AIM

Deliverable 4.1.2

User Applications

Editor:	Alessandro Corrente (Cefriel) and Antonio Capone (Politecnico di Milano)
Deliverable nature:	Prototype (P)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	31 December 2009
Actual delivery date:	19 March 2010 (Prototype was ready on 18 December 2009 and was validated on 9 February 2010 at the integration meeting.)
Suggested readers:	AIM consortium
Version:	1.0
Total number of pages:	45
Keywords:	User application, DVE, Sensor network, Utility application, Network operator application

Abstract

This deliverable contains the installation instructions of the prototype developed and a user manual with a detailed description of all DVE functionalities and settings. For all components of the use cases presented in the deliverable 4.1.1 there is a section in this document with the minimum requirements that the platform hosting the application has to provide and a section with the instructions to configure the component. Moreover we provide a description of the home user interface and of the mobile application designed to be easy to use and to allow a user to customize the DVE management logic in order to satisfy his needs. The installation and usage instructions are output of WP4 and are provided as input to WP5. The first ones are intended to support the installation phase in real households during the system evaluation period whereas the usage instructions are addressed to householders that will use the AIM system.

Disclaimer

Impressum

[Full project title] A novel architecture for modelling, virtualising and managing the energy consumption of household appliances

[Short project title] AIM

[Number and title of work-package] WP4 - Integration of the AIM architecture with the home network

[Document title] User Applications

[Editor: Name, company] Antonio Capone/Alessandro Corrente, POLIMI /CEFRIEL

[Work-package leader: Name, company] Thierry Milin, France Telecom

[Estimation of PM spent on the Deliverable] 40

Executive summary

The main objective of the AIM architecture is to offer a system for managing in real time energy consumption of home appliances. The AIM approach has the goal to develop an open and low cost solution with a user friendly interface that can be installed in every house. Home users can always have access to system settings and they can modify the operation mode of the AIM system or disable it at every time.

In previous documents (deliverable 3.1) it was described the system architecture and communication protocols with domestic appliances and technologies adopted.

In the deliverable 4.1.1 it was presented a report on the development phase of the use-cases with their software components. The use cases considered by AIM project are three: use-case for residential users, use-case for power distribution network operators, use-case for legacy service providers. These use-case are obviously interdependent and each one takes advantage of the services offered by the software modules of the others

This deliverable contains the installation instructions of the prototype developed and a user manual with a detailed description of all DVE functionalities and settings. For all components of the use cases there is a section in this document with the minimum requirements that the platform hosting the application has to provide and a section with the instructions to configure the component. Moreover we provide a description of the home user interface and of the mobile application designed to be easy to use and to allow a user to customize the DVE management logic in order to satisfy his needs.

The installation and usage instructions are output of WP4 and are provided as input to WP5. The first ones are intended to support the installation phase in real households during the system evaluation period whereas the usage instructions are addressed to the householder that will use the AIM system. A description of the measurements and the evaluation criteria evaluated in the test phase will be included in deliverable 5.1.1 because are part of the WP5.

List of authors

Company	Author	Contribution
CEFRIEL	Alessandro Corrente, Luca Sioli, Fabio Lavezzini	Chapters 1, 2, 4, 7.1
FRANCE TELECOM	Thierry Milin	Chapters 5, 7.2
POLIMI	Antonio Capone, Stefano Melzi, Luca Pietro Borsani, Antimo Barbato	Chapters 1, 2, 3, 8
PPC	Lars Schwendicke	Chapter 6

Table of Contents

List of figures

List of tables

Abbreviations

3D	3 Dimensional
API	Application Programming Interface
AS	Advanced Services
AUS	AIM Utility System
CO2	Carbon Dioxide
CPU	Central Processing Unit
DB	Data Base
DBMS	Data Base Management System
DVE	Device Virtualization Environment
EMD	Energy Management Device
GUI	Graphical User Interface
HTTP	Hyper Text Transfer Protocol
HVA	Home Virtualization Application
ID	Identifier
JAR	Java Archive File
JDBC	Java Data Base Connectivity
JDK	Java Development Kit
JNDI	Java Naming Directory Interface
JNI	Java Native Interface
JRE	Java Runtime Environment
KW	Kilo Watt
PC	Personal Computer
PIR	Passive Infra Red
PSA	Profiling System Application
RAM	Random Access Memory
RDBMS	Relational Data Base Management System
REST	Representational State Transfer
RFID	Radio Frequency Identification
RMI	Remote Method Invocation
SOAP	Simple Object Access Protocol
TV	Television
USB	Universal Serial Bus
WiFi	Wireless Fidelity
WSN	Wireless Sensor Network

Definitions

Web Server: is a computer program that delivers content, such as web pages, using the Hypertext Transfer Protocol.

***Web Service*:** is the communication between a client and a server through the web using the HTTP protocol. It consists of an application programming interfaces (API) that can be accessed over a network, such as the Internet, and executed on a remote system (web server) hosting the requested services. Web services are commonly divided in two category: SOAP Web Services that provides a machine-readable description of the operations offered and RESTfull Web Services.

Relational Database Management System: RDBMS is a set of computer programs that controls the creation, maintenance, and the use of databases. A DBMS is a system software package that helps the use of integrated collection of data records and files known as databases. It allows different user application programs to easily access the same database. The relational model is a system for storing and working with large databases based on tables of fixed-length records.

Java Virtual Machine is a set of computer software programs and data structures that use a virtual machine model for the execution of java language programs and scripts.

Energy Mix is the description of energy sources used and the amount of energy generated by every kind of source. It is provided for every hour of the day and it allows for example to chose to consume more energy when it is available from renewable sources and save energy during other periods.

1 Introduction

The main objective of the AIM architecture is to offer a system for managing in real time energy consumption of home appliances. The AIM approach has the goal to develop an open and low cost solution with a user friendly interface. Home users can always have access to system settings and they can modify the operation mode of the AIM system or disable it at every time.

The AIM system exchanges data about energy tariffs and domestic consumption with the utility provider to afford high level energy consumption management services and it communicates also with the telco operator enabling access to the system by a user from outside the home.

The architecture presented in Figure 1 is managed by three different users:

- the householders who control all appliances in the home and his DVE
- the telco operator who is in charge to provide connection to internet and remote access through its mobile application
- the utility provider who sends energy tariffs and energy mix information to the system and receives data about energy usage.

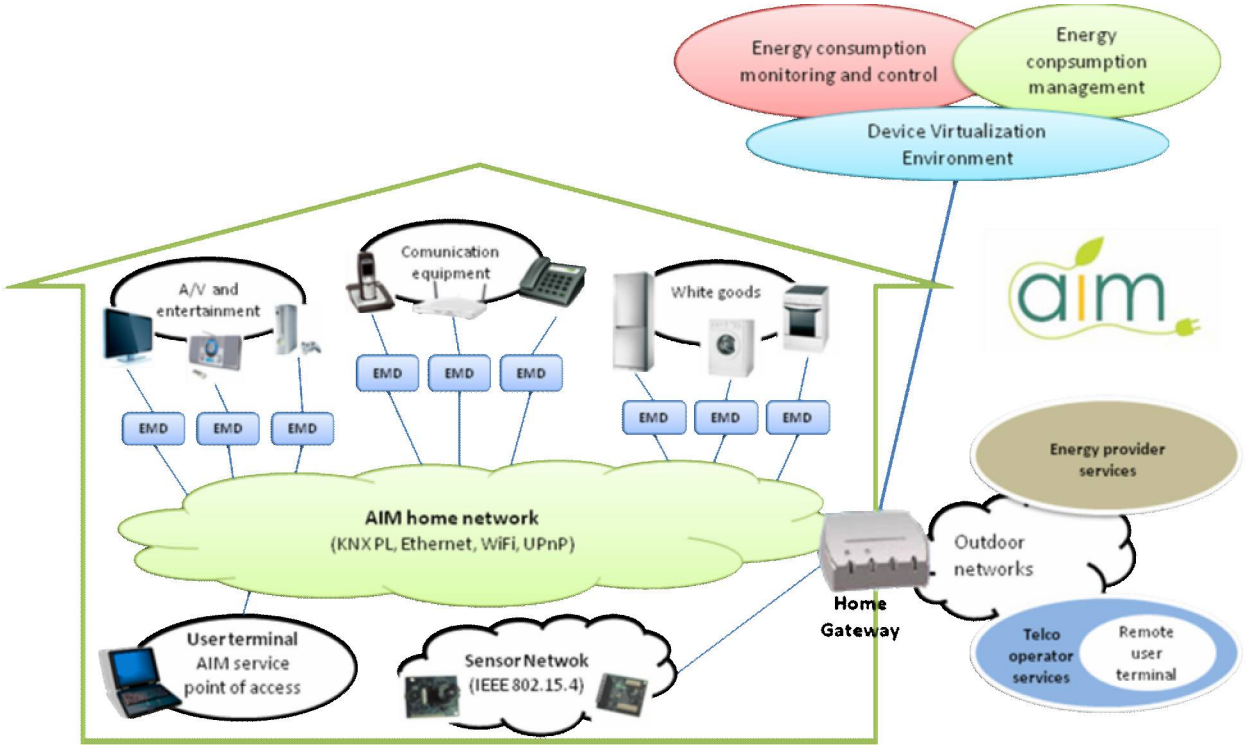

Figure 1: AIM system architecture

1.1 Objective of this document

The prototype developed during the project is a complete implementation of the architecture described in previous deliverables. This document is the manual of the prototype for users and it describes the installation phases of all components and the use of the AIM system.

In chapter 2 we present an overview of the features of our system and how a user can interact using different interfaces (the local GUI or the mobile application) with the DVE.

Chapter 3 describes the installation steps to set up the AIM wireless sensor network and to make it work properly. The chapter starts with the minimum system requirements to install the wireless sensor network software and in the following it describes the installation and sensors deployment procedure. In particular it focuses on how sensors should be positioned in a home environment to obtain the best performance.

Chapter 4 describes the installation and set up of the DVE following the same schema of the previous chapter. We provide instructions to configure the DVE in order to connect it with the wireless sensor network and with the telco platform.

Chapter 5 provides instructions to install the telco platform to enable the communication of the DVE with the mobile application and to retrieve information about energy cost from utility.

Chapter 6 describes installation and configuration of the utility web server that provides the energy tariffs and information about energy mix to DVEs.

The focus of chapter 7 is on the interaction of a user with the Web Interface. It describes all functionalities that can be accessed and all operation modes of the DVE. It includes also a description of the mobile application that allows a users to access to his DVE when he is not at home with his mobile device through the telco platform.

2 Application overview

A residential user can directly access the services provided by the AIM system through a local interface. The general objective, in the residential user use case, is to manage the energy consumption experimented in a generic house through a set of operations like:

- monitoring of the energy use
- personalization of energy use

In domestic environment users interact with DVE using a Web local Graphic User Interface (GUI). Through the local GUI a user can select all the desired functions, enforce actions on devices, monitor status or simply ask for energy saving functions. The Web local GUI is a Web-Based interface compatible with different HTTP browsers so the user can access the DVE from his/her Personal Computer. This GUI is designed for wide screen monitor like Personal Computers ones.

In home environment the DVE manages all appliances on the basis of user preferences, information received by the Wireless Sensor Network and energy tariffs received by utility.

The AIM Gateway is supplied with variable energy prices. This variable tariff has a given granularity, which is expected to be between 15 minutes and one hour. And it is supplied to every household one day ahead. The AIM Gateway can use this information to plan the power consumption for managed devices for the next day, as described in D2.2. In addition to the tariff information, environmental parameters to the generated electricity can be supplied to the AIM gateway. These information can show the ecological impact of energy consumption and thus encourage further energy saving.

Ecological parameters are:

- Energy Mix: the mixture of energy produced by a utility for a tariff
- CO2 production per energy unit
- nuclear waste production per energy unit

These information can be taken into account for device scheduling in the household as well.

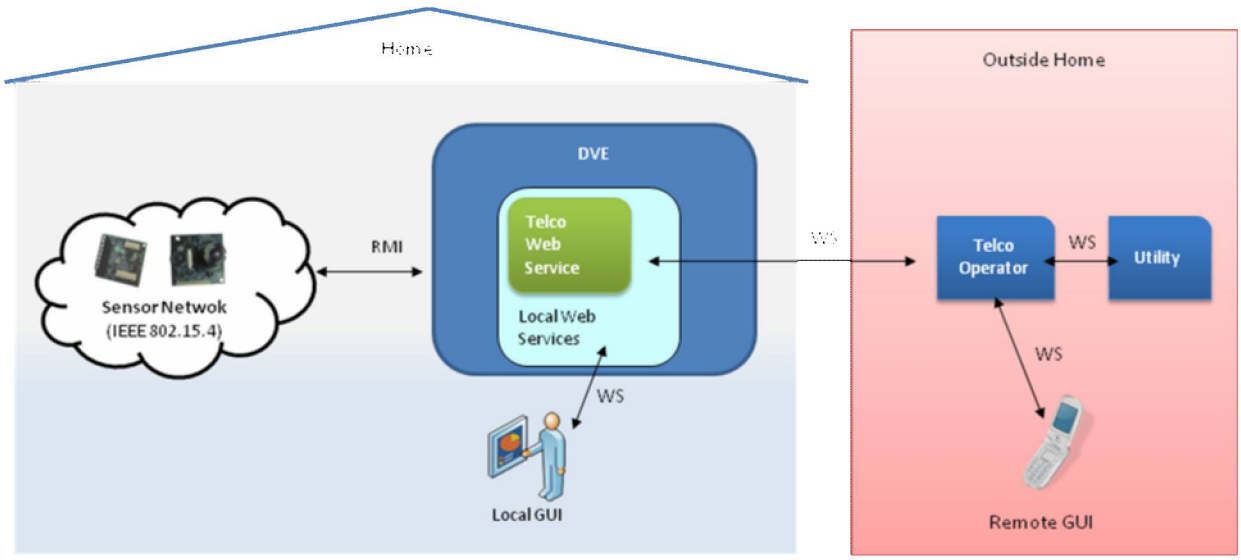

Figure 2: Interfaces with the DVE

Remotely a user can access and interact with his/her home DVE using the AIM mobile application. In this way a user can monitor the status of home appliances and home consumption through his/her portable device, can change some DVE settings or add tasks to the DVE (see 7.1.3). The system can also send alarms to the user if there are anomalous situations like excessive consumptions or appliances malfunctioning.

3 Sensor Network Configuration

3.1 Requirements

3.1.1 Basic System

The platform hosting the WSN Gateway (ref. deliverable 4.1.1) is a PC or an embedded platform, with these minimum requirements:

- One USB port
- Supporting Java Runtime Environment 1.5

The platform hosting the RFID application to control home access is a PC with these requirements:

- Windows XP or Vista installed
- One USB port
- Internet access

The WSN Gateway and the RFID application can be hosted on the same machine, and besides the two applications can be hosted on the same machine hosting the DVE.

3.1.2 Java

The WSN Gateway application is a Java based software and requires the installation of the Java Runtime Environment at least version 1.5. Java Native Interface libraries (JNI) are required to enable the communication with sensors through the USB port.

