University of Groningen

Software Patterns
Team 2

# Home Energy Monitoring System


**Authors**
Putra, Guntur
Fakambi, Aurélie
Schaefers, Joris
Menninga, Wouter


Sunday 15th November, 2015

Version 0.1

# Authors



| Name | E-Mail |
|---|---|
| Putra, Guntur | G.D.Putra@student.rug.nl |
| Fakambi, Aurélie | A.Fakambi@student.rug.nl |
| Schaefers, Joris | J.Schaefers@student.rug.nl |
| Menninga, Wouter | W.G.Menninga@student.rug.nl |



# Revision History

| Version | Author | Date | Description |
|---|---|---|---|
| 0.1 | Schaefers | 15-11-15 | Added a stakeholders section |
| | Schaefers | 15-11-15 | Added a key drivers section |
| | Putra | 15-11-15 | Added Introduction and System context |
| | Fakambi | 15-11-15 | Added non-functional req. and risks |
| | Menninga | 15-11-15 | Added high-level and functional req. |
| | Menninga | 15-11-15 | Improvements to risks |

# Contents

# List of Figures

# List of Tables

# 1 Introduction

This document depicts the software architecture of Home Energy Monitoring System as the first assignment of Software Patterns course. We proclaim ourself ~~as~~ a software architect team working on the Home Energy Monitoring System (HEMS).

Energy is one of main concern in household. The problem is~~,~~ how to monitor energy usage of a particular house in order to prevent useless consumption of energy and even how to save energy, resulting in lower household expenses for energy ~~cost~~. This system focuses on electricity, as one of the main energy consumption of a house.

This system allows the user to collect and store their electricity consumption. This system will also have a web dashboard that shows the graph of electricity consumption, which user can adjust the time range of the graph to see different data, and other useful data such as which device uses the electricity the most, etc. The data, which will be stored in our database, will be used to ~~do analysis that will help user to~~ predict their upcoming electricity bill in the next month and other required analysis. This system also provide a fault-tolerant computing platform to compute this analysis.

The data collection is not our part, we leave this portion of the system to ~~the~~ third party developers. Basically, we provide a cloud computing environment (SaaS) to do energy management system. Everyone can join this system through provided API.

## 1.1 System Context

This system provides monitoring for more than one house. Many houses can connect to the system through Internet. Sensors will be located at every device that users are willing to monitor. Real time measurement of electricity consumption will be sent to the HEMS server via Internet protocol by the sensor. Figure 1.1 shows the system context of the HEMS.

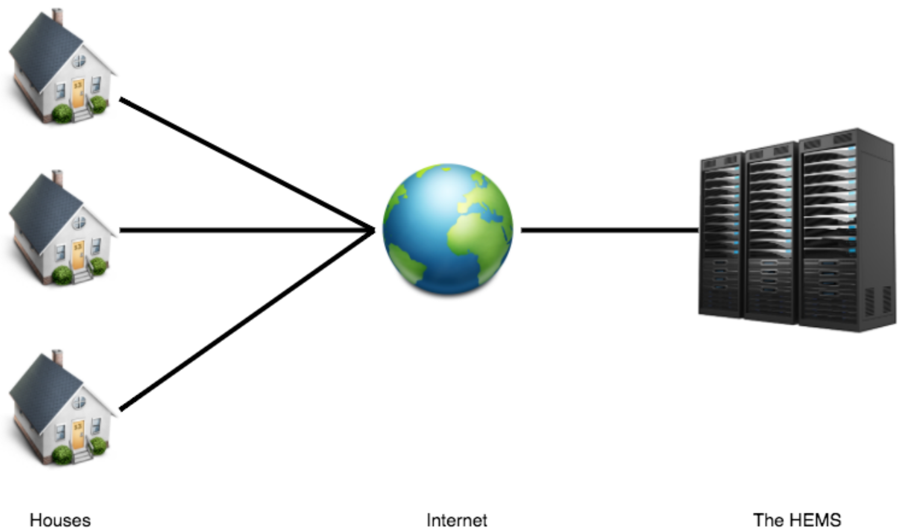


Figure 1.1: The HEMS system context.

As the sensors send the electricity measurement, no servers or any computing devices are needed to be installed in each house. This will make this system to be simple and easy to be implemented in each house. Users may see the energy consumption through the web dashboard via computers, tablets, smartphones, or any devices that are able to connect to the Internet. They may also place a TV or a screen in their house so that everyone in the house can see their energy consumption.

