Deliverable D 5.4
Cheap-GSHPs Web Platform
WP5

Grant Agreement number	657982
Project acronym	Cheap-GSHPs
Project full title	**Ch**eap and **E**fficient **Ap**plication of reliable **G**round **S**ource **H**eat Exchangers and **P**ump**s**
Due date of deliverable	31/05/2018 (M36)
Lead beneficiary	8 - ANER
Other authors	Robert Gast (ANER) Lucia Cardoso (ANER)

Dissemination Level

PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission Services)	
CI	Classified, as referred to in Commission Decision 2001/844/EC	

"Cheap-GSHPs project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No.657982".

Document History

Version	Date	Authors	Description
1	01/05/2018	Lucia Cardoso (ANER)	Creation of the document
2	25/05/2018	Robert Gast (ANER), Lucia Cardoso (ANER)	First draft
3	28/05/2018	Robert Gast (ANER), Lucia Cardoso (ANER)	Draft for reviewers
4	30/05/2018	Amaia Castelruiz (TECNALIA)	Reviewed final version
5	31/05/2018	Coordinator (A.Bernardi CNR-ISAC)	Final revision and upload in ECAS

This publication was completed with the support of the European Commission under the *Horizon 2020 research and innovation programme*. The contents of this publication do not necessarily reflect the Commission's own position. The document reflects only the author’s views and the Community is not liable for any use that may be made of the information contained therein.

Contents

Publishable summary

The document describes the development of the Cheap-GSHPs Web Platform, that is, the Cheap-GSHPs Decision Support System View module. The document includes an overall description of the system architecture, the workflows followed by the different modules developed and the definition of the interfaces between the Web Platform and the rest of components of the Cheap-GSHPs DSS.

Besides, the document shows the user interface obtained, with a description of the different pages developed in the platform.

Abbreviations

Cheap-GSHPs	**Ch**eap and **E**fficient **Ap**plication of reliable **G**round **S**ource **H**eat Exchangers and **P**ump**s**
DSS	**D**ecision **S**upport **S**ystem
HTML	**H**yper**T**ext **M**arkup **L**anguage
LCA	**L**ife **C**ycle **A**ssessment
RoI	**R**eturn **o**n **I**nvestment

1 Introduction

This report is about the development of the DSS Web application of the CHEAP-GSHPs project.

From the project webpage http://cheap-gshp.eu/ we have a link to the DSS web application. This web application is an online application, and to use it the user doesn’t need to download anything, and only an Internet connexion is necessary.

