



(FP7 614100)

D8.1 Specification of Proof of Concept Applications

01 Aug 2014 – Version 1.1

Published by the IMPReSS Consortium

Dissemination Level: Public



**Project co-funded by the European Commission within the 7th Framework Programme and
the Conselho Nacional de Desenvolvimento Científico e Tecnológico
Objective ICT-2013.10.2 EU-Brazil research and development Cooperation**

Document control page

Document file: D8.1_Specification_of_Proof-of-Concept_Applications.doc
Document version: 1.1
Document owner: Djamel Sadok (UFPE)

Work package: WP8 – Platform Evaluation and Application Development
Task: T8.1 Specification of Proof of Concept Applications
Deliverable type: R

Document status: approved by the document owner for internal review
 approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Lucas Lira Gomes (UFPE) Vinícius Fraga (UFPE)	10/02/2014	First Draft.
0.2	Lucas Lira Gomes (UFPE) Vinícius Fraga (UFPE)	21/02/2014	Created the Application Requirements section and revised the document.
0.3	Eduardo Souto (UFAM)	14/03/2014	Document sent for internal review.
0.4	Lucas Lira Gomes (UFPE)	18/03/2014	Addressed the concerns of the reviewers.
1.0	Djamel Sadok (UFPE)	18/03/2014	Final version submitted to the European Commission.
1.1	Lucas Lira Gomes (UFPE)	01/08/2014	Corrected some minor wording issues at the summary.

Internal review history:

Reviewed by	Date	Summary of comments
Sebastian Franken (FIT)	17/03/2014	Accepted with minor corrections and comments.

Legal Notice

The information in this document is subject to change without notice.

The Members of the IMPReSS Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the IMPReSS Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Possible inaccuracies of information are under the responsibility of the project. This report reflects solely the views of its authors. The European Commission is not liable for any use that may be made of the information contained therein.

Table of Contents

Executive summary 4

Introduction 5

 Purpose, context and scope of this deliverable5

1. Application Requirements..... 6

2. Scenario-based Descriptions 9

 2.1 Scenario 1: Adding New Areas 10

 2.2 Scenario 2: Areas’ Monitoring and Management 10

 2.3 Scenario 3: Adding New Electronics 12

 2.4 Scenario 4: Electronics’ Monitoring and Management 12

 2.5 Scenario 5: Adding Electronics to an Area 14

 2.6 Scenario 6: Adding New Groups 15

 2.7 Scenario 7: Groups’ Monitoring and Management 16

 2.8 Scenario 8: Adding New Simulations 16

 2.9 Scenario 9: Simulations’ Monitoring and Management 18

Summary and Conclusions..... 19

References 20

Executive summary

This deliverable describes the initial proposal for the specification of the proof-of-concept applications leveraging on the Impress cloud for the Teatro Amazonas and the UFPE campus. The approach is based on a mobile application, aiming to operate as a smart energy management device, which can access the IMPRESS cloud in order to obtain information about the devices connected to the cloud. The main goal is to give the users of the application a big picture of what is happening through a single interface, providing analytics and visualisations that can facilitate the decision process based on the available data in the IMPRESS cloud.

Since both scenarios are targeting smart energy management, we could use the same core technology to develop both applications, one for UFPE campus and other for Teatro Amazonas.

Despite that, we organise this deliverable as follows. First, section 2 contains the application requirements. Section 3 contains the scenario-based descriptions for an Internet of Things scenario. Finally in Section 4, we provide our summary and conclusion.

Introduction

Purpose, context and scope of this deliverable

There is a fast-growing need for more efficient use of energy, especially in countries that rely on energy sources that contribute more to pollution of the environment, e.g. coal and diesel powered thermoelectric power plants. Newly constructed buildings typically incorporate energy efficient facilities and equipment; however, those solutions do not provide a single interface for understanding their impact on the energy supply as a whole.

Besides that, there are less user dependent solutions. Smart grids, for instance, demand energy providers to significantly change the energy grid infrastructure. The smart energy grid could reason about the behaviours of suppliers and consumers in an automated fashion to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity.

In that context, the Impress platform, entailing both the cloud and its application, can be seen as an in-between solution. Providing analytics that are commonplace in smart grids, but with appeal to energy consumers requiring stricter control and understanding of their energy consumption as well. Especially because the Impress platform does not intend to be hardware-vendor specific, it can both be an intermediate step before smart grids large scale adoption or a foundation for energy providers during their own smart grid implementation process.

Finally, this deliverable aims to provide high-level specifications including requirements gathering and scenario-based descriptions. This helps the application development part in the WP8 tasks of the Impress platform, focusing on smart energy management, being carried out on a sound basis.

1. Application Requirements

As we are at the beginning of an implementation process, we limit ourselves to not specifying an extensive set of requirements formally. We rather propose guidelines to ensure our project vision is being met. This allows us to constantly update the application to be released through an iterative design process, integrating our stakeholder groups. The constant adjustment of the application to the users' needs makes it possible to acquire domain knowledge during the work on the application. Nevertheless, the proposed guidelines serve as a foundation for the development of the energy management and monitoring application that will demonstrate the IMPRESS cloud capabilities.

With these considerations in mind, we are envisioning the following for the application. First and foremost, the application must be capable of interacting with the Impress Cloud via a public API as depicted in Figure 1, so that the application can query the data available and access the Impress cloud modules and services. That API, however, is not defined at the time this deliverable is written; it is expected to be the output of Task 8.2.

