

Exploring MCDA methods with DecSpace

João Pedro Coelho Amador

Thesis to obtain the Master of Science Degree in
Informations Systems and Computer Engineering

Supervisor: Prof. José Luís Brinquete Borbinha

October 2018

Acknowledgments

I would like to start by thanking my supervisor Professor José Borbinha for the opportunity of developing this work and for the continuous support throughout this last year.

I also would like to thank Ana Sara Costa for the help and great suggestions of features and improvements.

To all my colleagues that helped me during the course and dissertation.

Finally, I would like to thank my family and friends as well, who supported me during this last year.

Abstract

Multiple Criteria Decision Aiding (MCDA) is a domain of Operations Research and Management Science mainly devoted to the development of methods, techniques, and tools aiming to help decision makers in complex decision processes. This dissertation has the purpose of creating and developing a web application: DecSpace, a framework to explore a wide range of MCDA methods. Towards this objective, the most appropriate tools and the latest version of DecSpace were investigated in order to understand their biggest features and avoid their biggest imperfections. After the design and implementation phase, DecSpace was evaluated by other users in order to understand if it was accomplishing its objectives. The results and recommendations were applied to achieve the concluding result. DecSpace is a web-based framework, making it possible to explore solutions for problems involving one or more methods. Persistent workspaces also make it possible to reuse solutions. In this sense, DecSpace is intended for use in teaching and researching in MCDA methods, as well for professional use as a Decision Support System for engineering and management scenarios where decisions based on those methods are to be considered. DecSpace has an open architecture, making it possible to use methods implemented locally, fully integrated as part of the local application, or methods available from remote servers, if exposed as web services.

Keywords

Multiple Criteria Decision Aiding; Decision Support Systems; Web Application; User Experience; Interaction.

Resumo

Apoio Multicritério à Decisão (AMCD) é um domínio da Pesquisa Operacional e da Ciência de Gestão, dedicado principalmente ao desenvolvimento de métodos, técnicas e ferramentas com o objetivo de ajudar os decisores em processos de decisão complexos. Esta dissertação tem o objetivo de criar e desenvolver uma aplicação web: DecSpace, uma estrutura para explorar uma ampla gama de métodos MCDA. Para esse objetivo, as ferramentas mais adequadas e a versão mais recente do DecSpace foram detalhadamente investigadas para entender as suas maiores características e evitar as suas maiores imperfeições. Após a fase de design e implementação, esta aplicação web foi avaliada por outros utilizadores para verificar se todos os objetivos estavam a ser cumpridos. Os resultados e recomendações foram aplicados para alcançar o resultado final. DecSpace é uma estrutura baseada na web, tornando possível explorar soluções para problemas envolvendo um ou mais métodos. Espaços de trabalho persistentes também tornam possível reutilizar soluções. Nesse sentido, DecSpace destina-se ao uso em ensino e pesquisa em métodos MCDA, bem como para uso profissional como um Sistema de Suporte à Decisão para cenários de engenharia e gestão onde as decisões baseadas nesses métodos devem ser consideradas. DecSpace tem uma arquitetura aberta, tornando possível usar métodos implementados localmente, totalmente integrados como parte da aplicação local, ou métodos disponíveis a partir de servidores remotos, se expostos como serviços da web.

Palavras Chave

Apoio Multicritério à Decisão; Sistema de Apoio à Decisão; Aplicação Web; Experiência de Utilizador; Interação.

Contents

1	Introduction	1
1.1	Problem Description	2
1.2	Objectives and Results	2
1.3	Document Structure	3
2	Overview of MCDA	5
2.1	Introduction to MCDA	6
2.2	MCDA Methods	7
2.2.1	Full aggregation approach	7
2.2.2	Outranking approach	8
2.2.3	Goal, aspiration or reference level approach	9
2.3	Conclusions	9
3	State of the Art of MCDA Tools	10
3.1	MCDA Tools	11
3.2	D-Sight	11
3.2.1	Main Features	11
3.2.2	Architecture	12
3.2.3	Application	12
3.3	Expert Choice	14
3.3.1	Main Features	14
3.3.2	Application	15
3.4	M-Macbeth	16
3.4.1	Main Features	16
3.4.2	Application	17
3.5	Visual PROMETHEE	18
3.5.1	Main Features	18
3.5.2	Application	19
3.6	Diviz	20

3.6.1	Main Features	20
3.6.2	Architecture	20
3.6.3	Application	21
3.7	Conclusions	22
4	DecSpace Original Prototype	24
4.1	Requirements: Use Cases	25
4.2	Requirements: Architecture and Technologies	26
4.3	Requirements: Domain Model	28
4.4	Available Methods	29
4.4.1	AdditiveAggregation	29
4.4.2	CAT-SD	29
4.4.3	Inquiry	29
4.4.4	Order By	30
4.4.5	Sort	30
4.4.6	SRF	30
5	New Requirements and Analysis	31
5.1	New Requirements	32
5.1.1	Required New Features	32
5.1.2	Required Framework: Vue	34
5.2	Requirements: Use Cases	34
5.3	Requirements: Architecture of the Solution	36
5.4	Requirements: Domain Model	37
6	Solution Implementation	39
6.1	DecSpace v1.0	40
6.2	Project Structure	41
6.3	Front-End	42
6.3.1	Main Components	42
6.3.2	Homepage	43
6.3.3	Sign Up Page	44
6.3.4	Login Page	45
6.3.5	FAQ Page	45
6.3.6	Method Catalog Page	47
6.3.7	My Projects Page	49
6.3.8	Public Projects Page	50
6.3.9	Settings Page	52

6.3.10 Device Mode	52
6.3.11 Workspace Page	54
6.3.11.A Sort Method	54
6.3.12 Administrator Page	57
6.4 Back-End	58
6.5 Database	60
7 Evaluation	61
7.1 System Evaluation Plan	62
7.2 Result Analysis	63
7.3 Conclusions	64
8 Conclusions	65
8.1 Conclusions	66
8.2 Future Work	66
A System Evaluation Guide	71
B DecSpace Survey	74
C Results of the Survey and Evaluation	82

List of Figures

3.1	D-Sight Architecture.	12
3.2	D-Sight Results.	13
3.3	D-Sight Profile Intervals.	14
3.4	Expert Choice Data Sensitivity.	15
3.5	Expert Choice Results	16
3.6	M-Macbeth Results	17
3.7	M-Macbeth Judgments Matrix.	18
3.8	Visual PROMETHEE Results	19
3.9	Diviz Interface.	21
3.10	Diviz Workspace.	22
4.1	DecSpace v1.0 Use Case Diagram.	26
4.2	DecSpace v1.0 Architecture Deployment Diagram.	27
4.3	DecSpace v1.0 Domain Model Diagram.	28
5.1	DecSpace v2.0 Use Case Diagram.	35
5.2	DecSpace v2.0 Architecture Deployment Diagram.	36
5.3	DecSpace v2.0 Domain Model.	38
6.1	DecSpace v1.0 Implementations	40
6.2	DecSpace v2.0 Structure	41
6.3	DecSpace v2.0 Homepage.	43
6.4	DecSpace v2.0 Homepage.	44
6.5	DecSpace v2.0 User Menu.	44
6.6	DecSpace v2.0 Sign Up Page.	45
6.7	DecSpace v2.0 Login Page.	46
6.8	DecSpace v2.0 FAQ Page.	46
6.9	DecSpace v2.0 Method Catalog Page.	47

6.10 DecSpace v2.0 SRF Table Example.	48
6.11 DecSpace v2.0 SRF Figure Example.	48
6.12 DecSpace v2.0 Sort Tutorial.	48
6.13 DecSpace v2.0 My Projects Page.	49
6.14 DecSpace v2.0 Add Project Dialog.	50
6.15 DecSpace v2.0 Public Projects Page.	51
6.16 DecSpace v2.0 Public Projects Preferences.	51
6.17 DecSpace v2.0 Settings Page.	52
6.18 DecSpace v2.0 Device Mode	53
6.19 DecSpace v2.0 Workspace Page	55
6.20 DecSpace v2.0 Workspace Methods Page	55
6.21 DecSpace v2.0 Workspace Boxes	55
6.22 DecSpace v2.0 Workspace Project Menu	56
6.23 DecSpace v2.0 Graphs Page	56
6.24 DecSpace v2.0 Sort Method	56
6.25 DecSpace v2.0 Administrator Files Page.	57
6.26 DecSpace v2.0 Administrator Users Page.	57
6.27 A snippet of the code of the "Login" function.	59
6.28 Database Structure	60

List of Tables

3.1	Features of studied tools.	23
C.1	Average Results of the System Evaluation.	83
C.2	Average Results of the Survey.	83

Acronyms

AHP	Analytic hierarchy process
ANP	Analytic network process
CAPSI	Conferência da Associação Portuguesa de Sistemas de Informação
CAT-SD	CATegorization by Similarity-Dissimilarity
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
DEA	Data Envelopment Analysis
DecSpace	Decision Space
DM	Decision-Maker
DSS	Decision Support Systems
ELECTRE	Elimination and Choice Expressing the Reality
JSON	JavaScript Object Notation
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
MACBETH	Measuring Attractiveness by Categorical Based Evaluation Technique
MAUT	Multi-attribute utility theory
MCDA	Multiple Criteria Decision Aiding
MVC	Model View Controller
OSGi	Open Services Gateway initiative

PROMETHEE Preference Ranking Organization Method for Enriched Evaluation

SOAP Simple Object Access Protocol

TOPSIS Technique of Order Preference Similarity to the Ideal Solution

XMCD Extensible Multi-Criteria Decision Analysis

XML Extensible Markup Language

1

Introduction

MCDA is a domain of Operations Research and Management Science mainly devoted to the development of methods, techniques, and tools aiming to help decision makers in decision processes [1].

It is applicable for solving problems that are characterized as a choice among alternatives, addressing problems where choices, alternatives, and items are evaluated on several conflicting criteria, i.e, decision making systems.

The main purpose of MCDA methods is to support a Decision-Maker (DM) during the decision-aiding process while taking personal preferences into account. The use of this kind of methods can be challenging for those that are not experts in MCDA. The number of MCDA methods and publications are increasing, and with it so does the software available, including spreadsheets containing method computations, web or smartphone applications, contributing to the increased use of these methods amongst researchers and professionals.

1.1 Problem Description

Everyone has to make decisions, we do it on our daily basis, but most problems we encounter are simple problems that don't require a great deal of thinking, and so not much criteria is applied, but when this problem requires the analysis of multiple criteria and the preferences of the decision maker, it is not so easy to perceive a solution by yourself. Decision aiding helps to reach answers to the various questions or problems, offering a recommendation for an action that will elucidate the decision.

It is not easy or affordable for the average user, to learn the concepts of MCDA with the tools that are available. Many software interfaces present the complexity of the methods in a really difficult way, and so the non-expert users don't have the knowledge to use or understand those concepts. The lack of methods available and the low presentation of the results are also problems, most of the frameworks offer one or two specialized methods. Ideally, there would be a solution with all the relevant methods. The lack of visual appealing can be a deadly factor in the market today, most of the software applies really old and outdated technologies. The user wants a visual appealing technology, and that can be a great factor in the decision of what solution to choose.

1.2 Objectives and Results

The fundamental objective is to support a decision maker to structure and to better understand this problem by providing him relevant information about the consequences of his choices, the synergies, and redundancies between criteria, and for this a visualization system that informs the user of the best solutions for the problem is needed. It has to be simple and effective, without requiring many complex decisions, this will aid the user in his choice.

Decision Space (DecSpace) is a framework for the use of MCDA methods, with the clear objective of lowering the complexity of using those methods. The main goals are the ability to reuse existing open reference implementations that already exist for some methods (which is the case for many of them, therefore they do not need to be re-implemented), the provision of an interface that welcomes users from diverse backgrounds and with different levels of experience with MCDA and to provide a web-based framework, making the best use of the cutting-edge related technology [2].

DecSpace is web-based, offering an accessible way to use it from any device with an Internet connection and a web browser, with a friendly interface to any user.

This dissertation started with the analysis and performance of some adjustments to the already implemented version of DecSpace, following this study, it was concluded that a new improved version needed to be implemented, correcting some architectural and implementation problems, as well as using a new framework, the decision came between Angular2, React and Vue, concluding with the choice of Vue.js.

The first step was to prepare the requirements definition along with the analysis and creation of the relevant diagrams for this dissertation: use case diagram; architecture deployment diagram; and the deployment diagram. This preparation served as a guide during the implementation of the software application.

The first phases of this project, executed in parallel, comprised the gathering of the requirements and the familiarity with the technological framework. That was followed by the implementation of the new solution, starting with the core features and homepage, followed by the "FAQ", "Method Catalog" and "Project Management" sections. It then was succeeded by the simultaneous development of the "Workspace", the "Sort" method, the graphical features to support the front-end and the back-end. At the same time, the database, for persistence, was redesigned from the ground up. Finally, it all was revised and readjusted to smartphones and tablets.

The application was evaluated with the help of 20 users, all IST students of different backgrounds (cyber-security, artificial intelligence, multimedia, etc.).

A paper describing this work was accepted and will be published in the proceedings of the Conferência da Associação Portuguesa de Sistemas de Informação (CAPSI) 2018 (the bibliographic reference for that is not yet available).

1.3 Document Structure

This dissertation is composed of a total of eight chapters and three appendixes.

Chapter 1, starts with an introduction to MCDA, the expected problems, objectives and results are also presented stating the reason for this document to be assembled.

Chapter 2, introduces a brief overview of MCDA and an analysis of the most important methods describing the main aspects and classifications.

Chapter 3, presents an analysis on some existing decision support tools, D-Sight, Expert Choice, M-Macbeth, Visual PROMETHEE, and diviz, to end this section a conclusion with all the research is summarized in a table.

Chapter 4, is devoted to the former version of DecSpace, the architecture, main features, technology, and methods from the original solution are analyzed, supplemented by the respective diagrams.

Chapter 5, is devoted to the present DecSpace, this new solution is described for its architecture, technology, and use cases, accompanied by the respective diagrams.

Chapter 6, presents the implementation of the new solution, starting by introducing the changes in the old version, the structure of the code, front-end, back-end and database developments.

Chapter 7, the evaluation process of DecSpace is depicted, as well as the interpretation of its results and a description of the changes that were applied to the system after the evaluation was completed.

Chapter 8, presents some concluding remarks and future work for DecSpace.

The appendixes present relevant information for conducting this dissertation. Appendix A exhibits the system evaluation guide, Appendix B exhibits the system evaluation survey, and Appendix C comprise the results of the evaluation.

2

Overview of MCDA

2.1 Introduction to MCDA

MCDA is useful for facilitating in decision situations that involve several criteria, while the preferences of the decision makers are taken into account. Adopting an MCDA approach is pertinent in several domains, such as healthcare [3], environment [4], finance [5], energy management [6], among many others (see, for example, [7]).

The literature in this field is rich (for the bases of MCDA, [8]; for an overview of the field of MCDA, see [9]; and for research trends, see [10]). Indeed, several researchers have proposed methods and techniques for aiding decision makers to handle decision situations effectively and efficiently, intervening in all phases of the decision process (i.e., representation of the problem situation, problem formulation, evaluation model, and final recommendation [11]).

Generally, a set of objects, actions or alternatives (hereinafter called actions) are assessed according to a multitude of characteristics, also called attributes, considered relevant for the decision situation at hand. In the case of the value sets of these attributes are preferentially ordered, they are called criteria ([12], chapter 9). For instance, one can face decision situations dealing with the choice among actions (e.g., choosing a car, see [13], chapter 6), or the rank of actions (e.g., ranking universities, see [14]), or the classification of them into predefined categories (e.g., classifying countries in terms of governance performance and efficiency, see [15]).

Decision making is present in our lives, from simple decisions to complex decision situations, usually involving conflicting criteria. In particular, for organizations, public or private, in several sectors, decision making is crucial for their business. Thus, managers (or stakeholders) need the support offered by MCDA methods to make effective decisions. Indeed, an MCDA approach can be suitable even for dealing with decisions related to complex systems presented, for example, in industrial processes, public policies, and supply chain management.

An MCDA approach allows to structure and have a better understanding of the decision situation at hand and handle it in a logical and systematic way. In addition, an interaction with the stakeholders enables a transparent process and a deeper knowledge of the whole decision situation. Besides that, having tools to support the decision-aiding process is an added value. In practice, the MCDA methods need to be supported by software tools or Decision Support Systems (DSS). Indeed, having DSS is pertinent to facilitate the collection of the data, all computations and the analysis of the results (including sensitivity and robustness analyses). DSS are relevant to all actors that participate in the decision-aiding process, in the sense that DSS can facilitate the application of the methods and techniques, even to those that do not have experience a priori with MCDA, providing an easy way of modeling the problem, and visualize the data, processes, and results [16]. Besides the applicability and relevance of MCDA for several domains, this reinforces the significance of the development of methods and their appropriate implementation.

2.2 MCDA Methods

It is necessary to understand how decision-aiding process operates and how the methods can be classified, before choosing which types of methods models adapt to problems, thus helping answer questions posed by a stakeholder of a decision process.

As declared in [8], a decision-aiding process consists of four basic steps:

- Representation of the problem situation: the problem needs to be understood and explicitly defined, who holds this problem, why is that a problem and who is accountable for this problem (i.e. identification of the stakeholders).
- Problem formulation: the problem is formalized employing decision support language, the use of natural language may not explicitly describe the problem. In this process, the criteria and actions are identified (which are the objects that are assessed on the criteria set).
- Evaluation model: the information collected in the previous steps is organized, and a formal answer to the problem statement is obtained.
- Final recommendation: assess the output of the model and understand if it should be accepted as the answer to the problem.

Decision aiding helps to reach answers to the various questions or problems, offering a recommendation for an action that will elucidate the decision. MCDA methods can be implemented to suit numerous types of problems [12]:

- Choice problem: select a collection of acceptable actions to decide the single best action.
- Sorting problem: specify and assign each action with similar behaviors or characteristics.
- Ranking problem: order actions by means of ranks (the order can be complete or partial).
- Description problem: describe the actions and their consequences.

Regarding the number of MCDA methods available, the decision maker is faced with the difficult task to choose the decision tool that better fits his/her needs, neither any of the methods are perfect, but with this increased interest and investigation in MCDA methods and the increasing development of new ones, it is essential to classify them into three different groups [17].

2.2.1 Full aggregation approach

A score is evaluated for each criterion and then incorporated into a global score, following an approach where a bad score in one criterion is compensated by a good score on another criterion. Some of the example methods are:

- Analytic hierarchy process (AHP): AHP is based on the motto "divide and conquer". Problems are broken down into sub-problems and solved one at a time. This breakdown is done in two phases of the decision process during the problem structuring and the elicitation of priorities through pairwise comparisons.
- Analytic network process (ANP): ANP is a generalization of AHP which deals with dependencies. In AHP, it is considered that the criteria are independent. If they are not independent, correlated criteria would result in an over-evaluated weight in the decision. The ANP method allows these dependencies, also designated feedbacks, to be modeled; they resemble reality and, as a result, yield more accurate results.
- Multi-attribute utility theory (MAUT): MAUT is based on the main hypothesis that all decision makers try to optimize, consciously or implicitly, a function which aggregates all their points of view. The decision maker's preferences can be reproduced by a function, designated the utility function U , this function is a way of measuring the desirability or the preference of objects, denominated alternatives, however, it is not necessarily known at the beginning of the decision process, so the decision maker needs to create it first.
- Measuring Attractiveness by Categorical Based Evaluation Technique (MACBETH): MACBETH is based on pairwise comparisons entered by the user, applying an interval scale. This method is based on three steps in order to achieve the ranking of the options. As with any MCDA method, the first step is to structure the problem, followed by entering pairwise comparisons into a judgement matrix. If the matrix is adequately consistent, the attractiveness can be calculated, otherwise, the user is obliged to revise their judgements. Lastly, an optional sensitivity analysis is recommended.

2.2.2 Outranking approach

There are no compensable scores, a bad score cannot be compensated by a good one. Two options can have the same score but they can't be compared because their behavior is different.

- Preference Ranking Organization Method for Enriched Evaluation (PROMETHEE): PROMETHEE presents the decision maker with a ranking of actions (choices or alternatives) based on the computation of preference degrees. A preference degree is a score (between 0 and 1) which shows how an action is preferred over another action, from the decision maker's point of view. The method yields into three main steps: i) the computation of preference degrees for every ordered pair of actions on each criterion; ii) the computation of unicriterion flows; iii) the computation of global flows. Based on the global flows, a ranking of the actions will be achieved as well as a graphical representation of the decision problem.

- Elimination and Choice Expressing the Reality (ELECTRE): The outranking methods are based on pairwise comparisons of the actions. Implying that every action is compared to all other actions to achieve a final recommendation. Reaching an advantage to avoid compensation between criteria and any normalization process, which distorts the original data. ELECTRE methods are appropriate when facing decision problems with more than two criteria and if at least one of the following conditions is satisfied.

2.2.3 Goal, aspiration or reference level approach

A goal is defined for each criterion and then the closest option is identified to the ideal goal or reference level.

- Technique of Order Preference Similarity to the Ideal Solution (TOPSIS): The TOPSIS method requires only a minimum number of inputs from the user and its output is easy to understand. The only subjective parameters are the weights correlated with the criteria
- Goal Programming: The main goal is to be achieved while also satisfying hard constraints. This goal is comprised of particular objectives that may be conflicting. The main challenge is the modeling of the problem: to find the goal and the soft and hard constraints while identifying the decision variables, goals and constraints.
- Data Envelopment Analysis (DEA): DEA is applied to estimate the performance of firms or entities which transform multiple inputs into multiple outputs. It is suitable for the use of both private sector firms and public sector organizations. The factor that distinguishes it is that the weights assigned to outputs and inputs are not allocated by users. Furthermore, it does not rely on a standard set of weights for all firms, instead, a different set of weights is estimated by a linear optimization procedure in order to show each firm in its best possible condition.