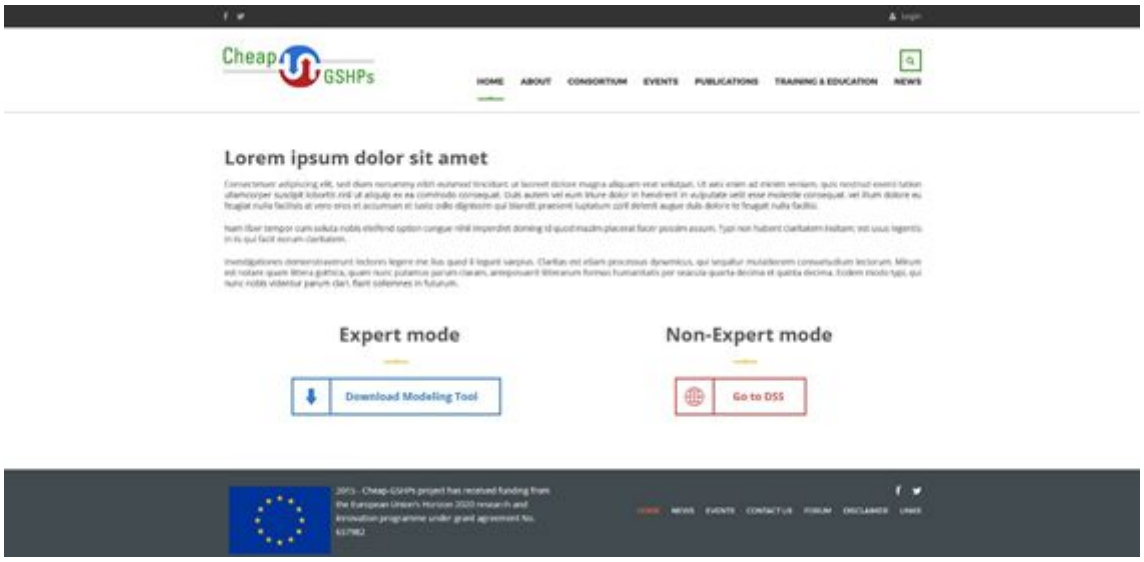

Figure 1 – Webpage to Access to DSS Web Application

As we show in the Figure 2 and it is explained in the D5.3 deliverable, the Cheap-GSHPs Web Platform is composed of three layers:

- DSS View:
  - The user interface, which is being developed in Task 5.4. Located in a Web Platform, the user accesses this view to input the required information.
  - Then, the software launches the different steps of the calculations to be performed. These steps are performed one by one, giving the user the opportunity to stop the process, save the project and resume the process at some other time.
  - Once the process is over, the results are shown to the user. Finally, the user can choose to save the information generated.

- Databases. The system uses three different databases in order to store the information related to each user and project, as well as for storing the information necessary to perform the calculations.

- DSS Engine: The software in charge of performing the calculations is composed of different modules, each of which is responsible for the execution of a specific step in the process. These modules or steps are:
  - Ground / Climate property retrieval
  - Energy Demand calculation
  - Possible Heat Pump alternatives
  - Possible Auxiliary Source alternatives
  - Possible Heat Exchanger alternatives
  - Solution ranking and best option determination

