



**AGENT-BASED
SUPPORT TOOL FOR
THE DEVELOPMENT
OF AGRICULTURE POLICIES**

D4.3 Validated design for the AGRICORE interface



Deliverable Number	D4.3
Lead Beneficiary	AAT
Authors	AAT, IDE
Work package	WP4
Delivery Date	M27* (Extension of the initial deadline (M21) accepted due to Covid-19 situation)
Dissemination Level	Public

www.agricore-project.eu



The Agricore project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No. 816078





Document Information

Project title	Agent-based support tool for the development of agriculture policies
Project acronym	AGRICORE
Project call	H2020-RUR-04-2018-2019
Grant number	816078
Project duration	1.09.2019-31.8.2023 (48 months)
Deliverable Authors	Juan Carlos Castillo (AAT), Rebeca Gutiérrez (AAT), Mercedes Pichardo (AAT), Pablo Báez (IDE), Pablo Gallegos (IDE)
Deliverable Reviewers	Sabine Delaitre (IDE), Fernando Dorado (IDE), Alex Hall (IDE), Peyman Uysal (AKD)

Version History

Version	Description	Organisation	Date
0.1	ToC definition	AAT and IDE	17 Mar 2021
0.2	Development of sections	AAT and IDE	21 Oct 2021
0.3	First validation phase completed	AAT and IDE	12 Nov 2021
0.4	Second validation phase completed	AAT and IDE	20 Nov 2021
0.5	Review of the established requirements	AAT and IDE	22 Nov 2021
1.0	Deliverable reviewed and complete	AKD, AAT and IDE	29 Nov 2021

Disclaimer

All the contributors to this deliverable declare that they:

- Are aware that plagiarism and/or literal utilisation (copy) of materials and texts from other Projects, works and deliverables must be avoided and may be subject to disciplinary actions against the related partners and/or the Project consortium by the EU.
- Confirm that all their individual contributions to this deliverable are genuine and their own work or the work of their teams working in the Project, except where is explicitly indicated otherwise.
- Have followed the required conventions in referencing the thoughts, ideas and texts made outside the Project.

Executive Summary

AGRICORE is a research project funded by the European Commission under the RUR-04-2018 call, part of the H2020 programme. AGRICORE proposes an innovative way to apply agent-based modelling to improve the capacity of policymakers to evaluate the impact of agricultural-related measurements under and outside the framework of the Common Agricultural Policy (CAP). The resulting AGRICORE tool will be tested in three use cases, each of them having a special focus on a specific type of impact assessment (IA): environmental IA, ecosystem services IA and socio-economic IA.

This document presents the design of the user interface (UI) of the future AGRICORE tool, as well as the process followed to obtain it. This design process includes the following phases: i) definition of requirements, ii) analysis of similar pre-existing tools for public policy impact assessment, iii) comparison and selection of tool(s) for the construction of the mockups that represent the design of each of the screens of the user interface, iv) design of the screens and construction of the mockups and v) validation of the design by the project partners and other external stakeholders. The last two phases have been executed in an iterative loop, so that inputs from subsequent user evaluation processes feedback into changes to the design, which are then re-evaluated by previous and new testers until a final design is reached.

The initial idea was to implement this (re)design-(re)evaluation process through face-to-face workshops in which the tool could be presented to its future users, in order to obtain their feedback directly. Due to the Coronavirus pandemic, this approach had to be delayed for a few months. Even so, the face-to-face evaluation approach could only be partially realised and had to be replaced by the implementation of online evaluation forms that were sent to potential users and external stakeholders, not always obtaining the desired number of responses.

The first section of the document is the Introduction, which presents the modular architecture of the AGRICORE suite and the importance of the User Interface as a means to access the different tools of the suite independently. It also explains the need for the different modules to work together, as the result of actions executed from the UI, to produce the expected results by the user.

Section 2 briefly reviews other tools previously developed in the EU framework for public policy impact assessment. EUROMOD and CAPRI were analysed as designs from which to draw inspiration.

Section 3 describes the design approach for the GUI. First, the usability principles that have guided the whole process of user-centred design are presented. Then, an analysis of the commercially available tools for mockup design is presented, along with the reasons to finally choosing MarvelApp as the one to be utilised for the design of the AGRICORE UI visual appearance. Finally, the exhaustive list of Functional and Non-functional Requirements for the GUI are listed.

Section 4, which is the core of this document, describes the proposed design by dividing it into as many subsections as there are elements in the tool. It also describes the possible operation flows that the user can follow through the interface. Beforehand, the main concepts of the GUI design (colour palette, table appearance, interacting files, etc.) are introduced.

Section 5 describes the changes that were made to the design of the different screens and windows as a result of the different iterations of evaluation by future users, until the final validated design was reached.

The final conclusions of the AGRICORE user interface design are summarised in Section 6.

Abbreviations

Abbreviation	Full name
ABM	Agent Based Model
ABM-e	Agent Based Model Engine
AH	Agricultural Holding
ARDIT	Agricultural Research Data Index Tool
CAP	Common (EU) Agricultural Policy
DCAT-AP	Data Catalogue Vocabulary - Application Profile
DWH	Data Warehouse
EAB	Expert Advisory Board
ES	Economic Size
EU	European Union
FR	Functional Requirement
GUI	Graphical User Interface
JRC	Joint Research Center
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
IAM	Impact Assessment Module
NFR	Non-functional Requirement
PD	Policy Description
SPG	Synthetic Population Generation
TEO	Techno-Economic Orientation
TF	Type of Farming
UK	United Kingdom
UI	User Interface
UX	User Experience
XML	Extensible Markup Language

List of Figures

Figure 1 Cycle of Policy Evaluation using the AGRICORE tool.....	7
Figure 2 AGRICORE's modular architecture with its modules grouped according to the user functionality to which each one contributes	8
Figure 3 EUROMOD desktop application interface	10
Figure 4 EUROMOD toolbar options.....	11
Figure 5 EUROMOD-JRC simulation setup	12
Figure 6 EUROMOD-JRC simulation results file.....	13
Figure 7 EUROMOD-JRC simulation history.....	13
Figure 8 CAPRI interface screenshots.....	14
Figure 9 Factors affecting interface design.....	16
Figure 10 Interface design lifecycle.....	19
Figure 11 AGRICORE interface template	26
Figure 12 AGRICORE interface top bar.....	27
Figure 13 AGRICORE interface colour palette.....	27
Figure 14 AGRICORE synthetic population generator tool colour palette	28
Figure 15 Detail of a table on the platform.....	28
Figure 16 AGRICORE interface operation flow	29
Figure 17 Login screen.....	31
Figure 18 Registration screen	31
Figure 19 Main page screen	32
Figure 20 Simulation setup lobby.....	33
Figure 21 Simulation setup lobby.....	33
Figure 22 Screen for searching existing populations	35
Figure 23 Screen for the synthetic population selection	35
Figure 24 Policy selection screen	37
Figure 25 Catalogues tables design.....	37
Figure 26 Pop-up to rate policy definitions.....	38
Figure 27 Policy rating elements	38
Figure 28 Pop-up to edit policy definitions	40
Figure 29 Simulation horizon screen	41
Figure 30 Solvers and biophysical models screen.....	42
Figure 31 KPIs selection screen	43
Figure 32 Synthetic population generator tool.....	45
Figure 33 Detail of the selection of datasets extracted from ARDIT	45
Figure 34 Synthetic population generation finished	46
Figure 35 Simulation setup completed	47
Figure 36 Simulation module screen	48
Figure 37 Visualisation module screen without any data loaded	49
Figure 38 Visualisation module screen with the results of a simulation loaded	49
Figure 39 Personal catalogue of policies screen.....	50
Figure 40 Platform's general catalogue of policies screen.....	51
Figure 41 Sidebar detail with the new changes.....	55
Figure 42 Some examples of data added	56
Figure 43 Policy rating icon change.....	57
Figure 44 MarvelApp project lobby.....	61
Figure 45 MarvelApp page to share the prototype	62
Figure 46 Tool palettes in MarvelApp	63
Figure 47 MarvelApp prototyping.....	64

List of Tables

Table 1 List of mock-up tools analysed	20
Table 2 Some examples of farm types according to their technical-economic orientation.....	44
Table 3 List of the 14 types of farms according to their economic size.....	44

Table of Contents

1 Introduction	7
2 Previous work on policy assessment.....	10
2.1 EUROMOD.....	10
2.2 CAPRI	14
2.3 Insights for the design of the AGRICORE GUI	14
3 Design approach	16
3.1 Design and usability principles	16
3.1.1 Usability principles.....	17
3.1.2 Colour as a key cognitive aspect of the interface.....	18
3.1.3 User-centred design.....	18
3.2 Design procedures and tools.....	19
3.2.1 Study and selection of design tools	19
3.3 Requirements of the AGRICORE User Interface	21
3.3.1 Functional requirements.....	21
3.3.2 Non-functional requirements.....	25
4 AGRICORE Graphical User Interface.....	26
4.1 Main concepts of the GUI design.....	26
4.1.1 AGRICORE desktop application.....	26
4.1.2 AGRICORE GUI colour palette	27
4.1.3 Files interacting with the AGRICORE GUI	28
4.1.4 Tables and icons	28
4.2 AGRICORE Graphical User Interface: user navigation and screen design	29
4.2.1 User workflow	29
4.2.2 User access management	30
4.2.3 AGRICORE platform main page	32
4.2.4 Simulation setup GUI sub-module	32
4.2.5 Synthetic population generator GUI sub-module.....	43
4.2.6 Simulation GUI sub-module	47
4.2.7 Visualisation GUI sub-module.....	48
4.2.8 General and personal catalogue of policies	50
5 Design validation procedures.....	52
5.1 First validation phase	54
5.2 Second validation phase.....	56
6 Conclusions.....	58
References	59
ANNEX I: Mockup prototyping using MarvelApp.....	61
MarvelApp tool.....	61
MarvelApp prototyping.....	63

1 Introduction

AGRICORE is a Research and Innovation Action funded by the European Commission under the call RUR-04-2018 whose objective is to design and build an Agent-based-modelling tool to carry out impact assessments of different measures and instruments alternatively implementable within the Common Agricultural Policy (CAP). These impact assessments fall into three main categories (socio-economic, environmental and delivery of ecosystem services) and will serve both to assess a posteriori the effects of measures already implemented (ex-post analysis) and to predict a priori the impact of measures in the design phase (ex-ante analysis). The analysis of the effect of the different policy measures is carried out through the computation of a series of key performance indicators (KPIs) within the respective impact assessment modules (IAMs). For this purpose, these modules need to receive as input arguments the results of the agent-based simulation, i.e. the sequence of actions and states that each agent (representing an agricultural holding) has followed during the simulation (see Figure 1, reproduced from AGRICORE's deliverable D6.1). In order to configure the simulation, to launch it and to visualise the resulting KPIs, the future AGRICORE user needs a graphical user interface (GUI) that allows him/her to interact easily with the main tool, but also to access other tools included in the suite (such as the ARDIT data source index, the module that enhances synthetic population generator (SPG), or the Policy Description (PD) catalogue, among others).

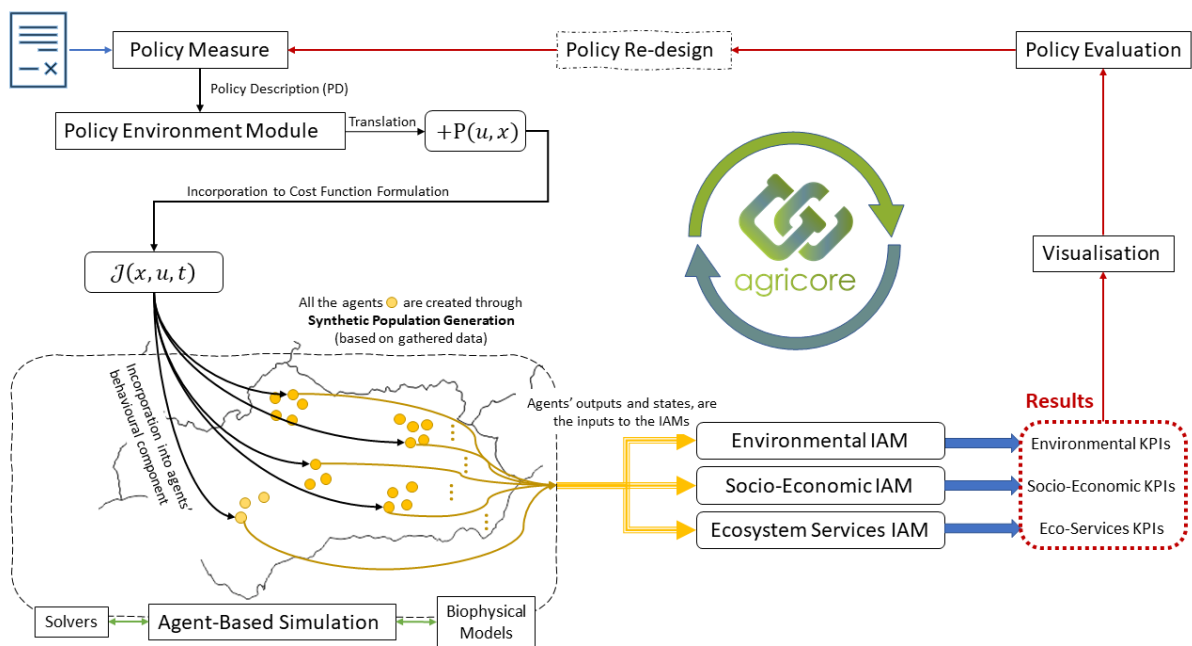


Figure 1 Cycle of Policy Evaluation using the AGRICORE tool

The main purpose of the present document is to present the design done for such AGRICORE GUI. To that end, a process was followed to gather requirements, to represent through graphical sketches (mockups) how the software interface would look like, and to finally develop and validate those designs. The AGRICORE interface will centralise user interaction with the AGRICORE suite, developed as a cross-platform desktop application using web technologies. The interface will allow users to configure and launch simulations based on several configurable elements. For example, selecting the synthetic population of agents representing the real farmers' population of interest, choosing the set of policies to be simulated or selecting the KPIs to be computed and calculated during the simulation process, among others. Once the simulations have

been completed, users will be able to use the interface to visualise the results graphically in tables, charts, graphs or plots.

In addition, the document includes a comparative analysis of other interfaces used in the field of policy assessment, a section dedicated to good practices and design principles applied during our interface construction process, and a final chapter describing how the interface design validation process has been carried out, based on the evaluation done by the AGRICORE project partners and other stakeholders.

Figure 2 shows the different modules that make up the AGRICORE architecture, grouped in three columns according to the functionality to which they contribute. The three sub-architectures, although represented separately, are connected to each other. The first column contains all modules related to the generation of synthetic populations, which is the main step in the process of setting up an AGRICORE simulation. In the middle column are the modules involved in the simulation process itself. Finally, on the right side, we find the module with the graphical user interface (GUI) of the AGRICORE suite, which allows the user to access the various tools of the suite, configure and run simulations and visualise the simulation results.

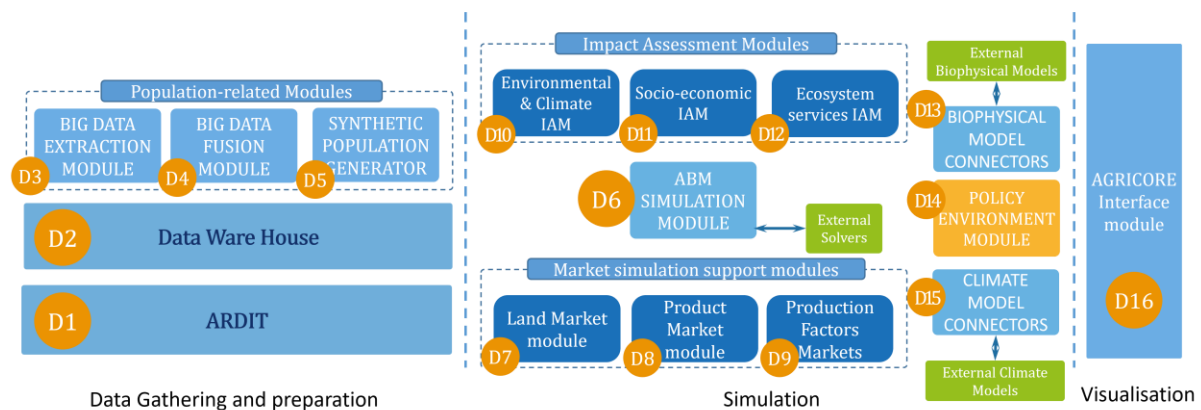


Figure 2 AGRICORE's modular architecture with its modules grouped according to the user functionality to which each one contributes

The 16 different modules have been defined as follows:

- **D1 - ARDIT (Agricultural Research Data Index Tool, previously referred to as European Datasource index tool):** The tool allows users to search for different sources of publicly available data on the Web, categorised by the methodology implemented according to the ontology AGRICORE DCAT-AP 2.0 extension.
- **D2 - Data Warehouse (DWH):** Tool responsible for the ingestion, storage, management and processing of data within the AGRICORE project.
- **D3 - Data Extraction Module:** This module extracts all the data of interest from multiple datasets considered in the project and stored in the DWH. Data extraction encompasses the capabilities for accessing different datasets, selecting the necessary data and formatting it for further processing.
- **D4 - Data Fusion Module:** Combine the individualised data with the probability distributions of the variables to generate the joint probability distributions.
- **D5 - Synthetic Populations Generator:** Module aimed to obtain a realistic synthetic population making use of the Synthetic Reconstruction method.

- **D6 - Agent Based Model Simulation Engine (ABM-e):** Instantiates agents for each farmer generated, evaluating its situation and making decisions based on its preferences.
- **D7, D8, D9 - Multi-Market Modules:** They allow the interaction of the agents present in the ABM-e through the main markets that affect the agricultural sector: the land market (D7), the market for products derived from agricultural and livestock activity (D8) and the markets for agricultural production factors (D9) such as labour, fertilisers and pesticides, or machinery.
- **D10, D11, D12 - Impact Assessment Modules:** Provides different modules used to evaluate the KPI's (Key Performance Indicators) related to the specific topics of each module (respectively Environmental / Climate KPI's/ Socio-economic and ecosystem services).
- **D13 - Biophysical Model Connectors:** They allow the ABM-e and other modules to be connected with the pre-existing biophysical models. Thus, these models can provide the AGRICORE suite with microservices for generating biological, edaphological and agronomic conditions for simulation, as well as for predicting the effects on the biotic and abiotic factors of the ecosystem as a consequence of the actions of the agents.
- **D14 - Policy environment module:** Defines different policies and translate them as input for the simulation engine.
- **D15 - Climate Model Connectors:** They allow connecting ABM-e and other modules with pre-existing weather and climate models. Thus, these models can provide the AGRICORE suite with microservices for the generation of climate conditions for simulation, as well as for the prediction of climate effects based on the decisions of the agents.
- **D16 - AGRICORE interface module:** Centralises all the interactions of the user with the AGRICORE tool, allowing to analyse the results of the simulations defined and performed along the simulation process.

It is this AGRICORE User Interface (D16) module that is described in this document. To do so, an analysis of pre-existing works and tools is first realised, and based on usability guidelines and principles, the methodology used for the design is detailed and the design itself is presented.

As a preliminary phase to the design of the AGRICORE interface, the state of the art of policy assessment tools were analysed, in order to learn how these projects had solved the problems or issues related to simulations, and to guide the AGRICORE design. In the end, EUROMOD and CAPRI tools were selected and analysed.

"EUROMOD is a tax-benefit microsimulation model for the European Union and the UK that enables researchers and policy analysts to calculate, in a comparable manner, the effects of taxes and benefits on household incomes and work incentives for the population of each country and for the EU as a whole. Cross-country comparability is enabled by coding the policy systems of the EU Member States and UK according to a common framework based on a standard set of modelling conventions...Underpinning the model is the purpose-built software comprising a user-friendly interface, supplemented by extended functionalities (plugins and add-ons) for special purpose analysis...EUROMOD is maintained, developed and managed by the Joint Research Centre (JRC) of the European Commission, in collaboration with Eurostat and national teams from the EU countries. The model is open source since December 2020, with an increasing number of users across Europe and beyond."^[1]

[illegible]

Previous work on policy assessment – 10

Figure 4 shows a close-up of the top toolbar of the interface:

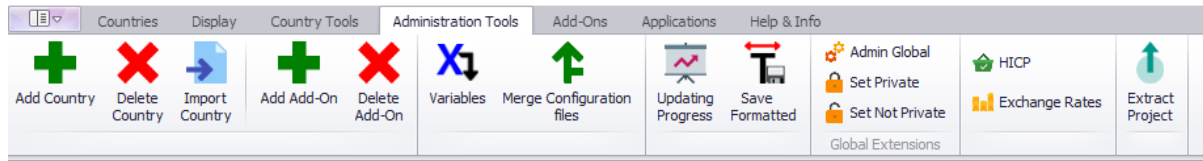


Figure 4 EUROMOD toolbar options

However, a more simplified version of the tool is also available as a web application. The EUROMOD-JRC provides a more user-friendly access to the simulation model allowing to simulate non-complex reforms without requiring any knowledge of the original tool. In order to use this tool, it is necessary to create a user account and to fill in a form to explain the reasons and uses that are wished to make of the application. Once logged in, the platform redirects the user to the configuration of the simulation parameters, selecting first, the country and the year on which we wish to base the simulation.