2.3 Conclusions

This chapter shows a concise description of MCDA and the existing methods. Several domains like healthcare, environment, finance, and energy management can take advantage of this perspective. Understanding the decision-aiding process, the explanation of the main steps, its stakeholders and the numerous varieties of decision situations, is crucial for future developments. All of this contributed and influenced the design of this new DecSpace solution, supporting the decision-aiding process and application of several MCDA methods.

3

State of the Art of MCDA Tools

3.1 MCDA Tools

The application of MCDA methods needs to be supported by adequate software. Numerous software solutions have been implemented to support the application of MCDA methods in real-world decision situations, but very few offer a great solution for visualization purposes.

To really understand the problem, we need to comprehend the visualization tools identified, using the following state-of-the-art web technologies, utilized to achieve the promise of an interactive browser-based system visualization application. Each tool is examined from a visualization representation, meaning that we are looking for the best evaluation of the objectives, architecture, technology applied, and main application functionalities. This analysis helps to figure out what are the required characteristics of these software programs and what are the main aspects that they are lacking, which this new framework tries to overcome, in order to make a very inclusive implementation.

3.2 D-Sight

D-Sight provides a selected amount of Strategic Solutions, with a specialization in the areas of MCDA and user-friendly software development [18]. This system provides responsiveness analyses to easily visualize and measure the impact of changing factors on projects. By developing a unique mathematically validated methodology for multi-criteria analysis and bound it in a state-of-the-art software, it is possible to see all relevant data under the model of a graph or chart or even automatically see every likely scenario.

3.2.1 Main Features

The following represent the main features of D-Sight that are relevant to our software project.

- **Advanced real-time simulation:** It is crucial to simulate all likely situations and immediately observing the results on screen. The visualizations provide the possibility to clearly see the combined output and to make a selection to the desired level of detail.
- **Clear visualizations and advanced analysis:** The capacity to visualize summaries and results of large quantities of data in real time enables an immediate yet precise understanding of the information. Allowing to make the best decision under any circumstances and provide a clear explanation.
- **Online collaboration with experts:** Real-time collaboration combined within the system allows all team members to be working simultaneously, despite their physical location.
- **Exclusive user-friendly design:** Created as an easy yet efficient system to manage and analyze data to support the decision-making process. Following the customer's requirements closely and

continuously upgrading the existing software. Remaining an intuitive online decision-making platform.

3.2.2 Architecture

D-Sight implements standard functionalities such as the methods PROMETHEE I and II, the GAIA visualization tool, sensitivity analysis (including, for instance, the walking weights or the decision maker's brain), and a Group Decision Support System. Providing a standard framework that allows conducting a deep analysis process, with the purpose of finding the best compromise solution, justify decisions and, consequently, save time. D-Sight is programmed in Java. Its structure has been designed following the best practices in programming to guarantee easy maintenance and evolution. The data layer is clearly defined enabling easy access for external applications, integration, and external access have been made easier thanks to the plugin system Open Services Gateway initiative (OSGi) framework. This structure is explained in Fig. 3.1.

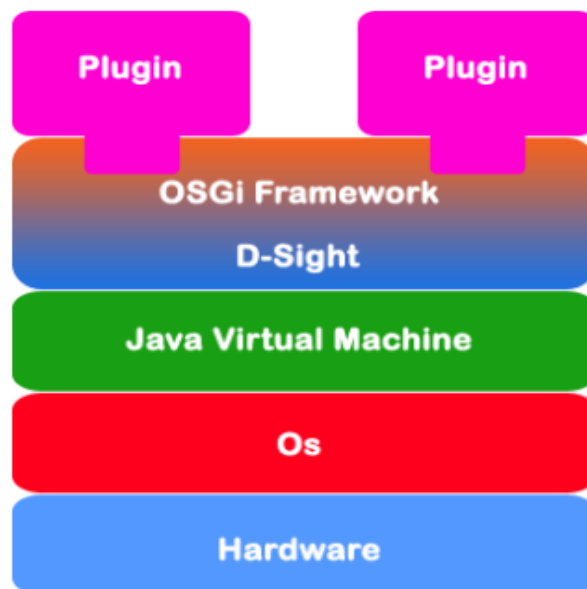


Figure 3.1: D-Sight Architecture.

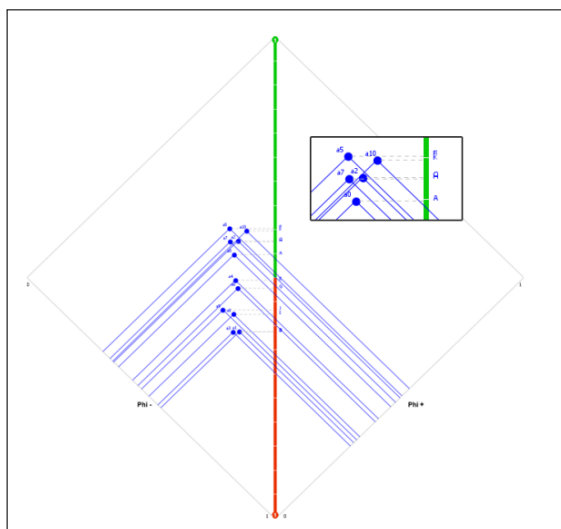
3.2.3 Application

D-Sight is launched by default with the five main inputs panels. The first one is the Evaluations panel gathering all the alternatives (Project A, Project B, ...) and all the criteria.

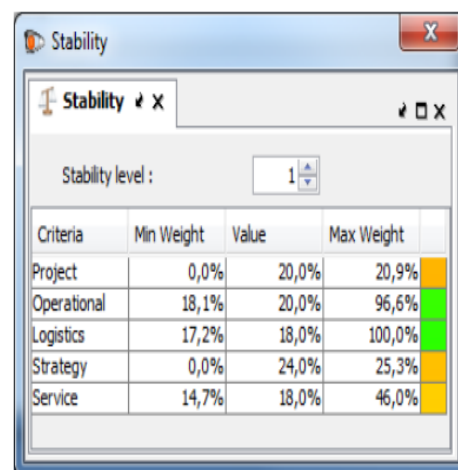
- GAIA analysis: The Global Visual Analysis offers a complete representation of the problem. The

alternatives are represented by the blue points and the criteria are represented by the different axes. The calculation of this chart depends on two things: The alternatives and their whole criteria evaluations and the "intra" criterion parameters representing the approach the decision maker wants to take to evaluate those alternatives each of them for different criteria.

- Profiles: The profiles of the alternatives can be compared in pairs using the Profiles tool Fig. 3.3. The comparison between two alternatives, is for all the level-2 categories, they do not have similar profiles, especially for the Project, Logistics and the Strategy criteria.
- PROMETHEE I Ranking: Represents the alternatives in a diamond. The two axes applied are the negative (red) and positive (green) flows as shown in Fig. 3.2(a). Following the rules of PROMETHEE I, the incomparability between two alternatives can be observed when their projections lines (on Phi+ axis and Phi- axis) are crossing. An alternative is preferred to another when it is above and without line crossing.
- Sensitivity Analysis: For any criterion or category (suggesting the decision maker's preference), the interval in which the weights may switch without affecting the ranking by applying stability intervals. The stability level is the rank to which the stability is measured, indicating that only the first alternative is shown and in which intervals it stays first, despite that, the third alternative becomes second as shown in Fig. 3.2(b). The stability level can be calculated up to n. The bigger the level is, the smaller are the intervals, as constraints are added.



(a) PROMETHEE I Diamond.



(b) Stability Intervals.

Figure 3.2: D-Sight Results.

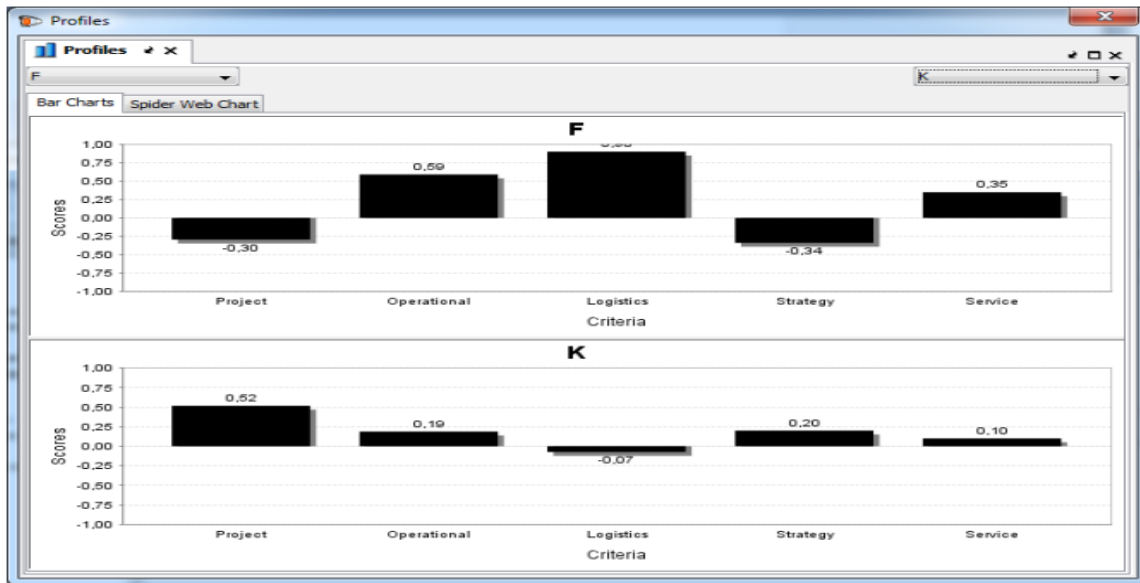


Figure 3.3: D-Sight Profile Intervals.

3.3 Expert Choice

Expert Choice is a real-time decision-making software that consolidates cutting-edge technology with time-tested mathematics to provide a well-informed decision [19]. The objective is to promote a collaborative, straightforward, and accurate approach to complex decision-making, changing complex and mathematically incorrect decision processes into a repeatable, understandable, and valid process. With easy-to-use software and minimal training, leverages decision-making practices that are established to deliver more optimal decisions.

3.3.1 Main Features

Expert Choice has several options in the visualization sector that can be used to view sensitivity analyses results, while focusing on features like establishing explicit decision goals, by using business demands and communicate the decision goal to all the evaluators. Set objectives and provide alternatives to evaluate them. Leverage the expertise of subject matter experts in decision making by designating specific roles. Cooperate either real-time or anytime with teams, partners, and customers all across the world to gather their intuition and hard operational data. Synthesize the team's intuition, and hard operational data easily, using the outcomes to make better decisions to meet demands and implement decisions with more confidence.

3.3.2 Application

The desire is to find an attractive and exciting way to visualize results, Expert Choice has several options in the visualization department that can be used to view sensitivity analyses, using the synthesize menu we are offered four visualization methods:

- **Dynamic Sensitivity:** Used to dynamically modify the priorities of the objectives and see how the priorities of the alternatives would change. To present the dynamic sensitivity based on the decisions made by the participants, it is possible to drag objectives priorities back and forth on the left making the priorities of the alternatives on the right change Fig. 3.4.
- **Performance Sensitivity:** Displays the priorities of the objectives, the performance of each alternative with regard to each of the objectives, as well as the overall priorities of the alternatives. By dragging any of the objective bars up or down, the overall priorities (and possibly their ranks) will change.
- **Gradient Sensitivity:** Presents the rate of change of the priorities of the alternatives with regard to changes in priority of one of the objectives. The curves shown in the Fig. 3.5(a) are the alternatives, the red vertical bar represents the priorities of the alternatives. The chosen objective on the x-axis is the "Cost", this can be changed from the pull-down menu and the alternatives are "Performance" and "Style".
- **2D Analysis:** Presents the performance of the alternatives in Fig. 3.5(b), with regard to any two selected objectives, the axis can be changed to any alternative chosen, between "Cost", "Style" and "Performance".

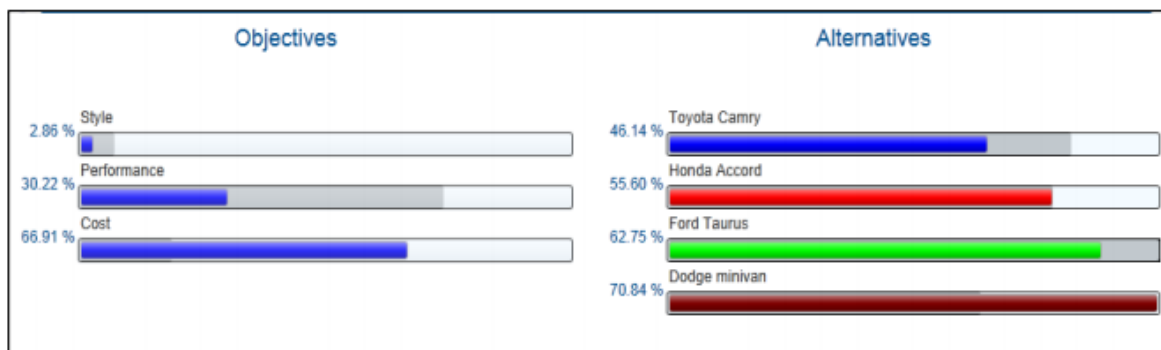
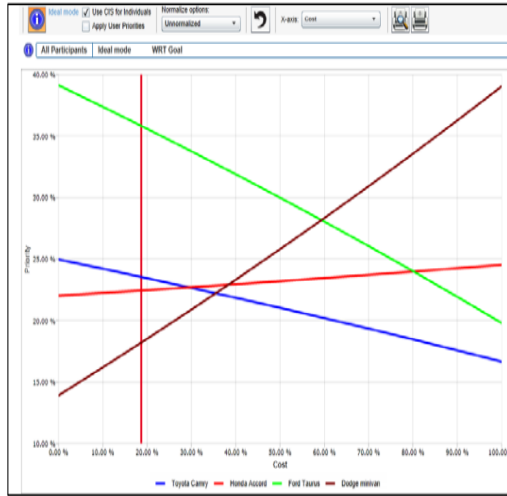
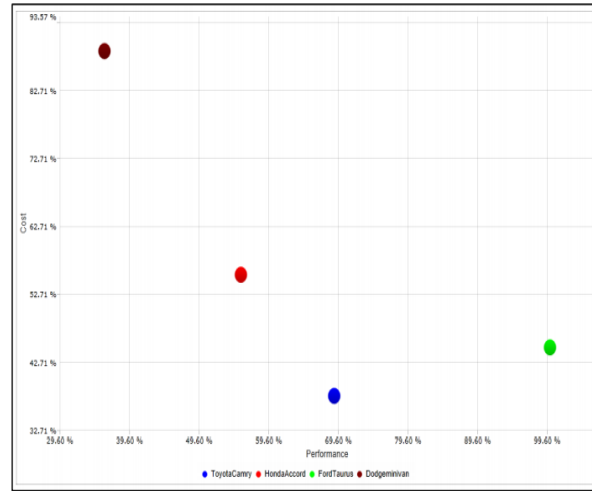


Figure 3.4: Expert Choice Data Sensitivity.



(a) Gradient Sensitivity.



(b) 2D Analysis.

Figure 3.5: Expert Choice Results

3.4 M-Macbeth

MACBETH is a non-numerical interactive approach that demands only qualitative decisions about "attractiveness differences" to support a decision maker or a combination of decision makers, evaluate and quantify the relative attractiveness of the alternatives [20].

This originally and interactively questioning approach where the elements are compared in pairs, requires only a qualitative preference judgment. As judgments are registered into the software, their consistency is automatically verified. As a consequence, it generates a numerical scale that is representative of the decision maker's decisions. Through a similar process, it allows the generation of weighing scales for criteria as well as a sensitivity analysis for each criterion.

3.4.1 Main Features

Macbeth offers resource allocation and a strategic plan development, while developing risk models and scenario analysis. Participative evaluation of social, economic and environmental impacts for major infrastructures, including cost, benefit and risk analysis of projects and programs, and performance evaluation for employees and suppliers.

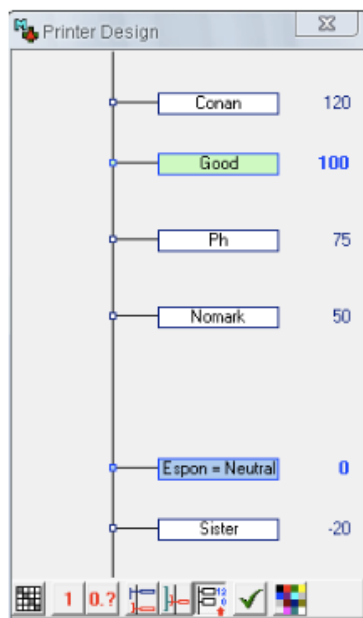
3.4.2 Application

In terms of application, the judgments window Fig. 3.7 is where the comparison judgments between options are inserted in an accessible way, this is one of the main information entry forms.

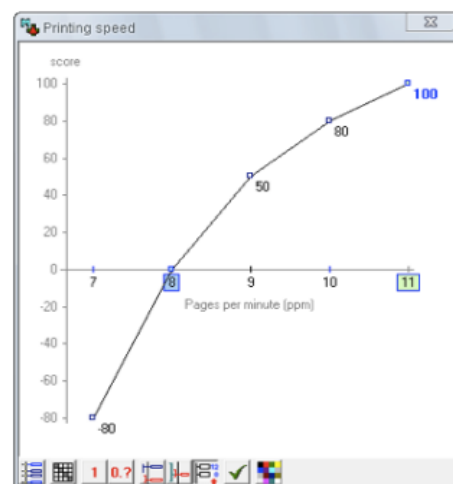
With this information, the software will later determine the options scores between the categories: no, very weak, weak, moderate, strong, very strong and extreme difference of attractiveness. As each judgment is entered into a judgments matrix, their consistency is automatically verified with the judgments previously inserted into the matrix, and suggestions are offered to resolve inconsistencies. The aid process then evolves to a construction of a quantitative evaluation model.

Descriptor of Performances in Fig. 3.6(a), facilitates the automatic scoring of options based upon their qualitative performance in a criterion. The thermometer illustrates each alternative's scores represented in vertical line.

Value functions represented in Fig. 3.6(b), allows an automatic scoring of options based upon their quantitative performance in a criterion. Where the performance levels are plotted on the horizontal axis and the scores on the vertical axis. the linear pieces serve to calculate the score of any option whose performance with respect to the criterion is between consecutive performance levels.



(a) Descriptor of Performances.



(b) Value Functions.

Figure 3.6: M-Macbeth Results

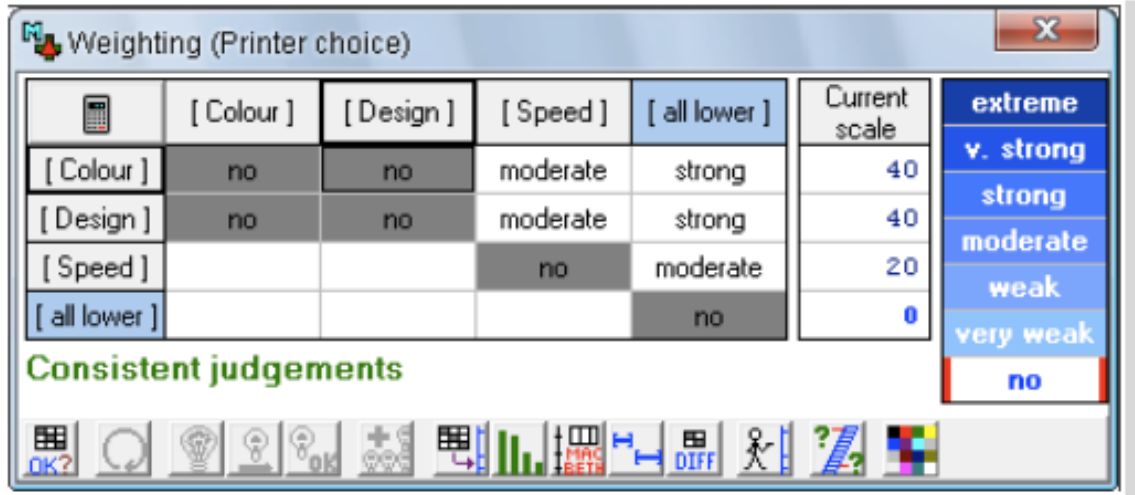


Figure 3.7: M-Macbeth Judgments Matrix.

3.5 Visual PROMETHEE

Visual PROMETHEE is an MCDA software, that offers an intuitive and comprehensive interface, by providing all the data available and editable, it offers a distinctive view of color and shapes to visualize groups of action or criteria, criteria statistics, and smart assistants are available to help the user to assess preferences and priorities, and multiple scenarios can be defined to reflect the points of view of several stakeholders or different hypotheses [21].

3.5.1 Main Features

The following represent the main features of Visual PROMETHEE that are relevant to the project.

- Evaluate various possible decisions or items according to multiple conflicting criteria, and identification of the best possible decision;
- Rank possible decisions from the best to the worst one;
- Sort items into predefined classes such as for instance: bad, medium, good, exceptional;
- Visualize decision or evaluation problems to better understand the difficulties in making good decisions;
- Achieve consensus decisions when several decision makers have conflicting points of view, to justify or invalidate decisions based on objective elements.

3.5.2 Application

Visual PROMETHEE presents some options to visualize information these are some of those options:

PROMETHEE Rankings: The PROMETHEE I and II Rankings window displays the rankings in two separate tabs. The PROMETHEE I Partial Ranking is based on the comparison of the leaving flow (Φ_i^+) and the entering flow (Φ_i^-). The PROMETHEE II Complete Ranking is based on the Φ_i net flow which is the difference between Φ_i^+ and Φ_i^- .

