E-LAND

D4.6 Visualisation Software Prototype

E-LAND

*Integrated multi-vector management system for **E**nergy is**LAND**s*

Deliverable nº:	**D4.6**
Deliverable name:	**Visualisation Software Prototype**
Version:	**1.0**
Release date:	**26/11/2020**
Dissemination level:	**Public**
Status:	**Submitted**
Author:	**Isidoros Kokos (ICOM)** **Viktor Alkalais(ICOM)** **Dimitris Karagkounis (ICOM)** **Ioanna Lazaridou (ICOM)** **Ilias Lamprinos (ICOM)** **Nikolaos Ioannidis (ICOM)**

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No 824388.

Document history:

Version	Date of issue	Content and changes	Edited by
0.1	14/10/2020	Table of Contents	ICOM
0.2	10/11/2020	1st Draft	ICOM
0.3	19/11/2020	1st Review	SIN, RLI, GECO
0.4	20/11/2020	2st Draft	RLI
0.5	25/11/2020	2nd Review	SIN, RLI, GECO
1.0	26/11/2020	Final version for TMT approval	ICOM

Peer reviewed by:

Partner	Reviewer
SIN	Farhan Farrukh
RLI	Martha Hoffmann, Pierre-Francois Duc
GECO	Thomas Mikkelsen

Deliverable beneficiaries:

WP / Task
WP4
WP6
WP8

Table of contents

Abbreviations and Acronyms

Acronym	Description
CSS	Cascading Style Sheets
DBMS	Database Management System
DER	Distributed Energy Resources
DVA	Data Visualisation Application
HTML	Hypertext Markup Language
EPA	Energy Planning Application
ESB	Enterprise Service Bus
ESS	Energy Storage System
JS	JavaScript
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LES	Local Energy System
MDB	Material Design for Bootstrap
MVS	Multi-Vector Simulator
oemof	Open Energy Modelling Framework
QoE	Quality of Experience
ORM	Object Relational Mapper
WAI-ARIA	Web Accessibility Initiative – Accessible Rich Internet Applications
WP	Work Package

Executive summary

The purpose of this document is to present the Energy Planning Application (EPA) and the Data Visualisation Application (DVA), two solutions which are part of the E-LAND Toolbox – a modular set of methodologies and ICT tools targeted at optimizing the operation and planning of multi energy Local Energy Systems (LES) and isolated communities. This report concerns on one hand the design - providing the implementation view of the solution- on the other, it provides an introduction to the concepts of each application and the guidelines of use.

The EPA is a web application providing a user interface to users interested in investment or operational planning for local energy assets in the context of a LES. EPA acts as the front-end, i.e. a graphical user interface for the Multi-Vector Simulator (MVS), another tool of E-LAND Toolbox, that enables the evaluation of local sector-coupled energy systems that include multiple energy carriers (e.g. electricity, heat and/or gas) via analyzing the energy system model and calculating future energy supply scenarios as well as the optimal sizing of future investments.

The application drives the end user through a number of steps for providing the input for the modelling and optimisation via the creation of:

- A *project*, supporting a plan for a specific site;
- *The scenarios*, representing different case studies of interest;
- *A grid model, the set of assets* (e.g. PV plants, wind plants, ESS, transformers, generators) of a scenario – either existing ones or new assets of interest (optimisation variables) – detailing their technical characteristics and their physical topology in the system.

Upon collecting the necessary data, MVS is communicated with the needed input and when the results are available to the EPA, the user can navigate to a dashboard and observe economic as well as technical indicators of the optimisation performed by MVS, as well as time series of the operation of the local assets.

The application can support multiple users, with each user having permission for own projects, as well as projects shared with them as part of a user group.

Data Visualisation Application is a web application that enables the visualisation of the project's KPIs and operation of local energy assets, providing insights on the operation of the pilots (local energy systems where the E-LAND solutions are validated) to the pilot

owners (end-users) and the local community, as well as third parties interested in the project's results through the project's dissemination activities.

A list of the KPIs that will be monitored via this application follows:

- Percentage reduction in CO2 emissions
- Reduction in peak power
- Percentage of demand covered by local renewable generation
- Percentage reduction in electric energy from grid
- Increase in storage usage efficiency
- Extended lifetime of battery storage

The details for the calculation of the above indexes are documented in deliverable D5.3 [1].

DVA provides a dashboard interface for the visualisation of the monitored indexes. The user can visualize indexes both at a project level, as well as at a pilot (i.e. LES) level. The application support both registered and unregistered users, with different access rights (permissions) to the webpages of the application.

1 Introduction

1.1 The E-LAND project

The continued decarbonisation of the energy sector through the use of renewable energy sources provides both interesting opportunities for Local Energy Systems (LES) and challenges for existing electricity networks. Mainland regions such as isolated villages, small cities, urban districts or rural areas oftentimes have issues with weak or non-existing grid connections. These areas are known as energy islands.

The goal of the European-funded H2020 project E-LAND is to provide a synergistic solution between technological, societal and business challenges that the energy sector faces. The main outcome will be the E-LAND Toolbox – a modular set of methodologies and ICT tools to optimize and control multi energy islands and isolated communities. The modular toolbox can be customized to meet local requirements and is expandable to incorporate new tools as new challenges arise.

1.2 Purpose

The purpose of this document is to document the Energy Planning Application (EPA) and the Data Visualisation Application (DVA), two solutions which are part of the E-LAND Toolbox. This documentation concerns on one hand the design, providing the implementation view of the solution, whilst on the other to provide an introduction to the concepts of each application and the guidelines of use.

1.3 Intended Audience and Reading Suggestions

The document is intended to address a diverse audience. The main target group is E-LAND project's partners, especially the pilot owners, which will be the end-user of both applications. It also addresses the dissemination team of the project, since the visualisation of DVA could be used for such purposes. Furthermore, given the public nature of this work, the authors of the document aimed at addressing interested readers such as possible end-users of E-LAND's solutions, Smart Grids domain experts (either academic researchers or industrial partners), as well engineers involved in the development of similar solutions.

Prior to reading this document it is highly recommended to have a clear overview of E-LAND´s vision and the scope of the specific applications with the toolbox. Therefore, readers outside E-LAND's consortium, are advised to refer to chapter 2 in order to better understand the project's main concepts and tools. The work documented in Deliverable D3.2 [2] is also a good reference to understand the requirements documented for each application during the requirement engineering process.

1.4 Report Structure

The document is structured as follows:

- The first chapter provides a short introduction of the E-LAND EU H2020 project, briefly highlights the purpose of the report, the intended audience and reading suggestions, as well as its structure.
- In the second chapter, E-LAND Toolbox is presented, detailing the component of the different layers and interrelations.
- Chapter 3 provides an overview of the Energy Planning Application (EPA) and its building blocks. It also briefly describes the Multi-Vector Simulator (MVS) - the tool (of E-LAND Toolbox) utilized to perform the economic and technical analysis, based on end user input provided via EPA. Guidelines of the application usage are also provided.
- Following, in chapter 4, describes the Data Visualization Application (DVA) of the toolbox. An overview of the application and its core building blocks is provided. It also provides information on the project KPIs that will be monitored via the application. Guidelines of application usage are also provided.

2 E-LAND Toolbox

Local Energy Systems (or Energy Islands) pose important challenges in their operation due to their weak interconnection with the bulk power system. Smart Grid solutions such as Demand Response (DR) schemes, smart EV charging and utilisation of Distributed Energy Resources (DER) assets can effectively reduce operational costs, equivalent CO2 emissions and increase the reliability in such isolated parts of the grid.

The E-LAND Toolbox comprises several components for enabling the future wide-scale deployment of Multi-Vector LES, enabling the co-optimization of electricity, gas and heat networks. A high-level view of the toolbox is presented in Figure 1 (as analysed in the work of [3]), whereas the main components of the system are presented in the following list:

- *Data Pre-processing (DP)*: An application able to analyse timeseries data set label missing/corrupted data, and provide correction or estimations whilst it is able to extract profiles.
- *Data Visualization Application (DVA)*: Application providing monitoring of the operation of local assets and of the project's KPIs.
- *Energy Forecaster (EF)*: An application offering the prediction of energy profile of local generation (e.g. wind turbines, PV panels), as well as consumption assets.
- *Energy Planning Application (EPA)*: An application providing a user interface for providing the input of the planning of the LES as well as accessing and visualizing the results of the calculations.
- *Multi-Vector-Simulator (MVS)*: An application responsible for simulating the different energy flows in a LES, as well as for enabling its future planning through optimal sizing/operational management of local assets.
- *Optimal Scheduler (OS)*: An application capable of calculating optimal (long-, short- and ultra-short-term) schedules considering the diverse type of assets of the LES and the various parameters (e.g. uncertainty of the forecasts, energy prices, operational constraints etc.).
- *Enterprise Service Bus (ESB)*: A system enabling the integration of the forecasting and optimisation tools, the EMS of the LES and the various external data providers.

A detailed documentation of functionalities of the toolbox is documented in [2].

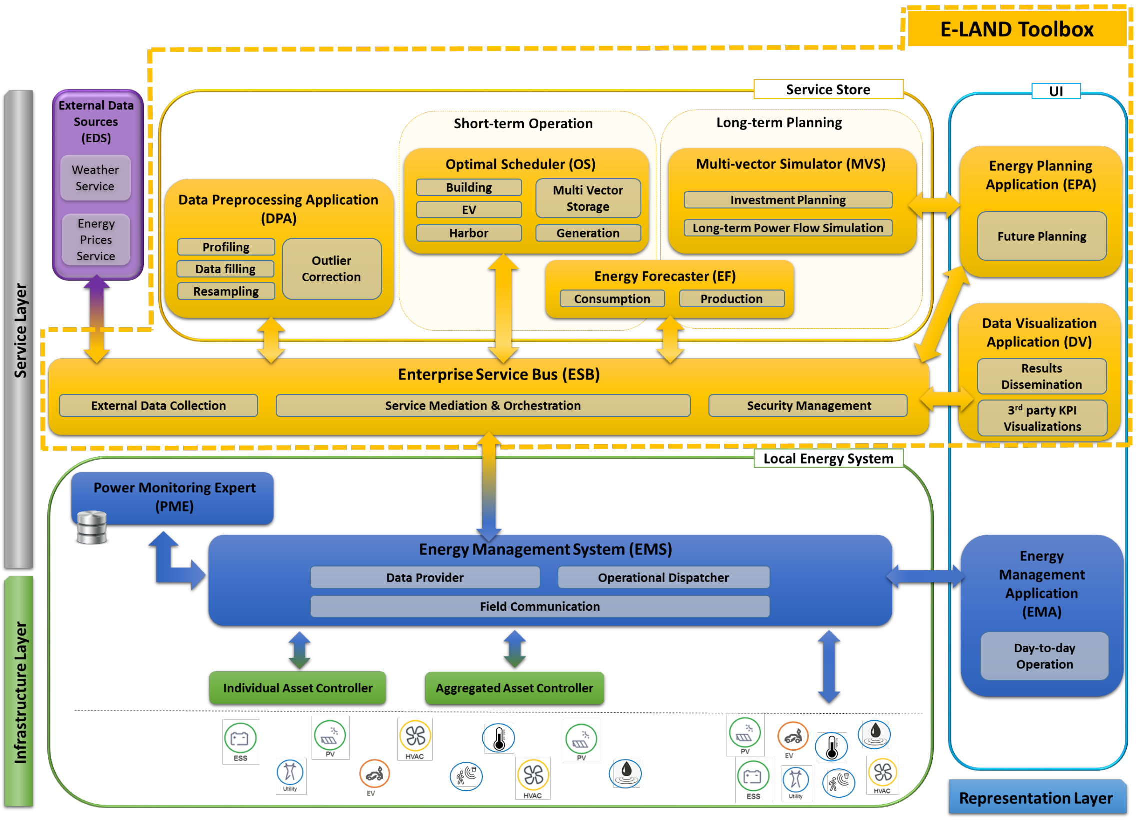

Figure 1 Context View (revised from [2])

3 Energy Planning Application (EPA)

3.1 Overview

The Energy Planning Application (EPA) is a web application providing a user interface to users interested in investment or operational planning for local energy assets in the context of a Local Energy System (LES). This section presents the conceptual design of the solution.

The application drives the end user through a number of steps for providing the necessary input to perform the simulation of the future operation of a LES. The user can create different *projects*, supporting planning of multiple sites.

Each project can have multiple *scenarios* (case studies), representing different solutions of interest for deployment, different constraints or different operational conditions (e.g. load/generation profiles of existing assets in the LES). It is possible to copy and edit an existing scenario, or import a scenario in file format. For each scenario a model of a local energy system is defined using an intuitive drag and drop design tool. By utilizing this interface, the user can create *assets* (e.g. PV plants, wind plants, ESS, transformers, generators) specifying existing assets as well as new assets of interest (optimisation variables). A pop-up dialogue (modal) allows to specify their characteristics. Assets are connected through system *busses* – detailing the physical topology of the system – to form the model that will be simulated.

