EPRO 3
Renewable Energy System
Project - Energy Hub

Counselor: Morten Opprud Jakobsen & Klaus Kolle

BY E10 - Team3:

Theis Christensen (10691)
Dennis Madsen (90248)
Paulo Fontes (10484)

Table of Content

Chapter 1

Introduction

This report contains information about the different development phases used to create an energy hub being part of the renewable energy system, which has been the task in Pro3 & Pro4. Tools given in classes during the first and second year of the EDE (Electronic Design Engineer) education, were used to create the energy hub.

The report is written in a chronologic manner, the first chapter describes the goal of the project and what would be the final product (pre project).

The next phase of the project development is listing requirement, creating overall pictures of the system, defining protocols and the design of the system (launch phase).

The creation of the product is described in time boxes, where different parts of the system will be described in a different time boxes. If something has changed in a part of the system, it is described in a later time box (after a time box has finished the section is locked and changed has been written in a following time box). The final stage of the project is handing over the product to the customer.

Chapter 2

PreProject

2.1 Rich Picture

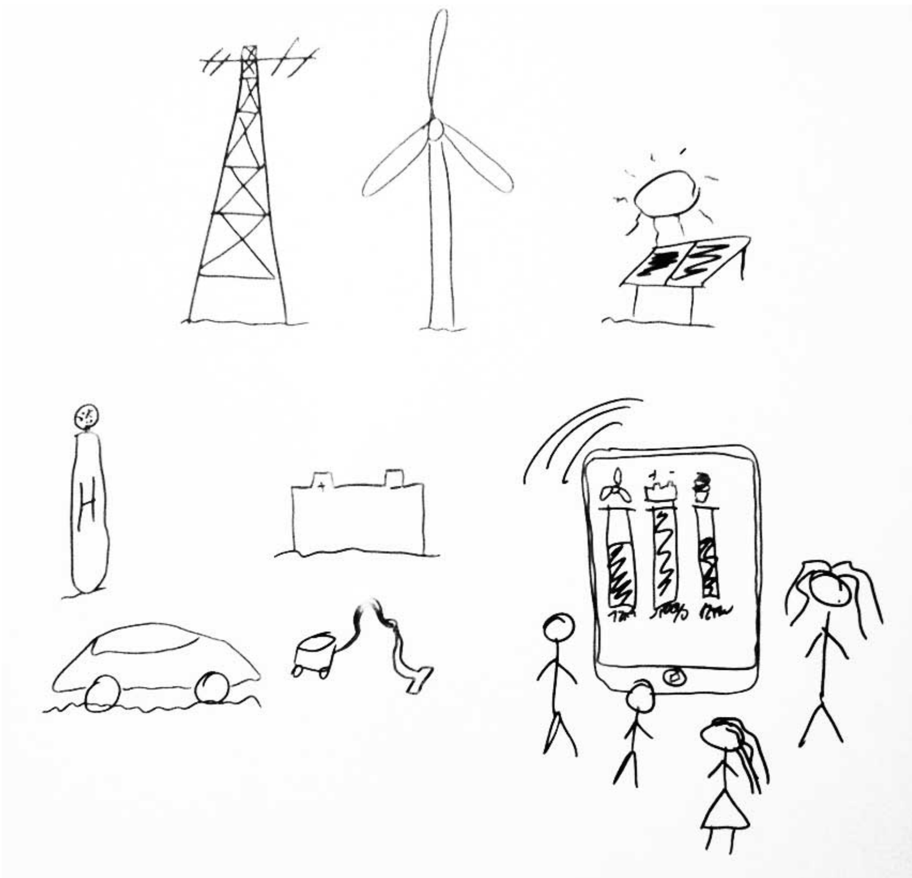

Figure 2.1: The surrounding environment is shown in the rich picture. The modules are: photovoltaic cells, CAES , battery-charger, wind-turbine converter, user-interface, the electric grid and different loads such as a electric engine, vacuum-cleaner etc.

2.2 Story Telling

Jan Nielsen just got a new energy module to connect to his system. Luckily there is plenty of space for new modules in the systems energy hub. He connects the device to the hub, where from the administrator web interface he can easily start the module. The connection of the new modules is straight forward as all the module plugs are similar and the energy hub finds out by itself if it is an input module, output module or both way module.

Aarhus University - Herning, a place full of innovation and great ideas. Jan welcome a class of high-school students to the green system simulator. There is possible to see the amount of energy that can be harvested in the green energy system such as wind and solar energy. When the system produces to much energy, the energy is stored as compressed air. Here people can really come an get an idea about how much they can help the environment, but also their own finances, if they invest in green energy for their own house.

Guests can interact with the system on a screen interface, they can see how much energy each module produces or consumes. This is shown in a intuitive manner, where everybody can follow, even persons without no special education or courses in the energy field.

2.3 Story cards

Story Card 1: Jan is looking at the web interface for the energy hub. From where he can see the status for each modules connected to it in a graphically way.

Story Card 2: A new energy module is connected to the hub, Jan opens the web interface for the energy hub, he logs in the administrator section of the system to start, stop or see a more detailed overview of each module. Jan really likes the graphically way that is also kept in the administrator section, this makes it possible for other non-technically persons to operate the system if Jan is not there one day that the system needs to be operated.

Story Card 3: It is one of many regularly autumn days in Denmark with rain and wind. The system is placed outside but still it operates perfectly under these weather conditions. To operate the system or see the status for one of the modules, it is not necessary for anyone to go outside to the module, they can simply log onto the system from their workstation pc or laptop.

Story Card 4: Jan arrives at the university in the morning and an email was send to him reporting a failure in the green energy system, he login to the administrator web interface, and he can see what the problem might be, and if it's possible to solve it directly on the interface.

2.4 Preliminary Use Cases

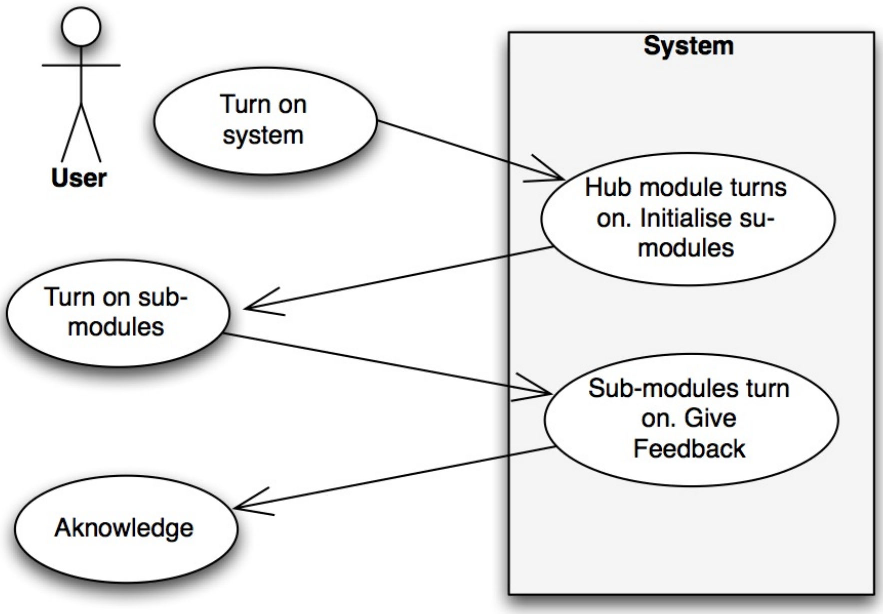

Figure 2.2: System initialization.

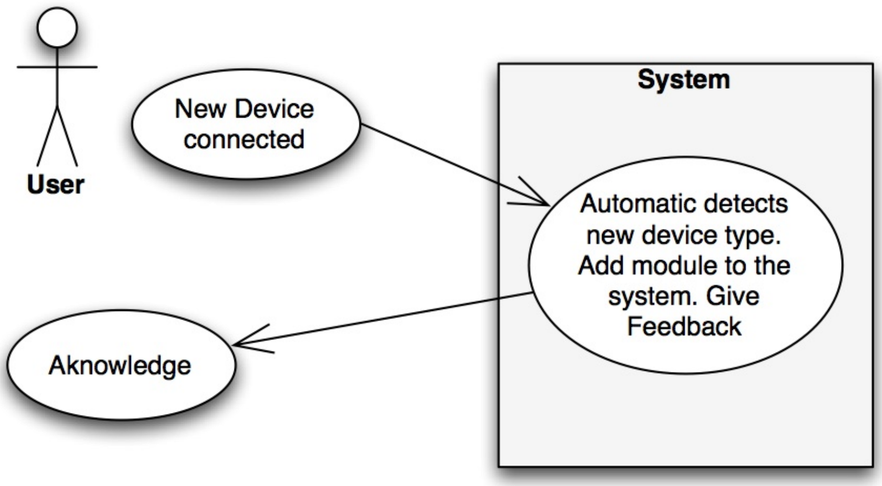

Figure 2.3: New device connected to the system.