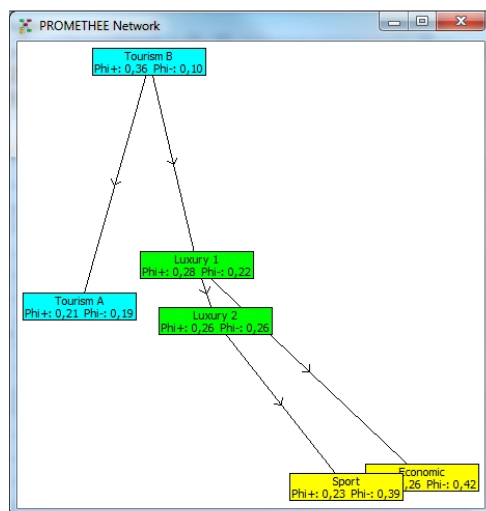
PROMETHEE Diamond: A two-dimensional alternative shared representation of both PROMETHEE I and II rankings. The square corresponds to the Φ_i^+ and Φ_i^- plane where each action is expressed by a point.

PROMETHEE Network: The network representation in Fig. 3.8(a) is a close-up of the diamond view where preferences are indicated by arrows. This makes it very easy to comprehend the proximity between actions and consequently the levels of incomparability in the partial ranking.

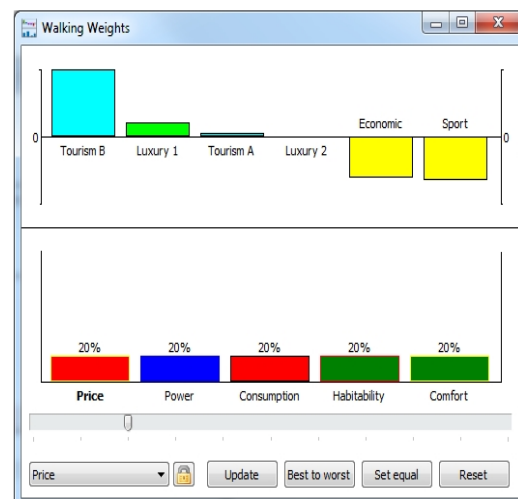
PROMETHEE Rainbow: Exposes the detail of the Φ_i net flow computation, highlighting the good and weak features of each action, and for each action, a different bar is conceived.

Walking Weights: Allows to change the weights of the criteria and see the impact on the Visual PROMETHEE analysis as exemplified Fig. 3.8(b). The upper part is a bar chart showing the PROMETHEE II Complete Ranking and the lower part is a bar chart showing the weights of the criteria.

PROMETHEE Geographical map: The PROMap, provides the combination of the Visual PROMETHEE analysis with Google Maps, where each action can be associated with a geographical location on the map.



(a) PROMETHEE Network.



(b) Walking Weights.

Figure 3.8: Visual PROMETHEE Results

3.6 Diviz

Diviz is an accessible software tool for building, executing and sharing complex workflows formed by MCDA methods, developed by the Decision Deck Project [22].

3.6.1 Main Features

One of the main features is that it supports the construction of academic MCDA methods, by combining several elementary calculation elements, offering the tools to consolidate new decision methods, in every implemented language, without having to rewrite them [22]. This design makes possible to efficiently integrate additional methods without great struggle Fig. 3.9. The design of the MCDA workflows is performed by an intuitive graphical user interface, allowing the user to execute it in order to obtain various possible outputs of the algorithms. These calculations are implemented on high-performance computing servers by using the Extensible Multi-Criteria Decision Analysis (XMCDa) web-services. After the execution of the workflow, the outputs of each component can be viewed and analyzed by the user, allowing for a complete concept of the algorithm, while the parameters can be changed.

Diviz can also be a useful tool to compare the outputs of various methods and algorithms on the equivalent input data, with a feature to export and import any workflow, with or without the data, as an archive, that can then be shared with any other user.

3.6.2 Architecture

In terms of architecture, diviz can be represented by a three-tier application composed of the following tiers: the client (graphical user interface), the component entering the XMCDa web services (reaches the necessary resources and other external platforms) and the server (plans and controls the execution of the workflows, divides the computations through the dedicated web services and returns the output of the executions to the client).

Diviz relies on two additional outcomes of the Decision Deck project:

- XMCDa data standard; The defined language that provides the representation of multiple MCDA concepts and data structures in a defined XML schema, the XMCDa markup language is written in XML, with the objective to support information systems in sharing structured data, particularly through the Internet and to encode documents.
- XMCDa web-services; This web service is an application which can be accessed via the Internet and is executed on a remote system to enable easy access to MCDA algorithms. The inputs and outputs are formatted with XMCDa, this guarantees high interoperable web services. Each

web service reveals a method for submitting a problem and another one for retrieving the corresponding results without having to stay connected, finally each web service can be made of any programming language.

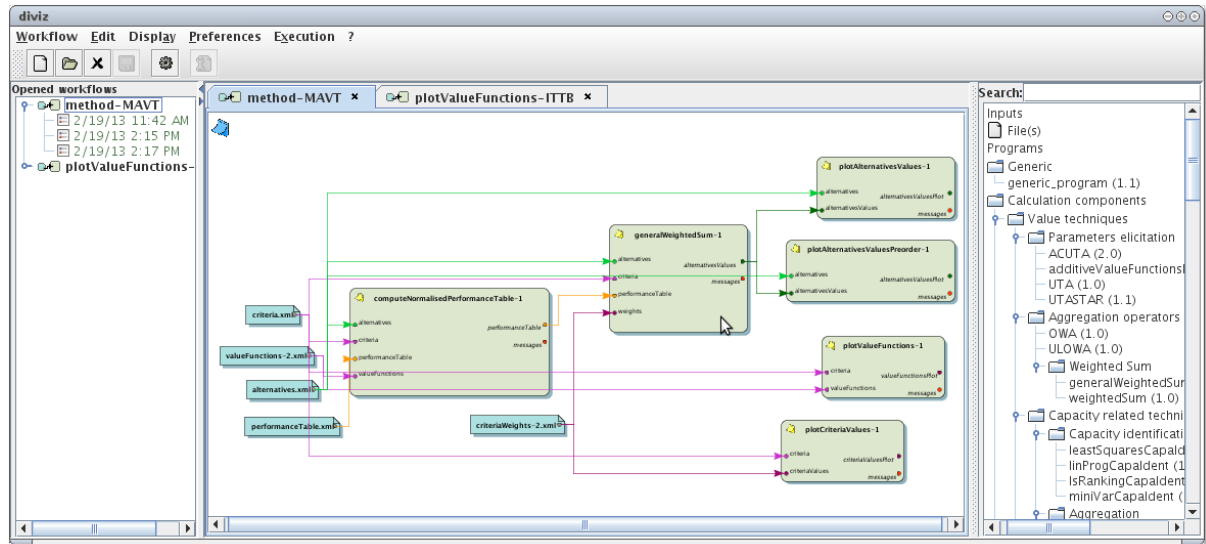


Figure 3.9: Diviz Interface.

3.6.3 Application

In Fig. 3.10, is represented the diviz workspace, this main workspace where all the workflows are created is quite intuitive with a drag and drop feature and a color scheme of the data input and output making it very simple. The right menu represents the list of all algorithmic components that can be dragged to the main workspace, the left menu represents the list of workflows and corresponding executions and in the lower menu the results of the workflow, this last menu can have many different styles to visualize the information like bar charts, tables, bar plots and GAIA plane, it offers a large scale of different visualization tools that really benefits the user. In terms of information on the screen, it may confuse the user, because of the excess of information also it should be welcome to input data manually and not have the necessity of always having an external file with the data correctly structured.

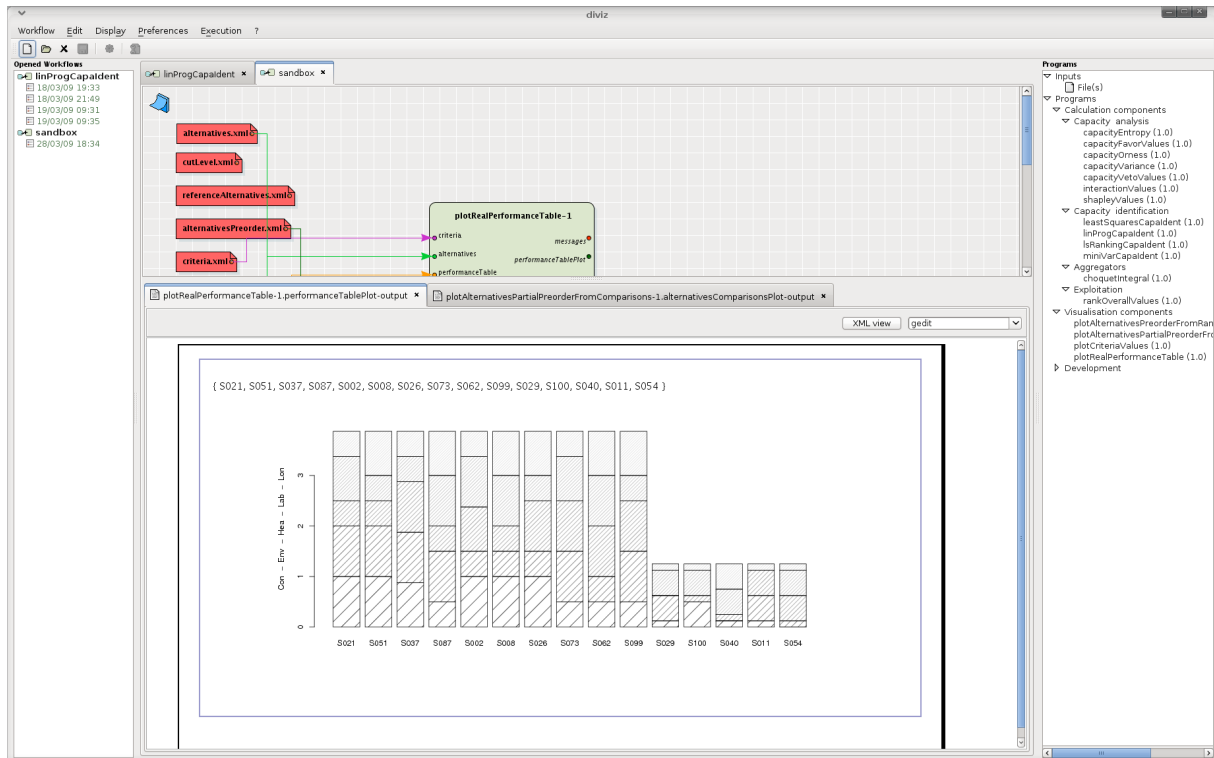


Figure 3.10: Diviz Workspace.

3.7 Conclusions

After an extensive analysis of the five tools taken into account, for being the most solid and better possibilities in the market, many of them have very interesting features but none offers the complete experience, it is important to point that for some tools a license is required (some offer a demo), that is expensive for the average user.

The main features are presented in Table 3.1, represented in six columns, the first column presents the core characteristics that are important for our analysis and the other five represent the frameworks studied [23].

D-Sight is divided into two components, "Web" is a collaborative decision-making platform that aids to solve challenges, analyze data, and manage results, and "Desktop" a dedicated software solution to support decision-making processes. They both are designed as an interactive and intuitive interface. D-sight provides a great deal of visualizations techniques to clearly see the combined output and to make a division to the desired level of detail, it really offers a lot of different visualization tools, but is not open-source and does not allow for easy development of new methods.

Expert Choice is a decision-making software that is based on multi-criteria decision making, and im-

Table 3.1: Features of studied tools.

Features	D-Sight	Expert Choice	M-Macbeth	Visual PROMETHEE	diviz
Open-source					y
Adding new methods					y
Results comparison			y	y	y
3D visualization techniques				y	
Data files import and export	y			y	y
Level of expertise required	2	3	3	3	2

plements Analytic Hierarchy Process (AHP), is a method of multi-criteria evaluation which organizes and simplifies decision-making. Expert Choice has a few visualization methods but with great representation, however, the information is not well organized and the interface feels really outdated.

MACBETH is an interactive approach that requires only qualitative judgments about differences to help a decision maker or a decision advising group quantify the relative attractiveness of options. M-Macbeth offers a really old interface, the way in which the information is presented really look outdated, however, it relies on a non-numerical interactive approach that works.

Visual PROMETHEE is a visualization tool for multicriteria decision aid methods that belong to the family of outranking methods, it offers a very large number of possible ways to see the results and it feels really fast while changing from one visualization technique to another, but the full version is not free and it's not web-based, also it is not possible to share your projects with other users.

Diviz is an open source tool that presents really unique and interesting ways of building workflows and visualizing them, using techniques which are meant to support complex decision aid processes offering interoperability in order to create a coherent ecosystem, on the other hand, it could offer more ways to visualize data and the interface is not very well organized.

4

DecSpace Original Prototype

This chapter starts by describing the first version of DecSpace, analyzed from the ground up, offering a detailed analysis of the implementation, architecture, main features, and technology.

The first version of the MCDA framework was developed in a previous project [2]. It is focused on being a very open web-based interface, easy to use for any user, that explores and offers several MCDA methods with various techniques and features. The biggest purpose is to have great interactivity, usability and to offer a great user experience while maintaining it highly accessible, despite the level of knowledge and experience of the user, establishing an easy process for developers to enlarge the list of available methods are also one of its aspirations.

4.1 Requirements: Use Cases

The use case diagram for DecSpace version 1.0 can be observed in Fig. 4.1.

DecSpace presents four users that define the service. Developers implement and add new MCDA methods to the workspace. The administrator manages projects and users that utilize the framework, having the permissions to delete any object. Anonymous users can test the service to see if the features are adequate to their judgments, although projects are temporary. Registered users are typical users that use the framework to test and implement their own projects, having access to all the features that Decspace offers.

To start using the framework with all the features available, a registration process is mandatory, inputting an email, username, and password. After a successful registration, various pages offer information about the service and MCDA.

Any user has its own project area, where projects can be created, deleted or managed, every project has its own information, project name, creation date, modification date, and, privacy. The public project area provides projects that other users share, any project can be observed but modifications are only allowed after the project is duplicated.

The workspace area is composed around the concept of building workflows. The owner of the project should be able to use and connect different modules so that the results of one method can be applied to another one. Those modules should be MCDA methods, that can be locally implemented by developers or remote methods using Simple Object Access Protocol (SOAP) messages within the diviz server [22]. Also, the input data can be imported in the format of Comma-Separated Values (CSV) and JavaScript Object Notation (JSON), or by importing a zip with other workflows, the workflows can also be exported to other formats. The user should be able to enter data into the MCDA methods manually or by importing data files and connecting them to the method modules.

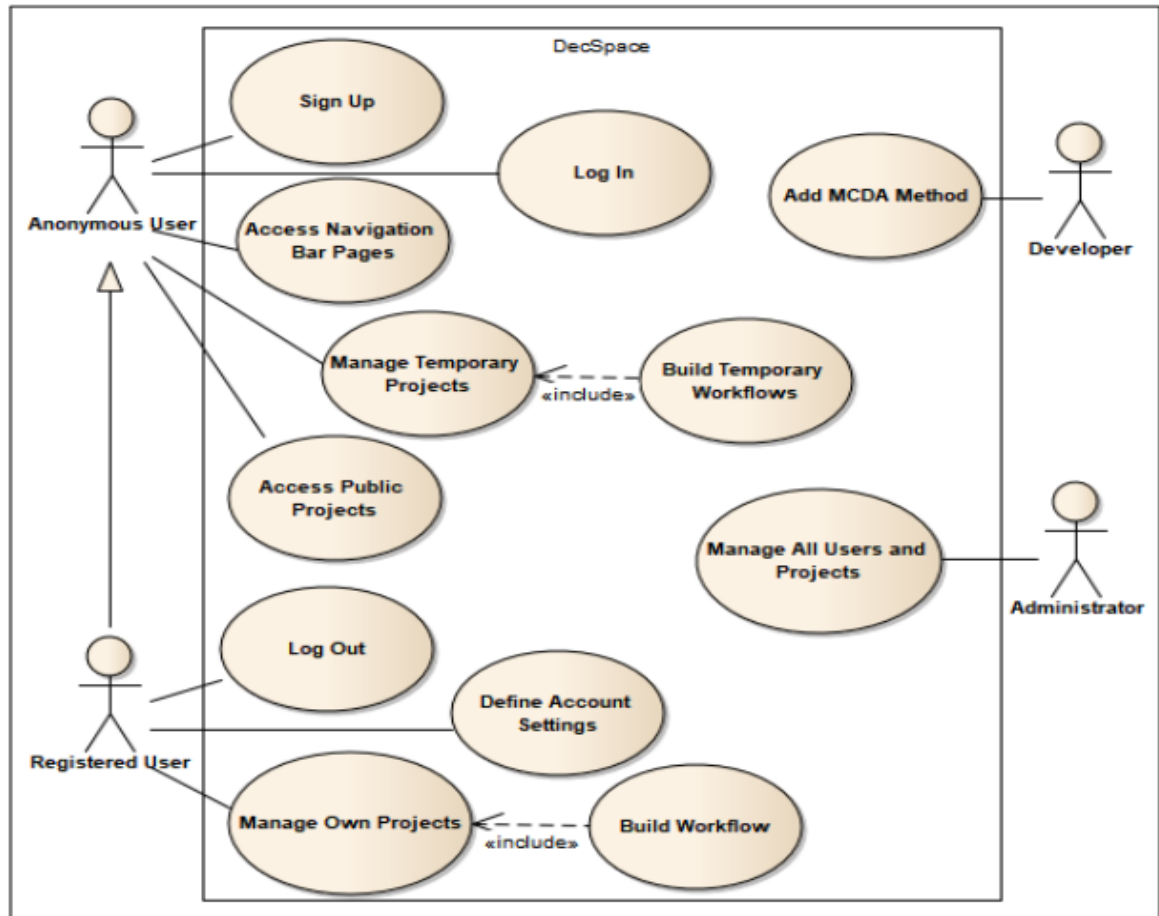


Figure 4.1: DecSpace v1.0 Use Case Diagram.

4.2 Requirements: Architecture and Technologies

DecSpace is designed following a three-tier architecture. Three tier architecture is a variety of client-server architecture, where the project is divided into three fundamental components: presentation tier, application tier, and data tier. The three layers are described below with the technologies used, along with the deployment diagram from the original DecSpace version 1.0, presented in Fig. 4.2.

- Presentation tier; The user interface runs client side only, the main function is to expose DecSpace's features and services to the user in a user-friendly way. It communicates with the application layer sending Hypertext Transfer Protocol (HTTP) requests and receiving a response, which is then processed and translated into information the user can handle.

From the technology used in the presentation tier, AngularJS is a complete JavaScript-based open-source application framework specialized in web front-end development, Hypertext Markup Language (HTML) is used for designing the basic structure of the web pages; Cascading Style

Sheets (CSS) for the visual style of the web pages; JavaScript for advanced handling on the front-end side of the application and jQuery for navigating the document and selecting its elements in order to modify them.

- Application tier; The application tier is where the business logic is established. It operates as a server, as in the client-server architecture, managing the requests received from the multiple clients. This is implemented in a single, standalone web server, which is deployed on server hardware.

This server is built upon two technologies, Node.js, an open source development platform for executing event-driven JavaScript runtime environment and Express.js, a minimal and flexible Node.js framework that provides a robust set of features for web and mobile applications.

- Data tier; The data tier is composed of the database, where all the information is stored. The database software MongoDB, an open-source cross-platform document database that provides high performance, availability and automatic scaling, is responsible for storing data accurately and retrieving any entries requested from the application tier.

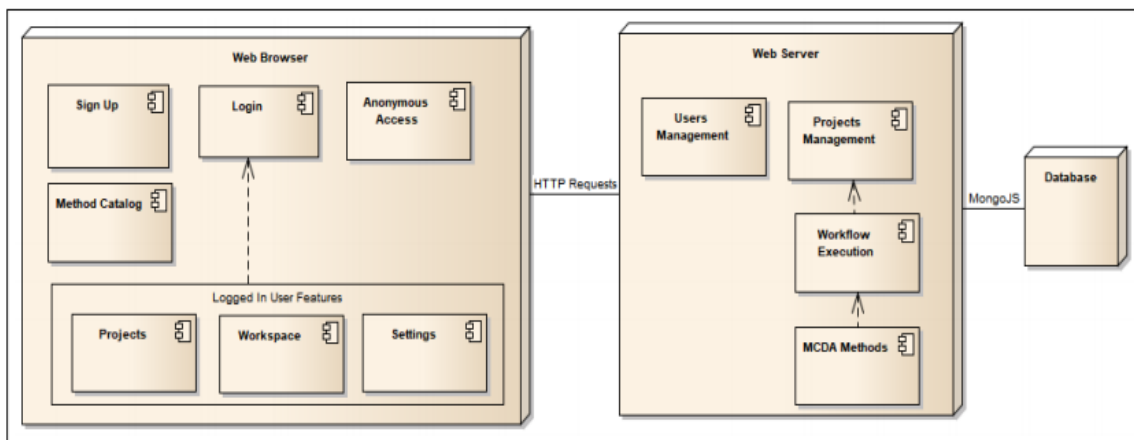


Figure 4.2: DecSpace v1.0 Architecture Deployment Diagram.

4.3 Requirements: Domain Model

The most relevant concepts and the relations between them are retracted in Fig. 4.3.

A user is identified by its unique email address, password, username and privacy setting, for each user is also recorded, the date of registering, date of last login and online status. A user may own various projects, which are distinguished by their singular identifier.

Projects have associated a name, owner, date of creation and date of latest update. A single project may contain several workflows, which are identified by their id as well. They also contain the date there were saved and, a save name made by the user.

A workflow consists of various modules, which may be connected with each other. A module can be of two kinds: method or data. Method modules are uniquely identified by the attribute "type count" and, as of the last version, there are six different types of local method modules that correspond to the available MCDA methods see Chapter 4.1.4, there is also a remote method that replicates the methods in diviz. The attributes of these methods correspond to their input data.

Data modules are simply distinguished by their name and include the data that was imported by the user or generated by a method module.