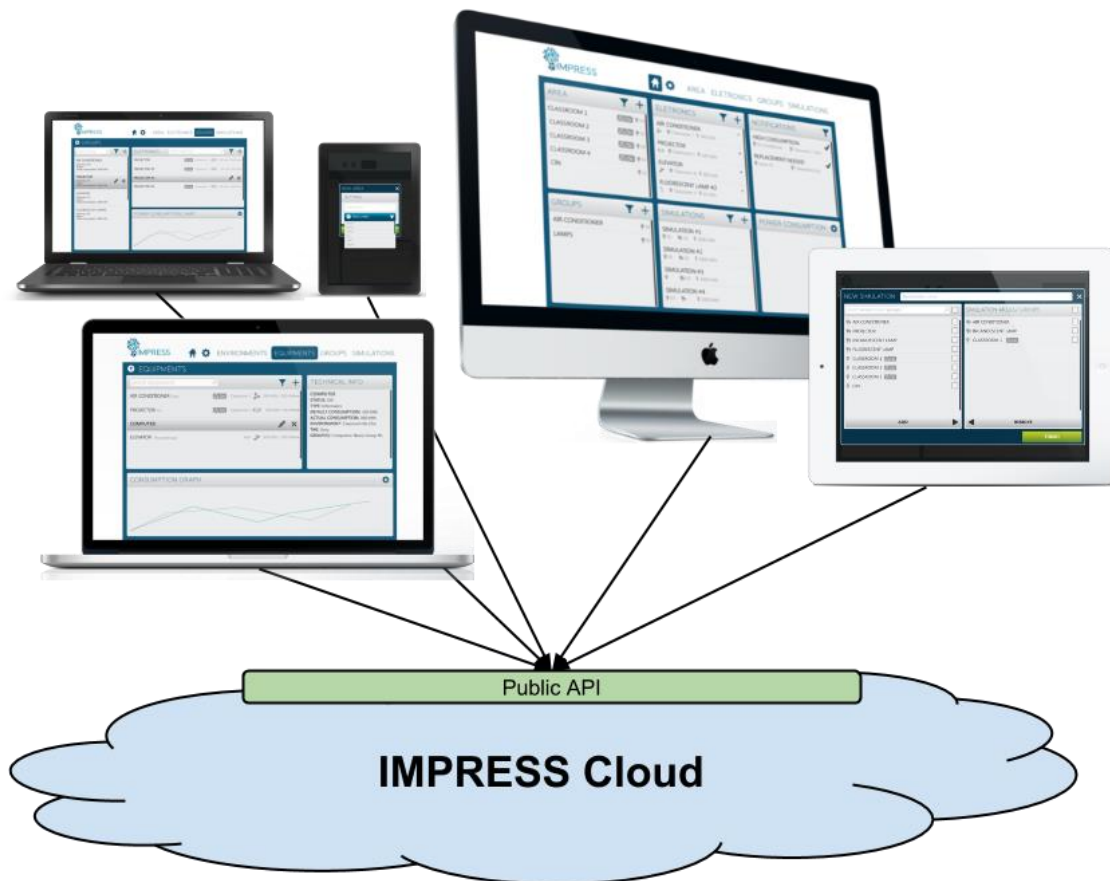


Figure 1: IMPRESS cloud integration with end users' applications.

To facilitate the management of the electronics being monitored, we propose two different organisational concepts: area and group. The former refers to the devices distribution on the spatial representation of the on-site facilities, while the latter does not have this limitation, it rather refers to a logical organization of devices. The reasons for having both of them can be justified by two major use cases. Areas are hierarchically organised as an acyclic tree, and that can be a great time saver for those who use the application. By nesting areas into other areas, power consumption charts for a given area can automatically include the consumption of the areas nested inside them

without having to manually manage that. Groups differ from that by being independent units by themselves. Their usefulness lies in the fact that one may want to put all light bulbs of a specific type into one group instance, independent of their location. Therefore it is possible to see consumption charts of all electronics of a certain type, as if it was an area, but without the nesting capability. Also, it is noteworthy that areas and groups are the cornerstones of the simulation feature, since one select areas and/or groups can be part of a simulation as a means to facilitate the process of choosing a set of devices and elements to be simulated.

The application should be also a simulation platform for end users. Otherwise, the application would be limited to just display current and previous energy consumption patterns. But what if all the light bulbs were changed to a more energy efficient alternative, without having to actually buy and deploy them to test? How would that change affect the current, past and future energy consumption patterns? Or, likewise, what if the current HVAC system employed is turned off during a certain time range during weekends? Having to actively turn off/on a system or to buy and install new electronic and wait to check its impact in the energy consumption is not the best approach. Those in charge of maintaining the energy infrastructure should be able to test their assumptions in a more straightforward way and, preferably, without having the time and deployment overhead factor. Note that simulations does not differentiate deployed electronics, used in production, from fictional ones, created just for the purpose of testing via simulations. It is up to the end user to select which devices will be part of a simulation, by selecting a set of groups and/or areas or by selecting electronics individually.

The application should have a notification system, so that the IMPRESS cloud could notify users about eventualities. An eventuality, for instance, could be a electronic with a current consumption profile that is very different than the expected considering its theoretical consumption, provided when adding new devices or changing their configurations. Likewise, end users would probably like to be warned if, among electronics of the same category (e.g. light bulbs, PCs, heaters, TVs, etc), there are outliers in terms of energy consumption. This use case can be useful to reveal less energy efficient electronic that should probably be replaced by a newer counterpart. Another scenario would be to spot sensors that are malfunctioning, whether by providing measurements that do not make sense given the context (e.g. a device that consumed zero Watts when in operation) or a sensor that broke and therefore did not report its measurements for a while. As a result, the energy infrastructure personnel could quickly pinpoint which meters/sensors need to be replaced for the IMPRESS cloud to keep operating consistently.

Another concern regards the target operating system and device platforms for the application. Normal PCs (i.e. Windows, GNU/Linux and Apple Macs) are certainly not the only option here, especially in our context of an IoT era. Mobiles have already a strong presence in the market that cannot be overlooked. We therefore aim to guarantee multiple platform support by providing a tablet-like interface. Our point here is that tablets' applications can be used in a PC without much inconvenience regarding the shift of interaction workflow, whereas a smartphone resolution would not provide a good experience when ported directly to PCs' higher resolutions. So, given our focus on PCs and tablets for the sake of implementation and design simplicity, let us explore a bit mobile operating systems options. To that extent, we present Table 1, which was part of Gartner's 2013 press release on worldwide combined shipments of devices (e.g. PCs, tablets, ultramobiles and mobile phones). It is clear that Android surpasses their competitors, followed by Windows and iOS/Mac OS with numbers above two hundred million units sold.