3.1.3 Database Management System – MySQL

Data retrieved by the Sensor Network and profiles calculated by the Profiling System are stored in a database. Every Database Management System with a JDBC connector can be used, we have chosen MySQL which is an open source DBMS with a freely downloadable version on the MySQL website [7]. It is also possible to find, in the MySQL site, the JDBC driver.

The installation of the MySQL server can be done manually following the instructions on the software guide but we suggest to use a package called WampServer [8] as described in section 3.1.4.

3.1.4 WebServer – Apache

When a user registers his presence at home, passing his RFID tag over the reader, a web page is displayed on the screen with a welcome message (Figure 3 and Figure 4) and the list of users in the home is updated on the database. A web server is required to display the web page and a PHP extension must be installed in order to update the database. We use the well known Apache web server and we suggest to install it with the WampServer [8] package.

3.2 Installation steps

The installation of the AIM wireless sensor network is split in two phases: the first is the preparation of the platform which acts as gateway and provides a user interface, the second phase is the deployment of the sensors in the home.

3.2.1 RFID Access Registration

The RFID access registration system requires to install on the hosting PC the WampServer [8] package following the wizard and the reader specific software client [9].

To configure the system it is necessary to copy the files in the Apache web directory and to configure the table “rfid_users” in the database with the names of the users and their tag.

TAG ID	User Name	Status (at home <1> or outside <2>)	Last access (date)
123456789	John	0	01/01/2010 10:30

Table 1: rfid_users table example

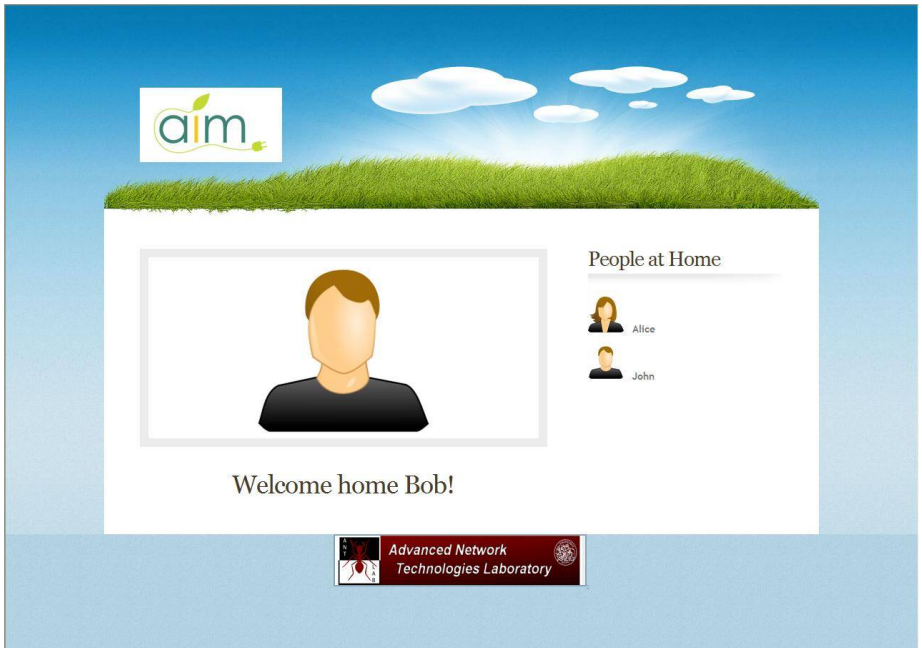

Figure 3: RFID welcome screen

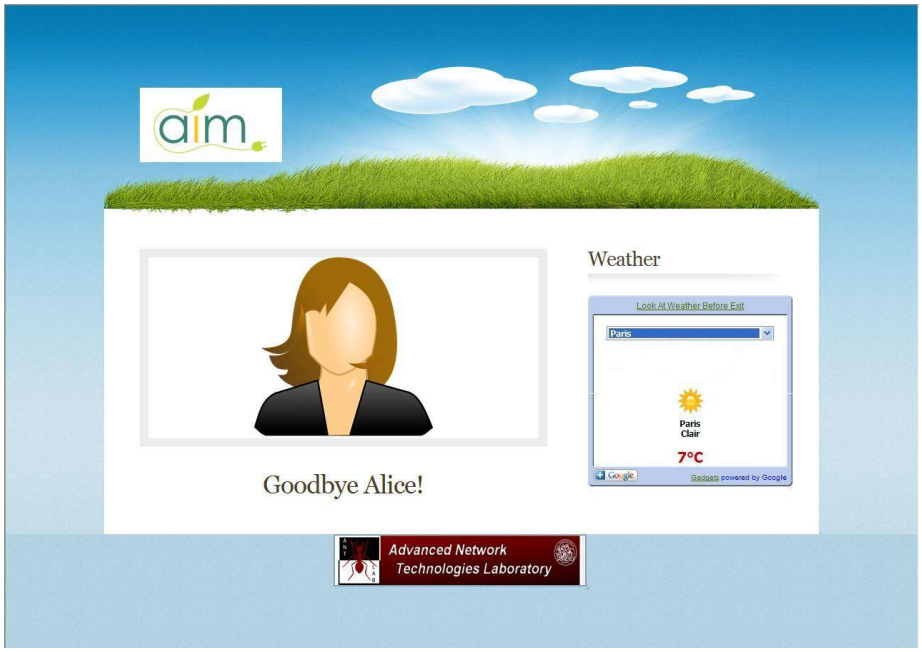

Figure 4: Message displayed when someone exits the home

3.2.2 Basic Java Applications

The first installation step is to start the WSNGatewayManager and the WSNGateway to interact with the sensor network. These two applications are Java based and they don't need any installation procedure.

Before starting these applications the user needs to start the Java Remote Method Invocation Registry, this can be done following the instructions on the Java Tutorial web site [10]. Moreover the Network Coordinator Sensor must be connected to the PC and turned on.

The WSN Gateway Manager and the WSN Gateway are released in two Java Archive File (jar) and can be launched from the operative system command line with these two commands:

```
java -Djava.rmi.server.codebase=file:/C:\\<path>\\ -
Djava.rmi.server.hostname=<rmiregistry_ip_address> gateway_manager.WSNGatewayManagerMain
<manager_ip_address> <gateway_manager_name>
```

```
java gateway.WSNGatewayMain <manager_ip_address> <gateway_manager_name>
```

The last application for configuring the sensor network is the Network Management Application (Figure 5). It can be launched with this command:

```
java -Djava.rmi.server.hostname=<manager_ip_address>  console.MobiWSNConsoleApp
```

3.2.3 Sensor Deployment

After the start of AIM basic java applications the wireless sensors must be positioned in the house.

In order to enable the living environment monitoring, the sensor network has to be created deploying three types of sensors (temperature, light and PIR sensors) in each room of the house. In every room must be at least one sensor for every type but can be deployed more sensors to have more accurate measures.

As described in deliverable D4.1.1 we use two types of motes (MicaZ and Imote2) respectively equipped with sensor board MTS300 (with temperature and light sensors) and IMB400 (with PIR sensor).

To monitor the light level and temperature in every room in a reliable way we suggest to deploy two or more MicaZ per room to avoid failures and to improve precision of the measured values. The number of MicaZ to be deployed depends on the dimensions of the room.

Whereas to detect presence of people in a room it is necessary to deploy the Imote2 sensors properly so that the union of the all PIR areas covers the whole house plan. Every Imote2 can detect movements in a semispherical area of about 5 meters of diameter, but the detection is more accurate within of a 2 meters area. We suggest to not overcome this threshold to detect also little movements.

3.2.4 Sensor Network Configuration

The wireless sensor network (WSN) can be managed through the *Network Management Application* that allows to show topological information, create mote groups and use Advanced Services (AS). This java application allows to manage the sensor network with a graphical interface (Figure 5).

The user must write in the Network Management Application window the WSNGatewayManager URL that he want to connect with, before clicking the connect button.

The AIM system to work properly needs the creation of a group of motes for each room of the house, composed by the sensors previously located in that room. This allows the system to know which sensors are located in every room.

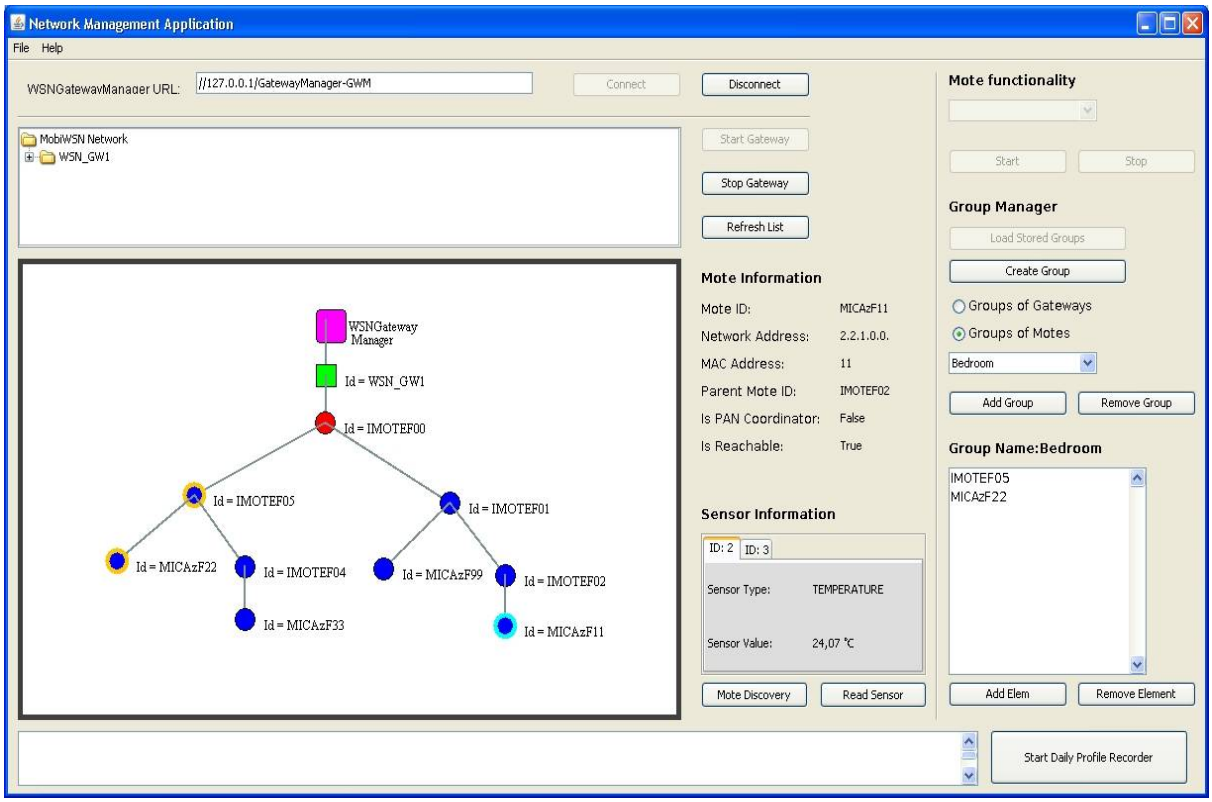

Figure 5: Network Management Application

To add a new group it is necessary to click on the "Add Group" button and to choose the group type and name, as shown in the next figure.

Figure 6: Mote group creation

Elements can be added to a group with the "Add Element" button. When the last element has been added, clicking the "Create Group" button the system sends information for group creation to the motes.

This graphic interface provides also details about the status of the sensor network, in particular when a node is selected it shows the topology information (mote ID, address, parent node, sensors list and AS list).

Finally when a group is selected it is possible to see some group information, among which nodes belong to the group, they are highlighted by orange rings.

3.2.5 Profiling System

At the end of the WSN configuration, it is possible to launch the Profiling System Application (PSA) that processes information provided by sensors in order to profile users behaviours. This java application uses a MySQL Database (DB) to store data and configuration information. The PSA is released in a Java Archive File (jar) and it can be launched from the operative system command line with this specific command:

```
java -Djava.rmi.server.codebase=file:/C:\\<path>\\ -
Djava.rmi.server.hostname=<rmiregistry_ip_address> profiling_system.ProfilingSystemMain
<PSA_ip_address> <PSA_name> <manager_ip_address>  <gateway_manager_name>
```

When the PSA is started a window opens (Figure 7), giving the user two possible choices (the first time the PSA is launched this window is not displayed because no database is available):

- Use the pre-existing database: the PSA will use data and configuration information previously stored in the database;
- Create a new database: the DB is cleared so that PSA can fill it starting from an empty database.

Figure 7: Database recovery window

If the "Use the pre-existing database" option is selected, the PSA is ready to interact with the sensor network and no other configuration steps are required. Instead, if the "Create a new database" option is chosen, the user is asked to fill the configuration information of the DB, editing a name for each created sensors group and specifying which rooms are directly connected through a door, as described in Figure 8 and Figure 9. After this configuration step the user profiling starts and no other interactions with the user are required.

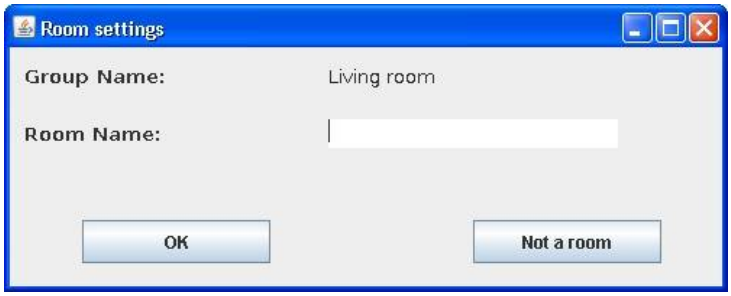

Figure 8: Rooms naming step

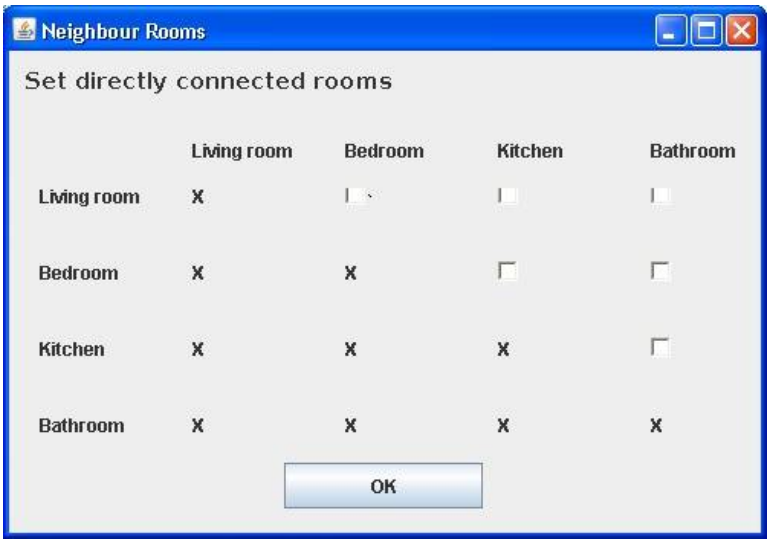

Figure 9: Neighbor rooms definition

3.2.6 Virtualized devices and home monitoring

For the house monitoring a graphical interface has been created called Home Virtualization Application (HVA). The HVA implements some virtualized devices (e.g. air conditioners, lighting systems, TV, WiFi routers) that are directly controlled by the DVE and allows monitoring temperature, luminance and presence status in each room of the house based upon data provided by the PSA and the sensor network.