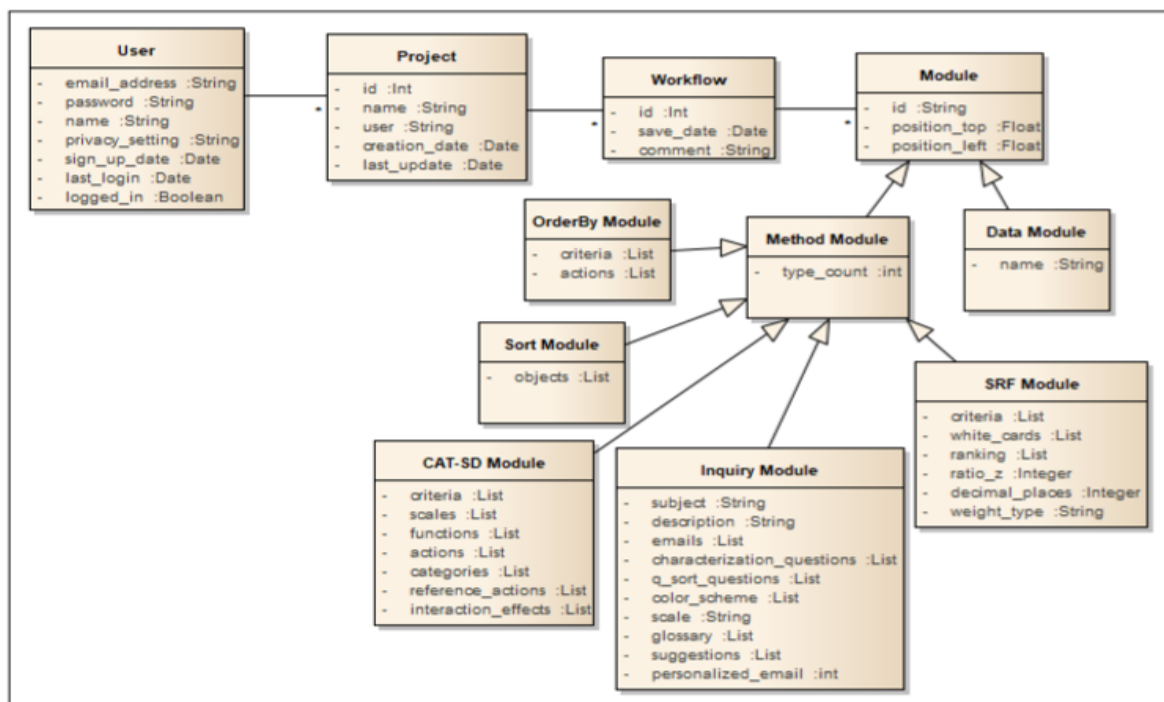


Figure 4.3: DecSpace v1.0 Domain Model Diagram.

4.4 Available Methods

DecSpace supports and has implemented six local methods by developers, these methods represent real-world problems that can be applied by users.

4.4.1 AdditiveAggregation

AdditiveAggregation employs an additive value function to evaluate an option according to the weights associated with each criterion [24]. In the DecSpace method, the criteria are established along with their weight. When analyzing the given options, an aggregated score is calculated, which consists of the sum of the product of each criterion value by their weight. The best option, according to this method, would be the one with the highest aggregated score.

4.4.2 CAT-SD

CATegorization by Similarity-Dissimilarity (CAT-SD) is an MCDA method for nominal classification problems [25]. This method aims at assigning actions into predefined and nominal categories (no preferential order exists among them), based on the concepts of similarity and dissimilarity to designate actions to nominal categories. Each category is characterized by the set of reference actions most representative of that category and likeness degrees are determined between the actions and the reference actions. An action is assigned to a certain category if it is similar enough to the reference actions representative of that category.

4.4.3 Inquiry

Inquiry method is a simplified approach of the Delphi method, this method objective is to reach an agreement from a group of selected experts on certain topics, by using a series of intensive surveys distributed with significant intermissions of time between them [26].

The Inquiry method deviates from the Delphi method, in the way that Inquiry does not concede consecutive rounds with the same module while being able to consolidate the new data in the new round. Inquiry uses the Q-Sorting technique when assembling the results from the experts. Q-Sorting is a process with three stages: study design, administration, and data analysis. In the end, the experts answer their surveys and with the answers a global score is calculated, involving the total score, average score, and standard deviation, the answers can be given from a scale from "least important" to "most important".

4.4.4 Order By

A very simple algorithm, that orders an input list of actions, by a certain criterion, described by a name, number or text, ascendant or descendant order, this criterion needs to be chosen by the user. Then the actions need to be inserted, by providing an action name and values.

4.4.5 Sort

Given a list of actions, characterized simply by their ordered names, the method creates as an output the initial list of actions, in the order that was chosen. For that is provided a simple drag-and-drop interface - the square that represents the inputted actions can be dropped on top of each other, effectively switching places.

4.4.6 SRF

SRF is an implementation of the revised Simos' procedure for the determination of the weights of criteria in the ELECTRE family of MCDA methods, proposed in [27]. It has been successfully used in various real-world decision scenarios in different areas (for a review of applications of the original and revised versions of Simos' method, see [28]). The application of this method starts by associating a "card" with each criterion. The decision maker has to handle the cards in order to rank them from the least to the most important, while also inserting white ones in order to increase the difference of importance of two successive criteria (or sets of criteria, if they are positioned in the same level). For determining the weights of the criteria it is necessary to provide the list of criteria, the ranking of the criteria and white cards, the parameter ratio between the importance of the most important subset of criteria and the importance of the least important one, the number of decimal places (it can be set to 0, 1 or 2), and the type of weight value (normalized, non-normalized or both).

5

New Requirements and Analysis

This chapter focus on the new solution, the details of the new requirements and features changes, the current implemented technologies, the new redesigned architecture, and infrastructure, and lastly the new domain model.

5.1 New Requirements

DecSpace architecture was rewritten from the ground up, to correct some errors and to enhance a better experience, the technologies changed to offer the most advanced and innovative frameworks, most of the code was not reused, which gave the opportunity to emphasize new positive aspects and efficiently implement the new usability and structural ideas. In this section, the new solution is analyzed from different perspectives, taking into consideration the architecture, features, and technologies.

5.1.1 Required New Features

This solution will be a progression over to DecSpace version 1.0, the implementation was successful, consequently, it has the need to expand and to become more robust while implementing new features and ideas.

Requirements are divided into three categories, the first category (G1) is to ensure the new architecture and interface is completely implemented, using the new Vue.js framework, improving the user experience. The second category (G2), will focus on adding new features, improving the method catalog, adding graphs and device modes experiences. Subsequently, the last phase (G3) will focus on implementing some of the required methods and connecting to external methods and improve and correct all the bugs and functionalities. A detailed list of objectives is presented:

- G1 - New Infrastructure and User Interface

G1.1 - New Core Application; The most critical goal, it is essential that the new architecture is rightly implemented, the front-end, back-end, and database need to be correctly connected.

G1.2 - New Image; The new image needs to be visually attractive, to old and new users.

G1.3 - Usability Improvements; Offer the best experience possible in terms of usability.

G1.4 - Error Reporting; Improve the error messages presented to the user, so that the users never feel lost in whatever they wish to do.

G1.5 - Privacy Settings; Give the possibility to adjust at any time the privacy setting of a project to private/public. Privacy setting for groups: give the possibility for a user to public his project to a restrict number of users.

G1.6 - Help and Tutorials; Implement tutorials for more inexperienced users both in DecSpace or MCDA, these tutorials can be implemented as step by step guides in the form of images with accompanying text.

G1.7 - Database; For the deployment of the new version of DecSpace it's necessary a local solution that is reliable.

- G2 - New Features and Improvements

G2.1 - Method Catalog; The catalog needs to be more robust, to present new options like references for each method, more information, tutorials.

G2.2 - Visualization of results; It is crucial nowadays to have a different way to show results instead of numbers or tables, there are many representation views that can be used to engage the user.

G2.3 - Gadgets; Create a better framework to work with smartphones and tablets. More touch, less keyboard, import and edit data with a more interactive approach.

- G3 - New Methods and Improvements

G3.1 - Management of Local Methods; Implement and increase the number of MCDA methods, DecSpace was designed with the purpose of having great user interaction and extensibility characteristics.

G3.2 - Management of External Methods; Implement an interface that communicates with external servers so that MCDA methods can be executed outside of the application, a suggestion is to implement the server of diviz, also implementing XMCDa in our framework.

DecSpace implementation uses the Mean Stack approach, instead of employing Angular a decision to use Vue.js was made, using the combination of the following programming languages and libraries that synergize well together, which simplifies and accelerates the development of web applications:

- MongoDB¹: It is an open-source cross-platform document database that provides high performance, availability and automatic scaling. MongoDB supports dynamic schema design, allowing the documents in a collection to have different fields and structures;
- Node.js²: It is an open source development platform for executing event-driven JavaScript runtime environment, specially designed for building scalable network applications;
- Express.js³: It is a minimal and flexible Node.js framework that provides a robust set of features for web and mobile applications, to help organize your web application into a Model View Controller (MVC) architecture on the server side;

¹<https://www.mongodb.com/>

²<https://nodejs.org/en/>

³<https://expressjs.com/>

- Vue.js⁴: a progressive framework for building user interfaces, Vue is designed from the ground up to be incrementally adoptable.

5.1.2 Required Framework: Vue

For this new version of DecSpace, it's imperative to start working with a modernized and renewed framework, three options were taken into account to offer the best possible interface. Vue is the dark horse of JavaScript Frameworks created by a team of dozen developers while Angular and React have been popularly used by giant companies like Facebook, Reddit, Airbnb, Netflix, and Google.

They offer various advantages and disadvantages between themselves, ReactJS⁵ has the average size between the three options, also with a medium learning curve, it offers a state management store called Redux that helps writing applications that behave consistently and run in different environments. Angular2⁶ has the biggest size among the three options, with the longest learning curve, because it uses TypeScript, Angular is perhaps the most used framework so there is a lot of information. VueJS has the smallest size among the three options and the smallest learning curve, making it perfect for software projects with fewer developers, the documentation is very well written, the component-based architecture suits perfectly in the project scope, and the fast coding speed and reusing CSS code make it the best option to choose.

5.2 Requirements: Use Cases

The following DecSpace use cases is analyzed with the support of one diagram, Fig. 5.1 describes the main features, there are definitely more cases but they were hidden to enhance the most important ones.

- Types of Users: There are four types of users with different permissions: the developer has the objective of adding new methods; the administrator manages all registered users and projects; finally, registered users and anonymous users, the difference between them is that registered users have an account and their work is persistent;
- My Projects: This area allows the user to manage projects, to create a project, and to choose if the project is public or private, the following fields identify each project: Name, Creation Date, Last Update and Privacy Setting, for each project, open, duplicate and delete features are available;
- Public Projects: This area contains all public projects that are shared by users, the following fields identify the projects: Project Name, Username, Creation Date and Last Update, for each project, a

⁴<https://vuejs.org/>

⁵<https://reactjs.org/>

⁶<https://angular.io/>

duplication, and open feature is available, the purpose of this area is to provide complete projects to all users;

- **Method Catalog:** All the information for the methods created are in this area, they are divided into two categories, local methods, created by the developers of the framework, and remote methods, through SOAP requests between the diviz server, it is possible to reuse all their different methods. Each method presents some relevant information, a practical example for guidance and a step-by-step tutorial, for the most inexperienced users;
- **Workspace:** This is where all the technical work happens. The objective is to have a simple working area, that executes users choice methods as fast and responsive as possible, as far as features go, it should be possible to import and export a CSV, XMCD or JSON file or use locally stored workflows, these can be saved, refreshed, executed and deleted. Any method on the catalog is ready to be used, they can be dragged to the workspace and connected between themselves.

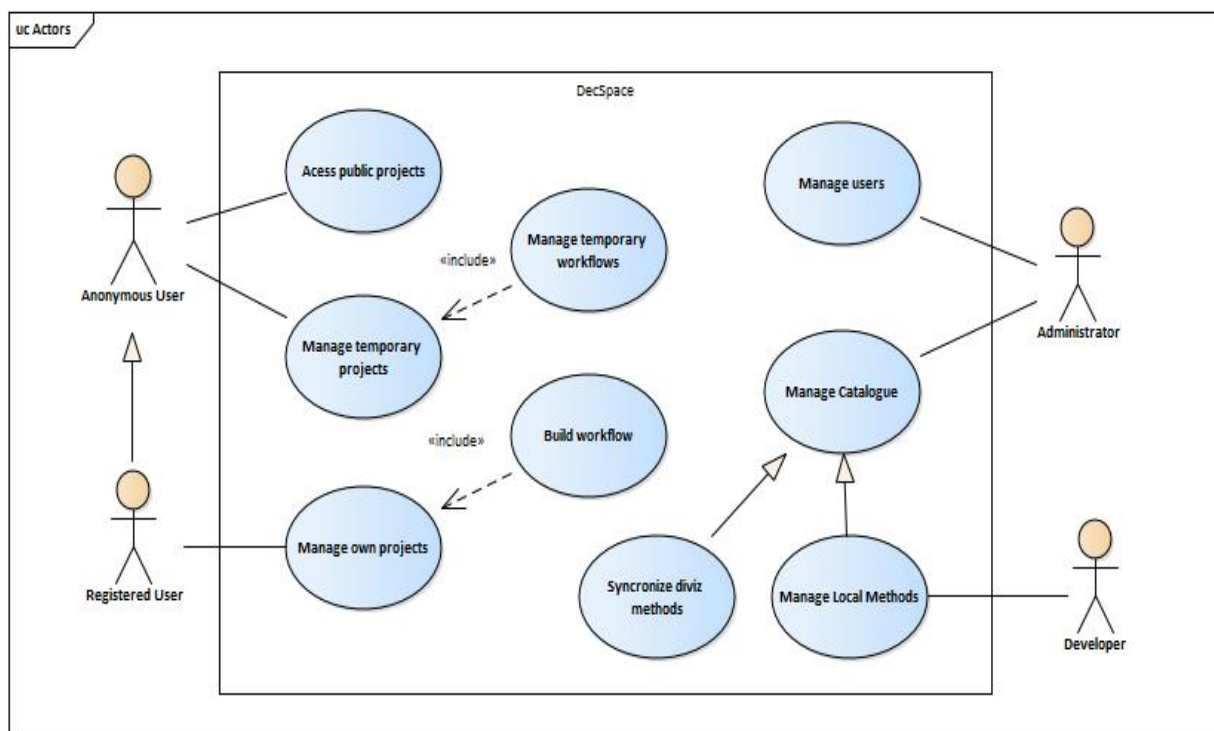


Figure 5.1: DecSpace v2.0 Use Case Diagram.

5.3 Requirements: Architecture of the Solution

The proposed architecture is structured into three main tiers as presented in Fig. 5.2. The objective is to confine most of the complexity to a single tier and restrain the interface details to another tier, which allows focusing on the usability and user interface:

- Client tier: The client tier consists of an interaction with the application tier to send user requests and after these requests are processed a response is returned to the client, which allows the user to perform the most various tasks the system has to offer. In terms of infrastructure, the interaction with the web server is done with simple HTTP requests;
- Application tier: The application tier defines the great complexity of the system, where most of the computational activity is performed. It acts as the server for client requests, by receiving and processing user requests sent by the client tier, and also carrying out the connections to the data tier. The location is a web server that executes the workflows and several MCDA methods;
- Data tier: The data tier is an entity that contains all the information of the application. It communicates with the application tier by receiving, processing and replying to the data requests. It is formed by a database and communicates with the application tier with the aid of the MongoJS library.

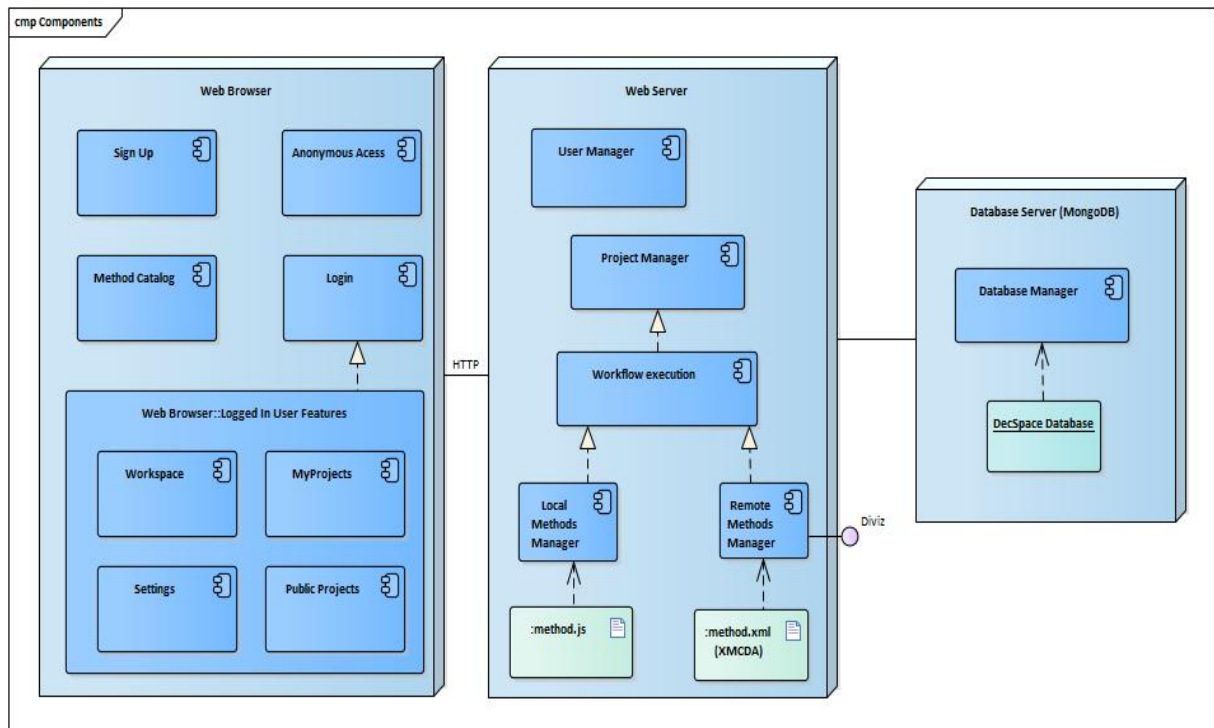


Figure 5.2: DecSpace v2.0 Architecture Deployment Diagram.

5.4 Requirements: Domain Model

The most consistent concepts and the relations between them should be described to adequately understand the necessity for this application, as represented in Fig. 5.3.

This application starts with users, which are identified by its unique email address and must have an associated password, username, privacy setting, date of registering, date of the last login are stored. This information is used to create an account, if all the necessary data are correctly introduced the account is created, if not an error message will help the user correct the error.

To access projects and information about personal user account it is necessary to be logged in, every registered user can own multiple projects and read multiple public projects. Each project is identified by its id, project name, owner, date of creation and last update date, it should provide the create, open, duplicate and delete features.

A single project may incorporate several workflows, which are identified by their id as well. They also contain the date in which they were saved and a save name entered by the user. A workflow consists of various modules that can be connected between them.

A connection is identified by its name and has both a source module and a target module.

A module has two types: data, input module or output module, or a method module, which are identified by the attribute method parameters, that differs for each respective method, and name, as of the last version, there are six different types of local method modules that correspond to the available methods (Additive Aggregation, CAT-SD, Inquiry, Order By, Sort, and SRF), there is also a remote method that corresponds to all available methods in diviz, the contents of this remote method are performed in XMCDa.

The method catalog contains all the methods for the users to choose the preferred ones.

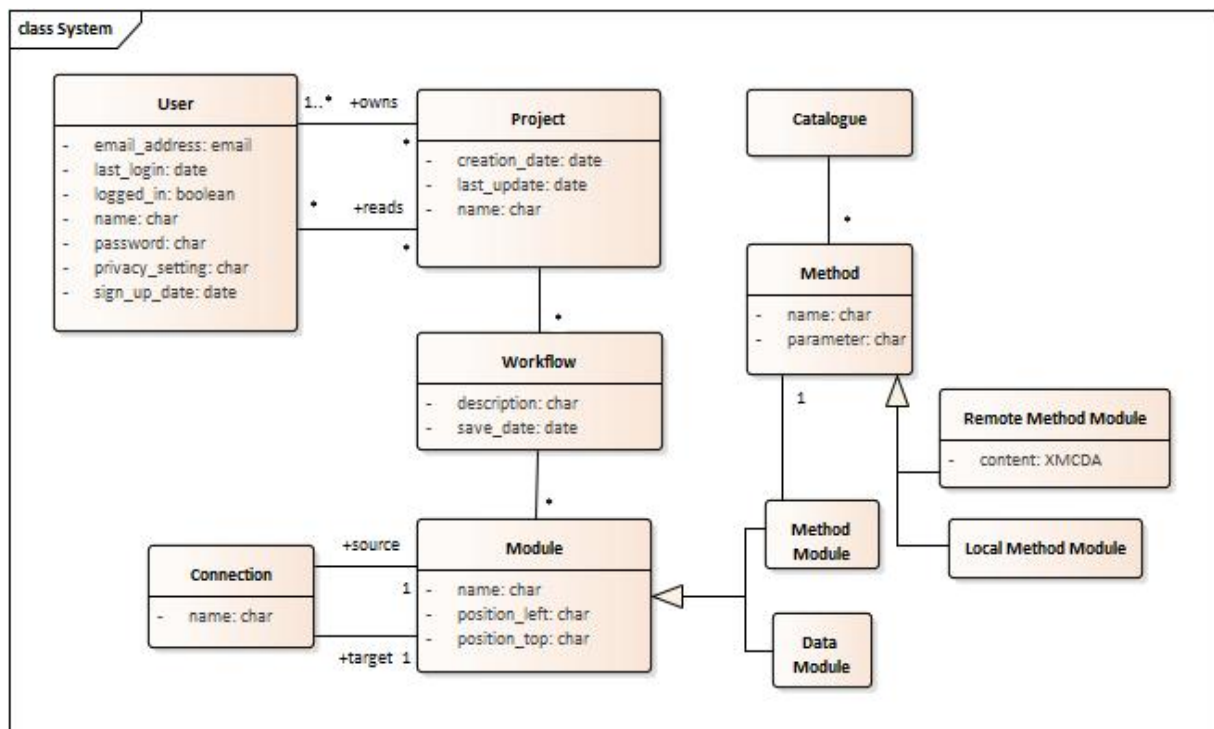


Figure 5.3: DecSpace v2.0 Domain Model.