Table 1:

Worldwide Device Shipments by Operating System (Thousands of Units)

Operating System	2012	2013	2014	2015
Android	503,690	877,885	1,102,572	1,254,367
Windows	346,272	327,956	359,855	422,726
iOS/Mac OS	213,690	266,769	344,206	397,234
RIM	34,581	24,019	15,416	10,597
Chrome	185	1,841	4,793	8,000
Others	1,117,905	801,932	647,572	528,755
Total	2,216,322	2,300,402	2,474,414	2,621,678


Source: Gartner (December 2013)

The support of the different mobile operating systems with the same code cannot be achieved through their standard SDKs. We would also have to deal with PC versions in separate, but to maintain many codebases can turn out to be too time consuming. Therefore, we plan to use the Qt framework [1] as it is a reasonable choice for solving this imbroglio. Currently, in Qt version 5.2, we can support Windows, GNU/Linux, Mac, Android, IOS and even Blackberry with virtually the same codebase. Additionally, it is important to mention that our team already have a solid experience with the Qt framework. Alternatively, a web-base HTML solution could be made but, even employing responsive design techniques, would not give the native feel, for end users, that can be achieved developing a real app. Besides that, the multi-platform factor can be a good marketing asset for the project, since portable devices (e.g. tablets, etc) are also things in an IoT sense.

Last but not least, the GUI should be user friendly. That entails being consistent, both in terminology and in the workflow. If the process of adding a new electronic device is very different from adding a new area, users would have twice the effort to get things done and that can limit the adoption of the proposed solution. Ideally, the Impress platform should be so straightforward to use that one cannot ignore its appeal.

2. Scenario-based Descriptions

In this section we will provide typical workflows with screens designed to make the whole application concept more tangible. Figure 2 depicts the main window of the application, which is a dashboard with a menu at the top right hand corner and several dock widgets occupying most of the window. These dock widgets are basically a summary of what a user can find by clicking on the equivalent menu. The only exceptions are the notifications and power consumption dock widgets at the right hand corner. The former can spot possible concerns that a user would like to be aware of. For example, when an energy measurement device has stopped to work or the like. The later displays a line chart of the total energy consumption. To that extent, this dashboard aims to give the overall picture of the monitored environment through the eyes of the Impress platform.

Also, it is worth to mention that a user can return to the dashboard at any time by simply clicking on the  button.

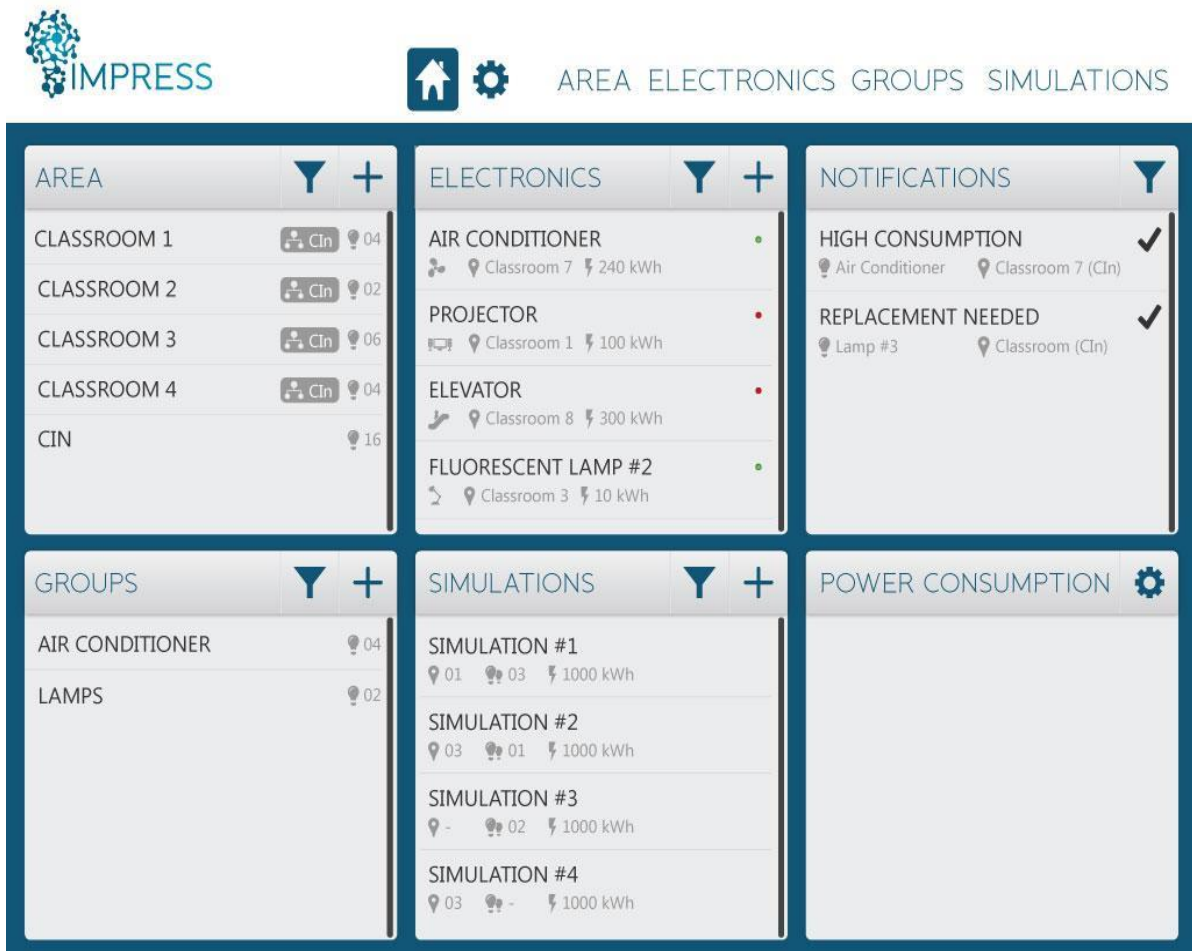


Figure 2: Impress' Application Dashboard Window.

2.1 Scenario 1: Adding New Areas

To add new areas, one could click on the **+** button at the area dock widget in the application dashboard. As a result of that, a modal dialog will appear, like the one depicted in Figure 3.

After filling the desired name for the area in the edit box and specifying the other area where this new area to be created will be nested under in the combo box, as depicted in Figure 4, then the user just has to click on the "Finish" button.