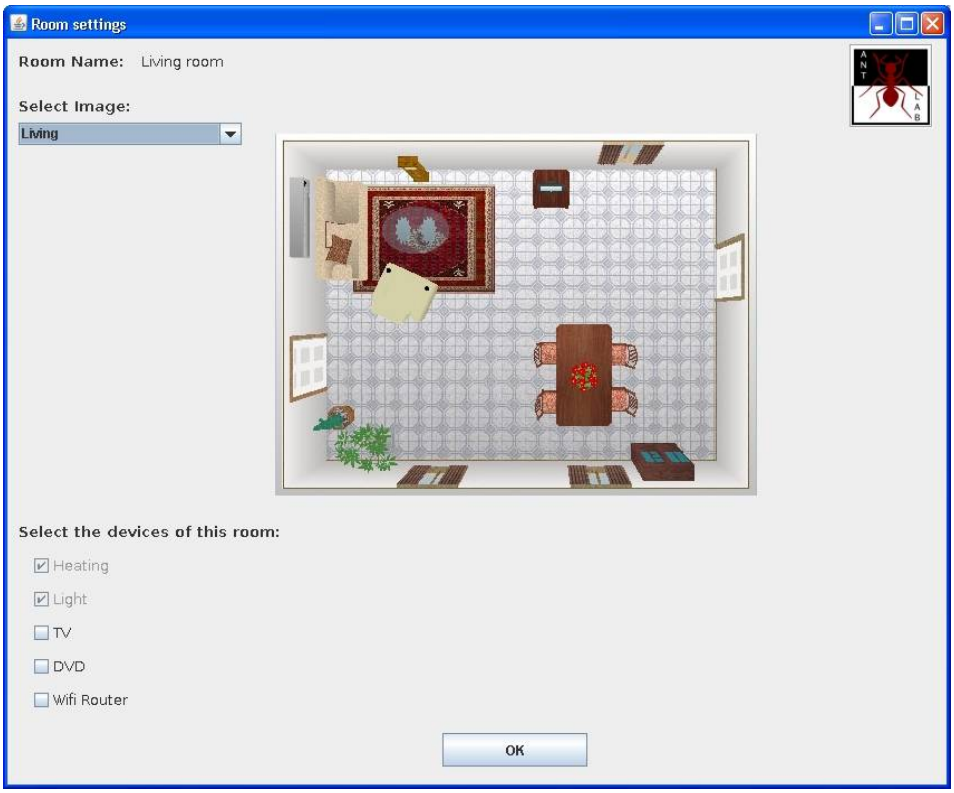

Figure 10: Home Virtualization Application set up

The HVA is released in a Java Archive File (jar) and can be launched from the operative system command line with this specific command:

```
java -Djava.rmi.server.codebase=file:/C:\\<path>\\ -Djava.rmi.server.hostname=<rmiregistry_ip_address>
home_virtualization_application.HomeVirtualizationApplicationMain <HVA_ip_address> HVA
<PSA_ip_address> PS
```

When the Home Virtualization Application is started a configuration step is required in order to associate a 3D image with each room of the house (Figure 10).

After this configuration step, the HVA window is displayed (Figure 11) and the house monitoring can be performed through the *java buttons* located at the top of this window. The first button on the left of the panel allows the application to upload information stored in the database in order to represent the house map; the green and red buttons can be used to start/stop the house monitoring and, finally, the last two buttons allow the users to manage the heating/cooling and lighting systems.

To start the HVA monitoring, the blue button has to be pushed so that the house rooms can be represented in the house map section through 3D images. These images can be moved by the user in order to locate them in the preferred positions. After this configuration step, the green button can be used to start collecting and representing information provided by the sensors and the PSA.

The HVA window is structured so that data can be showed through two main panels: House panel and Room panel. The first one represents the house map and some house information while the second panel is composed by a 3D animated image associated with the selected room and four java panels:

- *Information*: shows information provided by sensors located in the selected room;
- *Room Devices*: shows the virtualized devices located in the selected room and their status;
- *Consumption Panel*: represents the consumption chart of the selected room for the current day; this chart is computed using the devices status information and consumption data;
- *Presence Panel*: represents the predicted presence profile and the daily presence profile of the selected room for the current day.

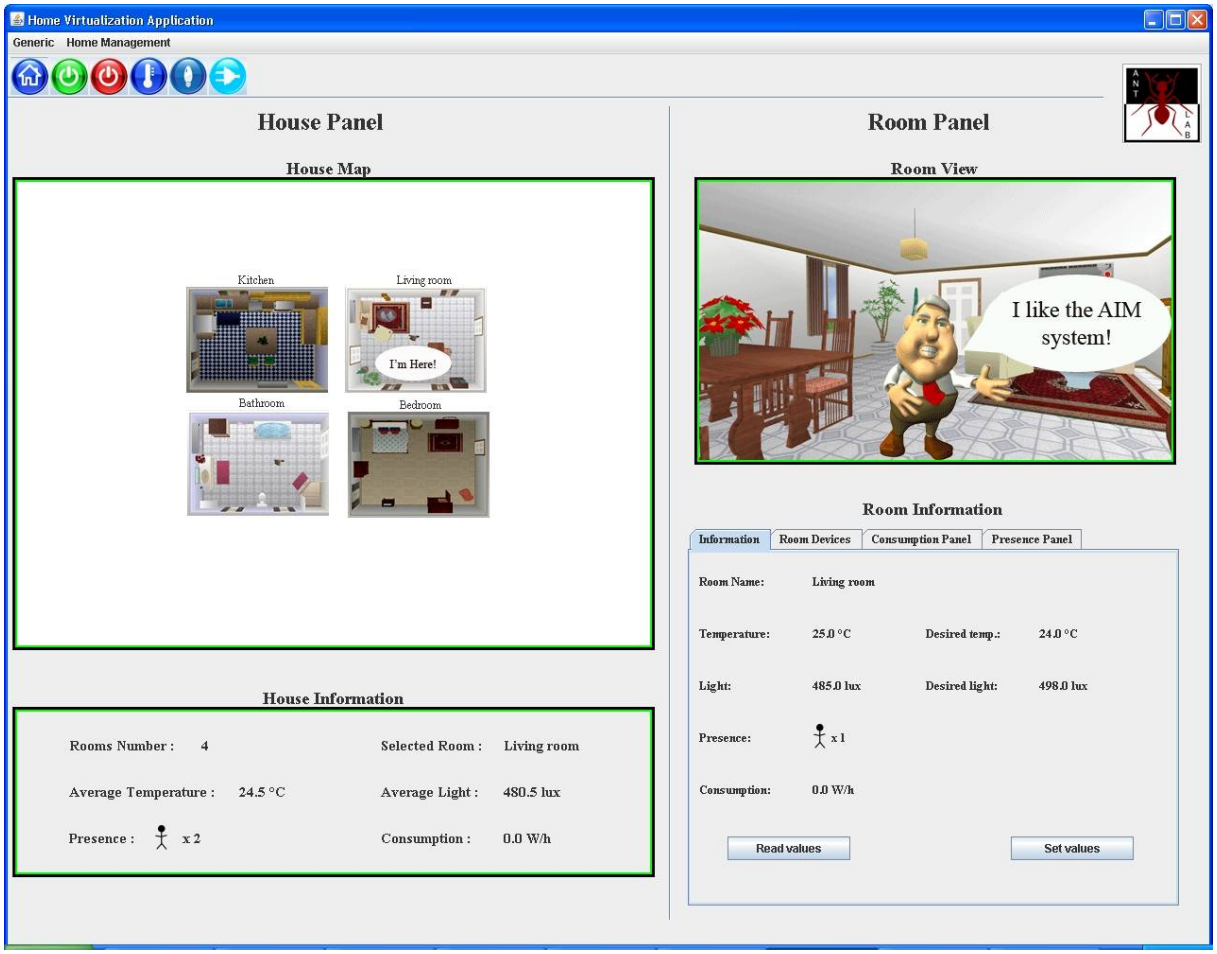

Figure 11: Home Virtualization Application GUI

4 Device Virtualization Environment Configuration

4.1 Requirements

4.1.1 JDK

The Device Virtualization Environment software is a Java based software and it can be installed over all the most common operative system platforms. DVE requires the installation of a Java Virtual Machine 1.6 or higher versions. The JDK is available at the download page of the java web site [5] referring to the JDK installation. Therefore, a full JDK has to be available and the **JAVA_HOME** directory has to be configured properly.

4.1.2 Tomcat

The DVE provides access to energy saving servicers through a web service interface. The DVE requires the installation of the Apache tomcat web server 6.0 or higher version available at the tomcat web site [6] download page.

4.1.3 MySQL

The DVE uses a relational database to store all data persistently. It is based on the MySQL RDBMS that must be installed on the hosting machine before installing the DVE. The MySQL DBMS is available at the MyQSL site [7] download page.

4.2 Installation steps

Before starting the DVE installation, the operator must verify that JDK, Tomcat web container and MySQL are already installed on the machine and verify that MySQL service is running.

The DVE installation is divided in two phases:

1. To complete the first one it is necessary to run the DVE installer and following the instruction of the installation wizard.
2. In the second phase the configuration.ini file, that is stored in the DVE installation path, must be modified to set parameters if needed.

Internet connectivity should be available for the communication with the utility and the telco platform.

5 Telco platform Configuration

5.1 Requirements

The modem router must have Ethernet ports for :

- DVE
- Keletron residential gateway
- Doebelt power meter.
- Phillips TV set.

It must support NAT/PAT and must be accessible from the internet with a fixed address.

5.1.1 AIM Operator platform

The AIM operator platform runs on a standard x86 PC with at least following performance requirements:

- 2 GHz Single-Core CPU
- 1 Gigabyte of RAM
- 20 Gigabyte of hard disc space.

The required operating system is Linux.

The AIM operator platform :

- must allow to send SMS
- must be accessible from the internet by Utility platform.

5.1.2 AIM Mobile application

The application is designed for an Android 1.5 version.

The screen must have 3.2 inch size and support 320 x 480 resolution.

The operating system settings must to be configured to accept unknown sources to allow non-Marked applications.

5.2 Installation steps

5.2.1 Internet access

The local IP address of the gateway should be 192.168.1.1. Access to the router from the internet is supported by a dyndns.org account. The address to use is "http::/aimgrenoble.dyndng.org".

Configuration of the modem router is performed by local administrator with "admin" / "admin" for login and password. NAT and PAT have to be defined according to local IP address for each device.

5.2.2 AIM_Operator platform

Utility platform access to operator portal is :

- http://192.44.60.172:8080/AimOperator.

The operator platform is based on Orange background technology running on Tomcat. Adaptation of the platform is mainly supported by AIM web services deployment.

The web services to be deployed are :

- http://192.44.60.172/axis/services/RemoteLoadControl
- http://192.44.60.172/axis/services/IdentInfoMgmtServerService

These web services proceeds data format conversion between internal database format between DVE and Utility platform.

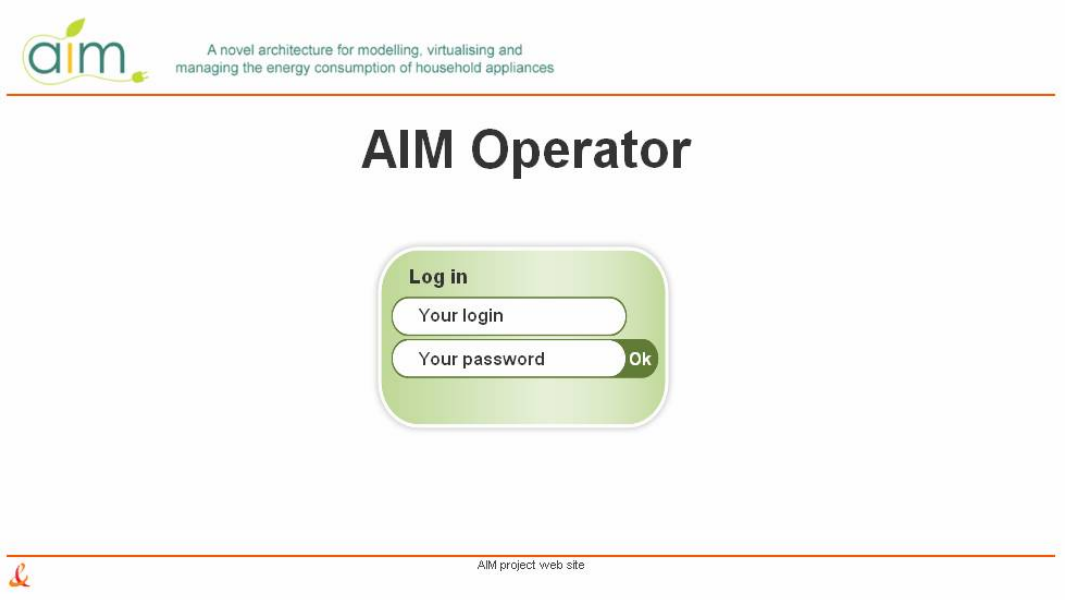

Figure 12: AIM operator login

5.2.3 AIM_Mobile application

The application is installed manually on the mobile by the Telco representative.

Figure 13: AIM Mobile interface

6 Utility platform Configuration

6.1 Requirements

6.1.1 Basic System

The AIM utility system (AUS) runs on a standard x86 PC with moderate performance. Indication for requirements are as follows:

- 2 GHz Single-Core CPU
- 1 Gigabyte of RAM
- 10 Gigabyte of hard disc space

Since the AUS is based upon Glassfish, a Java EE Server, it is platform independent. An installation is definitely possible on most Linux distributions, Microsoft Windows and MacOS operating systems for PCs. The further requirements and installation steps are based on an Ubuntu Linux installation. The installation of Ubuntu Linux, Version 9.10, is a standard Desktop or Server choice. The only additionally installed software are the JDK (see section 6.1.2) and the database (see section 6.1.3) with the dependant software.

6.1.2 JDK

Since the AUS is based on Java, it requires a standard Java Runtime Environment. The Runtime Environment has to conform to the Java JDK 6 definition. It is advisable to use the Sun Java JDK6 which is available from the Ubuntu software repositories.

6.1.3 PostgreSQL

For data storage the AUS requires the availability of a database. All SQL databases with a JDBC connector module can be used. For this installation (see section 6.2) a PostgreSQL database is used, which is also available from the Ubuntu repositories. The installation of the afore mentioned JDBC connector module is described in section 6.2.