6

Solution Implementation

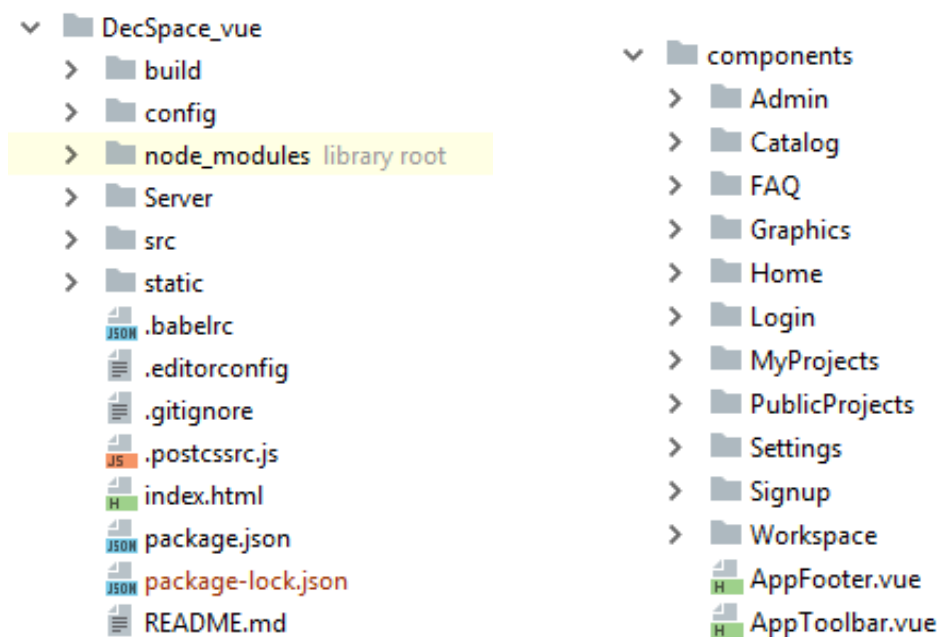
6.2 Project Structure

DecSpace code is structured as illustrated in Fig. 6.2, all the folders are separated depending on their relevance for the software project. The main folder "DecSpace_vue" stores all the information about the project. The "build", "config", "static" folders and all the other packages are automatically included by Vue.js to facilitate the usage of the core functionalities.

The "node_modules" folder simply contains the scripts of the various NPM packages that are automatically installed in the library root.

The folder "Server" contains all the functions that should be performed on the back-end side of the application, e.g. start the server, connect to the database, among many others. It also includes the models for the user base, user, and project.

The folder "src" is where all the client-side of the application is deployed, i.e. in the web browser. This folder is comprised of the "assets" folder, all the images used in the application are stored where, the "auth" and "store" folder, this is where the authentication and storage of local variables are done, the "components" folder, contains all the important Vue.js pages, the "router" folder ensures the connection between pages and finally the "services" folder is the connection between the front-end and the back-end.



(a) DecSpace v2.0 Folder Structure

(b) DecSpace v2.0 Components Structure

Figure 6.2: DecSpace v2.0 Structure

6.3 Front-End

The front-end of the application is comprised of components, each component represents a website page or feature, each file is separated in different concerns, HTML, JavaScript, and CSS.

The main goal is to attract different users, to an innovative framework that is responsive, very quick, secure, that adapts to all devices like smartphones or tablets, and as an innovative yet simple interface.

The concept of this final design decisions went to a lot of different ideas, different prototypes and decisions, and lastly through user tests and suggestions, concluding in the application provided. The following sections describe the detail of every single component in DecSpace.

6.3.1 Main Components

The App.vue file is where global components are defined, in this case, the application toolbar, the router-view and the footer bar are all defined here, as well as the style for the application.

The main.js file imports all the packages used and creates the application, the most relevant packages are: 'vue'¹ allows the integration of Vue.js, 'vue-resource'² a plugin that provides services for making web requests and handle responses, 'vue-session'³ a simple plugin for session storage, 'vuex-router-sync'⁴ allows to synchronize both router and storage, 'vuetify'⁵ a semantic component framework for Vue with the objective to provide reusable components, 'v-dragged'⁶ directive plugin for drag event detection and 'vue-chart.js'⁷ for reusable chart components.

The store manages local variables that need to be stored, using 'vuex'⁸ a centralized store for all the components in an application, with rules ensuring that the state can only be mutated in a predictable fashion. The variables stored are token (indicates if the user is online or offline), username, email, project name, messages for the login, sign up and changing the password.

The router deeply integrates with Vue.js core to make building single page applications easier, the path, name, and component of each application page are defined here.

The services provide HTTP client connections for the browser and node.js using 'axios'⁹.

¹<https://www.npmjs.com/package/vue>

²<https://www.npmjs.com/package/vue-resource>

³<https://www.npmjs.com/package/vue-session>

⁴<https://www.npmjs.com/package/vuex-router-sync>

⁵<https://www.npmjs.com/package/vuetify>

⁶<https://www.npmjs.com/package/v-dragged>

⁷<https://www.npmjs.com/package/vue-chartjs>

⁸<https://www.npmjs.com/package/vuex>

⁹<https://www.npmjs.com/package/axios>

6.3.2 Homepage

The Homepage is the user first impression and contact with the application, it gives a global presentation of DecSpace while introducing the user to the most basic features. The color scheme and layout have been carefully studied.

The information is presented in one-page style, with the logo in shades of blue representing the colors of the application (this is not a definitive logo), the title "A multi-criteria decision analysis framework" explains what is DecSpace is all about, and the subtitle "Many more features to come!" represents the improvements being made, the button "Try It Now" allows for unregistered users to experience the application, as depicted in Fig. 6.3.

The second part of the Homepage explains in more depth some features of the application, together with three different icons complement the information about characteristics. The last piece of the layout are the contacts and information about DecSpace, as depicted in Fig. 6.4. At all times a footer with the year of production and the brand name is displayed.

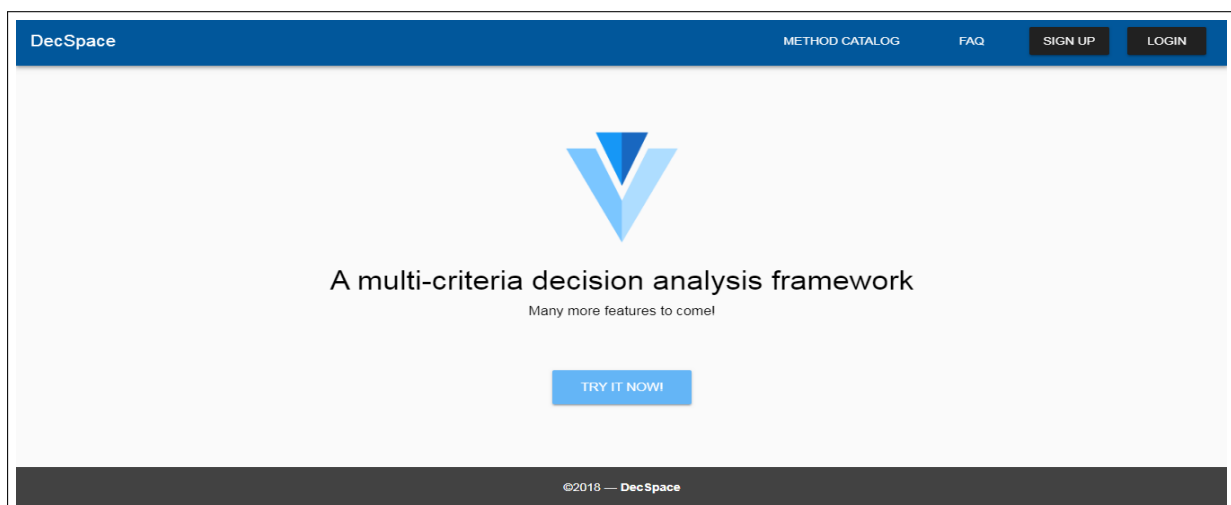


Figure 6.3: DecSpace v2.0 Homepage.

The toolbar connects all the application, the DecSpace name appears with a different type of font and redirects the user to the main page, there are other links to the FAQ, Method Catalog, Login and Sign Up page. If a user is authenticated, the respective menu is presented, this user menu provides access for all project actions, and is composed by an avatar and username, and three links with the respective icons for My Projects, Public Projects, and Settings, as shown in Fig. 6.5.

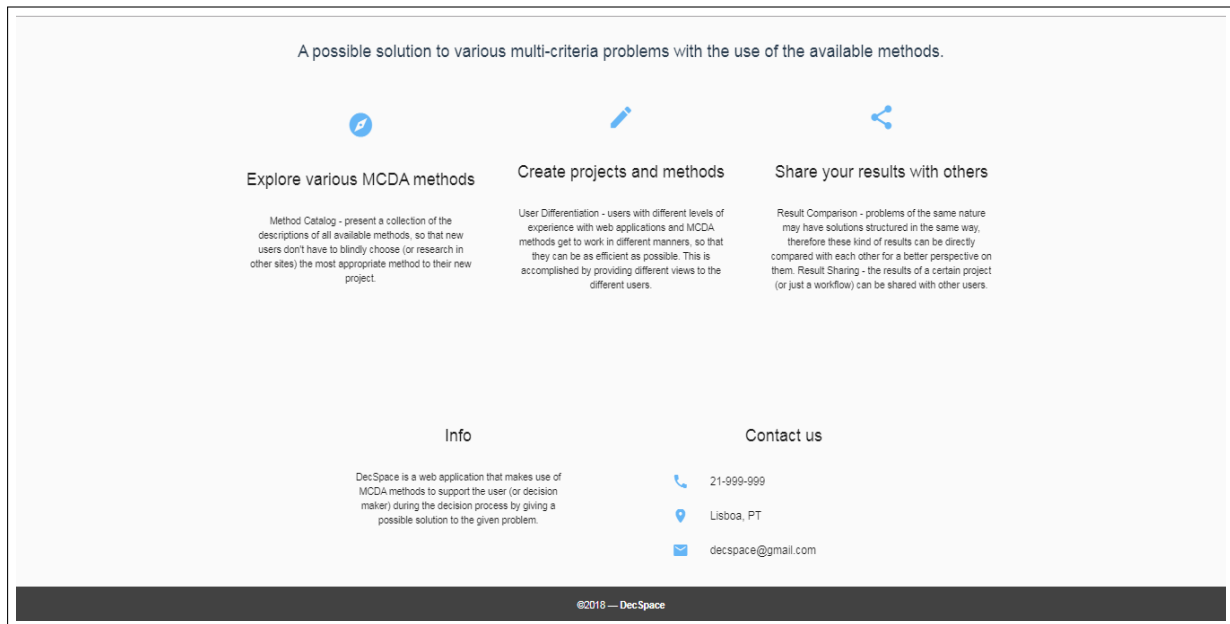


Figure 6.4: DecSpace v2.0 Homepage.

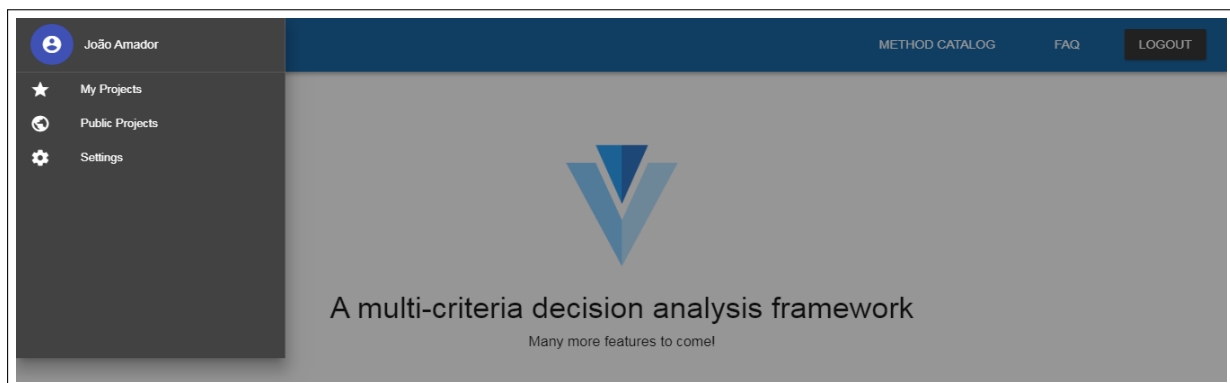


Figure 6.5: DecSpace v2.0 User Menu.

6.3.3 Sign Up Page

The Sign Up page allows the creation of an account to access the main features of DecSpace, the form is very simple with only three required fields, "Username", "Email" and "Password". The page interacts with the user, highlighting the chosen fields, and displays error messages like "Invalid Password" or "Invalid Email", or success messages like "Successfully Sign Up", this page is illustrated in Fig. 6.6.

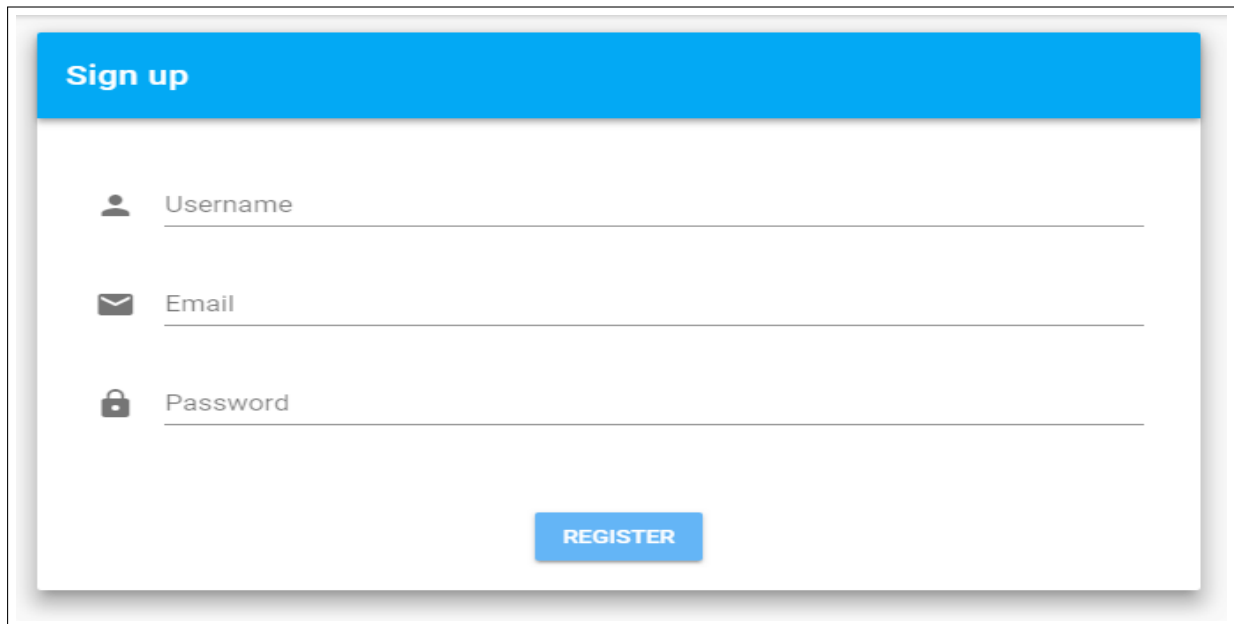
The image shows a web form for signing up. At the top, there is a blue header bar with the text "Sign up" in white. Below the header, the form has three input fields, each with a small icon to its left: a person icon for "Username", an envelope icon for "Email", and a lock icon for "Password". Each field is followed by a horizontal line for text entry. At the bottom center of the form, there is a blue button with the word "REGISTER" in white capital letters.

Figure 6.6: DecSpace v2.0 Sign Up Page.

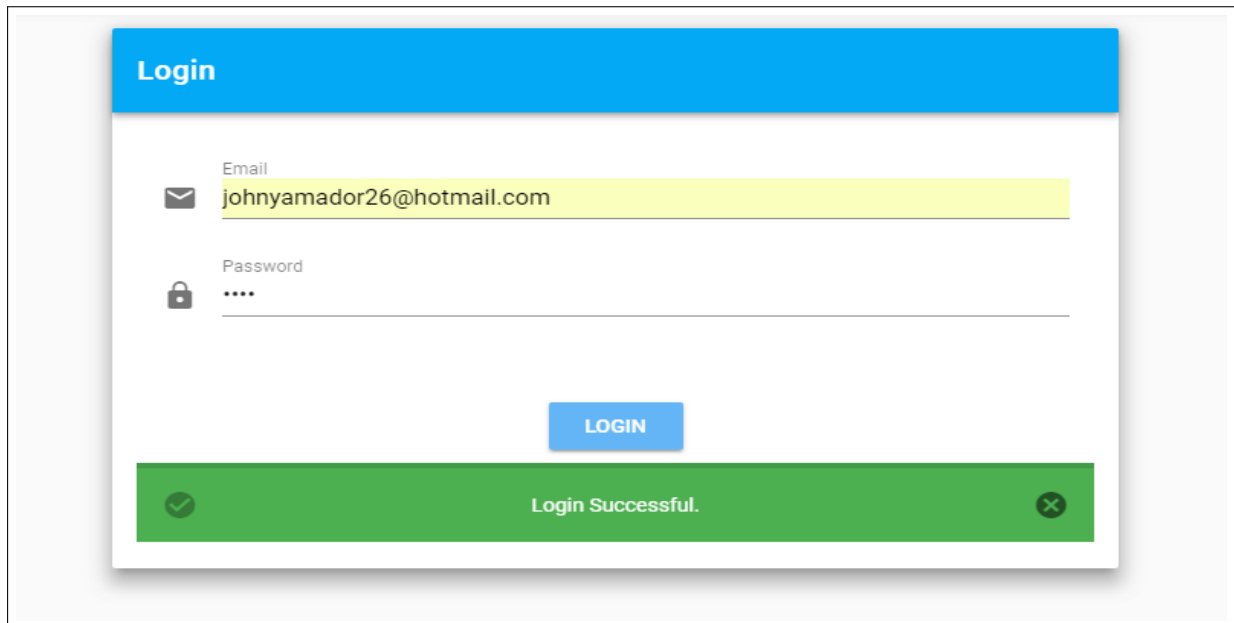
6.3.4 Login Page

The Login page style is similar to the Sign Up, again the form is very simple with only two required fields, "Email" and "Password". If the user successfully logs in, local tokens are stored with different information to be handled by the application, this page is illustrated in Fig. 6.7.

6.3.5 FAQ Page

The Frequently Asked Questions (FAQ) section is a part of the website where general concerns, questions or objections can be addressed. The go-to destination for finding answers to specific questions about the product or business operation.

It addresses questions from "What is DecSpace or MCDA?", "How do I sign up or login?" to "What are the main functionalities and methods of this application?". This interface is presented in Fig. 6.8.



The image shows a login form titled "Login" in a blue header. Below the header, there are two input fields: "Email" with an envelope icon and "Password" with a lock icon. The email field contains the text "johnyamador26@hotmail.com" and is highlighted in yellow. The password field contains four dots. A blue "LOGIN" button is centered below the fields. At the bottom, a green banner displays a checkmark icon, the text "Login Successful.", and a close icon (an 'x' in a circle).

Login

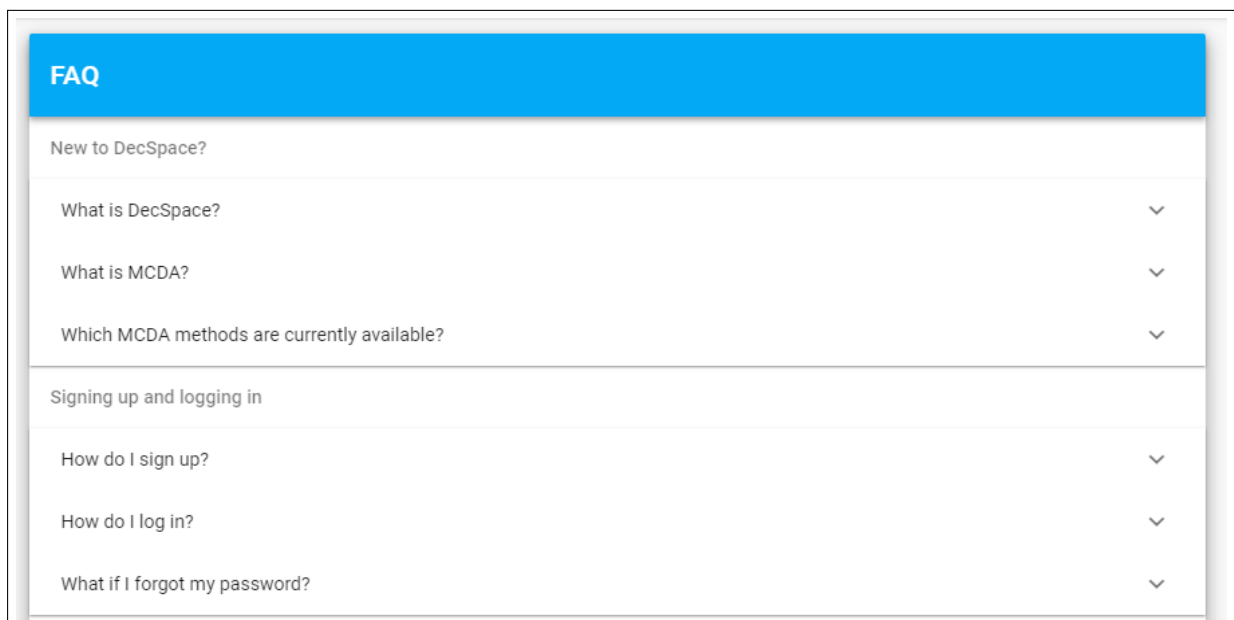
Email
johnyamador26@hotmail.com

Password
....