The user can optimize the scenarios by providing required data and assumptions (project-specific economic and technical data). Upon collecting the necessary data, a request for a *simulation* can be sent by EPA to the Multi-Vector Simulator (MVS). The data in the request are formulated in a JSON file. As soon as the MVS simulation is completed and the results are available to the EPA, the user can navigate to a dashboard and assess time series of the operation of the local assets as well as the economic technical key performance indicators of the energy system resulting from the optimisation performed by MVS.

The application can support multiple users, with each user having permission for own projects as well as user groups, sharing common projects. A class diagram of the "building blocks" of the main application is presented in Figure 2, while a second related to the classes of the dashboard - where the simulation results are visualised – is presented in Figure 3.

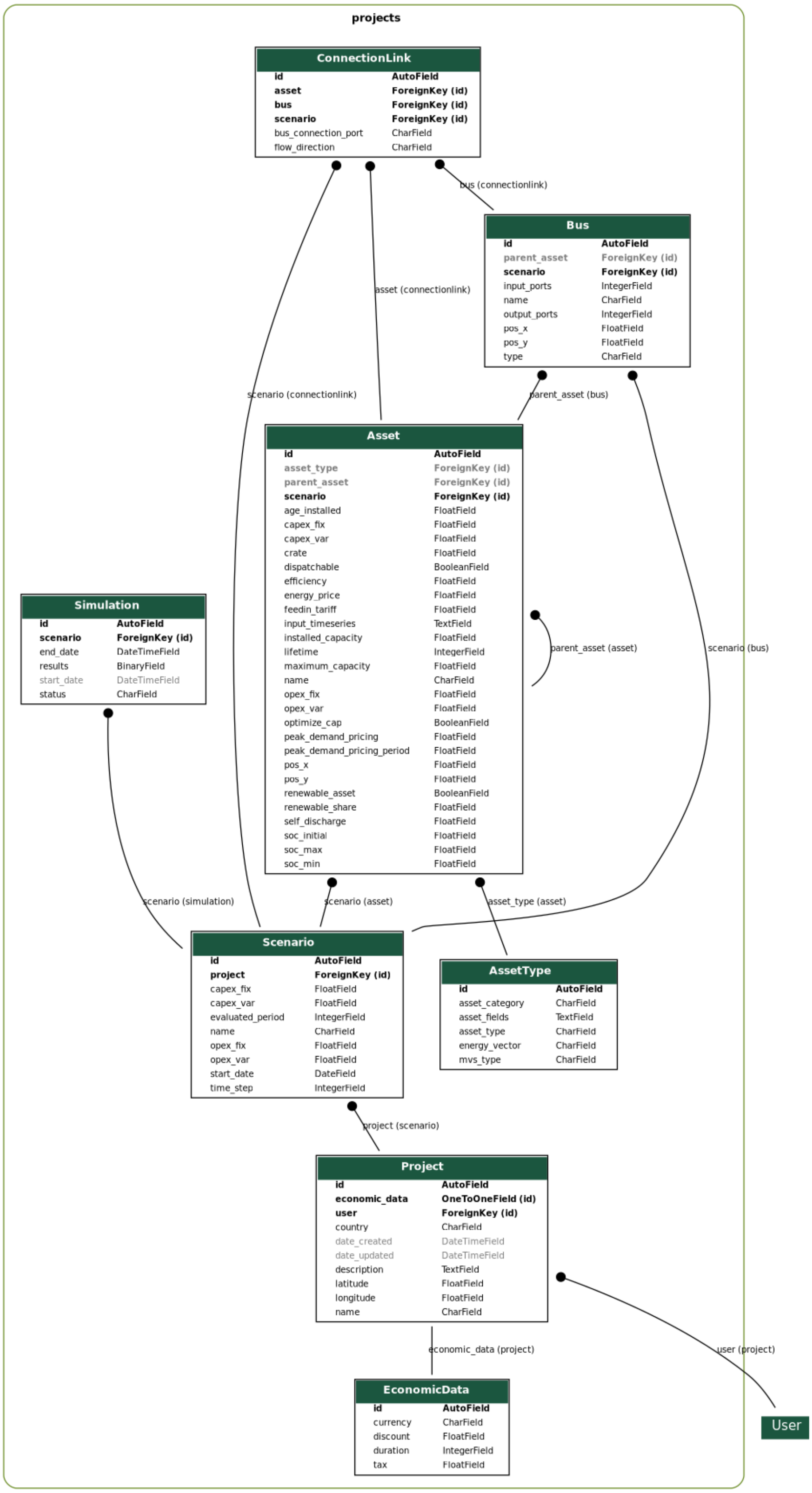

Figure 2 Project Main Class Diagram

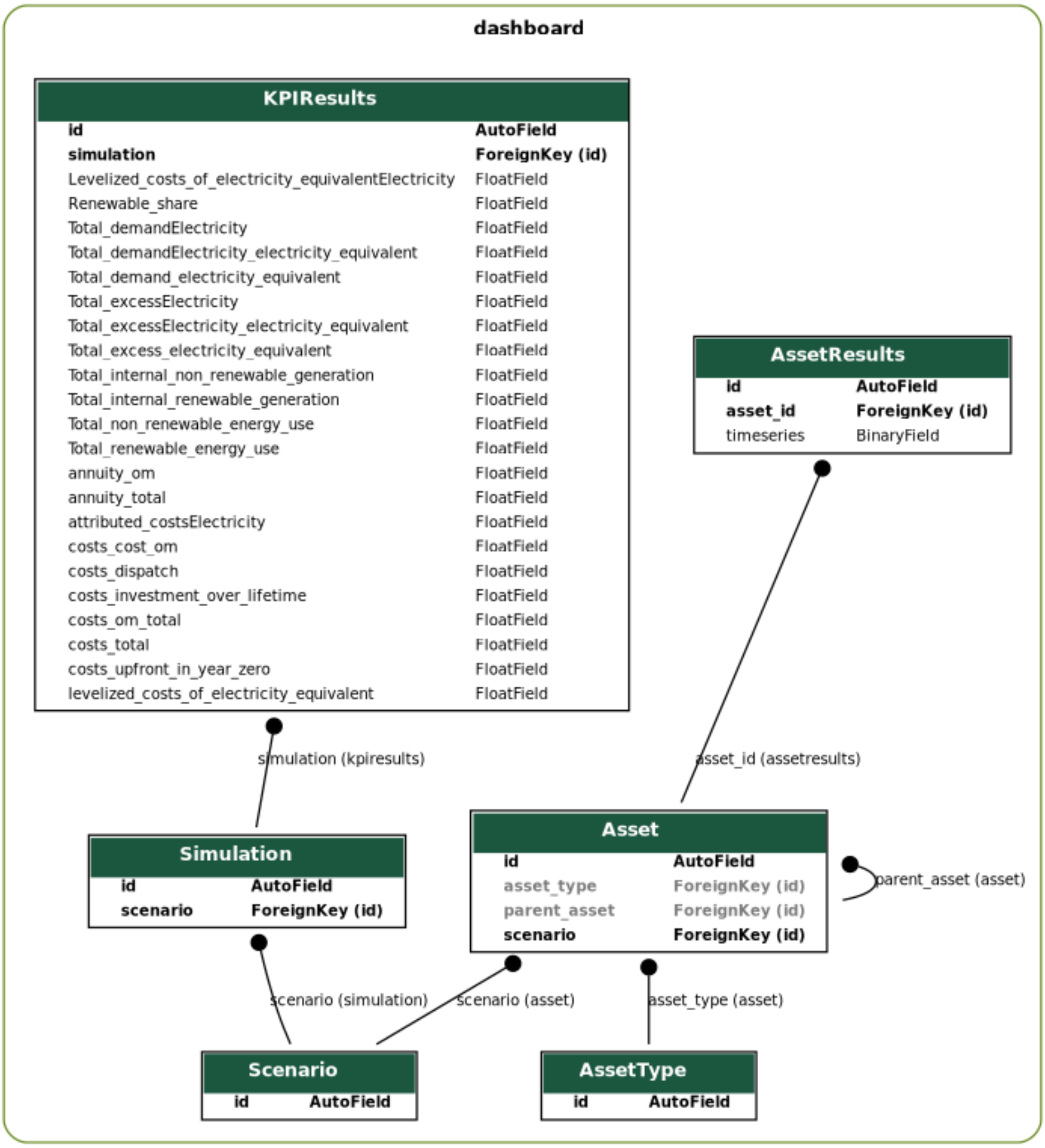

Figure 3 Simulation Results Class Diagram

3.2 Multi-Vector Simulator (MVS)

The Multi-Vector Simulator [4] is a tool developed within the scope of the H2020 project E-LAND and allows the evaluation as well as investment and dispatch optimization of local sector-coupled energy systems that include multiple energy carriers: electricity, heat and/or gas.

The MVS has three main features:

- Analysis of a current energy system, which can be defined from csv or JSON files, including its costs and performance parameters.

- Near-future (months up to years) investments into power generation and storage assets can be optimized aiming at least-cost supply of electricity and heat.
- Future energy supply scenarios that integrate emerging technologies helping to meet sustainability goals and decrease adverse climate effects can be evaluated, e.g. through high renewable energy shares or sector-coupling technologies.

The MVS utilizes the Open Energy Modelling Framework (oemof) [5], a programming framework for energy system modelling and optimisation. Within this, the library *oemof-solph* allows building and solving linearized energy system models. The MVS will be provided as an open-source standalone tool, allowing it to be adapted to future needs by other researchers and developers.

The EPA will provide the graphical user interface for the MVS, facilitating the interaction of end-users with the tool. Therefore the EPA grid model, i.e. the model of the local energy system defined by the end user, as well as all assets defined in it, are limited to the MVS energy assets types and oemof types alike.

3.3 Design

The EPA utilizes various technologies in order to provide reliable and high Quality of Experience (QoE):

- The web application core is implemented in *Django* [6], a web framework written in Python;
- *MySQL* [7] is the selected Database Management System (DBMS), in which all required data (inputs of user, output of MVS) are stored;
- *Bootstrap 3* [8], *JQuery* [9], *Drawflow* [10], *Selectize* [11], *PlotlyJS* [12], *SweetAlert2* [13] and *JQuery-UI* [14] are the major *JavaScript* libraries used either to achieve specific front-end functionalities or to enrich the appearance of the application to the end use;
- *Nginx* [15] is utilized as a reverse proxy and load balancer.

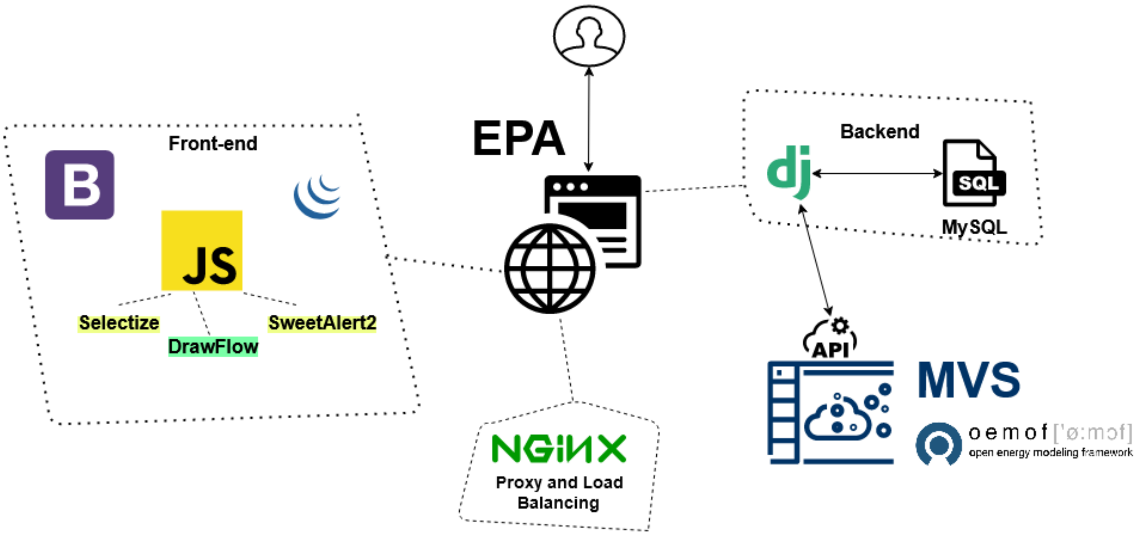

Figure 4 EPA Technology Dependency Stack

3.3.1 Django

Django is a well-established, scalable, secure and performant web framework. The framework provides a vast set of built-in functionalities required in most web applications, thus simplifying the application creation process to the bare minimum.

Apart from *Django Core*, there are many Python modules targeting the *Django* framework, which aim to enhance the created application. Such a module is *Django-Crispy-Forms* [16], used throughout EPA for the creation of bootstrap ready HTML forms.

Django provides a powerful Object Relational Mapper (ORM), which allows to interact with the underlying database directly through python code. Another important feature of the framework is the ability to explicitly set access rules (through *Mixin classes*) for each view of the application. As a result it is easier and more straightforward to identify possible security issues.