6.1.4 JDBC Connector

A JDBC Connector is needed to provide database access from Java to any database. For PostgreSQL the JDBC connector can be found from the jdbc.postgresql.com website [2]. The Glassfish requires a JDBC connector in version 3. This .jar file can either be placed directly into the Java classpath directory or put anywhere on the hard drive. In this case it needs to be added to the classpath used by Glassfish.

6.2 Installation steps

6.2.1 PostgreSQL

The Postgres database has to be set up for the AUS. This requires creating a new database user for the AUS and setting up a new database for this user. The anonymous login has to be enabled by configuring the database as a trusted user.

6.2.2 Glassfish Installation

The Glassfish application server can be obtained from glassfish.org [3]. Installation steps for Glassfish can be obtained from [4].

After successful installation of Glassfish the server and its default domain “domain1” can be started with `asadmin` from the `bin` directory:

```
Asadmin start-domain domain1
```

The Glassfish frontend is now available via webbrowser and the URL: http://localhost:4848/

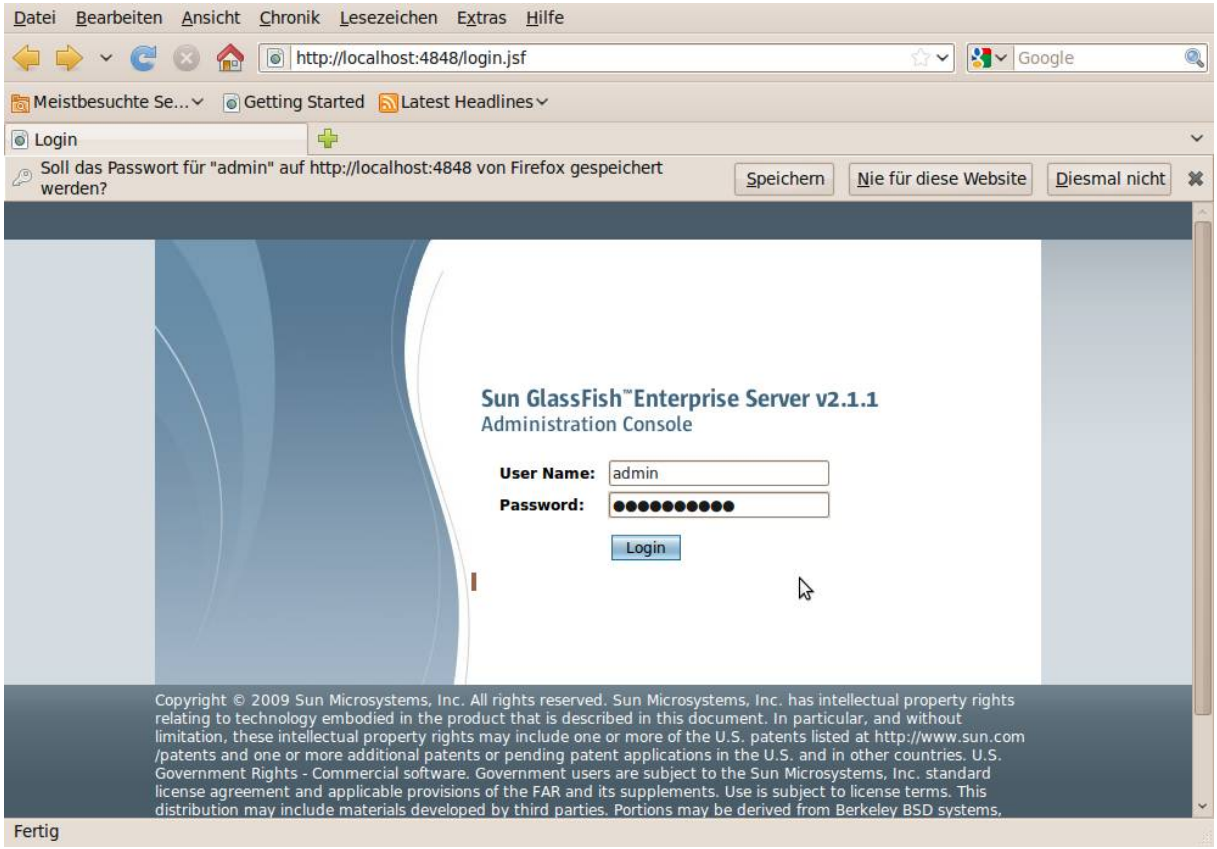

Figure 14:Glassfish login screen

6.2.3 Glassfish configuration

The Glassfish configuration is done in a browser via the web front-end as shown in Figure 14. The standard username and password is *admin* and *adminadmin*. These should be changed immediately for security reasons.

6.2.3.1 Configuring the classpath

Once the Glassfish is running correctly, the first step should be setting the classpath in Glassfish, if the JDBC connector module was not inserted into the Java classpath directory (see section 6.1.4). The classpath setup can be found under Application Server > JVM Settings > Path Settings (see Figure 15). Add in the System Classpath window the complete path to the JDBC .jar file e.g.:

```
/opt/glassfish/postgresql-8.4-701.jdbc3.jar
```

and save.

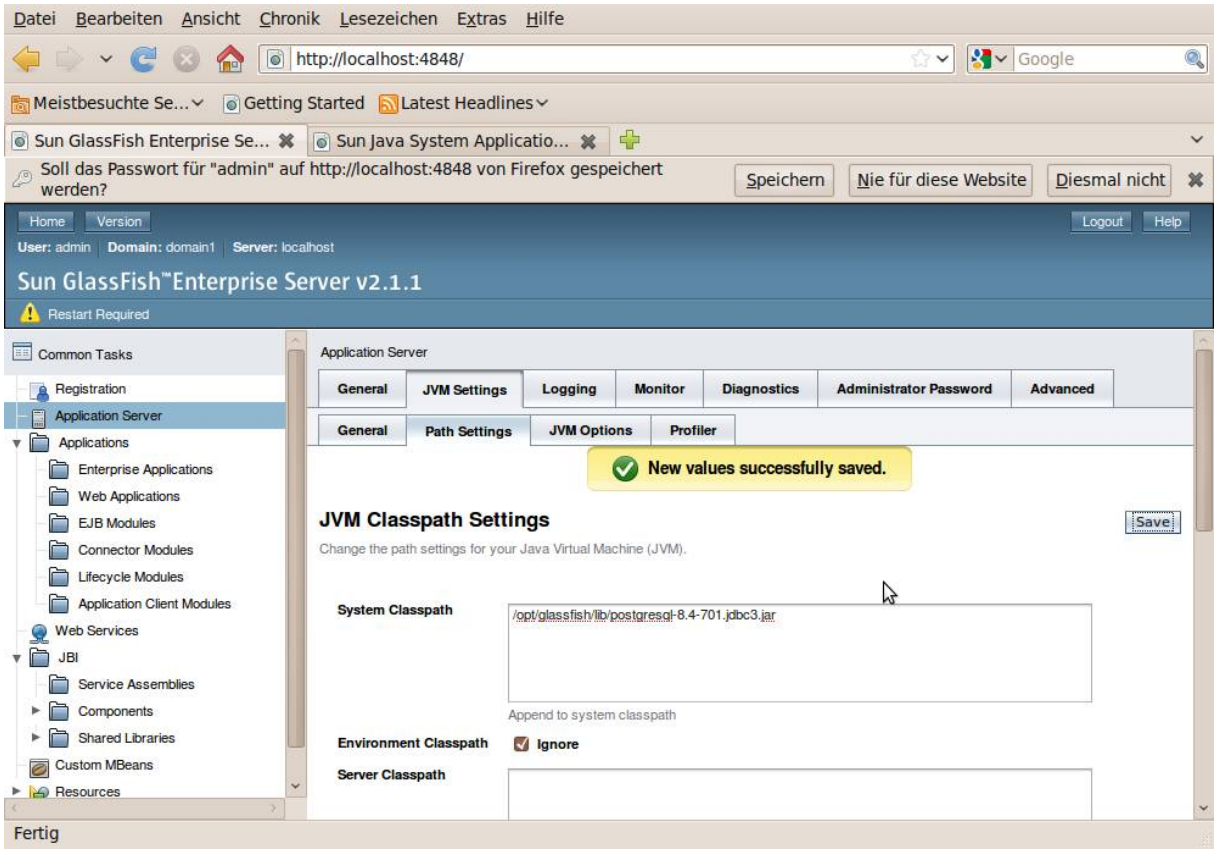

Figure 15: Setting the Classpath for JDBC in Glassfish

6.2.3.2 Setting up the connection resource

Adding a connection resource to the connection pool (see Figure 16) enables the Glassfish application server to communicate with the database through a standardized JDBC interface. Clicking on the Button “New” will open a configuration page (see Figure 17). In this first configuration page the name of the new resource is set and the database vendor and SQL interface chosen. On the second page (see Figure 18) the parameters for database access are entered. Even though the database can be accessed without a password, a password parameter is needed.

After having set up the database as a new resource for the application server, it has to be mapped to the JDBC interface used by the AIM utility application. The JNDI Name used in Figure 19 is defined as an interface in the application source code. Once the JDBC resource is set up completely it can be tested from the JDBC resources page.

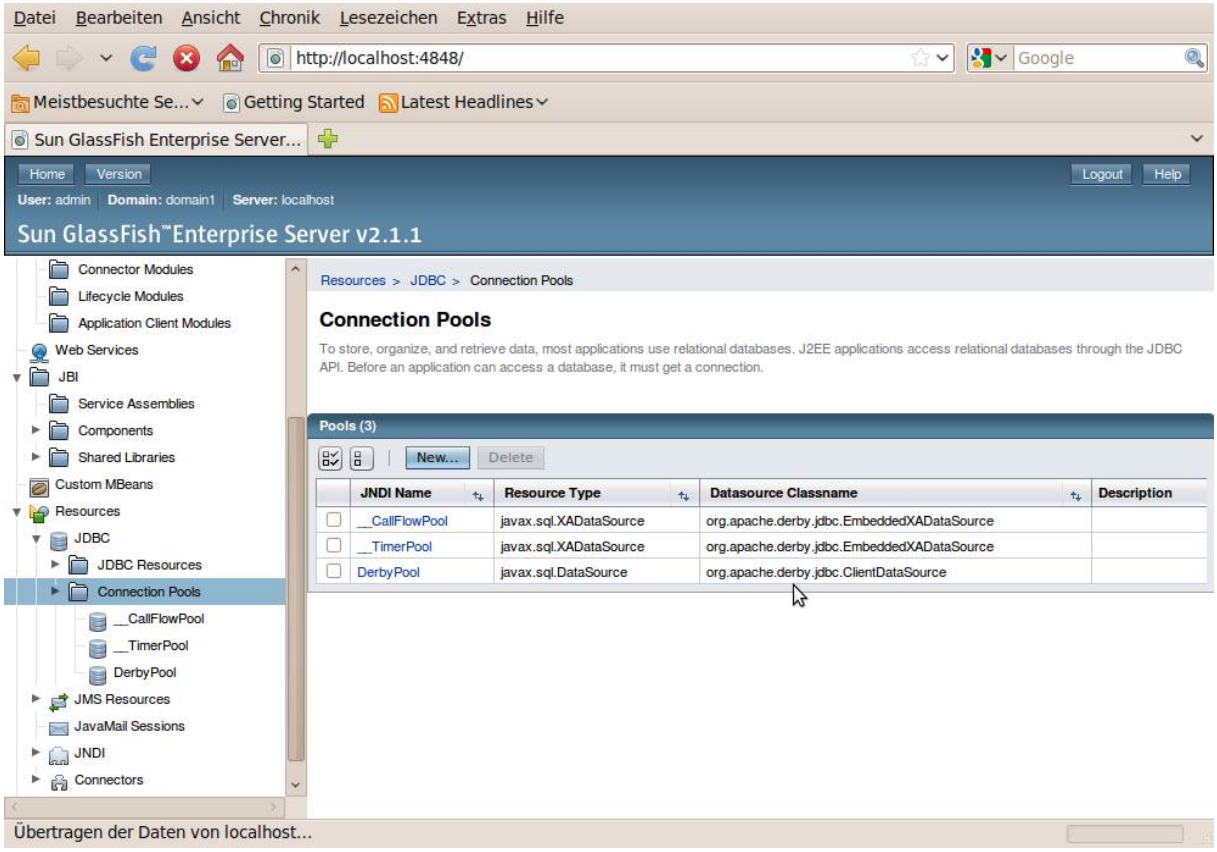

Figure 16: Connection Pools Configuration Page

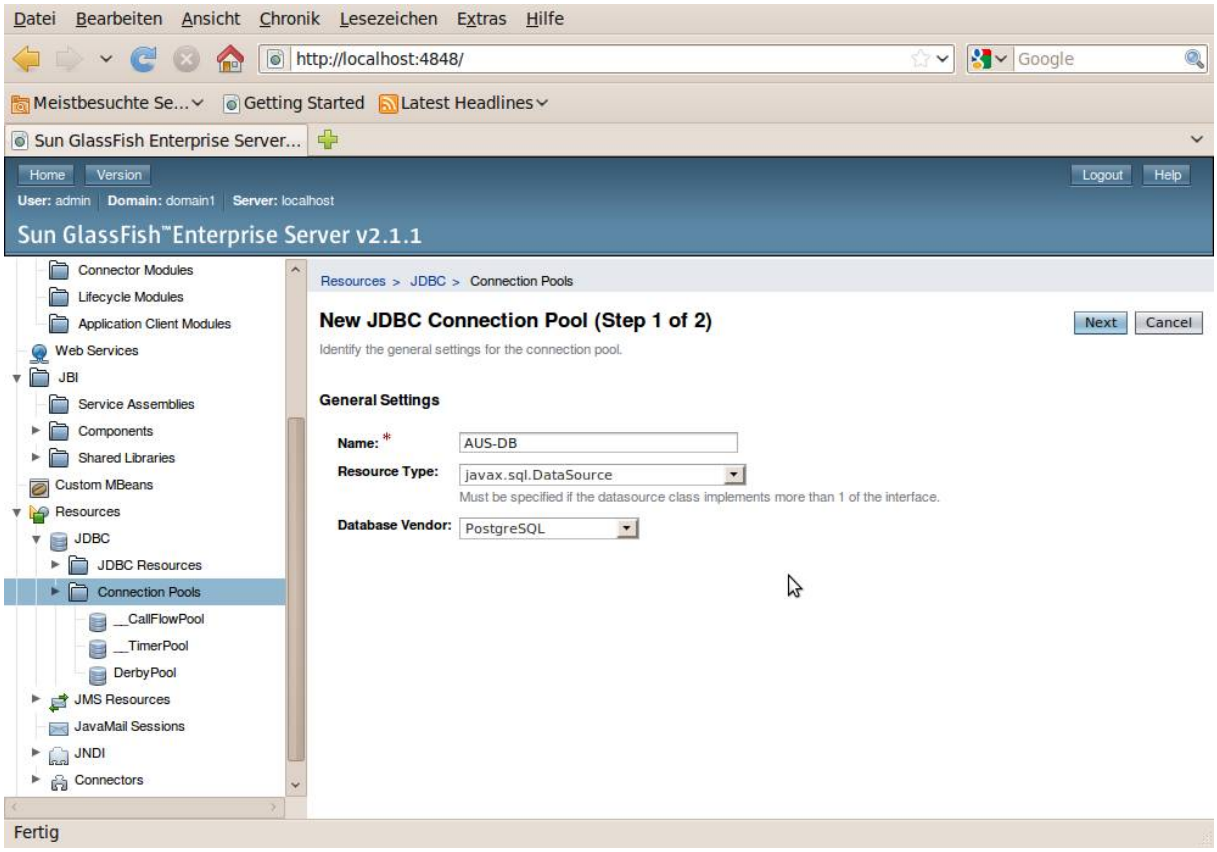

Figure 17: Creating a new connection resource for an SQL datasource with Postgres DB backend