LOGIN

✓ Login Successful. ✕

Figure 6.7: DecSpace v2.0 Login Page.



The image shows an FAQ page with a blue header labeled "FAQ". The content is organized into two sections. The first section, "New to DecSpace?", includes three questions: "What is DecSpace?", "What is MCDA?", and "Which MCDA methods are currently available?", each with a downward arrow icon. The second section, "Signing up and logging in", includes three questions: "How do I sign up?", "How do I log in?", and "What if I forgot my password?", each with a downward arrow icon.

FAQ

New to DecSpace?

What is DecSpace? ▼

What is MCDA? ▼

Which MCDA methods are currently available? ▼

Signing up and logging in

How do I sign up? ▼

How do I log in? ▼

What if I forgot my password? ▼

Figure 6.8: DecSpace v2.0 FAQ Page.

6.3.6 Method Catalog Page

Presents a list of the available MCDA methods, each method is presented with a bookmark icon in "amber" color, the method title, a small description, links for important references and the creation date. All of this information can be treated as the user wishes, the preferences button allows for a customization of information views, as represented in Fig. 6.9.

Each method presents its unique information in different pages, the method name is highlighted, with sub-titles bars that represent the description, an example, and a step-by-step tutorial, an "arrow-left" icon is included to facilitate the backtrack for the catalog. For each section, the description presents a short presentation about the method. The example section presents tables, text or figures that resembles the implementation in the workspace as seen in Fig. 6.10 and Fig. 6.11, for the user to get an acquaintance with the core processes. And finally, the step-by-step tutorial section supports even more beginner users, presented in Fig. 6.12, by providing small and easy steps to conclude an experiment, by presenting images and support text that facilitates the learning of each user.

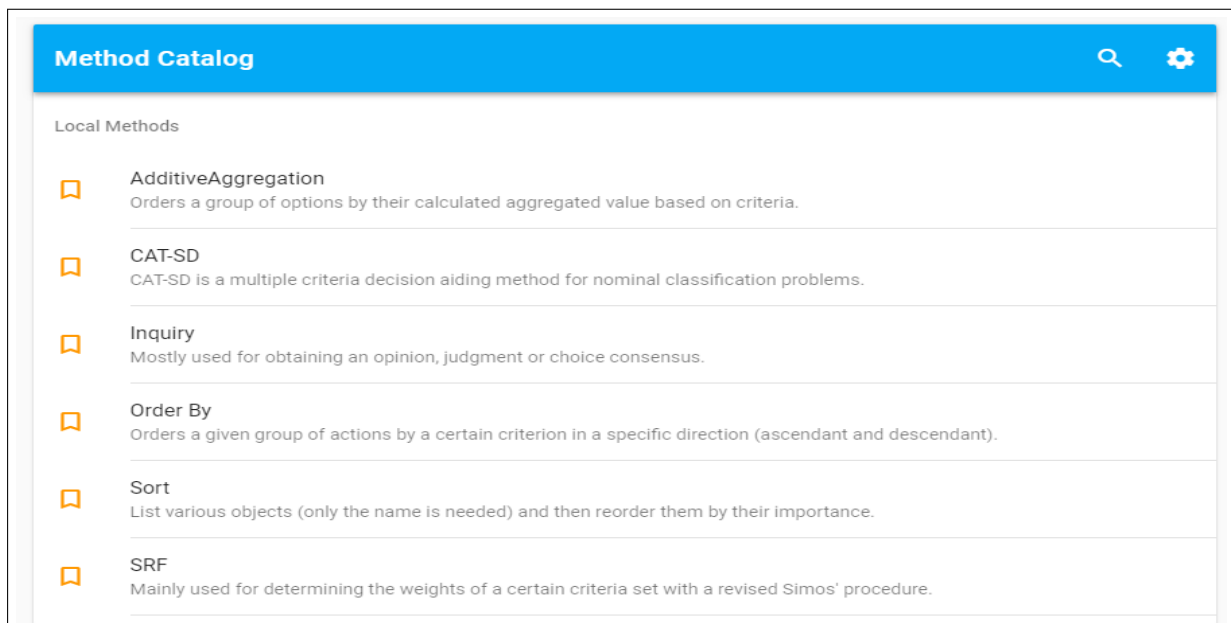


Figure 6.9: DecSpace v2.0 Method Catalog Page.

←

SRF

DESCRIPTION

EXAMPLE

STEP BY STEP

Criteria

Criterion Name
0-100
CO2
Consumption
Power
Price
Safety

©2018 — DecSpace

Figure 6.10: DecSpace v2.0 SRF Table Example.

Ranking

Most Important

Consumption

Safety

0-100

Power

Price

CO2

Least Important

©2018 — DecSpace

Figure 6.11: DecSpace v2.0 SRF Figure Example.

DESCRIPTION

EXAMPLE

STEP BY STEP

✓ Import "Sort By" Method

✓ Fill in the Data

✓ Run the Workflow

4 Check Results

DecSpace

METHOD CATALOG

FAQ

LOGOUT

Output File

Table

Order

1

2

3

4

Most Important

Least Important

OK

©2018 — DecSpace

BACK

FINISH

©2018 — DecSpace

Figure 6.12: DecSpace v2.0 Sort Tutorial.

6.3.7 My Projects Page

This is where the users can create and manage all their own projects, the focus is to bring a simple but yet complete view on project management, this page can be accessed in the user menu.

First, the user can control what he wants to see, through the settings icon it is possible, to adjust the preferences, this dialog window allows to display the project privacy and creation date.

The project list displays all the created projects and an example folder for group projects or projects with more than one file for organization purposes Fig. 6.13. Each project has a unique identifier, a project name, a creation date, and an icon that represents the privacy of each project, the privacy subtitle is also displayed.

For each project there are four functions, open project, can also be opened by clicking in the project line, edit privacy, changes the privacy and the icon, duplicate project, creates a copy of the respective project, and changes the project name depending on the copy number, and delete project, deletion of that project must be confirmed, this mechanism avoids accidental deletions of projects.

To create a project is as simple as touching the add button, the dialog window in Fig. 6.14 will open and a form with the project name and privacy (public or private) is requested to the user. All of this information is always synchronized with the database.

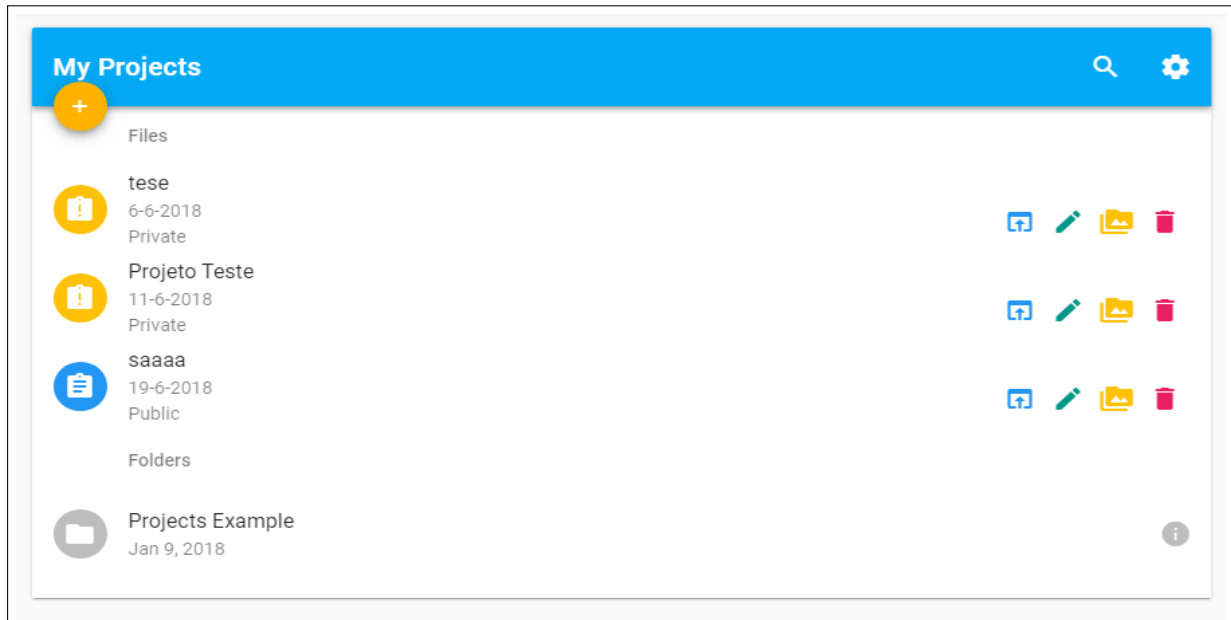


Figure 6.13: DecSpace v2.0 My Projects Page.

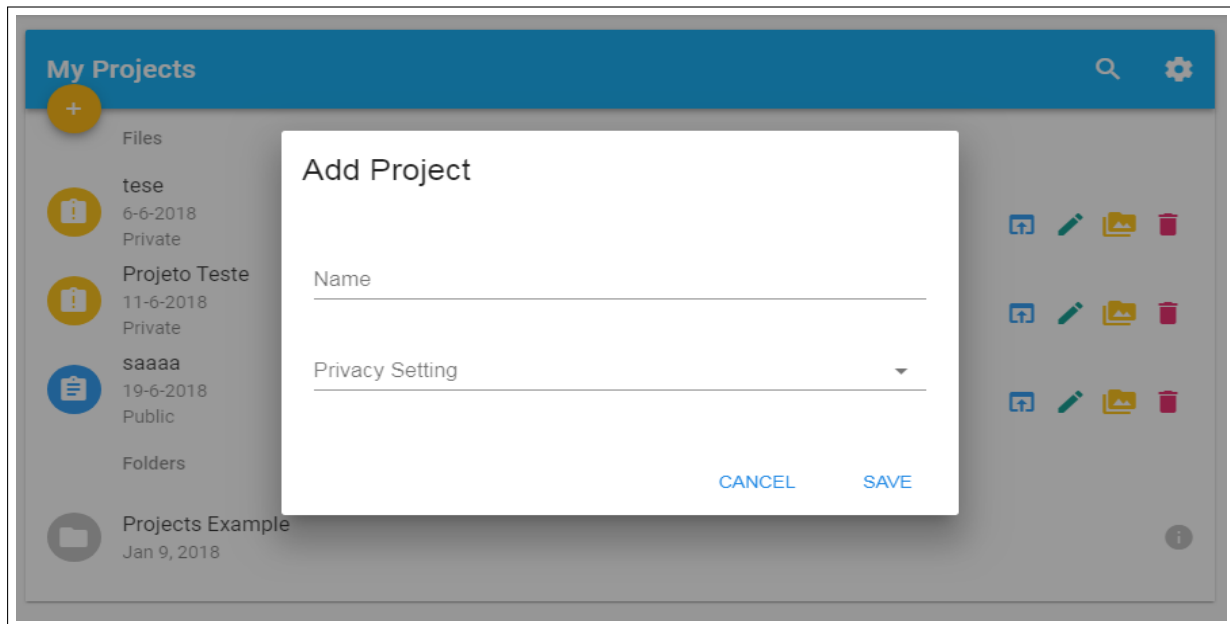


Figure 6.14: DecSpace v2.0 Add Project Dialog.

6.3.8 Public Projects Page

The Public Projects section, allows users to search for projects marked as public, this encourages any user to find already implemented projects, to help them support their idea. The projects can be cloned for their own My Projects area, and freely work with them, or they can be opened and examined, without saving alterations, exemplified in Fig. 6.15.

The user can also control is own preferences, through the settings icon it is possible, to adjust the preferences, this dialog window allows to display the project privacy, creation date, and the owner of the project, Fig. 6.16.

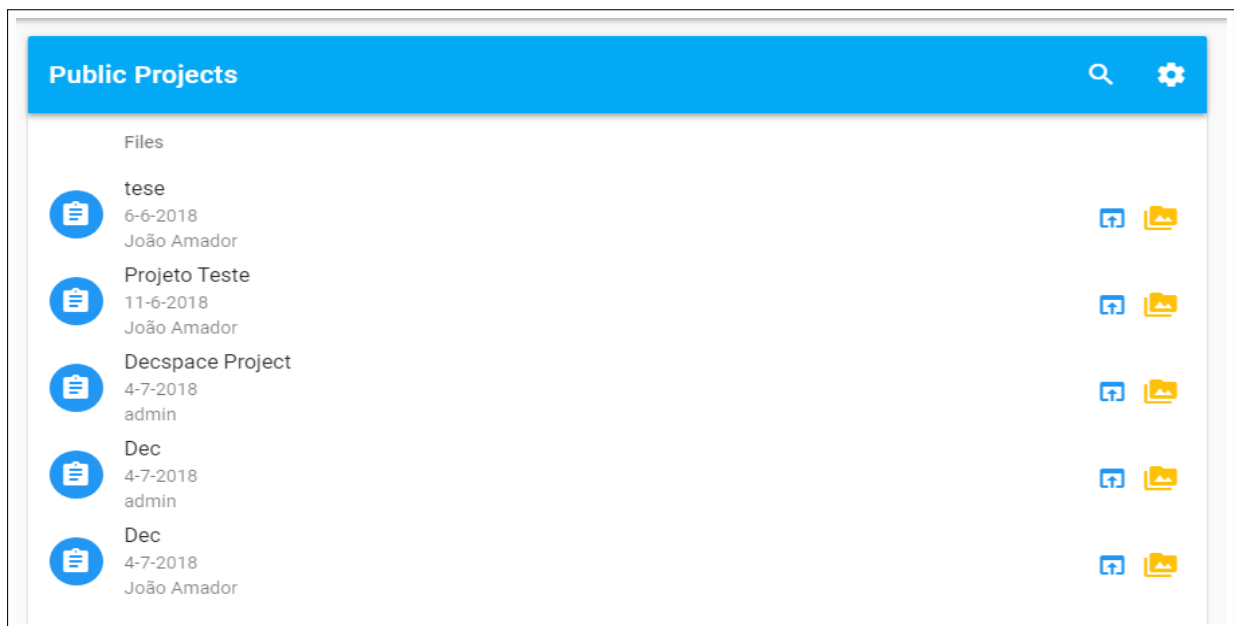


Figure 6.15: DecSpace v2.0 Public Projects Page.

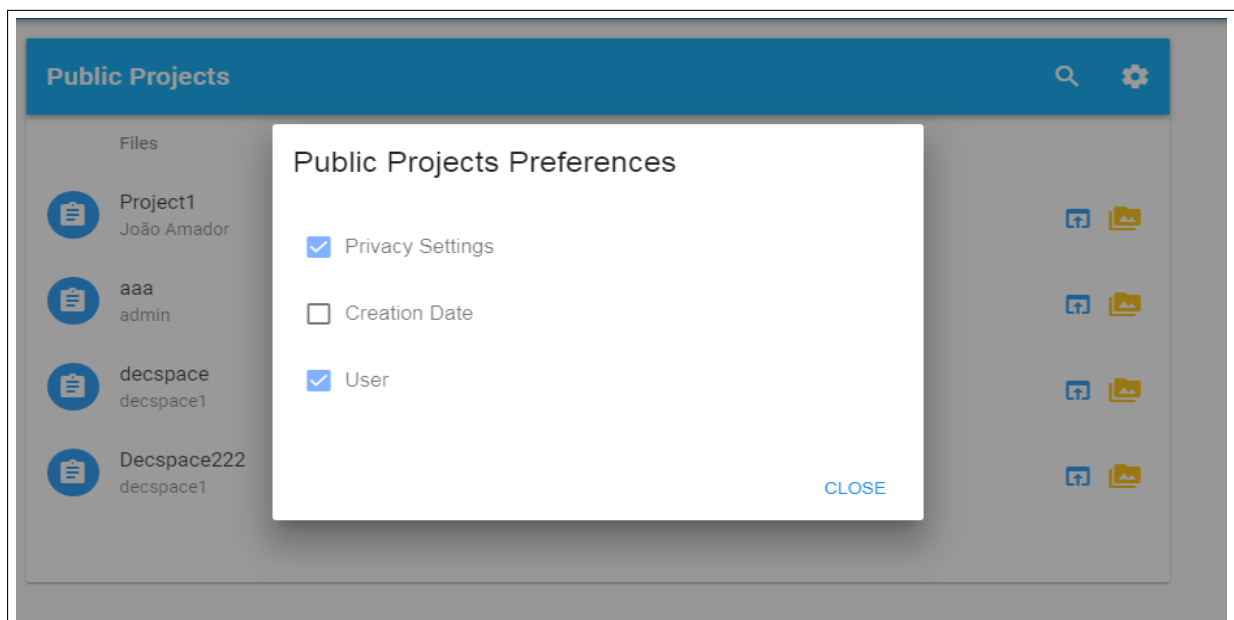
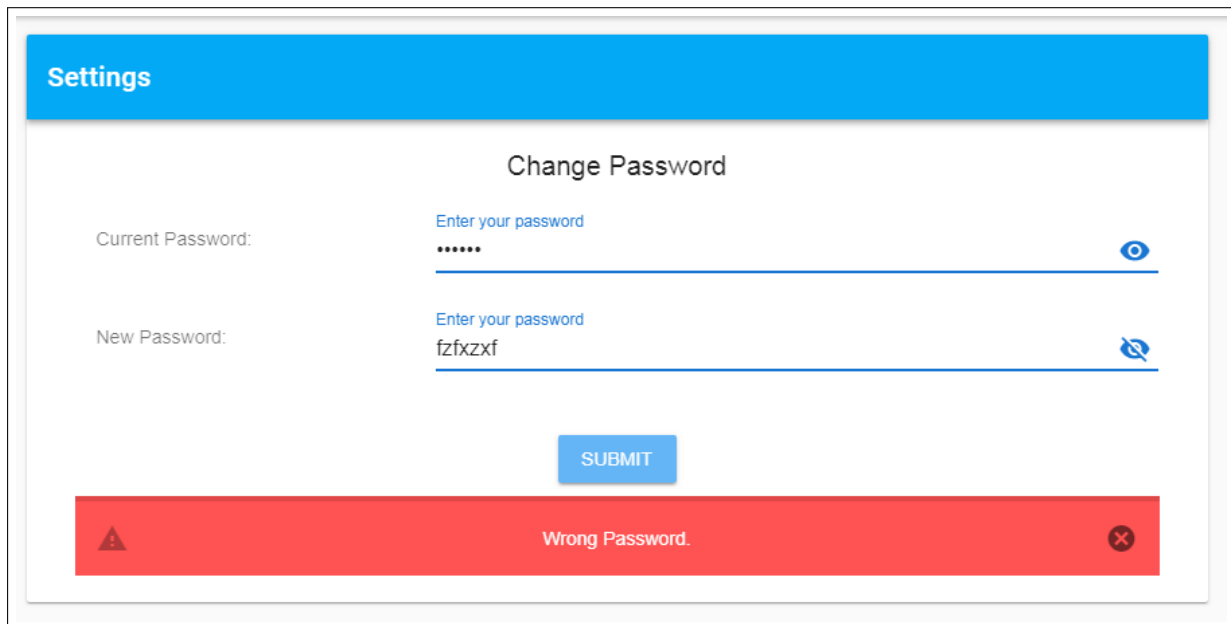


Figure 6.16: DecSpace v2.0 Public Projects Preferences.

6.3.9 Settings Page

The Settings page provides password change, through a secure process, without the need to confirm the new password, the user can control the conceal of each input, and interpret the result through various information messages. This section is exemplified in Fig. 6.17.



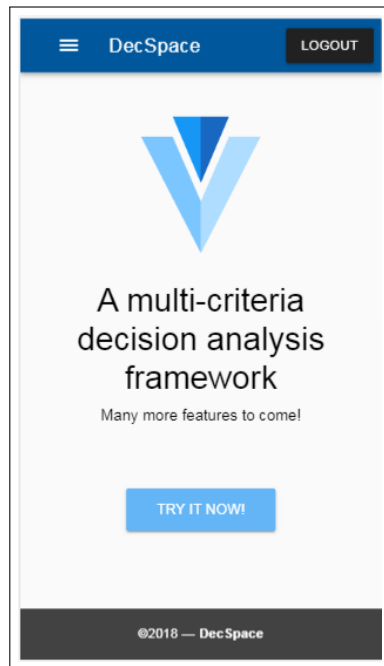
The screenshot displays the 'Settings' page with a blue header. The main section is titled 'Change Password'. It contains two input fields: 'Current Password' and 'New Password'. The 'Current Password' field is masked with dots and has a toggle icon to its right. The 'New Password' field contains the text 'fzfxzxf' and also has a toggle icon. Below the input fields is a blue 'SUBMIT' button. At the bottom, there is a red error message bar with a warning icon on the left, the text 'Wrong Password.' in the center, and a close icon on the right.

Figure 6.17: DecSpace v2.0 Settings Page.

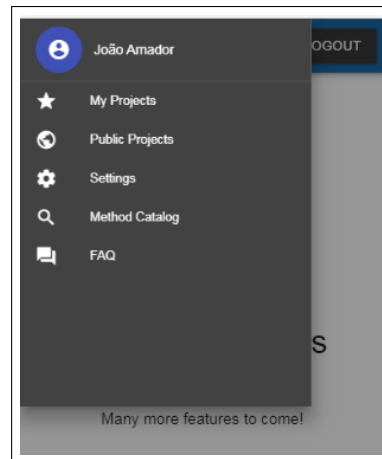
6.3.10 Device Mode

Device Mode was tested in Chrome DevTools to enable streamlined device display emulation of various devices that included iPhone, Samsung Galaxy, and iPad. The goal is to provide the best experience possible in each type of device, to make a pleasant activity of using DecSpace. The improvements were made from the scratch, providing a different experience for each page, and improving the usage of smaller screens and touch features.