2.5 Stakeholder Analysis

Stakeholder Analysis is an important tool for developers, as the different involvement of all person on the project and on the final product is clarified. This is done by identifying persons or groups which are relevant and their level of influence on the project.

Project coordinators:
Morten Opprud
Klaus Kolle

Customers/Users:
Jan Nielsen - Customer (Primary User)
High school students (Secondary Users)
Rene A. S. Josefsen (Web interface customer)

Suppliers:
Jens Mortensen
Per Lysgaard

Theory Advisors:
Henning Slavensky
Ulrich Bjerre
Kristian Lomholdt
Per Lysgaard

	Has decision power	**Has no decision power**
Directly involved stakeholder	Klaus Kolle Morten Opprud Jan Nielsen	Rene A. S. Josefsen
Not directly involved stakeholders	Per Lysgaard	Jens Mortensen High School Students

Figure 2.4: Stakeholder Analysis table

Klaus Kolle and Morten Opprud, are the project coordinators/managers, they have decision power over the final product and are directly involved on the development. As academic project, they are the persons that have to be satisfied with the final product. As project managers they guide the developers trough all the phases of the development process.

Jan Nielsen is the customer, the primary user of the system so he has the decision power over the final product and is enrol in all the development and design process.

Rene A. S. Josefsen is our web-interface customer, has no decision power in the overall system, but is direct involve in the design of the system interface.

In a real world project, Jan and Rene satisfaction as clients, would be very important. In this project their feedback is used as requirements for the final product.

High School Students have no decision power over the final interface since they are the secondary users of the system. They will be one of the final users to test the system.

Jens Mortensen is the component supplier, is not directly involved in the project and has no decision power over the final product.

Per Lysgaard is not directly involved in the development of the system, but has the final decision of the budget for the system.

2.6 System Definition

Proposal 1 - Fully automatically

The Energy-Hub is the central device in the green-energy system. All inputs and output devices are automatically routed inside this device. When devices are connected to the Energy Hub, it automatically detects if it's an input or an output module.

On the web interface, the user have the possibility of using between two modes:
- Green System profile.
- Fast charging profile.
The Green profile only makes use of the green power devices such as photovoltaic-cells, wind power, energy from the charged batteries or from the compressed air module. The fast charging profile on the other hand charges the compressed air module and the battery directly from the grid.

Proposal 2 (Optimising):
Extra output extension model: the user can connect devices, and switch each output on and off. This module will be used to present to high school students the interface navigation.

Chapter 3

Launch Phase

An initial analysis of the final system is made in order to estimate the development time it takes to develop the system. When knowing the development time of a product, this can be translated into an estimation of the total development cost. In the launch phase, everything not concerning the realization phase of the system have to be considered, in order to make a good time estimation. The black box which it will all ends up with have been defined in the Pre-Project, now one layer is pilled of the box at a time, which is done in the next 3 sections. Before digging in to the realization phase a detailed development plan for the product is then defined.

3.1 General Analysis

In the general analysis section the requirement is found and tests to check if these requirements have been fulfilled created. Overall function blocks for the system are drawn, different ways of interacting with the system should be defined together with the design of the final product.

3.1.1 Requirement Analysis

From the teachers wiki [1] requirements have been found and listed together with the requirements defined by the customer. The requirements have been divided in groups:

- Functional Requirements ("system shall do 'requirement' ")
- Non-functional Requirements ("system shall be 'requirement' ")
- Behavioural Requirements ("how the system shall react")
- Performance Requirements ("how well does it have to be done")

Common requirements for the energy system can be found on the E10 wiki [2]

ID	**Requirement**	**Description**
F-1	Communication	
F-1.1	Web Server	The hub shall be able to put data on a web-server.
F-1.2	System	The hub shall be able to communicate with a connected module through power line communication.
F-1.2.1	Protocol	The Power Line Communication shall be done according to the common protocol (see appendix: Protocol)
F-2	Routing	
F-2.1	Direction	When a module is connected the system shall automatically find out if it is an producer-, storage- or consumer module.
F-2.2	Consumer	Only allowed to take energy from the system.
F-2.3	Producers	Only allowed to give energy to the system.
F-2.4	Storage	Must acts either as a consumer or a producer (can be changed dynamically).

Table 3.1: Functional: System shall do...

[1] http://teachers.ede.hih.au.dk/index.php/Main_Page
[2] http://e10.ede.hih.au.dk/index.php/Common_Requirements

ID	Requirement	Description
NF-1	User Interface	
NF-1.1	Web Interface	Maximum 2 click to go where you want on the website.
NF-1.2	HW Interface	Able to connect 10 modules to the hub.
NF-1.3	HW Interface	A locker shall be unlocked to access the physical hw.
NF-1.4	HW interface	An Emergency stop shall be visible on the hub and shall shut down the system if pushed.
NF-1.5	HW Interface	1 start button for the hub. 10 buttons to each start a module.
NF-2	Electrical	
NF-2.1	Voltage	The input and output ports must work in the range 30V +/- 10%.
NF-2.2	Current	The maximum current on each port is defined to maximum 30 Amperes.

Table 3.2: Non-Function: System shall be...

ID	Requirement	Description
P-1	Hardware	
P-1.1	Housing	The housing have to be water-proof, to not harm the system.

Table 3.3: Performance: How well does it have to be done...

ID	Requirement	Description
B-1	Status	
B-1.1	Web Interface	Warnings are posted in the web interface.
B-1.2	Physical	1 diode showing when a module is off and one showing when it is on.
B-1.3	Report	An e-mail is sent to the defined user if an error occurs.
B-2	Energy Control	
B-2.1	Over-production	Overproduced energy is wasted in a dummy load connected to the hub.
B-2.2	Over-production	If two or more producers are connected and one can be without, it is stopped.
B-2.2	Under-production	If there is no overproduction (dummy load is turned off), the grid is connected to the power line in case the producers cannot produce enough energy.
B-3	Errors	Humidity and Temperature sensor will be placed inside the system housing.
B-3.1	Humidity	If the humidity is above the maximum level 70%, the system shuts down.
B-3.2	Temperature High	If the temperature is higher than 55 degrees, the system shuts down.
B-3.2	Temperature Low	If the temperature is below 0 degrees, the system shuts down

Table 3.4: Behavioural: How the system shall react...

3.1.2 Requirement tests

In order to verify that the requirements have been fulfilled, tests should be done. Most of the tests should be done together with the customer in order to get his acceptance that he received what he ordered.
After a test has finished, a binary grade will be given (pass / fail), together with a comment telling how the test went: where it was done, with whom and when it was done.

Test of the Functional Requirements

ID	Test Description	Grade	Comment
F-1.1	Check if the parameters on the website is updating. The easiest parameter to validate is the uptime of the system. Check the website with half a minute in between and verify that the parameter has been updated.		
F-1.2	Connect a module to the hub, by connecting the module to the hubs power line (plugs on the back of the hub). If the module respond to a ping signal send from the hub, the two modules are communicating through power line. The response of the ping can be seen by using an oscilloscope and simply analyzing the packages on the power line according to the protocol.		
F-1.2.1	A test of the protocol is placed together with the description of it. This test is verified in corporation with the rest of the teams.		
F-2.1	Connect 3 modules: a producer, consumer and a storage device one at the time. Verify that the hub has correctly read the modules by checking connected modules on the website.		
F-2.2	Connect a consumer to the hub. Verify that the current flow is going to the consumer.		
F-2.3	Connect a producer to the hub. Verify that the current flow is going to the hub.		
F-2.4	Connect a storage device together with a producer. Verify that the current flow is only going to the storage device. Turn of the producing module and connect a consumer (note that the storage device shall have some energy stored). Verify that the current flow is only going from the storage device.		

Table 3.5: Functional requirement test

Test of the Non-Functional Requirements

ID	Test Description	Grade	Comment
NF-1 .1	The user is asked to go through some simple steps. See production of a connected module. Login, start a module, stop a module. Check alternative energy production (money, CO2, bulbs). Together with the user it is accepted that there is maximum 2 clicks to find a certain thing on the website.		
NF-1 .2	Check if there are 10 module connectors on the back of the hub module. Verify that all connectors work by connecting a module to one input at the time and check that a red diode for the module input turns on.		
NF-1 .3	Together with the user it is verified that he cannot control the front interface without unlocking the system with a key.		
NF-1 .4	Check that an emergency button is visible on the hub (the user decides if it is visible enough). The button is checked by getting the system up and running and pressing the emergency button. The system shall now shut down.		
NF-1 .5	It is visible inspected that 11 start buttons can be found on the physical hub interface		
NF-2.1	The voltage found on the power line is handled in the modules. Test can be found in the common requirement document. However the power line is measured with a voltmeter (without any modules on, only the grid) and shall be in the range 27-33V		
NF-2.2	The maximum current given or taken is handled in the modules. Test can be found in the common requirement document.		