With the year and country selected, the application automatically fills in all the fields that compose the configurable parameters or attributes with default values, which can be freely modified by the user. Once the configuration of the parameters has been completed, the simulation can be executed. EUROMOD-JRC has two types of simulations depending on the type of configuration selected to feed data to them. Firstly, there is the survey microdata, which include datasets based on microdata from EU Statistics on Income and Living Conditions survey (EU-SILC). Secondly, there are the hypothetical data, in this case, the simulation makes use of experimental data that are built through a plugin of the tool. The process of running the simulation is a blocking process. Once the simulation is launched, the browser window is locked until the simulation is finished, which can take a couple of minutes.

EUROMOD screen for configuring the simulation parameters can be seen in Figure 5:

EUROMOD-JRC Interface

[Home](#)
[Country Selection](#)
[News](#)
[My Simulations](#)
[Previous Systems-Archive](#)

Spain

Spain

2020

☒ Survey microdata (SILC)
 ☐ Hypothetical data (HHoT)

Submit
 Cancel

Simulations based on HHoT are experimental

Personal Income Tax

Schedule	Annual income components (EUR)		Marginal tax rate
Tax Allowances	Bracket 1	> 0 ≤ 12450	0.095
	Bracket 2	> 12450 ≤ 20200	0.12
	Bracket 3	> 20200 ≤ 35200	0.15
	Bracket 4	> 35200 ≤ 60000	0.185
	Bracket 5	> 60000	0.225

Social Insurance Contributions

Employees	Pensions rate (full-time workers)	0.047
	Pensions rate (part-time workers)	0.047
Employers	Pensions rate (agrarian workers)	0.047
Self-Employed	Unemployment rate (full-time workers)	0.0155
	Unemployment rate (part-time workers)	0.0155

Figure 5 EUROMOD-JRC simulation setup

After finishing the simulation, the application redirects the user to a new screen where the results can be downloaded in an xlsx format file, with all the data organised in tables. The interesting thing about the results file is that it contains a sheet where the simulation parameters can be observed, allowing the comparison of the values taken by default with the values that the user has introduced or modified. Figure 6 shows the results file of a simulation launched in EUROMOD-JRC, where the comparison between the values taken by default and those introduced by the user, can be observed marked with a blue colour. Also, at the bottom, the different sheets that classify the results obtained can be found:

SIMULATION PARAMETERS - ES 2020
 9 Nov 2021 09:22:06

PERSONAL INCOME TAX

Schedule

Field	Original Value	Modified Value
Bracket 1 - Marginal tax rate	0,095	0,08
Bracket 1 - Upper threshold	12350	12350
Bracket 2 - Marginal tax rate	0,12	0,12
Bracket 2 - Upper threshold	20200	20200
Bracket 3 - Marginal tax rate	0,15	0,15
Bracket 3 - Upper threshold	35200	35200
Bracket 4 - Marginal tax rate	0,185	0,2
Bracket 4 - Upper threshold	60000	60000
Bracket 5 - Marginal tax rate	0,225	0,225

Tax Allowances

Field	Original Value	Modified Value
Personal - Basic	5550	5550
Personal - Basic - older than 65	1150	1150
Personal - Basic - older than 75	1400	1400
Personal - Basic - joint taxation for lone-parents	2150	2150
Personal - Basic - joint taxation if married	3400	3400
Family - Deduction - 1st child	2400	2400
Family - Deduction - 2nd child	2700	2700
Family - Deduction - 3rd child	4000	4000
Family - Deduction - 4 or more children	4500	4500
Family - Complement for children less than 3	2800	2800
Family - Dependent parent - amount	1150	1150
Family - Dependent parent - income limit	8000	8000
Family - Dependent parent complement - older than 75	1400	1400
Work-related - Employment income - amount below 1st limit	5700	5700
Work-related - Employment income - 1st limit	11250	11250
Work-related - Employment income - 2nd limit	14450	14450
Work-related - Employment income - amount over 2nd limit	2000	2000
Work-related - Employment income - slope in between	1,15625	1,15625
Work-related - Employment income - limit for other income	6500	6500

Housing - No tax allowances reforms are offered for Housing.

Education - No tax allowances reforms are offered for Education.

Index	Simulation Parameters	Without children - data	With children - data	Without children - shares	With children - shares	Without children - graphs	With children - graphs	Metadata	⊕
-------	-----------------------	-------------------------	----------------------	---------------------------	------------------------	---------------------------	------------------------	----------	---

SOCIAL INSURANCE CONTRIBUTIONS

Employees

Field	Original Value	Modified Value
Pensions rate (full-time workers)	0,047	0,047
Pensions rate (part-time workers)	0,047	0,047
Pensions rate (agrarian workers)	0,047	0,047
Unemployment rate (full-time workers)	0,0155	0,0333
Unemployment rate (part-time workers)	0,0155	0,0333
Unemployment rate (agrarian workers)	0,0155	0,0155
Other SIC (full-time workers)	0,001	0,001
Other SIC (part-time workers)	0,001	0,001
Other SIC (agrarian workers)	0,0003	0,0003

Employers

Field	Original Value	Modified Value
Pensions rate (full-time workers)	0,236	0,236
Pensions rate (part-time workers)	0,236	0,236
Pensions rate (agrarian workers)	0,191	0,191
Unemployment rate (full-time workers)	0,055	0,055
Unemployment rate (part-time workers)	0,055	0,055
Unemployment rate (agrarian workers)	0,055	0,055
Other SIC (full-time workers)	0,008	0,008
Other SIC (part-time workers)	0,008	0,008
Other SIC (agrarian workers)	0,0025	0,0025

Self-Employed

Field	Original Value	Modified Value
Pensions basic rate (non-agrarian workers)	0,283	0,283
Pensions reduction coefficient (non-agrarian workers already contributi)	0,045	0,045
Pensions basic rate (agrarian workers)	0,1875	0,1875
Pensions additional rate (agrarian workers)	0,033	0,033
Work-related injuries (non-agrarian workers)	0,009	0,009
Work-related injuries (agrarian workers)	0,01	0,01
Other SIC (non-agrarian workers)	0,008	0,008

Figure 6 EUROMOD-JRC simulation results file

The tool also has a section where the user can access a list with all the simulations that have been launched by him/her. Being able to edit titles and descriptions, delete them or access again to the simulation configuration page, loading the simulation parameters with those used when it was executed for the first time. Figure 7 illustrates the historical of simulations launched by the user:

European Commission > Business, Economy, Euro > Accounting and taxes > EUROMOD-JRC Interface					
EUROMOD-JRC Interface					
Home	Country Selection	News	My Simulations	Previous Systems-Archive	▼
My Simulations					
Title	Description	Country	Year	Date	
CY_2020_20211109_134018		CY	2020	09/11/2021	✗ ✎
ES_2020_20211109_092239		ES	2020	09/11/2021	✗ ✎
FR_2019_20211026_170144		FR	2019	26/10/2021	✗ ✎

Number of results: 3

Figure 7 EUROMOD-JRC simulation history

2.2 CAPRI

"CAPRI is a success story of an economic model developed by European Commission research funds. Operational since almost a decade, it supports decision making related to the Common Agricultural Policy based on sound scientific quantitative analysis. CAPRI is only viable due to its Pan-European network of researchers which based on an open source approach tender together for projects, develop and maintain the model, apply it for policy impact assessment, write scientific publications and consult clients based on its results" [2]

Like EUROMOD, CAPRI also has a desktop application to launch the simulations and exploit the results. But, in order to use it, the user needs to have installed **Java** and **GAMS** programming languages, on which the platform is built. Given the complexity of GAMS and the lack of knowledge that many users may have towards it, CAPRI organises training sessions to familiarise the users with the GUI.

However, being GAMS a programming language that needs a licence to use it, and because of the complexity of the tool, it has not been possible to test CAPRI or see the organisation and functionalities of its interface completely, except some screenshots and descriptive texts found in user manuals on its official website. Some of these screenshots are shown in the Figure 8:

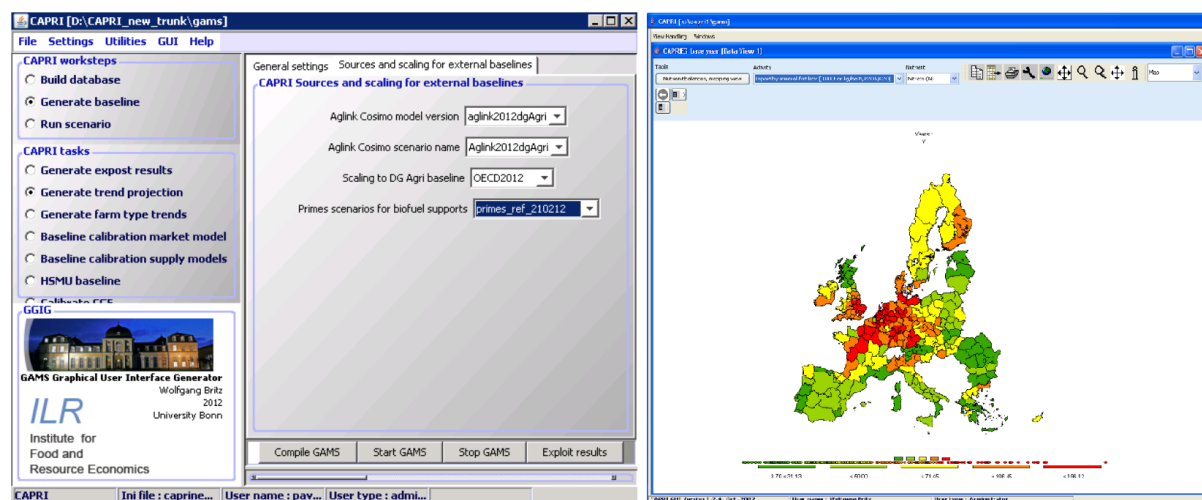


Figure 8 CAPRI interface screenshots

2.3 Insights for the design of the AGRICORE GUI

From the analysis of the EUROMOD and CAPRI desktop applications, not many similarities or features applicable to the objectives and ideas of the AGRICORE interface could be extracted. The more accessible and simpler character of the AGRICORE interface is, somewhat, more related to the web version of EUROMOD (EUROMOD-JRC), where two main sections can be identified, the simulation setup and the list of completed simulations. Unlike the web version of EUROMOD, which blocks the user interaction until the simulation is completed, AGRICORE should allow the user to continue working in the tool once a simulation has been launched, as it could take several hours to complete. Since the simulation will be in the background, the platform must have a section in which the user can visualise, at any moment, the state of those simulations that have been launched and have not been completed yet.

In addition, EUROMOD, once the country and year have been selected, automatically loads the data on the attributes of the policies applied at that time. AGRICORE, on the other hand, should give the user the possibility to select the policies he/she wants to include, even policies that are applicable at different regional levels. For this reason, the platform should have a section where all policies that can be evaluated on the platform are stored. Users will be able to use this collection to customise it when adding them to the simulation setup.

3 Design approach

3.1 Design and usability principles

The interface comprises all aspects and processes that serve as a link between users and a given software, including all the ways in which the user interacts and all the elements present in it. The process of designing an interface is a fundamental part when developing an application or platform, not only because the interface constitutes a large part of the programming code of the project, but also because it is the visible and usable point for the users, which they will use to achieve the desired goals. For this reason, the interface must be pleasant, comprehensive and easy to use for the user, as this can guarantee an increase in traffic and number of users, the use of the tool, the achievement of its objectives and, in short, the success or failure of a project.

Nowadays, there are standards and excellent practice guides that allow developers to improve the usefulness and effectiveness of the interface. To make a usable design it is necessary to know not only patterns commonly used for good aesthetics, but also, psychological and organisational factors to know how people work and make use of the interfaces. The term **user experience (UX)** is a complex concept that encompasses all those factors that affect the interface. From the styles, *look & feel* and layouts (UI), to the navigation, interaction and the thoughts, emotions and reasoning that users feel when using the interface.

In the end, for software to fulfil its objectives, it has to be usable and accessible to as many users as possible. This means that it must be easy to learn, allowing the user to perform tasks faster and faster, and easy to use, the use must be simple and natural[3][4][5][6].

The following picture (Figure 9) shows an example of the factors that comprise the UX[7]:

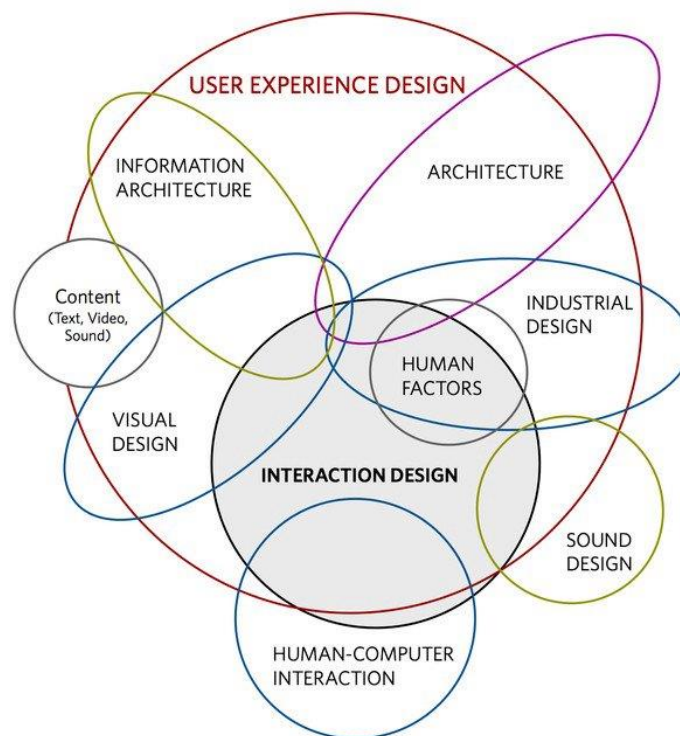


Figure 9 Factors affecting interface design

In the case of the interface designed for AGRICORE project, all possible recommendations and best practices for styling, navigation and usability have been followed. Additionally, the component libraries for web-based interfaces **Angular Material**[\[8\]](#) and **Bootstrap**[\[9\]](#), have served as inspiration and basis for the design of certain components and elements of the interface. Some general principles that have been applied during the designs are described below:

3.1.1 Usability principles

The main points when designing software and making it easy to use, include:

- **Easy to learn:** the time that the user takes from the first access (without any knowledge) to the productive use of the application should be reduced to a minimum. In addition, the system should include support for less-advanced users, in order to achieve the maximum level of knowledge and productivity. To achieve this objective, the application needs to be:
 - **Synthesisable:** any change in the application must be captured by the user.
 - **Familiar:** there must be a correlation between the user's knowledge of the real world, metaphors (icons, terms, concepts) and other similar applications or systems, with the knowledge that is required to use the application in question.
- **Flexibility:** all the ways in which the user can interact with the interface or exchange information. This is done by giving control to the user, allowing, for example, undoing, cancelling or suspending operations and processes,
- **Consistency:** all mechanisms and concepts of the system should always be used in the same way. This applies to buttons, icons, descriptions, menus, processes, or even how to navigate between different screens. And is achieved by making use of style guides or a common '*look & feel*' for the entire platform. A simple example of bad consistency would be, if a button to remove an item from a list appears red on one screen and grey on another. Or, if a button that performs the same operation in two or more different views has a different label in each one.
- **Recoverability:** the system should help users to recognise and recover from errors, using an understandable language and visual elements, and providing the users with mechanisms to reset an erroneous action performed by themselves.
- **Response time:** the response time required by the system to notify changes should be reasonable for the user.
- **Cognitive load reduction:** the interface should maintain a minimalist and simple design, without overloading the user with content and reducing the cognitive load. Also, one main action per screen should be maintained, being secondary actions less prominent than the primary ones.
- **Feedback:** the system should keep the user aware of the status of the platform and the actions he/she performs on it.
- **Positioning:** there are also techniques that aim to help developers to distribute the elements on the screen to highlight the most important areas or to make the reading easier. For example, by placing the most important content, actions or buttons at the top of the screen, at eye level, or by facilitating the reading from left to right.
- **Navigation:** providing a natural interaction and navigation between the views or screens that compose the platform.

3.1.2 Colour as a key cognitive aspect of the interface

Colour also plays an essential role in the users' perception of the information and help them to understand better the meaning of the different elements presented on the graphical interface. Therefore, from a number of good practices and guidelines related to colour^{[10][11]}, the following ones are applied to design the AGRICORE interface ergonomics:

- Colour should not be used as a coding medium. For example, use colour only to notify users whether or not an action has been performed correctly in the interface. Typically, notifications combine colour, along with text and icons that identify the nature of the notification (success, warning, error).
- High contrast between letters and background should be used.
- Blue-yellow and red-green combinations should be avoided.
- It is recommended to limit the number of colours from 3 to 7 (5 ± 2), in total.
- Avoid saturated colours as much as possible.

3.1.3 User-centred design

The user-centred design (UCD) is an iterative process that should include the end-user approval and validation at each stage. So, the users must be involved in the design and development stages. In the case of the design process, users can cooperate in three different ways:

1. **Identifying the user's needs and context:** user feedback helps the developer to know who the users are and for what the interface will be used.
2. **Specifying requirements and providing design solutions:** users can help to identify requirements that are beneficial to a correct design and to the project objectives.
3. **Evaluating the results:** they can also help to evaluate the design by providing possible improvements and changes that could better align to the objectives, using mockups, system sitemaps or interaction flow diagrams, among others.

Figure 10 shows an example of a user-centred design lifecycle^[12], where all the steps involving users are identified, from user identification, scenarios definition, interface interaction flow, design, mockups and prototyping, to the evaluation and approval of the results:

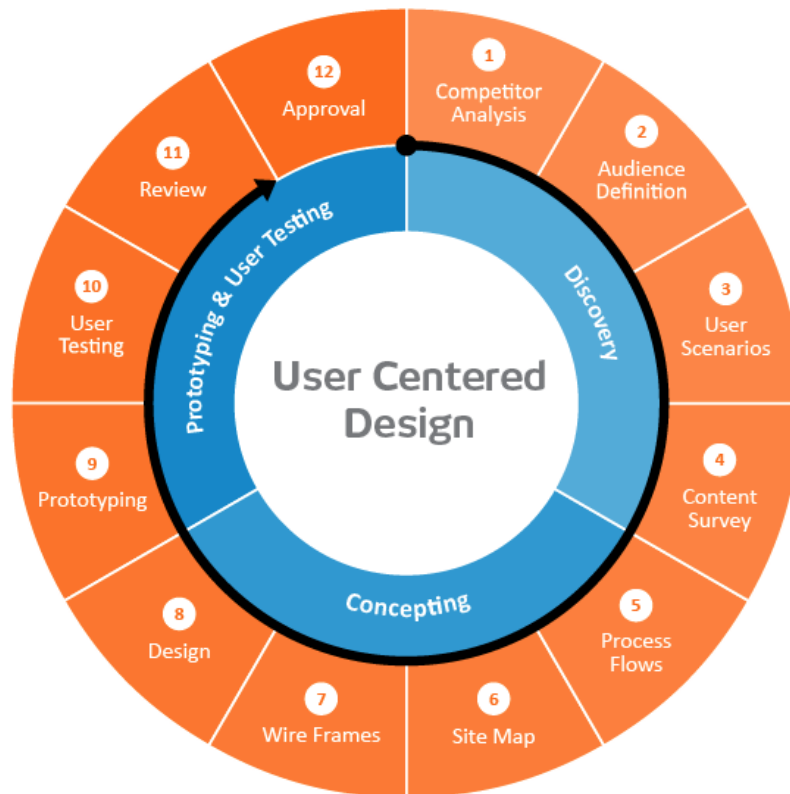


Figure 10 Interface design lifecycle

3.2 Design procedures and tools

3.2.1 Study and selection of design tools

As a previous step to the interface design, a search of the available tools for the construction of the screens that would compose the complete interface was carried out, considering the following predefined objectives for its choice:

- The tools should be free of charge, open source or a free version of them could be used without time or use limit.
- It should enable teamwork or allow several developers to work on the same project.
- The selected tool cannot be limited in the number of screens that can be created or in the amount of elements or content that can be added to each one of them.
- It should be user-friendly and should include all the necessary components and styles to make the designs as close as possible to the final product.
- It should include the possibility to export or share the designs made, to facilitate the subsequent validation process.