# 2 Requirements

This chapter describes our vision and uses it to derive stakeholders. This information helps us to be able to properly write use cases and stories. These will be used to extract functional, commercial, technical and evolution requirements. Afterwards, a risk assessment will take place, to ensure that the project is not at great risk.

## 2.1 Vision

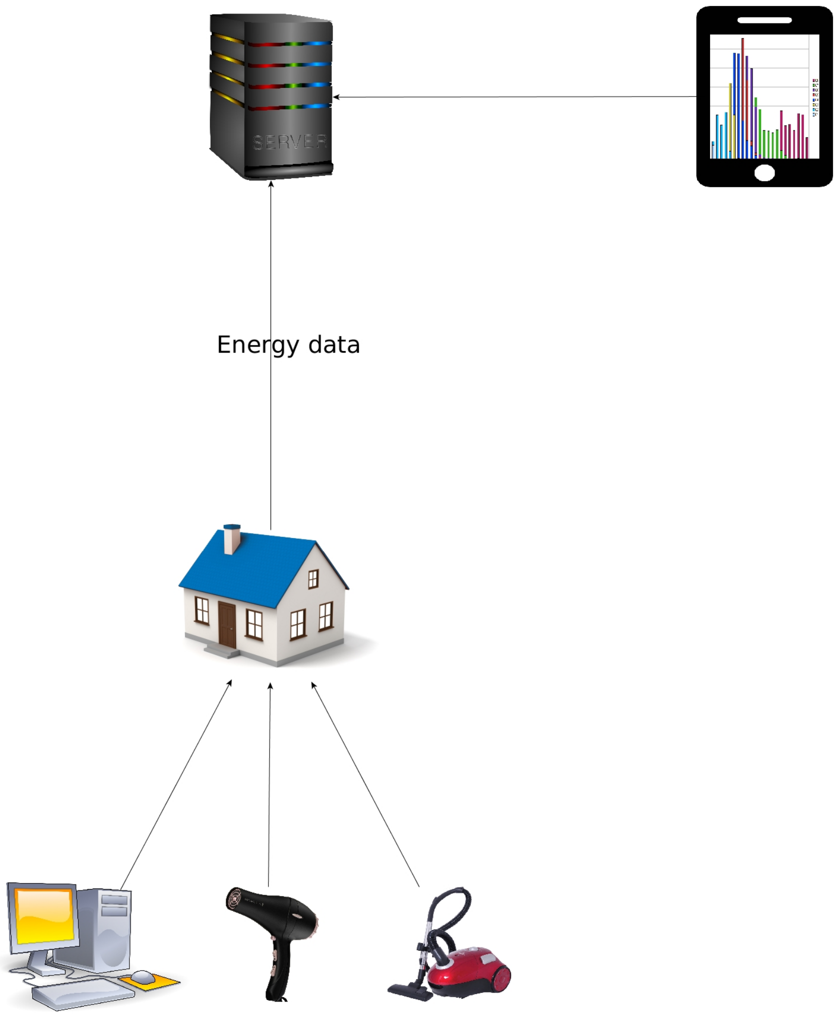


Figure 2.1: Architectural vision

## 2.2 Stakeholders and their concerns

This section defines all stakeholders of our system and describes the concerns of the stakeholders. A stakeholder might be a person, group of persons, or organization that are involved in our system. There are ten stakeholders, ranged from first parties to third parties stakeholders. Several quality standards from the "Software Requirements" book by Microsoft [1] are used. Those quality standards are described in Table 2.1.

| Quality Attributes | Brief description |
|---|---|
| Availability | The extent to which the system's services are available when and where they are needed |
| Interoperability | How easily the system can interconnect and exchange data with other systems or components |
| Performance | How quickly and predictable the system responds to user inputs or other events |
| Reliability | How reliable the results of the system are (accuracy). |
| Security | How well the system protects against unauthorized access to the application and its data |
| Usability | How easy it is for people to learn, remember, and use the system |

Table 2.1: Quality attributes of Software Architecture from "Software Requirements" Book [1].

There are six quality attributes, as can be seen in Table 2.1, for measuring stakeholders' concern regarding our system. Furthermore, profitability has been added as another quality standard to improve measuring stakeholders' concern. The stakeholders are listed below and are then explained in more detail below.