Table 3.6: Non-Functional requirement test

Test of the Performance Requirements

ID	Test Description	Grade	Comment
P-1.1	Inspect the housing together with the customer, while water is being dumped on the roof and the sites of it		

Table 3.7: Performance requirement test

Test of the Behavioural Requirements

ID	Test Description	Grade	Comment
B-1.1	A module is connected to the hub, depending on the available module: -Battery module: disconnect all batteries (no batteries) -Wind module: turn the turbine away from the wind (low wind) Photo-cell module: Cover the cells (low sun). If one of the above states are done, a warning message should be posted on the website		
B-1.2	Check that two diodes are shown next to a module start button (one red and one green). Connect a module, now check that the red diode turns on. Go to the website and turn on the module, check that the green diode turns on.		
B-1.3	Connect a module to the hub and start it. When the status of the module is running (see on the website), the module is unplugged. Now check the mail box of the user and verify that a mail has been received from the energy system.		
B-2.1	A dummy load and a producing module is connected to the system. As no consumers is connected, the system is overproducing. The current flow to the dummy load is measured. If the current flowing to the dummy load is the same as the one coming from the producer (maximum -10%), the test is valid.		
B-2.2	A dummy load and a producing module is connected to the system. As no consumers is connected, the system is overproducing. A variable consuming module is now connected. The resistance in the consuming module is decreased (increasing load). The current flow to the dummy load shall now fall. When no current is flowing to the dummy load, it shall be observed that the grid is connected. This is done by increasing the load of the consuming module and measuring the current flow from the grid.		
B-3.1	A water boiler is placed below the humidity censor. A humidity interment is placed next to the hubs humidity sensor. The hubs reaction when the humidity comes above 70% is observed. If the hub shuts off, the test has been successful.		
B-3.2	If a climate cabinet is available, the heat tests will be performed there, otherwise a heating gun will be used to increase the temperature. A temperature instrument is placed next to the hubs temperature sensor. The hubs reaction when the temperature comes above 55deg is observed. If the hub shuts off, the test has been successful.		
B-3.3	If a climate cabinet is available, the heat tests will be performed there, otherwise cooling spray will be used to decrease the temperature. A temperature instrument is placed next to the hubs temperature sensor. The hubs reaction when the temperature comes below 0deg is observed. If the hub shuts off, the test has been successful.		

Table 3.8: Behavioural requirement test

3.1.3 Problem Domain Analysis

Everything that should be controlled by the system-to-be will be defined here.

3.1.3.1 Block Diagram:

Block diagram of the hub. Showing all the things that the hub should interface with.

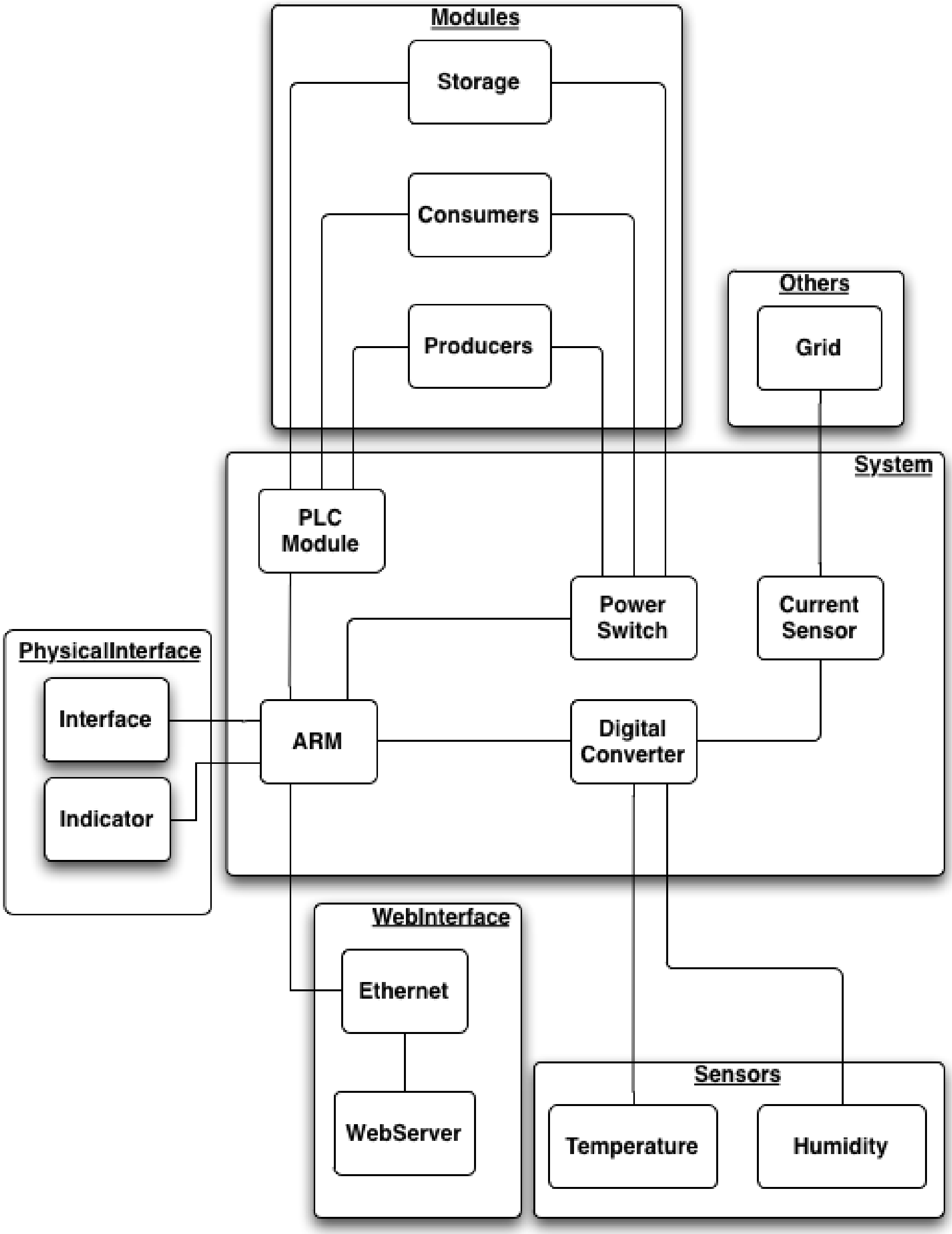

Figure 3.1: Block diagram

3.1.3.2 Block Diagram: Candidates in the system

- **Power Switch:** Part of the hub that powers the connected modules on or off.
- **Consumers** Modules that consumes power (e.g. light, washing machine, electric car)
- **Producers:** Modules that delivers energy to the hub (e.g. Wind-turbine, photovoltaic-cells)
- **Storage:** Modules that both consumes when the system over produces and "produces" (gives energy to the consumers) when needed (e.g. battery, air compressor).
- **Grid:** The grid is used to start up the whole system. It is also used if producing modules does not produce enough energy to the consumers.
- **Current Sensor:** Unit that measures the current taken from the grid. To calculate the amount of green energy produced.
- **Temperature:** Unit that measures the temperature where the system is placed. If the temperature is too high or low it might damage the hardware.
- **Humidity:** Unit that measures the humidity where the system is places. If the humidity is too high the hardware might be damaged.
- **Ethernet:** Unit that has connection to an internet database.

3.1.3.3 Block Diagram: Events in the system

All candidates shown in the block diagram.

- TemperatureBelowLimit
- TemperatureAboveLimit
- HumidityAboveLimit
- CurrentMaxLimit
- CurrentMinLimit
- EnergyAboveLimit
- EnergyBelowLimit
- InitModule
- StartModule
- StopModule
- StandbyModule
- CreateModuleLogFile
- LoadModuleLogFile
- UpdateModuleLogFile
- UpdateServer

Table with all the above candidates and event combined: All the different events that can happen in a block or sent between blocks.

	P SW	Cons.	Prod.	Storage	Grid	Curr Sens	Temp	Humi	Ether.
TempBellowLimit							X		
TempAboveLimit							X		
HumidityAboveLimit								X	
InitModule		X	X	X					
StartModule		X	X	X	X				
StopModule		X	X	X	X				
StandbyModule			X	X					
CurrentMaxLimit					X	X			
CurrentMinLimit					X	X			
EnergyAboveLimit			X		X				
EnergyBelowLimit			X		X				
CreateModuleLogFile			X						
LoadModuleLogFile			X						
UpdateModuleLogFile			X						
UpdateServer									X

Table 3.9: Class diagram of all the hubs events and candidates.