The EPA is based on *Django* v3.1.1.

3.3.2 Drawflow

Drawflow is a JavaScript library, licenced under the MIT licence, without any external dependencies. *Drawflow* enables the creation of graphical flow diagrams using predefined drag and drop objects.

The EPA utilizes the library in the definition of the model of the local energy systems, where the user is able to define all energy assets of a project's scenario. *Drawflow* provides built-in methods to gather and export editor nodes data to JSON format. The generated JSON dictionary is sent to the backend through an *Ajax* post request for further processing, validation and database updating.

3.3.3 MySQL

MySQL is an open source DBMS. The technology provides the ability to explicitly define relational schema in a secure manner, while maintaining high performance and out of the box scalability support. All EPA data are stored in the relational database by utilizing tables and foreign key for the tables' inter-association.

3.3.4 SweetAlert2

SweetAlert2 is a JavaScript (JS) library which aims to improve the visual representation of JS built-in alert boxes for aesthetic and accessibility purposes (WAI-ARIA) alike. EPA utilizes the library mostly in the energy system model definition page in order to enhance end user QoE.

3.3.5 Selectize

This JS library provides a customized version of the default <*select*> tag. It best fits in cases when there is a big list of options to select from. EPA utilizes the library in the simulation results visualization dashboard for facilitating the user to select what data to visualize for analysis.

3.3.6 Bootstrap 3

Bootstrap is a full and open source CSS framework, which provides a vast collection of responsive controls. An important feature of the framework is the extended support it provides out of the box for most of the major web browsers. The framework enables the front-end of each and every EPA page. In EPA version 3 of the framework is utilized.

3.3.7 JQuery-UI

JQuery-UI is a JQuery library which delivers an enriched collection of widgets aiming to introduce visual functionalities to the front-end. Accordion - a collapsible content of panels - is a widget which can be found in EPA, listing the elements in an intuitive manner for the end user.

3.3.8 Plotly JS

Plotly is an open source, MIT licenced, JavaScript chart generation library. The library is built on top of D3.js, a powerful visualization library. Timeseries, pies and other chart-based visualizations shown in the simulation results visualization (dashboard) page are created using *Plotly*.

3.3.9 NGINX

Nginx [15] is an open source, BSD 2-clause licensed, software. It is a web server mostly used as load balancer and reverse proxy to enhance server performance. A major advantage of *Nginx* is its asynchronous event driven approach to handle request. The EPA utilizes *Nginx* as a reverse proxy and load balancer, two features enhancing the application reliability, performance and security.

3.4 Guidelines of use

The EPA aims to function as the driving tool for the energy system planning and the economic and technical analysis defined by the end user. The user is required to follow a number of steps in order to fill in the required input to carry out the analysis of the system. An updated workflow [3] of the possible operations is illustrated in Figure 5.

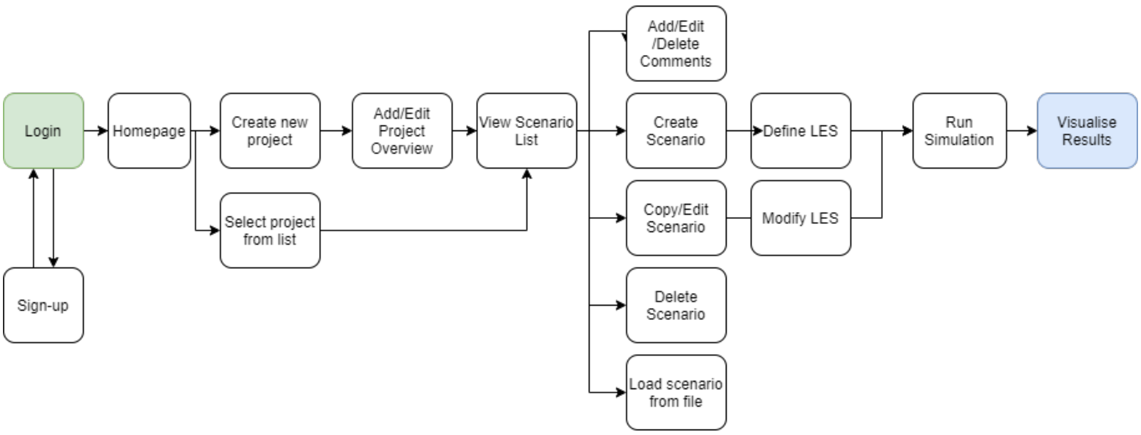

Figure 5 EPA Workflow

Once the user navigates to the EPA, the web app validates if the user is logged-in and redirects to the proper page. If the validation is unsuccessful the login page is shown (Figure 6). The page asks the user credentials. This landing page also provides general information regarding E-LAND project and the application.

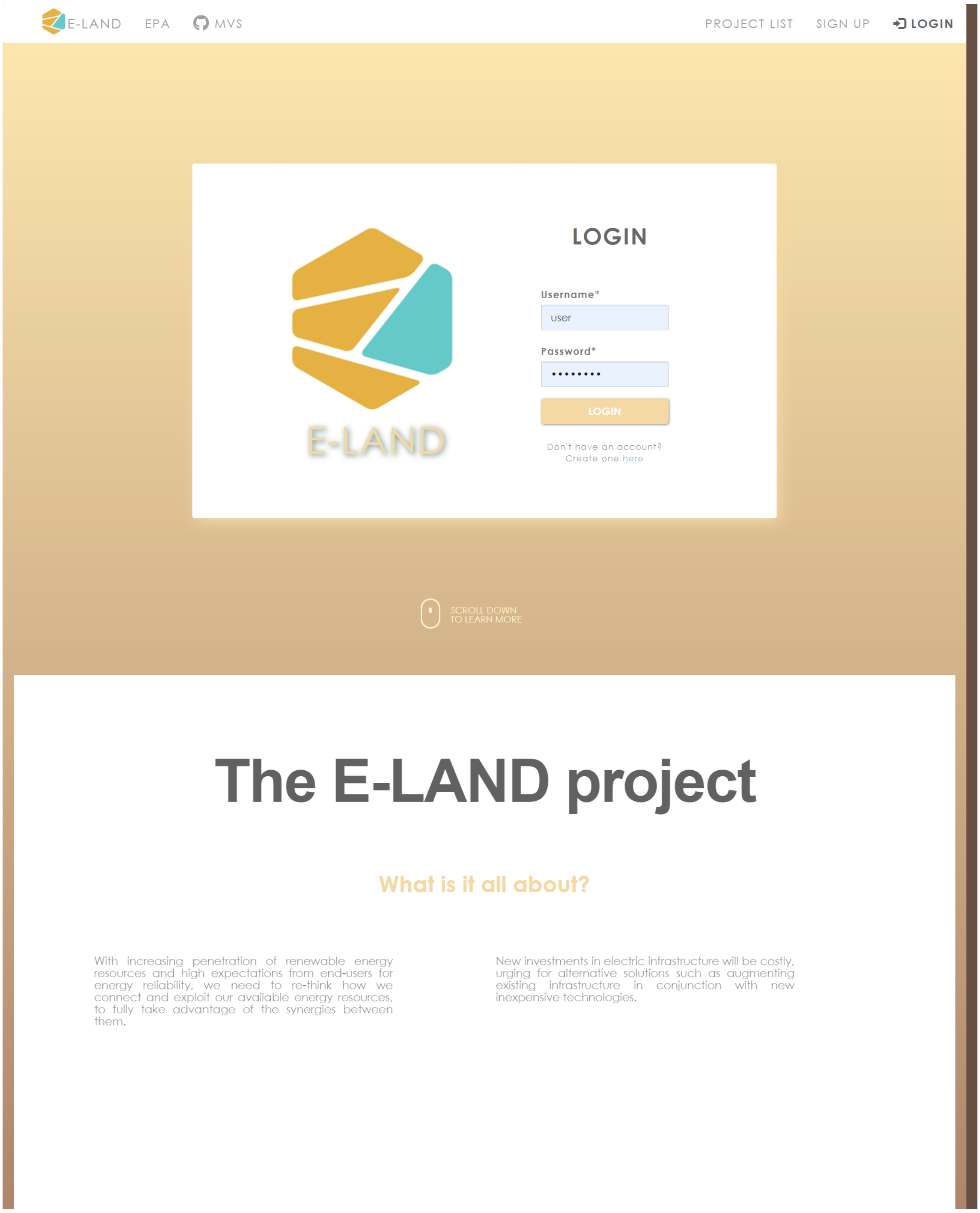

Figure 6 EPA: Login Page

If the user doesn't have login credentials the application asks the user to register an account. An admin account can manage user acceptance and management. The signup page is presented in Figure 7.

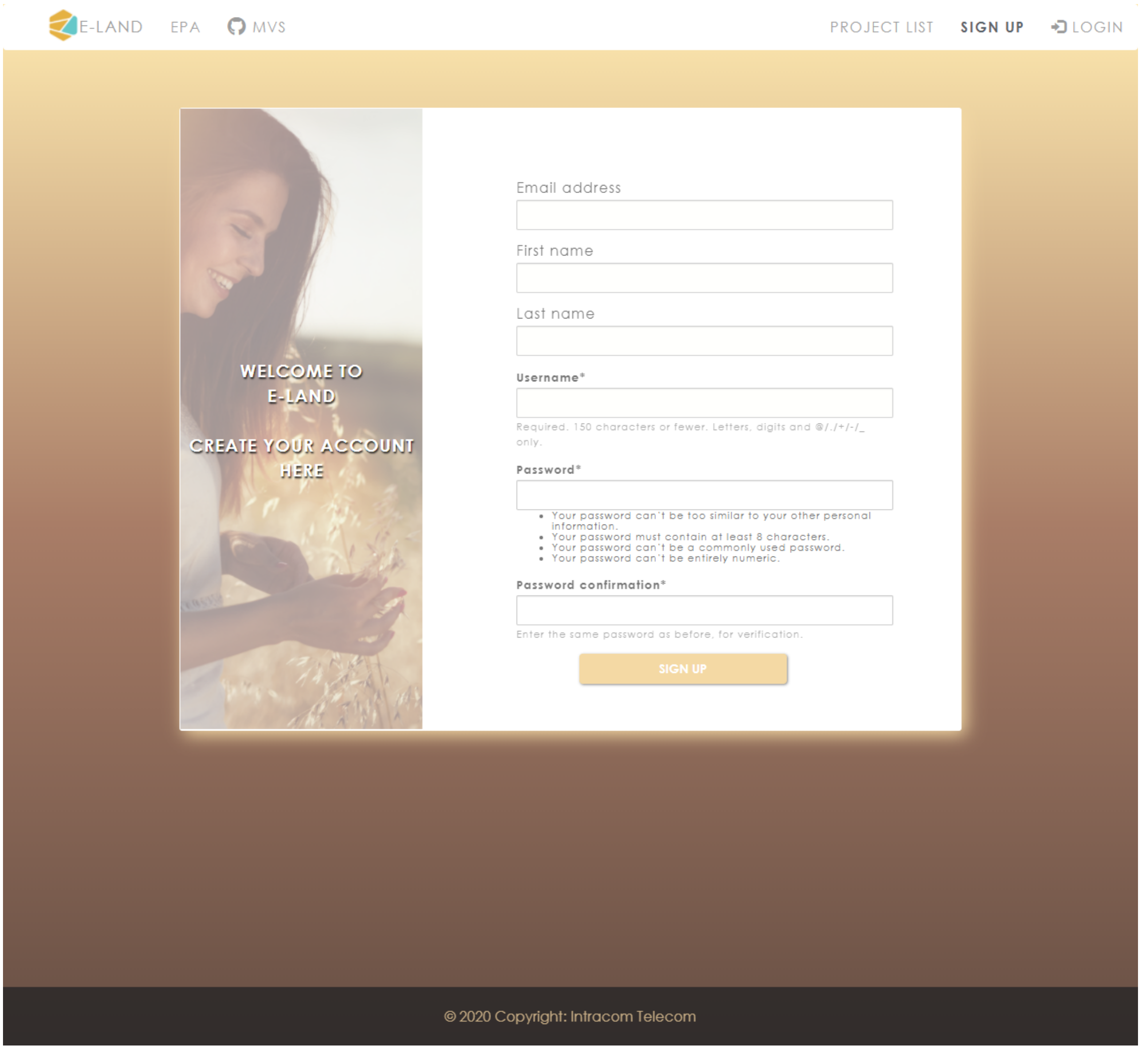

Figure 7 EPA: Signup Page

Once the user successfully logs-in, is redirected to *Projects list* (Home) page as shown in Figure 8. The navigation bar, placed on top of each page, provides quick access links to:

1. E-LAND project landing page;
2. EPA home page;
3. MVS GitHub repository [4];
4. The list of projects, to which the user has access to;
5. User specific menu (user info and password change);
6. Logout functionality.

The user's projects (if any), are presented in a list and their location displayed on a map as well. End users have full access to create, edit, view, open and delete their own projects. The edit and show buttons shown in project list page navigate to corresponding pages containing specific project data.

