Master thesis report

Web-based AHP and CPC evaluation system

YEAR 2008/2009

Team members:
Gianluigi Ciamбриello
Paolo D’Amelio

Supervisor and examiner:
Peter Wallin
Stig Larsson
Abstract

Making a decision today is quite important, especially in companies, where the consequences can bring benefits or costs. Sometimes the decisions to be made are complex and we need to evaluate many criterions that can involve several factors. In those cases, it is not easy to have a good overview of all choices and for this reason it is important to have a system that helps us in making complex decisions, when it is not possible to evaluate all the alternatives manually. This thesis describes different algorithms and tools that can aid in the process of making decisions. Based on the review of existing tools a new web based application is proposed. The new tool, Decision Maker, uses both the full Analytical Hierarchical Process approach and Chainwise Paired Comparisons method to evaluate different alternatives.
Acknowledgements

For the development of this Master Thesis, first of all we should be grateful to Peter Wallin to his support throughout the project. A special thanks to Stig Larsson who helped us with report and general improving of system. Even more important we want to remember our families, special people who supported us in our entire university carrier, especially in the last year abroad, in this wonderful land.

Grazie,

Paolo e Gianluigi
Table of Contents

1 INTRODUCTION .................................................................................................................. 7
2 BACKGROUND ..................................................................................................................... 9
3 DECISION MAKING THROUGH ALGORITHMS ............................................................... 10
   3.1 ANALYTIC HIERARCHY PROCESS ........................................................................... 10
   3.2 CHAINWISE PAIRED COMPARISON ....................................................................... 12
   3.3 OTHER ALGORITHMS ............................................................................................... 14
4 RELATED WORK ............................................................................................................... 16
   4.1 OTHER TOOLS AVAILABLE ..................................................................................... 16
5 METHOD ............................................................................................................................. 22
   5.1 WEB SITE BASED: PROS AND CONS .................................................................... 22
   5.2 TECHNOLOGIES INVOLVED .................................................................................. 23
   5.3 ORGANIZATION – FILES AND FOLDERS ................................................................. 24
   5.4 WEB SITE OVERVIEW ............................................................................................. 25
   5.5 WEBSITE DESCRIPTION .......................................................................................... 28
6 SOFTWARE ENGINEERING PROCESS .......................................................................... 36
   6.1 PROJECT PLAN .......................................................................................................... 36
   6.2 REQUIREMENTS ......................................................................................................... 38
   6.3 DESIGN ...................................................................................................................... 39
   6.4 DEVELOPMENT .......................................................................................................... 40
   6.5 VERIFICATION AND TEST ...................................................................................... 41
7 RESULTS ............................................................................................................................ 42
   7.1 AHP VS. CPC – CASE STUDY ................................................................................ 42
8 DISCUSSION ....................................................................................................................... 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>ARE AHP AND CPC USEFUL?</td>
<td>47</td>
</tr>
<tr>
<td>8.2</td>
<td>IMPLEMENTATION AND ALGORITHMIC CHOICES</td>
<td>48</td>
</tr>
<tr>
<td>8.3</td>
<td>PROBLEMS</td>
<td>49</td>
</tr>
<tr>
<td>9</td>
<td>CONCLUSIONS</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>FUTURE WORK</td>
<td>52</td>
</tr>
<tr>
<td>11</td>
<td>APPENDICES</td>
<td>53</td>
</tr>
<tr>
<td>11.1</td>
<td>APPENDIX A – REQUIREMENTS</td>
<td>53</td>
</tr>
<tr>
<td>11.2</td>
<td>APPENDIX B – DESIGN</td>
<td>57</td>
</tr>
<tr>
<td>11.3</td>
<td>APPENDIX C – DATABASE</td>
<td>65</td>
</tr>
<tr>
<td>11.4</td>
<td>APPENDIX D – SOURCE CODE</td>
<td>66</td>
</tr>
</tbody>
</table>
Figure Index

Figure 1 AHP - Comparison values ................................................................. 11
Figure 2 AHP - Table example ................................................................. 11
Figure 3 Comparison with other tools ....................................................... 22
Figure 4 Website Map .............................................................................. 26
Figure 5 Web site home page ................................................................. 28
Figure 6 Admin main page ........................................................................ 29
Figure 7 Admin adding of new survey ...................................................... 30
Figure 8 Admin showing survey results ................................................... 31
Figure 9 Admin add administrator ........................................................... 32
Figure 10 Admin categories management ............................................... 33
Figure 11 User first page ........................................................................... 34
Figure 12 User insert values...................................................................... 35
Figure 13 Working hours planned before starting ..................................... 37
Figure 14 Effective working hours ............................................................ 37
Figure 15 Example Inputs for AHP case study ......................................... 43
Figure 16 Example Inputs for CPC case study ......................................... 45
Figure 17 Result comparison AHP vs. CPC .............................................. 46
Figure 18 Functional Requirements .......................................................... 55
Figure 19 Non Functional Requirements .................................................. 56
Figure 20 Use case administrator side ...................................................... 57
Figure 21 Use case user side ..................................................................... 60
Figure 22 Sequence diagram administrator side ....................................... 62
Figure 23 Sequence diagram user side ...................................................... 64
Figure 24 Entity-relationship diagram ...................................................... 65
Figure 25 CPC - Array preparation and function call (user_survey_last.php) ........................................................................ 66
Figure 26 CPC – Function to calculate CPC results (cpc_core.php) .......... 68
Figure 27 AHP – Array preparation and function call (user_survey_last.php) ........................................................................ 68
Figure 28 AHP – Function to calculate AHP results (ahp_core.php) ........ 69
Figure 29 AHP – Function to calculate the opposite values (ahp_core.php) ........................................................................ 69
Figure 30 Query to add a new survey (administrator.php) ...................... 70
1 Introduction

Everyone, in his workaday life has to make decisions, both simple and complicated. A decision is the output of a process in which two or more alternatives have to be considered. Some decision can be taken very quickly, because they are easy or only one alternative is acceptable, but often there is more than one factor that influences the choice, and the results are hard to predict.

The important thing about decision making is that decision elements have different weight, sometimes one of them is so important that it is the only one to consider. Moreover, for each element it is necessary to evaluate the benefits it brings, the new opportunities it can create, the risks related and the costs. All these trade-offs and evaluation points are too much for the human brain to handle in a rational way.

In these cases, people use their instincts or rationality to obtain the best results, asking friends for advice, collecting different opinions or accepting decisions taken by others that might be more knowledgeable in the area. Sometimes there are too many different parameters to evaluate so they prefer to make random choice like flipping a coin, throwing a dice or rely on astrology and tarot cards.

When decisions affect group of people, like in politics, it is not possible to use instinct or random choice, but all the alternatives must be analyzed in details. That is why some “decision making” processes and algorithms were developed during the years. Decision-making techniques have been used in several fields, but the main field where these processes originated was economics, because the economic system is complex and therefore it requires methods to make decisions.

Tools that help people to make decisions compose the class of Decision Support Systems\textsuperscript{[1]}. This class includes different applications. Some applications consider
also the consequences of a choice, other focus only on helping people to define their preferences.

The Decision support systems were born in the middle 1960s when different people perceive the field of Decision Support Systems from various vantage points and report different accounts of what happened and what was important.

Different system were developed exploiting also the new computer technologies, and researchers inside the universities started to analyze and discuss methods and solutions.

In the 1980s industrial companies implemented some algorithms in their organizations, so the field developed quickly.