For this purpose, a number of tools used for mockup design were explored, compared and analysed, discarding those that did not meet the objectives described above. Although, emphasis was placed on those with no limits on the number of screens to be built or the number of elements that could be added to each one. In summary, the Table 1 present a comparison of tools analysed:

Table 1 List of mock-up tools analysed

Tool (free or trial versions tested in October 2020)	Advantages	Disadvantages
Moqups ^[13]	<ul style="list-style-type: none"> Online web application, offers templates, styles and elements with great detail and possibility for editing. One of the best solutions to make designs as close as possible to the final product. Introduces the prototyping functionality, you can simulate the user interaction in the designs. 	<ul style="list-style-type: none"> 1 project and only 1 person editing it. The project/mockups can be saved in the cloud, but you cannot download it or export it as images, for example. Has a limit of 200 objects (buttons, panels, forms, figures) per project. So it severely limits the possibility of creating projects with multiple and very detailed screens. 30 days limit. After that, you need to purchase a usage plan.
Lucidchart ^[14]	<ul style="list-style-type: none"> None that have been tested. 	<ul style="list-style-type: none"> 1 project and only 1 person editing it. Has a limit of 3 screens or mockups with 60 objects per screen. With a trial version you need to activate a premium trial introducing your credit card data, in order to design UIs.
MarvelApp ^[15]	<ul style="list-style-type: none"> Web application, allows you to make designs very similar to the final product. There is no limit on the number of screens that can be created or the number of elements that can be added to each one. Allows for the creation of teams with several members who can work on the same project. Introduces the prototyping functionality, you can simulate the user interaction in the designs. The prototype can be shared publicly via a link to MarvelApp. Very simple to use. 	<ul style="list-style-type: none"> 1 project limit. The project/mockups can be saved in the cloud, but you cannot download it or export it as images, for example. Elements, templates and editing options are a bit worse than Moqups.
Justinmind ^[16]	<ul style="list-style-type: none"> Desktop application, no time limit in the free version. There is also no limit on the number of screens that can be created or the number of elements that can be added to each one. Designs very similar to the final product. 	<ul style="list-style-type: none"> The user interface is complex and the learning curve seems high.
Pencil ^[17]	<ul style="list-style-type: none"> Desktop application, very simple UI and easy to work with. Completely free, is open source. The open source nature of the tool allows it to be extended by adding collections of components, styles and icons 	<ul style="list-style-type: none"> Being a desktop application, teamwork becomes very difficult, unless the designs are shared between developers. So, it is impossible for two or more developers to work on the same project simultaneously.

	that are not in the software, by default. <ul style="list-style-type: none"> • I also introduces the prototyping functionality. 	<ul style="list-style-type: none"> • A priori very similar to other tools like Balsamiq or MockFlow, but this one is completely free.
Balsamiq[18]	<ul style="list-style-type: none"> • Simple to use. 	<ul style="list-style-type: none"> • Poor designs, elements and templates available. • 30 days trial limit.
MockFlow[19]	<ul style="list-style-type: none"> • None that have been tested. 	<ul style="list-style-type: none"> • Poor designs, elements and templates available. Very similar to other tools like Balsamiq. • 1 project limit. • Has a limit of 3 screens/mockups.
Proto.io[20]	<ul style="list-style-type: none"> • 15 days trial version with all functionalities available. 	<ul style="list-style-type: none"> • Focused on mobile applications. • 15 days trial limit.
Mockplus[21]	<ul style="list-style-type: none"> • 15 days trial version with all functionalities available. • You can create unlimited projects during the trial version. 	<ul style="list-style-type: none"> • The project/mockups can be saved in the cloud, but you cannot download it o export it as images, for example. • 15 days trial limit.
Fluid UI[22]	<ul style="list-style-type: none"> • Lifetime trial version. 	<ul style="list-style-type: none"> • 1 project limit and only 1 person editing it. • Has a limit of 10 screens/mockups.

The final decision was between Pencil, Justinmind and MarvelApp, as they were the only ones that offered more functionalities without any time limit. Finally, it was decided to choose **MarvelApp**, because it was the only tool that complied with all the objectives, but mainly because of the collaborative work and the facilities it offered to share the designs. A more detailed description of the MarvelApp, as well as a brief overview of the prototyping process using the tool, is provided in ANNEX I: Mockup prototyping using MarvelApp.

3.3 Requirements of the AGRICORE User Interface

3.3.1 Functional requirements

Functional requirements (FR) define descriptions and behaviours of what the software/system offers, i.e. what it does or not. Including all the functionalities that the system must incorporate to meet the objectives of a project or end-users [23]. The functional requirements for the AGRICORE interface design are the following:

- **AG.D16.FR.001. Centralise the interaction of the users with the AGRICORE suite:** the AGRICORE interface module should centralise all the interaction of the users with the AGRICORE suite.
- **AG.D16.FR.001-1. User Access Management:** users must be able to login, register and recover their credentials.
 - **AG.D16.FR.001-1-1. Login:** previously registered users with a valid account should be able to access the suite by introducing their username and password. Once logged, they should be directed to the AGRICORE Suite Lobby view.
 - **AG.D16.FR.001-1-2. User Registration:** users must be able to register and request an account for the AGRICORE Suite. To that end, they should provide a name, country, valid email address, and affiliation (company, institution).

- **AG.D16.FR.001-1-3. Recover account credentials:** users must be able to recover their credentials (normally the password) if they lost them.
- **AG.D16.FR.001-2. Simulation Setup:** users must be able to configure the simulation to be run. The user can start a new simulation setup, complete it and launch it, or save an incomplete setup to resume it later. To that end, it should be possible:
 - **AG.D16.FR.001-2-1. Simulation Setup Page:** users should be shown a screen (lobby) with as many buttons as components need to be configured to allow the simulation. These buttons should have an empty background (or another alternative colour) as long as the component to which they correspond has not been defined; once the user has configured the component, the background of the button should change to filled (or another alternative colour). If the user tries to close the SPG without downloading the setup file, a message should be prompted warning about the possible loss of work. The '*Launch Simulation*' button will be disabled as long as the simulation setup has not been completed (all its components have been defined)
 - **AG.D16.FR.001-2-2. Synthetic Population Selection:** general users (basic and advanced) should have two options (buttons/tabs/another alternative):
 - One to select an existing population stored in a repository. In principle, this would be the user's private repository, stored in the space reserved for him/her in the DWH. However, it is proposed as a mere possibility the existence of a "public" repository, with a series of populations provided by their authors (public administrations, academic institutions, JRC or others) provided that the requirements of anonymisation, data protection and use of the datasets allow it. Some filters (Geographical Scope, TEO, Economic Size, etc) should be offered to discriminate among available populations, both in the private or public SP repositories.
 - Another option is to load a synthetic population directly from a continent file available on a physical medium. In this case, they should be provided with a pop-up window to the file browser to select the location of the file for uploading.
 - For advanced users, a third option should also be provided that allows them, via a button, to open a new window with the Synthetic Population Generator interface.
 - **AG.D16.FR.001-2-3. Selection of Policies to be included in the simulation:** this window should have two subsections/tabs, depending on whether you want to perform a simulation in positive (P) or normative (N) configuration. In the case of positive configuration, the screen is divided into three sections:
 - P.1: A 'General Catalogue' of Policy Descriptions (PDs), from where these descriptions, previously made public by their authors, can be imported to the user's Personal catalogue.
 - P.2: A 'Personal Catalogue' of Policy Descriptions, containing both PDs imported from the General catalogue and PDs built by the user itself.
 - P.3: A 'Selected policies' section containing the policies to be actually incorporated into the simulation setup. Once a PD has been selected, the user can edit the main parameters of the Policy by directly editing the PD file.

Once in the Personal catalogue, the user can:

- Add the policy described by a certain PD to the simulation setup.
- Rate (evaluate) a PD imported from the General catalogue.
- Delete a PD from the Personal catalogue.

Adequate filters should be provided in all sections to discriminate among PDs.

The options for the policy configuration case are yet to be defined.

- **AG.D16.FR.001-2-4. Selection of Simulation Period:** this is a very simple view/tab in principle, which should allow the user to simply define the number of years/agricultural seasons over which the simulation should run. It should be noted that the starting year must coincide with the year of the data on the basis of which the synthetic population has been constructed. This data could be extracted from the SP file and displayed in this view as well, in order to have complete information on the temporality of the simulation.
- **AG.D16.FR.001-2-5. Selection of Solver for the Simulation:** this section, as well as the previous one, is a simple screen where the users can choose the solver and the biophysical model to be used in the simulation.
 - The solver is the mathematical software (external to AGRICORE) that is invoked during the simulation to solve the optimisation problem (constrained objective function maximisation) that controls the actions of each individual agent. The combined selection of the policies to be simulated and of the synthetic population makes the resulting optimisation problem of a specific type (LP, NLP, MILP, MIQP, MINLP, etc.). Once the system detects the type of resulting optimisation problem, it should provide the user with a list of commercial solvers capable of solving that particular type of problem. At this point, the user must select one of them and provide the AGRICORE simulation engine with a path (on his own computer or in a network location) to the executable file of the selected solver. If once a solver has been selected, changes are made to the selected SP or policies combo making the resultant type of optimisation problem incompatible with the selected solver, the system will display an error message.
 - Biophysical models are simulation software elements (external to AGRICORE) whose function within the simulation environment is to allow each agent to predict the effect that control actions (agro-management decisions) (may) have on the biophysical state of their land and livestock. The selection of biophysical models is therefore directly dependent on the selected synthetic population, as these model(s) must be able to simulate the types of farming (crops and or livestock) that are relevant for the impact assessment to be carried out (those TFs present in the different synthetic agents).
- **AG.D16.FR.001-2-6. Selection of default KPIs to be computed:** this view/tab should have, for all users, a selector that allows choosing one or several from a list of pre-existing KPIs. These selected KPIs will be computed automatically during the simulation and stored by default in the results file so that they can be directly displayed in the data display window(s) as soon as the simulation is finished. For advanced users, a selector should also be provided to take them to another view/tab from where they can define, using a controlled and predefined language

and format, their own KPIs. Once created and selected, they will also be computed by default during the simulation.

- **AG.D16.FR.001-3. My Simulation Module:** users must be able to launch a simulation based on a simulation setup file. This file can be preloaded directly from the output of the simulation setup module, or it can be loaded manually by the user into the simulation module using a previously generated configuration file. This screen must have two sections:
 - S.1. Ongoing simulations must appear under the 'Simulations in progress' section, with their corresponding progress bar an percentage, and buttons to watch (live results), pause or cancel the simulation.
 - S.2. Previously completed simulations must appear listed under the 'Finished simulations' section, along with buttons to visualize the results (opening Visualisation Module), download the result data into a plaintext/rich text file, or delete result data.
- **AG.D16.FR.001-4. Visualisation Module:** users must be able to visualise the results from a finished simulation. These results can be automatically immediately displayed after the end of a simulation, or uploaded by the user using a previously obtained results file. It should be possible to download each visualisation individually as an image file or all the visualisation together as a PDF report. Advanced users must have the possibility of opening a Jupyter IDE to create their own customised visualisations by directly processing the raw result data.
- **AG.D16.FR.001-5. Synthetic Population Generator Module:** advanced users should be able to generate their own Synthetic Populations using the interface and linking with ARDIT, and with the Data Extraction, Data Fusion, Bayesian Network Generation and SPG Modules.
 - **AG.D16.FR.001-5-1. Synthetic Population Generator Configurator:** advanced users landing on this window of the synthetic population generator should be able to configure at least the following parameters of the synthetic population to be generated:
 - Geographic Scope of the farms to be synthesized.
 - Technical-economic orientation(s) of the farms to be synthesized.
 - Economic Dimension Range of the farms to be synthesized.
 - Base year of the simulation, which will define the data to be used in the SPG process.
 - (Desirable) other parameters (e.g. gender of owners, etc.).

It should also be possible to configure how the remaining "non-interest" farms are modelled. That is, if a super agent is created to group them all together, or by TEOs, etc. Once all the parameters have been selected, the user should be able to launch a search for available data sources. This search is powered through ARDIT, and the result is a list of potential datasets to be used for each one of the Attributes of Interest of the agents.
 - **AG.D16.FR.001-5-2. Selection of Datasources for SPG:** the result of the (multi)query to ARDIT should be a view/tab listing all the Attributes of Interest of each agent (according to the WP3 ABM definition). When selecting each attribute, the possible data source(s) available to extract the information to assign values to that attribute should be displayed. Ideally, there should be a mechanism (colours, ticks, others) to visualize for which attributes a data source has already been selected, and for which ones the selection is still to be made. Once at least

one data source has been selected for each AoI, a selector/button should be enabled to trigger the generation of the SPG itself.

- **AG.D16.FR.001-5-3. Synthetic Population Generator (Down)Loader:** once the synthetic population generation process is completed, this view/tab should allow the advanced user:
 - Download the generated population to his/her computer as a file with format and extension yet to be defined.
 - Load the generated population in the Simulation Setup module and be redirected to the simulation setup lobby.
- **AG.D16.FR.001-6. AGRICORE Suite Main Page:** Once logged, users must be able to directly navigate to any of the interface modules (simulation setup, synthetic population generator, simulation run, visualisation of results and (optionally) commuting to/from ARDIT). The user should also be able to access his/her Catalogue of Policies and their past executed simulations (under My Simulations).

3.3.2 Non-functional requirements

Non-functional requirements (NFR) define how the system should be or how it should perform through constraints or quality objectives that it must satisfy^[24]. The non-functional requirements for the AGRICORE interface design are the following:

- **AG.D16.NFR.001. Developed as a cross-platform desktop application web technologies:** The AGRICORE interface module will be implemented as a cross-platform desktop application (available for Windows, macOS and Linux systems) using web technologies.
- **AG.D16.NFR.002. Warning pop-ups:** On any user interaction that involves closing, stopping or cancelling a process that may cause the user to lose information or work previously done, the system will display a pop-up warning.
- **AG.D16.NFR.003. Help popovers:** The interface will provide users with help buttons, which, when clicked, display information about sections, functionalities or actions to be performed.
- **AG.D16.NFR.004. Help section:** The interface will provide a section with documentation or guidelines for using the platform.
- **AG.D16.NFR.005. Open-sourcing:** Following the open-source paradigm of the AGRICORE Project, the platform and its code will be publicly available for download to any interested user.
- **AG.D16.NFR.006. Shared user accounts:** The entire AGRICORE suite will share the same user accounts. The user accounts for accessing the AGRICORE interface will be the same as those that can be used to access ARDIT or other project tools.
- **AG.D16.NFR.007. Network connection error:** The platform will notify the user with an error message if any connection to other tools or modules of the project do not return data due to connectivity, availability or operational problems.

4 AGRICORE Graphical User Interface

This section describes the mockup design done using MarvelApp, based on the requirements established for the interface, the objectives of the project and the design and usability principles applied. Firstly, the operational flow designed on the basis of the established requirements is explained, along with notions of desktop application design and general concepts of the design carried out. Finally, the analysis of each designed screen and its usefulness within the platform is described.

4.1 Main concepts of the GUI design

4.1.1 AGRICORE desktop application

The AGRICORE interface will be implemented as a cross-platform **desktop application** using web technologies, for this reason, the proposed design takes the general characteristics of a desktop application. The designed mockups simulate the execution of the platform on a Windows operating system, as can be seen in the upper bar of the programme window. The aesthetics of this bar is something that is automatically provided by the framework used to develop the platform, so it is not something that the developer has to worry about. Figure 11 shows an example of the template used by all the mockups designed, where it can identify the top bar mentioned above and a **collapsible sidebar**, a very common element in dashboards and desktop applications, which has links to each of the sections, serving as a support element to navigate from any section of the platform from any other. The content of the screen would always follow the same order, finding first a header to identify the section in which the user is, a small descriptive text of it and finally, the content of that section which, could include other less highlighted headers and descriptive texts.

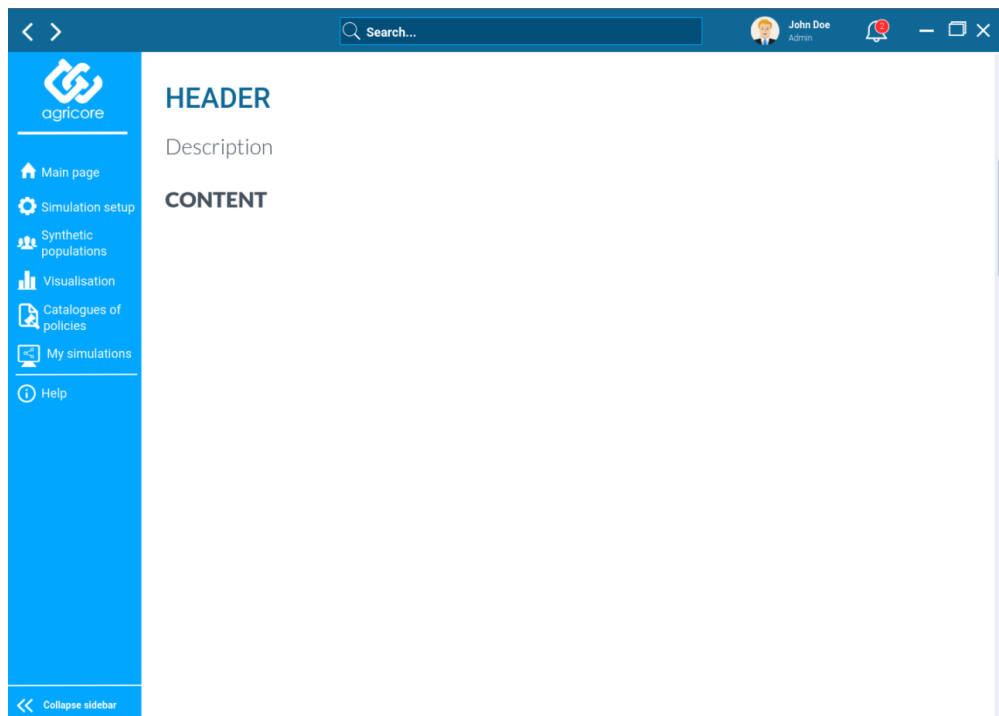


Figure 11 AGRICORE interface template

In Figure 12 a more detailed view of the platform's top bar can be seen. From left to right, it can be found:

- Buttons to go back or forward in the history of visited pages.
- A search box to search content in the platform.
- A button with the user's profile picture (element added provisionally, it may not be in the final development), name and role. Clicking this button would open a drop-down menu showing options such as the profile configuration or logout.
- Another button represented by a bell icon, to alert the user of new notifications. For example, to let the user know that a simulation process launched by him/her, has been completed.
- Windows operating system buttons to minimise or close the application window.



Figure 12 AGRICORE interface top bar

4.1.2 AGRICORE GUI colour palette

The colour range traditionally associated with the rural and agricultural world is that of greenish and toasted yellows, which evoke the colour of forests and cultivated fields. However, AGRICORE is a tool that links the rural environment with Community Institutions, and how the latter's political work influences the improvement of socio-environmental conditions in the former. It is therefore a tool whose object of simulation is the agricultural rural environment, but which incorporates other aspects such as the description of policies at the European level and the analysis of their socio-economic influence. In addition, ARDIT, one of the tools contained in the AGRICORE suite, already uses a yellow-green colour range in its design. For all these reasons, and in order to highlight the techno-economic nature of AGRICORE as a means of simulating European public policies, a palette of blue-blue colours, traditionally linked to the European identity and to a more administrative and financial context, has been initially chosen (Figure 13). In any case, this choice is not necessarily definitive and may be changed later if the Commission, through the agencies that will presumably govern the tool in the future (e.g. DG-AGRI), considers another corporate visual image to be appropriate.