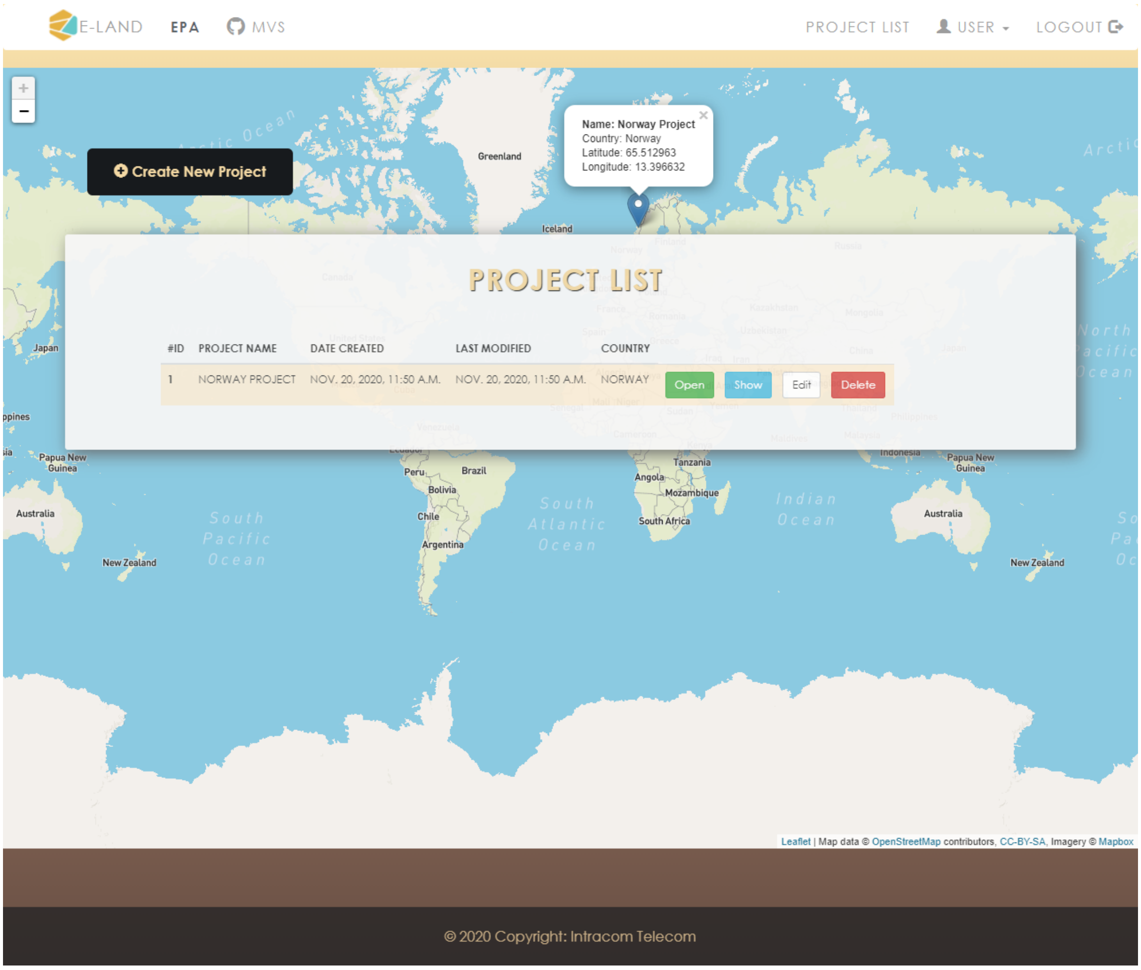

Figure 8 EPA: Projects List

Figure 9 illustrates the project creation page. Apart from the required fields, an interactive map aids in the longitude and latitude input filling procedure.

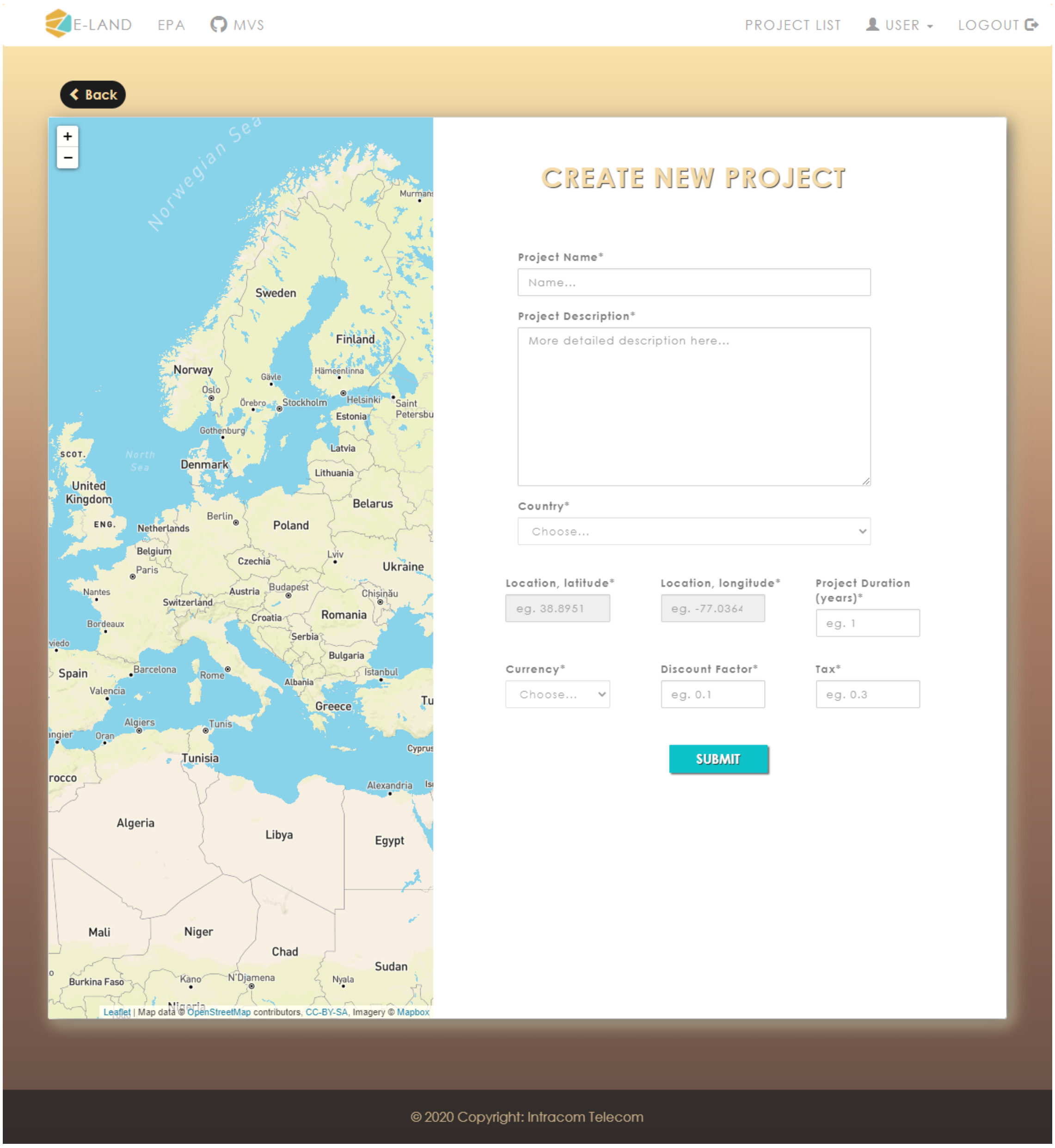

Figure 9 EPA: Project creation Page

After a user defines a new project or edits an existing one, it is possible to navigate to the project's scenario list by clicking on the "Open" button next to each project in the project list page. A new page appears, a list of all project's scenarios (Figure 10) in one tab and another list of all project comments (Figure 11).

Along with the list of all scenarios, Figure 10 shows the available options for each scenario. More specifically, a user can *review* the scenario information, *visualize* scenario simulation results, *define the model* of the local energy system, *run the simulation* of the defined model (communication with MVS), copy a selected scenario (which can be later modified) - and *remove* (delete) the scenario from the list.

Similar to the scenario list, the user can add, edit and delete comments to the project.

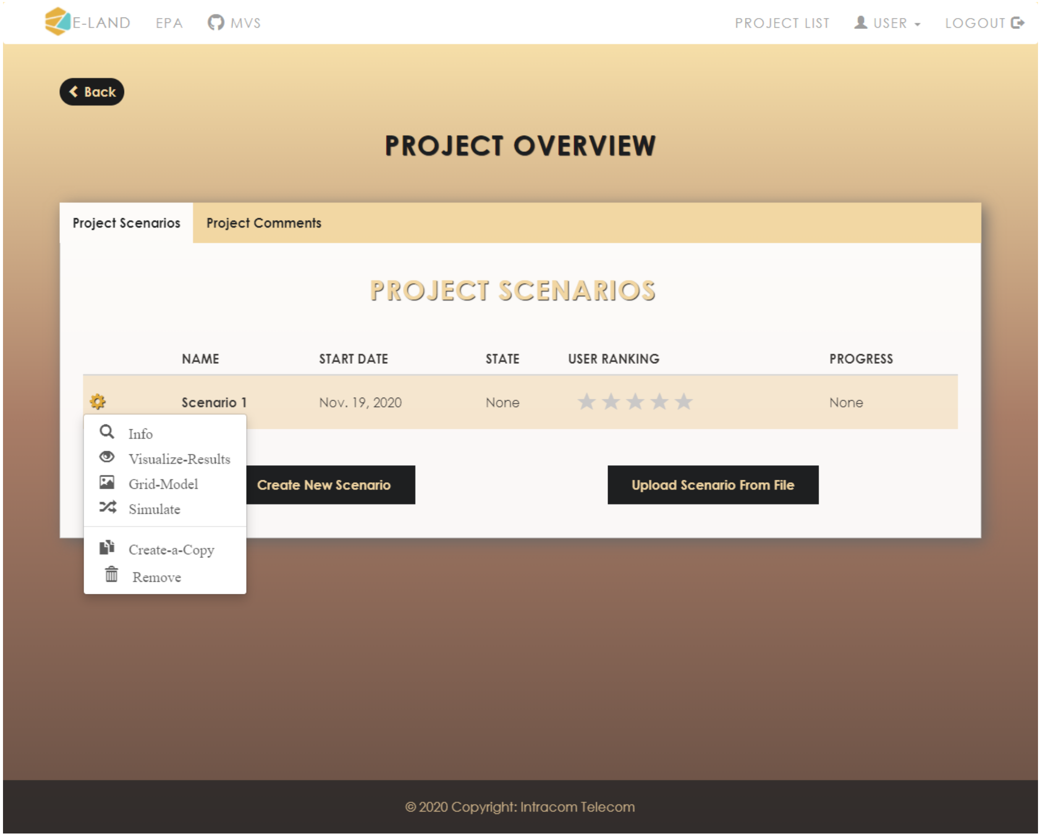

Figure 10 EPA: Scenario List

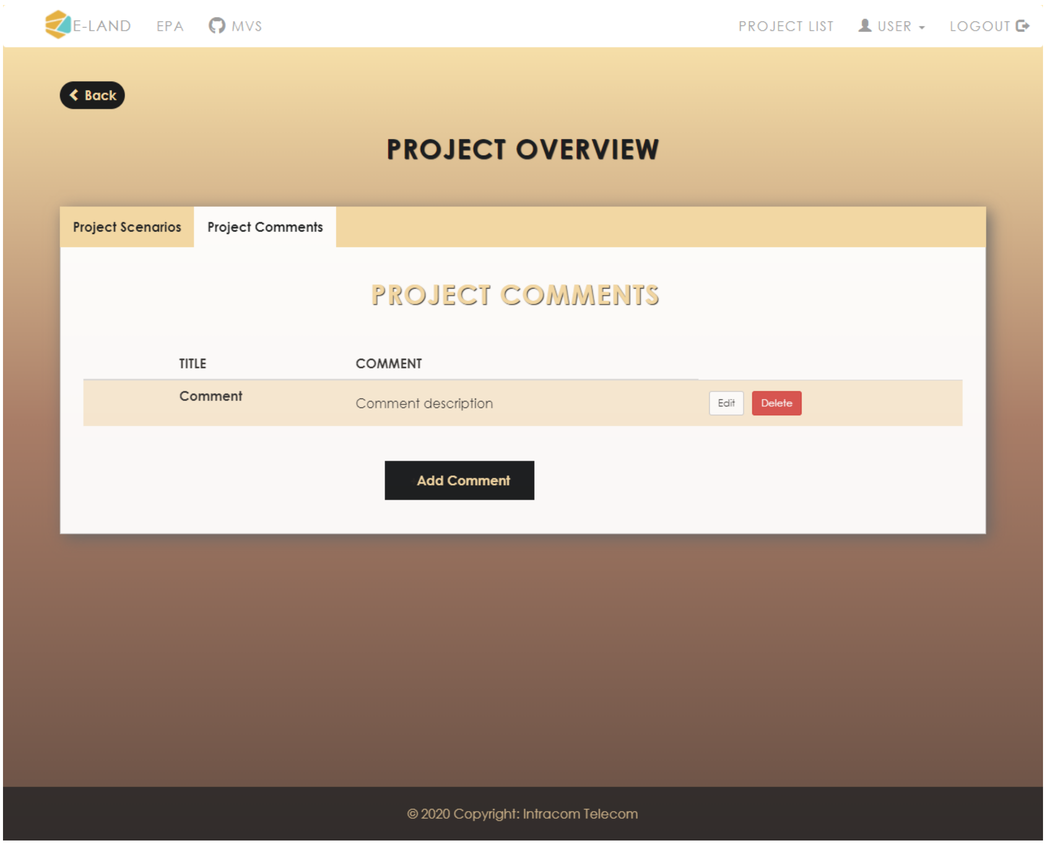

Figure 11 EPA: Comment List

In order to create a new scenario, the user clicks on the “Create New Scenario” button and the modal of Figure 12 fades in. In case a user has a locally created, valid scenario file, it is possible to load file data and create a new scenario from file, using the modal shown in Figure 13.