Today there are a lot of decision support systems, and they are clustered in five categories: communications-driven, data-driven, document driven, knowledge-driven and model-driven decision support systems [2]. Moreover, we can claim that actually the Decision-Making is a wide-scale used technique, which helps us to take the best choice in several and important cases.

Our work is about the development of a website that provides the possibility to create survey for administrator, and to fill data for users. The website uses two different methods to help the decision process. It is as simpler as possible to use, so it can be used by anyone.
2 Background

Sometimes is necessary to make decisions that involve many criteria, sub criteria and alternatives. In that case, the complexity can grow up exponentially and usually is required an accurate decision schema that considers all the dependencies among factors involved. Therefore, it is advisable to use decision analysis instead of manual approaches (sometimes too complex) or random approaches (which are not good to get the best solution).

The idea behind this work is to provide a useful tool that helps people in making decision. The tool uses two of the main algorithms in this field.

The system should ask people to insert some values related to comparisons, and then analyze data in order to choose the best elements.

There are several techniques based on Multi-Criteria Decision Analysis (MCDA). We have selected two of them, one is the popular Analytic Hierarchy Process (AHP)\(^3\) and the second one is Chainwise Paired Comparison (CPC)\(^4\). Both these techniques support decision making through “Pairwise Comparisons”, as introduced by L.L. Thurstone in 1927\(^5\), which uses comparison between pairs to understand which decision elements have higher relevance on others. In our work, we suppose to have only a criteria so we will not handle more than one criteria per survey.

An example with real data is also provided in this thesis, to show how the two algorithms work and the differences in their results.
3 Decision making through algorithms

In this chapter, the two algorithms used in our work: AHP and CPC will be described in detail. In addition, other algorithms somehow similar to AHP and CPC are introduced.

3.1 Analytic hierarchy Process

Analytic hierarchy Process (AHP) \[^6\] is a technique developed by Thomas L. Saaty in the 1970s and it helps the decision makers to understand which, out of a number of decision elements is the most preferable one. AHP uses pair wise comparisons to create a mesh where each element is compared with all the others.

The algorithm consists in four main steps.

The first one is to set the elements, which eventually can be divided again in sub-elements. After this division, we can organize them as matrix NxN, where N is the number of decision elements (and each eventual subset is defined as another matrix).

The second step consists of collecting data from users, thus each element is compared with all the others. As you can see in Figure 1, the value to insert is in the range 1/9 to 9 (even if we use -9 to indicate 1/9, -8 to indicate 1/8 and so on) where the user can choose the value in this way: he can select 9 if he prefers the first element extremely more compared with the second, 1/9 for the opposite case and 1 if they have equal relevance.
The resultant table (Figure 2) will be symmetrical on the main diagonal, because all the values below the diagonal will be the opposite of the values above on the top of the matrix. The values on the diagonal will always be one, because they are the results of an element compared to itself.

The third step is to calculate the sum of each row, to obtain $N$ sums. Next, divide each one of these sums with the total sum of the values (all the values in the table).

The last step consists only in assigning the decision elements to the respective values in order to know the relevance of each element.

Moreover it is possible to normalize the list of sums to get a kind of percentage, and it is possible in the third step, dividing each element in the matrix by the sum of the column the element is a member of, and calculate the sums of each row.
Then normalize the sum of the rows (divide each row sum with the number of requirements). The result of this computation is referred to as the priority matrix and is an estimation of the eigenvalues of the matrix\[^7\]. In this case if the calculation is correct and accurate we will have precisely consistent values.

The only problem of AHP is that we need a lot of comparisons (since it requires \((N\times(N-1))/2\) for \(N\) elements to compare) to do, but this redundancy of the pairwise comparisons makes the AHP much less sensitive to judgment errors; it also lets you measure judgment errors by calculating the consistency index of the comparison matrix, and then calculating the consistency ratio \[^7\].

\[3.2\quad \text{Chainwise Paired Comparison}\]

Chainwise Paired comparison (CPC) is based on comparisons among two elements each time, but requires less information from the users than AHP does. In fact, it needs only \(N\) comparisons, in a system with \(N\) decision elements, compared to AHP that needs \((N\times(N-1))/2\).

We can say it move the workload from the user to the system. To get the final result the algorithm calculates some intermediate values, so it can take a long time to compute large arrays of elements.

Each element is compared with the next one, and the last one with the first (this is just a standard that we used, is the same if we had chosen other pairs, keeping the chain structure). The last comparison, where the last element is compared to the first, is only to be able to check the consistency. Therefore there are \(N\) comparisons defined as
To have consistency among comparisons, the product of all the $R_i$ must be 1. This product is defined as:

$$\prod_{i=1}^{n} R_i = \left(\frac{W_1}{W_2}\right) \left(\frac{W_2}{W_3}\right) \left(\frac{W_3}{W_4}\right) \cdots \left(\frac{W_{n-1}}{W_n}\right) \left(\frac{W_n}{W_1}\right)$$

If it is not 1, there are two different set of comparison values: Direct Values and Indirect Values. Direct values (also called $D_i$) are directly inserted by the users, whereas indirect values called $I_i$, are generated automatically from the values in $D_i$.

To calculate $I_i$ is necessary to use the product of $D_i$ and then divide each element by the product.

$$I_i = \frac{D_i}{\prod D_i} \text{ for } i = 1 \text{ to } n,$$

Now next step is to adapt the $D_i$ to consistent values. These values are obtained calculating the weighted geometric mean of $D_i$ and $I_i$. This is the value used as $R_i$, when direct values are non-consistent.
After these operations Ri are consistent, and it is possible to calculate for each decision element the relative weight based on the nth element (can be done on one of the elements, setting its value to 1 and calculating the others, to choose the last one make calculations easy). The relative weight of the nth element Mn=1. For each i except i=n, Mi is defined as the product of all the following Ri.

The last step consists in normalizing the relative weights. To do so each Mi is divided by the sum of all the Mi.

The Vi set contains the final values, which are the solution of the problem. The bigger value indicates the element is more favorable.

3.3 Other algorithms
In the first two paragraphs of this chapter an explanation of how AHP and CPC work is provided. Both algorithms are used in the developed system. Some other algorithms could be considered before start working on one of them.

One possible solution, proposed by Chan WK and Tong TKL[^8], is called Grey Relational Analysis. They propose a system in which well known things are white, things with no information are black, and all the other things are in a grey scale. The goal of their algorithm is not to say which one is the best thing, but to provide information to make a good choice.

Thomas L. Saaty, the creator of AHP, is also known for the “Analytical Network Process”[^9]. The Analytic Network Process allows both comparison within group of elements and between groups. Such comparisons are good to define the complex situations in human life, especially when risk and uncertainty are involved. Both the AHP and the ANP calculate priorities for elements and groups of elements by making paired comparisons of elements on a common property or criterion.

The ANP has been applied to a large variety of decisions: marketing, medical, political, social, forecasting and prediction and many others. Its accuracy of prediction is good in applications that have been made to economic trends, sports and other events for which the outcome later became known.
4 Related work

In the available literature, there are several documents related to decision making. Most of them are about studies to solve specific problems or for specific industry situations.

4.1 Other tools available

Making Decisions in Integration of Automotive Software and Electronics: A Method Based on ATAM and AHP\textsuperscript{[10]}

This document describes how a combination of AHP and the Architecture Tradeoff Analysis Method (ATAM)\textsuperscript{[11]}, showing that used together they provide more benefits than used separated. They focus on the design of automotive in-vehicle electronic system. Produce a good quality product with low cost and in the same time, respect the functional and non functional requirements is always a challenge and it always requires making some decisions.