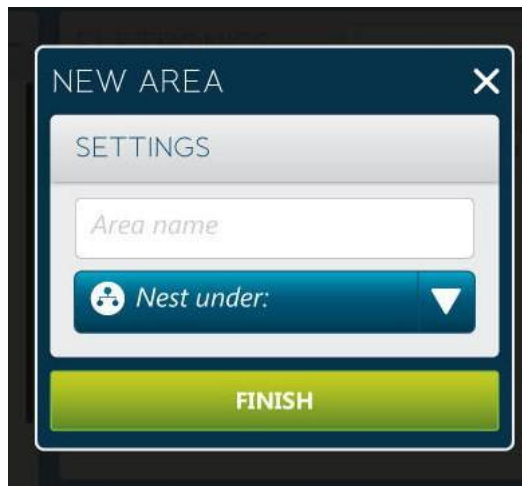


Figure 3: Impress' Application New Area Modal Dialog.

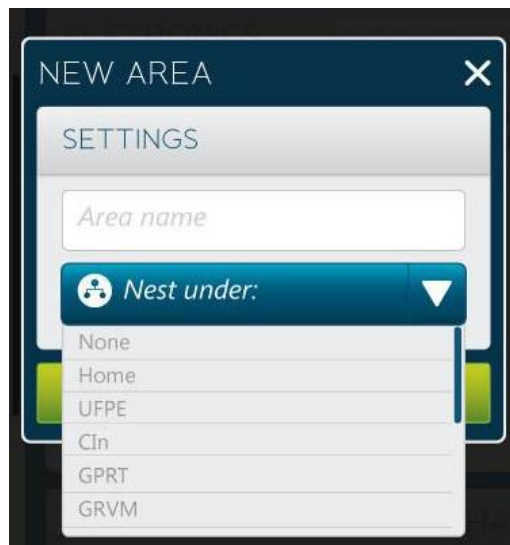




Figure 4: Impress' Application New Area Modal Dialog (nesting feature).

Note that it is not mandatory to nest an area inside another one, but this can help to manage and visualise electronics' energy consumption per area afterwards. Obviously, this nesting is only adopted when the areas are organised hierarchically in areas equivalent to the spatial representation of the on-site facilities.

2.2 Scenario 2: Areas' Monitoring and Management

To monitor and manage the areas already registered in the system, the user should press the "Area" button at the top right hand corner menu. This would lead to the area window, depicted in Figure 5.

At the left side of the window, there is a list of the registered areas in the Impress Cloud. By clicking on any of the listed areas, the user would see the selected area's electronics and the area's power consumption graph, in the electronics and power consumption's dock widgets, respectively. The power consumption chart entails a line chart, showing both the expected consumption, based on the theoretical power consumption provided by the user when creating the devices that are part of the selected area, and the real consumption measured by the WSN that feeds the Impress Cloud regarding the selected area.

Note that it is possible to add new areas and attach electronics to an area directly through the area window, following steps similar to the ones described in scenario 1 (Adding New Areas), via the **+** buttons. Besides that, it is possible to edit or remove an area or electronic through the  and  buttons respectively.

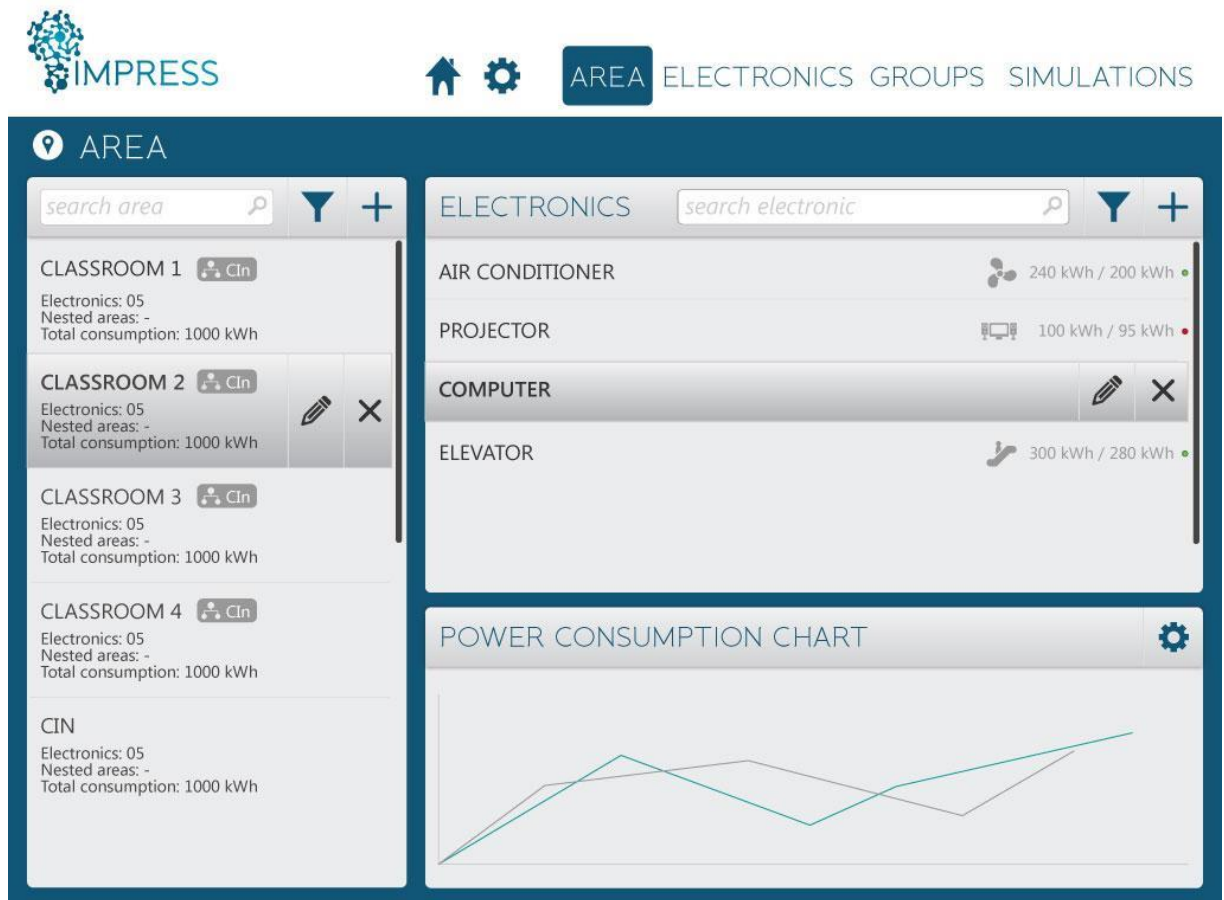


Figure 5: Impress' Application Area Window.