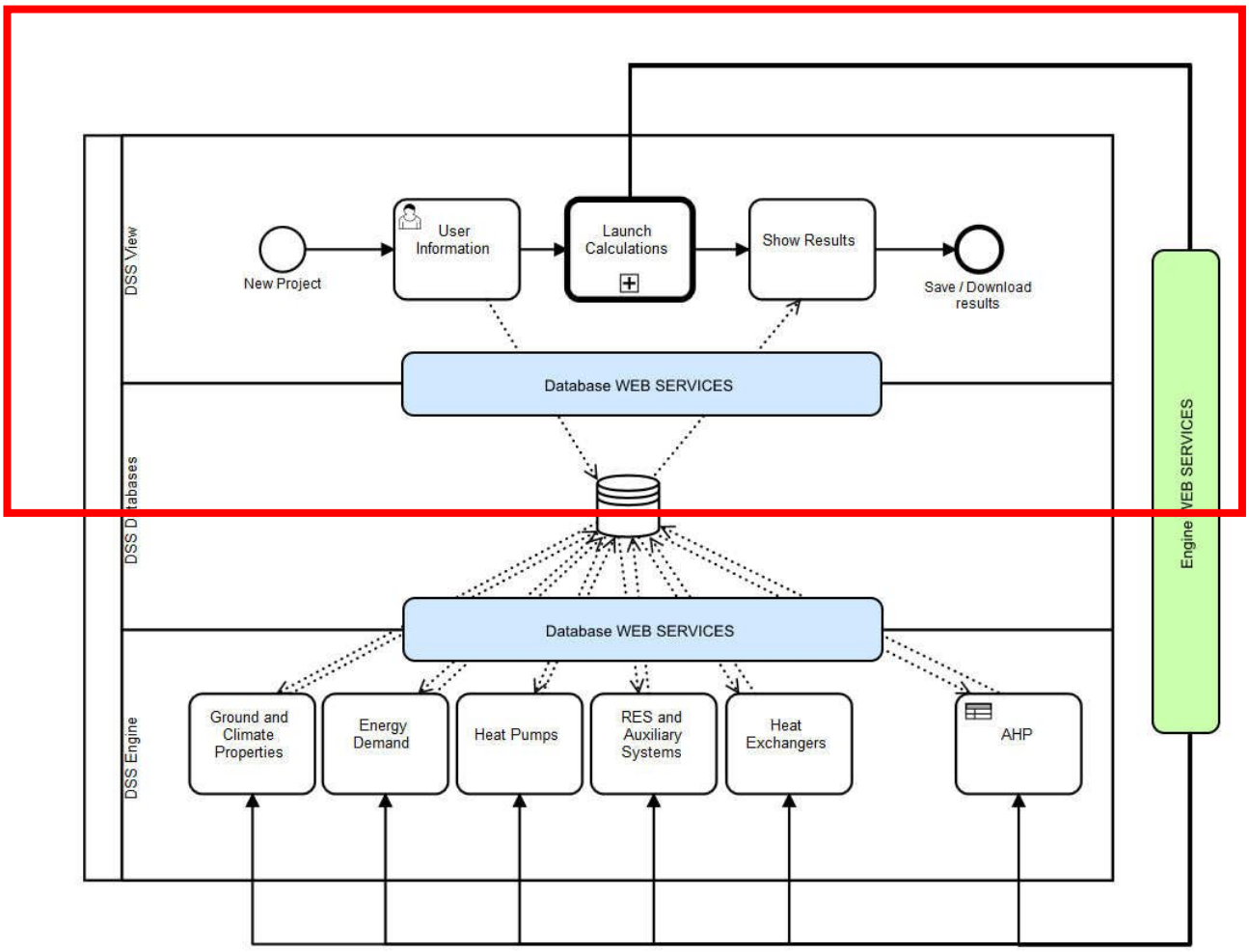

Figure 2 - Cheap-GSHPs Web Platform Structure

The three modules (View, Databases and Engine) need to interact between each other.

- **The View needs to store the information introduced by the user in the Databases.**
- **The View needs to launch each of the different steps of the calculations in the Engine.**
- The Engine needs to get the information necessary for the calculations from the Databases and also to store the results of each step of the calculations.
- **The View needs to access the results of the calculations and show them to the user.**

The D5.3 deliverable contains the description of the DSS Databases, the DSS Engine and the Web Service Engine, while in this deliverable (as we show in Figure 2), we will explain mainly the DSS View module.

2 Overall Description

2.1 DSS View Architecture

Figure 3 shows the architecture of the web application. A HTTP Server communicates through a Web Service (REST Protocol [1]) with the DSS Engine. The user inputs and the results are stored in a Database, and we can access to these data using the Web Service too.

For getting or inserting data into the databases we use the Database Web Services defined, and for launching the different steps of the calculations we use the Engine Web Services defined.

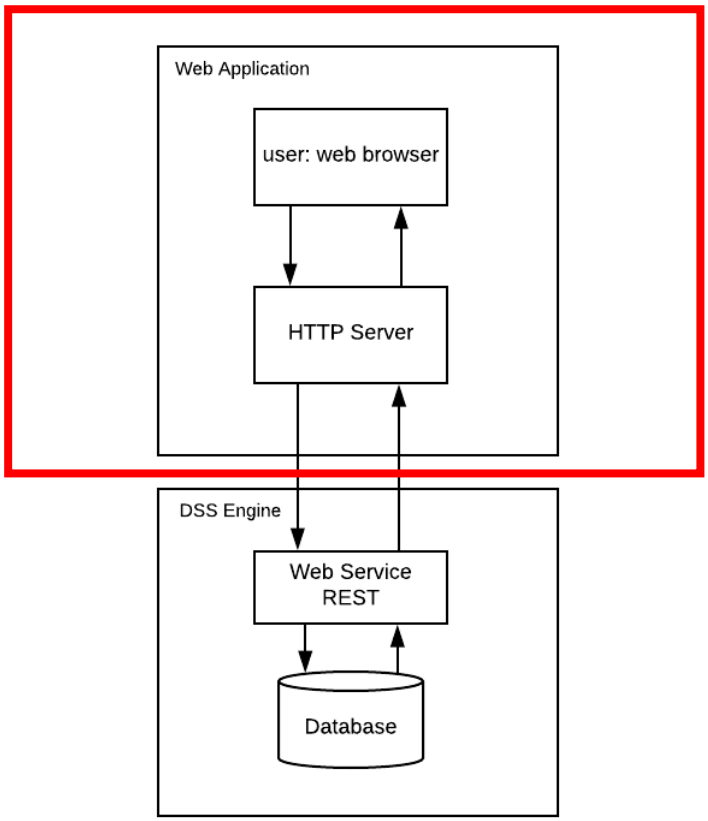

Figure 3 – Architecture Web Application and DSS Engine

2.2 Workflow

Figure 4 shows the workflow implemented by the DSS View.