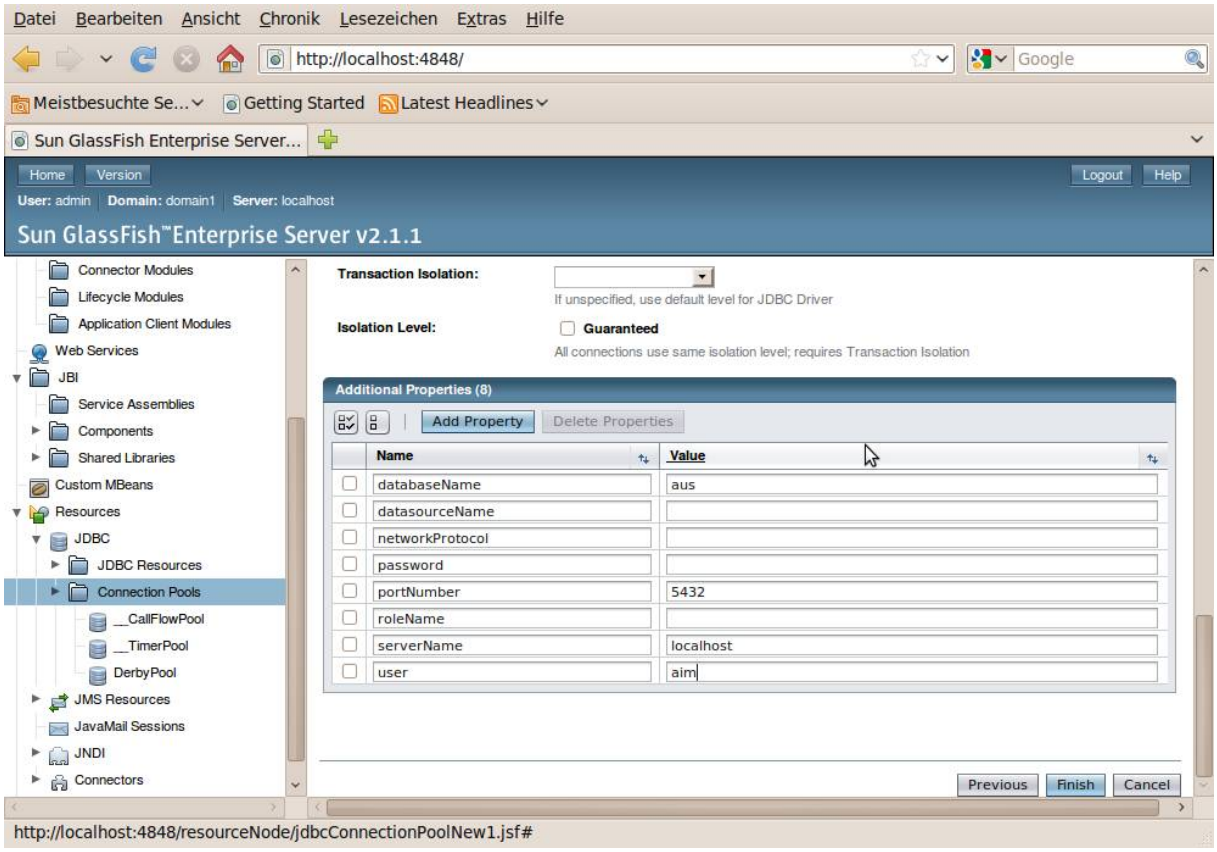

Figure 18: a new connection resource: Setting the access parameters for the Postgres database

6.2.3.3 Deploying the AIM Utility System

With the previously described steps to set up the database connection it is now possible to deploy the AUS on the Glassfish. The screenshot in Figure 20 shows all information necessary for the deployment:

- Application type: Enterprise Application
- Location of the application: the path to the application binary
- Application name: name for the application (here AUS)
- Virtual Servers: context in which the application should run. Since no additional servers were created, the default is used.
- Enabled: Check to enable running the application

Once this information is entered and saved, the AUS is ready for use.

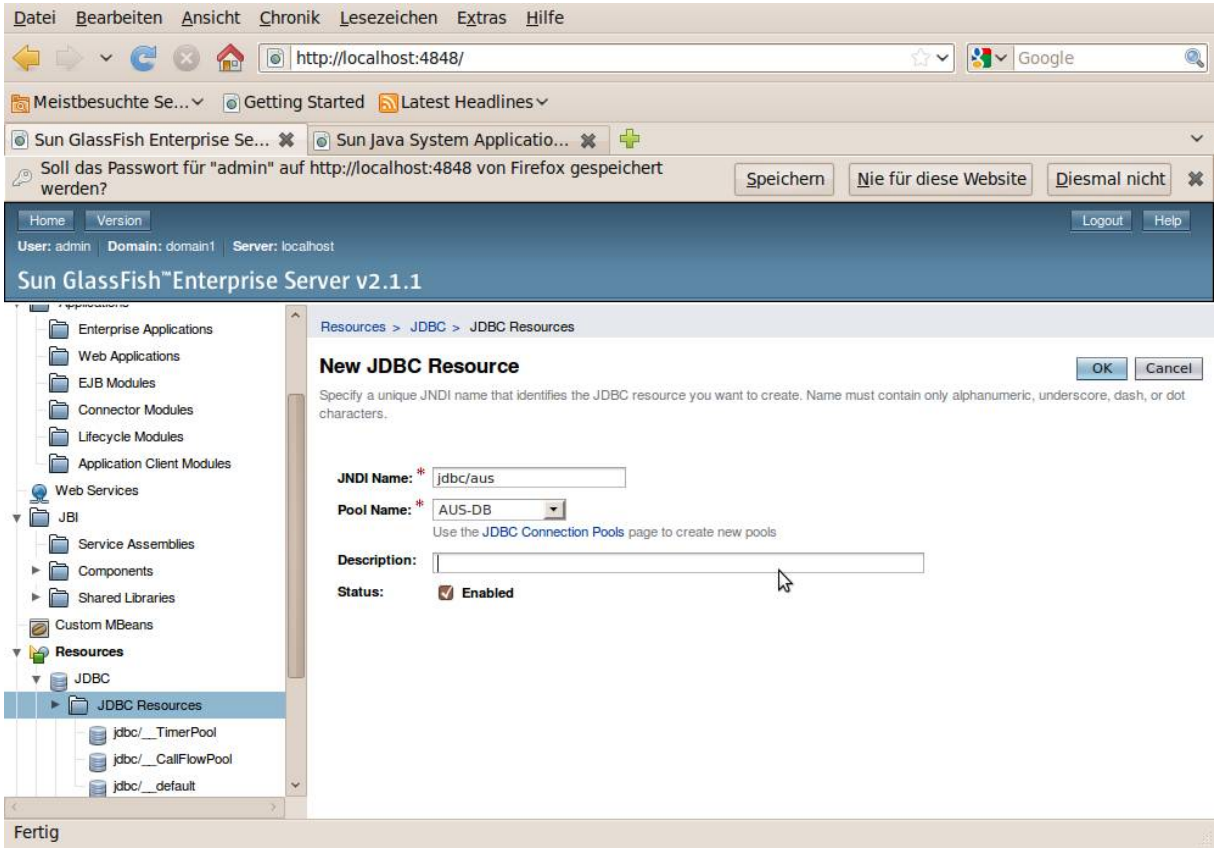

Figure 19: Mapping the resource to a JDBC name

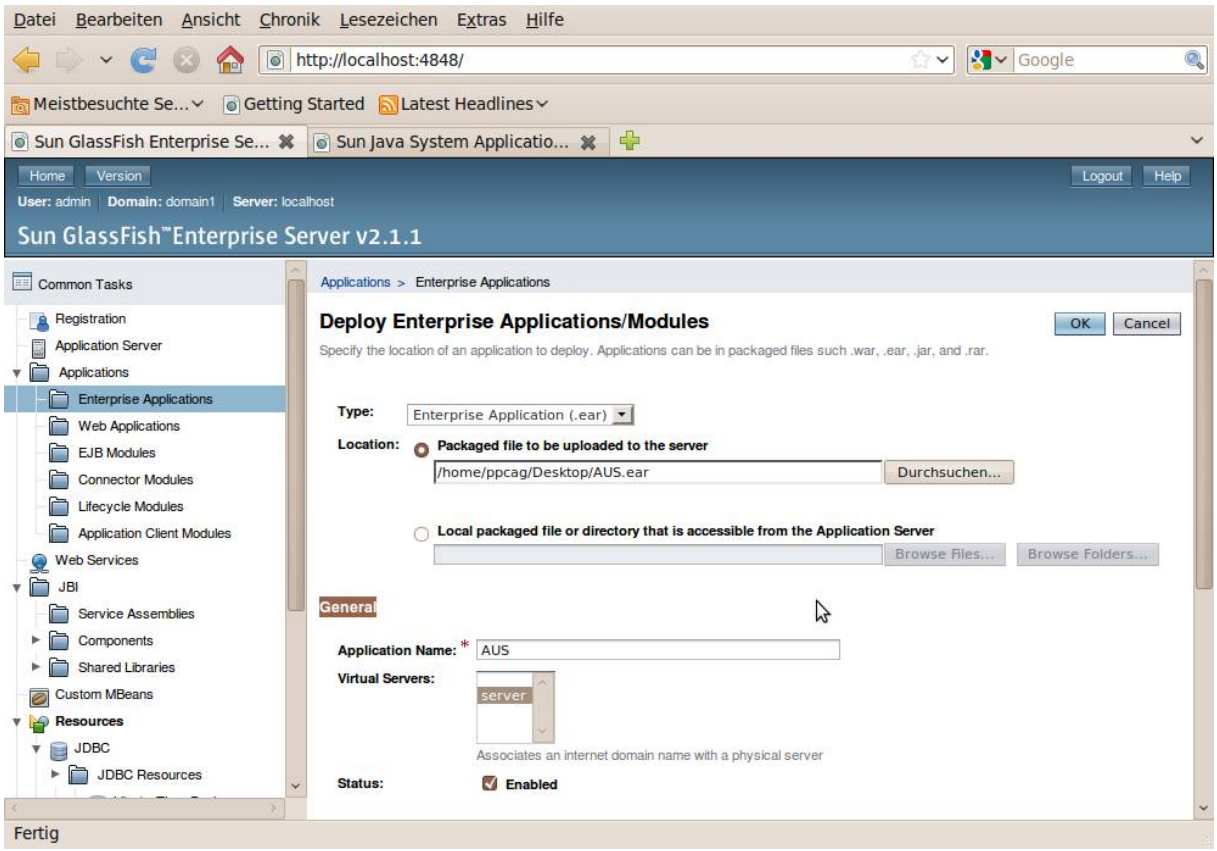

Figure 20: Deploying the AUS as an Enterprise Application on the Glassfish

7 User manual

7.1 Local GUI User manual

This chapter includes the DVE usage instructions for end users. Each section of the local Graphical User Interface is described with the available functions.

The picture below shows in blue the pages of the interface accessible by all the users, in green the pages accessible by the administrator and the advanced users and in orange the pages only accessible by the administrator user.

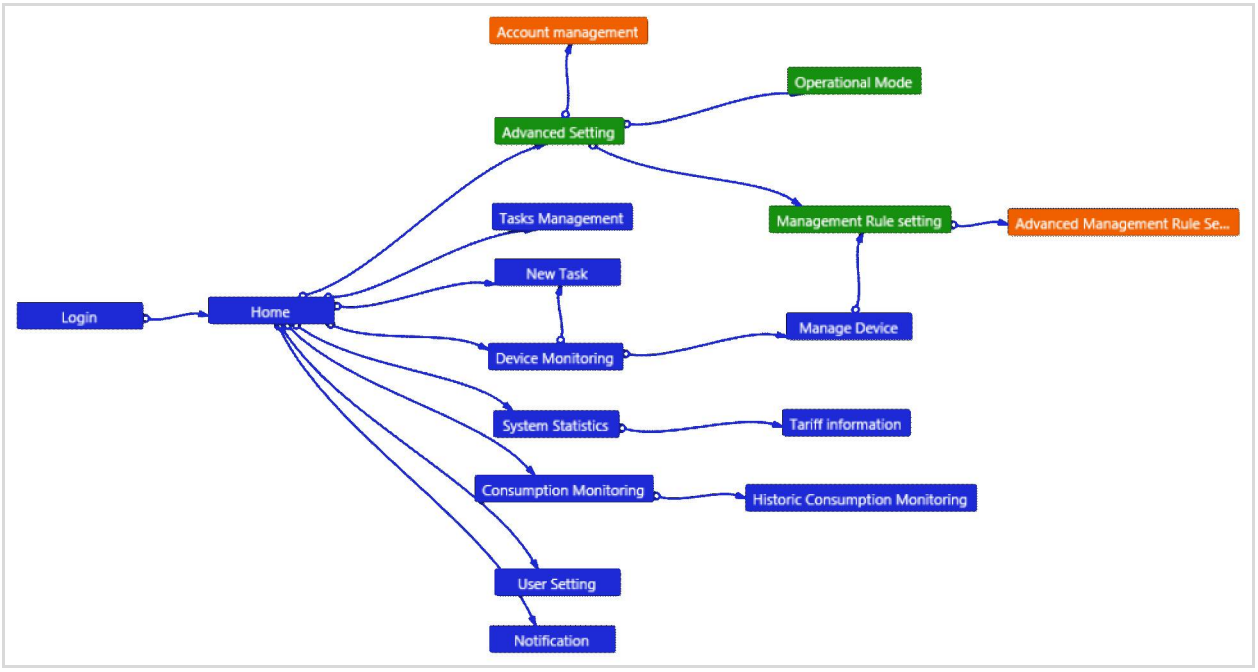

Figure 21: navigational three of the local GUI

7.1.1 Login

This page allows the user to authenticate him in the system inserting its own credential (login and password). In case of error the user will be prompted to verify its data and reinsert it in the correct fields.

Based on predefined roles for users, the authenticated user may will not have access to some pages (see Figure 21 for details)

Figure 22: Login Page

7.1.2 Homepage

The homepage is a domestic environment overview that is made of three elements that helps the user viewing the entire system status:

- A box containing a summary of the most important data as the one related to the consumption in real time or the active operational mode in the system (Basic, Green, Comfort, Cost minimization).