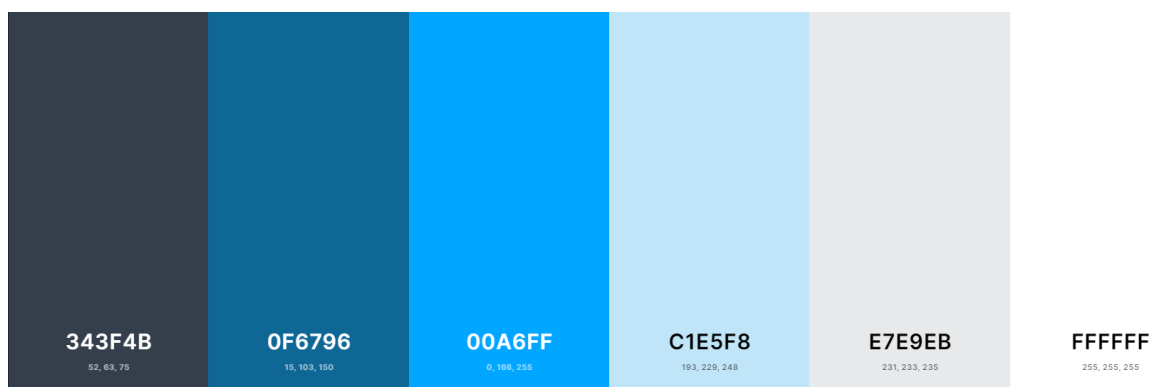


Figure 13 AGRICORE interface colour palette

Similarly, a colour palette was also chosen to differentiate the AGRICORE interface from the support tool for the creation of synthetic populations. In this case, using three shades of grey which, like the AGRICORE interface, are not necessarily the definitive shades for the tool. They were simply chosen to differentiate them (Figure 14).

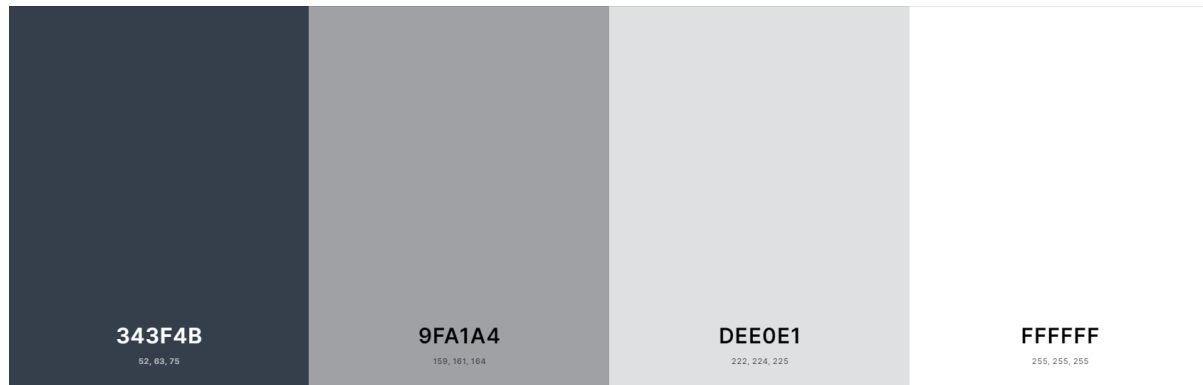


Figure 14 AGRICORE synthetic population generator tool colour palette

4.1.3 Files interacting with the AGRICORE GUI

Besides providing the user with the graphic elements to work on the interface, the platform includes the possibility of using files in which the configuration or data generated in the different work processes included in the platform are stored. In general, there are three types of files:

- **Simulation configuration files:** can be downloaded by the user during the simulation setup process, containing the data introduced to configure the synthetic population, policies, solvers or KPIs. They can be used to save an unfinished configuration, enabling the user to re-upload the file and resume it.
- **Simulation results files:** these files can be obtained once the simulation is finished and store all the results of the simulation process. Then, they can be loaded from the visualisation section, in order to display the results in a graphical way.
- **Synthetic population files:** these files, generated by the synthetic population generator support tool, store all the configuration, parameters and data of the population. They can be uploaded by users in the simulation setup section to load a synthetic population into it.

4.1.4 Tables and icons

All the tables displayed on the platform follow a similar style. The headers contain the different data columns, with buttons on the right side to sort the listed data alphabetically or numerically. Some of the designed tables contain a second line in the header with inputs to filter the tables at the column level. On the right side of the table, there is always a column of options for each row, where there is a button to clear all search filters applied so far, represented by a brush (Figure 15).


Name	Policy maker	Modeled by	State	Start Date	Derogation	Type	Options
<input type="text" value="Search by name"/>	<input type="text" value="Search by author"/>	<input type="text" value="Search by date"/>	<input type="text" value="Search by state"/>	<input type="text" value="Search by start date"/>	<input type="text" value="Search by date"/>	<input type="text" value="Search by type"/>	

Figure 15 Detail of a table on the platform

4.2 AGRICORE Graphical User Interface: user navigation and screen design

4.2.1 User workflow

Once logged in, the user will navigate to the main page where links to each of the sections or modules that compose the platform can be found. The following diagram (Figure 16) reflects the operational flow with the different screens/sections that compose the platform (blue). In it, we can also identify the sections restricted to a limited number of users (red with padlock), such as the external application to create new synthetic populations or the platform's general policy catalogue.

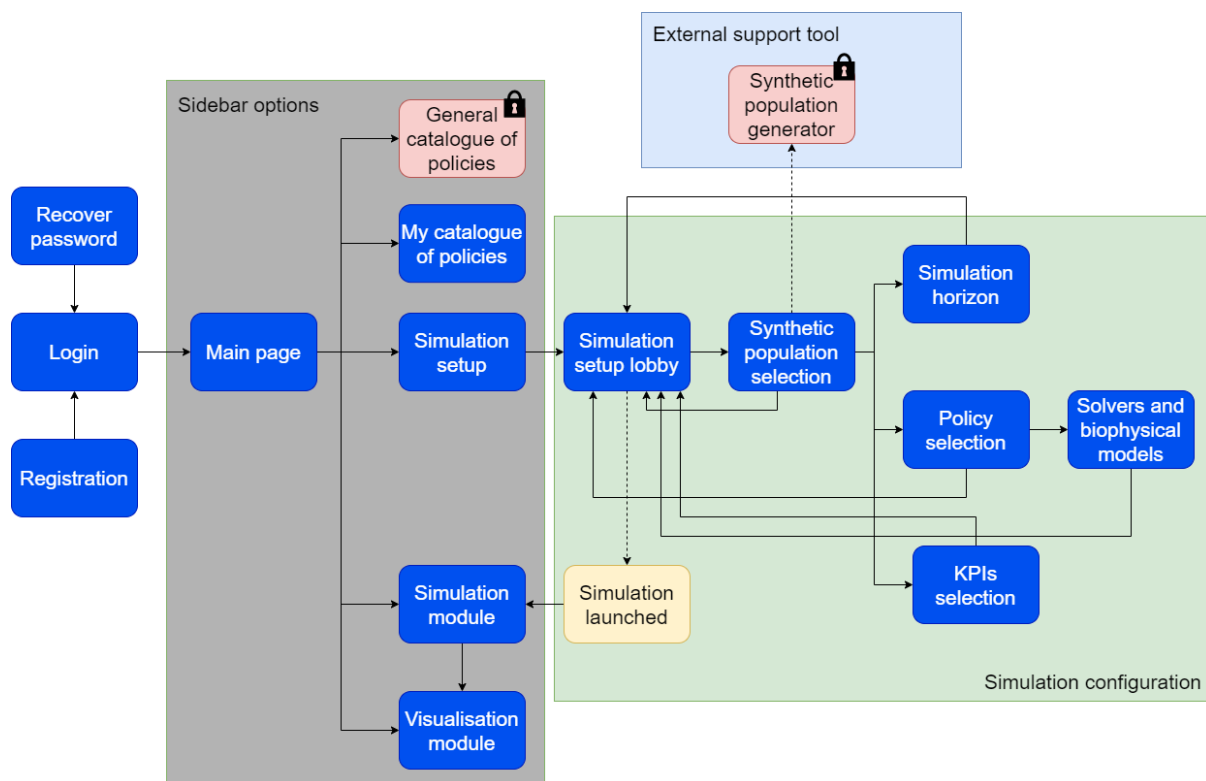


Figure 16 AGRICORE interface operation flow

From the main page, the user can access the different sections of the tool:

- **Simulation setup (implemented by the simulation setup module)** will allow the user to configure and launch new simulations on the platform based on different parameters and components. In turn, this module will provide the user access to the following actions (each one linked to its corresponding screen):
 - **Synthetic population selection:** for the selection of the population of agents on which the simulation is to be applied.
 - **Policy selection:** for the selection of the agricultural policies to be evaluated.
 - **Simulation horizon (timeframe selection):** to simply define the number of years/agricultural seasons over which the simulation should run.

- **Solvers and biophysical models selection:** to let the user select the solvers and biophysical models to be applied in the simulations.
- **KPI selection:** selection of KPIs that will be computed automatically during the simulation.
- **Simulation module:** used to monitor the execution of simulations already launched by the user, access a history of completed simulations or launch a completely new simulation based on a configuration file.
- **Visualization module:** used to graphically present the results obtained from the simulations in graphs, charts or plots.
- **Synthetic population generator module:** used to create synthetic populations to feed the simulations. This module has its own dedicated application, separated from the AGRICORE interface, due to its complexity and its advanced-user purpose.
- **Policy catalogues module:** the AGRICORE platform will provide two different catalogues for the selection of agricultural policies during the configuration of the simulation. In this section, divided into two parts, users will be able to access their personal policy catalogue to manage it, facilitating the policy selection process during the simulation setup. There will also be a general and public policy catalogue, managed only by more advanced users, which will have a more extensive and complete set of policies. The rest of the users will be able to use the general catalogue to populate their personal catalogues.

4.2.2 User access management

A user account is required to utilise the AGRICORE tool. So, it will be necessary to log in by introducing the user credentials (email and password). If the user does not have an AGRICORE account, he/she will have to fill in a registration form to create a new one, providing the name and surname, country, valid email address and affiliation (company or institution). In addition, it will also be possible for users to recover their password if they forget it.

The registration, login and password recovery screens have been designed using some common practices nowadays when designing desktop applications of this type, using a smaller window type to display the forms and not giving the user access to the rest of the interface until he/she has validated it.

Figure 17 shows a screenshot of the mockup designed for the login screen of the platform:

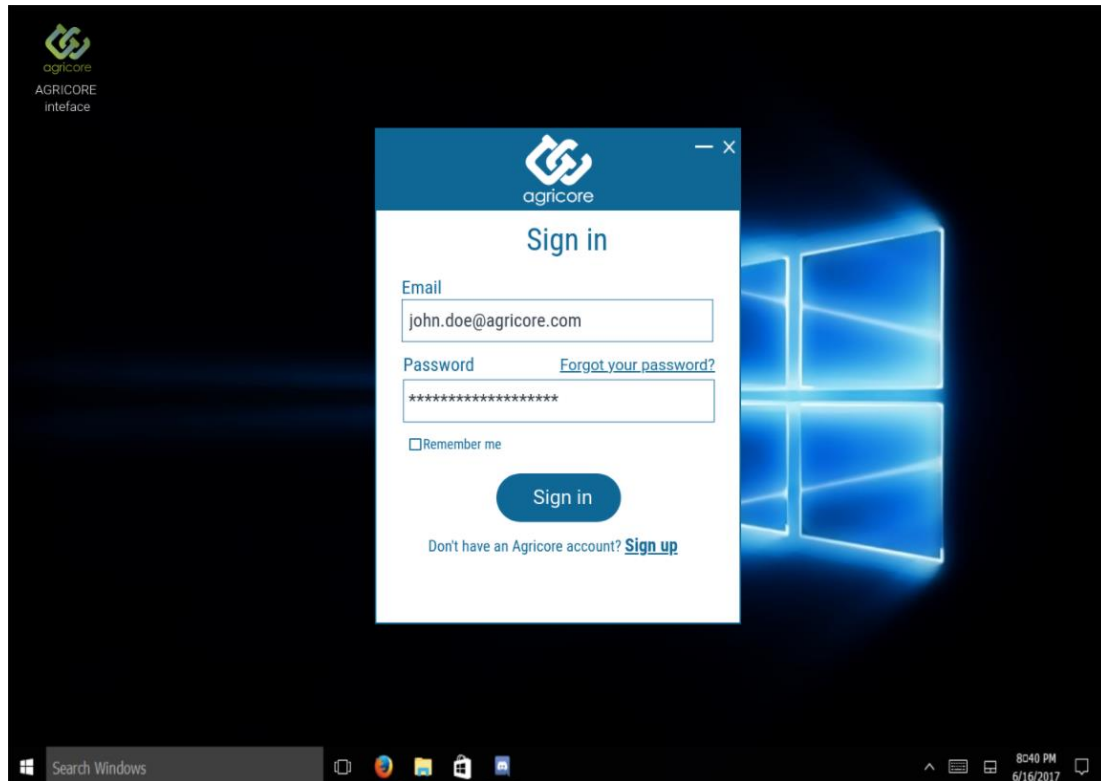


Figure 17 Login screen

In turn, Figure 18 shows the mockup designed for the registration screen of the platform:

Figure 18 Registration screen

4.2.3 AGRICORE platform main page

Once logged, users will be redirected to the main page. On this screen, the idea is to provide the users with links and descriptions to the different sections and modules that compose the platform. The design shows a list of available options, allowing users to decide between starting a new simulation process, initiating the creation of a new synthetic population through the Synthetic Population Generator external tool, or navigating to the visualisation and simulation modules. Figure 19 shows the design of the AGRICORE platform lobby:

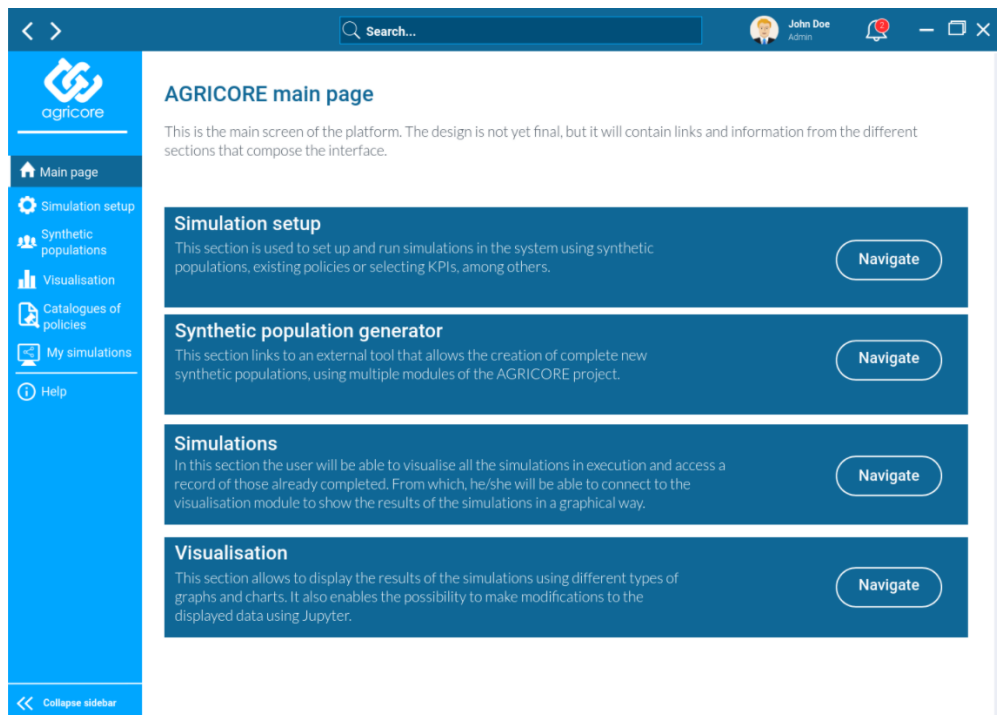


Figure 19 Main page screen

4.2.4 Simulation setup GUI sub-module

In this section, users will be able to configure and launch simulations on the platform. The simulation setup section comprises the lobby to link to each of the configurable components of the simulation and the dedicated sections for each component itself, being the following:

- Synthetic population setup.
- Policies setup.
- Simulation horizon setup.
- Solvers and biophysical models' setup.
- KPIs setup.

4.2.4.1 Simulation setup lobby

Initially, it was thought that any component of the simulation could be configured independently of the others, so a circular layout was designed with the buttons to access the different configurable components. However, in order to configure the policies or the simulation horizon, it is necessary to first establish the synthetic population on which run the simulation. The same issue was happening between the policies and the solvers, the user needs to set up the policies first and then the solvers. For this reason, the design was changed to a list of buttons, shown in

the Figure 20. Then, if the user tries to access a section that requires others to be previously configured, the system will display an alert message.

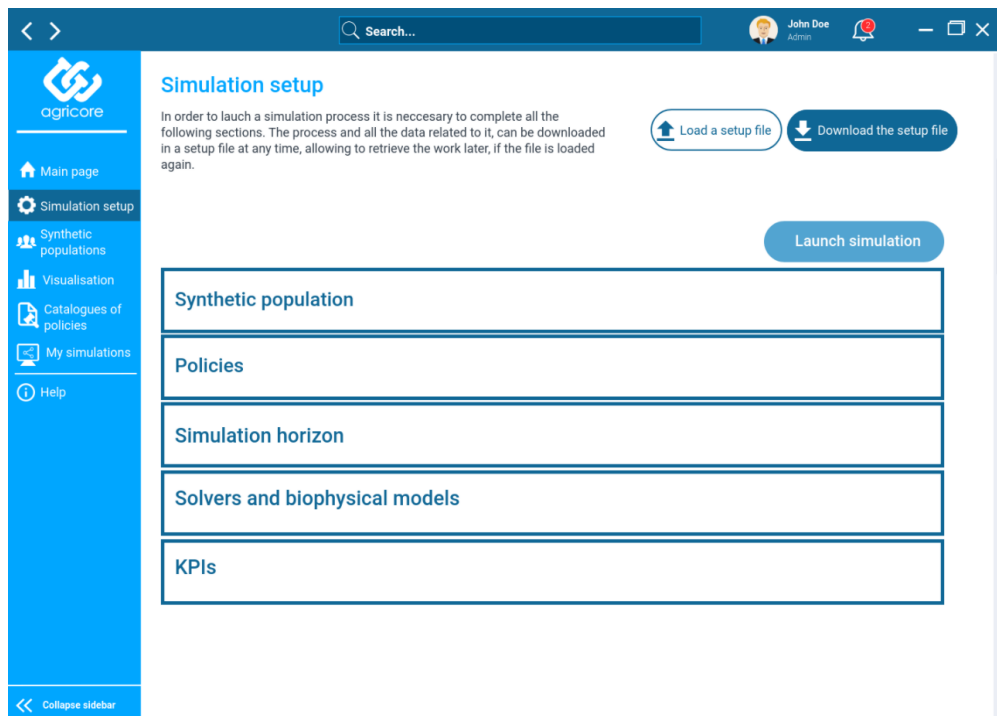


Figure 20 Simulation setup lobby

These buttons have a white background as long as the component to which they correspond has not been configured. Once the user has configured one component, the background of the button changes to blue and a completion text is displayed. On the other hand, the button to run the simulation, located above the list of buttons, will be disabled until all sections are configured or completed (Figure 21).

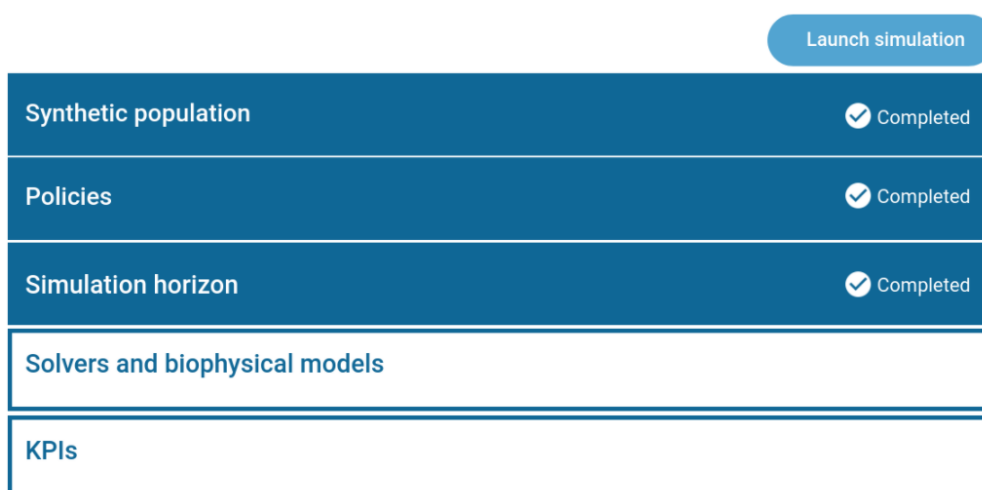


Figure 21 Simulation setup lobby

At any time, users can download a file with the configuration done so far, which can be uploaded again to finish it, thus avoiding the user to lose his work. This functionality is reflected by two buttons at the top of the screen, which opens a pop-up or alert window where users can select the file from their computers. In addition, if the users try to navigate to any other different screen(outside of the simulation setup process), the system will notify them if they want to download the configuration file to avoid losing the progress made during a configuration process. Similarly, if the user attempts to upload a configuration file and there are already some completed components on the screen, the system will notify the user if he/she wishes to replace the entire configuration with the one contained in the file.