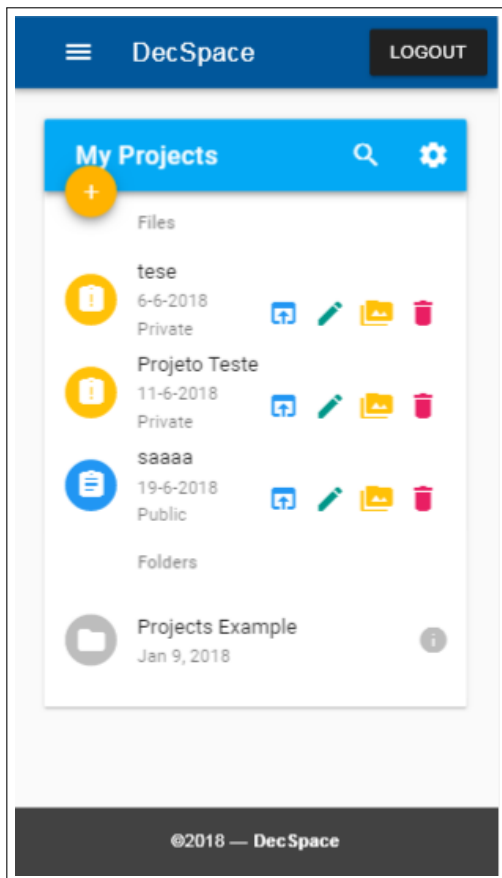
The homepage pictured in Fig. 6.18(a), represents a web application with all the information about DecSpace, the page fits perfectly on the screen and adapts to each device. The user menu depicted in Fig. 6.18(b), was changed to feature all the different links, leaving the toolbar for the logs. All the features in DecSpace exhibit a new way to be used, with an improved interface for portable devices Fig. 6.18(c) and Fig. 6.18(d).



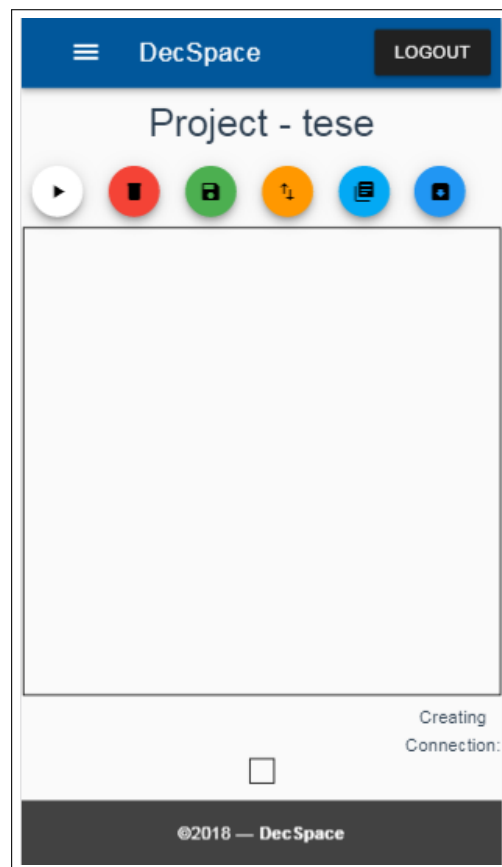
(a) Device Mode v2.0 Home Page



(b) Device Mode v2.0 User Menu



(c) Device Mode v2.0 My Projects Page



(d) Device Mode v2.0 Workspace Page

Figure 6.18: DecSpace v2.0 Device Mode

6.3.11 Workspace Page

The most particular area, this is where all the technical work happens. After the creation and opening of a project, the user is redirected to the Workspace page, where it is possible to build and execute workflows. The workspace is composed of six buttons, each with its own color and activity, a whiteboard and a create connection indicator complete the main working area, as indicated in Fig. 6.19.

A variety of methods are available for the user to explore, but only one is implemented presently. Of the methods button, the user can select the favored method, and when opened, the user can manually add data to them, or consult the references from each specific method, Fig. 6.20. The methods are represented in small boxes with a different color if they are output files, methods or input files. They are also draggable, around the workspace Fig. 6.21.

DecSpace presents other features like execute workflow, runs the workflow identifying errors and producing an output file, delete all modules, eliminates all module boxes from the workspace, save workflow, creates multiple saves that can be accessed from the project menu, import workflow, import a comma-separated values (CSV), XMCD or javascript object notation (JSON) file, a zip file can also be added, containing a workflow that was already used, methods, presents a list of local implemented methods, references for those methods and remote methods, through simple object access protocol (SOAP) requests between the diviz server, project menu, accesses the project saves for user administration, Fig. 6.22.

Finally, the output files, have some new improvements, not only the solution of the problem is represented, the user can alternate between the normal view and graphs view, there are four types of graphs, line chart, bar chart, bubble chart, Fig. 6.23, and doughnut chart, all of them fully interactive and ready to adapt to any solution.

6.3.11.A Sort Method

The Sort method was developed with the purpose of testing the architecture of the workspace, it is a simple method that handles the drag-and-drop functionality made available by the v-dragged library.

The requirements are names of objects submitted by the user, and, then ordered using the drag-and-drop functionality. To order the objects, the user can drag any object along the object line and drop in the favored position, an example is demonstrated in Fig. 6.24.

The output of this method is simply the equivalent set of ordered objects that was inserted by the user, or any graphical viewpoint the user wants to access.

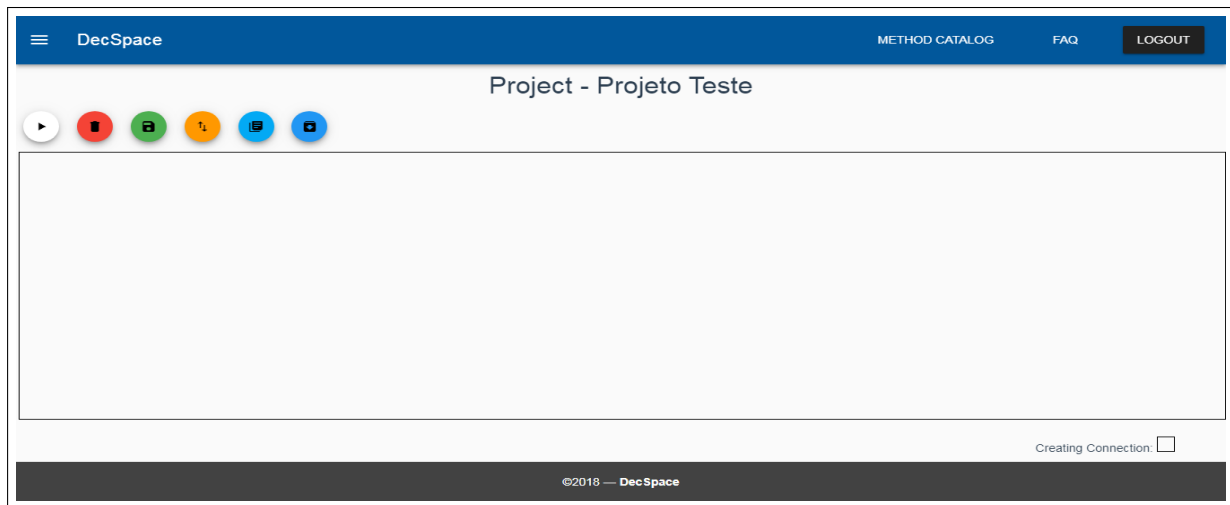


Figure 6.19: DecSpace v2.0 Workspace Page

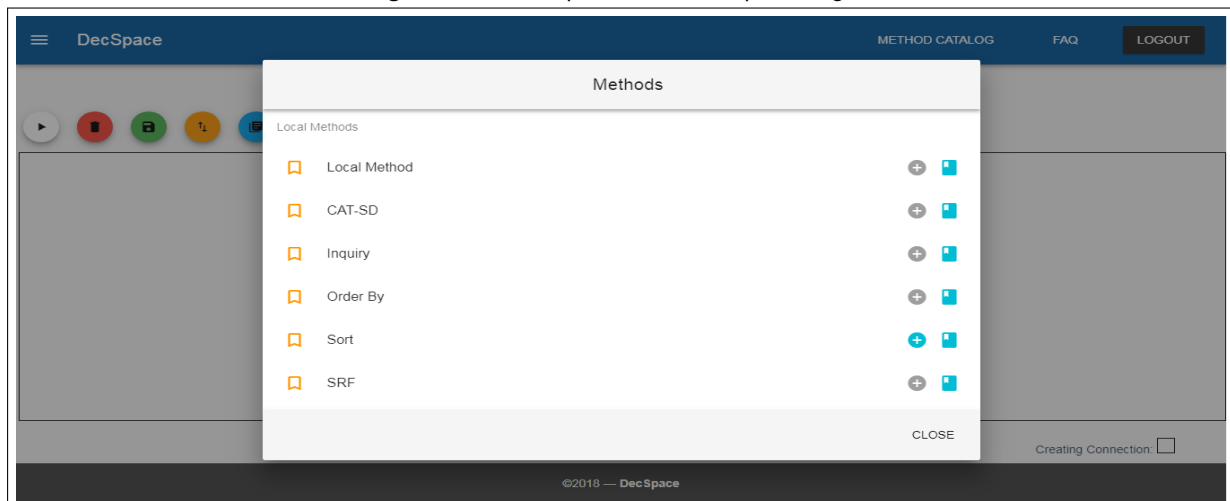


Figure 6.20: DecSpace v2.0 Workspace Methods Page

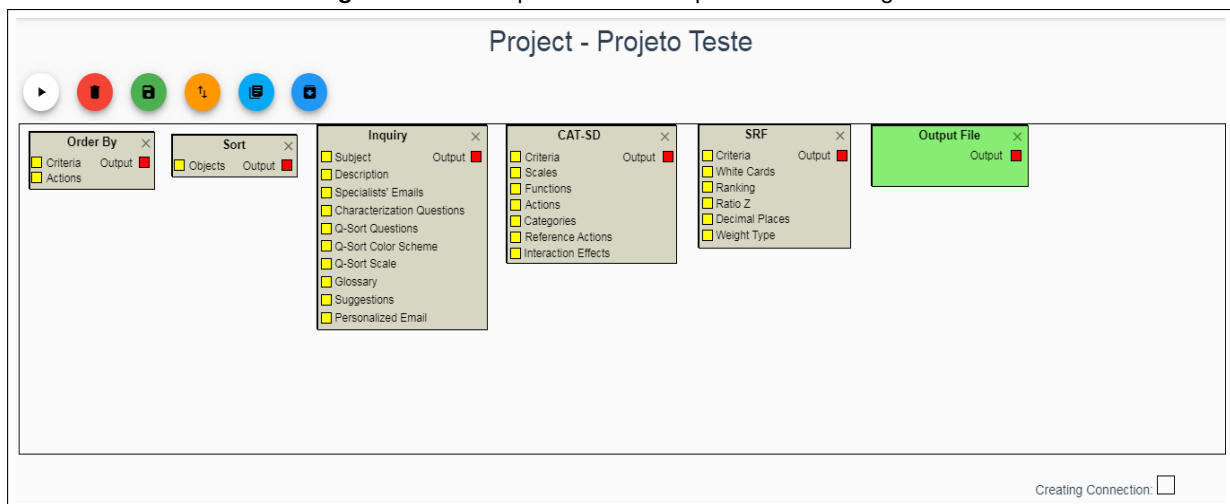


Figure 6.21: DecSpace v2.0 Workspace Boxes

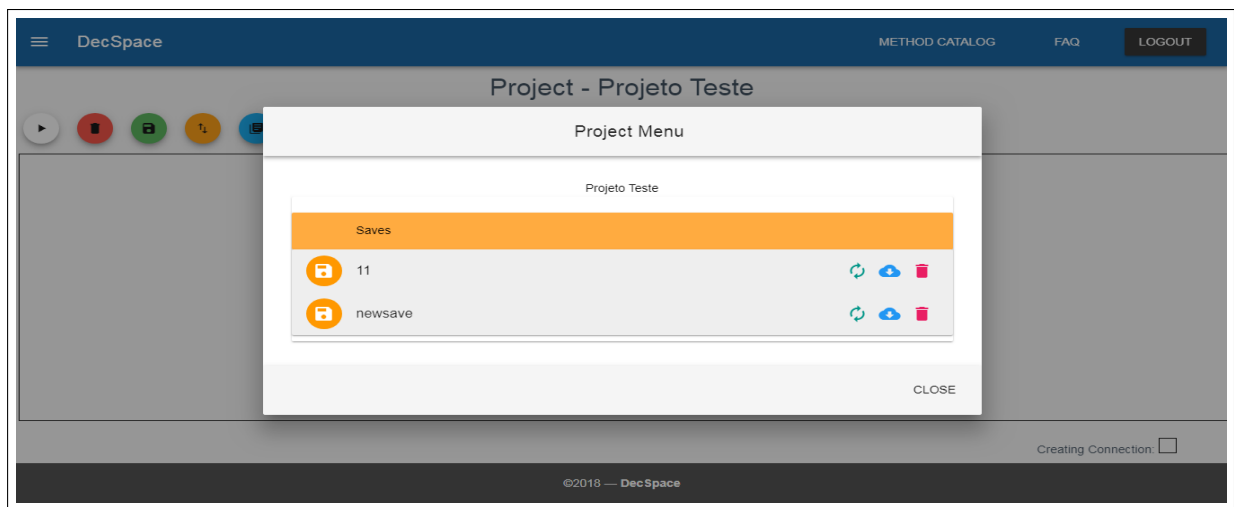


Figure 6.22: DecSpace v2.0 Workspace Project Menu



Figure 6.23: DecSpace v2.0 Graphs Page

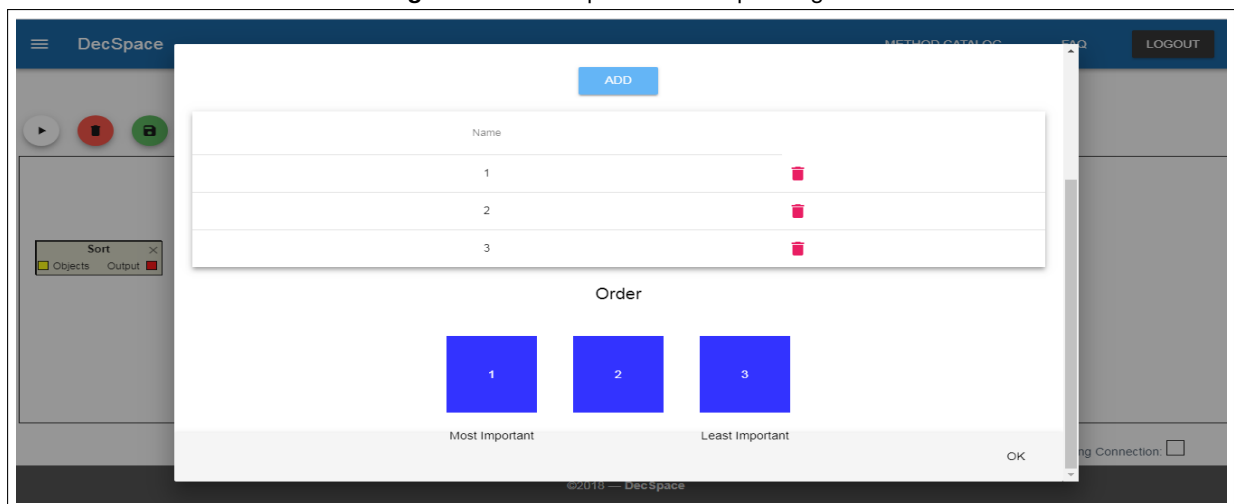


Figure 6.24: DecSpace v2.0 Sort Method

6.3.12 Administrator Page

The Administrator page is only accessible to the DecSpace admin. It presents all the information about the projects and users of the framework, as it allows to delete any information it must be guarded in a secured location. For each user is displayed the username and email, and for each project is revealed the project name, creation date, and owner. The administrator has permission to remove the data of any user or project. The Administrator page is exemplified in Fig. 6.25 and Fig. 6.26.

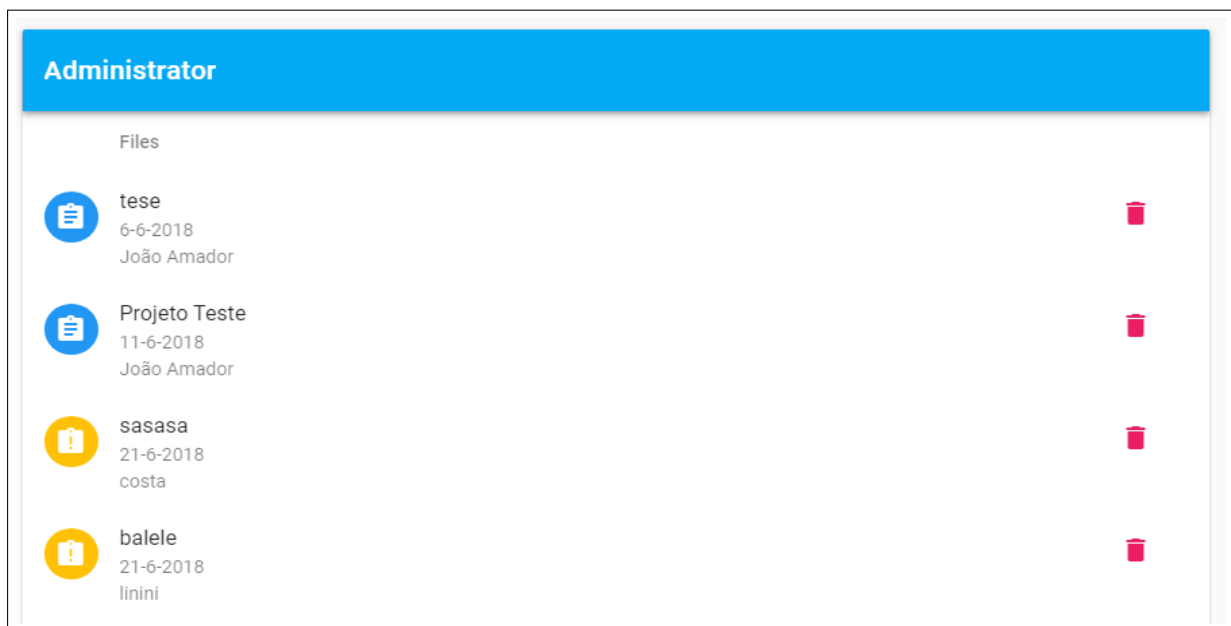


Figure 6.25: DecSpace v2.0 Administrator Files Page.

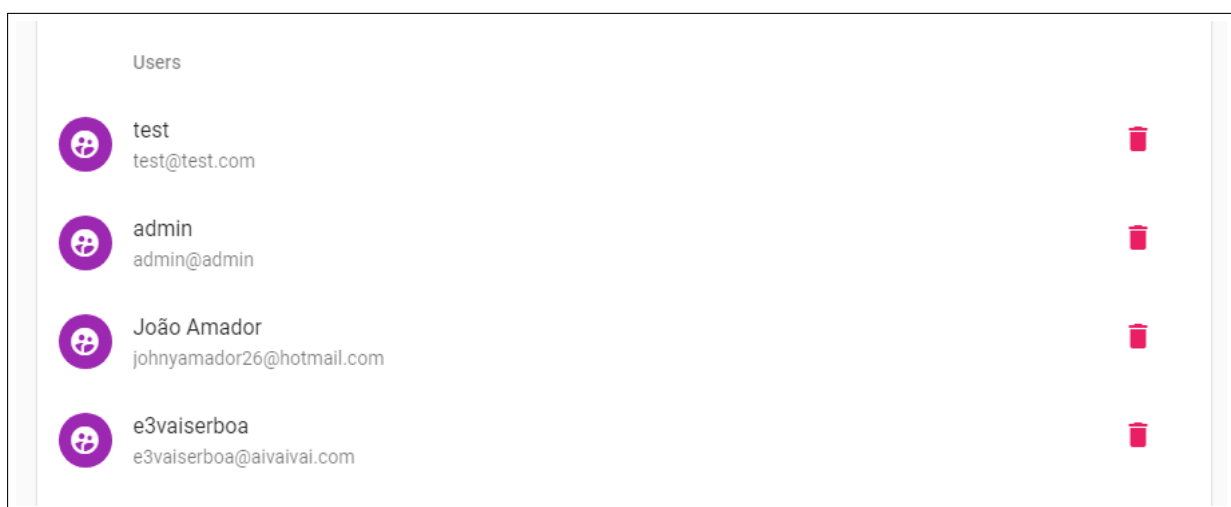


Figure 6.26: DecSpace v2.0 Administrator Users Page.

6.4 Back-End

A "back-end" application serves indirectly in support of the front-end services, usually by being closer to the required resource or having the capability to communicate with the required resource. In this case, it's necessary to interact with both the front-end and the database.

For the back-end side of DecSpace, the most relevant packages installed are: 'express'¹⁰ is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications, 'body-parser'¹¹ parses incoming request bodies in a middleware before handlers, 'bcrypt'¹² is a password hashing function, and 'mongoose'¹³ an object modeling tool designed to work in an asynchronous environment.

For the main application back-end, two models were designed, User model requires three attributes: email and user-name. are both from the type string, unique for each user, mandatory and can't possess white spaces, the final attribute is the password that has to be a string and mandatory. The Project model requires nine attributes: id, project name, owner, creation date, privacy, icon and icon class are all type string and mandatory, finally a project could have a copy status or an array of save files.

The main functionalities of the core back-end start with establishing a connection between the database using mongoose, a username, and password, using a secret session for tracking logins.