2.3 Scenario 3: Adding New Electronics

To add new electronics, one could click on the **+** button at the electronic dock widget in the application dashboard. As a result of that, a modal dialog will appear, as depicted in Figure 6.

After specifying the device’s type in Figure 7, which can vary from led light bulbs to Uninterruptable Power Supply (UPS) units in its combo box, a user may provide next the desired name for the electronic, its theoretical power consumption as specified by its vendor, and some additional information that the user may find important to store, in the edit boxes that are part of Figure 6. Next, the user simply has to click on the finish button. As a side note, albeit the theoretical power consumption seems to be redundant, since meters/sensors would be actively measuring energy consumption, it is necessary for electronics that were not deployed to be simulated beforehand.



Figure 6: Impress’ Application New Electronic Modal Dialog.



Figure 7: Impress’ Application New Electronic Modal Dialog.

2.4 Scenario 4: Electronics’ Monitoring and Management

To monitor and manage the electronics already registered in the system, the user could select “Electronics” at the top right hand corner menu. This would lead to the electronics window, depicted in Figure 8.

In the upper half of the window, at the left side, there is a list of registered electronics in the Impress Cloud. At the right side, a window containing some technical information on the currently selected electronic device, some details related to the data provided at the creation of the device and its current operation state are provided.

In the lower half of the window, there is the power consumption chart of the selected electronic device. More specifically, this entails a line chart with two pieces of information: the expected consumption, based on the theoretical power consumption provided by the user when creating the selected device, and the real consumption measured by the WSN that feeds the Impress Cloud.

Alternatively, it is possible to add new electronics from the electronics dock at the left side of the upper half of the window, similarly to the process described in scenario 3 (Adding New Electronics).

Besides, it is possible to edit or remove an electronic through the  and  buttons respectively.

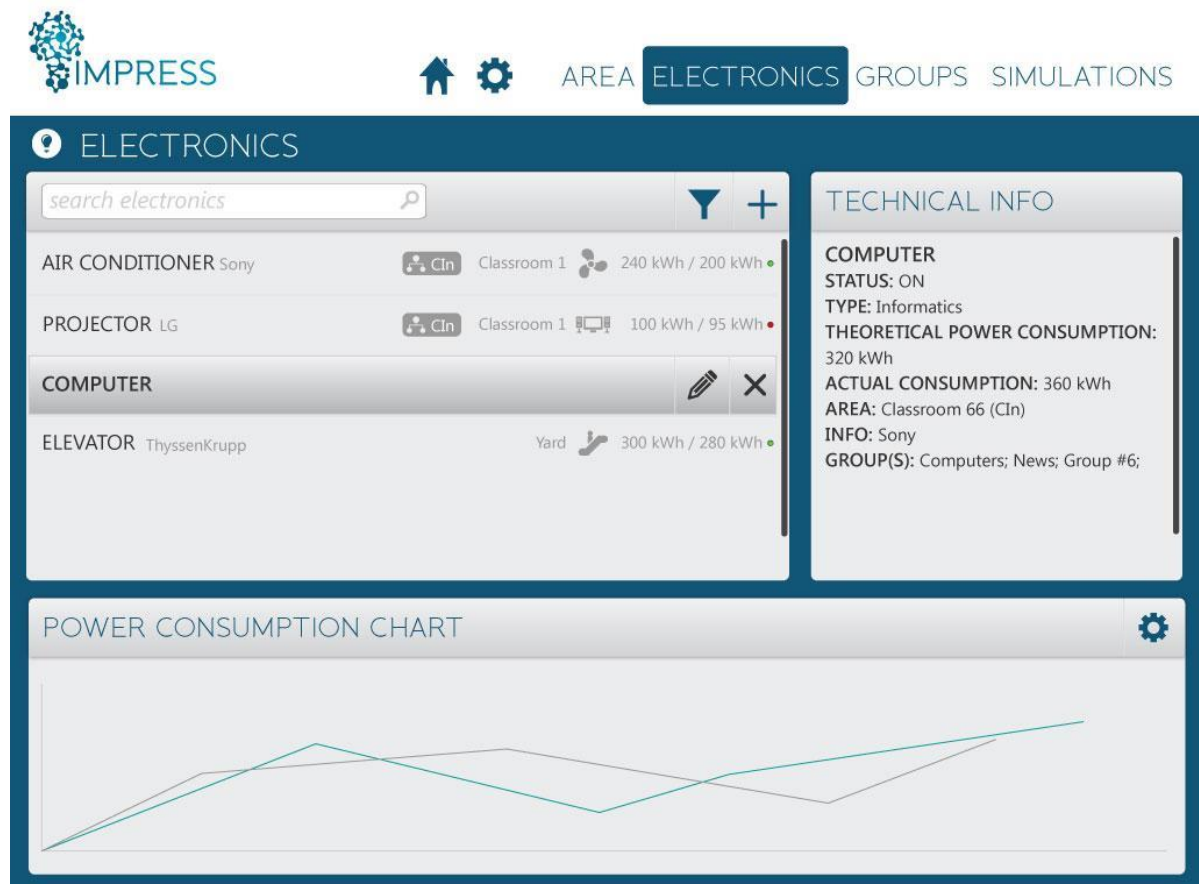


Figure 8: Impress' Application Electronics Window.

2.5 Scenario 5: Adding Electronics to an Area

To attach an electronic to an area, the user should click on the **+** button at the right top corner of the Electronics' dock in the Area window. A modal dialog will appear (Figure 9), showing a list of the existing electronics capable to be added and the possibility of creating a new one from the "New Electronic" button, which leads to a process similar to the one described in scenario 3 (Adding New Electronics). Once this is performed, the last step is to select the "Finish" button. Note that it is possible to select multiple electronics at once by checking multiple check boxes.