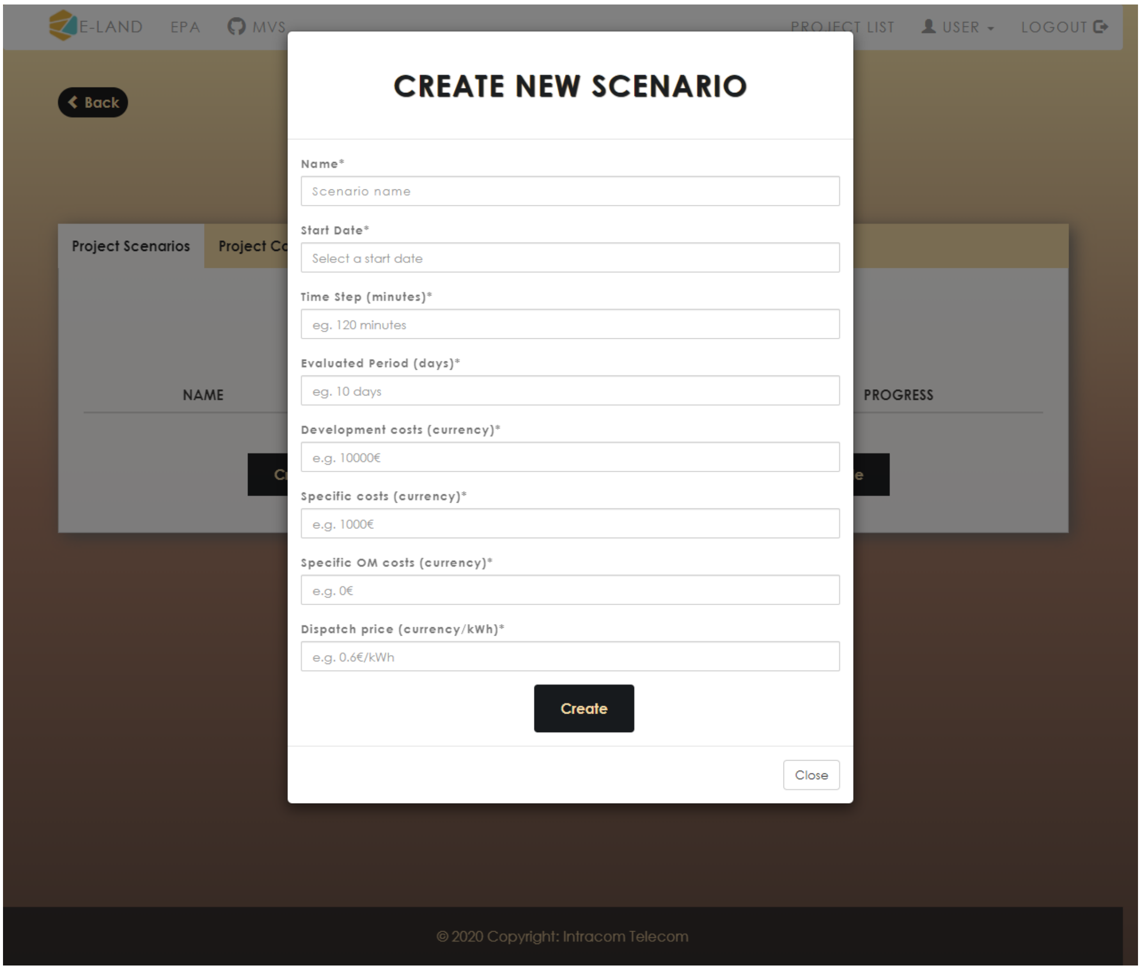

Figure 12 EPA: Scenario Creation

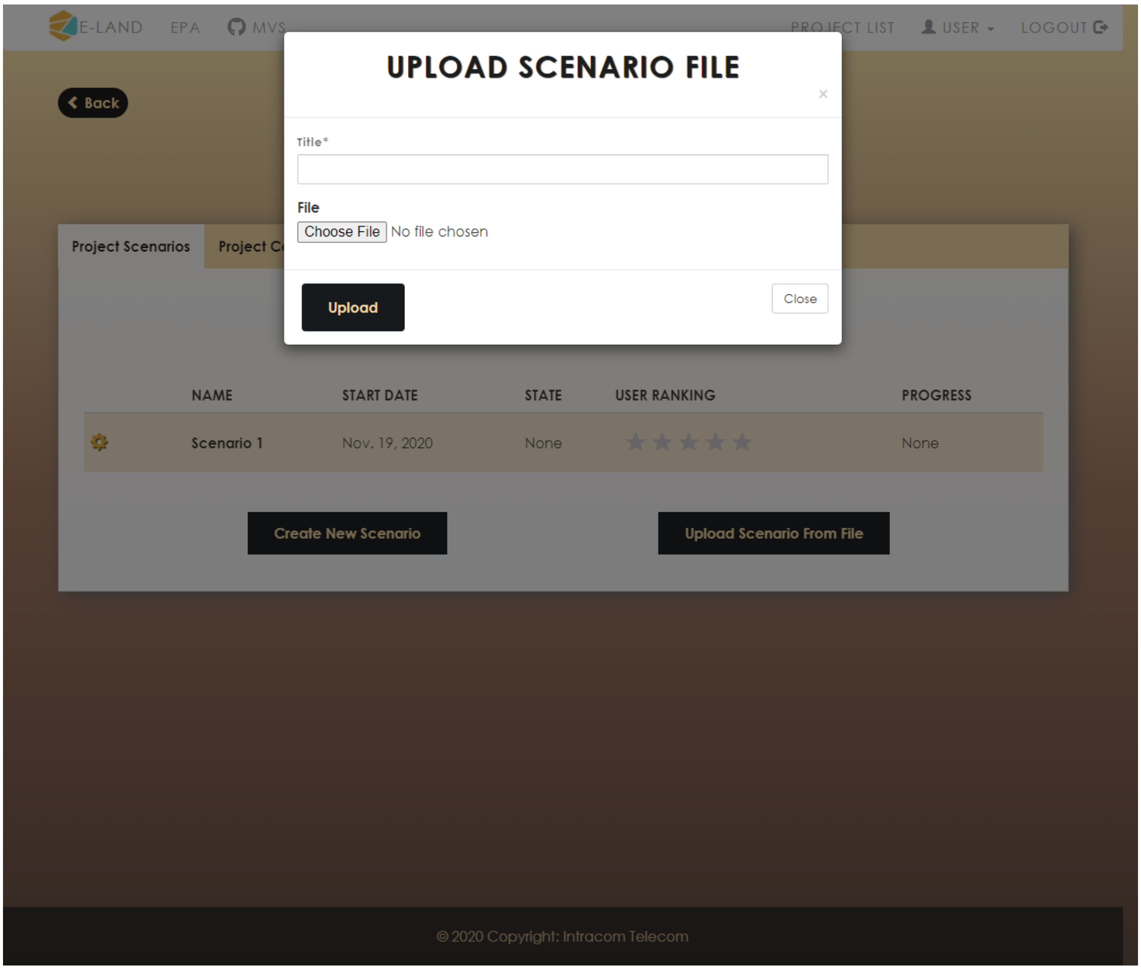

Figure 13 EPA: Import Scenario from file

Following the scenario creation step, the user has to define the model of the desired energy system. This is the page to define all the assets of the simulated energy system. There are 4 major asset types, *production, conversion, storage and consumption*. A number of different assets are listed under each type category. Apart from the assets, energy busses are also defined in the grid model. A rule which needs to be respected is that the interconnection of assets can only be allowed through a bus.

Assets and busses are represented as nodes in the grid model. For the creation of the model, an intuitive drag and drop editor is utilized. To set the properties of a single asset, the user needs to double click on the asset. After this action a modal will appear to fill-in specific asset data (Figure 15). Once the user has finished with the definition of the grid model, the “Save” button needs to be pressed to save model information to EPA’s database. In case the user needs to clear all defined assets, it is possible through the “Clear” button.