It contains also information about the user and its role (basic, advanced, administrator)

- A graphical overview of the house with the rooms and the devices connected to the AIM system
- A notification box including latest messages generated by the DVE system logic

From the homepage the user can access all other section available in the local GUI:

- New task
- Tasks Management
- Managed Devices
- System Statistics
- Consumption Monitoring
- System Settings
- User Settings

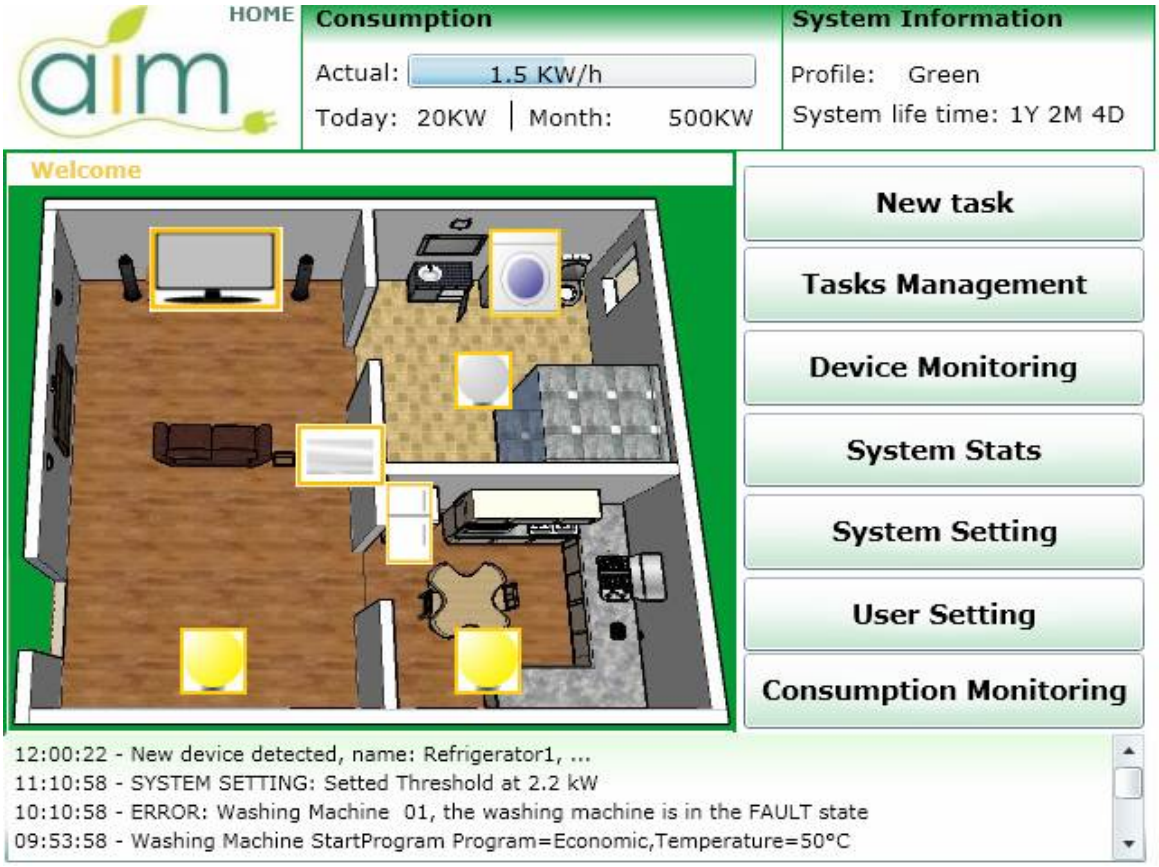

Figure 23: Homepage

7.1.3 New Task

Accessing this section, the user will be able to select on which device he wants to program a new task so it will see the list of the devices actually connected to the AIM system.

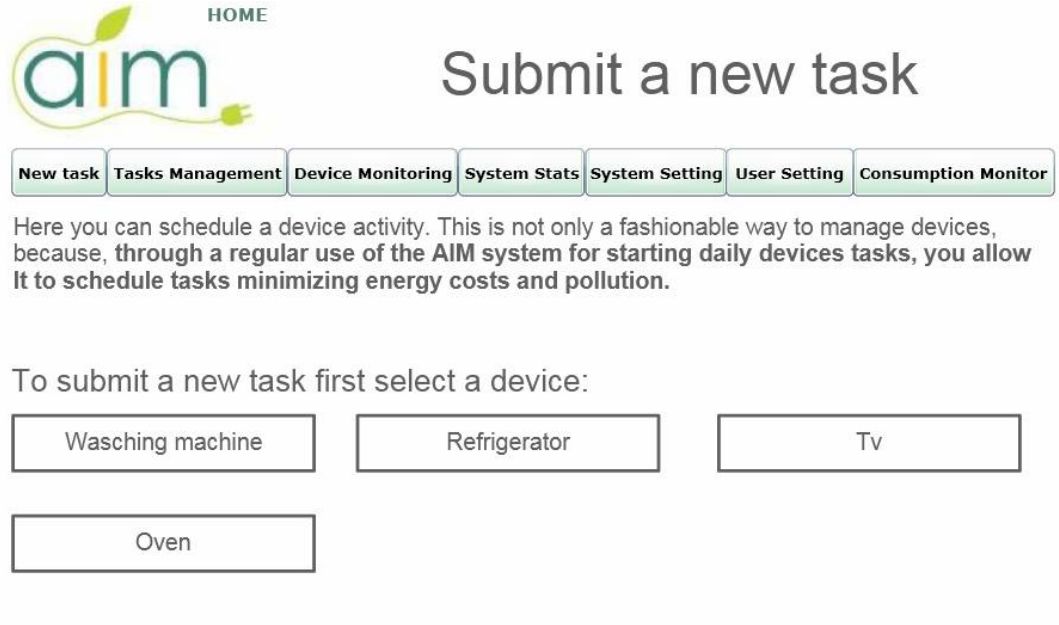

Figure 24: Selecting a device for creating a new task

The second step of the program process will ask to the user to select on the desired device the program to perform and the parameters that are requested eventually.

The user MUST set the deadline for the task to be completed.

The AIM system logic in the DVE will perform evaluation of the data given by the user and will calculate the best solution to execute the task based on the active operational mode.

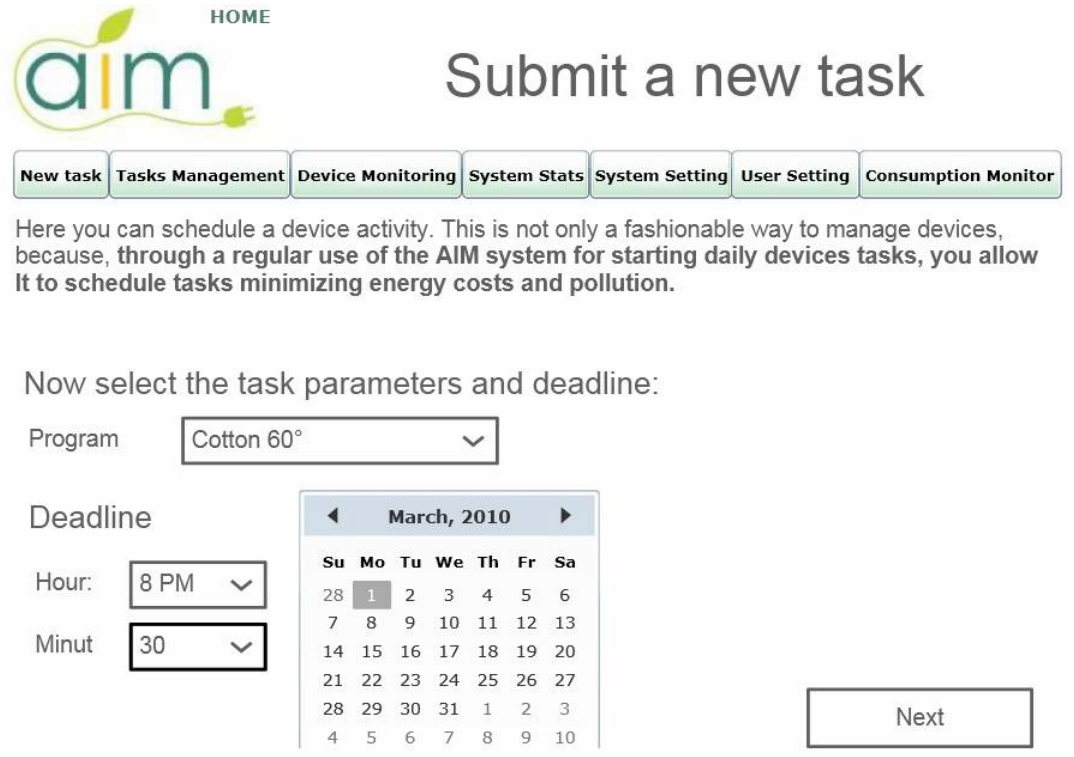

Figure 25: Setting task parameters and deadline of a new task

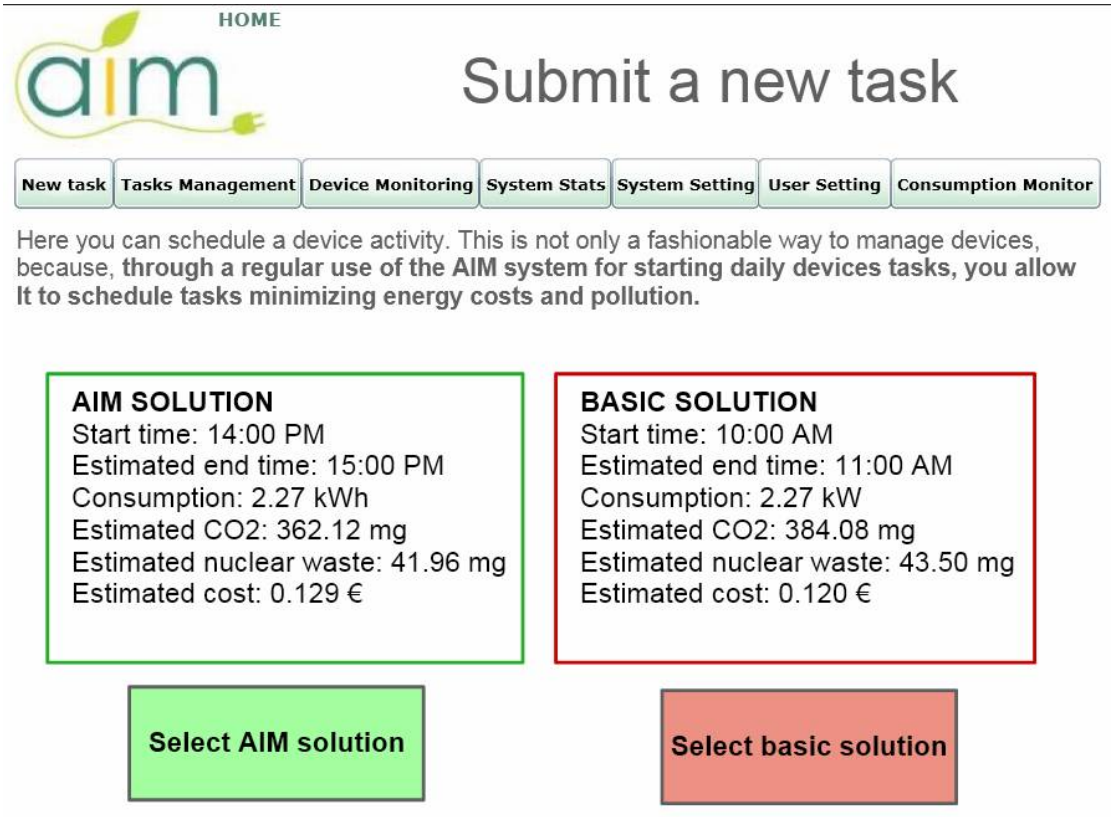

Figure 26: Result of the programming of a new task

The user can choose between two solutions available:

- AIM Solution – with energy saving
- Basic Solution – without energy saving

He will see the comparison between the solutions so he will know how much each solution costs in term of energy costs and emission.

When the user confirm a solution pressing the green or red button, the system saves it in the database and it will take care to start the task at the calculated time.

If the system detects an error while programming the task, possibly due to some conflicts with other tasks, the system will propose two new solutions to be confirmed by the user.

7.1.4 Scheduled Task

This section includes a graphs that shows on the x axis the time (hours) and on the y axis the devices programmed consumption. The graph is divided in time slot and will show for each time slot the programmed task and relative consumption in that slot.

Each task has different colours in order to be easily identified from the others. The user can select a task and he will see details a page about the selected task. At this time the user will be able to edit the task or to remove it.

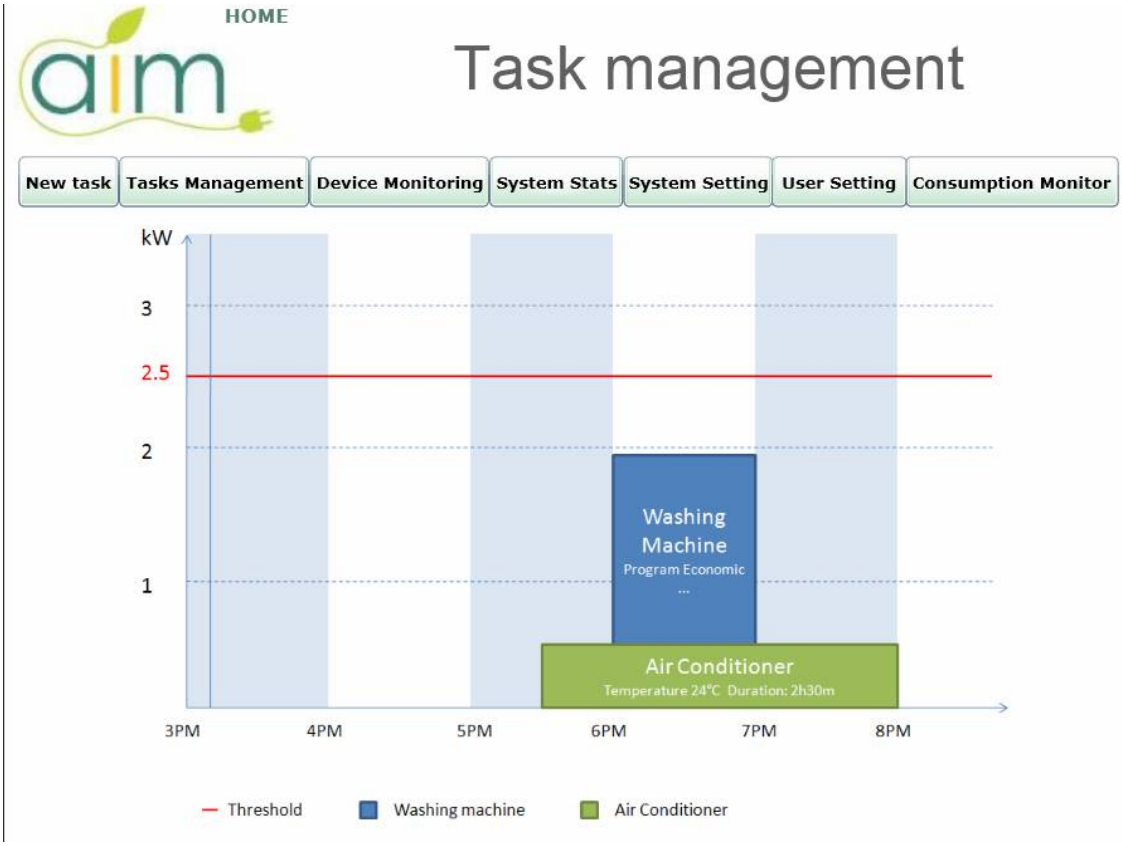

Figure 27: Scheduled tasks view

7.1.5 Consumption Monitoring

This section includes a time-consumption graph, divided in time slot, for the actual day. It shows the consumption for each hour of the day of all the tracked devices. Last monitored hour is the one before the actual one.