Within each of the sections that compose the simulation configuration, there are two buttons at the bottom of the screen. The first one, to go back to the configuration lobby and the second one, to confirm the configuration made in that section. The latter allows us to return to the configuration lobby, but shows the section as already configured or completed.

4.2.4.2 Synthetic population selection

In the first configurable section of the simulation setup, users can select the farm population on which to run the simulation. Here, the users have three options:

- **Select an existing synthetic population from the repository:** the user can search for existing synthetic populations stored in a repository of the project DWH, using some filters and parameters. This screen, which is displayed as a pop-up or alert window, is composed of three sections, one for the search filters, with different inputs based on the attributes of the populations (title, geographical scope, economic dimension, etc.), a second section to display the results found in a table, where users will have to select one of the populations through a checkbox, and finally, the third section with two buttons, one to close and cancel the process, and another one to confirm the selection of the population, closing the pop-up. In addition, the synthetic populations may be public (populations provided and released by advanced users) or private (populations stored in a space reserved for the user in the DWH). The screen for filtering the populations is shown in Figure 22.
- **Load a population from an external file:** the user will be able to load a file containing the synthetic population coming from the synthetic population generator support tool.
- **Create a new population:** advanced users will be redirected to the synthetic population generator tool.

The screen for selecting one of the three options, is shown in Figure 23.

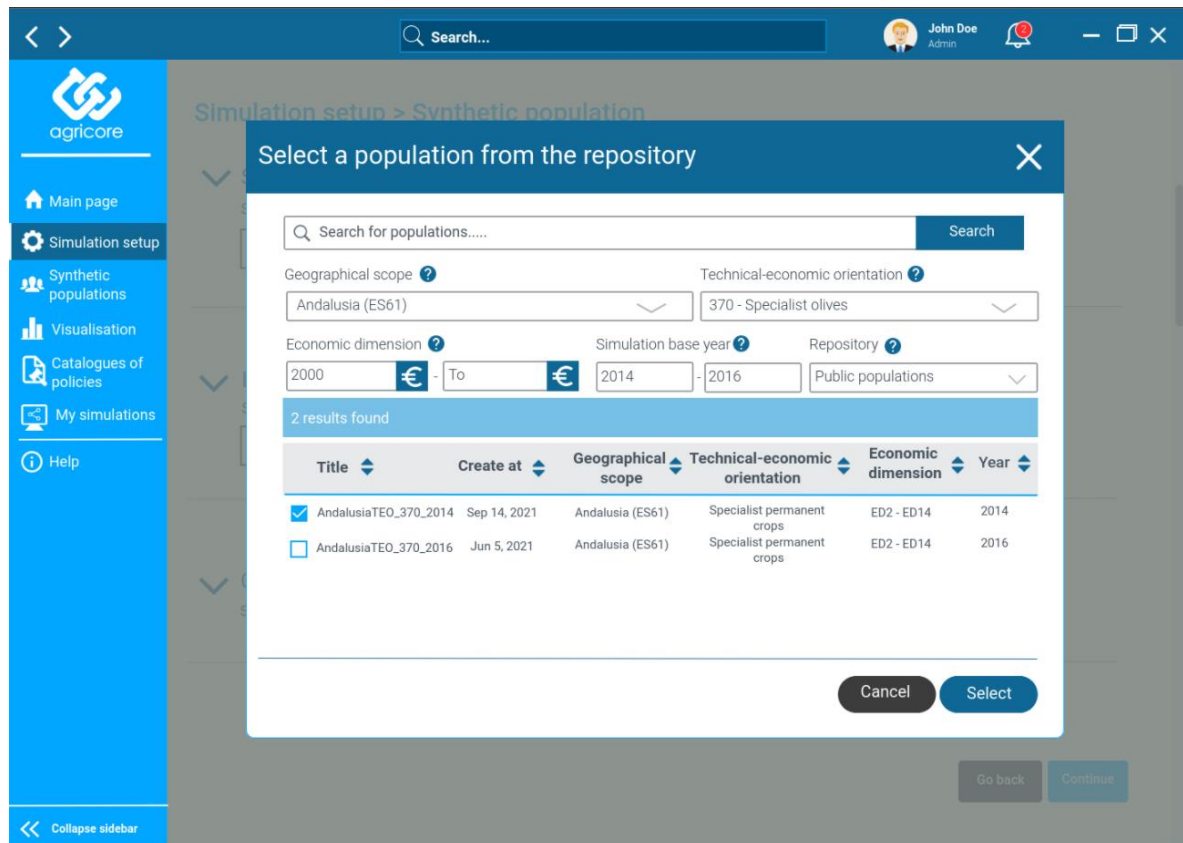


Figure 22 Screen for searching existing populations

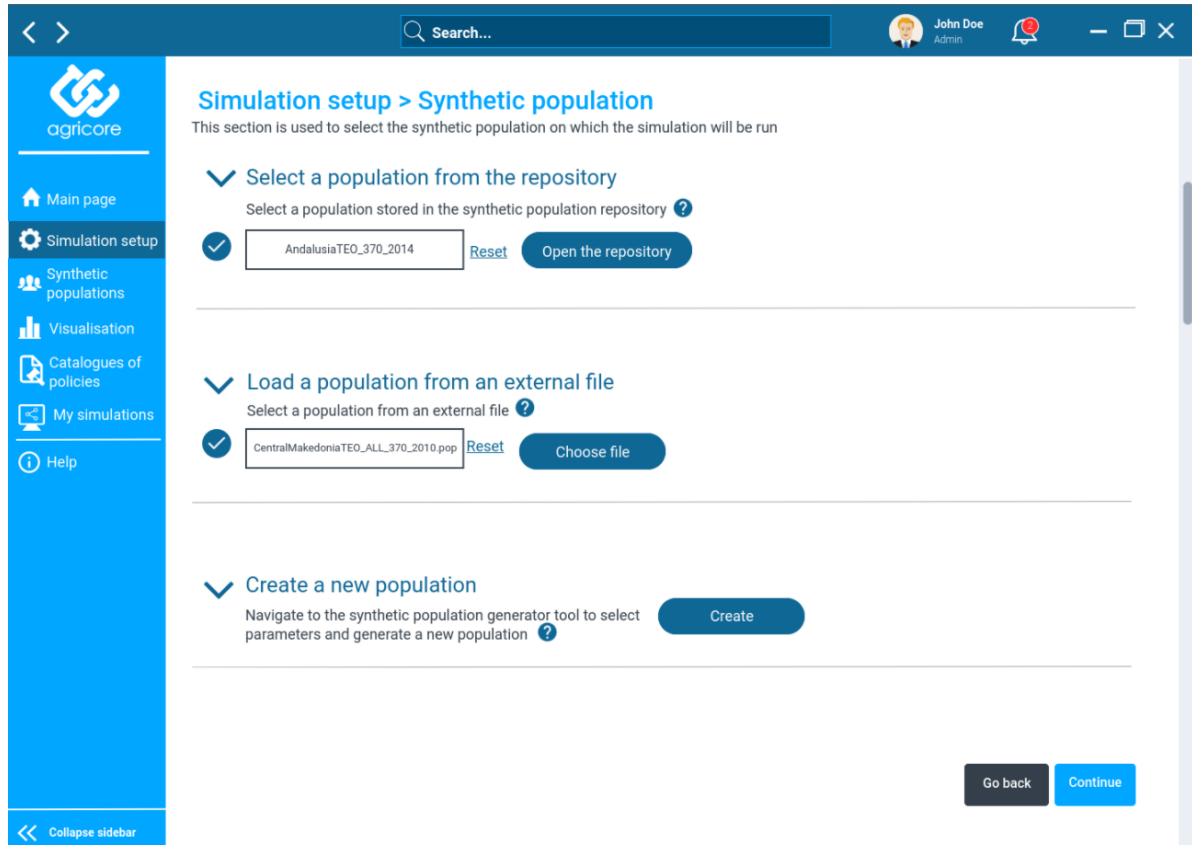


Figure 23 Screen for the synthetic population selection

4.2.4.3 Policies selection

On this screen, users can select the set of agricultural policies to run the simulation on. AGRICORE platform provides two different types of policy configurations, **Positive** (monitoring, ex-post) or **Normative** (policy design, ex-ante). The positive policy configuration run simulations using existing policies, whereas in the normative one, the users must choose between some parameters/constraints to obtain the better possible policy given that parameters/constraints. The first thing users will do when accessing this section, is to select which type of configuration they want to fill in. For this purpose, the interface has a selector with two options, one for each type of configuration. By clicking on each of them, the content of the screen changes completely. At the time of drafting this document, the screen for the normative tab configuration has not been designed yet.

In the positive configuration the screen is divided into two sections. At the top are the policy catalogues, personal (of the user) and general (of the platform). At the bottom of the screen there is a table with the list of policies selected for the simulation ("**Selected policies**"). By default, when accessing this screen, this table will be empty. The user can fill it in in two ways:

- Creating new policies using the button "**Create a new policy**", which will open a pop-up screen with a form to fill in with the definition of the policy to be created.
- Adding existing policies from the user's personal catalogue, by clicking on the **plus icon** in this catalogue. This button will open another pop-up displaying the selected policy definition details and asking the user for a confirmation to add the policy to the selected policies table.

In addition, another possibility would be to add policies from the personal catalogue and, once in the selected policies table, edit their configuration by clicking on the **pencil icon**. Normal users will be able to consult this catalogue and filter the contained policies using different parameters and attributes, and then, if they wish, they will be able to add policies from the general catalogue to their personal catalogues. The user would use the general catalogue of policies to search and add desired policies to his own catalogue, but never directly to the selected policies table.

Figure 24 shows a complete screenshot of the policy selection screen for a positive configuration.

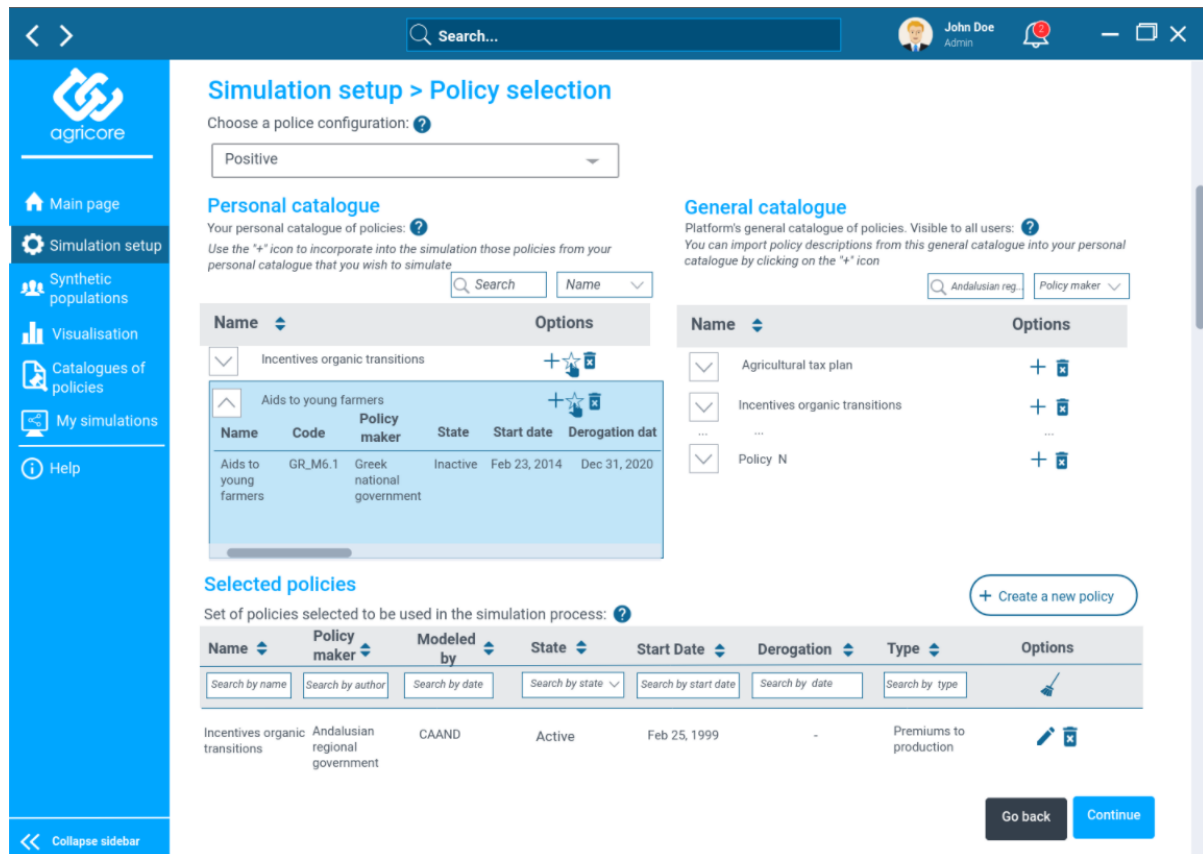


Figure 24 Policy selection screen

Although several layouts have been designed for this screen, the preferred layout, presents both catalogues opened and side by side. In order to make both visible on the screen and at the same height, the selected layout displays a shortened version of the policy tables, showing only the name and the available options. To display the full policy definition, users will have to click on the dropdown button on the left of each policy; this action will open a menu (in blue) with all the attributes of the policy, as can be seen in Figure 25.

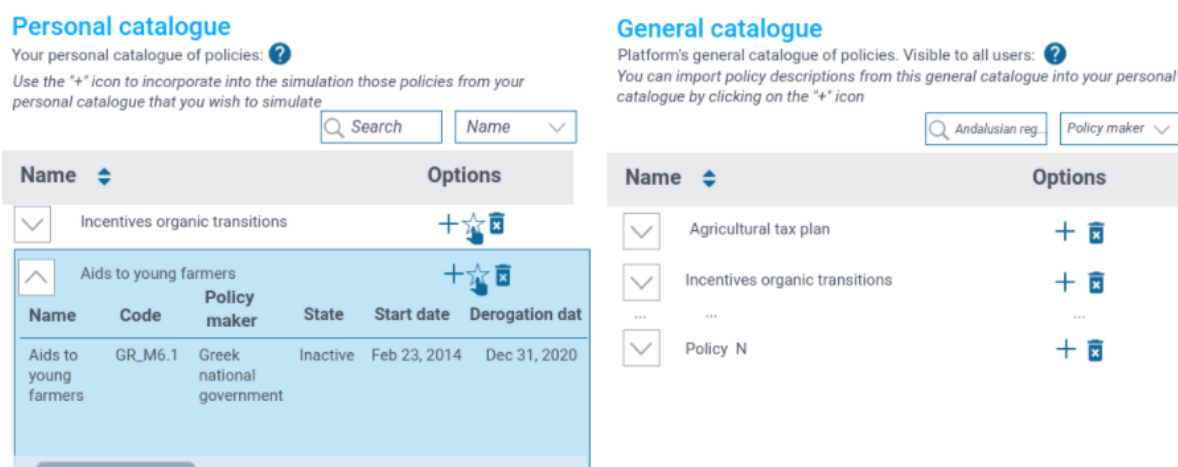


Figure 25 Catalogues tables design

In addition, the AGRICORE platform will enable users to rate policy definitions made by other users. Within the personal catalogue, the user will find a button represented by a **star icon**, that will open a new pop-up screen with the complete policy definition and an input to enter a numerical rating of the policy. Figure 26 shows a screenshot of the pop-up to rate policies.

Rate policy definitions

Rate ? * Specify a number between 0 and 10

Policy name ? Aids to young farmers

Type ? Direct subsidies

Derogation date ? Dec 31, 2020

Start date ? Feb 23, 2014

Active/Inactive ? Inactive

Country/region ? Greece

XML file ?

```
<instrument no="1" ref="GR_M6.1" type="Direct subsidies">
  <Requisites_of_Beneficiary_Holdings>
    <Minimum_EconomicSize> 0 </Minimum_EconomicSize>
    <Maximum_EconomicSize> 50000 </Maximum_EconomicSize>
  </Requisites_of_Beneficiary_Holdings>
</instrument>
```

Cancel Rate

Figure 26 Pop-up to rate policy definitions

In Figure 27, some interface elements related to policy rating are shown, such as, the number of ratings for a policy definition, the average rating obtained or the icon to open the rating pop-up.

Average rating	Number of ratings	Options
Search by rating	Search by n° ratings	
7	13	
8	57	

Figure 27 Policy rating elements

In short, on the policy selection screen, the tables displayed will have the following options:

- General catalogue of policies table:
 - Plus icon: adds the specific policy to the user's personal catalogue.
 - Bin icon: deletes the specific policy from the general catalogue (only available for advanced users).
- Personal catalogue of policies table:
 - Plus icon: opens a pop-up with the detailed information of the specific policy in order to add it to the selected policies table.
 - Star icon: opens a pop-up with the detailed information of the specific policy to rate it.
 - Bin icon: deletes the specific policy from the user's personal catalogue.
- Selected policies table:
 - Pencil icon: opens a pop-up to edit the specific policy definition.
 - Bin icon: deletes the specific policy from the set of policies selected for the simulation.
 - Create a new policy button: located above the table, opens a pop-up to create a completely new policy and add it to the selected policies list.

The pop-up mockups for the functionalities to create, edit, rate, or view the definition of a policy have all a very similar aspect. In the upper part of the screen, can be found the attributes of the policy and in the lower part, the complete definition of the policy in a XML or JSON format. At the time of drafting this document, it is still not yet definite how we are going to represent the policies virtually, because there are no standards that allow us to do it, it was thought to use XML or JSON files to store the policy definitions. The user will be able to modify the definition using the form inputs or by editing the JSON/XML file directly. Both options cannot be used simultaneously, i.e., if the user edits the XML, the form will be disabled and vice versa. In Figure 28, an example of the pop-up to edit a policy definition can be seen:

Figure 28 Pop-up to edit policy definitions

Finally, the user's personal catalogue and the general catalogue have two dedicated and separate sections from the simulation processes, accessible through the sidebar, where the users will be able to manage them at any time.

4.2.4.4 Simulation horizon

This section allows the users to simply define the number of years/agricultural seasons over which the simulation should run. The starting year, "**Base year**", must coincide with the year in which the synthetic population has been constructed. Then, the user will have to select whether the simulation will be based on years, or on agricultural seasons. By finally, select the number of time units to simulate. An example configuration for this section would be the following:

- **Base year:** "2004" (extracted from the selected synthetic population base year).
- **Time unit:** open the drop-down menu and select the option "years".
- **Number of time units to simulate:** by entering the number "13", it would indicate that the user wants to simulate from the base year, 2004, to 2017.

The mockup designed for this screen can be seen in Figure 29.

Simulation setup > Simulation horizon

Simulation horizon

In this section the user can define the number of years/agricultural seasons over which the simulation should run. It should be noted that the starting year (base year) must coincide with the year of the data on the basis of which the synthetic population has been constructed

Base year [?]

Time unit (year, season) [?]

Number of time units to simulate [?]

[Go back](#) [Continue](#)

Figure 29 Simulation horizon screen

4.2.4.5 Solvers and biophysical models

This section, as well as the previous one, is a simple screen where the users can choose the solver and the biophysical model to be used in the simulation.

The solver is the mathematical software (external to AGRICORE) that is invoked during the simulation to solve the optimisation problem (constrained objective function maximisation) that controls the actions of each individual agent. Normally, the different commercially available solvers are able to tackle more than one type of optimisation problem (LP, NLP, MILP, MIQP, MINLP, etc.), but they are also normally specialised only in some of them. The combined selection of the policies to be simulated and of the synthetic population (which may involve an additional set of policies "native" to that population) implies that the agents' objective function and/or the constraints to which they are subjected adopt a specific mathematical structure, making the resulting optimisation problem of a specific type. Once the system detects the type of resulting optimisation problem, it should provide the user with a list of commercial solvers capable of solving that particular type of problem. At this point, the user must select one of them and provide the AGRICORE simulation engine with a path (on his own computer or in a network location) to the executable file of the selected solver. If once a solver has been selected, changes are made to the selected SP or policies combo making the resultant type of optimisation problem incompatible with the selected solver, the system will display an error message.

Biophysical models are simulation software elements (external to AGRICORE) whose function within the simulation environment is twofold. On the one hand, they allow each agent to predict the effect that their future control actions (agro-management decisions) may have on the biophysical state of their land and livestock, mimicking the agronomic knowledge of AH managers that governs the decisions they make. On the other hand, it allows to simulate the real effect of

the actions taken, once combined with the biophysical disturbances (weather conditions, pests, animal diseases, etc.) external to the farm, and thus to update the agronomic state of the farm after each simulation step. The selection of biophysical models is therefore directly dependent on the selected synthetic population, as these model(s) must be able to simulate the types of farming (crops and or livestock) that are relevant for the impact assessment to be carried out (those present in the different synthetic agents).