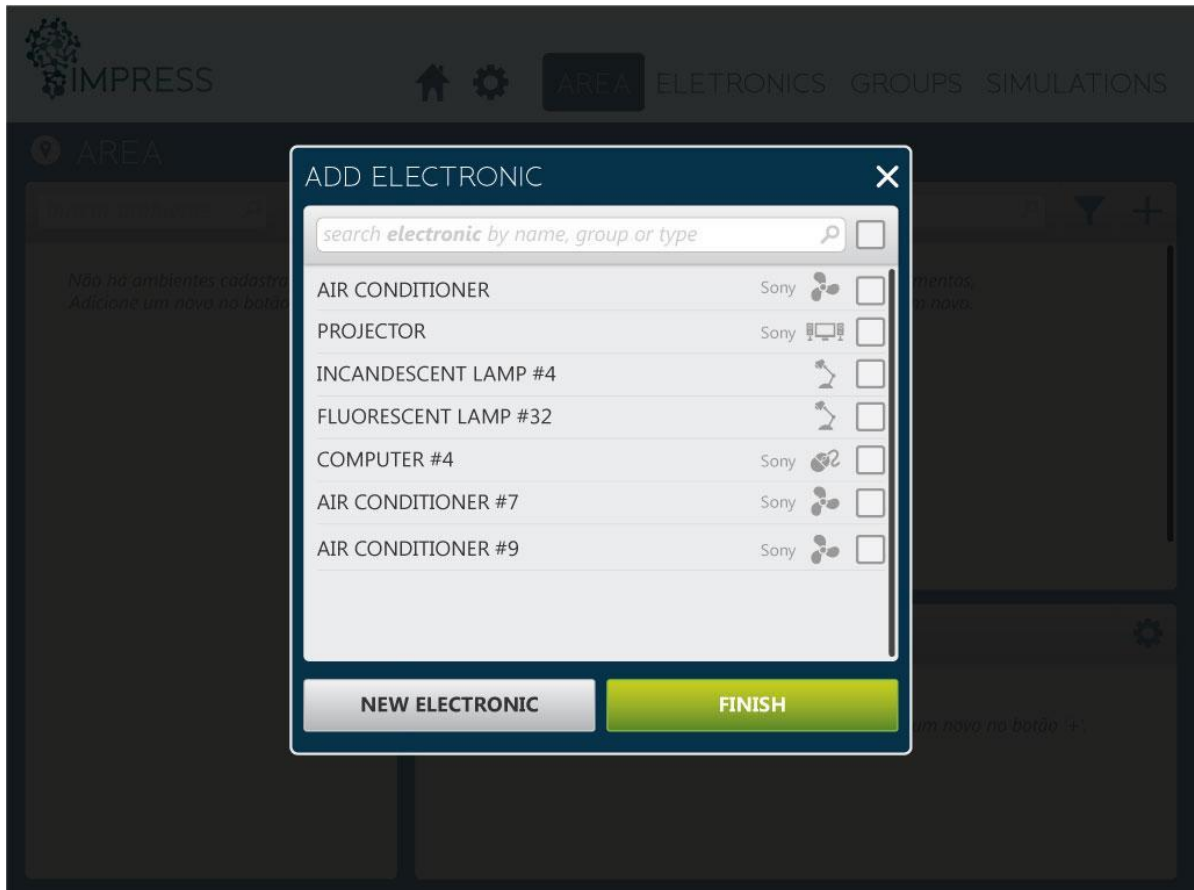


Figure 9: Modal dialog to attach an electronic to an area.

2.6 Scenario 6: Adding New Groups

To add new groups, one could click on the **+** button at the group dock widget. Because of that, a modal dialog will appear, as depicted in Figure 10.

In this modal dialog, the user could fill the desired name of the group and attach electronics to it using the list at the left side of the window, which is possible to be filtered by name, type or area, or remove electronics that are already in the group using the list at the right side of window.

To add/remove electronics, the user should click on the "Add"/"Remove" buttons at the bottom of the respective lists. The last step requires the user to select the "Finish" button, to store the selection as a new group.