In our work, we use to call AHP the classical AHP algorithm (because AHP can include also other variants of that method but we are going to use the standard approach), with the \((n*(n-1))/2\) comparisons, but here, in this paper, the AHP used is what we are calling CPC which is a subset of AHP.

This tool is directly focused on a particular research area (automotive electrical and electronic system architecture however we can still compare it with our system because they both are using the decision analysis and analytic hierarchy processes. The differences are that our system is using two algorithms, AHP and CPC independently, so is possible to choice which one of them to use. Whereas this approach will uses ATAM and AHP together, with the output of ATAM
used as input of AHP in way to obtain a robust evaluation of both scenarios and how well an integration strategy fits a certain scenario\(^{[10]}\).

To merge those two decision methods there are three steps to follow: elicit scenarios from system stakeholders, then rate importance of scenarios and finally assess scenario fulfilment of each design choice.

Main characteristics:

- Algorithms used: ATAM and CPC
- Application field: Automotive software
- Year: 2007
- Platform: Only method

DecisionLens®

Decision Lens®\(^{[12]}\) is a commercial tool based on advanced analytical methods designed to support decision makers in structuring decisions, quantifying intangible factors and evaluating choices in a comprehensive and rational framework.

It uses the AHP algorithm to manage the different possibilities, and other algorithms (Analytic Network Process) to organize data.

The DecisionLens team proposes a solution in five steps to the problem of resource allocation decision.

The first step consists in gather the critical data needed to make the decision evaluating all the alternatives. The second Phase, called Criteria Development, consist in brainstorm objectives and criteria and then structure them into a hierarchical model. This model will be used to make comparisons. Then a definition of budget rules is required to fix the target. Finally, in the last phase, a decision is taken executing the algorithms.

In our opinion, this tool concentrates more in the analysis than in executing algorithms, so the differences with our work are clearly visible. We do not provide help in defining the elements and analyzing the problems; instead our
tool gives detailed information about the choices once the elements are well defined. We also provide a system that allows multiple users input, while DecisionLens is more for a company manager who has to take decision alone.

Main characteristics:

- Algorithms used: AHP and ANP
- Application field: Marketing, performance, strategy, finance
- Year: 2007
- Platform: Web-based and tool-based

Logical Decision®

Logical Decision® is a tool available since 1991 that allows evaluating and selecting best solution among different difficult decisions. It is designed for one-of-a-kind decisions where you need to think about many concerns at once and make judgments about which concerns are most important to you[13].

The decision making is composed by three simple steps as: organize information, making value judgments and display results. This tool allows choosing different algorithms such as "Smarter" method, "tradeoff" method, "Analytic Hierarchy Process" method and others.

Logical Decision® tool is quite similar to ours; it has the same way to store and manage data but it is able to allow choosing among five algorithms instead of two as our Decision Maker. Therefore Logical Decision® can show few graphs useful for data organizing and result displaying. Although our tool allows the insertion of user data in according with kind of survey, selecting which kind of data the user needs to insert.

Main characteristics:

- Algorithms used: AHP, Smarter, tradeoff and other 2 not specified
- Application field: Business evaluation, Weapon system, Airplane systems
- Year: 1991
• Platform: Stand alone application

Super Decisions

The Super Decisions\textsuperscript{[14]} software is used for decision-making with dependence and feedback (it implements the Analytic Network Process, ANP, with many additions). Such problems often occur in real life. Super Decisions extends the Analytic Hierarchy Process (AHP) that uses the same fundamental prioritization process based on deriving priorities through judgments on pairs of elements or from direct measurements. In the AHP, the elements are arranged in a hierarchic decision structure while the ANP uses one or more flat networks of clusters that contain the elements. Most decision-making methods assume independence between the criteria of a decision and the alternatives of that decision, or simply among the criteria or among the alternatives themselves. The ANP is not limited by such assumptions. It allows for all possible and potential dependencies.

The ANP models have two parts: the first is a control hierarchy or network of objectives and criteria that control the interactions in the system under study; the second is the many sub-networks of influences among the elements and clusters of the problem, one for each control criterion.

The major advantage of this tool is the algorithm; it uses ANP, which is very accurate to evaluate the decision elements in way to take the best choice. The main difference with our Decision Maker is that Super Decision works with only one person, whereas our system is basically based on other people who will fill the surveys.

Main characteristics:

• Algorithms used: ANP
• Application field: Marketing, Medical, Political, Social
• Year: 2000
• Platform: Stand alone application
Telelogic Focalpoint®

“Telelogic® Focal Point™ is a web-based solution for market-driven product managers. It helps organizations to select the right product features and deliver them to the right market segment at the right time by facilitating improved product decision making”[15].

Organizations that develop products and applications software often find it a major challenge to ensure that the products they develop have the right features that meet market or user requirements.

By focusing on customer and business value, Focal Point can help improve the product decisions that managers make throughout the lifecycle, from product conception to delivery[15].

Usually one of the most difficult parts of the product-planning process is deciding whether a feature will be valued by the market segments you are targeting. With Focal Point’s competitive analysis and win/loss reports, you can better understand how your competitors are approaching the same market segments and where their strengths and weaknesses are. [15]

We did not have the possibility to try this tool, but reading their brochures and some customers reviews, we find that their main goal is to automatically capture product requirements from customer e-mails and have them tagged with customer, market segment and product information for easier analysis. This is very different from ours, who is more direct to the goal. The automation in this tool provides a good product that requires only little man work, but in our opinion, it lacks of direct human control.

Main characteristic:

- Algorithms used: AHP
- Application field: Portfolio management
- Year: 2002
- Platform: Web-based
Comparison among tools

In Figure 3 it is possible to see a comparison among the examined tools. Their main characteristics have been merged in the table shown above. The values of comparison are: algorithm used, application field (for few of them the application field is very wide, so we reported only the topic fields where they have been utilized), type of tool (platform used) and year of development. In the first row there is our system, because is interesting to compare it with other existent tools.

In the table we compared only the most common characteristic, because even if there are many other interesting characteristics to mention, they are different in some way, so few aspects are not comparable.

<table>
<thead>
<tr>
<th>Tools</th>
<th>Algorithms</th>
<th>Application field</th>
<th>Platform</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Maker</td>
<td>AHP or CPC</td>
<td>All fields</td>
<td>Web-based</td>
<td>2009</td>
</tr>
<tr>
<td>Making Decisions in Integration of</td>
<td>ATAM and AHP</td>
<td>Automotive software</td>
<td>Only method</td>
<td>2007</td>
</tr>
<tr>
<td>Automotive Software and Electronics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DecisionLens®</td>
<td>AHP and ANP</td>
<td>Marketing, performance, strategy, finance</td>
<td>Web-based and tool-based</td>
<td>2007</td>
</tr>
<tr>
<td>Logical Decision®</td>
<td>AHP, Smarter,</td>
<td>Business evaluation, Weapon system, Airplane systems</td>
<td>Tool-based</td>
<td>1991</td>
</tr>
<tr>
<td></td>
<td>tradeoff and other algs.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Super Decisions</td>
<td>ANP</td>
<td>Marketing, Medical, Political, Social</td>
<td>Tool-based</td>
<td>2000</td>
</tr>
<tr>
<td>Telelogic Focalpoint</td>
<td>nd.</td>
<td>Portfolio management</td>
<td>Web-based</td>
<td>2008</td>
</tr>
</tbody>
</table>
5 Method

This section describes why the system was implemented as a website and not as a standalone application, which technologies we used for the implementation and how we structured the system. There is also an explanation of the system, how it works and how to use it.