The screen for selecting the solver and the biophysical model is seen in Figure 30.

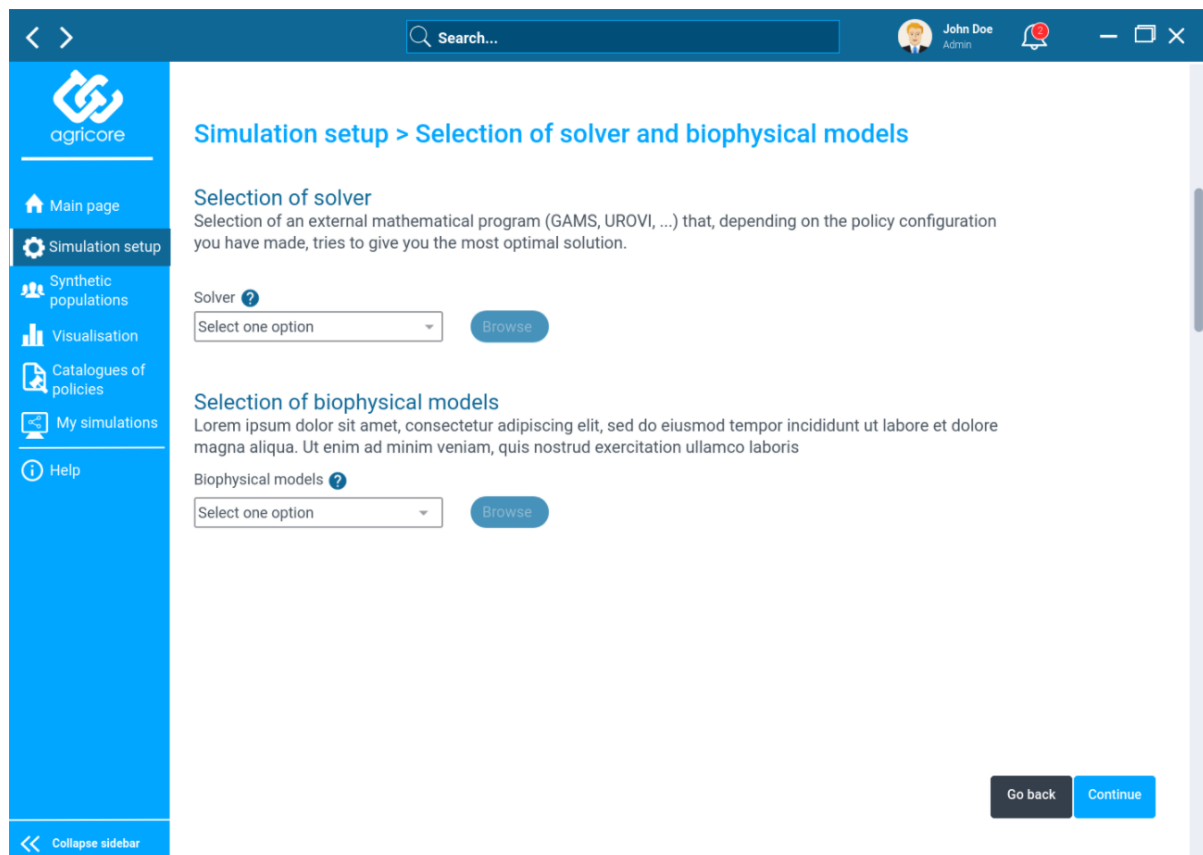


Figure 30 Solvers and biophysical models screen

4.2.4.6 KPIs selection

In the last section corresponding to the simulation configuration, the users will be able to select a set of pre-existing KPIs from a list. These selected KPIs will be computed automatically during the simulation and stored by default in the results so that they can be directly displayed in the visualisation screen as soon as the simulation is finished. The design of this screen is very similar to the one for configuring policies. First, we find a table with all the existing KPIs in the platform and then, a table where the user must add the KPIs that will be computed, by default, in the simulation. The user can add KPIs to the selected ones by simply clicking on the "plus" button. Some functionalities associated with this screen are still under analysis at the time of writing this document, such as the possibility for users to create or edit their own KPIs. Similarly, although no mockup has been designed for this, there will be a screen where advanced users can define the list of KPIs available on the platform.

In the tables included in the design, there are icons to delete the KPIs from them. In the case of the KPIs selected for the simulation, this functionality is clear, remove them from the selected

table. However, the button to remove them from the table of available KPIs, would only be available for advanced users who can manage this list in the platform.

Figure 31 shows the mockup designed for the selection of KPIs.

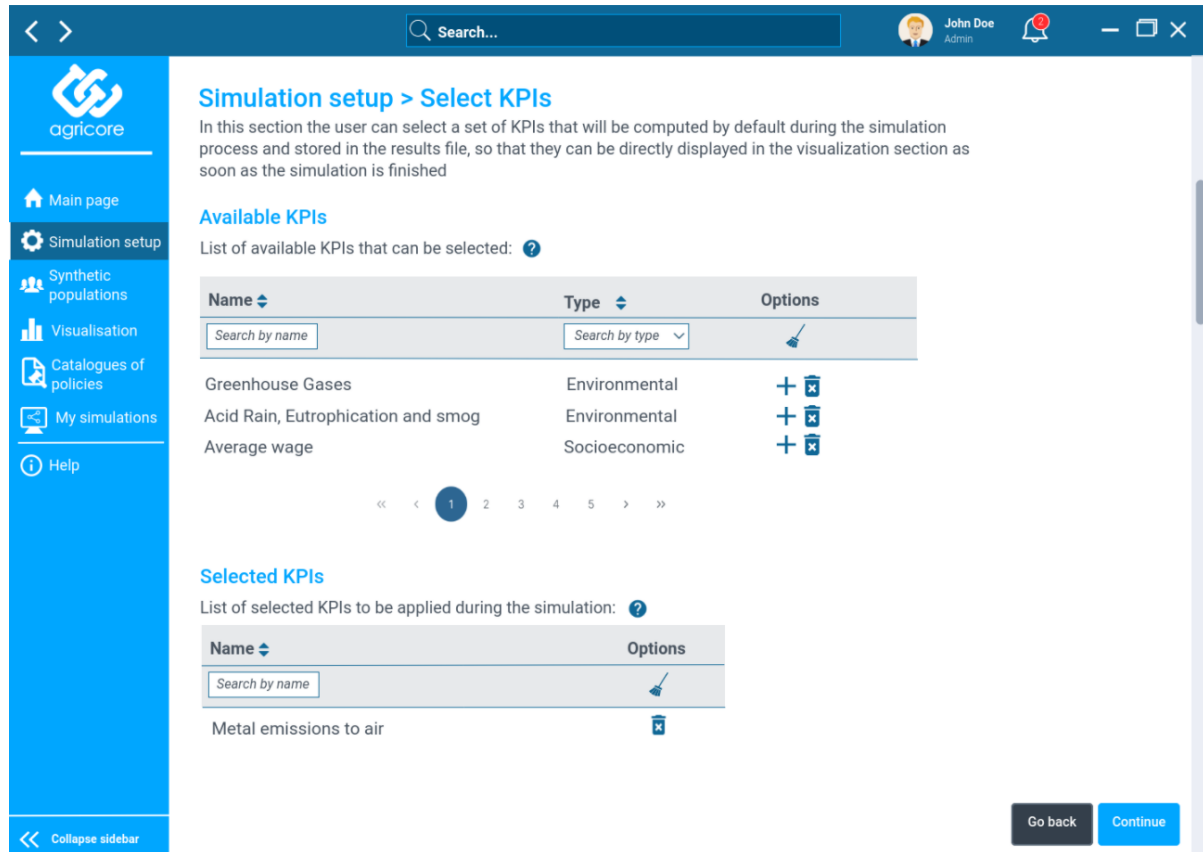


Figure 31 KPIs selection screen

4.2.5 Synthetic population generator GUI sub-module

The process of generating a new synthetic population may include extensive and fully manual data processing actions not based on the use of the graphical interface, which in practice limits the use of this sub-module to advanced users. For this reason, the synthetic population generator has been designed as a completely different application, with its own interface, separated from the AGRICORE interface, to which only more advanced users will have access due to its complexity and its advanced-user purpose, which involves **ARDIT**, the **Data Extraction**, the **Data Fusion** and the **Synthetic Population modules** of the project. The design of this screen has been divided into two parts, firstly the attributes or parameters that define the population, including the following:

- **Geographical scope:** the user can select the geographical region(s) in which the real holding population to be synthesised is located. The different regions (up to NUTS3) can be added individually or in aggregate (all NUTS2 of a NUTS1).
- **Technical-economic orientation:** the user can restrict the synthesised population to include only those real agricultural holdings whose main type(s) of farming are of interest for the simulation. The 62 different particular farm types can be added individually or aggregated by main farm type (TF14).

Table 2 Some examples of farm types according to their technical-economic orientation

TF14	
15	Specialist Cereals, Oilseeds and Protein (COP) crops
16	Specialist other fieldcrops
35	Specialist wine
36	Specialist orchards-fruits
37	Specialist olives
38	Permanent crops combined
45	Specialist milk
49	Specialist cattle
48	Specialist sheeps and goats
20	Specialist horticulture
50	Specialist granivores
60	Mixed crops
70	Mixed livestock
80	Mixed crops and livestock

- **Economic dimension range** of the farms: the user can restrict the synthesised population to include only those real agricultural holdings whose Economic Size (ES) classes are of interest for the simulation. The 14 different particular farm types can be added individually or through a 'select all' button.

Table 3 List of the 14 types of farms according to their economic size

ES classes	
1	< 2.000 €
2	2.000 € - < 4.000 €
3	4.000 € - < 8.000 €
4	8.000 € - < 15.000 €
5	15.000 € - < 25.000 €
6	25.000 € - < 50.000 €
7	50.000 € - < 100.000 €
8	100.000 € - < 250.000 €
9	250.000 € - < 500.000 €
10	500.000 € - < 750.000 €
11	750.000 € - < 1.000.000 €
12	1.000.000 € - < 1.500.000 €
13	1.500.000 € - < 3.000.000 €
14	> 3.000.000 €

- **Base year of the simulation:** the user must select the year to which the data from the datasets to be used for the construction of the synthetic population refer. The synthetic population is a static 'snapshot' of the state of the real population in that year.
- **Other parameters:** the user can further restrict the real population to be synthesised according to her simulation interests, e.g. gender (only AH whose managers are women), or age group (only AH whose owners are young people under the age of 35), etc.)

Figure 32 shows a screenshot of the synthetic population generation tool.

The screenshot shows the 'Synthetic population generator application' window. It has a title bar with navigation and window control icons. The main content area is titled 'Synthetic population generator' and contains a section 'Create a new population' with the instruction 'Select parameters to generate a new synthetic population'. Below this are several input fields: 'Geographical scope' with a dropdown showing 'Andalusia (ES61)' and 'Murcia (ES62)'; 'Technical-economic orientation' with a dropdown showing '211 - Specialist vegetables indoor'; 'Economic dimension' with a dropdown showing 'ED4 - ED14' and a Euro symbol; 'Simulation base year' with a text input '2015'; and 'Other' with a text input 'Female owners' and a plus button. There is also a 'Data sources selection' section with the instruction 'Connection to ARDIT to retrieve potentially useful datasets to feed the synthetic population. At least one dataset needs to be selected for each attribute of interest of the agents:'. It features a list of attributes: 'Economic size' (checked), 'Physical size', 'Soil characteristics', '...', and 'Ownership gender'. A dropdown for 'Physical size' is open, showing a list of datasets: 'EU FADN' (checked), 'Spanish agricultural census' (checked), 'ESYRCE', '...', and 'Andalusian SIPEA'. A 'Search data sources' button is located to the right of the instruction. At the bottom right, there is a 'Generate' button.

Figure 32 Synthetic population generator tool

Once all the parameters have been selected, the user will be able to launch a search for available data sources (reflected in the interface through the button "**Search data sources**") that can feed data to the attributes of interest (AoI) of the agents that compose the population. This search is powered through ARDIT, and the result is a list of potential datasets to be used for each one of the attributes of interest. The result of the multi-query to ARDIT will be shown as a view/tab, listing all the AoI of the agents. When selecting one attribute, the possible data source(s) available to extract the information to assign values to it will be displayed. Once that, at least, one data source has been selected for each AoI, a button will be enabled to trigger the generation of the synthetic population itself (reflected in the interface as "**Generate**"). Figure 33 shows the tab used to select the data sources or datasets:

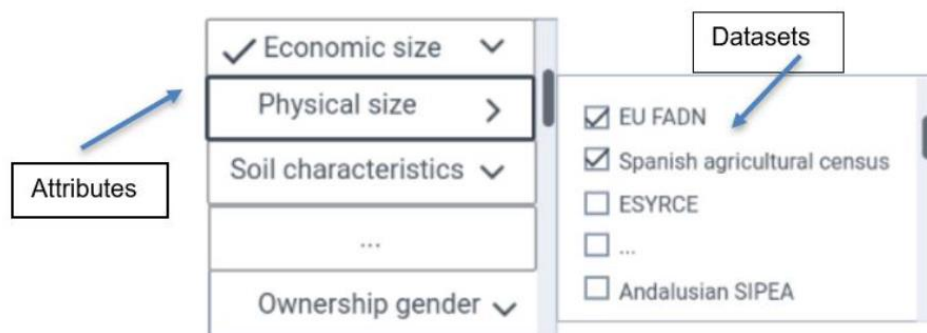


Figure 33 Detail of the selection of datasets extracted from ARDIT

The process of generating the synthetic population displayed in a pop-up or alert window, which would be blocking the user, i.e. the user would not be able to perform any action while it is running (except cancel the generation of the population). Once the synthetic population generation process is completed, the tool offers the user two possibilities (Figure 34):

1. Download the population data in a file, so that the user can import it later in the simulation setup section of the AGRICORE interface.
2. Load the population automatically in the simulation setup lobby, completing the population selection step.

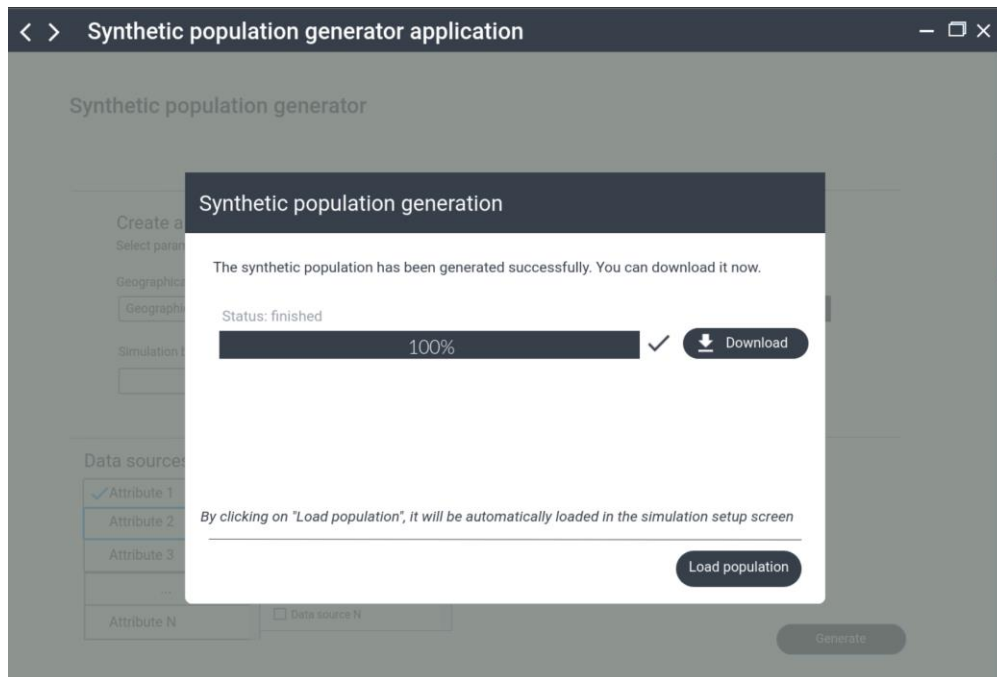


Figure 34 Synthetic population generation finished

4.2.6 Simulation GUI sub-module

Once the user completes all the sections regarding the simulation setup, the button to launch the simulation will be enabled, as can be seen in Figure 35.

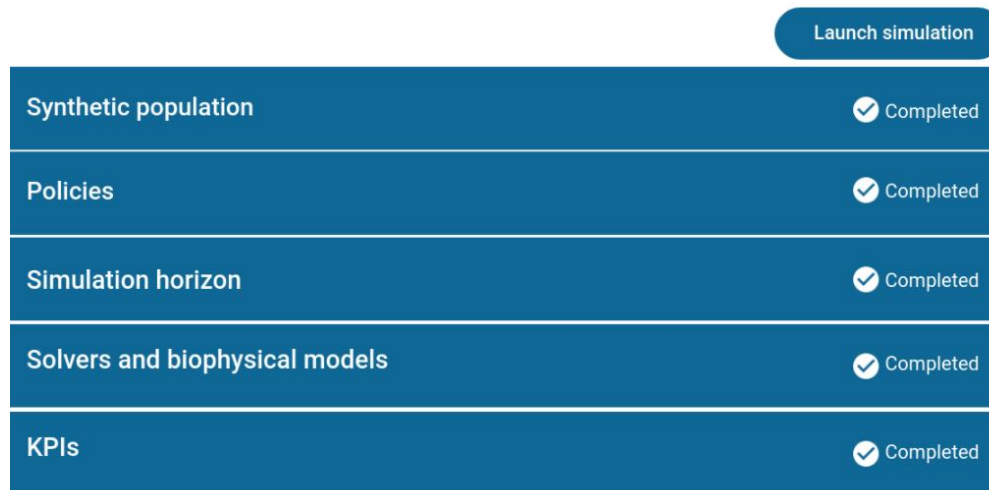


Figure 35 Simulation setup completed

Given that this process can take several hours, it was decided to create a new section in the platform from which users can monitor the status of their launched simulations, as well as view the history of simulations already completed. This solution also allows the simulation process to be non-blocking, allowing users to continue using the platform while the simulation is running.

For this reason, the simulation module was designed, identified in the mockups as "**My simulations**". This screen is again composed of two tables; in the first one, the user can manage the simulations that have been launched and are still running, being able to pause them, stop them or open a window that shows more detailed status information. In the second table, the user will have access to a history of the simulations launched by him/her that are already completed. Here, the user can download the simulation results file, delete the simulation, or navigate to the visualisation module, loading the results automatically in graphs, charts and plots.

In addition, the platform will allow the user to load a setup file from a simulation that has not been executed, by clicking on the "**Load a simulation setup file**" button, which will open a screen to select the location of the file on the user's computer. Once the simulation file has been selected and launched, it will be added to the table of "**Simulations in progress**".

The design of the simulation module is shown in Figure 36.