From this section a user can access the sub-section relative to the historical consumption data.

7.1.6 Historic Consumption Monitoring

This sub-section still includes a time-consumption graph, divided in time slot, for the actual selected time. It shows the consumption for each slot of time for all the tracked devices.

It is possible to specific of which device the user wants to see the graphs. The list of connected devices is available at the bottom of the page below the graph.

The user can change the view of the graphs in order to shows data related to the last year consumption, last month, last week or last day.

It is also possible to show on the graphs either the costs sustained or the CO_2 emissions generated.

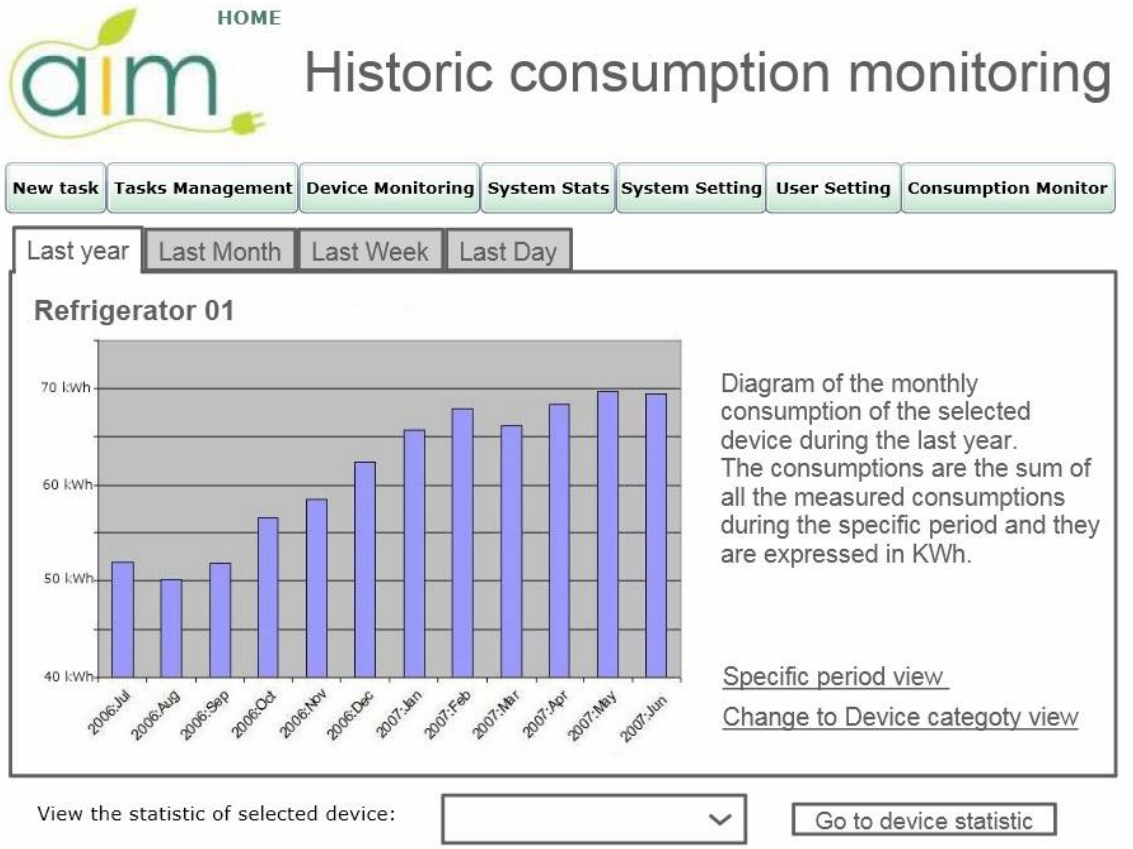

Figure 28: Graphs of historical consumption monitored

7.1.7 Device Monitoring

This section includes the list of all connected devices and for each one of them it shows the state. User will be able to see where a device is located.

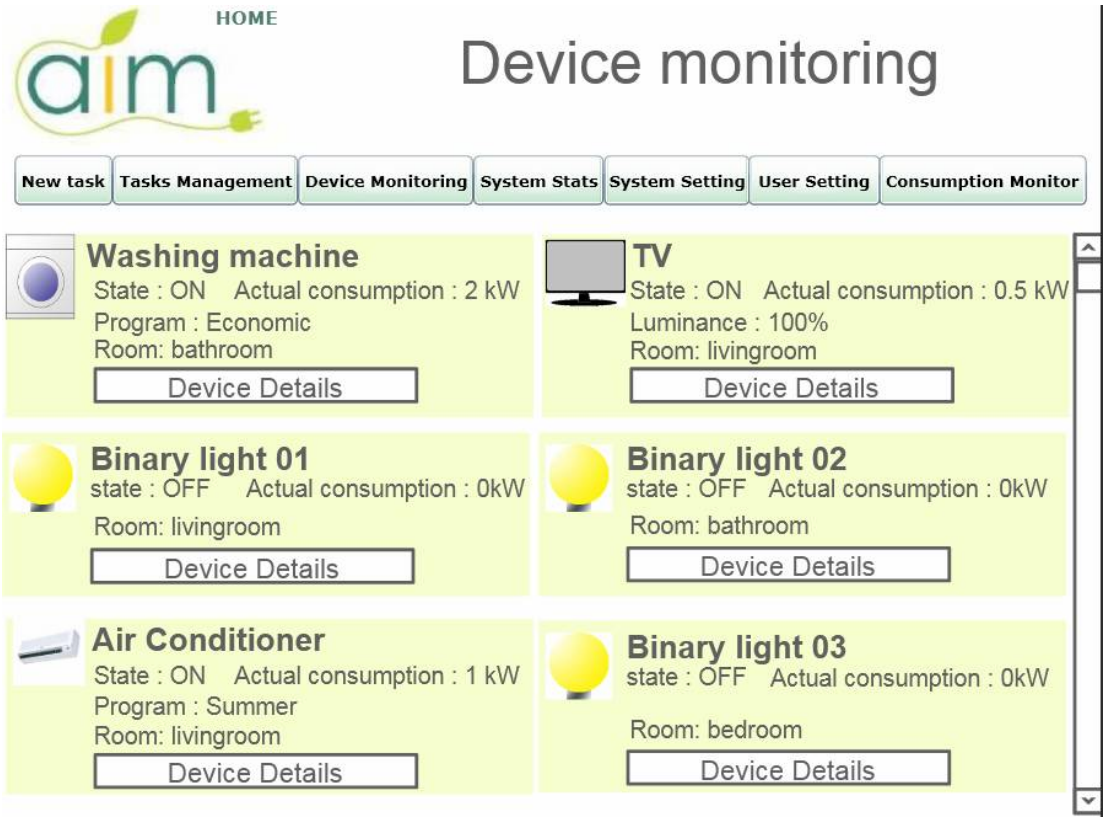

Figure 29: Connected devices list

In this page, all users are able to see detail of the selected device. They can see the name, the type, the actual consumption and status of each device. All programmable devices contain also information related to eventually programmed task on them with their start and end time.

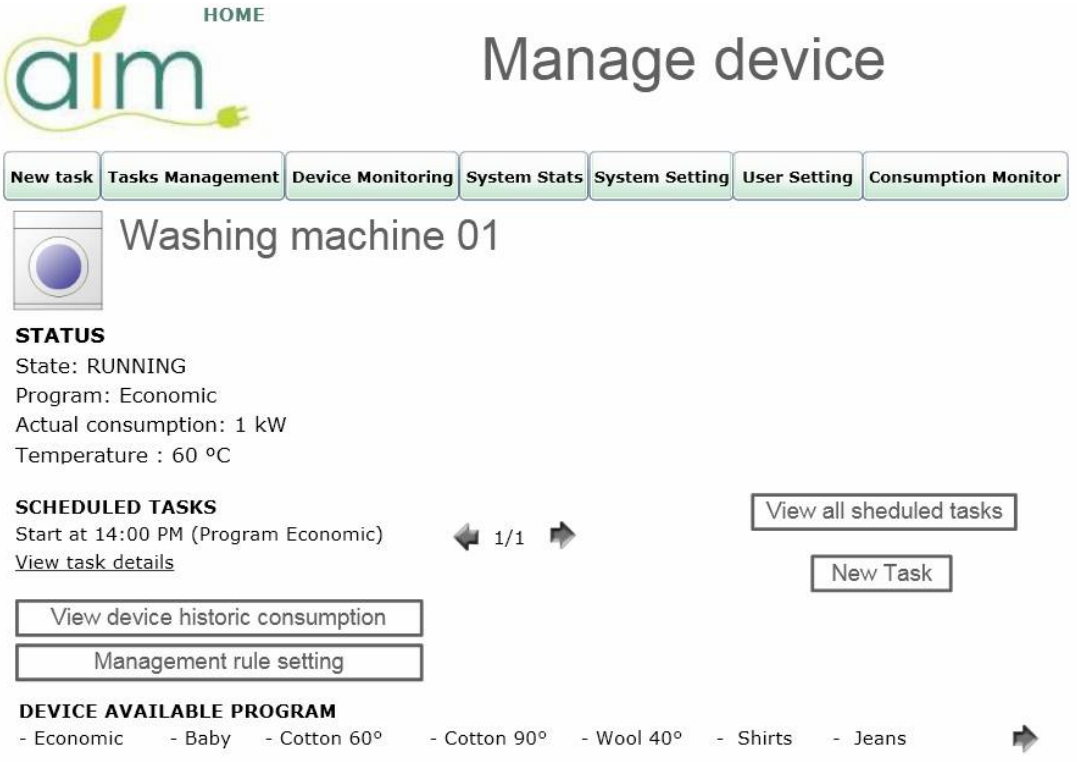

Figure 30: Device details

Advanced users or administrators are able to modify the rules for a device, for example the association of the device with a room so they are able to access the Management Rule Settings page.

7.1.8 User Settings

This section allows the user to edit some information about its account in the system and set two main environmental parameters like the luminance and the temperature desired in each room while he is in the house.

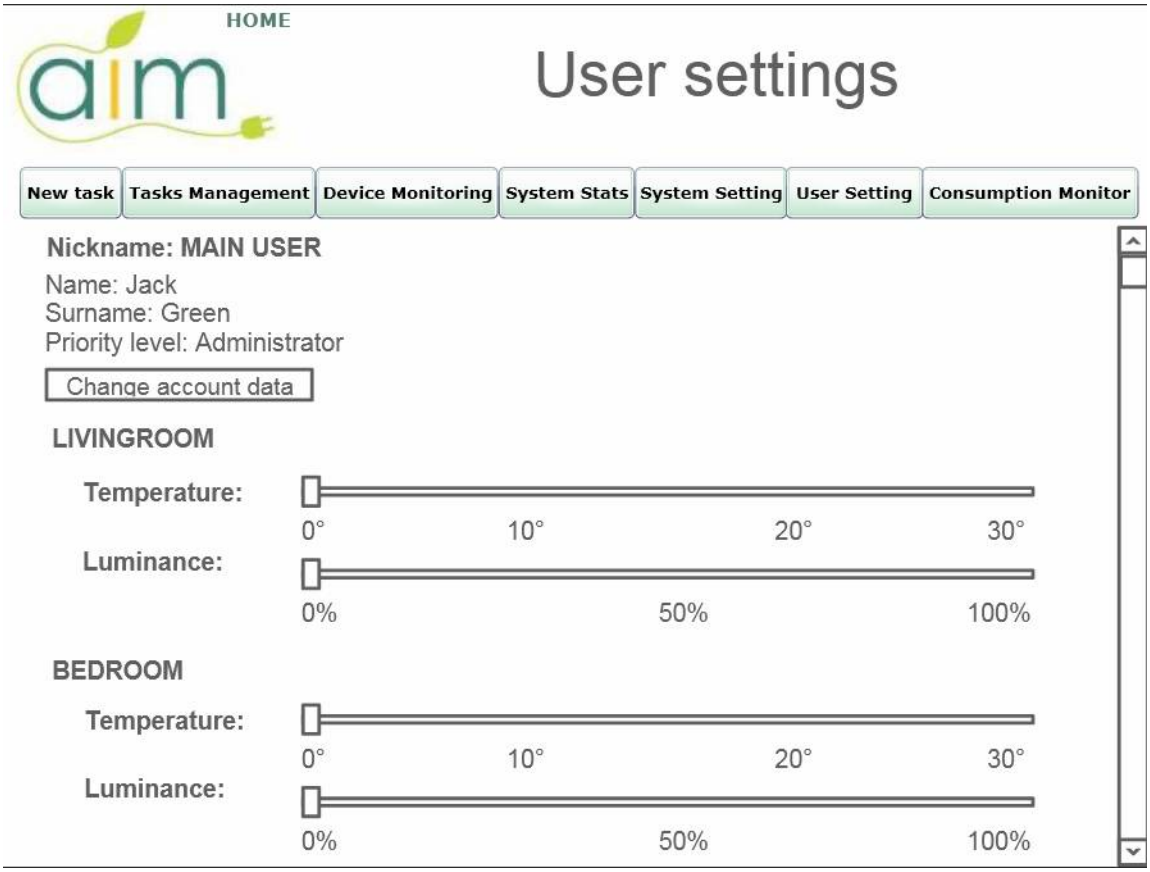

Figure 31: User settings

7.1.9 Notifications

Notifications section includes the list of all notifications generated during the actual day.

Each notification include the time at which it has been generated and a description of the event.

There are two types of notifications:

- Abnormal situations
- Event

Errors can be generated, for example, if a task has not been started at the time at which it has been programmed.

Event notes are generating when an event occurs, for example the starting of a scheduled task.

7.1.10 System Statistics

This page is a summary of all the benefits obtained using the AIM system. It shows the comparison of the data between the house under control of the AIM system and the house without it.

All users are able to see the amount of energy saved using AIM system from the startup of the system and also the emissions generated. For each device the users are able to see how much they saved using AIM.