3.1.3.4 State Machine Diagram

Digging deeper into the hub module, states in the different blocks is defined.

Micro Processor (ARM) states:

Description of the different states:
After the initialization of all modules connected to the hub, the system is in *Idle Mode*.

Idle: After initialization the hub goes into idle mode. The hub stays in idle mode until a start signal is given from the user (from the web or the physical hub interface).

Run mode: The system stays in Run Mode until an event happens, e.g. temperature or humidity reaches their limits, a new module is connected, a module should be removed, the system is over-/underproducing, the log should be updated etc.

Start: Whenever a new module is connected it waits for the user to start the module from the web interface.

Stop: To securely disconnect a module, the user must use the disconnect button (on web interface) to make sure all data is saved.

Shut down: If the system finds itself in a critical condition (temperature, humidity, communication problems) it shutdown it's connected modules, tries to save all available data to the log. Then it sends an message (in form of e-mail) to the user, describing the problem, and the module powers off itself.

Connect module: The system checks if it has seen exactly that module before, if that is the case it updates variables from the database (uptime, production etc.). If the module is new for the system, it creates it in the database and gives it an unique id.

Warning: Is sent if the system tries to set a submodule in start- or stop mode, but after a while it has not changed (timeout). After the timeout state, the system returns to run state.

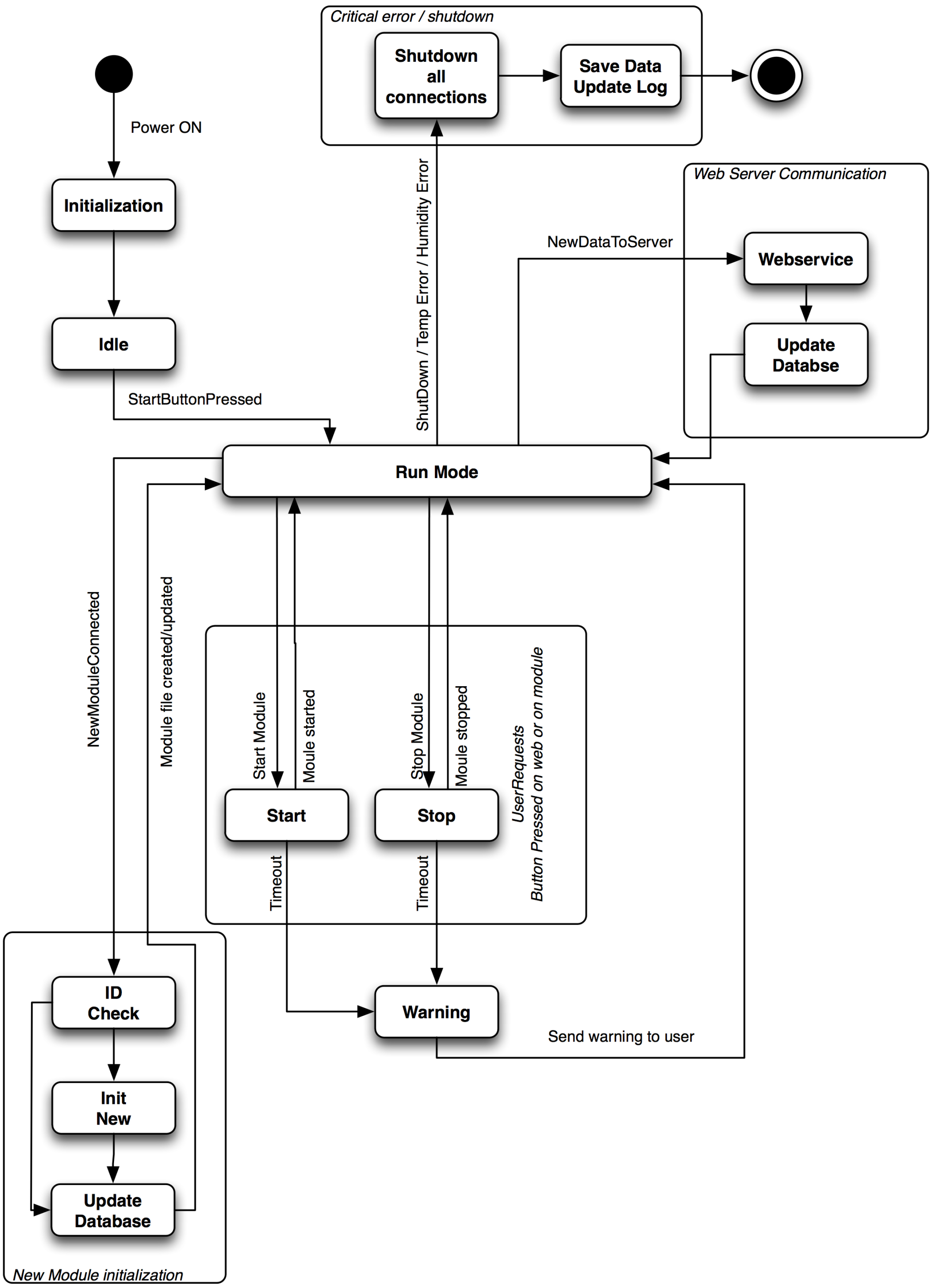

Figure 3.2: State Machine Diagram of the system

Web Server

Description of the different states: **Listening:** The system will listen the TCP / IP port until an event happens, e.g. a new module is connected, a module should be removed, new data to the data base, user event from the web interface etc. **Webservices:** The web server will execute a PHP script, this will perform requests to the database and send start and stop commands to the hub.

Webservices:
- insertData: inserts new data to table (module_id, current, voltage, time of measure, status, efficiency) this are the data common to all modules.
- selectData: show data to the user on the web page, this is a dynamic request.
- clearData: resets database, by erasing the database and recreating and populating.
- sendCmd (Start / Stop / Cmd): commands send from the web interface to control the system and different modules.
- sendWarning: shows an warning in the web interface and the administrator receives a warning in the defined email address.