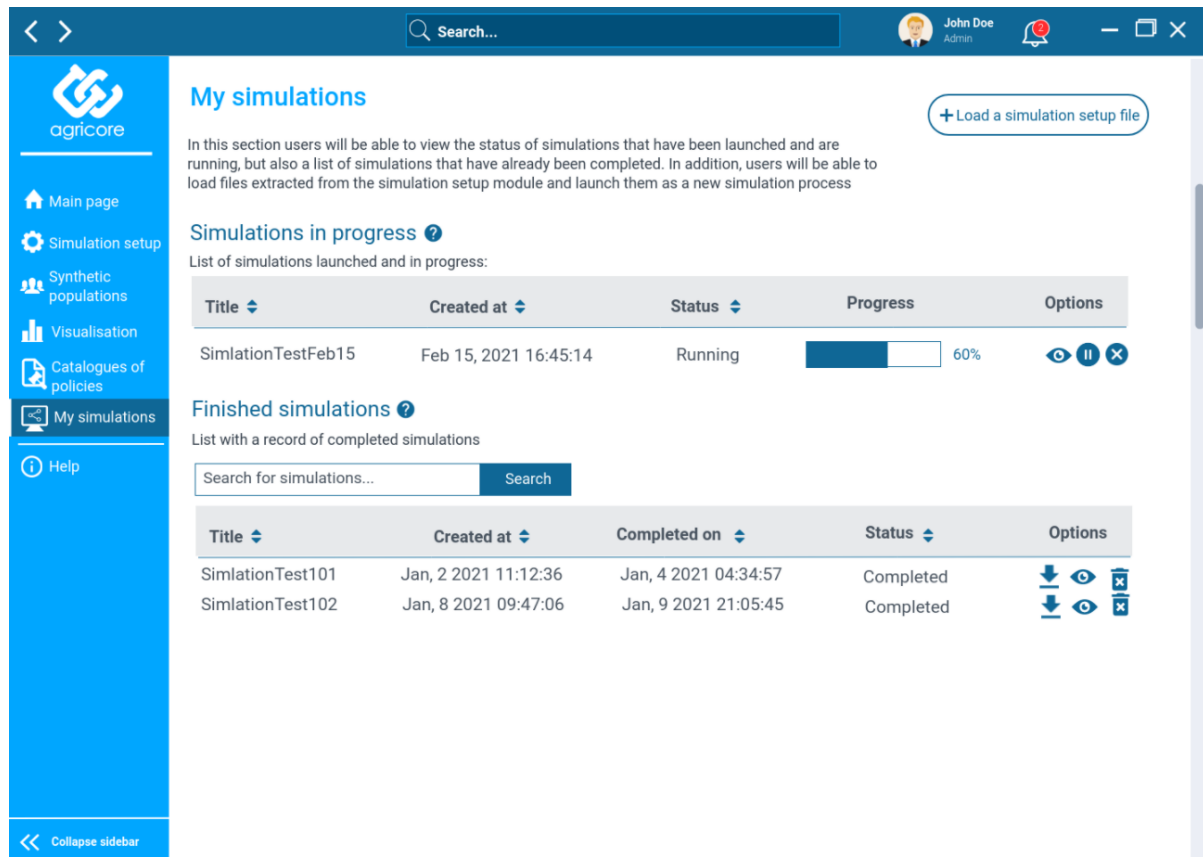


Figure 36 Simulation module screen

4.2.7 Visualisation GUI sub-module

In the visualisation module, users can load the results of the simulations to visualise them graphically in graphs, charts or plots. By default, if we access this module through the sidebar or from the main page of the platform, no graphs will appear, as we must first select which simulation we wish to visualise the data on. In order to load the results of a simulation, there are two options:

- Navigate to the visualisation module and click on the **"Add a results file"** button. Here, the user will be able to select a results file of a completed simulation stored on his/her computer, and load it on the platform.
- From the simulation module, select a **finished** simulation and click on the **eye button** to load the results in the graphs automatically.

By default, graphs will be loaded on the screen using the KPIs selected during the simulation setup. However, it will also be possible to run **advanced visualisations** using **Jupyter notebook**. Thanks to this tool, it will be possible to create interesting and complex graphs using **Python**. The graphs shown in the designed mockup are simply a visual representation of how the screen of this section would look like. The graphs that will be displayed and the possibilities to work or interact with them, will depend on the libraries that will be chosen for the graphical representation of the data.

Figure 37 shows the default design of the visualisation module, with no data loaded, while Figure 38 presents the design with the simulation results already loaded:

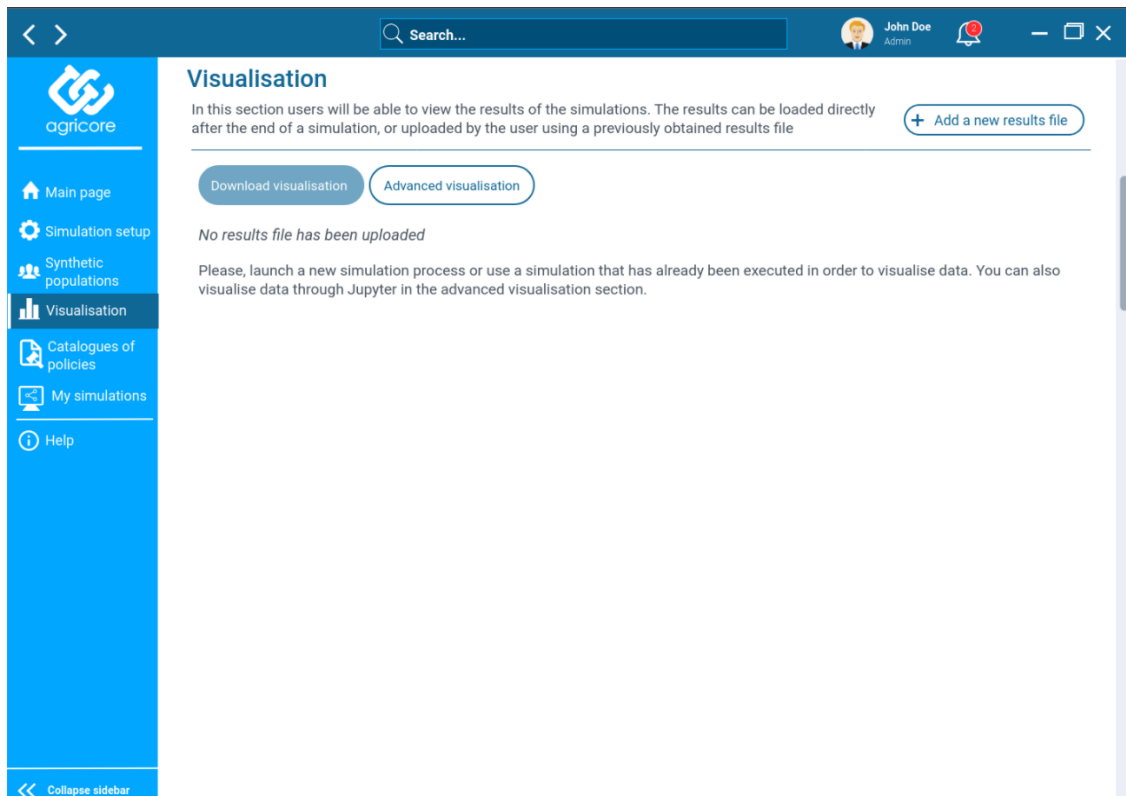


Figure 37 Visualisation module screen without any data loaded

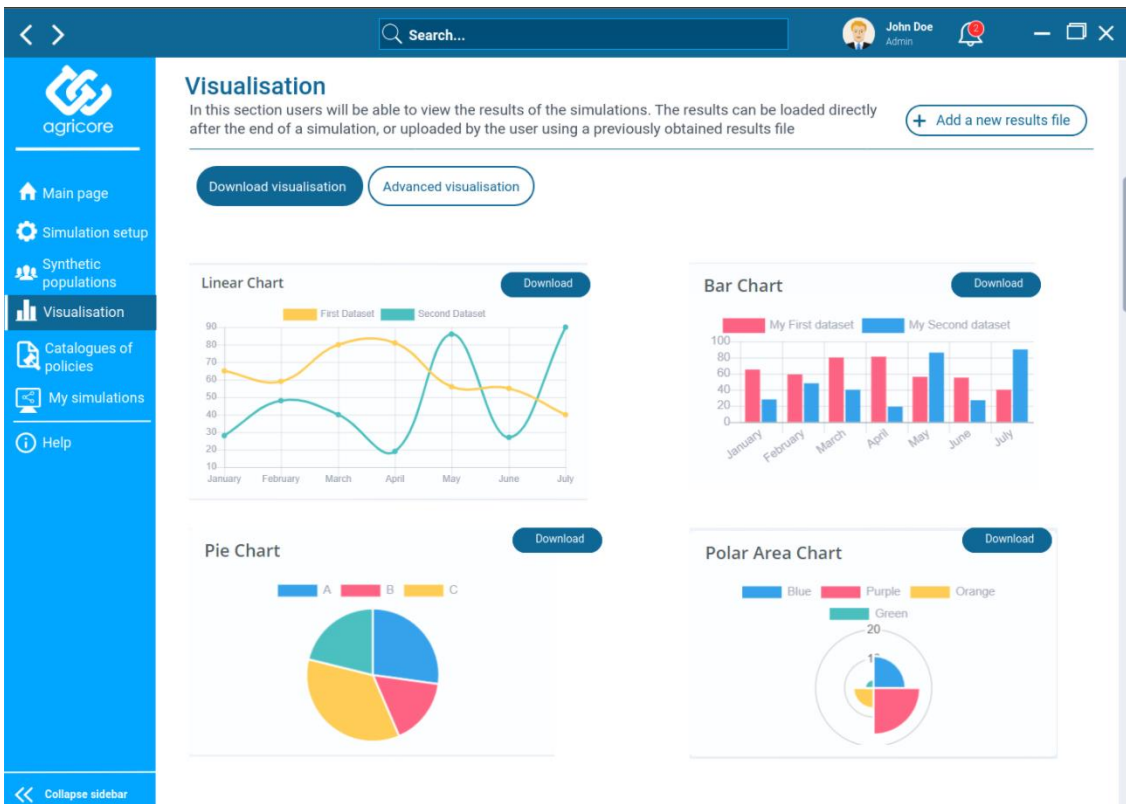


Figure 38 Visualisation module screen with the results of a simulation loaded

4.2.8 General and personal catalogue of policies

In addition, the platform will have separate sections of simulation processes where the users can manage their policy catalogues, facilitating the subsequent incorporation of these policies during the configuration of the simulation. In the section for personal catalogues, users will find a table with all the policies included in them. From there, they will be able to perform the same operations as from the simulation setup screen, being able to create new policies, edit or delete the existing ones. Also, through the "Search for policies" button, shown in the following figure, the platform will open a pop-up where users can filter the general catalogue of policies in search of those they want to add to their own. Figure 39 shows a screenshot of the mockup for the personal catalogue of policies:

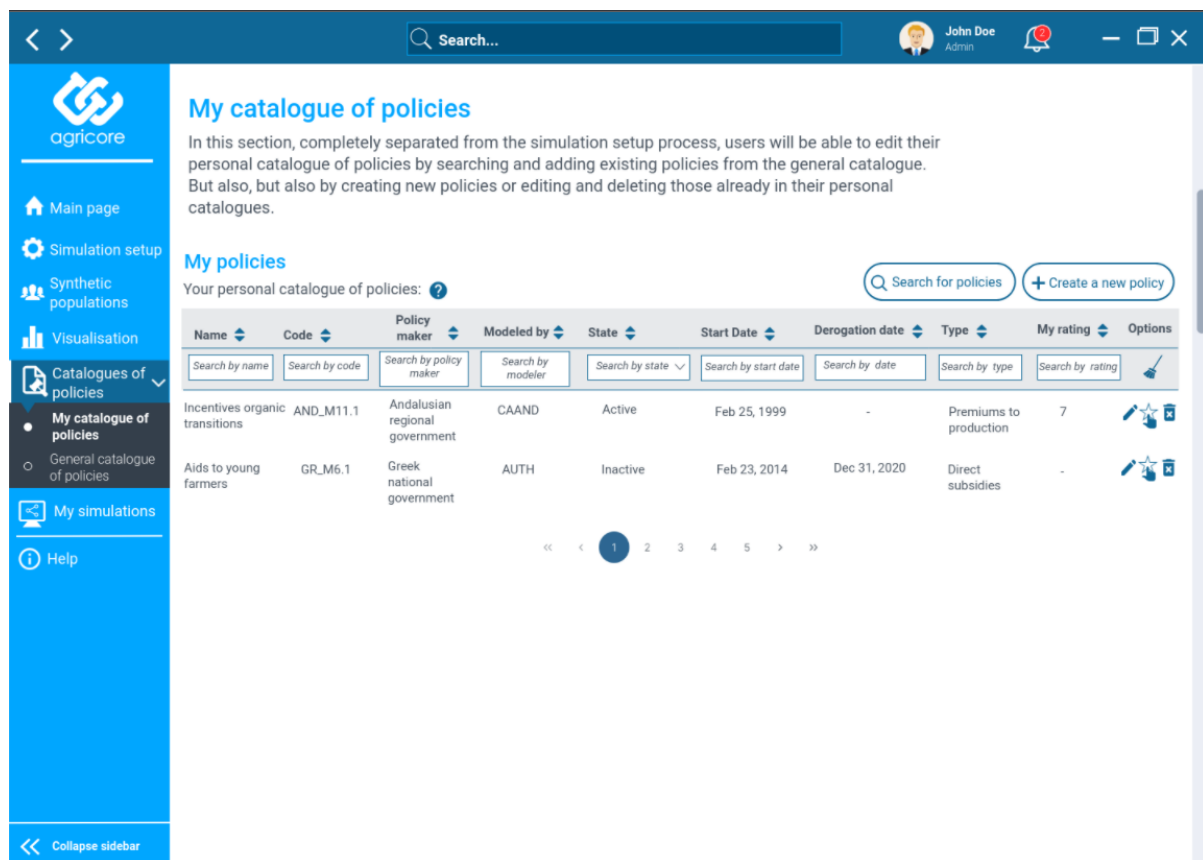


Figure 39 Personal catalogue of policies screen

In the same way, there will also be a separate section dedicated to the general catalogue of the platform, where more advanced users will be able to manage the policies stored in it by creating new ones, editing or deleting existing ones. Figure 40 shows a screenshot of the mockup for the independent section of the general policy catalogue:

General catalogue of policies

In this section, completely separated from the simulation setup process, users with the appropriate permissions can manage the general policy catalogue of the platform adding, editing or deleting the policies contained in it.

Platform's general catalogue of policies. Visible to all users: ? [+ Create a new policy](#)

Name	Code	Policy maker	Modeled by	State	Start date	Derogation date	Type	Average rating	Number of ratings	Options
Search by name	Search by code	Andalusian regional gov...	Search by modeler	Search by state	Search by date	Search by date	Search by type	Search by rating	Search by n° ratings	
Agricultural tax plan	AND_F12	Andalusian regional government	University of Malaga	Active	Feb 25, 2006	Feb 25, 2025	Tax reduction	7	13	Edit Delete
Incentives organic AND_R32 transitions	AND_R32	Andalusian regional government	CAAND	Active	Feb 25, 1999	-	Premiums to production	8	57	Edit Delete

« < 1 2 3 4 5 > »

« Collapse sidebar

Figure 40 Platform's general catalogue of policies screen

5 Design validation procedures

Once the mockups were completed, the process of validating the designs by the end-users of the platform began. It was decided to divide the validation process into two phases, the first, consisting of an internal validation by the project partners to determine their degree of satisfaction with the designs and their opinion on certain points which, at that time, were not clear or finalised. So that their feedback could help to introduce improvements or changes. The second phase would consist of following the same process but with potential users, agents and experts external to the project, including the EAB, policy-makers and researchers contacted within the clustering activities. In order to facilitate this task, a user guide and a form with questions about the interface were elaborated to collect all the answers in a unified and organised way. The form contains questions about the usability, the difficulty of using the interface and certain unclear aspects, to know the opinion of the users in order to make improvements to the designs.

Both the form and the link to the mockups in MarvelApp were sent for validation. The questionnaire, shown below, consists of 24 questions, some of them one choice and multiple choice questions and others, to briefly explain the user's opinion through free text inputs questions.

1. **Name and surname.**
 - Free text input.
2. **Institution or company.**
 - Free text input.
3. **How would you rate the interface in general?**
 - Rating from 1 to 5. Being 1 the minimum rating and 5 the maximum rating.
4. **In general, how would you rate the difficulty of the interface?**
 - Rating from 1 to 5. Being 1 very difficult to understand/use and 5 the opposite.
5. **How would you rate the difficulty of the simulation setup lobby?**
 - Rating from 1 to 5. Being 1 very difficult to understand/use and 5 the opposite.
6. **How would you rate the difficulty of the synthetic population generator tool?**
 - Rating from 1 to 5. Being 1 very difficult to understand/use and 5 the opposite.
7. **How would you rate the difficulty of the user's personal policy catalogue?**
 - Rating from 1 to 5. Being 1 very difficult to understand/use and 5 the opposite.
8. **How would you rate the difficulty of the simulation section?**
 - Rating from 1 to 5. Being 1 very difficult to understand/use and 5 the opposite.
9. **How would you rate the difficulty of the visualization section?**
 - Rating from 1 to 5. Being 1 very difficult to understand/use and 5 the opposite.
10. **Do you consider the content and labels displayed on the screens (descriptive texts, paragraphs, buttons) to be clear and understandable?**
 - Rating from 1 to 5. Being 1 not clear or understandable and 5 the opposite.
11. **Leaving aside the navigation between the mockups in MarvelApp, how would you rate the transitions between the different windows, pages and sections that compose the interface?**

- Rating from 1 to 5. Being 1 the minimum rating and 5 the maximum rating.
- 12. Any navigation/transition detail that you would change?**
 - Free text input.
- 13. Have you encountered problems using MarvelApp navigation?**
 - a. Yes.
 - b. No.
- 14. If you have experienced problems using MarvelApp please, describe them briefly.**
 - Free text input.
- 15. Do you understand the different options for adding a synthetic population to the simulation process? Do you feel that this is well reflected in the interface?**
 - a. Yes, both.
 - b. I understand the options available, but they are not well reflected in the interface.
 - c. I do not understand the options available, nor are they well reflected.
- 16. Do you think the policy selection design for the simulation is adequate? Considering the personal and general policy catalogues, the possibility to create/edit policies, evaluate them, ...**
 - a. Everything is clear for me.
 - b. I don't understand anything in this section.
 - c. I don't understand how to add policies to my simulation process.
 - d. I don't understand the difference between the personal and general catalogue of policies.
 - e. I don't understand how to edit/create policies.
 - f. I don't understand the icons on the buttons and what they are for.
 - g. I don't understand how to rate policy definitions.
- 17. Which of the three different design options or alternatives for the policy selection section do you think is the most appropriate?**
 - a. Catalogues tables above, both opened and in line.
 - b. Selected policies table above and catalogues tables tabs below.
 - c. Catalogues tables tabs above and selected policies table below.
- 18. If you could change anything in the policy selection section what would it be?**
 - Free text input.
- 19. Do you consider that the design of the launched simulations module is adequate and provides the necessary information and details?**
 - a. Everything is clear and fine for me.
 - b. I don't understand the difference between the "launched simulations" and the "finished simulations" tables.
 - c. I don't understand the icons on the buttons and what they are for.
- 20. Do you understand and see reflected in the interface the difference between "simulation setup files" and "simulation results files"?**

- a. Yes, it's clear for me.
 - b. I don't understand the difference.
 - c. It's difficult for me to see the difference in the interface.
- 21. Do you understand the mechanics of importing/exporting simulation setup and simulation results files?**
- a. Yes
 - b. No.
- 22. Do you consider that the design provide the necessary features to achieve the tool's objectives within the AGRICORE project?**
- Rating from 1 to 5. Being 1 the minimum rating and 5 the maximum rating.
- 23. Do you consider that the modules that compose the tool (SPG, simulation setup, visualization, personal catalogue of policies) and the different sections of the simulation process are adequate?**
- Rating from 1 to 5. Being 1 the minimum rating and 5 the maximum rating.
- 24. If you could change something in the interface (design, sections, components, ...), what would it be? And, why?**
- Free text input.

5.1 First validation phase

As a result of the first validation phase, the project partners reported some possible improvements on aspects that were not clear in the interface. In general, the results were quite positive, finding among them the following:

- Average interface rating: 4.5/5
- Average difficulty rating: 4.33/5 (being 5 easy to understand)
- Average difficulty rating on the simulation setup lobby: 4.67/5 (being 5 easy to understand)
- Average difficulty rating on synthetic population generator tool: 4/5 (being 5 easy to understand)
- Average difficulty rating on the user's personal policy catalogue: 4.17/5 (being 5 easy to understand)
- Average difficulty rating on the simulation module: 4.67/5 (being 5 easy to understand)
- Average difficulty rating on the visualisation module: 4.83/5 (being 5 easy to understand)
- Average rating on the descriptive texts, buttons, icons displayed on the designs: 4.5/5
- Average rating on the usefulness of the designs within the project objectives: 4.67/5
- Average rating for the modules that compose the platform: 4.67/5

In addition, regarding the comments made by the partners on possible improvements or changes, the following suggestions were obtained:

- A suggestion was made to highlight on the platform's sidebar the section or module that was open in each mockup, to make it easier for reviewers to locate each mockup within the platform's sitemap.

- As a solution, the sidebar was modified in all the mockups to highlight which section represents each one in the platform, adding a background colour and a small underline on one side (Figure 41).