Then the application can start processing requests, for the user model it is possible to register a user through an HTTP request "Post" validating if all the fields have been filled in and if the email doesn't already exist, if something wrong occurs a non success token and a message detailing the error are sent to the user, if the process succeeds a new user account is created along with the hashed password, this method was implemented to improve security in the application, bcrypt is the algorithm chosen to securely hash and salt passwords, building a password security platform that can evolve alongside hardware technology to guard against the threats that the future may bring.

The login request, works through an HTTP request "Post", again validating if all the fields are answered, if the email exists in the database, and if the password is equal to the one stored. If the login is not accepted, a message is sent to the user, with the success token set to false and an error message, if the login is successful the token is set to true, a message with the login details and the user-name are sent, an example of the code can be seen in Fig. 6.27.

There is also an HTTP request "Get" to retrieve all users and a change password request that modifies the user password if all the validations are correct.

The project model follows the same structure it is possible to create, edit and delete a project through an HTTP request "Post", retrieve all projects from the respective user, and to add save files for each

¹⁰<https://www.npmjs.com/package/express>

¹¹<https://www.npmjs.com/package/body-parser>

¹²<https://www.npmjs.com/package/bcrypt>

¹³<https://www.npmjs.com/package/mongoose>

project if all the validations are correct.

```
app.post('/login', function (req, res) {
  if (!req.body.password || !req.body.email) {
    console.log('miss')
    return res.json({success: false, msg: 'Please input username and password.'})
  }
  User.findOne({email: req.body.email}, function (err, user) {
    if (err) {
      console.log('email miss')
      return res.json({success: false, msg: 'Email does not exist.'})
    }
    if (user === null) {
      console.log('email miss')
      return res.json({success: false, msg: 'Email does not exist.'})
    }
    bcrypt.compare(req.body.password, user.password, function (err, isMatch) {
      if (err) {
        console.log('pass mal')
        return res.json({success: false, msg: 'Wrong Password.'})
      }
      if (isMatch === true) {
        console.log('tudo certo')
        res.send({success: true, msg: 'Login Successful.', username: user.username})
      } else {
        console.log('pass mal')
        return res.json({success: false, msg: 'Wrong Password.'})
      }
    })
  })
})
```

Figure 6.27: A snippet of the code of the "Login" function.

6.5 Database

The solution devised for the database was setting the development version in an open cloud service without worrying about performance and looking for a simple and effective way to control and manage data, for the deployment version, the database was changed to a local database to present better performance and fewer issues with services.

MLab¹⁴ was chosen for creating and managing the development database of DecSpace. This service provides a fully managed cloud database service that hosts MongoDB databases, this was preferred to a usual local database because it provides a secure data storage and retrieval, as well as a very intuitive and valuable interface.

The database stores users Fig. 6.28(a) and respective projects Fig. 6.28(b), for the users, an email, user-name, and an encrypted password are all collected in aggregation with a unique identifier. The Projects are stored with their respective unique identifier, name, owner, creation date, icon and class, privacy and an array of save files.

```
1 {
2   "_id": {
3     "$oid": "5afc6bb145165e0688a708ac"
4   },
5   "username": "João Amador",
6   "email": "johnyamador26@hotmail.com",
7   "password": "$2b$10$Uj0BhlxkJ1Jv4ZTAvfXy70hg1
8   "__v": 0
9 }
```

```
1 {
2   "_id": {
3     "$oid": "5afedd38476dd81df4e131e4"
4   },
5   "id": "1",
6   "name": "Project1",
7   "username": "João Amador",
8   "creation": "18-5-2018",
9   "privacy": "Private",
10  "icon": "assignment_late",
11  "iconClass": "amber white--text",
12  "__v": 7,
13  "saveFile": [
14    "dasdsddsadsads",
15    "1",
16    "save2",
17    "save3",
18    "save3"
19  ]
20 }
```

(a) Database User Structure

(b) Database Project Structure

Figure 6.28: Database Structure

¹⁴<https://mlab.com/home>

7

Evaluation

After the definition and implementation of the requirements, the new framework created needs to be evaluated, three different perspectives endure usability, functionality and user experience [29]. Our evaluation takes into consideration these perspectives, for this system to be reasonably judged and confronted with others.

The objective was to create a new framework that presents information in the most simple way possible so that users can easily understand, learn and work with the system. To guide them in this evaluation a system evaluation guide (Appendix A) was created, detailing all the steps for a fundamental test, this plan covers the most relevant aspects of the evaluation, including the necessary equipment, the various phases of the evaluation and the tasks that will be performed by the participant, the system evaluation survey (Appendix B) for users to give important feedback and opinions on what is wrong or can be improved and finally the average results of the system evaluation survey (Appendix C) are displayed.

7.1 System Evaluation Plan

A system evaluation plan has the objective to test the framework on all crucial features, our desire with this evaluation is to understand everything that feels wrong or confusing for all users, also correcting bugs and errors that emerge. The system evaluation guide was designed with the objective of explaining the various steps of the evaluation, so that valid results could be achieved and that the test sessions would occur as active as possible. After each step of the guide, the user is encouraged to discuss the best and worst mechanics, all the steps are recorded and the metrics collected. After the completion of the guide, it is mandatory to answer the survey, composed of thirteen questions about usability, aesthetics, recommendations.

All of the tests were executed in the laboratories of Instituto Superior Técnico - Campus Taguspark, always in the presence of the administrator to guide and record all of the progress. These tests were performed during the month of July 2017, with the maximum duration of 60 minutes, it is important to state that none of the 20 participants had little to no experience with MCDA. The initial idea was to have two rounds of tests, but do to a weak response of the new participants during August 2018, the second phase was abandoned.

The following metrics were recorded: time to complete each task, successful task completion, user errors per task (critical or non-critical), and user experience (recommendations).

It is necessary to stipulate some objectives for the results to be properly analyzed, the results should not exceed the following values:

- Time to complete task 1: 2.00.00 minutes
- Time to complete task 2: 3.00.00 minutes

- Time to complete task 3: 2.30.00 minutes
- Time to complete task 4: 2.15.00 minutes
- Time to complete task 5: 5.00.00 minutes
- Time to complete task 6: 1.30.00 minutes
- Number of non-critical errors per task: 0.5
- Number of critical errors per task: 0.15

7.2 Result Analysis

The analysis of the results are divided into two segments, the first studies the recommendations and alterations requested by users at the end of each step, and the second segment analyses the results of the survey.

These are the main adjustments or modifications addressed by the users that tested the framework:

- Change the name of "User Preferences" to "Method Catalog Preferences", "My Projects Preferences" and "Public Projects Preferences";
- More information in error messages;
- Change the execute icon;
- Change "Edit Privacy" to "Change Privacy";
- In tutorial change the placement and color of the "Finish" and "Back" buttons;
- Apply some outline in "OK" buttons;
- Change methods avatar;
- Open project by clicking on the line of the project;
- Graphs interface redesigned;
- Fixes in icons placements in all the framework;

All of these issues were addressed and implemented resulting in the first version of the new DecSpace.

Regarding the analysis of the survey, MCDA is not a simple concept it requires learning and understanding, our main motivation is to keep it as simple as possible. Our survey reflects this, the average score of the difficulty of all the features, scored above 3.5, which indicates that some features present

more challenges than others, but the framework keeps its simplicity, for the aesthetics the average score of 4.0 on all features, demonstrates the visual power and attractiveness of DecSpace, which can be one of the most important features. The other questions reflect the user preferences in usefulness, innovation, and efficiency all of them scoring more than 4.0. The last feature is the device mode experience, it scored 4.4, the highest of all the features, expressing the satisfaction and usability of using this feature in your smartphone or tablet.

7.3 Conclusions

The evaluation conducted aims to stabilize and secure the framework the best way possible, the tests were done in a closed environment with the support of a system evaluation guide and survey to accomplish the best and most authentic results possible. The evaluation is prepared for 20 participants, the main accomplishments for this preparation are the correction of errors and bugs, implementation of new features that help users, test the user interface and study tendencies and suggestions.

The results obtained suggest that the framework is well structured and attracts users, for its new interface, design, and usability, offering a service that users would like to use in the day to day life.

8

Conclusions

This chapter resumes and concludes all the work done in the course of a year, it takes into account all the research performed, the implementation and evaluation produced and the written paper to produce this dissertation. To conclude the future work is detailed.

8.1 Conclusions

The purpose of MCDA is to provide support to decision makers during decision processes with methods and techniques, as well as frameworks for computational support. Handling these frameworks can be challenging for users who are not experts in MCDA methods. For that reason, requirements related to usability and user-friendly issues need to be taken into account in the development of these frameworks.

In this dissertation, we proposed a new approach to the MCDA phenomenon with a framework that employs forefront technology, well-studied usability, and user interaction techniques. The fundamental objective is to support a decision maker to structure and to better understand the decision problem at hand.

This framework resulted from the investigation of the current MCDA frameworks and from the analysis of their greatest and worst features, the first version also had an enormous influence on the ideals and requirements of DecSpace.

DecSpace is a web-based framework that provides MCDA methods and allows to create projects with multiple workflows using one or more of such methods. Indeed, an increasing number of MCDA methods are going to be available, implemented locally or remotely exposed by diviz server and as web services. We show that DecSpace is intended to be a user-friendly solution for supporting the decision processes in different scenarios, for both non-expert users and MCDA expert users. We presented the main features of DecSpace and highlighted that is focused on the usability and user interface.

8.2 Future Work

This dissertation was focused on creating a new framework with new technologies, converging to usability and user interface that differs from what already exists. The architecture was restructured and the structure of the project redesigned, the core front-end, back-end, and database built from the ground up.

Still, DecSpace is a continuous project that is always evolving and needs constant maintenance for new bugs or errors that may arise, for that it was decided to divide the future implementations into two parts.

The first part, it is to improve and align some use cases, the administrator page needs to be in a private URL, in a secure location that cannot be accessed by other users.

The workspace still needs some improvements and features, in the area of save files, the importation and exportation for the database, the connection between methods, importing and exporting data files, deleting modules.

The tutorials on the method catalog need to be completed as soon as the methods are implemented and the references for each method.

The device mode and the version deployed show minor bugs.

The second part is about suggested features, the service should expand an interface that is able to communicate with external servers. This way, some MCDA methods can be executed outside of the application, which implies that there is not the need of implementing those methods in the framework, diviz server can be re-implemented for this new version.

The original methods or new ones can be implemented, expanding the service with many different MCDA methods.

The current version of DecSpace supports four data visualization graphs, different ones can be implemented to show new ways of information, in parallel with the implementation of methods, the graphics should adapt to those methods.

The concept of group projects and projects folders, allowing for shared projects is a good idea for inter-usability.

Bibliography

- [1] V. Belton and T. J. Stewart, *Multiple criteria decision analysis : An integrated approach*, 2002.
- [2] A. Barbosa, "DecSpace : A Multi-Criteria Decision Analysis Framework Master Project in Information and Software Engineering," 2017.
- [3] K. Marsh, M. Goetghebeur, P. Thokala, and R. Baltussen, "Multi-Criteria Decision Analysis to Support Healthcare Decisions," *Cham: Springer International Publishing*, 2017.
- [4] I. Huang, J. Keisler, and I. Linkov, "Multi-criteria decision analysis in environmental sciences: Ten years of applications and trends," *Science of the total environment*, 409(19), 3578- 3594., 2011.
- [5] M. Doumpos and C. Zopounidis, "Multicriteria analysis in finance," *Cham, Switzerland: Springer*, 2014.
- [6] A. Mardani, E. Zavadskas, Z. Khalifah, N. Zakuan, A. Jusoh, K. Md Nor, and M. Khoshnoudi, "A review of multi-criteria decision-making applications to solve energy management problems: Two decades from 1995 to 2015," *Renewable and Sustainable Energy Reviews*, 71, 216–256, 2017.
- [7] P. Guarnieri, "Decision Models in Engineering and Management," *Cham, Switzerland: Springer*, 2015.
- [8] R. B. and D. Bouyssou, "Aide Multicritère à la Décision: Méthodes et Cas," *Paris, France: Economica*, 1993.
- [9] S. Greco, M. Ehrgott, and J. Figueira, "Multiple Criteria Decision Analysis: State of the Art Surveys (2nd ed.)," *New York, NY: Springer Science + Business Media, Inc.*, 2016.
- [10] M. Ehrgott, J. Figueira, and S. Greco, "Trends in multiple criteria decision analysis," *New York, NY: Springer Science + Business Media, Inc*, 2010.
- [11] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukias, and P. Vincke, "Evaluation and Decision Models with Multiple Criteria. Stepping Stones for the Analyst," *New York: Springer*, vol. 86, p. 446, 2006.

- [12] B. Roy, "Multicriteria Methodology for Decision Aiding," *Dordrecht, The Netherlands: Kluwer Academic Publishers*, 1996.
- [13] D. Bouyssou, T. Marchant, M. Pirlot, P. Perny, A. Tsoukiàs, and P. Vincke, "Evaluation and decision models: A critical perspective," *Dordrecht, The Netherlands: Kluwer Academic Publishers*, 2000.
- [14] S. Corrente, J. Figueira, S. Greco, and R. Słowiński, "A robust ranking method extending ELECTRE III to hierarchy of interacting criteria, imprecise weights and stochastic analysis," *Omega, The International Journal of Management Sciences*, 73, 1–17, 2017.
- [15] A. Costa, J. Figueira, C. Vieira, and I. Vieira, "An application of the ELECTRE TRI- C method to characterize government performance in OECD countries," *International Transactions in Operational Research*, 2017.
- [16] D. Liu and T. Stewart, "Integrated object-oriented framework for MCDM and DSS modelling," *Decision Support System*, 38, 421–434, 2004.
- [17] A. Ishizaka and P. Nemery, *Multi-Criteria Decision Analysis - Methods and Software*, 2013.
- [18] Q. Hayez, Y. D. Smet, and J. Bonney, "D-Sight : A New Decision Support System to Address Multi-Criteria Problems CoDE-SMG – Technical Report Series Technical Report No .," no. August, 2011.
- [19] M. B. Barfod, "Graphical and technical options in Expert Choice for group decision making," *DTU Lyngby: Technical University of Denmark, Transport*, vol. 1, pp. 1–35, 2014.
- [20] "Macbeth," *Working Paper LSEOR*, pp. 1–40, 2003.
- [21] B. Mareschal, "Visual PROMETHEE manual," 2013.
- [22] P. Meyer and S. Bigaret, "Diviz: A software for modeling, processing and sharing algorithmic workflows in MCDA," *Intelligent Decision Technologies*, vol. 6, no. 4, pp. 283–296, 2012.
- [23] J. Mustajoki and M. Marttunen, "Comparison of Multi-Criteria Decision Analytical Software - Searching for ideas for developing a new EIA-specific multi-criteria software," *IMPERIA Project Report*, pp. 1–85, 2013.
- [24] L. C. Dias and J. N. Climaco, "Additive aggregation with variable interdependent parameters: The VIP analysis software," *Journal of the Operational Research Society*, vol. 51, no. 9, pp. 1070–1082, 2000.
- [25] A. S. Costa, J. R. Figueira, and J. Borbinha, "A Multiple criteria nominal classification method based on the concepts of similarity and dissimilarity," *European Journal of Operational Research*, 2018.

- [26] A. Habibi, A. Sarafrazi, and S. Izadyar, "Delphi Technique Theoretical Framework in Qualitative Research," *The International Journal Of Engineering And Science*, vol. 3, no. 4, 2014.
- [27] J. Figueira and B. Roy, "Determining the weights of criteria in the ELECTRE type methods with a revised Simos' procedure," *European Journal of Operational Research*, vol. 139, no. 2, pp. 317–326, 2002.
- [28] E. Siskos and N. Tsotsolas, "Elicitation of criteria importance weights through the Simos method: A robustness concern." *European Journal of Operational Research*, 246(2), 543–553, 2015.
- [29] M. Fonseca, P. Campos, and D. Gonçalves, *Introdução ao Design de Interfaces*, 2012.



System Evaluation Guide

System Evaluation Guide

This document has the objective of giving the necessary tips and tasks to the participants of the system evaluation, so that they are able to successfully complete the tests.

Setup

This evaluation takes into account some elements that need to be present:

- A personal computer for the user to test the application.
- This system evaluation guide for the each task.
- The survey to be answered after the completion of all the tasks.

Tasks

Task 1

Scenario:

Create an user account and login into DecSpace.

Guidelines:

- Explore the front page and get some information about DecSpace.
- Sign up
Create an account by providing an "Username". In the field "Email Address" provide a valid email address (i.e. that has not been previously used to sign up into DecSpace). And the last entry is the "Password".
- Log in with the "Email" and "Password" provided.

Task 2

Scenario:

Explore the "FAQ" and "Method Catalog", to better understand the "Sort" method.

Guidelines:

- Explore the "FAQ" page, to learn about the main functionalities.
- Explore the "Method Catalog"
Adjust the method catalog preferences by displaying the References and Date of each method.
Select the "Sort" method, read the description and example, and finally complete the tutorial for the method.

Task 3

Scenario:

Explore the User area, by changing the password and duplicating a public project.

Guidelines:

- Open the user menu.

- Change your password in Settings.
- In the “Public Projects” area, open then return and duplicate the “Projeto Teste” project created by the user “João Amador”.
- In “MyProjects” area, check if the project has been successfully duplicated.

Task 4

Scenario:

Create, duplicate, edit and delete a project in “MyProjects”.

Guidelines:

- Adjust the user preferences, by displaying the privacy and date.
- Create a new project, with the project name and set the “Privacy” to private.
- Duplicate the project.
- Delete the duplicated copy.
- Edit the “Privacy” of the original project.
- Open the project.

Task 5

Scenario:

In the workspace, explore the “Sort” method and observe the results, and logout.

Guidelines:

- Add the Sort method from the “Methods” button.
- Drag the method to your favorite position.
- Access the method information.
- Add four names “2”, “5”, “1”, “6”, delete the number “5”, and finally rearrange them by this order, by dragging the boxes, “1”, “2”, “6”, from “Most Important” to “Less Important”.
- Run the workflow.
- Open the Output File.
- Check the results order and graphics.
- Log out.

Task 6

Scenario:

Explore DecSpace in the device mode.

Guidelines:

- Log in with your account.
- Explore the Method Catalog
- Open a project.
- Use the workspace.

B

DecSpace Survey

DecSpace Survey

The purpose of this survey is to understand and analyse the user experience and satisfaction with DecSpace.

Thank you for your collaboration!

***Obrigatório**



Rank the degree of difficulty of using these features *

	Very Difficult	Difficult	Average	Easy	Very Easy
Signing up	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Logging in	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
FAQ page	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method Catalog "User preferences"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method Catalog "Description"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method Catalog "Example"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method Catalog "Step by step"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "User Preferences"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Add Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Open Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Duplicate Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Edit Privacy"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Delete Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PublicProjects "Open Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PublicProjects "Duplicate Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Settings "Change Password"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Workspace "Accessing Methods"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Workspace "Adding Methods"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



Workspace "Input Data"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Workspace "Running Workflow"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Workspace "Graphics"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Device Mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



Rank the degree of aesthetics of using these features *

	Very Bad	Bad	Average	Good	Very Good
Signing up	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Logging in	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
FAQ page	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method Catalog "User preferences"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method Catalog "Description"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method Catalog "Example"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Method Catalog "Step by step"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "User Preferences"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Add Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Open Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Duplicate Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Edit Privacy"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MyProjects "Delete Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PublicProjects "Open Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PublicProjects "Duplicate Project"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Settings "Change Password"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Workspace "Accessing Methods"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Workspace "Adding Methods"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



Workspace "Input
Data"☐☐☐☐☐Workspace
"Running
Workflow"☐☐☐☐☐Workspace
"Graphics"☐☐☐☐☐

Device Mode

☐☐☐☐☐

How easy it was to learn how to use DecSpace? *

1

2

3

4

5

Very Difficult

☐☐☐☐☐

Very Easy

How would you classify DecSpace in terms of usefulness? *

1

2

3

4

5

Very Useless

☐☐☐☐☐

Very Usefull

How satisfied are you with DecSpace? *

1

2

3

4

5

Very
Unsatisfied☐☐☐☐☐

Very Satisfied

SEGUINTE

Nunca envie palavras-passe através dos Formulários do Google.

Este conteúdo não foi criado nem aprovado pela Google. Denunciar abuso - Termos de Utilização - Termos adicionais

Google Formulários



DecSpace Survey

*Obrigatório

User Experience

Classify the sentence: "DecSpace is a innovative framework with a friendly interface" *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Classify the sentence: "DecSpace was very efficient" *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Classify the sentence: "DecSpace has responsive features" *

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

How do you rate the device mode experience *

	1	2	3	4	5	
Very Useless	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very Usefull

What did you like the most about DecSpace?

A sua resposta



What did you like the least about DecSpace?

A sua resposta

Do you have any recommendations for DecSpace?

A sua resposta

Do you have any other observations regarding DecSpace?

A sua resposta

ANTERIOR

SUBMITER

Nunca envie palavras-passe através dos Formulários do Google.

Este conteúdo não foi criado nem aprovado pela Google. Denunciar abuso - Termos de Utilização - Termos adicionais

Google Formulários





Results of the Survey and Evaluation

Table C.1: Average Results of the System Evaluation.

Task	Time	Non-Critical Errors	Critical Errors
1	1.58.00	0.2	0
2	2.33.00	0.15	0
3	2.17.00	0.15	0.1
4	1.48.00	0.2	0.15
5	4.20.00	0.1	0
6	1.05.00	0	0

Table C.2: Average Results of the Survey.

Question	Description	Score
1	Rank the degree of difficulty of using DecSpace features	Above 3.5
2	Rank the degree of aesthetics of using DecSpace features	Above 4
3	How easy it was to learn how to use DecSpace?	4.3
4	How would you classify DecSpace in terms of usefulness?	4
5	How satisfied are you with DecSpace?	4.3
6	Classify the sentence: "DecSpace is a innovative framework with a friendly interface"	4.2
7	Classify the sentence: "DecSpace was very efficient"	4.2
8	Classify the sentence: "DecSpace has responsive features"	4.3
9	How do you rate the device mode experience	4.4