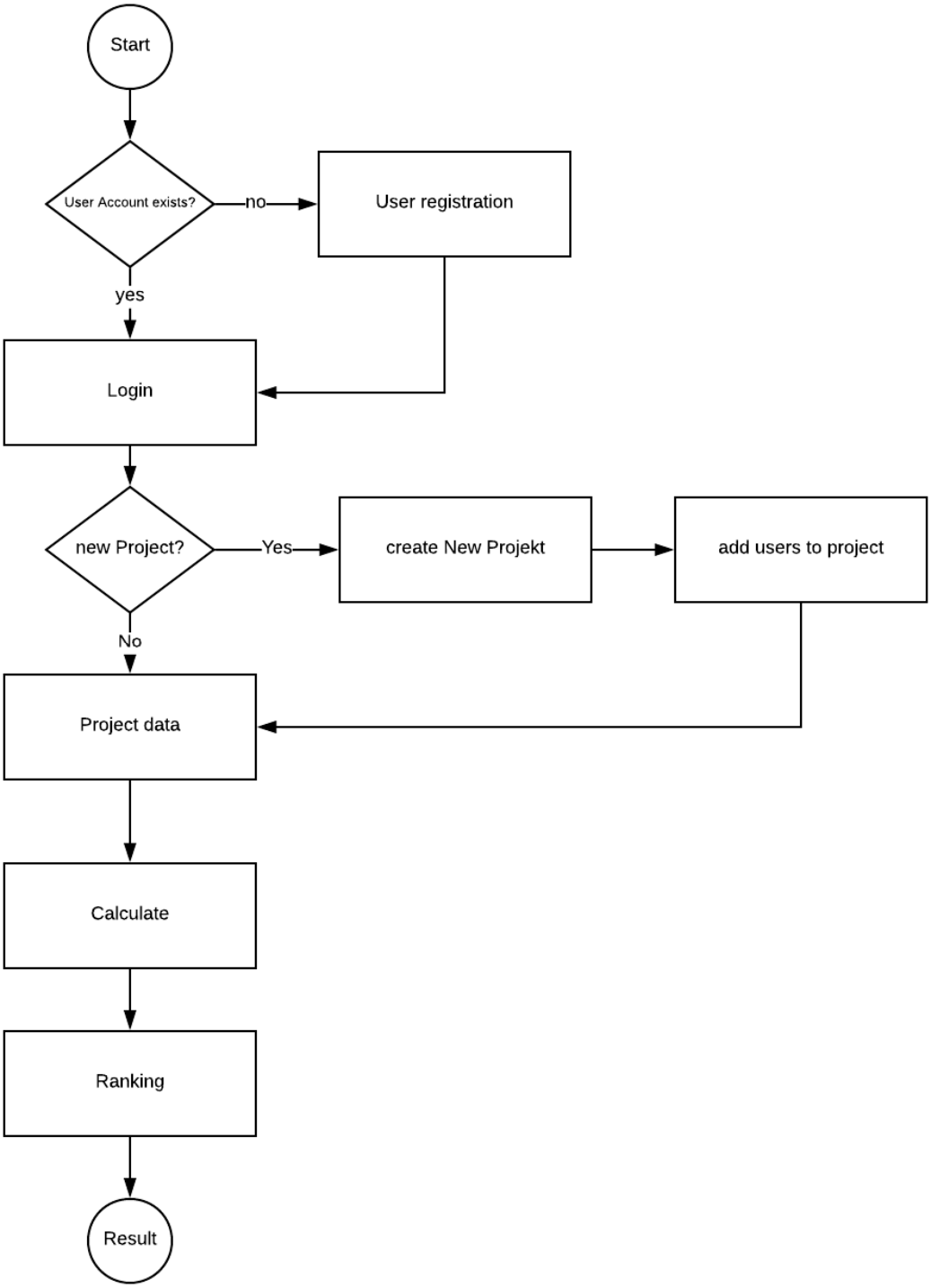

Figure 4 – DSS View Workflow

The user can start the web application, registering if a user account has not been created yet or loging in if the account already exists. Then the user can create a new project or enter in an existing project. In both cases, the user will need to complete the input data (project data) that are necessary for the calculations. Once the user has done this, the application will launch the calculations to obtain the possible solutions. In the next step, when the users obtain the possible solutions, he/she can apply his/her preference from the list of availablecriteria (Cost, RoI, LCA, Used surface), and the application will show the rank of possible solutions according to their properties and the preferences of the user.