- Product owner
- Developers
- Maintainers
- Government
- Home owners

**Product owner** is the owner of the website. This stakeholder funds the website and its main concern is creating a profit. This affects the quality attributes usability and availability.

**Developers** have to make sure the system provides the functionality that the users expect the system to have. This means that their main concern is usability. But also interoperability is their main concern, because users will want to connect any energy consuming device to this site in order to monitor the consumption.

**Maintainers** are mainly concerned that the website is up and running at all times. Meaning their concern is the availability and security of the system.

**The government** wants to lower the energy consumption of the citizens. It wants to comply with the aims of the EU to reduce the energy consumption by 20% by 2020. Their main concern is the usability and reliability of the system, because the system is only useful to the government if it is used by allot of people and the statistics of the system are reliable.

**Home owners** want to get an insight in their energy usage. They want to know how to effectively reduce their energy consumption. In order to do this, they might want to receive alerts about sudden increases of energy consumption in certain devices. These alerts have to be accurate and reliable. The main concern of the home owners, thus, is the usability and reliability of the system.

## 2.3 Key-drivers

The key drivers of this system are:

1. Usability
2. Reliability
3. Availability

**Usability** has to be the main focus of the system. Users will want to stick to using our system if the usability is better then similar systems of the competitors. Even if competitors have a better availability and or even reliability, there is a good chance that customers will stick to this system if the usability is better.

**Reliability** is very important, because for this kind of system the trust everyone has is crucial. Making the stakeholders trust the system is the most important aspect of this system. If the system at some points does not provide reliable data, without specifically informing about it, the entire system is useless. This not necessarily mean that the system has to be very precise in calculating the statistics. Even if system might be off by 30%, as long as the user is aware of this and it never exceeds this marge of 30%, it is fine. The increases and decreases of the energy consumption will still be clear.

**Availability** is crucial because if the system isn't up at the times when it should, users will lose trust in the system. If a user set an alarm on a device, alarming them if the device exceeds a certain energy consumption, then ideally that alarm has to go any time the device indeed exceeds the given limit of energy consumption. Because of maintenance to the system or dependent systems, the up time of the system might not be 24/7 at all times. But if the system is not up at a certain point, the users have to know about this scheduled downtime.

## 2.4 High-level requirements

The high-level requirements describe the high-level functionality of the system. The high-level requirements are used to derive the functional requirements in section 2.6. The high-level requirements are also used to classify the severity of the risks in section 2.10.

This table uses different priorities. First there is the 'must' priority, this requirement is an absolute must. The system could not function without the 'must' requirement. Besides 'must' there is the 'should' priority. This priority is highly desirable for the system, but the system could function without this requirement. The last priority is 'could', these requirements are nice to have but not really necessary for the core functionality of the system. These different priorities are used through the rest of the document.

| Nr. | Prio | Description |
|---|---|---|
| HL-1 | **Must** | **Collecting electricity usage data**<br>The system must collect electricity usage data, which has to be stored so it can be used to compute statistics and detect changes in energy usage. |
| HL-2 | **Must** | **Computing usage statistics**<br>The system must be capable of computing statistics about the energy usage, like total usage per month, but also periods of peak energy usage. It has to be able to compute such statistics not only per house, but also for individual devices. |
| HL-3 | **Must** | **Displaying usage data / statistics**<br>Data which has been gathered and statistics that have been computed must be displayed to the user in an understandable way. One of the main goals of the system is to make users of the system aware about their energy usage. The data and statistics should be displayed in a way that is intuitive and easy to understand for average consumers. |
| HL-4 | **Must** | **Configuring the system**<br>The users of the system must be able to configure the system, i.e. register their house/add new devices/subscribe to alerts etc., using the web interface. This web interface should be user-friendly and intuitive, such that an average consumer is able to use it. |

Table 2.2: High Level Requirements

## 2.5 Stories and use-cases

## 2.6 Functional requirements

This section lists the functional requirements of the system.