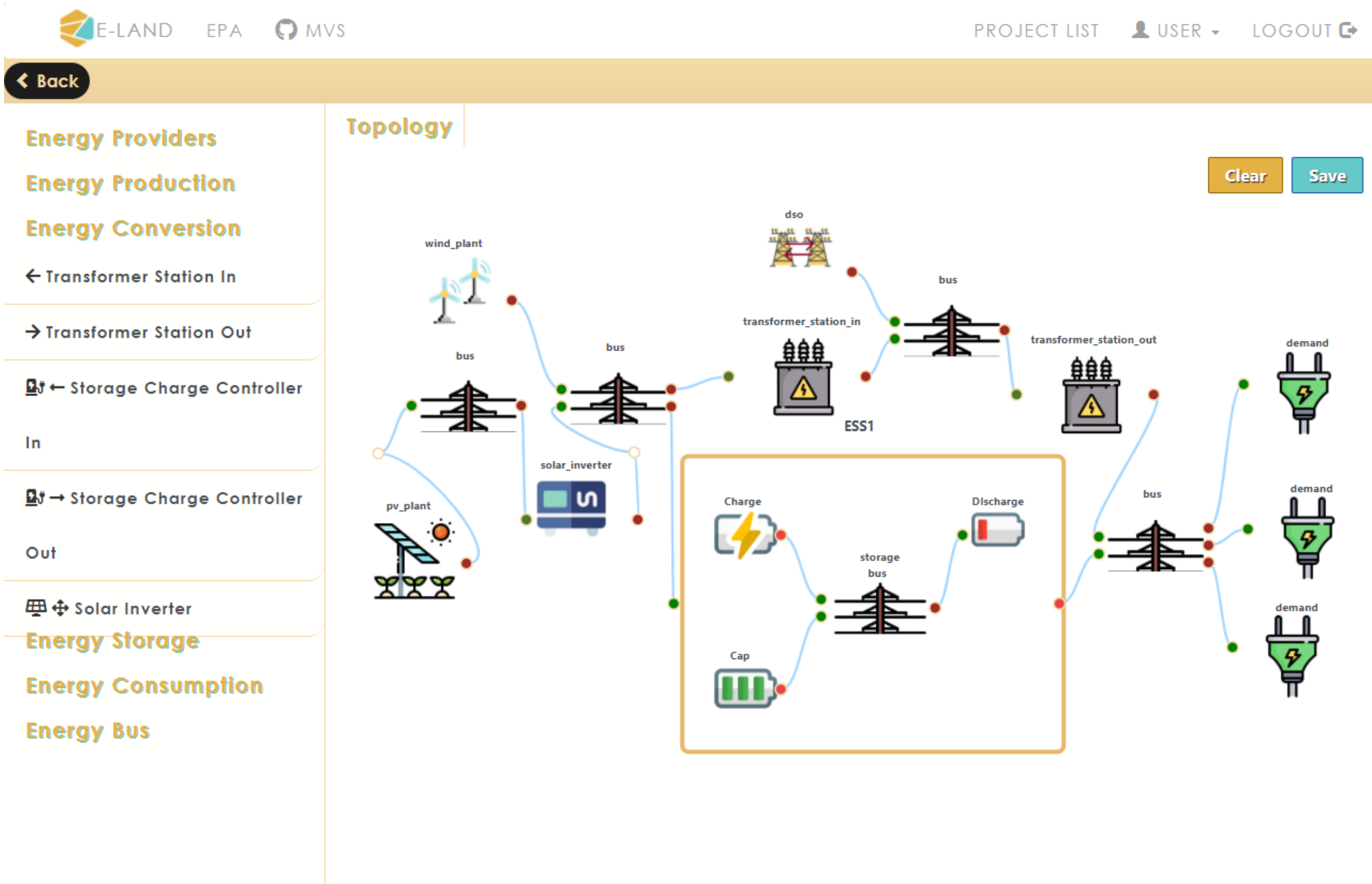

Figure 14 EPA: Grid Model Definition

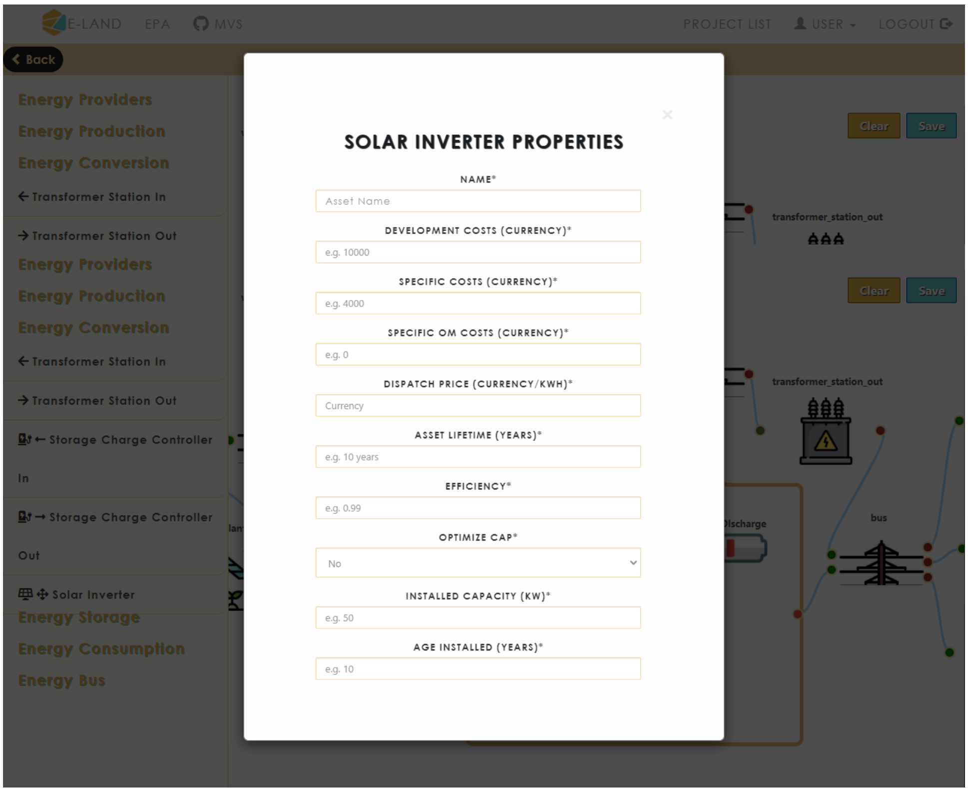

Figure 15 EPA: Asset specific data

Once the grid model is defined, the user can request a simulation from the MVS by clicking on the “Simulate” option shown in Figure 10. This is a non-blocking process, meaning the user sends the request and can continue on other desired task on the application (e.g. create/edit other scenarios).

Once the simulation results are ready the “Visualize-Results” option shown in Figure 10 gets enabled. Figure 16 shows the simulation results dashboard. On top of the dashboard, the user may observe pie charts presenting economic metrics related to the studied scenario’s investment. Following the visual representation, various economic KPIs are provided in table format, concerning the investment as a whole, such as: the upfront investment cost, the total lifetime investment cost and the annuity maintenance cost. The table also provides more technical indexes of the analyses, such as: the total renewable impact (share in demand supply), the total electricity demand and the internal renewable generation. The definitions of the KPI is available on the *readthedocs* [17] page of the MVS.

At the bottom of the dashboard, the user may observe the timeseries of operation for each assets in the LES and for each energy vector (e.g. electricity, heat and gas, H2).

For comparison purposes it is possible to add the timeseries of multiple assets - as long as they belong to the same energy vector - in the same chart.

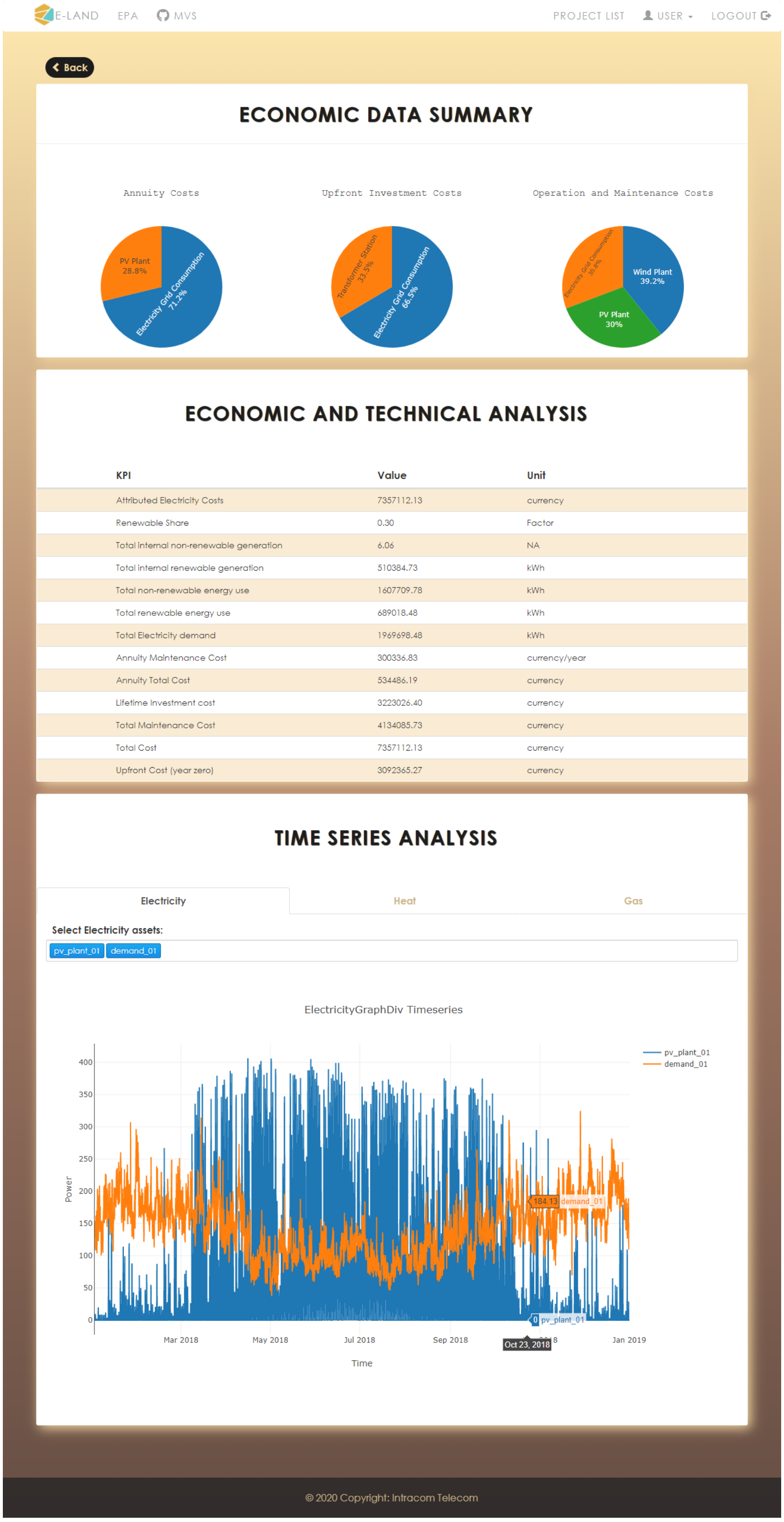

Figure 16 EPA: Scenario Simulation Results

4 Data Visualisation Application (DVA)

4.1 Overview

DVA solution enables the visualisation of the project's KPIs and operation of local energy assets, providing insights on the operation of the pilots (local energy systems where the E-LAND solutions are validated) to the pilot owners (end-users) and the local community, as well as third parties interested in the project's results through the project's dissemination activities.

A list of the KPIs that will be monitored via this application follows:

- Percentage reduction in CO2 emissions;
- Reduction in peak power;
- Percentage of demand covered by local renewable generation;
- Percentage reduction in electric energy from grid;
- Increase in storage usage efficiency;
- Extended lifetime of battery storage.

The details for the calculation of the above indexes are documented in deliverable D5.3 [1]. DVA interfaces the toolbox's ESB to get access to the data needs for the KPIs' calculations.

4.2 Design

DVA utilizes a set of technologies for providing reliable and high QoE to the end user:

- The web application core is implemented in *Django* [6];
- *MySQL* [7] is the selected Database Management System (DBMS), in which all required data, KPIs and pilot specific information such as asset data, are stored;
- *Bootstrap 4, Material Design for Bootstrap (MDB)* [17], *JQuery* [9], *Selectize* [11], *Charts.js* [18] and *JQuery-UI* [14] are the major JavaScript libraries used either to achieve specific front-end functionalities or to enrich the appearance of the application to the end user;
- *Nginx* [15] is utilized as a reverse proxy and load balancer.

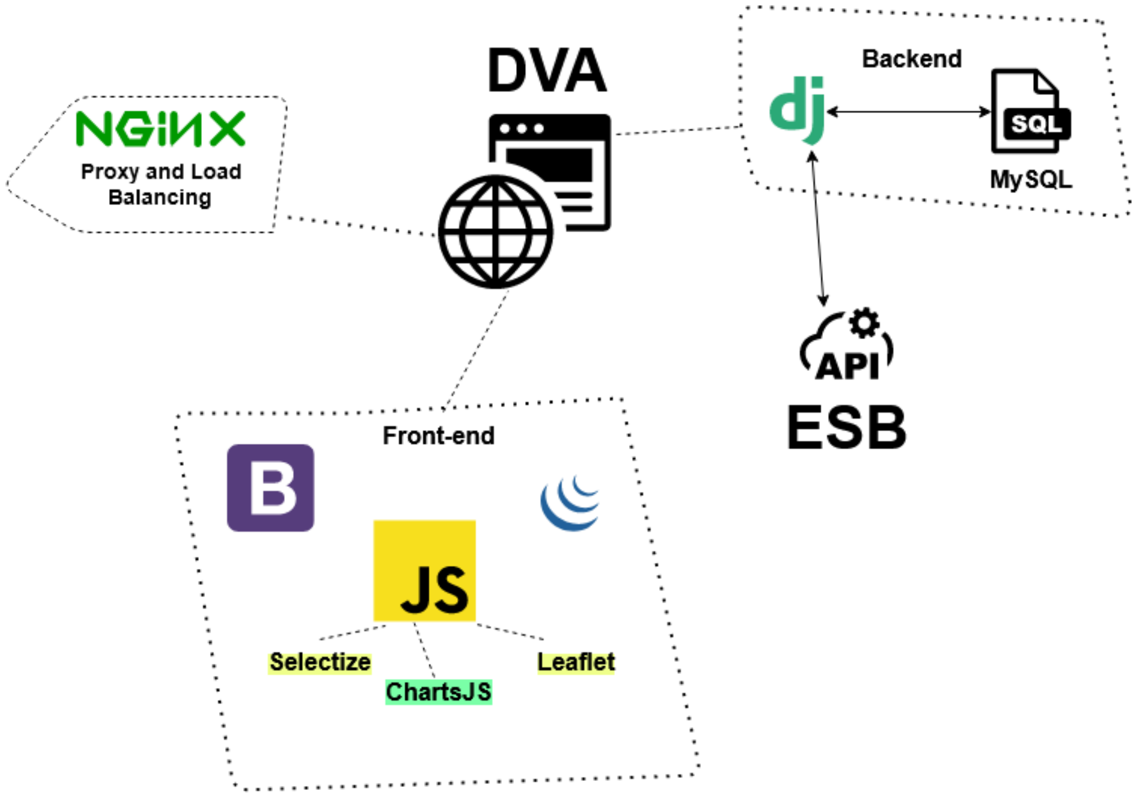

Figure 17 DVA Technology Dependency Stack

Django, MySQL, Selectize, JQuery-UI and *Nginx* are already described in the EPA design section (§3.3) of this document.

4.2.1 Bootstrap 4

Bootstrap 4 [19] is the successor of *Bootstrap 3*, described above (§3.3.6). It is the current most up to date version of the framework. *Bootstrap 4* offers a number of new features to the existing rich framework, such as cards, flexbox and fully responsive tables out of the box.

4.2.2 Material Design for Bootstrap

Material Design for Bootstrap (MDB) [17] is an open source toolkit based on Bootstrap for developing apps following the Material Design idea for user experience, leveraging HTML, CSS and JS. This framework is a third party wrapper to *Bootstrap* that provides such features and enhanced styling to DVA's front-end.

4.2.3 Chart JS

Chart.js [18] is a free open-source JS library for data visualization, which supports eight chart types: bar, line, area, pie (doughnut), bubble, radar, polar, and scatter. *Chart.js* renders in HTML5 canvas and is available under the MIT license. Time series, pies and other chart-based visualizations shown in the simulation results visualization page are created using this library.