3 Web Application (DSS View)

Based on the mock ups from previous tasks we have designed the graphical user interface. To do that the following technologies have been used:

Name	Description
HTML	HTML is the standard markup language for creating web pages and web applications. [2]
JavaScript	JavaScript enables interactive web pages and this is an essential part of web applications. [3]
Ajax	Ajax is a set of web development techniques using many web technologies on the client side to create asynchronous web applications. [4]

The main objective in the task has been mainly to create a user friendly and intuitive interface.

Now, we will briefly explain the workflow that follows the DSS view:

3.1 Register

A user need an account to use the Web Application.

The Web Application use an encryption technique to encrypt the password before its stored in the database.

Figure 5 – Register Form

3.2 Login

With his email address and password, the user can start a session in the DSS Web Application.

In the Login, the system uses the same encryption technique to encrypt the password before it is transmitted to the Database by the Web Service.

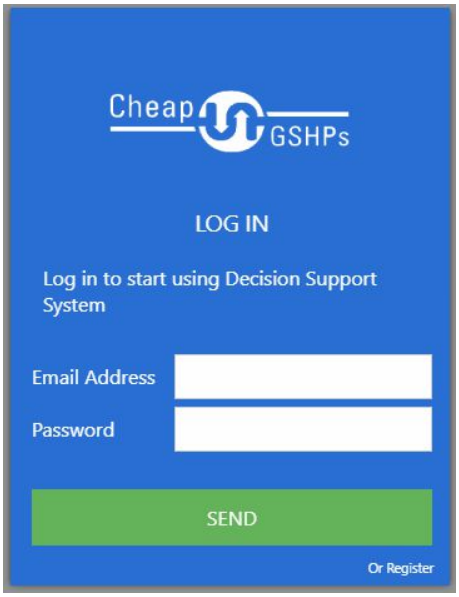

Figure 6 – Login Form

3.3 New Project

Then, as we show in the workflow figure, the next step is to create a Project by Name and Description:

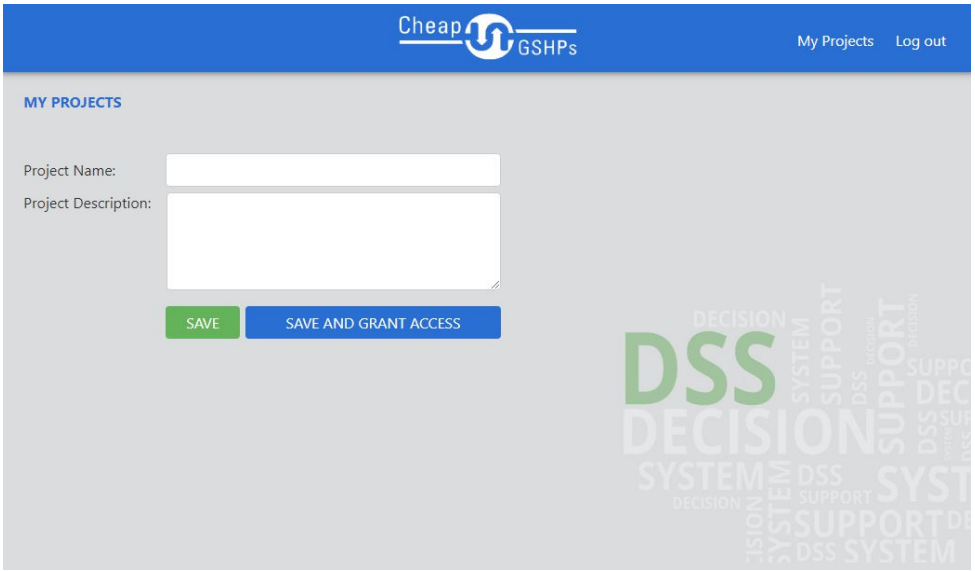

Figure 7 – Create Project Form

In a second step the user can invite other people to join the project, only with read permissions or with read and edit permissions