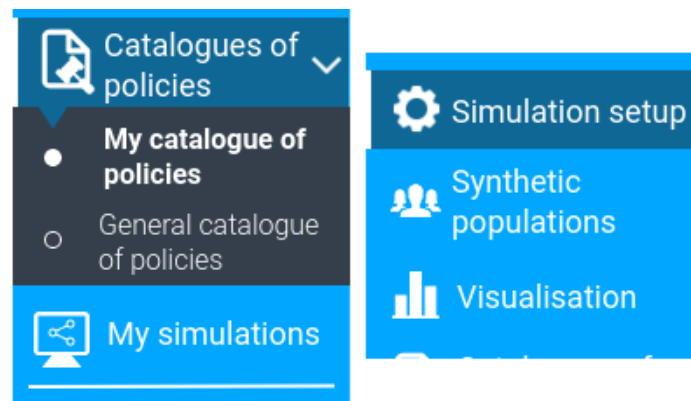


Figure 41 Sidebar detail with the new changes

- In the policy selection section of the simulation setup, some partners suggested to change the order of the catalogues and the selected policy tables, showing the catalogues at the top of the screen. On the other hand, there were some suggestions to display both catalogues, general and personal, completely open and on the same line. Previously, these were separated in two tabs, and when a user clicked on one tab, the other one was hidden.
 - Following the suggestions, the design of the screen was completely changed. A new layout was created displaying the catalogues at the top of the screen and on the same line. In addition, two mockups were created with alternative designs, showing the catalogues in the tab system.
 - A new question was created in the form to ask, in the second phase of validation, which design was the most appropriate for the policy selection screen.
- Given the doubts that some partners showed in some comments and choice questions, some descriptive texts that appear in the mockups were reviewed and updated to make them clearer and more understandable. For example, some additional explanations were added in the policy selection screen to clarify how policies could be added from the general catalogue to the personal one, or from the personal catalogue to the list of policies selected for the simulation.
- Other suggestions from the partners were to try to add more understandable content in the inputs or tables of the platform with possible "real" data. So that, the reviewers would find a more common language.
 - In the screens concerning the generation of synthetic populations and the selection of policies for simulation, which were the lowest-rated due to their difficulty, some examples of what could be real data that the user would find on the screens were added to the inputs and tables (Figure 42).

Name	Code	Policy maker	Modeled by	State	Start Date
Incentives organic transitions	AND_M11.1	Andalusian regional government	CAAND	Active	Feb 25, 1999
Aids to young farmers	GR_M6.1	Greek national government	AUTH	Inactive	Feb 23, 2014

Search by name

Search by code

Search by policy maker

Search by modeler

Search by state

Search by start date

Create a new population

Select parameters to generate a new synthetic population

Geographical scope

Andalusia (ES61) X

Murcia (ES62) X

Technical-economic orientation

211 - Specialist vegetables indoor

Simulation base year

2015

Other

Female owners

Figure 42 Some examples of data added

Finally, in order to facilitate the understanding of the design and the validation process in the second phase, the user guide was modified to consider all the changes made. Some paragraphs were also modified to improve the explanations in those interface sections where the partners had shown more doubts or lower ratings.

5.2 Second validation phase

Once the first validation phase was completed and the changes suggested by the partners were implemented, external stakeholders were contacted to receive new points of view on the designs. The link to the mockups in MarvelApp, the interface user guide and the validation form were sent to the project's External Advisory Board staff and to partners of other H2020 European projects with a background related to the agricultural sector.

After further reviews and feedback, the cumulative results were the following:

- Average interface rating: 4.43/5
- Average difficulty rating: 4.43/5 (being 5 easy to understand)
- Average difficulty rating on the simulation setup lobby: 4.57/5 (being 5 easy to understand)
- Average difficulty rating on synthetic population generator tool: 4/5 (being 5 easy to understand)
- Average difficulty rating on the user's personal policy catalogue: 4.29/5 (being 5 easy to understand)
- Average difficulty rating on the simulation module: 4.71/5 (being 5 easy to understand)
- Average difficulty rating on the visualisation module: 4.29/5 (being 5 easy to understand)
- Average rating on the descriptive texts, buttons, icons displayed on the designs: 4.29/5
- Average rating on the usefulness of the designs within the project objectives: 4.71/5
- Average rating for the modules that compose the platform: 4.57/5

Due to the doubts that some stakeholders had in recognising the icon to rate policies, it was decided to change the icon so that it could be better identified with the associated functionality. In Figure 43 both the original icon (left) and the new and definitive one (right) can be found:

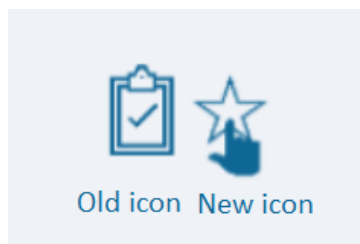


Figure 43 Policy rating icon change

Without any more significant changes in the second validation phase, apart from small adjustments in texts or labels of some buttons to give more coherence and comprehension to the mockups, the validation phase was concluded with quite positive responses and with some suggestions that allowed to increase the usefulness of the designs made.

6 Conclusions

The deliverable D4.3 Validated design for the AGRICORE interface provides a detailed design guide for the AGRICORE interface, which is a fundamental tool for the achievement of the project's main objective, the evaluation of agricultural policies. In the tool, end-users will be able to set up a simulation, launch it, track its progress and visualise its results upon completion. In addition, it will also allow direct access to other tools of the AGRICORE suite (ARDIT, Synthetic Population Generator, Repositories of Policy Definitions) that can be used independently of the simulation workflow.

The final design submitted is the result of a validation process carried out through iterations of evaluation by future users, followed by readjustments to improve gaps or unclear elements. The next steps include:

- Analysing and selecting the most appropriate tools and software libraries for the graphical visualisation of the simulation results
- Implementing semantic and natural language processing capabilities to the platform.
- Starting the development of the user interface itself through the implementation of the designs and functionalities described in this document.

However, the designs depicted might not constitute the final version of the tool. Until the delivery of the final product of the project (AGRICORE Suite) the designs and the platform could undergo some small changes to adapt them to the final developments and designs of the tools that compose the suite, which are currently being developed in parallel. Despite this, following the open-source paradigm of the AGRICORE Project, all developments related to the User Interface will be accessible in due course through a dedicated repository in GitLab.

References

- [1] [^](#)I. for Social and E. C. Economic Research University of Essex; Joint Research Centre, EUROMOD. 2021. [Online]. Available: <https://euromod-web.jrc.ec.europa.eu/>
- [2] [^](#)CAPRI, “Common Agricultural Policy Regionalised Impact Modelling System.” [Online]. Available: <https://www.capri-model.org/dokuwiki/doku.php>
- [3] [^](#)J. Nielsen, “Enhancing the explanatory power of usability heuristics,” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 152–158, Apr. 1994, doi: 10.1145/191666.191729.
- [4] [^](#)ISO, “ISO 9241-210:2019 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems.” [Online]. Available: <https://www.iso.org/standard/77520.html>
- [5] [^](#)W3C, “Web Content Accessibility Guidelines.” [Online]. Available: <https://www.w3.org/TR/WAI-WEBCONTENT/>
- [6] [^](#)S. H. Patrick J. Lynch, Web Style Guide, Foundations of User Experience Design. Yale University Press, 2016. doi: 10.5555/3002831.
- [7] [^](#)D. Saffer, Designing for interaction. New Riders, 2009, p. 21. doi: 10.5555/1795836.
- [8] [^](#)Google, “Angular Material official website.” [Online]. Available: <https://material.angular.io/>
- [9] [^](#)Bootstrap, “Bootstrap official website.” [Online]. Available: <https://getbootstrap.com/>
- [10] [^](#)L. Moreno, “Fundamentals of color in user interface design (UI).” [Online]. Available: <https://uxdesign.cc/fundamentals-of-color-in-interface-design-ui-8127149f13e6>
- [11] [^](#)S. Vaniukov, “Colors in UI Design: A Guide for Creating the Perfect UI.” [Online]. Available: <https://usabilitygeek.com/colors-in-ui-design-a-guide-for-creating-the-perfect-ui/>
- [12] [^](#)V. Xia, The Do’s and Don’ts of User-Centered Design. [Online]. Available: <https://medium.com/@Vincentxia77/the-dos-and-don-ts-of-user-centered-design-4e1cde38e3c7>
- [13] [^](#)Moqups, “Moqups official website.” [Online]. Available: <https://moqups.com/>
- [14] [^](#)Lucidchart, “Lucidchart official website.” [Online]. Available: <https://www.lucidchart.com/pages/>
- [15] [^](#)MarvelApp, “MarvelApp official website.” [Online]. Available: <https://marvelapp.com/>
- [16] [^](#)Justinmind, “Justinmind official website.” [Online]. Available: <https://www.justinmind.com/>
- [17] [^](#)Pencil, “Pencil official website.” [Online]. Available: <https://pencil.evolus.vn/>
- [18] [^](#)Balsamiq, “Balsamiq official website.” [Online]. Available: <https://balsamiq.com/wireframes/>

- [19] [^](https://www.mockflow.com/)MockFlow, “MockFlow official website.”
[Online]. Available: <https://www.mockflow.com/>
- [20] [^](https://proto.io/)Proto.io, “Proto.io official website.” [Online]. Available: <https://proto.io/>
- [21] [^](https://www.mockplus.com/)Mockplus, “Mockplus official website.”
[Online]. Available: <https://www.mockplus.com/>
- [22] [^](https://www.fluidui.com/)F. UI, “Fluid UI official website.” [Online]. Available: <https://www.fluidui.com/>
- [23] [^](https://www.nuclino.com/articles/functional-requirements)Nuclino, “A Guide to Functional Requirements.”
[Online]. Available: <https://www.nuclino.com/articles/functional-requirements>
- [24] [^](https://thetradewiki.com/en/Non-functional_requirement)T. R. V. of Wikipedia, “Non-functional requirement.”
[Online]. Available: https://thetradewiki.com/en/Non-functional_requirement
- [25] [^](https://www.researchgate.net/publication/293756327_User_Interface_Prototyping_Techniques_Methods_and_Tools)P. Weichbroth and M. Sikorski, “User Interface Prototyping. Techniques, Methods and Tools,” *Studia Ekonomiczne. Zeszyty Naukowe Uniwersytetu Ekonomicznego w Katowicach*, vol. 234, pp. 184–198, Dec. 2015, [Online]. Available: https://www.researchgate.net/publication/293756327_User_Interface_Prototyping_Techniques_Methods_and_Tools

Apart from the references mentioned along the document, the current deliverable took into account following inputs:

- AGRICORE Proposal: project proposes a novel tool for improving the current capacity to model policies dealing with agriculture by taking advantage of the latest progresses in modelling approaches and ICT.
- AGRICORE Grant Agreement ANNEX 1 Part A and B, Research and Innovation action, Number-816078: Official Grant Agreement of the AGRICORE project, which defined the terms and conditions of the project, as well as the main requirements of the project.

Deliverable Number	Deliverable Title	Lead beneficiary	Type	Dissemination Level	Due date	Reason
D1.1	Standardised Methodology and Set of Ontologies for the Characterisation of the Data Source	UNIPR	Report	Public	M11	Gather requirements and specification for ARDIT
D6.1	AGRICORE Architecture and Interfaces	IDE	Report	Public	M24	Understand the connection between different modules of the project and the AGRICORE interface
D11.3	Other Requirements Ethics	IDE	Report	Confidential	M18	Accesibility

ANNEX I: Mockup prototyping using MarvelApp

MarvelApp tool

MarvelApp is a web application that allows the design of mockups for mobile, desktop or web applications. It has a small learning curve due to the simplicity of its interface when designing and working with mockups. Although it has several payment options, the free version offers the possibility of working on a single project for an unlimited time, without restrictions when creating new screens or limits on the number of elements or components that the user can add in each one. It also includes the possibility for several people to work on the project simultaneously, but not on the same mockup at the same time. To be able to use MarvelApp, users only need to create an account specifying a username, an email and the password. Once logged in, they can create a project and add the mockups directly from the web browser.

Although the functionality to export the designs in a PDF file or in image format is not available in the free version of the tool, MarvelApp includes the prototyping functionality, which consist of organising and joining the mockups to simulate how the user would actually interact with interface. Since it was necessary to be able to share the AGRICORE interface designs for its validation, it was considered more useful to use the prototyping functionality rather than taking screenshots of each mockup and adding them to a user guide. Furthermore, MarvelApp's prototyping functionality can be shared publicly to anyone using a link, without the need for the recipients to create an account on the tool. And also, with the incentive that, with the prototype, they could navigate and interact with mockups as if they were doing it in the real interface.

The process of working with the mockups in MarvelApp is very simple. Once a user account and a project have been created, there are two main working screens, the main screen or the project lobby and the screen for editing each specific mockup that composes it. The first one list all the existing mockups in the project, with the possibility of adding new ones, copying or deleting them, or opening the editing screen to modify their content. Other options can be found such as links to the project settings page, where the project owner can add new co-workers to the team, or links to the prototype section and the page to share it.

Figure 44 and Figure 45 reveal the project lobby and the page to share the prototype:

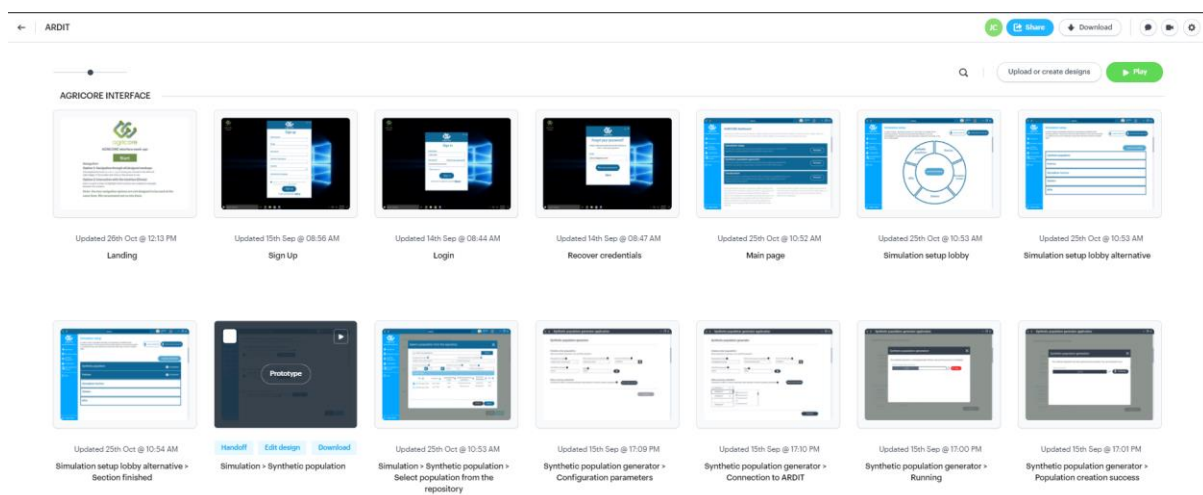


Figure 44 MarvelApp project lobby

Share link

Copy link Send SMS Send Email Embed

<https://marvelapp.com/prototype/6ab0ch2>

Copy

Password protect

Anyone viewing the link will be asked to enter a password



Invite-only

Upgrade

Only visible to admins and collaborators added to this project



Figure 45 MarvelApp page to share the prototype

On the other hand, there is the mockup editing screen which provides menus and toolbars to edit the content of a specific mockup. The process of designing and building mockups consist of adding elements to a canvas, positioning them, editing their sizes and styles, and applying interface design principles to achieve a result as close as possible to what the real platform would look like. To this end, MarvelApp provides different elements for mockup construction such as polygons, circles, lines, text boxes, an icon library and the possibility to add images from the user's computer. In addition, the tool has templates for complex components like a complete login screen. However, these templates do not offer all editing possibilities. So, in the case of the AGRICORE interface designs, all the mockups have been built from scratch using only the elements listed before.

Figure 46 shows the toolbar for editing a text block in MarvelApp, where the positioning options within the canvas, the size, text fonts and styles can be seen:

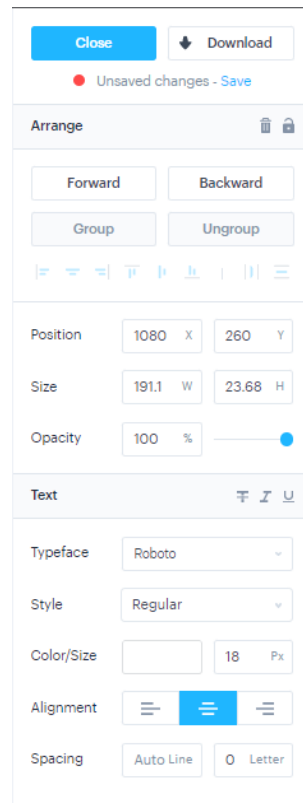


Figure 46 Tool palettes in MarvelApp

As an advantage for the reutilisation of components and to facilitate the construction of mockups, MarvelApp provides three options:

- Create a complete copy of an existing mockup and then edit it according to the requirements.
- Open the editing screens of two mockups, select and copy the element or elements to duplicate in the first one, and "paste" them in the second one.
- Select a number of elements in a mockup and save them in the project's component library. This is the most suitable option for the reutilisation of elements and components, since it allows grouping elements and importing them into any new mockup directly from the library.

An example of using the project's component library would be when there is a menu or button that has a customized size or style and it is necessary to add it to different screens of the platform. The library feature allows the user to maintain the same look & feel and coherence in the design with minimal effort. Moreover, it facilitates the editing of these components as they can be modified in a provisional mockup, saved in the library again and re-imported in the necessary mockups without any difficulty, as the elements that compose the component are grouped together, which makes it easier to position them in the canvas.

MarvelApp prototyping

The prototyping process can be understood as:

"...an operational model of developed system, which shows not only interaction between a user and a system, but also exposes some subset of its functionality... a testing and evaluation approach which is a crucial component of User-Centred Design (UCD) methodology, widely used in IT projects since the 1990s [Sharp, 2005]...Prototyping not only helps to visualize design concepts in an

interactive way, but also supports expressing new requirements and expectations towards a prospective system. Usability testing of user interface prototypes validates user requirements and functionality of the prototype under evaluation"[25]

MarvelApp tool has a prototyping functionality that allows transforming static mockups into interactive screen designs, allowing to add transitions and interactions in the elements that compose the mockup, such as buttons, menus, toolbars, links, etc. This functionality allows end-users to test the designs and navigate between them, simulating as if they were doing it in the real application or platform. Facilitating the user experience, the understanding of the designs and the possible validation of them. The process of building the prototype is simple, it is only necessary to highlight the parts of the mockup that need to be interactive, using rectangles that can be linked later to a different mockup in order to create the transition between them. In the Figure 47, the red squares delimit the interactive areas of the mockup that allow the user to navigate to other mockups:

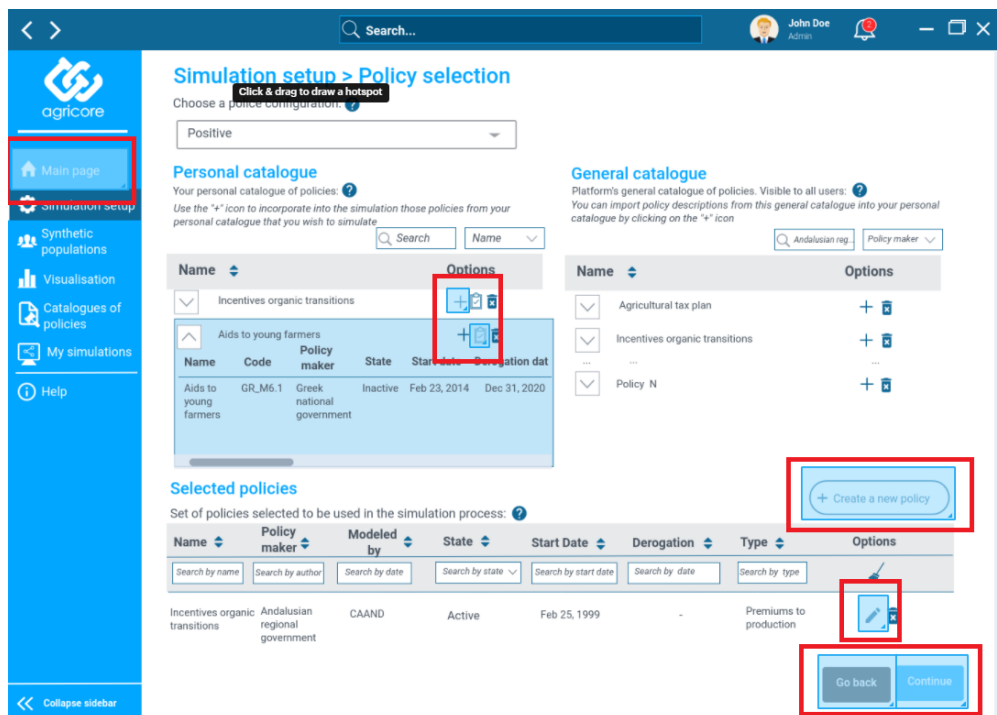


Figure 47 MarvelApp prototyping

In short, through the link to share the interface designs, users can interact with the mockups in two possible ways:

1. Iterating over each of the designed mockup, using the keyboard arrows or moving the mouse to the left and right edges of the screen and clicking on the arrows that appear on the screen.
2. Using the elements enabled and highlighted on the prototype to simulate how the real navigation between the screens would be.