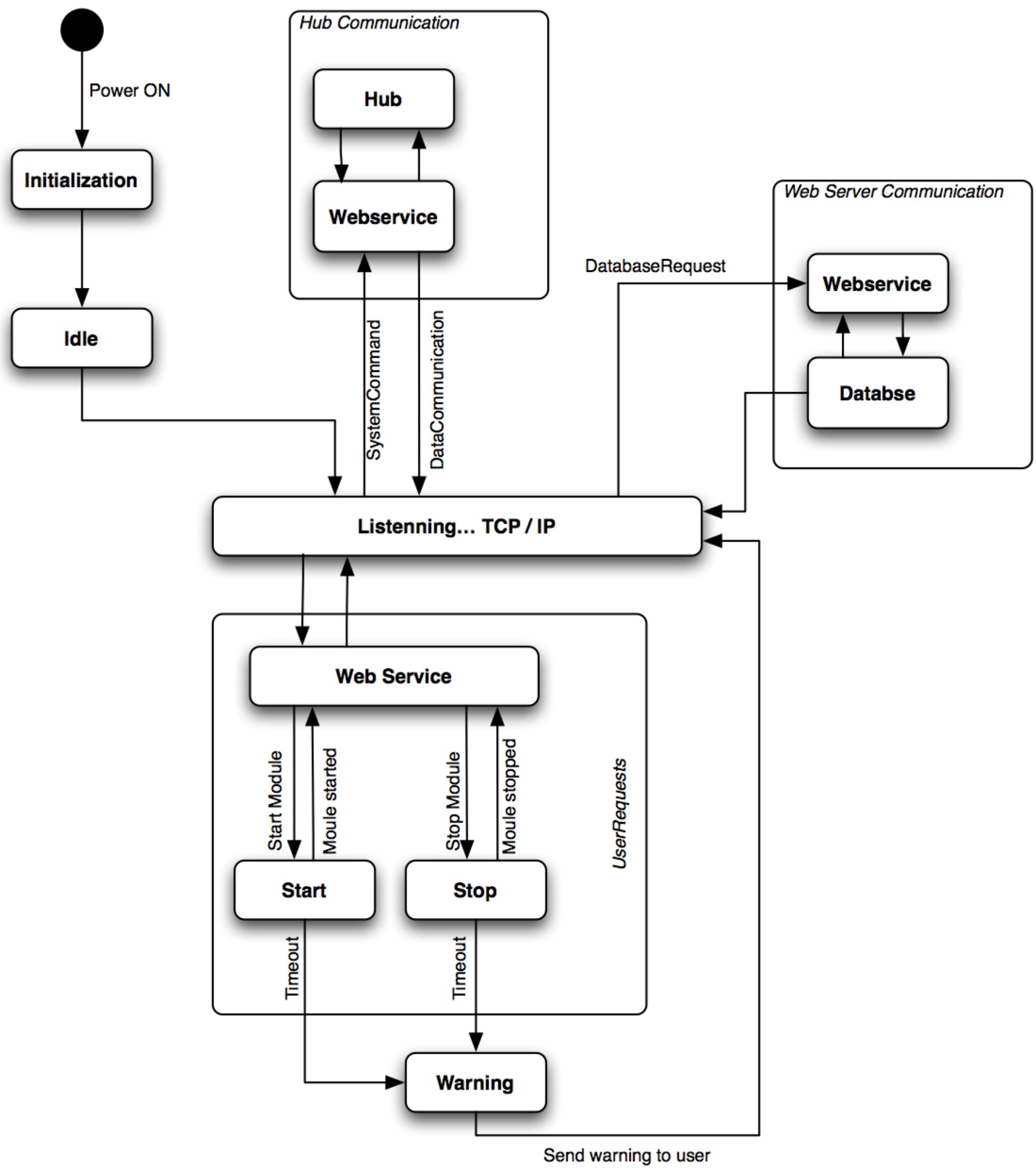

Figure 3.3: State Machine Diagram of the Web Server

Database structure:

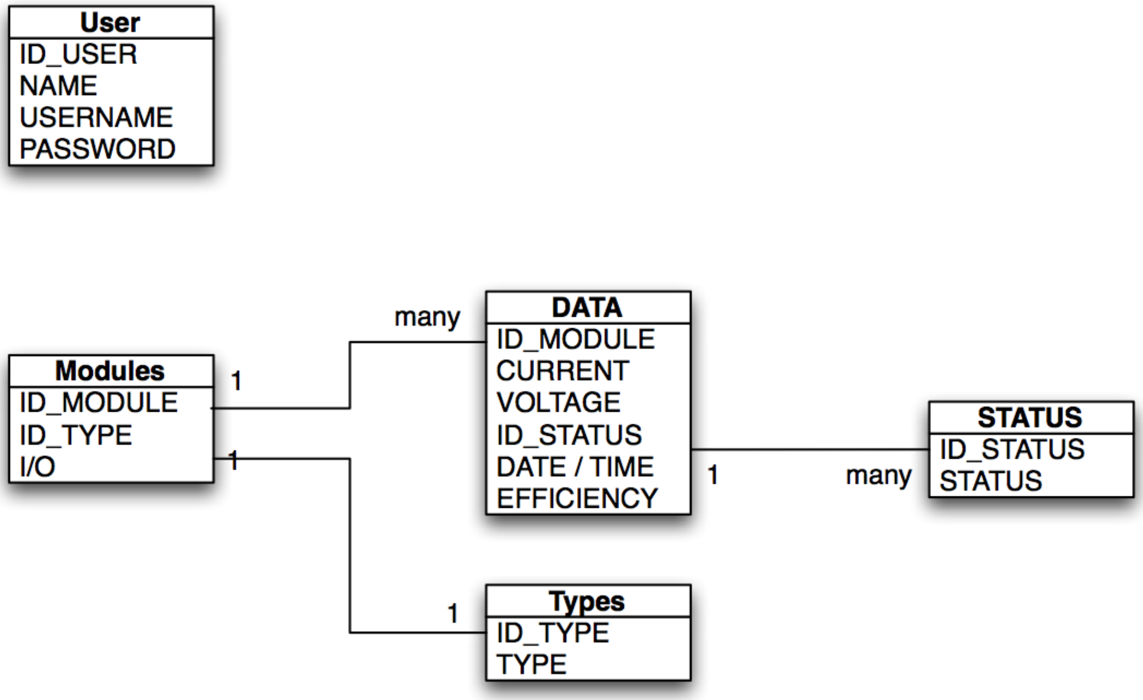

Figure 3.4: Database Structure

3.1.4 Usage Domain Analysis

A description of all the different users to the system and the actions each user are allowed to do.

Web user - Log in
Description: An user with administrator privileges that can login to the web interface of the system. This user can manage the energy hub from a computer or mobile device via the web page and have permission to start and stop modules.

Web user - Visitors
Description: This visitor of the web interface can only see info and data on the web page for the system. This user do not have permission to manage anything.

Hub
Description: The Hub actor is the system, that have the ability to make action on its own, for example it have to be able to shutdown a module if an error occurs on that module.

Engineer
Description: The engineer is any person with knowledge about how the system works on a technical level. This user is able to reprogram the whole system and have permission to everything.

On system user
Description: On system user is any user that go to the physical system and operates on it. This user is able to turn modules connected to the hub on or off.

Plug in module
Description: The user should be able to plug in different modules to the system, this is only possible when you stand physically next to the system, so this use case is only included for the *On system user*.

Unplug module
Description: Like plugging in modules this use case is only included for the *On system user*.

Start/Stop module
Description: The capability of turning modules on and off is included in the *Engineer* and the *Web user - Log in*.

Start/Stop system
Description: The capability of turning the system on and off is included in the *Engineer*. The *Web user - Log in* do not have the option to turn the system on.

View Production/Consumption of module
Description: It is possible for the *Web user - Log in*, *Web user - Visitors*, *Hub* and the *Engineer*.

View temperature
Description: It is possible for the *Web user - Log in*, *Web user - Visitors*, *Hub* and the *Engineer*.

View log of module
Description: It is possible for the *Web user - Log in*, *Hub* and the *Engineer*.

View humidity
Description: It is possible for the *Web user - Log in*, *Web user - Visitors*, *Hub* and the *Engineer*.

Initialize module
Description: It is only possible for the *Hub* to initialize modules.

View energy level of storage modules
Description: It is possible for the *Web user - Log in*, *Web user - Visitors*, *Hub* and the *Engineer*.

Create log file
Description: Creating a log file for a module is only possible for the *Hub*.

Disconnect module
Description: Disconnecting a module without unplugging it is only possible for the *Hub* and the *Engineer*.