| Nr. | Prio | Description |
|---|---|---|
| FR-1 | **Must** | The system must be able to receive electricity usage data from devices. |
| FR-2 | **Must** | The system must be able to store electricity usage data. |
| FR-3 | **Must** | The system must be able to retrieve previously stored electricity usage data. |
| FR-4 | **Must** | The system must be able to compute the total electricity consumption for a given time period for a particular house. |
| FR-5 | **Must** | The system must be able to compute the electricity consumption per device for a given time period. |
| FR-6 | **Must** | The system must be able to compute an estimated electricity bill for the current month based on the electricity consumption to that point. |
| FR-7 | **Must** | A user of the system must be able to select which statistics to compute in a web interface. |
| FR-8 | **Must** | The system must be able to show the computed statistics in a web interface. |
| FR-9 | **Must** | The system must allow users to register an account on the web interface. |
| FR-10 | **Must** | The system must require users to be logged in, before the user can view electricity usage information about his/her house. |
| FR-11 | **Must** | A user of the system must be able to register a new house using the web interface. |
| FR-12 | **Must** | A user of the system must be able to register a new device for his house using the web interface. |
| FR-13 | **Must** | A user of the system must be able to configure the price of a kWH in the web interface. |
| FR-14 | **Must** | The system must be able to send feedback to registered devices about the current electricity usage. |
| FR-15 | **Must** | The system must be able to estimate the accuracy of the computations, such that when data might be inaccurate, the user will be informed about this. |
| FR-16 | **Must** | The system must be able to display the history of electricity usage. |
| FR-17 | Should | Users of the system should be able to subscribe to alerts in the web interface, alerting them about sudden energy increases or when they are using more energy than in previous months/weeks. |
| FR-18 | Should | The web interface of the system should allow the user to compare the energy usage of different months. |
| FR-19 | Should | The system should send alerts to users by mail when the user is subscribed for this alert and the condition for the alert is met. |
| FR-20 | Should | The system should be able to show the computed statistics in a graph. |

Table 2.3: Functional Requirements

## 2.7 Commercial non functional requirements

## 2.8 Technical non-functional requirements

### 2.8.1 Usability

The sytem will be hopefully use by a lot of people who don't necessarly have knowledge in the technology field so the Usability is an important requirement so the end-user can acces all the information he needs.

| | |
|---|---|
| **US-1** | An application is available for tablets and phone with those OS Windows Phone,Android, and Iphone. |
| **US-2** | The end-user needs maximum ten minutes to get a basic understanding of system features through the UI. |
| **US-3** | The UI has a FAQ and a help section. |
| **US-4** | Every feature/major option of the system can be accessed through the home page(Single Page Application). |

### 2.8.2 Reliability

| | |
|---|---|
| **RE-1** | A margin of error of 5% in the energy measurements is tolerated. |

### 2.8.3 Security

Ensuring the security for the system is a major issue so all the data needed for its good functioning remain protected and consistent.

| | |
|---|---|
| **SEC-1** | Each user is identified and has to log in in order to view his "Home Energy Monitor" account. |
| **SEC-2** | The data in the database is encrypted. |
| **SEC-3** | Passwords of the users are stored and transmitted as a hash |

### 2.8.4 Availability

| | |
|---|---|
| **AVA-1** | The sytem should be available 99.9%. of the time. |
| **AVA-2** | The system should be down for maximum ten minutes when the user installs a new release(version). |

### 2.8.5 Scalability

The system should make the increase of workload, resources and users easy that's why the scalability is an important Key driver.

| | |
|---|---|
| **SCA-1** | The system must be able to monitor several kind of energy consumption for example: gas ,solar, temperature. |
| **SCA-2** | The system should function as efficiently even if the number of users increase |

## 2.9 Evolution requirements

## 2.10 Risk assessment

The system is confronted by several risks which are determined and mitigated in this section. Taking those risks into account allows to avoid them or at least reduce their impact. The risk management involves the identification of the risks, their probability and potential impact or consequences.

The tables below explain the meaning of the definition for probability and consequence.

| **Probability** | **Likelihood of occurrence** |
|---|---|
| High | 0.65 - 1.00 |
| Medium | 0.35 - 0.65 |
| Low | 0.00 - 0.35 |

Figure 2.2: The different probabilities used to classify risks

| **Severity** | **Explanation** |
|---|---|
| Severe | A risk that can lead to loss of live or casualties. |
| Significant | A risk that can lead to damages, can delay the project more than 3 months or causes one of the high-level requirements not to be fulfilled. |
| Moderate | A risk that can lead to one of the high-level requirements not to be fulfilled to an acceptable level. |
| Minor | A risk that can lead to one of the high-level requirements not being fully fulfilled, but still fulfilled in an acceptable level. |