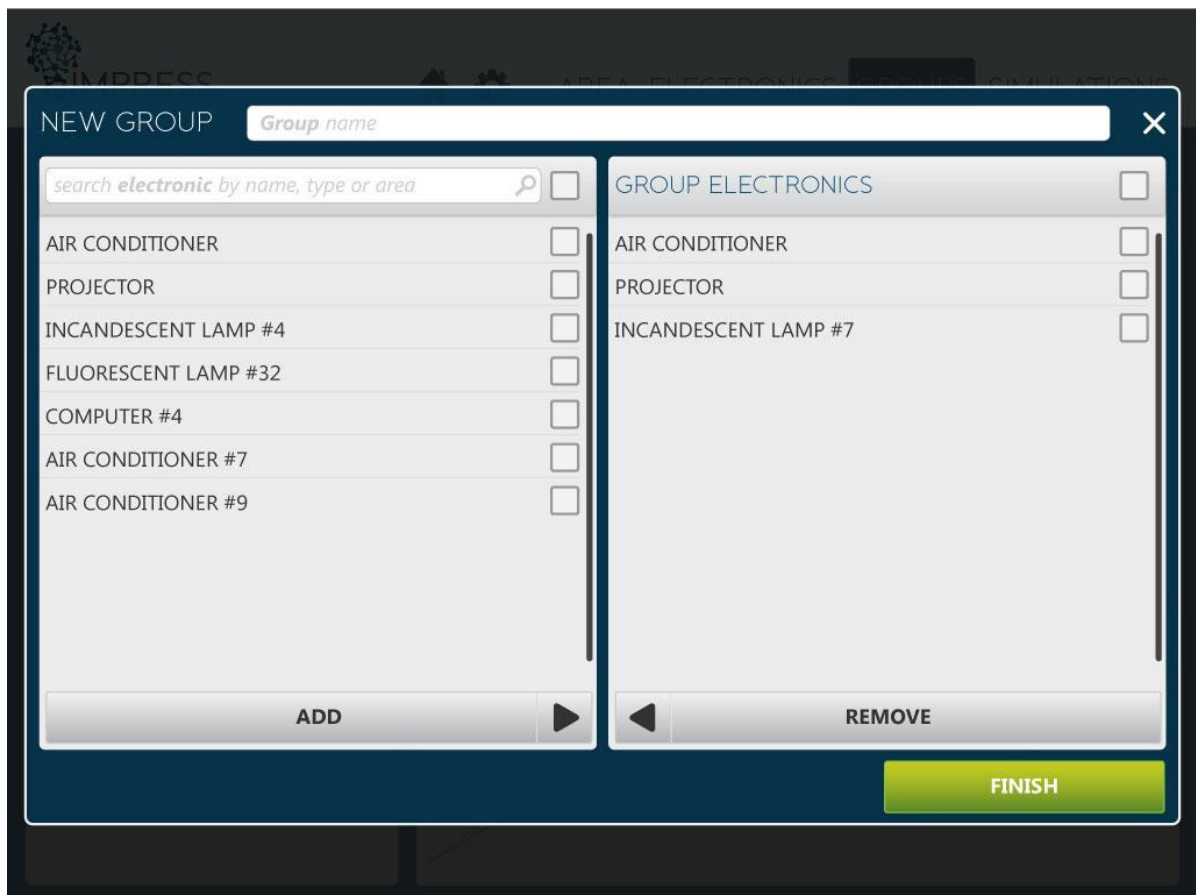





Figure 10: Modal dialog to add a new group.

2.7 Scenario 7: Groups' Monitoring and Management

To monitor and manage the groups that are already registered in the system, the user could select the menu option "Groups" at the top right hand corner menu. This would lead to the groups window popping up as depicted in Figure 11.

The groups window is similar to the area window, depicted in scenario 2 (Areas' Monitoring and Management). At its left side, there is a list of registered groups. At the upper half of the right side, a dock to manage the electronics of the currently selected group and, at the lower half, the power consumption chart of the whole group which is currently selected.

Note that it is possible to add electronics to a group through the electronics dock in a process similar to the described in scenario 5 (Adding Electronics to an Area). It is also possible to add new groups by clicking on the **+** button at the top right hand corner of the groups' list, to edit any of the existing groups through the  button, next to the name of each group and to remove an existing group through the  button next to the  button. The same editing/removing process is applicable to the electronics of the group.

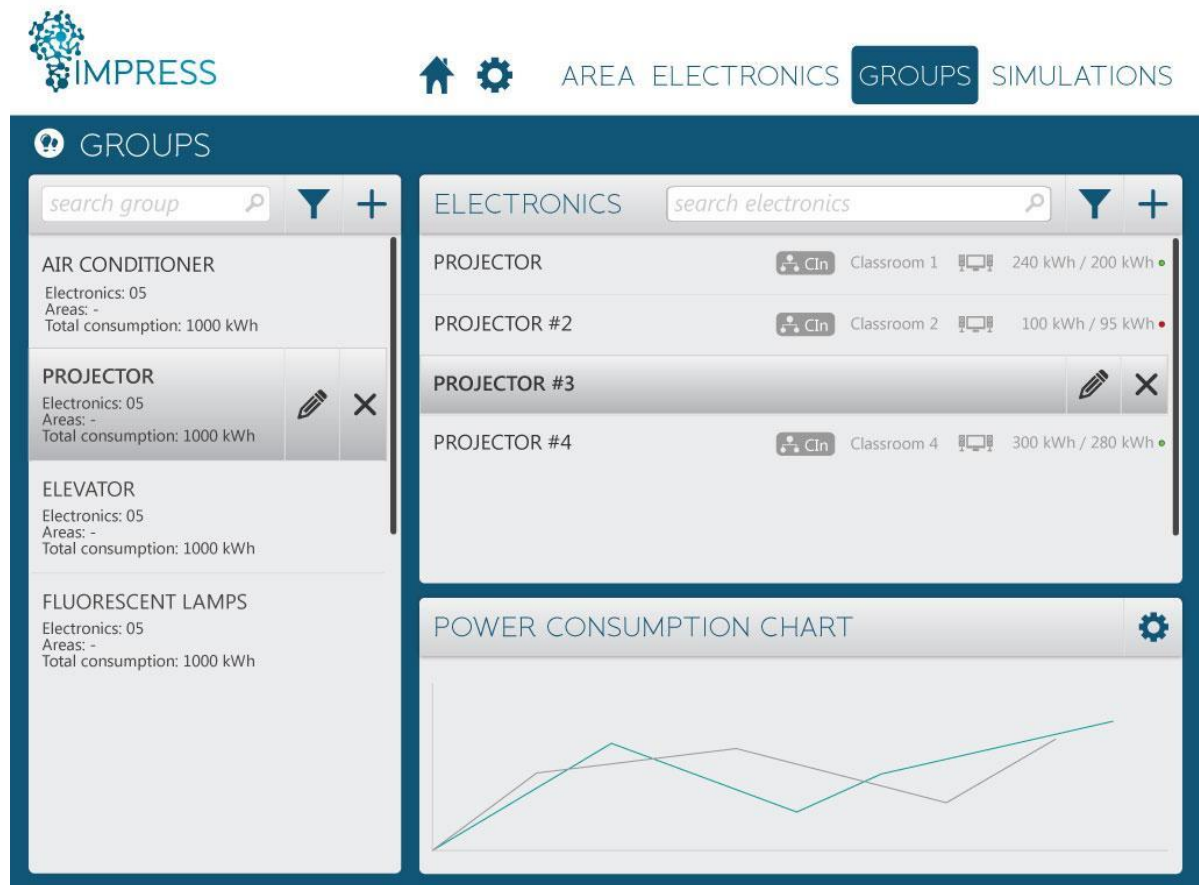


Figure 11: Impress' Application Groups Window.

2.8 Scenario 8: Adding New Simulations

To add a new simulation, the user could choose the **+** button at the simulation dock widget at the dashboard. A modal dialog as depicted in Figure 12 would then appear. Similarly to the process described in scenario 6 (Adding New Groups), the user can fill the desired name of the simulation in the edit box at the top of the window; using the left side list, it is possible to add groups or/and areas to the simulation, while at the right side list it's possible to remove areas/groups that have already been added to the simulation. The last step as usual requires selecting the "Finish" button.

