combined? It's very much the same principle, but before doing that, lets take a look at the patients_outcome table.

**command:** SELECT * FROM patients_outcome
Ok, as you see, this is how the table look likes. Consider this table, and the two tables that was earlier combined together. So three different tables will be combined or connected. Below, is an example of how to do that.

**command:**

```sql
SELECT * FROM patients INNER JOIN patients_solution ON patients.P_ID = patients_solution.P_ID INNER JOIN patients_outcome ON patients.P_ID = patients_outcome.P_ID
```

And tada! This is the result, three tables combined, note that, it's also possible to add some additional specification on the command, adding something like WHERE P_ID=1. Probably it would then show the entire full first row in the three tables combined.

However, this was a very basic short tutorial on how to use SQL queries, to get started with the coding part, which is done in PHP. However, some basic SQL knowledge is required in order to be able to code in PHP MySQL, since making the queries in PHP requires some basic SQL knowledge. See the example below please, where the php code actually connects to the database with the "mysql_connect()" function and selects a datables with the "mysql_select_db()" function, and then, the query is made and the result is printed with the "echo" function. At the end, the connection to the database is closed with the "mysql_close()" function.
As you can see, there is both PHP and SQL knowledge required. Hope at least this small tutorial was helpful in learning some basics.
Appendix D

Other Links


[23] SQL Constraints http://www.w3schools.com/SQL/sql_constraints.asp


[26] SQL NOT NULL Constraint http://www.w3schools.com/SQL/sql_notnull.asp

Arash Kianifar -64- Bachelor Thesis – B Level
[27] The CREATE DATABASE Statement
http://www.w3schools.com/SQL/sql_create_db.asp

[28] SQL INNER JOIN Keyword
http://www.w3schools.com/SQL/sql_join_inner.asp

[29] SQL CREATE TABLE Statement
http://www.w3schools.com/SQL/sql_create_table.asp

[30] SQL UPDATE Statement "The UPDATE Statement"
http://www.w3schools.com/SQL/sql_update.asp

[31] SQL INSERT INTO Statement "The INSERT INTO Statement"
http://www.w3schools.com/SQL/sql_insert.asp

[32] SQL DELETE Statement "The DELETE Statement"
http://www.w3schools.com/SQL/sql_delete.asp

[33] SQL SELECT Statement "The SQL SELECT Statement"
http://www.w3schools.com/SQL/sql_select.asp

[34] SQL FOREIGN KEY Constraint
http://www.w3schools.com/SQL/sql_foreignkey.asp

[35] SQL SELECT DISTINCT Statement "The SQL SELECT DISTINCT Statement"
http://www.w3schools.com/SQL/sql_distinct.asp

[36] 3.1.Connecting to and Disconnecting from the Server

[37] 3.3.Creating and Using a Database

[38] 3.3.1.Creating and Selecting a Database

[39] 3.3.3.Loading Data into a Table

[40] 3.3.4.Retrieving Information from a Table

[41] 3.3.4.7.Pattern Matching

[42] 3.3.4.3.Selecting Particular Columns
[43] 3.3.4.1. Selecting All Data

[44] 3.3.4.4. Sorting Rows

[45] 3.3.4.8. Counting Rows

[46] 3.3.4.9. Using More Than one Table

[47] Primära och främmande nycklar i MySQL
http://www.republic.nu/utveckling/primara-och-frammande-nycklar-i-mysql/

[48] AJAX ASP/PHP Example
http://www.w3schools.com/ajax/ajax_aspphp.asp