5.1 Web Site based: pros and cons

The system was developed as a website. An alternative was to make it as a standalone application, but as a website, it has several advantages and some small disadvantages that can be easily solved.

Pros:

- It does not require an installation, so it is free for all the users and always available online. In a standalone application users will be probably indisposed to install the application or to go in some particular place to fill the survey. Whereas a web-based system is always, ready to use and require just few minutes!
- New versions can be made available immediately for all the users, this means that when new features are added, everyone can use them with no cost. The idea is that in a world where all the things change quickly, it is important to keep the system always up-to-date, and do it automatically.
- It works on all operating systems and modern browsers, thus users and administrators do not have to use the same system. This is one of the most important reason, especially because new operating systems are ready to be launched (different Linux versions, Google probably will release an OS, etc.).
- It offers the possibility to reach a lot of users to collect huge data (exploiting internet)

Cons:
- There is only one database and it can be a problem if a huge amount of users starts to input data. It can be resolved providing the system to companies that can install it on their servers.
- Internet connection is always required and there can be problems with security issue.

5.2 Technologies involved

Since the system is designed to be a website, the choice of programming languages was done among the most common technologies, to be sure to utilize the latest advantages from the current technologies.

First, the system has been developed with PHP technology, since it is the most common web programming language and allows us to implement everything we need. PHP is a widely used general-purpose scripting language that is especially suited for Web development. The current version is PHP 5, and it provides a very good control on the dynamic logic. It is also good for managing the SQL queries.

Then we chose HTML, which is the web pages standard language. It is very powerful since it provides all the basic things to create user interfaces. Together with the Cascade Style Sheet (CSS) it is possible to create interesting graphical effects and to show all the content coming from other languages.

On the client side, we used the JavaScript (also called JS) to make html pages more dynamic, and to check the validity of forms. JavaScript is a powerful language to access object inside the html code, so we used it also to modify some object without reloading the page every time.
Exploiting the powerfulness of JS, we implemented some AJAX functions. They provide the way to execute some PHP code after the page is load. For example, we used them to update the list of decision elements when the administrators set their number.

To manage persistent data we used PostgreSQL version 8. An alternative to Postgre is MySQL, but we preferred to have a more robust system even if the performance is somehow lower.

Finally, one of the most important services we used is Smarty. Probably Smarty is the best template engine in the world. It allows the organization in folder to keep separate HTML, JavaScript and PHP, providing all the functions to show data together and to control the data flow among pages.

5.3 Organization – Files and folders

Since the start of the project, our experience helped us to decide a policy to organize the work and simplify the structure.

Before starting the implementation, the system was divided in folders. Having a good structure reduces the time to find where to work, and it guarantees extendibility.

The system (implementation part) is divided in these folders:

- css – it is the folder that contains the Cascade Style Sheet. Here there are all the graphics information
- images – it contains all the images needed to show the template
- include - it contains PHP files with function which can be called in the other files, such as the connection to the database.
- js – it contains JavaScript files
- libs – it is folder with all the Smarty libraries that compose the template engine’s core
- modules – it contains the PHP files with the queries for the database
• sql – it contains the definition of the database and all the tables. It is useful to deploy the system on a new server
• templates – it contains the outline of the page. An outline can be used for several pages, putting in it the content of the page
• templates_c – it is the Smarty folder to compile the templates
• cache – this folder is another Smarty folder, used to save some data, we did not put files inside this folder

Finally, in the root folder we store the PHP files which are the core of the system.

We also decide to organize the documentation in sub-folders, so we have:

• Week reports – it contains the working hours per week, also used to create some statistic of required time in each field.
• Design – it is the folder with all the design files, as use cases, database design, sequence diagrams and so on.
• Math – it is the folder used to put dome Excel files to few files useful to understand the math logic before implementation.
• Documents – all the documents we produced.

5.4 Web Site Overview

Administrator area

Since it is necessary to collect data from different users belonging to different working categories, it has been decided to give the administrator the possibility to add and delete categories and then to assign a category to each survey. After creating a new category or selecting an existent category the new survey can be created, so there is a form where the administrator can choose the name, which data users are required to provide (among first name, last name, gender and age), the number of decision elements and to use AHP or CPC. We implemented an AJAX module to display a list of textboxes needed to insert the decision elements. This list is automatically generated each time the administrator changes
the number of decision elements. Each decision element has a name and a comment. The comment can be useful sometimes to explain details about the element, but can also be empty. When all the data in the page are set, the new survey is stored in the database and available for the selected category of users.

Other feature available for an administrator is to add new administrators, in order to add other people to be able to manage the system.

**User Area**

The first page of the user area asks the user to select his working category, when he choose it, the surveys available for him/her are shown (again we used AJAX to manage it dynamically, because the user can belong to more than a category and can change it to before selecting the survey). He can also add some comments about the survey or the category. He clicks to go forward and he is required to add his data (data depend on what the administrator ask for the chosen survey). Then the data are stored and the user inserts his preferences in the comparison page. Here one of the two possible algorithms is executed and the results stored.

**Website Map**

![Website Map](image)

*Figure 4 Website Map*
In Figure 4 show a map about the main pages of web site. The two main pages are the user and administrator home pages, where they can choose all features to do. The user has limited number of options since most choices are determined by the administrator. The user should basically just answer the questions, but the administrator has several pages about other administration functionalities.
5.5 Website Description

The web site home page (Figure 5) is very simple as it only has a short description of the goal of the system and a link to the survey, where users can put their data. There is a banner on the top, which contains few buttons to navigate between user and administrator side (through login section). To the left there is a general explanation of decision making in the main section, followed by a short slogan to “present” the system.
Using the login panel in the top right, it is possible to access to the administrator main page (Figure 6), most features of the system can be accessed, including the ability to add a new administrator, show survey results, show user result, add a new survey and manage the categories.
The Figure 7 shows how an administrator can create a new survey. The web page is divided in two sides; the first one is for the general information about the survey (as name, category, algorithms, etc) and the second is generated afterwards by AJAX, in according with number of elements.
If the administrator wants to check the results from a survey or the results of a user, he can use the two relative functionalities, which lead the administrator on a page necessary to select the survey/user. After selection, the results are shown through a list as shown in figure 8.

Moreover, it is possible to visualize the user who filled the survey shown and visualize his results for that survey.
Figure 9 is about the creation of a new administrator. A simple form requests all information necessary for the query that will create the new administrator into database.
The categories are very important for the surveys, because an administrator should be able to manage those as he wants, both deleting and adding others.

To do this, it is useful to visualize a list of existing categories that is dynamically updated when a new category will be added or deleted (Figure 10).
For the user side, shown in Figure 11, the first page is to insert the category of the user and then to select one of surveys dedicated to his category.
In the Figure 12, the user can insert the comparison values about the pairs of elements.

The pairs will be visualized as “Element 1 – Element 2 – Value”, in order to be easy and understandable for users (there are also comments below elements).

The value that he inserts will be stored in the database and then when the administrator will want to show the results an arithmetical mean will be made (we use an SQL function called AVG()).
6 Software engineering Process

When developing the Decision Maker system, a waterfall model as development process was used in order to organize all the development phases. Following this model, we divided the work in 5 phases:

- Project Plan
- Requirements
- Design
- Development
- Verification and Test

6.1 Project plan

We started this work by analyzing the available literature, so we created our own background about the topic. We found that generally, the tools available are created for specific purposes, but often the ideas behind them are similar and sometimes they use the same algorithms.