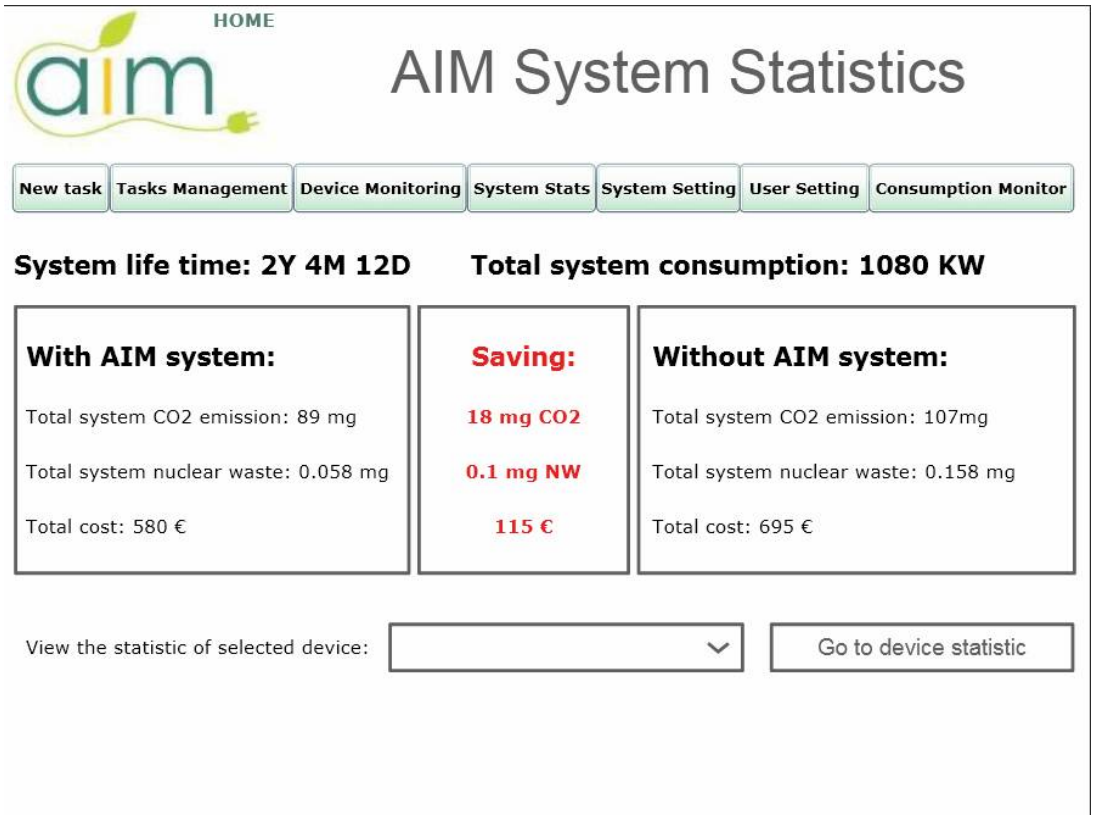

Figure 32: System Statistics

7.1.10.1 Tariffs Information

In this sub-page users can find two graphs, the first one shows the tariff in each time slot in terms of KW, second one that shows the emissions of CO_2 in each time slot. Both graphs have a resolution of 15 minutes.

7.1.11 Advanced Settings

Only advanced users and system administrators can access this main section.

In this section users have access to some general information about the operational mode, the tariff information from the energy provider and so on.

From this section the users enabled are able to access the operational mode sub-section and the Account Management sub-section.

7.1.11.1 Management rule setting

In this page the users enabled can edit, enable or disable the rules for the management of the devices. For each device there is the list of the rules related to it. Users are also able to add new rules or to

delete existing ones. Only system administrators can access from this page the advanced management rules settings.

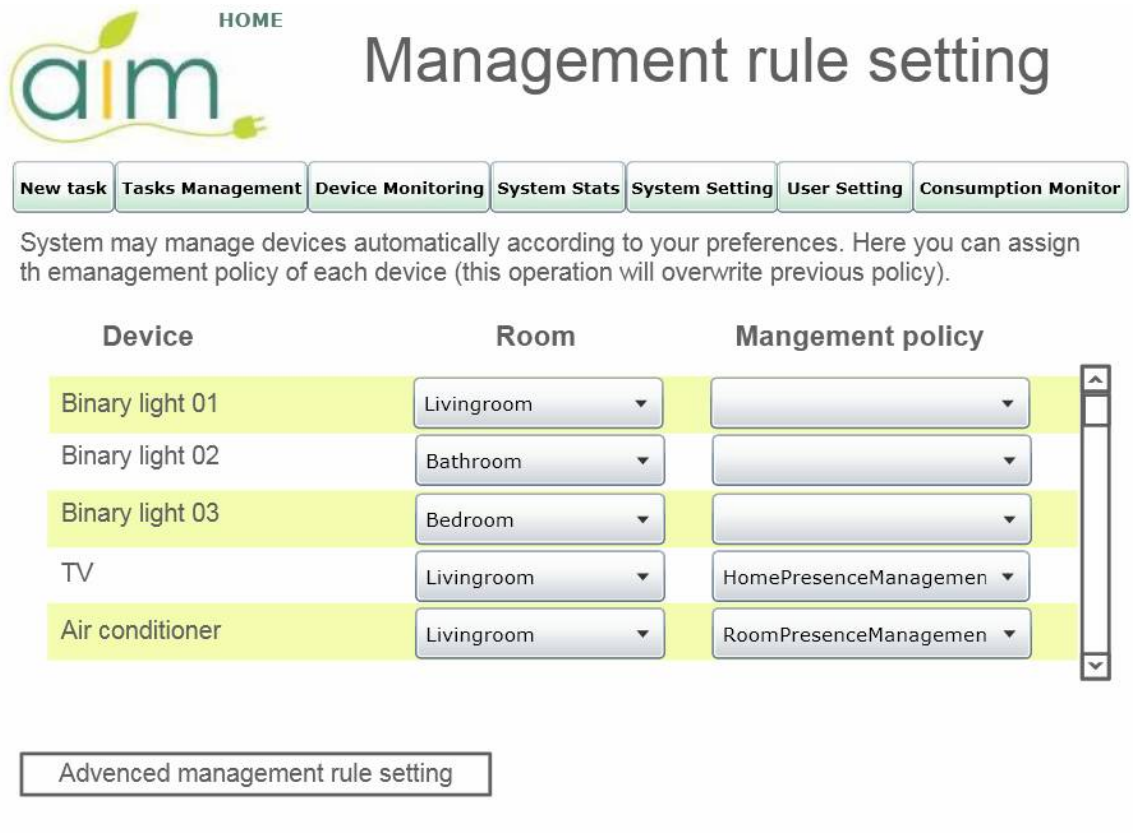

Figure 33: Rules management of devices

7.1.11.2 Advanced management rule setting

In this page, accessible only to system administrators, it is possible to edit the default value of the rules for the device management.

7.1.11.3 Account management

In this page, accessible only to system administrators, it is possible to manage existing user accounts and create new ones. For each account the administrator is able to set the role of the user (basic, advance, administrator).

7.1.11.4 Operational mode

In this page, reserved to advanced users and administrator, the users are able to change the operational mode of the entire system. They can see the active mode and change it between green, cost minimization, comfort and on vacation.

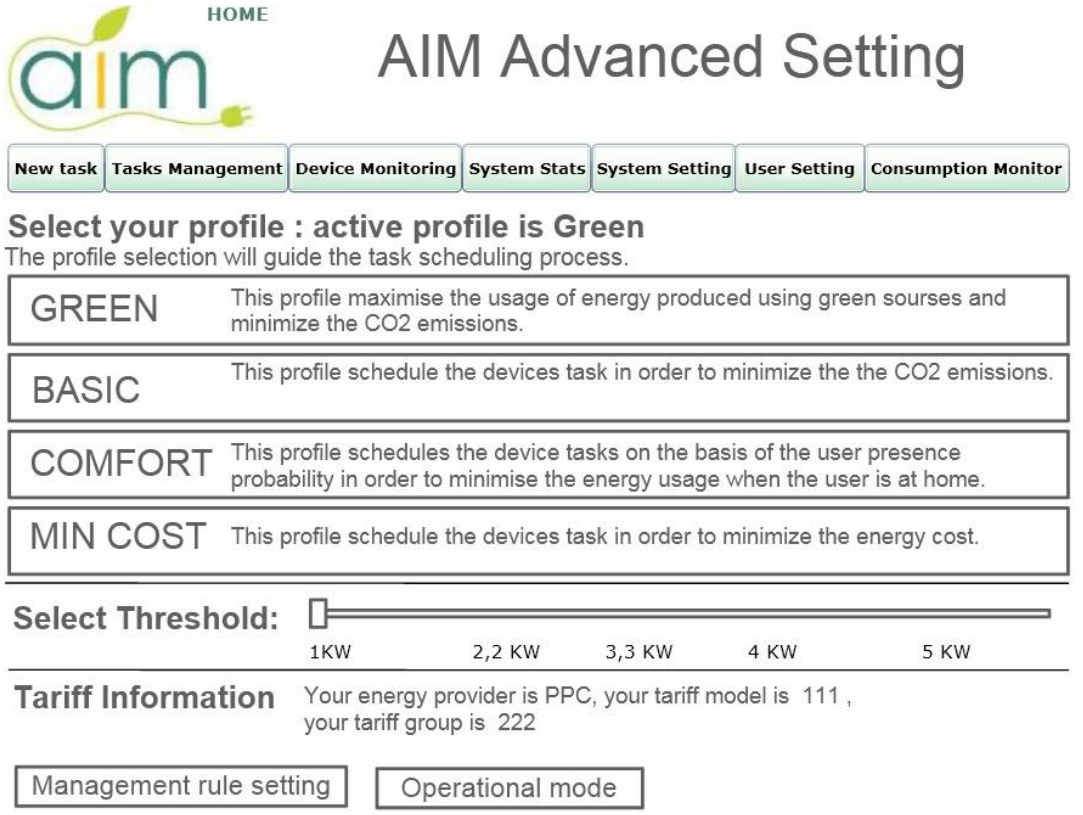

Figure 34: Operational Modes

If the user decides to activate the "on vacation" mode, it will be prompted to a page where it will be asked to set the period of vacation during which the system will manage the house ignoring the profiles of presence of the users

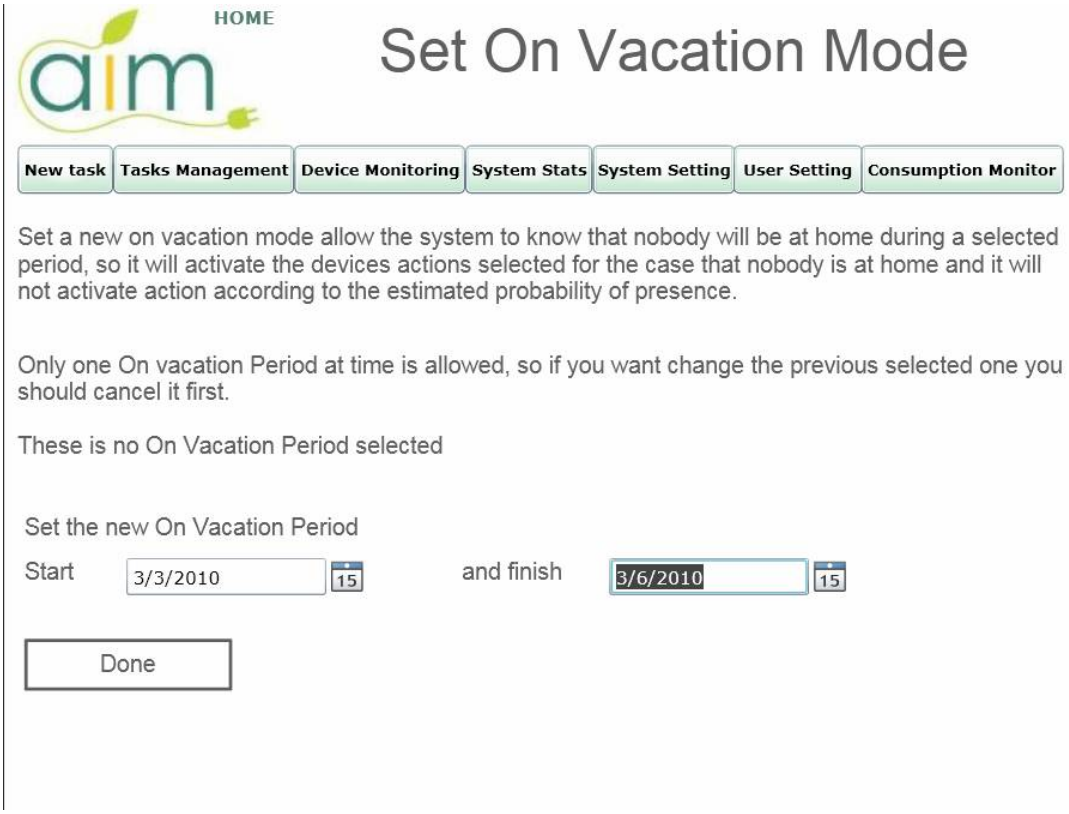

Figure 35: "On vacation" operation mode

7.2 Mobile GUI User manual

The mobile interface was designed by ergonomic experts to be intuitive enough for technology early adopters. It is based on touch screen technology.

The interface follows Android best practices :

- http://developer.android.com/guide/practices/ui_guidelines/index.html

A carrousel allows the user to select functions :

- to monitor and control appliances,
- to receive alerts to in case of abnormal consumption (eg when the door of the fridge stays opened)
- to configure his AIM energy saving profile ("green" , "mincost", or "basic")
- to promote energy saving through his/her community.

8 Conclusion

This deliverable contains the installation instructions of the prototype developed and a user manual with a detailed description of all DVE functionalities and settings. For all components of the use cases presented in the deliverable 4.1.1 there is a section in this document with the minimum requirements that the platform hosting the application has to provide and there is a section with the instructions to configure the component. Moreover we provide a description of the home user interface and of the mobile application designed to be easy to use and to allow a user to customize the DVE management logic in order to satisfy his needs.

An overview of the features of our system is presented and the user interaction with the DVE through different interfaces (the local GUI or the mobile application) is described. This document includes also a description of the mobile application that allows a users to access to his DVE when he is not at home with his mobile device through the telco platform. The mobile interface was designed by ergonomic experts to be intuitive enough for technology early adopters.

The prototype developed has to be tested in real home environments to evaluate the energy saving capabilities and usability. The introduction of the AIM system in a home doesn't intend to enforce people living there to change their habits, but it should improve comfort and cost reduction. The evaluation of the AIM system is part of WP5.

References

[1] http://www.ict-aim.eu/

[2] http://jdbc.postgresql.org/download.html

[3] http://download.java.net/javaee5/v2.1.1_branch/promoted/Linux/glassfish-installer-v2.1.1-b31g-linux.jar

[4] https://glassfish.dev.java.net/downloads/v2.1.1-final.html

[5] http://java.com

[6] http://tomcat.apache.org/

[7] http://www.mysql.com/

[8] http://www.wampserver.com/en/

[9] http://www.touchatag.com/downloads

[10] http://java.sun.com/docs/books/tutorial/rmi/index.html