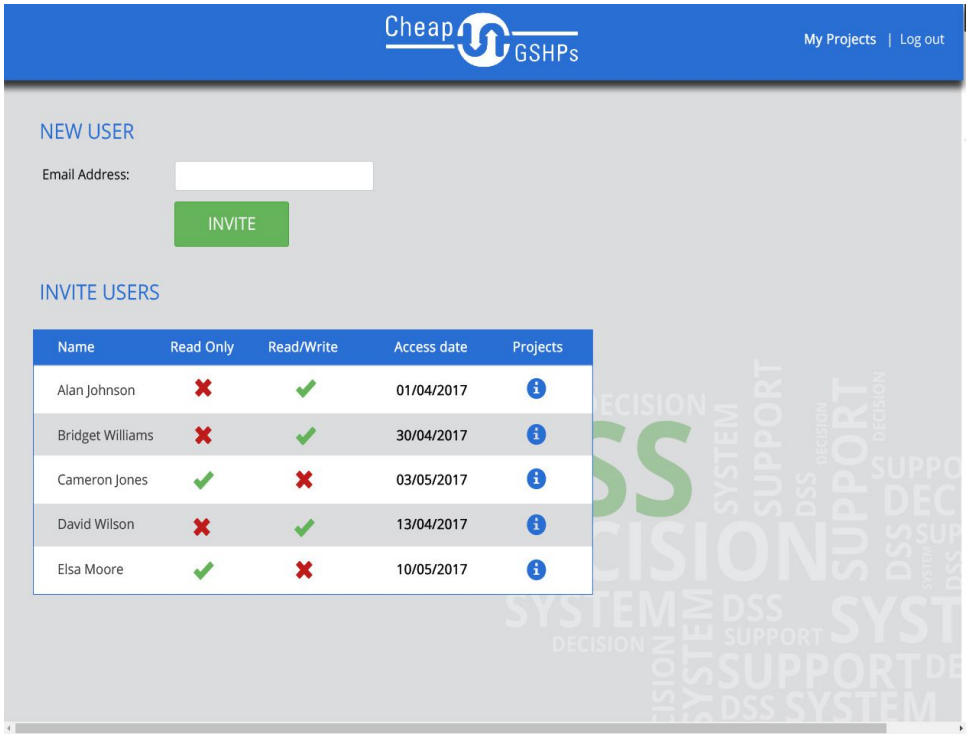

Figure 8 – Invite User Form

3.4 Project Details

A non-expert user can work with this form and indicate his data in an easy way.

The DSS Engine will later use these values for the calculations.