We tried to define the time needed for each phase; and based on conversations with our supervisor we decided how the work should be organized.

To understand which part of the work that would consume most of our time, we divided the working hours in different categories:

1. Understanding and Study
2. Design
3. Html and CSS (including also JavaScript)
4. PHP and Database
5. Algorithmic part
6. Documentation
By filling a week report, we could check how our work was progressing and which parts required more time.

Figure 13 Working hours planned before starting

Figure 14 Effective working hours
As shown in Figure 13 and 14, we spent a different amount of time with respect to the plan.

Actually, we thought the algorithmic part would take more time, and we planned around 25%. On the other hand, we spent more time than planned in the PHP and HTML programming.

6.2 Requirements

As for each software engineering process, we divided requirements in functional and non-functional.

The functional requirements are the features to provide to the administrator and customers; we divided them in four categories:

1. General
2. User area
3. Administrator area
4. Login area

Of course, the requirements we selected at the beginning changed a little, but we did not lose the waterfall model because there were only small changes and some new not determinant requirements. Dividing the requirements in categories was really a good idea, so we could concentrate on each specific category during the development.

We did not divide the non-functional requirements since they are only few. They are about the responsiveness time, the system requirements, the display constraints and so on.

For detailed information on requirements, see the appendix A.
6.3 Design

Another important phase is the system design. After collecting all the requirements, we had a faded idea of how to implement the system, and we composed all the things creating some use cases.

Following the requirements, we divided user from administrators and listed all the functionalities they can use. For each part of the functionality, we prepared a short description to add some details and the scenario, to state the preconditions, post conditions and some other important information.

After creating use cases, we understood the situation enough to start the development phase, but we also created some dynamic models, to state the data flow. In fact, some sequence diagrams helped us to organize the structure.

In this phase, we decided also which data structure to use and in addition, how to allow interactions among technologies.

The meeting with our supervisor(s) every week speeded up our work, because we collect feedbacks even before starting the implementation.

We draw a schema of the database, which required only small changes during the development. The database is composed by eight tables where one is to manage the administrators, three to operate on survey and four to organize user data and results. We decided to store results for each user and for each survey, so administrator can check data in two different ways.

The work is gone very well in this part, because we did not have big problems in the development phase, in fact the structure of the system we created at the starting of the development was good throughout the project.

In the Appendix B, you can read and analyze in detail our design phase while in Appendix C you can see the database design with description.
6.4 Development

The development part was where we spent most time. Working on the development of the system was done in different parts.

First, we defined all the pages creating HTML layout. They were only static pages, without any logic behind, but gave us the possibility to see how the final result would be. In this part we iterated with our supervisor, to avoid misunderstanding, which would have created problem in the future.

At the end of this part, we had the schemas of all the pages, and we started to work on the PHP programming part.

We used the Smarty templates engine to keep the HTML and PHP separate and in the same time have the advantages of using pre-defined functions to manage arrays, and other PHP structures.

During the PHP implementation, we worked in parallel creating the algorithm with static input and preparing the real data in the administrator and user part. In this way, we could control both things and we did not have to wait before implement the algorithms. This made us work more effectively so we could manage to finish on time.

While writing PHP code, we also implemented SQL queries to put and fetch data from the database, still respecting our policy to keep separated different technologies in different files.

When all the data were ready we joined the algorithmic part with the user data, and fortunately, we did not have particular problems with it.

After these operations, the system was almost ready, so we started to put some JavaScript checks, to avoid wrong input from users, and other general controls.

Appendix D contains some snippets of the most important part of the system, such as the algorithms.
6.5 Verification and test

Once the implementation was finished, we started to test the system. During the implementation, we did continuous module and unit testing.

In this way, we applied a lot of modification and we added several checks with JavaScript and PHP deleting the web site’s bugs.
7 Results

7.1 AHP vs. CPC – Case Study

In order to understand how these algorithms work with real data, we can make a case study creating a survey with data about foods. We will compare the different results from both algorithms, AHP and CPC, so that we can have a feedback from how they work, how they are useful and of course, which one is better.

We should consider that the two algorithms are not exactly the same because of the inputs, which are different considering AHP has more comparison than CPC. So they do not match enough to say which one of them is the preferable, but at least we can compare their results to understand how they react with their inputs.

The case study is about foods, we will compare several kinds of foods such as meat, vegetable, soup, pizza, pasta, fish and potato. As already mentioned, we a different number of comparison for two algorithms but we are still using same values for the comparisons in order to keep similar proportions among them.

We will present only one insertion of data, as if data are filled only by a user and not more, because is enough to understand how algorithms are going to work, even if system manages big amounts of data using averages calculation.
AHP algorithm

For the AHP side, we have a number of comparison of \((n*(n-1))/2\) so in our case (with 7 elements) we have \(7*6)/2=21\) comparisons.

In the following table (figure 15), there are inputs for the AHP test. The table is symmetrical, so on the main diagonal we will find always 1 and, below the diagonal, there are the opposite values of top side colored with red font.

<table>
<thead>
<tr>
<th></th>
<th>Meat</th>
<th>Soup</th>
<th>Potato</th>
<th>Vegetable</th>
<th>Pizza</th>
<th>Fish</th>
<th>Pasta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meat</td>
<td>1</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1/2</td>
</tr>
<tr>
<td>Soup</td>
<td>1/9</td>
<td>1</td>
<td>1/3</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
<tr>
<td>Potato</td>
<td>1/5</td>
<td>3</td>
<td>1</td>
<td>1/3</td>
<td>1/4</td>
<td>1/3</td>
<td>1/9</td>
</tr>
<tr>
<td>Vegetable</td>
<td>1/2</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1/2</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>Pizza</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1/2</td>
</tr>
<tr>
<td>Fish</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1/2</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>Pasta</td>
<td>2</td>
<td>9</td>
<td>9</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 15 Example Inputs for AHP case study**

From those inputs, the algorithms calculate the final results and can generate a list of element with relative values. Higher values correspond to higher relevance.
- Meat  - 0.183471858501434
- Soup  - 0.017772201448679
- Potato - 0.049187183380127
- Vegetable - 0.144268468022347
- Pizza  - 0.183471858501434
- Fish   - 0.148972868919373
- Pasta  - 0.272855579853058

Sorting the list by values and we will have the result we wanted to obtain, Pasta, Pizza, Meat, Fish, Vegetable, Potato, Soup.

**CPC algorithm**

In the CPC algorithms we have n comparison, where n is the number of elements, so in this case there will be 7 comparisons between pairs, handled as chain style (first element with second, second with third, and so on).

Those input that you can see below, are a subset of comparisons used by AHP. So of course the AHP will be more precise with calculations, but CPC is faster because it requires almost half the number of comparisons. Depending on the importance of the decision, what level and robustness you need, how much time it is allowed to take (cost), it will be decided which one of the two algorithms to use.

In the list of input data presented in figure 16, the data are not consistent (to be consistent the last comparison should be 3 instead of 2) but the CPC algorithms can manage some inconsistency by itself, but if too large a new evaluation is needed.
As for AHP, CPC algorithm generates a list of elements with their values in way to understand the highest ranked elements.

- Pasta - 0.316478
- Pizza - 0.2369
- Meat - 0.149334
- Vegetable - 0.125513
- Fish - 0.111784
- Potato - 0.0443328
- Soup - 0.0156588
Differences of results

The inputs of the two experiments were quite similar (as we already mentioned CPC comparisons are a subset of AHP comparisons, with same values). Even if CPC has fewer comparisons, the results in the list are similar, there are only few different values and two elements (Vegetable and Fish) are sorted in opposite way.