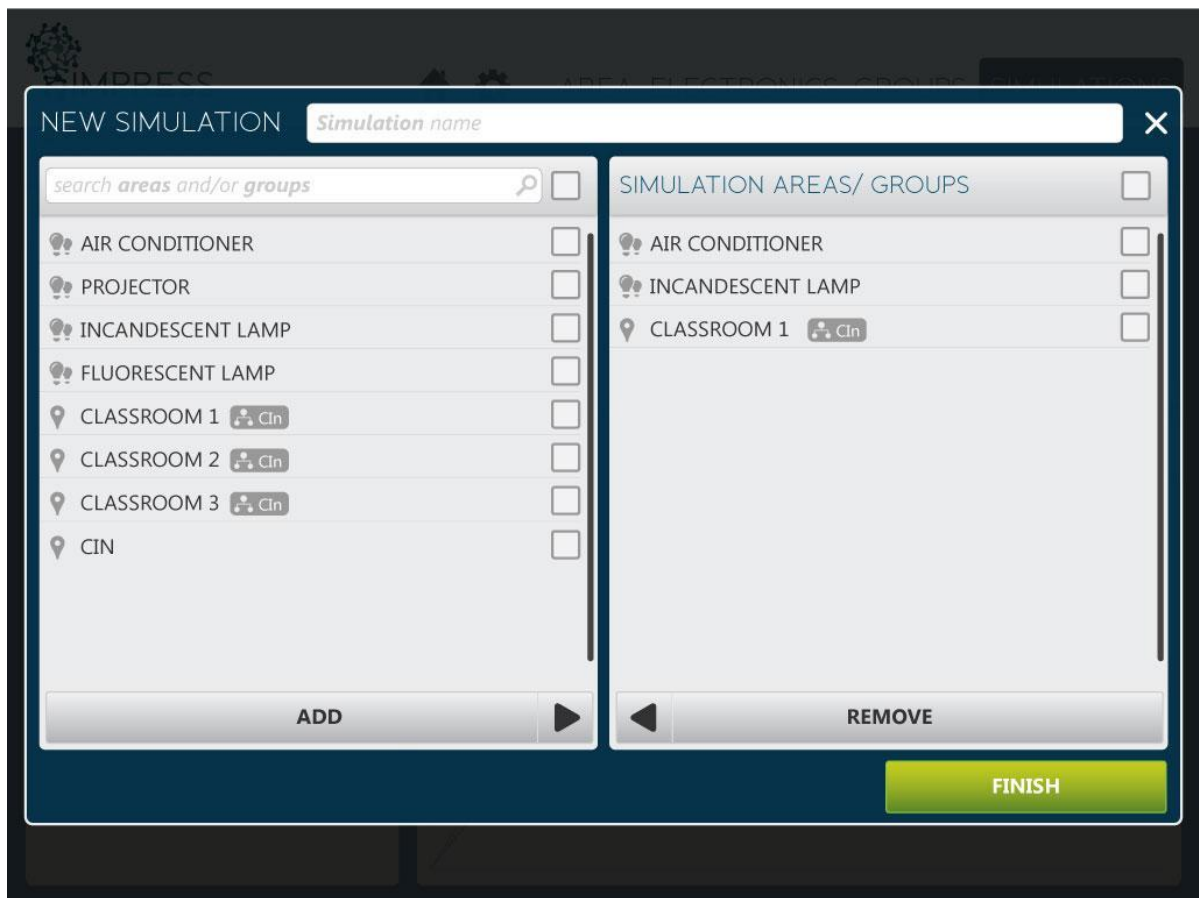


Figure 12: Modal dialog to create a new simulation.

2.9 Scenario 9: Simulations' Monitoring and Management

To monitor and manage the simulations that have been created, the user should click on "Simulations" menu option at the top right hand corner menu. This would lead to the simulations window, depicted in Figure 13.

The simulations window has a simulations' list at its left side, a technical info dock at the upper half of its right side, as well as the simulations' power consumption charts of the selected group in the lower half.

Notice that it is possible to add new simulations in a process similar to the described in scenario 8 (Adding New Simulations) by clicking on the **+** button at the top right hand corner of the simulations' list. It is also possible to edit any of the existing simulations through the **✎** button, next to the name of each simulation, and to remove it as well, clicking on the **✕** button.



Figure 13: Impress' Application Simulations Window.

Summary and Conclusions

In this deliverable, we presented the overview of a possible smart energy management application envisioned to be integrated with the IMPRESS cloud.

The IMPRESS cloud aims to provide an abstraction layer, by the means of its API, so that IoT applications can be developed in a more straightforward way. In this sense, it is exactly seen like a platform-as-a-service (PaaS) for IoT applications. To validate the IMPRESS project vision, we produced a high-level specification that will be used as a guide when carrying out the development of an energy management application using the IMPRESS cloud infrastructure in the next tasks of the work package 8.

The application was conceived in order to facilitate for the users to get an understanding of the data available in the IMPRESS cloud. So that, applications' users can understand the overall impact of the energy consumption in both a per electronic/device or clusters manner. These clusters can represent a physical area, organised in a hierarchical manner, or simply a flexible group of heterogeneous devices. Groups usefulness lies in the fact that not always electronics spatial representation is the better representation for specific kinds of analysis. For instance, one might want to create groups for the two different brands of light bulbs installed on site, therefore being easier to identify common patterns in the energy consumption trends of those two groups in particular. Or, alternatively, one may simulate what-if scenarios just to compare how changing the current light bulbs for more efficient ones would impact the overall energy consumption in the long run, without having to really change the current facilities being measured. To make that last comparison example possible, we proposed a simulation mechanism.

A notification system was also envisioned for the application to detect possible concerns that a user would like to be aware of. Like when an energy measurement device has stopped to work or is delivering implausible data values. Asynchronous messages should then be used to notify users and/or administrators.

Also, we reviewed similar applications regarding smart energy management. Many of these approaches, however, are focused in the energy supplier implementing smart grids or are simply if-this-than-that (IFFT) applications for end users.

In conclusion, we believe that the application should focus on getting meaning and creating useful knowledge from the data we have available in the cloud, so as to make it easier to manage an energy infrastructure by those in charge of such activity. Such management should not require energy suppliers intervention and therefore, avoiding these from being a barrier for adoption of the IMPRESS platform.

References

- [1] Qt framework. <http://qt-project.org/>