3.1.4.1 Use cases

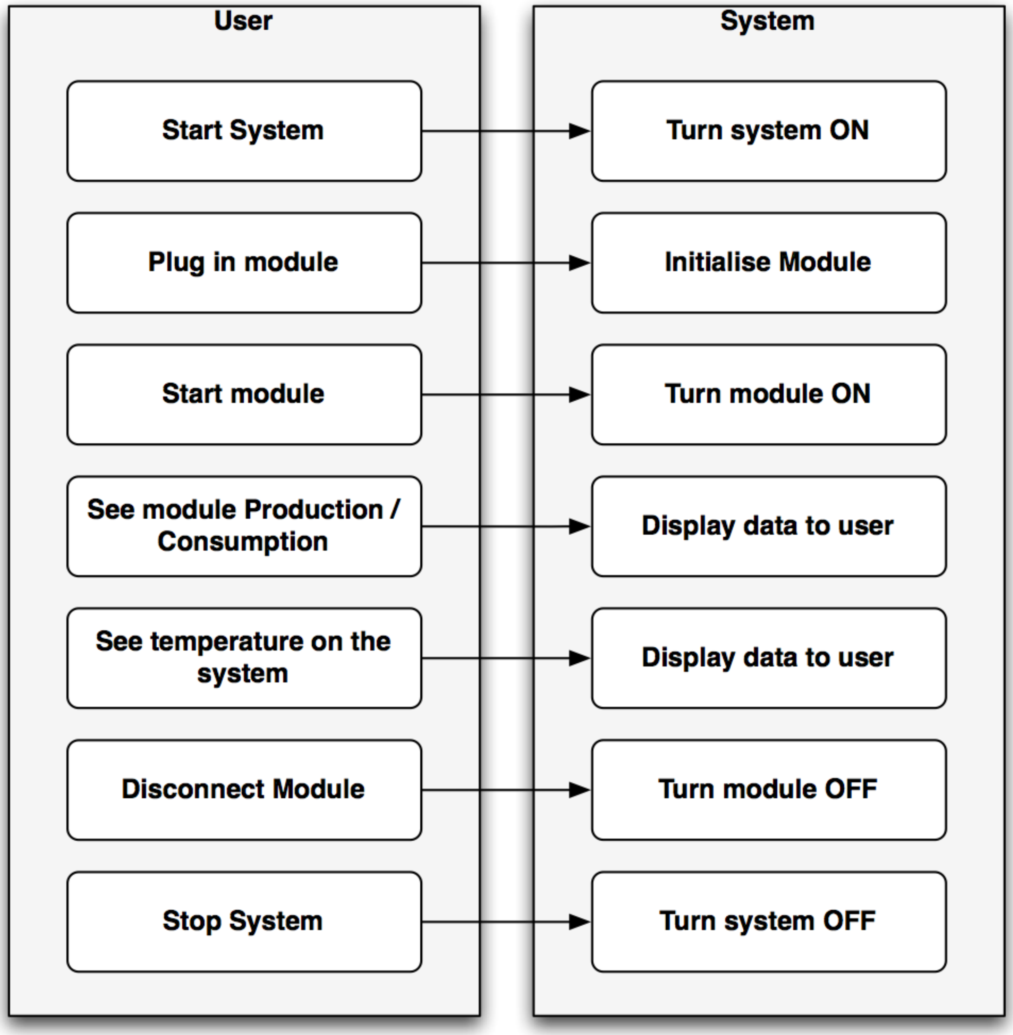

Figure 3.5: Example of *on system user* interaction

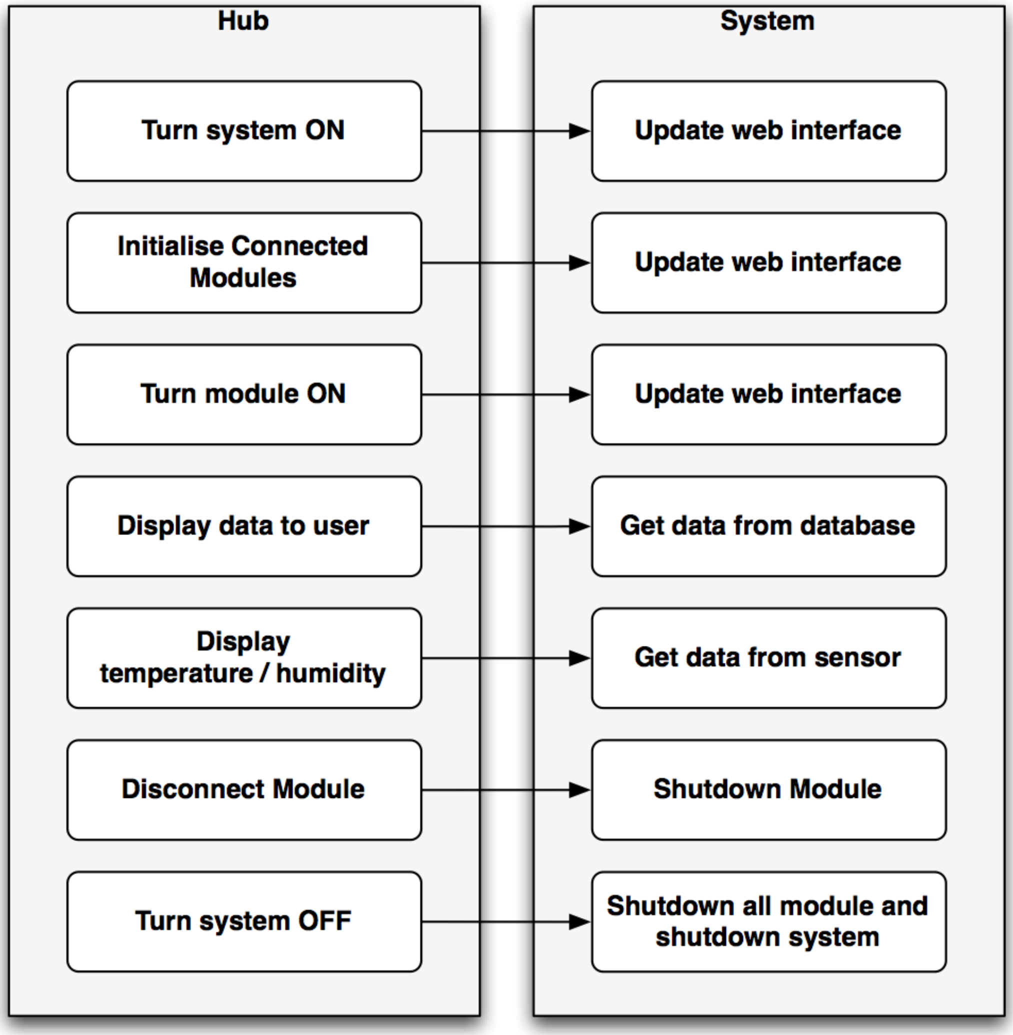

Figure 3.6: Example of *hub* user interaction

3.1.5 Interface Analysis

In the EIDE1 course the interaction design was made for the web interface and the physical interface for the hub, with different design sessions. The user needs and the design for the two interfaces was determined to be made. The IDE report is included in the appendix.

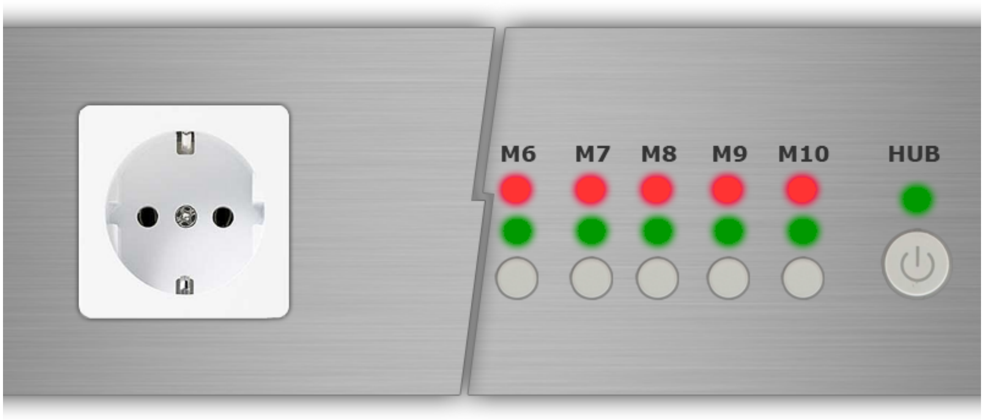

Figure 3.7: Final design of the hubs physical interface.

WebInterface: In EWEB course a static web interface was developed with the knowledge gained from the interaction design sessions and developed in HTML, the WEB report is included in appendix.