The differences between AHP and CPC in Figure 17 come from the reason that CPC has fewer comparisons compared with AHP so, having less information, it cannot have exactly the same precision of AHP. However, those differences between the results of algorithms could be greater if more elements are compared.

<table>
<thead>
<tr>
<th></th>
<th>CPC</th>
<th>AHP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pasta</td>
<td>0.316478</td>
<td>Pasta - 0.272855579853058</td>
</tr>
<tr>
<td>Pizza</td>
<td>0.2369</td>
<td>Pizza - 0.183471858501434</td>
</tr>
<tr>
<td>Meat</td>
<td>0.149334</td>
<td>Meat - 0.183471858501434</td>
</tr>
<tr>
<td>Vegetable</td>
<td>0.125513</td>
<td>Fish - 0.148972868919373</td>
</tr>
<tr>
<td>Fish</td>
<td>0.111784</td>
<td>Vegetable - 0.144268468022347</td>
</tr>
<tr>
<td>Potato</td>
<td>0.0443328</td>
<td>Potato - 0.049187183380127</td>
</tr>
<tr>
<td>Soup</td>
<td>0.0156588</td>
<td>Soup - 0.017772201448679</td>
</tr>
</tbody>
</table>

Figure 17 Result comparison AHP vs. CPC
8 Discussion

In this chapter we will discuss if AHP and CPC are useful and why. Moreover we will explain our decisions and choice about the implementation and problem found during the project.

8.1 Are AHP and CPC useful?

During our work, we tried the system with different inputs, and executing the two algorithms we find out that for small decisions people can use their intuition to make decision. When the decision elements increase, every choice become difficult to make, while it is still easy to compare elements in pair.

We can claim the pairwise comparison (which is the main feature of AHP and CPC) makes the algorithms very useful and simple to use.

In the administrator side, the topic part is to define a new survey defining all the elements and data about the survey. Then the user finds directly the elements in pair and has only to put their value. For the administrators it is not any difference if a survey is created for AHP or CPC.

The main advantage of AHP is the possibility to divide the criteria (decision elements) in sub criteria, which can be useful when more details must be considered for an element. Moreover, the AHP algorithm provides very precise results, since it has a complete control on all the couple of elements.

On the other hand, AHP has a big disadvantage that it requires \((N^2-1)/2\) comparisons, and users are often bored to fill out long surveys.

From a time perspective, CPC is much better because it requires only \(N\) comparisons, where \(N\) is the number of elements. For example, if the survey is
done with 12 elements to compare, only 12 pairwise comparisons are required to execute CPC, while 66 using AHP

8.2 Implementation and algorithmic choices

During the design and development, we had to make some decision about the implementation and the results management.

The comparisons between two elements are done giving a number that indicates how many times the user prefers the first element over the second. The values from 2 to 9 indicates that the first element is better than the second while the negative values (-2 to -9) are to express preference on the second. The value 1 indicates that the two elements have the same weight. The number 0 and -1 are not valid since -1 is equal to 1 and 0 has no meaning. In the web-site the values are shown as described from -9 to 9, but in the calculations they are converted in decimal numbers for the algorithms (for example -2 becomes ½ so 0.5, -3 becomes 1/3 and then 0.3333 and so on). We did this transformation to increase usability, because negative numbers have a better impact on users.

In addition, it has been decided to show results after the evaluation for each user and for the whole survey. We stored the results for users and then we calculate the average among the entire user to get the survey result.

The results are decimal values, so the biggest value means the most favorable element. Another important thing is that is possible to know not only which the best element is, but also how much better it is. So it could happen that the first element is much better than the others, but also that the first and the second are almost on the same level. This helps the administrator to have a deep look in the results, which are more than a simple classification.
8.3 Problems

There were no major problems. In the beginning it was hard to get a good overview of the requirements of the system. The administrator area and the user area were confused. Nevertheless, the frequent meetings with our supervisor helped us a lot.

At the end of the project, one of us was in Canada to participate to a challenge, so we were in a hurry to finish all the documentation, but fortunately, we finished the main part of the implementation two weeks before, so only some small checks were needed to implement.
9 Conclusions

The lesson we learned from the case study in previous chapter is that we should consider which kind of survey we are analyzing before to choice an algorithm instead of another. AHP is performing and accurate whereas CPC can manage many decision elements but it has not an accurate effectiveness with the returned values.

This lack of precision of CPC algorithm is supplied by saving effort at filling time, when people have to insert values. In detail, if we have a survey of 20 elements the AHP will ask 20*19/2 = 190 comparisons!

Of course, this limit about number of comparisons makes AHP too difficult to be used in cases where we have many elements to compare, in those it is better to use CPC.

Nevertheless AHP return values extremely accurate and it can be used in almost all cases (is not so common to have more than 10 elements) obtaining a quality decision support.

We can claim there are several methods and approaches, each with its own benefits and drawbacks, depending on the situation and therefore we should choice it considering which kind of decision and in what field we are going to use it.

Considering all the differences, we built a system where the administrator can choose the algorithm to use every time he creates a survey.

To fill a survey it takes less than a minute to insert personal data plus about 30 seconds for each comparison. This means that an average survey in CPC is filled in less than 10 minutes while AHP in the same time can manage only survey with 4-5 elements. If we raise the time to 25 minutes (a time generally accepted by
most of users) we can do survey with more than 40 elements in CPC and 9 elements in AHP.

Our policy was to keep the system as simple as possible, especially in the user area, so everyone can fill out surveys. No special ability is required, because users only have to read the name of the two comparison elements and put the value. We also added the possibility for administrators to describe each element in a comments field to let the users get a deeper understanding of each element.
10 Future work

The field we worked on is quite open. AHP was created to help organizations and people to make economic and political decisions. In the following years it was applied to different fields, applying different modifications. Our application aim to provide a general support to company managers, professors at university and group leaders or anyone else that would like to make structured decisions. We think the idea behind it is very adaptable.

In future, it will be good to create similar web applications for specific purpose, such as hospitals and schools (for example to help to decide which subject is more important than others).

Moreover, we did not manage the sub criteria in AHP, and having such feature in this system can be useful to adapt to more complex systems.

Since we found advantages and disadvantages in both AHP and CPC, a good idea is to implement an intermediate solution among them, to guarantee a good level of consistency and in the same time not ask the user to pent a lot of time providing data.

Finally, we think people really like to have visual results, not only numbers. So we are planning to add some graphs to our surveys, to make the tool more intuitive and usable.
11 Appendices

11.1 APPENDIX A – REQUIREMENTS

This appendix provides some information about requirements of our system. We classify requirements in two groups as usual: Functional Requirements and Non-Functional Requirements.

Functional Requirements contain all the functionalities we want to provide while Non-Functional state quality issues.

In the two tables below (Figure 18 and 19), we report all the requirements. For each requirement, we put a unique identifier, a short description and a number that indicates priority: 1 indicates it must necessary be done, 2 means it is good to do, 3 means we will do it if we have time.