4.2.4 Moment.JS

Moment.JS [20] is a widely used JS library which supports out of the box parsing, validation, manipulation and display of dates and times in various formats. The library has no external dependencies, a feature which makes it even more attractive and straightforward to utilize. It is licenced under the MIT license and entirely open source.

4.2.5 Leaflet

Leaflet [21] is an open source (BSD 2-Clause) JS library designed for mobile-friendly interactive maps development. Among other features, *Leaflet* offers various map controls to interact with, multiple layers addition flexibility, visual customization features and performance features. DVA utilizes the library to present pilots and their relevant assets positioned on a map.

4.3 Guidelines of use

DVA enables the visualisation of the project's KPIs and asset monitoring data through a dashboard interface. The user is navigated through a homepage to project aggregated results and pilot specific results. The workflow of the DVA application is illustrated in Figure 18.

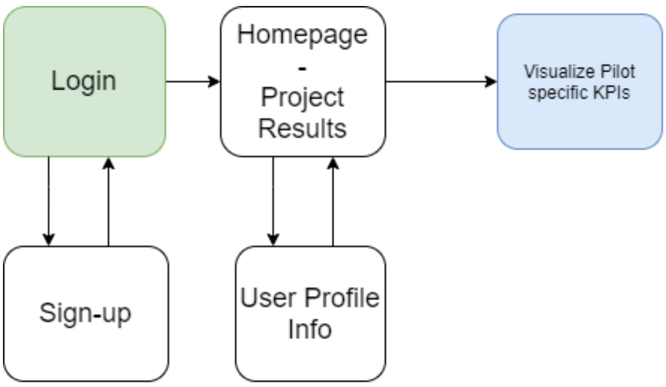

Figure 18 DVA: Workflow

Once the user navigates to the DVA, the web app shows the Homepage of the application with the list of all pilots of the project. This page is the only one accessible to non-authenticated users. In order to acquire higher access privileges and be able to navigate to all application pages, the user needs to login using personal credentials as shown in Figure 19 DVA: Login Page below.

If the user has no login credentials, the application asks the user to register an account. The signup page is presented in Figure 20. In order to complete the registration successfully, the user is required to verify the provided email address.

Once successfully logged in, the user can view and edit profile information by following the profile link at the top left of the sidebar (see Figure 21).

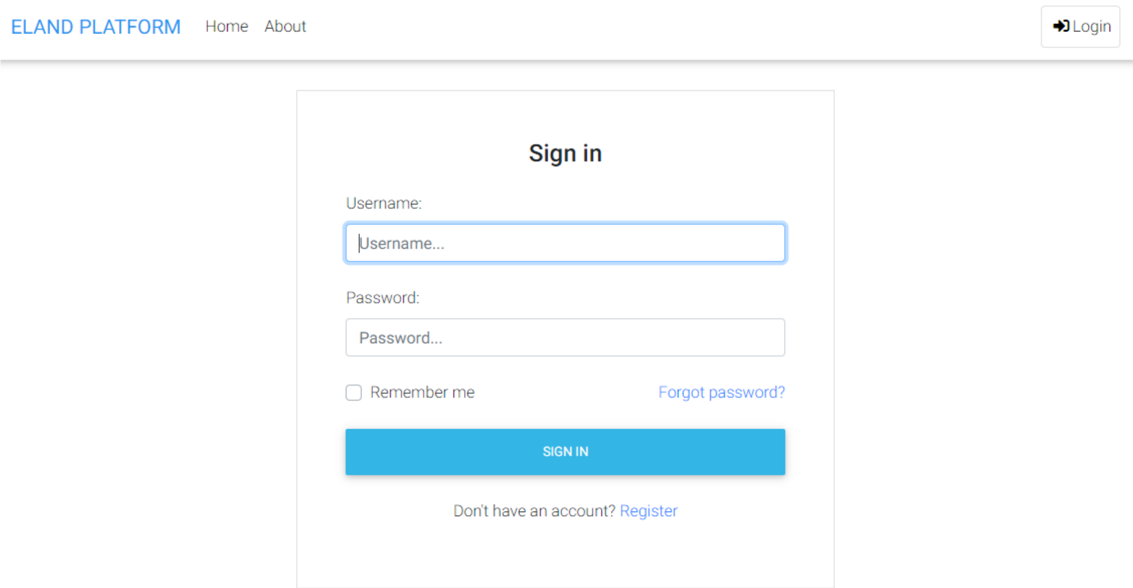

Figure 19 DVA: Login Page

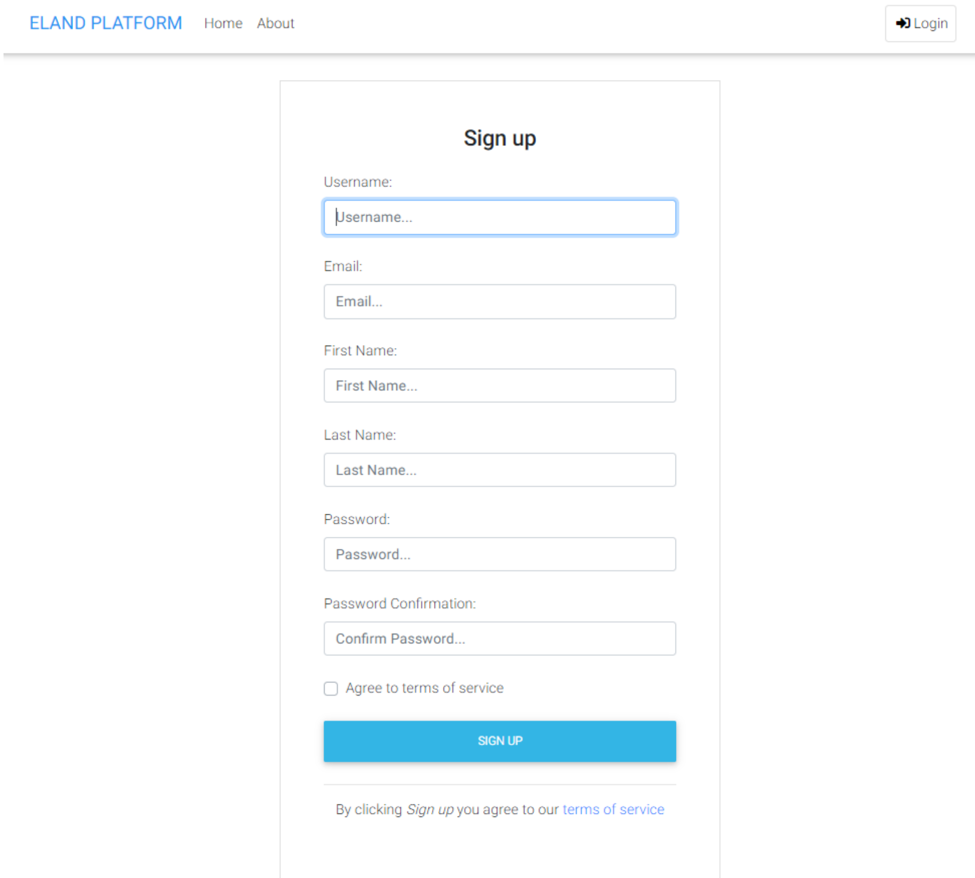

Figure 20 DVA: Sign-up Page

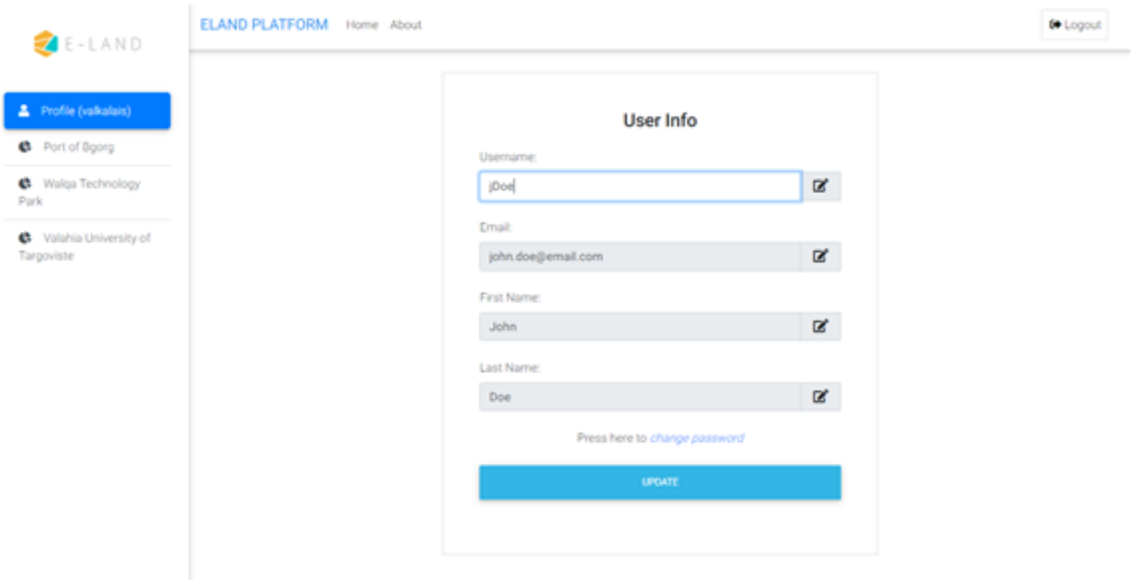

Figure 21 DVA: User Profile Information Page

The Projects List (Home) page is shown in Figure 22. The pilot sites that the user has authorization to view (if any), are presented in a list and a map visualization. The following information are presented about each pilot site:

- Name: pilot site descriptive name;
- Location: the geographical location of the pilot;
- Country: the country where the pilot is located;
- Latitude and Longitude: exact coordinates of the pilot site.

Additionally, a link corresponding to each of the pilot sites also appears on a sidebar to the left of the page, assuming the user has the relevant access rights. Using these links, the user can navigate to a page demonstrating pilot’s specific KPIs (i.e. Figure 25, Figure 26 and Figure 26). Finally, two bar charts are presented at the bottom of the page, illustrating the equivalent CO_2 emissions and the energy footprint on a pilot basis during the piloting phase lifetime.