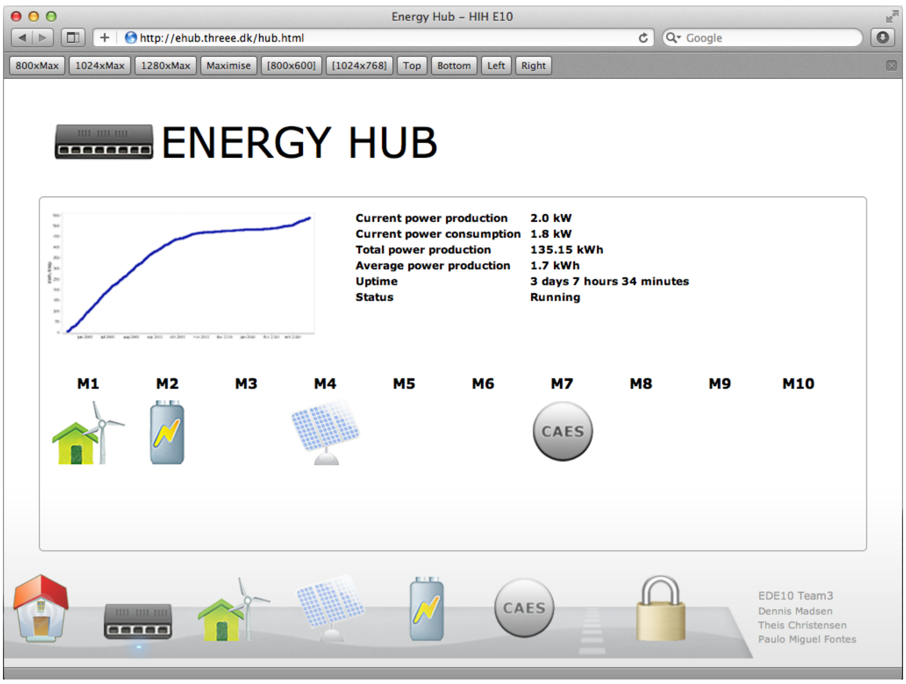

Figure 3.8: The final web interface design, showing the hub module page

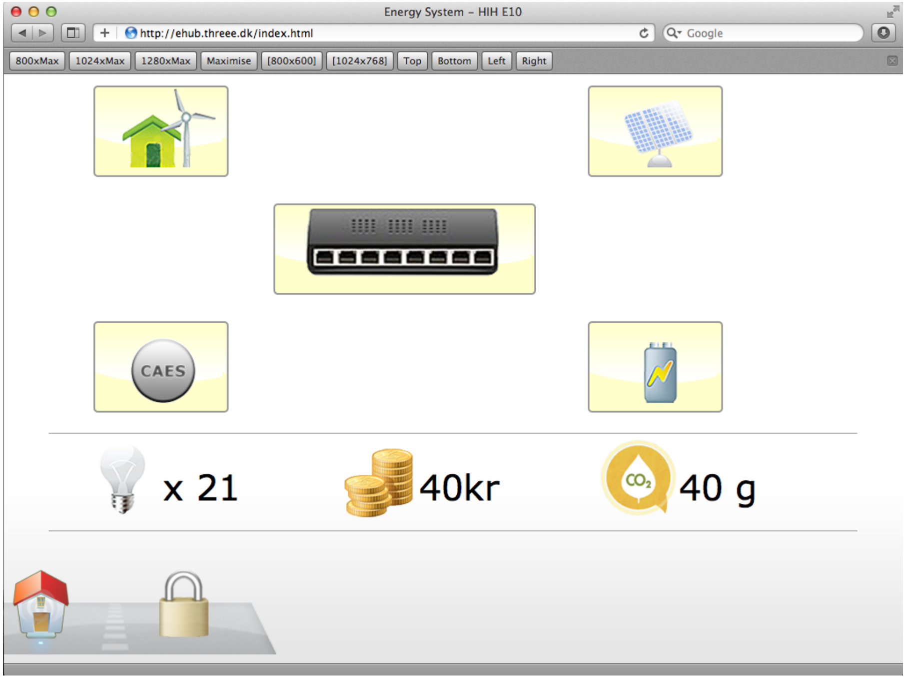

Figure 3.9: The final web interface design, showing index page

3.1.6 Function Analysis

From an analytic perspective, functions are very useful as they are intended to elaborate the objective of the system. When defining functions the question is, *what is the system supposed to do?* In the usage part, it was concerned *how* the system should be used. This makes the usage and functions closely connected, since it is difficult to talk about how a system is being used without discussing what it should do.

When analyzing functions the object is to get a complete list of all the functions the system must implement. The goal is not to describe every single function in detail, quite the contrary the goal is to identify the functions.

Functions can be grouped into types. There are four different types of functions:

Update is a type defining functions which are activated by a problem or application domain event, and which results in a change in the models state.

The type *Signal* defines a function which is triggered by a change in the models state. Running the function always results in a reaction to its surroundings: that is either the reaction is a display to the actors in the problem domain or else the reacting a direct intervention in the problem domain.

When an actor has need for information the function of type *Read* is activated. The function is displaying the relevant data of the model to the actor.

Finally, the *Calculating* function is activated when an actor provides information, which should be included in a computation which also involves data from the model. The function returns its computed result to a display.

When all functions are found they must be defined by their type and complexity, as shown below.

Title	**Complexity**	**Type**
connect_module Modules can be connected no the back of the hub	S	Update
disconnect_module Modules can be disconnected on the back of the hub	S	Update
start_system Start the hub from the physical interface	S	Update
stop_system Stops all modules including the hub	S	Update
power_off_system Critical error, shut down it self	S	Update
start_module Modules can be started from the hubs web page	M	Update
initialize_module Modules are automatically initialized when connected	C	Update/Calculating
stop_module Modules can be stopped from the hubs web page	M	Update
get_module_production Read data from module	M	Read/Calculating
get_module_consumption Read data from module	M	Read/Calculating
get_module_energy_level Read data from module	M	Read/Calculating
get_temperature Read temperature sensor	M	Read/Calculating
get_humidity Read humidity sensor	M	Read/Calculating
create_module_log _file Create a log file for a module	VC	Read/Update
load_module_log _file Load an already created log file of a module	C	Read/Update
save_module_log _file Save a new log file of a module	C	Read/Update
send_module_command Send a command to a module	M	Update
read_module_command Read a command from a module	M	Read/Update
webserver_write_data Write data to the web server	M	Update
webserver_read_data Read data from the web server	M	Read/Update

Table 3.10: Table showing the different functions in the system.
S = simple. M = medium. C = complex. VC = very complex

3.1.7 System Dynamics

Diagram of how to put data on the web server.

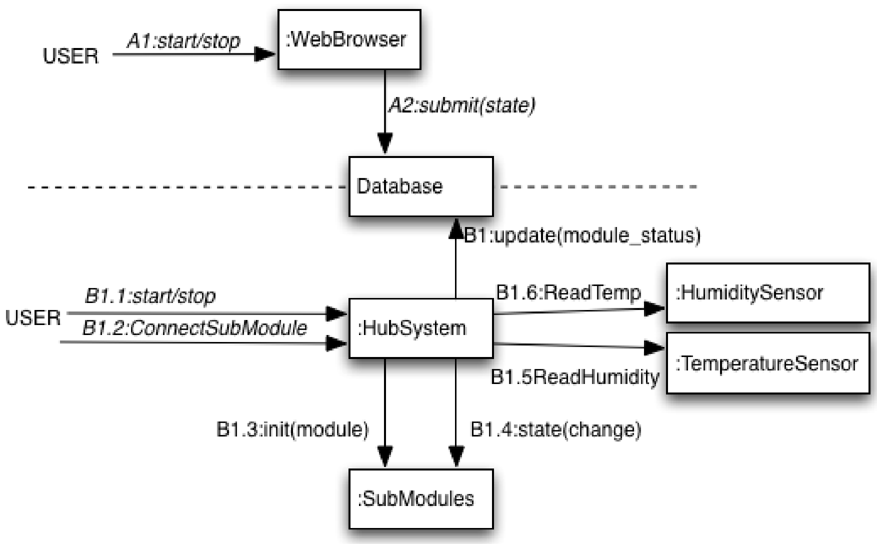

Figure 3.10: Interaction between systems; Web browser and Hub both connected to the database

3.2 General Architecture Design

Design criteria is important to consider in order to choose the right technical platform to perform the criteria. When marketing the product design criteria is very important, in making the specification for the product.

3.2.1 Design Criteria

Write in what parts should be used (mostly off the shelf things).

Issue	Critical	Very Important	Important	Less Important	Notes
Safe	X				1
Performance			X		2
Usage	X				3
Reliability	X				4
Easy serviceable		X			5
Remote maintenance			X		6
Cost effective				X	7
Learnability		X			8
Memorability		X			9
Effectiveness			X		10
Utility				X	11

Notes:

1. The safety is a very important factor, as the system is meant as a showoff system for high-school students, who should be able to be near the system without hurting themselves or the system.

2. The performance in the system is not critical when considering it from the users perspective. It doesn't matter if it takes a few seconds before the system responds the user. But the system should still be able to react fast in order not to harm it self, but also to get a high effectiveness.

3. The system should be easy to use. All kinds of people should be able to use the system without any specific training. Connecting of new devices and doing administrative jobs on the system a small walk through of the system is required.
4. Errors must not occur in the system. The hub is the central nerve in the whole system, therefore if the hub does not run, non of the subsystems does and no energy is routed.
5. The user of the system should be able to find eventual errors on the system by him self, with help from a good error log created by the system.
6. The only thing possible to maintain remotely is start and stop of submodules + power down the whole system, which is less important as the system is placed locally.
7. Only one kind of the system is to be produces, but still the price should be kept at a relatively low level (as we have an undefined maximum price of the system).
8. As the system is meant to be operated by a non-technically persons, learning the system shall be fast and operating the system shall therefore be very intuitive. Also visitors should be able to do simple tasks on the system, such as turn a device on or off.
9. Mainly the system is supposed to run by it self without any interaction. But when the system is going to be operated on, or the system is showed for visitors, the instructed person(s) shall be able to do so without any preparation time.
10. The effectiveness of the system is not the most important factor, but still quite important as the talk is about a green system, which must be affordable for the user to implement the system. When investing in a green system, it will have information about estimated lifetime and buy time (the time it takes the system to 'buy home itself'). If the buy time is to long, maybe longer than the lifetime, the price for having such a system will be way to high. The goal is that visitors can see the advantage in buying a green system (environmental and money friendly), so it might affect them to be more environmental friendly.
11. The possibility of changing parameters in the system should be kept as low as possible to easy the interaction with the system. If the user have too many ways of setting up the system it can easily confuse more than necessary. Therefore only the most important things that the user should be able to change is implemented, rest of the settings is placed as hidden utilities (only to be set up by the developers during tests).

3.2.2 Architecture Dynamics

The sequence diagram explain the dynamics of more complex parts of the system, this diagram is important for estimating the price and time for the development, by giving the developer a better understanding of how the system works.