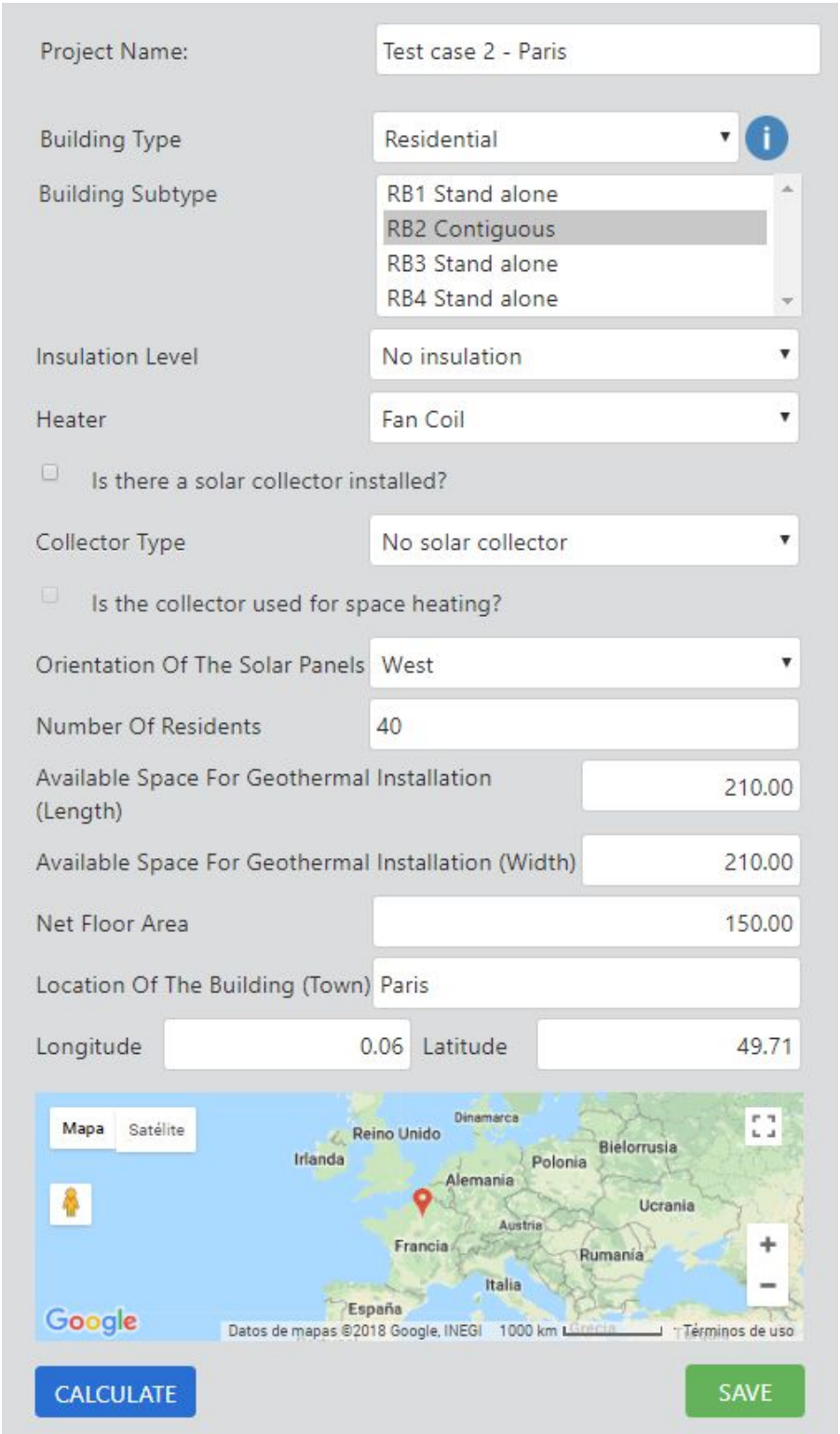

Figure 9 – Project Data or Input Data Form

3.5 Calculate

With the values introduced by the user in the project the DSS Engine will calculate valid scenarios for the user.

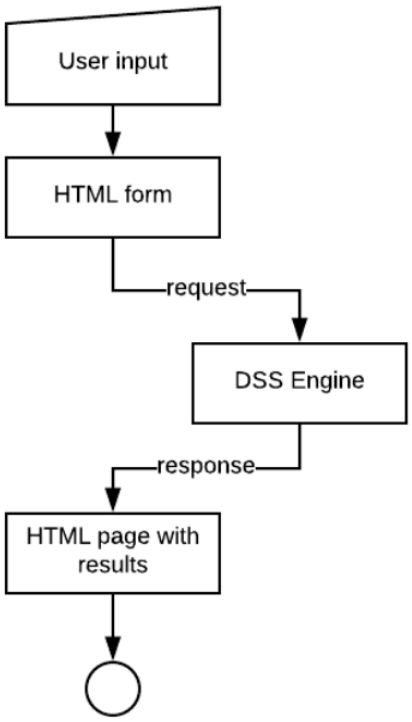

Figure 10 – Workflow Between Web Application and DSS Engine

This example shows how the web application calls the DSS Engine using the web service:

General structure of the web services:

FUNCTION	PROTOCOL	CALL	RESPONSE
Execute action	POST	http://localhost:8080/cheapgshpsengine/rest/[calculationmodule] {”projectid”: ID}	OK / NOK

3.6 Ranking & Results

After the calculations by the software, the web application offers to the user possible solutions.