Legend:

G= General
U= User area
A= Administrator area
L= Login area
## Functional Requirements

<table>
<thead>
<tr>
<th>Requirement no</th>
<th>Requirement group</th>
<th>Description</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>G</td>
<td>The system should contain an Administration section and a user section</td>
<td>1</td>
</tr>
<tr>
<td>F2</td>
<td>A</td>
<td>The administrator can add surveys</td>
<td>1</td>
</tr>
<tr>
<td>F3</td>
<td>A</td>
<td>The administrator can choose approximately which algorithm to use (CPC or AHP) for a survey</td>
<td>1</td>
</tr>
<tr>
<td>F4</td>
<td>A</td>
<td>An administrator can chose the category for a survey and add/remove categories.</td>
<td>1</td>
</tr>
<tr>
<td>F5</td>
<td>A</td>
<td>The administrator should visualize the results for each one user who filled a survey</td>
<td>1</td>
</tr>
<tr>
<td>F6</td>
<td>L</td>
<td>The system should allow a login feature for administrators</td>
<td>1</td>
</tr>
<tr>
<td>F7</td>
<td>G</td>
<td>Each one element could have a description in way to allow understanding of the users.</td>
<td>1</td>
</tr>
<tr>
<td>F8</td>
<td>U</td>
<td>A user is be able to fill only survey available for his working category or surveys available for all categories</td>
<td>2</td>
</tr>
<tr>
<td>F9</td>
<td>G</td>
<td>All comparison values between two elements must be in the range from -9 (equivalent to 1/9) to 9.</td>
<td>2</td>
</tr>
<tr>
<td>F10</td>
<td>A</td>
<td>The administrator can add different</td>
<td>2</td>
</tr>
</tbody>
</table>
information to require when a user fill the survey

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>F11</strong></td>
<td>A</td>
<td>The final results should be a list of elements sorted by relevance trough the values calculated with algorithms.</td>
</tr>
<tr>
<td><strong>F12</strong></td>
<td>G</td>
<td>The system should handle the data consistency.</td>
</tr>
<tr>
<td><strong>F13</strong></td>
<td>G</td>
<td>It should be possible to register a new administrator.</td>
</tr>
<tr>
<td><strong>F14</strong></td>
<td>U</td>
<td>The user page could show the decision elements like a table for AHP algorithm.</td>
</tr>
<tr>
<td><strong>F15</strong></td>
<td>U</td>
<td>The user page could show sequentially all the selected pairs for CPC algorithm.</td>
</tr>
</tbody>
</table>

Figure 18 Functional Requirements
Non Functional Requirements

<table>
<thead>
<tr>
<th>Requirement no</th>
<th>Description</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>NF4</td>
<td>The system should support all the resolutions in the range 1024x800 to 1920x1440</td>
<td>1</td>
</tr>
<tr>
<td>NF1</td>
<td>The algorithm must take less than a minute, even for big input</td>
<td>2</td>
</tr>
<tr>
<td>NF3</td>
<td>The system should support most of operating system such as Windows Xp, Windows Vista, Leopard, Linux</td>
<td>3</td>
</tr>
<tr>
<td>NF2</td>
<td>The website should have a good rendering with at least Mozilla Firefox, Microsoft Explorer, Google Chrome and Opera.</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 19 Non Functional Requirements
11.2 APPENDIX B – DESIGN

USE CASE – ADMINISTRATOR SIDE

Figure 20 Use case administrator side

Login

**Description:** The Login feature allows having one or more administrators who can manage the private sections of the site.
**Website path and Scenario required:** This is reachable from home page and from each user page, it doesn’t need any constraint, it can be applied in whatever situation.

**Add new survey**

**Description:** Probably is the main feature which administrator can do, it is to add new surveys (choosing algorithm, selecting which data user are required to put, add elements…)

**Website path and Scenario required:** From button “Add survey” in the admin home page is possible to add survey after the administrator is logged in.

**Show survey results**

**Description:** An administrator can select a survey and visualize all results of his elements, in way to get a list of elements with correspondent values.

**Website path and Scenario required:** There is a button “Show survey results” in the admin home page, there are no particular constraint for that, but of course if there are no any surveys it will be not possible to select one of them.

**Show user results**

**Description:** The administrator can select a user and visualize all his results.

**Website path and Scenario required:** This feature is in the admin home page, with a dedicated button, it is similar to the survey show feature, so if you want to visualize any result you should have at least one person who filled the survey.
Add category

**Description:** With this feature, the administrator can add new categories that are necessary to group the surveys.

**Website path and Scenario required:** As previous functionalities, the add category is reachable from admin page with no constraints.

Delete category

**Description:** This is necessary in case an administrator wants to delete some categories that are not in use.

**Website path and Scenario required:** This feature is in the admin home page, with the button “delete categories”, to let it work, is necessary to have any categories in the database.

Add administrator

**Description:** It is possible to add a new administrator, giving him all the rights to manage the website.

**Website path and Scenario required:** This is included among the other administrator features from administrator page. The scenario is that at least an administrator must be present on db.

Logout

**Description:** The Logout feature allows to exit from administrator mode coming back to index page.

**Website path and Scenario required:** This is reachable from each admin pages so with all scenarios where the administrator is logged in.
USE CASE – USER SIDE

Chose and fill a survey

Description: The user can choose one survey from a list of available surveys and so he can fill the survey selecting a value for each comparison.

Include: This functionality includes “Insert own data”. It is necessary because in some cases, users are required to put their personal data in the survey.

Website path and Scenario required: Directly from the home page, the user can start clicking to user side pages, then he can choose his working category and can select which survey wants to apply. To chose a survey to fill, is necessary to have at least a survey available according to the working category the user selects.
SEQUENCE DIAGRAM – ADMINISTRATOR SIDE
In UML (Unified Modeling Language), the sequence diagram is describes the dynamical behavior expressed in term of sequence message as requests. The objects on the top of figure are all the components that interact in our system and they are the administrator, the GraphicalUserInterface, the Javascript, the PHP and the Database.

In Figure 22, you can see all the functionalities that the administrator can do and first of all, he must login in the system. There are few frames in the figure, we
can see one of them in login feature, because is the administrator is not present or if the password is wrong, the login will fail so the frame is an alternative, that manage the successful or the failure login.

If the login is correct, the administrator has many features to apply such as:

- Add new administrator
- Add new survey
- Show survey results
- Show user results
- Manage Categories

Those features are described in the sequence diagram as well.

Finally the administrator can do the Logout.
About the user side you can see in Figure 23 that all the interactions among user system objects. The user has not much functionality to do, he can only fill a survey, and to make this possible, he is required to select his working category (in order to visualize only the surveys available for his category). The first frame “loop” represents the interactions between PHP and HTML (GUI) in order to show all the comparisons and the second is to allow all the insertions from user.
11.3 APPENDIX C – DATABASE

The database in Figure 24 is composed by 8 tables where one is to manage the administrators, three to operate on survey and four to organize user data and results. The “administrator” table contains data to let the administrators access their area. The table “Survey” contains information inserted by the administrator, such as the type of algorithm to apply, the data requested and so on. “Survey_data2 contains users input for each survey (only personal data) while “user_values” contains the values inserted by the user for each element. The result of the execution is stored in “user_results”, always user by user. Whereas the table “results” contains the final information for each element. The list of
elements is contained in the table “Element” and the list of categories in the table category.