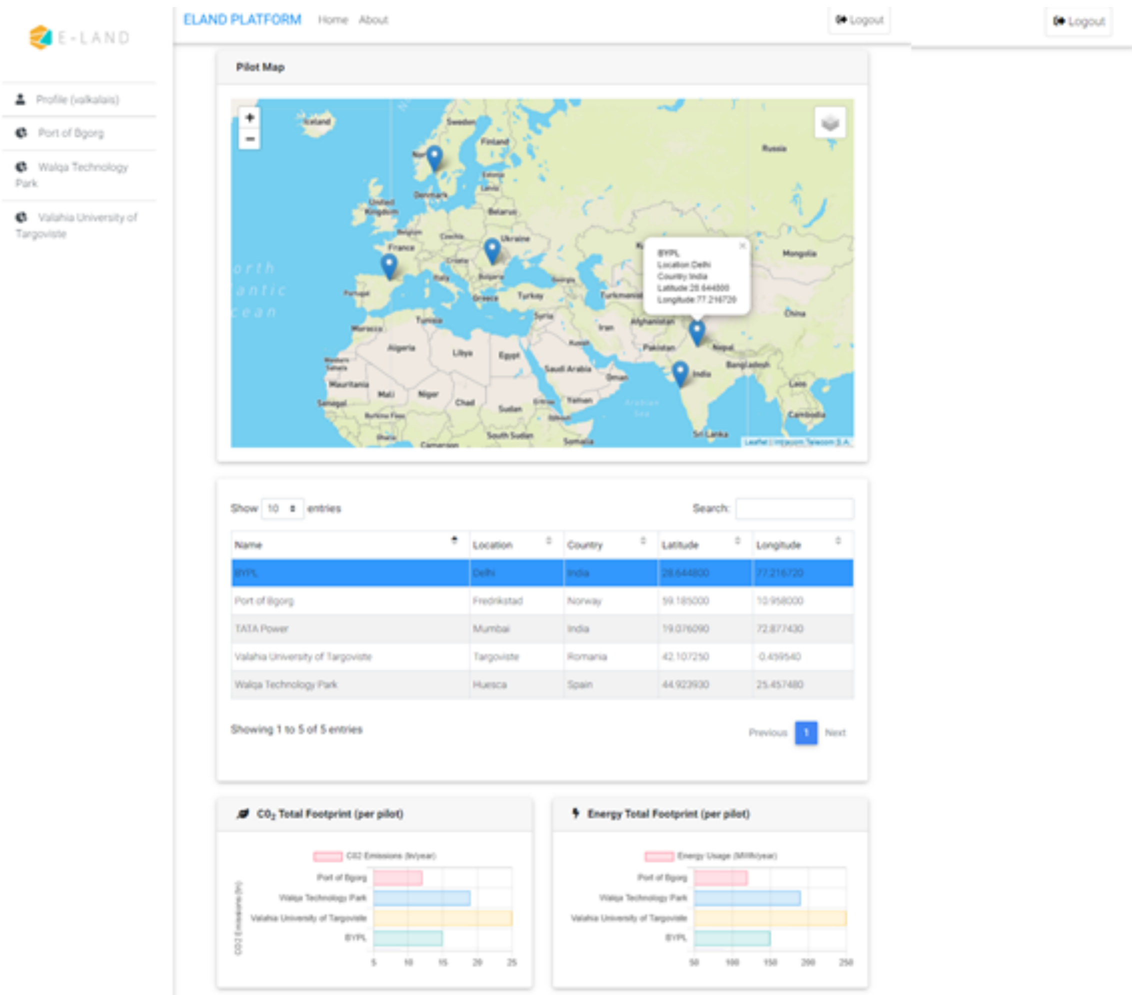

Figure 22 DVA: Pilots List

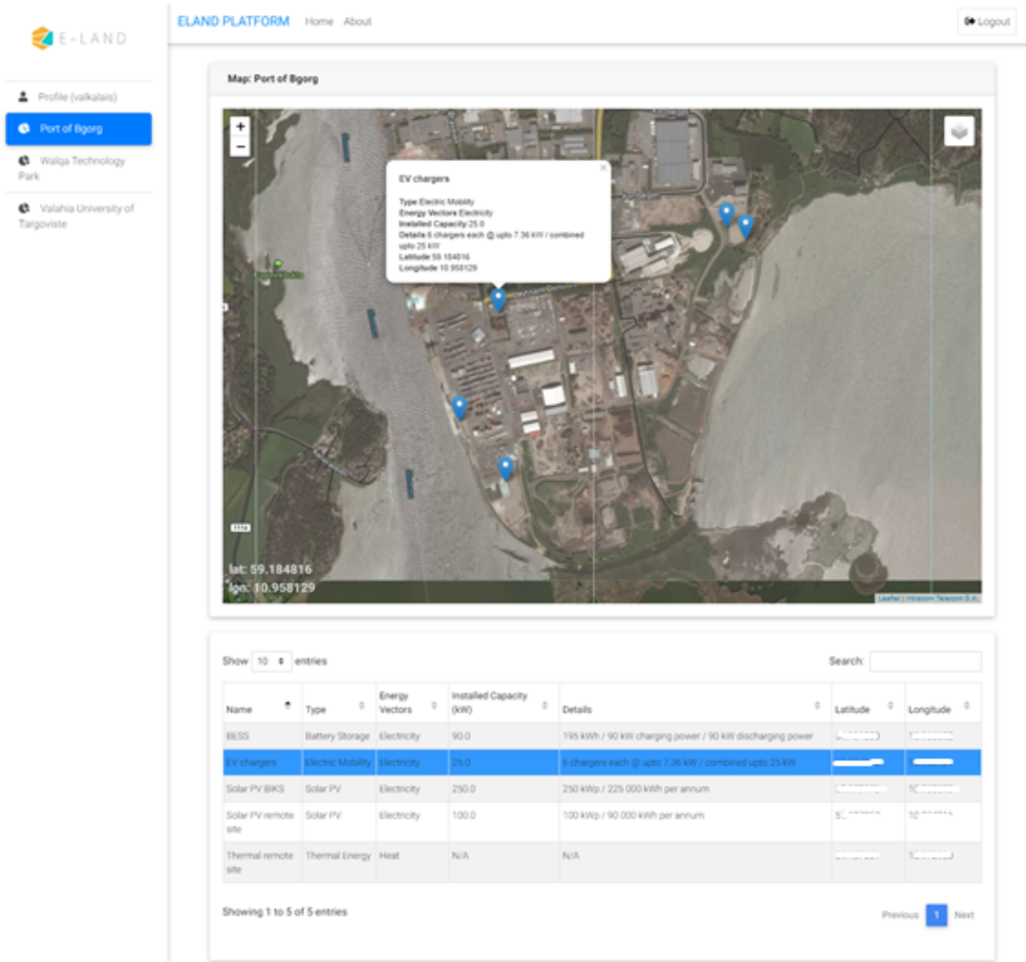

Figure 23 DVA: Pilot Dashboard Page

After the user selects a specific pilot site, the pilot site assets are presented in a list and a map visualization of their respective locations within the site zooms in to the selected asset as illustrated in Figure 23. The following information are presented about each asset:

- Name: the descriptive name of each asset;
- Type: the asset type (e.g. Electric Mobility, Solar PV, Battery Storage);
- Energy vector: the energy vector of the specific assets (i.e. heat, gas, electricity);
- Installed capacity: the maximum capacity of the asset;
- Details: a short description to better understand the asset specifics;
- Latitude and Longitude: coordinates of the asset in the map.

In the same page, pilot specific KPI visualizations are also included. Depending on their vertical position within the page view, the visualizations are presented the following three figures.

Figure 24 presents the following visualizations:

- A line chart depicting the pilot site CO_2 footprint for each energy vector as well as the total CO_2 footprint;
- A line chart depicting the pilot site power usage for each energy vector and in total;
- A doughnut chart illustrating the pilot site consumed energy mix for each energy vector;
- A radar chart illustrating the contribution of each asset to the total energy demand/generation of the pilot site.

Figure 24 Pilot specific KPI's part 1

Figure 25 groups the following visualizations:

- A bar chart depicting the percentage reduction in CO_2 emissions, the percentage reduction in peak power and the percentage reduction in total power consumption;
- A bar chart depicting the of energy demand covered by local renewable generation and reduction of the imported electric energy from the grid;

- A bar chart depicting the total number of cycles of pilot site battery storage and number of events where the battery temperature exceeds the range of predefined limits set for the pilot;
- A bar chart depicting the battery storage state of health percentage.

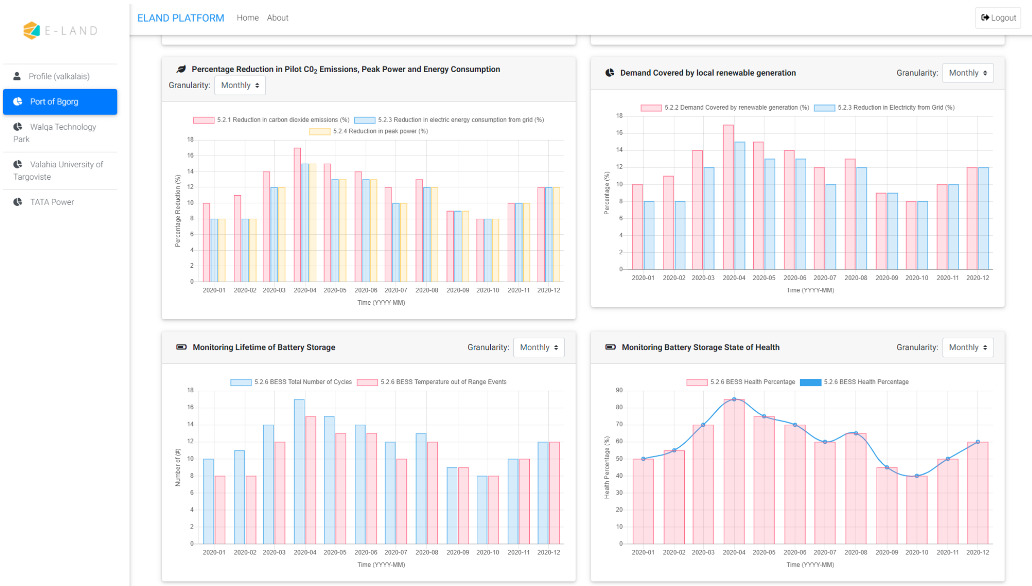

Figure 25 DVA: Pilot specific KPI's part 2

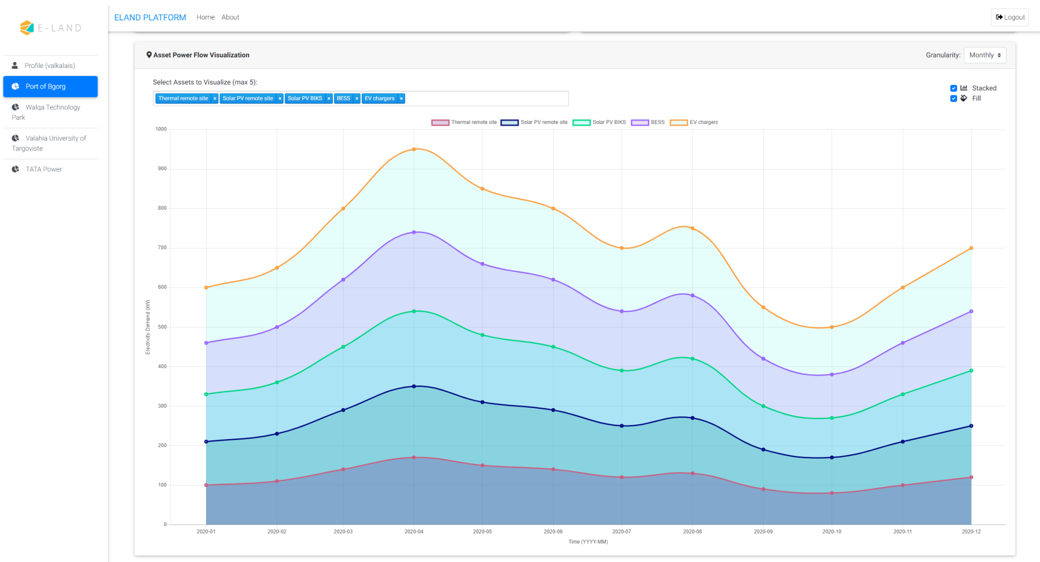

Figure 26 DVA: Pilot specific KPI's part 3

Finally, Figure 26 shows a line chart illustrating the power flow for any number of assets designated by the user. Negative power flow designates generation and positive power flow demand. The user may select any number of assets from the asset selection list to be presented.

5 End-user Assessment & Tool Evolution

The tools documented in this report, the Energy Planning Application (EPA) and the Data Visualisation Application (DVA), are addressing on one hand the needs of the project partners for assessing future investments in the LES and performance impact of E-LAND's optimization processes (respectively); on the other hand they are aiming a wider audience.

Following the paradigm of MVS, the EPA prototype will be under an open-source license enabling a wide audience to use, assess and evolve the tool; offering the energy communities with an easy to use solution to plan their future steps for achieving their sustainability goals.

The DVA exposes the impact of the project in the local community of the LES, providing the opportunity to individuals "outside" of the project to assess the performance of the actions under the umbrella of ELAND to the local energy system.

Towards facilitating the evolution of these extrovert tools, interactive workshops will be realised in the next phase of the project in order to fine-grain the prototypes through co-creation with the various stakeholder in the project ecosystem. Following this "maturing" process, the Second E-LAND Toolbox workshop is scheduled for M31 (June 2021) towards sharing the outcomes of this work, facilitating collaboration and gathering useful feedback for the tools' evolution and future reusability.

6 Conclusions

This report documents a high-level implementation view of the Energy Planning Application (EPA), a web application providing a user interface to users interested in investment or operational planning for local energy assets in the context of a LES; and the Data Visualisation Application (DVA), a web application that enables the visualisation of the project's KPIs and operation of local energy assets, providing insights on the operation of the pilots (local energy systems where the E-LAND solutions are validated) to the pilot owners (end-users) and the local community, as well as third parties interested in the project's. It also documents a quick user guide for each application.

Following an integration testing period, the solutions will be deployed in a production environment and shall be used for monitoring the performance at each pilot site during the piloting phase of the project in the context of WP6 – Piloting, Validation and Replication Guidelines. During this period the tools will evolve based on the feedback of the various end-users. Furthermore, they will be used for disseminating the results of the project both at a local community level, but also to a wider audience.

7 Bibliography

[1] E-LAND, “D5.3 Pilot and System Implementation Plan,” 2020.

[2] E-LAND, “D3.2 Functional and Operational requirements,” 2019.

[3] E-LAND, “D3.3 Technical Specification,” (restricted access), 2019.

[4] “https://github.com/rl-institut/multi-vector-simulator,” [Online].

[5] “https://oemof.org,” [Online].

[6] “https://www.djangoproject.com,” [Online].

[7] “https://www.mysql.com,” [Online].

[8] “https://getbootstrap.com/docs/3.3,” [Online].

[9] “https://jquery.com,” [Online].

[10] “https://github.com/jerosoler/Drawflow,” [Online].

[11] “https://github.com/selectize/selectize.js,” [Online].

[12] “https://plotly.com/javascript,” [Online].

[13] “https://github.com/sweetalert2/sweetalert2,” [Online].

[14] “https://jqueryui.com,” [Online].

[15] “https://www.nginx.com,” [Online].

[16] “https://github.com/django-crispy-forms/django-crispy-forms,” [Online].

[17] “https://mdbootstrap.com,” [Online].

[18] “https://www.chartjs.org,” [Online].

[19] “https://getbootstrap.com/docs/4.5/getting-started/introduction,” [Online].

[20] “https://momentjs.com,” [Online].

[21] “https://leafletjs.com,” [Online].