Figure 2.3: The different severities used to classify risks

### 2.10.1 Technical

| **T-RISK1** | **The statistics provided by the data processing framework are wrong** |
|---|---|
| **Probability of occurrence** | Low |
| **Consequences** | Moderate. If the statistics computed by the system are not accurate, this will lead to loss of thrust of the end user in the system. |
| **Prevention** | Make sure the algorithms used to compute the statistics are correct. |
| **Reaction** | Correct the algorithm, if the change is significant also inform end users about the error. |

Table 2.9: T-RISK1 – The statistics provided by the data processing framework are wrong

| **T-RISK2** | **The data center storing the energy measurements becomes unavailable** |
|---|---|
| **Probability of occurrence** | Low |
| **Consequences** | Significant |
| **Prevention** | Store the data in a redundant way, preferably in multiple data centers, so that one data center going offline does not lead to downtime of the system. |
| **Reaction** | If the data storage does become unavailable, new incoming data should be cached so it is not lost and users should be informed in the web interface that viewing the statistics is temporarily unavailable. |

Table 2.10: T-RISK2 – The data center storing the energy measurements becomes unavailable

| T-RISK3 | **Somebody gains unauthorized access to someone else's data** |
|---|---|
| **Probability of occurrence** | Low |
| **Consequences** | Significant. Data about electricity usage can be used, e.g. to derive when people are home. Unauthorized data access will lead to a loss of thrust in the system by consumers. |
| **Prevention** | Make sure access to the data requires authentication using a password at all time. Enforce users to use a strong password. |
| **Reaction** | Make sure the unauthorized access is removed. Inform end users about which data was accessed by the unauthorized party. |

Table 2.11: T-RISK3 – Somebody gains unauthorized access to someone else's data

### 2.10.2 Business

| B-RISK1 | **Wrong estimation of the budget** |
|---|---|
| **Probability of occurrence** | Medium |
| **Consequences** | Significant. The final product does not have the features expected. |
| **Prevention** | The team needs an accountant or at least someone taking care of the follow-up of the money. Make sure there are regular evaluations to keep track of the money flow. |
| **Reaction** | Remove some requirements or features of the product, or change the hardware components used. |

Table 2.12: B-RISK1 – Wrong estimation of the budget

| B-RISK2 | **Wrong estimation of the budget: the money invested in the fabrication and achievement of the product/system is not covered by the sales (shortfall/deficit)** |
|---|---|
| **Probability of occurrence** | Medium |
| **Consequences** | Moderate. Stopping the sale |
| **Prevention** | The team needs an accountant or at least someone taking care of the follow-up of the money. |
| **Reaction** | Adding more features to the product in order to make it more competitive in the market. |

Table 2.13: B-RISK2 – Wrong estimation of the budget: the money invested in the fabrication and achievement of the product/system is not covered by the sales (shortfall/deficit)

### 2.10.3 Schedule

| S-RISK1 | **The project is not finished at the deadline** |
|---|---|
| **Probability of occurrence** | Low |
| **Consequences** | Significant. Pressure for all the team members, loss of credibility regarding the customers, selling a product with less features than expected. |
| **Prevention** | SRA , Schedule Risk analysis : Estimation of the duration of the project by its manager ( with the use of probability and statistics ) . Meeting for the team members every week to keep track of the timing and take decisions according to the deadline. |
| **Reaction** | Postpone the deadline or remove some features when the deadline can't be postponed. |

Table 2.14: S-RISK1 – The project is not finished at the deadline

# 3 Analysis

# 4 System architecture

## 4.1 System context

## 4.2 Verification

## 4.3 Elaborated Model

# 5 Hardware Architecture

# 6 Software Architecture

# 7 Architecture evaluation

# 8 System evolution

# A Time Tracking

## Week 1

| Person | Task | Hours |
|---|---|---|
| Putra | Meetings, writing introduction and system context. | 4 |
| Fakambi | Meeting, First approach on non functional requirements and risks | 4 |
| Schaefers | Meetings, Added a stake holders and key drivers description. | 5 |
| Menninga | Meeting, high-level and functional requirements and improvements to risks | 5 |

# Bibliography

[1] K. Wiegers and J. Beatty. *Software Requirements*. Developer Best Practices. Pearson Education, 2013.