11.4 APPENDIX D – SOURCE CODE

CPC MAIN CODE

```php
/* we will prepare the array to give as input to cpc function */
$table=$nelements=$_SESSION['elemnumber'];
for($i=0;$i<$nelements;$i++) {
    $firstname = "first".$i;
    $secondname = "second".$i;
    $par['id_survey_data']=$_SESSION['id_user'];
    $par['firstelement'] =$_REQUEST[$firstname];
    $par['secondelement'] =$_REQUEST[$secondname];
    $par['value'] =$_REQUEST[$par['firstelement']] ;
    /* query which inserts user values into user_values table */
    insertValues($par);
    $next=$i+1;
    $next2=$i+2;
    /* code lines to insert the name of the row (like: s1-s2) and the value */
    if($i!=$nelements-1) $table[$i]=array ( 'cel' => 's'.$next.'-s'.$next2,
    'di' => $par['value'],
    'ii' => "",
    'ri' => "",
    'mi' => "",
    'vi' => "",
    'firstelement' => $par['firstelement'] );
    else $table[$i]=array ( 'cel' => 's'.$next.'-s1',
    'di' => $par['value'],
    'ii' => "",
    'ri' => "",
    'mi' => "",
    'vi' => "",
    'firstelement' => $par['firstelement'] );
    } /*call the cpc function (the other columns such as ii,ri,mi,vi*/
$result = calculateValuesCPC($table);

/* query to insert the results into user_results table*/
for($i=0;$i<$nelements;$i++) {
    insertUserResultsCPC($result[$i],$_SESSION);
}
```

Figure 25 CPC - Array preparation and function call (user_survey_last.php)
function calculateValuesCPC($table) {
    $di_prod=1;
    $num_rows= sizeof($table);

    /*calculates the product of all the Di (values set by the user)*/
    for ($row=0; $row< $num_rows; $row++){
        $di_prod = $table[$row]['di']*$di_prod;
    } //End for row

    /*calculates ii (di/di_product)*/
    for ($row=0; $row< $num_rows; $row++){
        $table[$row]['li']= $table[$row]['di']/$di_prod;
    } //End for row

    /*calculates Ri (di^[((n-1)/n)] * ii^[1/n])*/
    $exp1=($num_rows-1)/$num_rows;
    $exp2=1/$num_rows;
    for ($row=0; $row< $num_rows; $row++){
        $table[$row]['ri']=
        pow($table[$row]['di'],$exp1)*pow($table[$row]['li'],$exp2);
    } //End for row

    $ri_prod=1;
    for ($row=0; $row < $num_rows; $row++){
        $ri_prod= $ri_prod * $table[$row]['ri'];//Equal to 1
    } //End for row

    /*calculates Mi (prod(ri:rn-1))*/
    for ($row=0; $row< $num_rows-2; $row++){ //End for row
        $res=1;
        for($row2=$row; $row2< $num_rows-1; $row2++){
            $res=$res * $table[$row2]['ri'];
        }
        $table[$row]['mi']=$res;
    } //End for row

    $table["num_rows-2"]['mi']=$table["num_rows-2"]["ri"]; //End for row
    $table["num_rows-1"]['mi']=1;
    $mi_sum=0;
    for ($row=0; $row < $num_rows; $row++){
        $mi_sum= $mi_sum + $table[$row]['mi'];
    } //End for row

    /*calculates Vi(mi/mi_sum)*/
    for ($row=0; $row< $num_rows; $row++){ //End for row
        $table[$row]['vi']=$table[$row]['mi']/$mi_sum;
    } //End for row

    $vi_sum=0;
    for ($row=0; $row < $num_rows; $row++){
        $vi_sum= $vi_sum + $table[$row]['vi']; //Equal to 1
    } //End for row

    return $table;
}
AHP MAIN CODE

$ncomparisons=(\text{n}e\text{l}e\text{m}e\text{n}ts*(\text{n}e\text{l}e\text{m}e\text{n}ts-1))/2;

for($i=0;$i<$ncomparisons;$i++)
{
    /* building of names to recover data from Session and Request */
    $firstname= "firsthidden".$i;
    $secondname="secondhidden".$i;
    $valuename="value".$i;
    $par[‘id_survey_data’]=$_SESSION[‘id_user’];
    $par[‘firstelement’]= $\_REQUEST[$firstname];
    $par[‘secondelement’]= $\_REQUEST[$secondname];
    $par[‘value’]= $\_REQUEST[$valuename];

    /* query which inserts user values into user_values table */
    select and increments the number of user that filled the survey */
    insertValues($par);

    /* setting diagonal values */
    $table[$\_REQUEST[‘firsthidden’].$i][$\_REQUEST[‘firsthidden’].$i] = ‘1’;
    $table[$\_REQUEST[‘secondhidden’].$i][$\_REQUEST[‘secondhidden’].$i] = ‘1’;

    /* setting upon side of matrix */
    $table[$\_REQUEST[‘firsthidden’].$i][$\_REQUEST[‘secondhidden’].$i] = $\_REQUEST[‘value’].$i;

    /* setting bottom side of matrix */
    $table[$\_REQUEST[‘secondhidden’].$i][$\_REQUEST[‘firsthidden’].$i] = opposite($\_REQUEST[‘value’].$i);
}

/*call the ahp main function */
$result = calculateValuesAHP($table);

/* query to insert the results into user_results table*/
foreach($result[1] as $i){
    insertUserResultsAHP($result,$i,$_SESSION);
}

Figure 27 AHP – Array preparation and function call (user_survey_last.php)
function calculateValuesAHP($table){
    $num_rows= sizeof($table);
    $ncomparisons=($num_rows*($num_rows-1))/2;
    
    /* id of the elements */
    for($i=0;$i<$ncomparisons;$i++){
        $array[$_REQUEST['firsthidden'.$i]]=$_REQUEST['firsthidden'.$i];
        $array[$_REQUEST['secondhidden'.$i]]=$_REQUEST['secondhidden'.$i];
    }
    
    /* calculates sums of values */
    foreach($array as $i){
        foreach($array as $j){
            $sum[$i]=$sum[$i]+$table[$i][$j];
        }
        $sumvar=$sumvar+$sum[$i];
    }
    
    /* division by sum*/
    foreach($array as $i){
        $result[0][$i]=$sum[$i]/$sumvar;
        $result[1][$i]=$i;
    }
    return $result;
}

Figure 28 AHP – Function to calculate AHP results (ahp_core.php)

function opposite($par){
    /* calculates the opposite of the value, to obtain the other half matrix*/
    if($par=="0.111111111111") return '9';
    if($par=="0.125") return '8';
    if($par=="0.142857142857") return '7';
    if($par=="0.166666666667") return '6';
    if($par=="0.2") return '5';
    if($par=="0.25") return '4';
    if($par=="0.333333333333") return '3';
    if($par=="0.5") return '2';
    if($par=="9") return '0.111111111111';
    if($par=="8") return '0.125';
    if($par=="7") return '0.142857142857';
    if($par=="6") return '0.166666666667';
    if($par=="5") return '0.2';
    if($par=="4") return '0.25';
    if($par=="3") return '0.333333333333';
    if($par=="2") return '0.5';
    if($par=="1") return '1';
}

Figure 29 AHP – Function to calculate the opposite values (ahp_core.php)
QUERY CODE EXAMPLE

```php
function addSurvey($par,$id_admin){
    /* it makes atomic more queries */
    new sql(" BEGIN WORK ");
    /* this query will insert all information in survey table */
    $query = new sql("INSERT INTO survey (\'name\', nelements, levelcomparison, id_category, id_administrator, ch_name, ch_surname, ch_gender, ch_age) VALUES(\'', \'', \'', \'', \'', \'', \'', \'', \'', \''\');
    new sql(" COMMIT WORK ");
    return true;
}
```

Figure 30 Query to add a new survey (administrator.php)
REFERENCES:


[12] www.decisionlens.com


[14] www.superdecisions.com

[15] Telelogic Focal Point Focus on Product Management, Rational software, IBM