Then, the user can specify his priorities or preferences for the project. Examples: Roi vs. Space, Roi vs. Cost, etc., and the system will sort the solutions by the criteria from the user.

The calculation and presentation of the possible solutions is done with Shiny, an open source R package that provides an easy and powerful web framework for building web applications using R. [5]

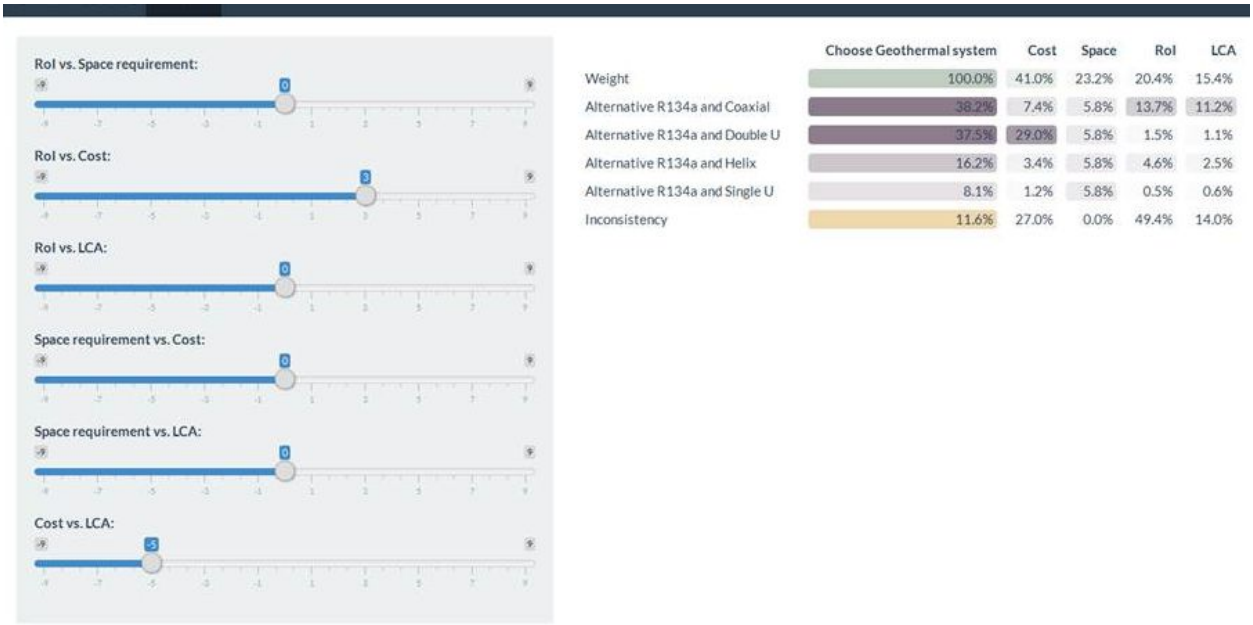

Figure 11 – Ranking and Result Form

4 Conclusion

In this document the development of the Cheap-GSHPs Web Application has been presented. This Web Application is an easy way for non-expert users to use the Cheap-GSHPs DSS Engine.

An overall description of the Web Platform has been presented, including the definition of the architecture of the web application and of the workflow implemented by the DSS View. Besides, the different pages that form the Web Application – including the different options offered to the users - have been presented.

References

[1] Wikipedia: The Free Encyclopedia. Representational state transfer. Wikimedia Foundation, Inc. 25. Mai 2018. Web https://en.wikipedia.org/wiki/Representational_state_transfer
[2] W3school: THE WORLD'S LARGEST WEB DEVELOPER SITE https://www.w3schools.com/html/
[3] W3school: THE WORLD'S LARGEST WEB DEVELOPER SITE https://www.w3schools.com/js/default.asp
[4] W3school: THE WORLD'S LARGEST WEB DEVELOPER SITE https://www.w3schools.com/js/js_ajax_intro.asp
[5] RStudio Inc: https://shiny.rstudio.com/