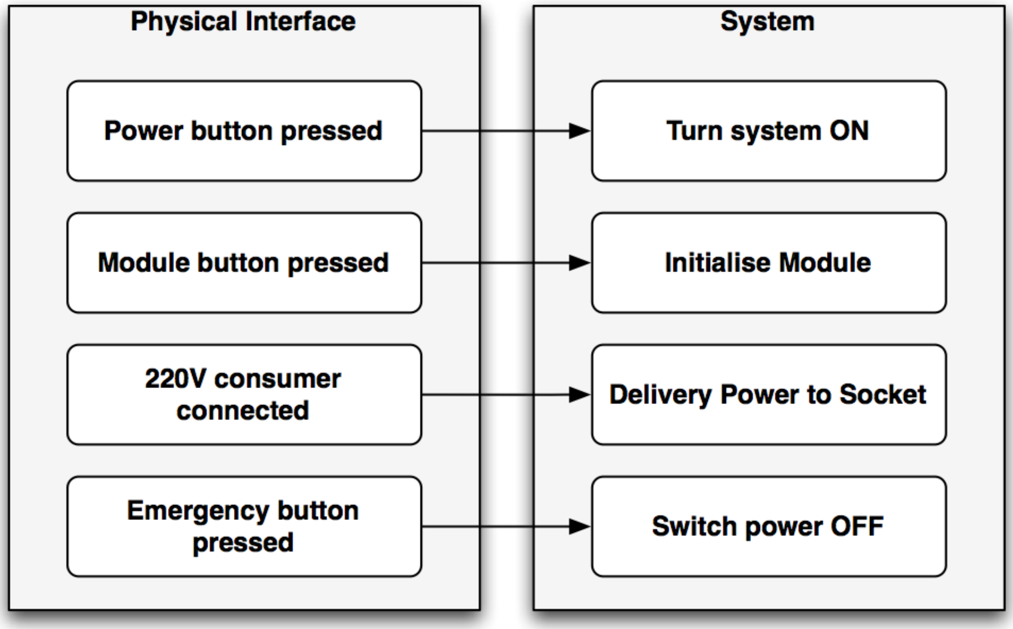

Figure 3.11: Sequence diagram; Communication between the system and external systems.

Sequence Diagrams

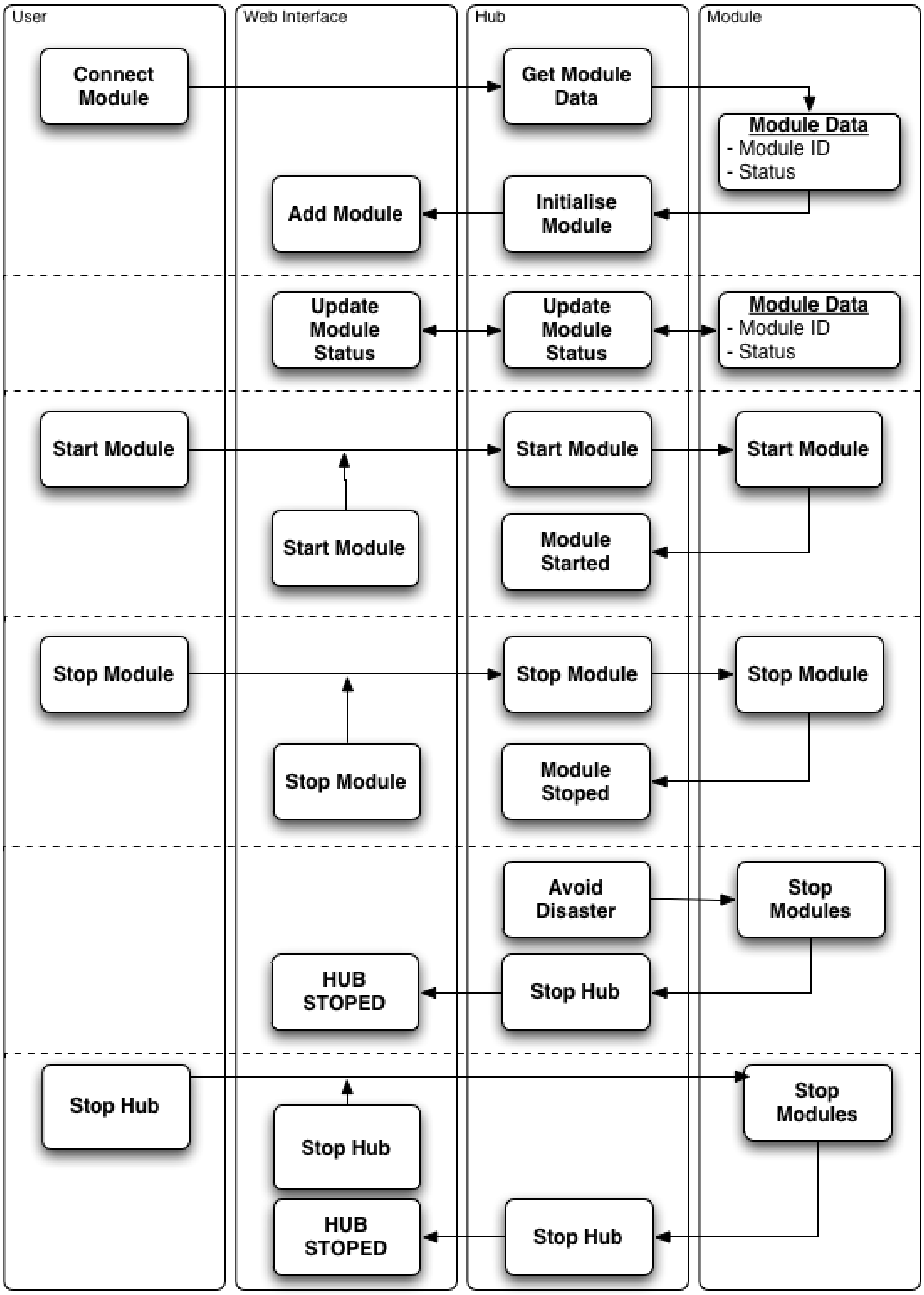

Figure 3.12: Sequence diagram; Communication between the system and external systems.

3.3 Technical Platform

Selecting the technical platform is very important. The wrong platform may cause negative effects on the final product, or in the cost of the development of the product. The hardware have different specifications and lifetime. Here the design criteria is a big help for selecting the right platform, to fulfill the requirements

3.3.1 Hardware Specifications

As hardware we are going to use the following:

- Embedded Artists LPC2478-32 Developer kit.
- Temperature sensor
- Humidity sensor
- Current sensor
- Digital Converter
- Line communication module

3.3.2 Software Specifications

In the software part we are using the ARM processor, and we have to develop drivers for the different hardware above, we are also making the following in software:

- Web
- Ethernet communication
- Storage module driver
- Producer module driver
- Consumer module driver

Class	**Interface**	**HW**	**SW**
System:			
ARM	DIG		X
PLC module	DIG/ANA	X	X
Power switch	DIG/ANA	X	X
Digital Converter	DIG/ANA	X	X
Current sensor	DIG/ANA	X	X
Sensor:			
Temperature	ANA	X	
Humidity	ANA	X	
Web interface:			
Ethernet	DIG		X
Web server	DIG		X
Other:			
Grid	ANA	X	
Modules:			
Storage	DIG		X
Producer	DIG		X
Consumer	DIG		X
Physical interface:			
Interface	DIG/ANA	X	X
Indicator	DIG/ANA	X	X

Table 3.11: HW + SW Specifications. DIG = Digital. ANA = Analog. HW = Hardware. SW = Software

3.4 Contracting

Week	Task	Note
5		EMC course, Meeting with MOJ and KK
6		
7		Meeting with MOJ and KK
8		
9		Meeting with MOJ and KK
10		
11		Meeting with MOJ and KK
12		
13		Meeting with MOJ and KK, Project week
14		Easter holiday
15		Meeting with MOJ and KK
16		
17		Meeting with MOJ and KK
18		
19		Meeting with MOJ and KK
20		Project week
21		Meeting with MOJ and KK, Project week, submit project

Table 3.12: Time plan for 4. semester

Chapter 4

Conclusion

Until now the PreProject and Launch Phase has been made. The PreProject gives a short project analysis to see the cost and time the project would take to develop, there is few technical words in the PreProject for the customer to understand the analysis. This has been fulfil and show that it is possible to develop.
In the Launch Phase the project has been analysed in detail, and description of system function and implementation has been made.

Chapter 5

Appendix

Team 3:
Interaction Design Report + videos
Web report
All teams:
Protocol document - Communication protocol between the modules
PLC HW document - Hardware circuitry for the PLC modules
Common requirements - Common system requirements for all teams