I.BA_IOT.H1801: Final Report

C-Rooms

Tim Bolzern, Pascal Keusch, Raphael Husistein, Fabio Francioni
HSLU, Department Informatik, Switzerland

tim.bolzern@stud.hslu.ch
pascal.keusch@stud.hslu.ch
raphael.husistein@stud.hslu.ch
fabio.francioni@stud.hslu.ch

Document information

Project	**C-Rooms**
Document	FinalReport_Group2.docx
Authors	Pascal Keusch, Tim Bolzern, Raphael Husistein, Fabio Francioni
Professor	Dr. Angela Nicoara
Last change	20/12/2018 18:23:00

Change log

What	**Who**	**Version**
Initial start	Fabio	0.1
Created all the content	All	0.2
Finalize documentation	All	1.0

Content

1 Abstract

Open rooms are often crowded, and available seats are limited. Instead, people go from room to room to find an empty seat. The search for a spot takes time and disturbs people in the room who are studying in the rooms. This is where our project will focus on: people should be able to know the availability of a room or just simple find a free seat in a room. Our team developed a solution to monitor the room availability based on real-time data. Additionally, people can see temperature and the luminosity of a room. This data can also be used by the Janitors to maintain the rooms. The whole system is called C-Rooms whereas the "C" stands for "Clever". Due to its modular approach, it can be expanded with other sensors if requested.

The data will be made visible to the user over a web interface, so the people are able to choose a room they like to work.

Unfortunately, during the project we found out that the motion sensors used are not appropriate for our use case. At the time the problem was discovered it was already too late to adjust the solution respectively, other sensors could not be purchased in this short time. This issue is described in a separate chapter.

2 Introduction

With the C-Rooms (Clever-Rooms) project, we would like to make meeting rooms smarter. At HSLU Rotkreuz, people are often having difficulties due to crowded rooms – at the time they arrive. So, they need to seek for other (free) rooms. On the way finding some free place you'll need to open the doors and may disturb some classes. If only somehow the rooms capacity could be tracked, people could find free seats in rooms faster and more efficient. Furthermore, the capacity of the rooms would be used in a more effective and efficient way.

As the described problems may also occur in daily business, we would like to find a solution with IoT, which can remedy the situation.

In the beginning, the team thought about tracking people inside the room with a camera. This might be difficult to accomplish as people will move and expensive installations must be prepared (multiple cameras, special light installations, expensive computing hardware etc.). On the other hand, tracking people with cameras inside closed (private property) rooms will result in legal difficulties.

The second approach includes detecting incoming and outgoing people at the door with motion sensors. Privacy is guaranteed as no pictures are taken and no personal data will be stored. Additionally, with other sensors useful information about the room could be provided to the end user. Useful information can contain temperature and luminosity, so the user is able to choose the most accurate room.

The second approach will be followed and described in this report. The next chapter contains some information about related works. Chapter 4 will enclose the overall design of the C-Rooms system as well as the implementation part. In chapter 5 the solution was evaluated and tested to show the functionality of the system.

3 Related Work

Most systems available today are focusing on a more efficient reservation system for the rooms. But as the rooms are open for everyone without reservation, a reservation system doesn't solve the problem.

Hacks to create more conference room capacity:

- ➔ Here people try to create a solution for the unscheduled meeting spaces. By creating that, people might get to the meeting room even though it's already full (highfive.com, 2014).

Room occupancy to save energy:

- ➔ Case study to save energy with IoT. In the study they mention several techniques to monitor occupancy: Wi-Fi, Camera, Sensor-Network based, Data Fusion based (Akkaya, Guvenc, Aygun, Pala, & Kadri, 2015).

Room system University Lucerne (German only):

- ➔ Students can search for free places on a website. 96 places are being tracked currently (tafelrunde, 2018).

So, the system at University Lucerne is really close to the C-Room system. Nevertheless, they had to modify tables with sensors. The cost of the modifications seemed to be high. Additionally, it's not very flexible: new tables need to be modified with the sensors and it's harder to re-use such tables in other rooms if necessary as they have a fixed cable management.

4 System Design and Implementation

4.1 System Overview

Following table lists the sensors and controllers used in this project.

Sensors	
Luminosity Sensor: TSL2561	Provides the Lux value of the current environment
Temperature Sensor BME280 I2C/SPI	Provides the temperature
PIR Motion Sensor	Detects motions
Controllers	
Photon RedBoard	Used as an IoT Hub / concentrator to send data to the particle cloud

Table 1 - C-Rooms System Overview

Below drawing illustrates how the C-Room system was thought to be setup before the re-design.

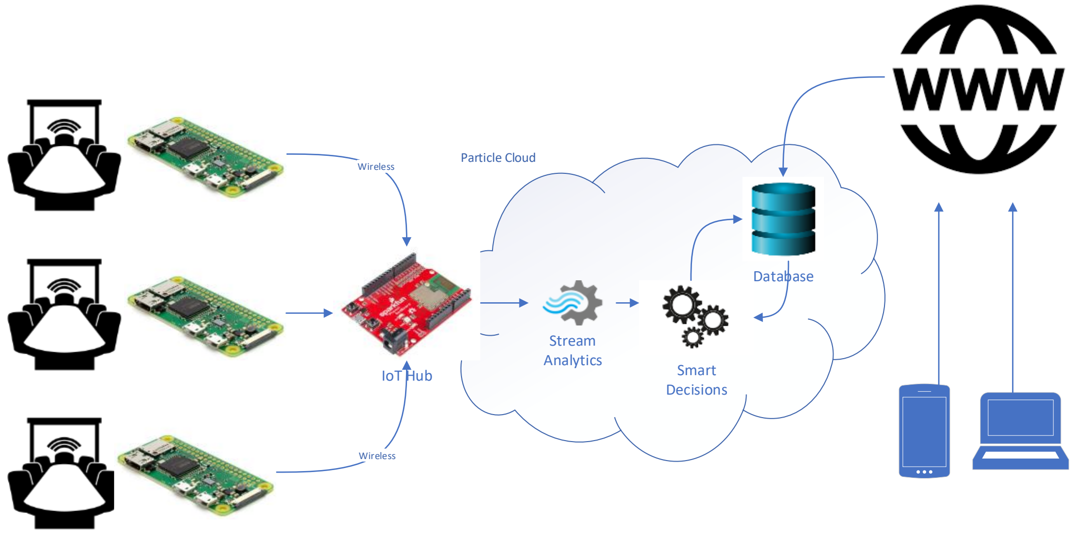

Figure 1 - Overview with Photon RedBoard

Due to its modular approach, an additional room can be added to the system easily. The Raspberry Pi Zero resides in the room and retrieves data from the sensors. It will concentrate the data and send it to the IoT Hub, in this case a Photon RedBoard from Sparkfun. The IoT Hub sends the data to the cloud where it's analysed and saved to a database. A webserver can access the database in the cloud and display the necessary data to the end user who is accessing the webserver with the laptop or smartphone. Due to complexity and a lack of time, the team found a solution to use the Photon Redboard only to send data directly to the cloud instead of using an intermediate device (Raspberry Pi Zero). The new architecture can be found below:

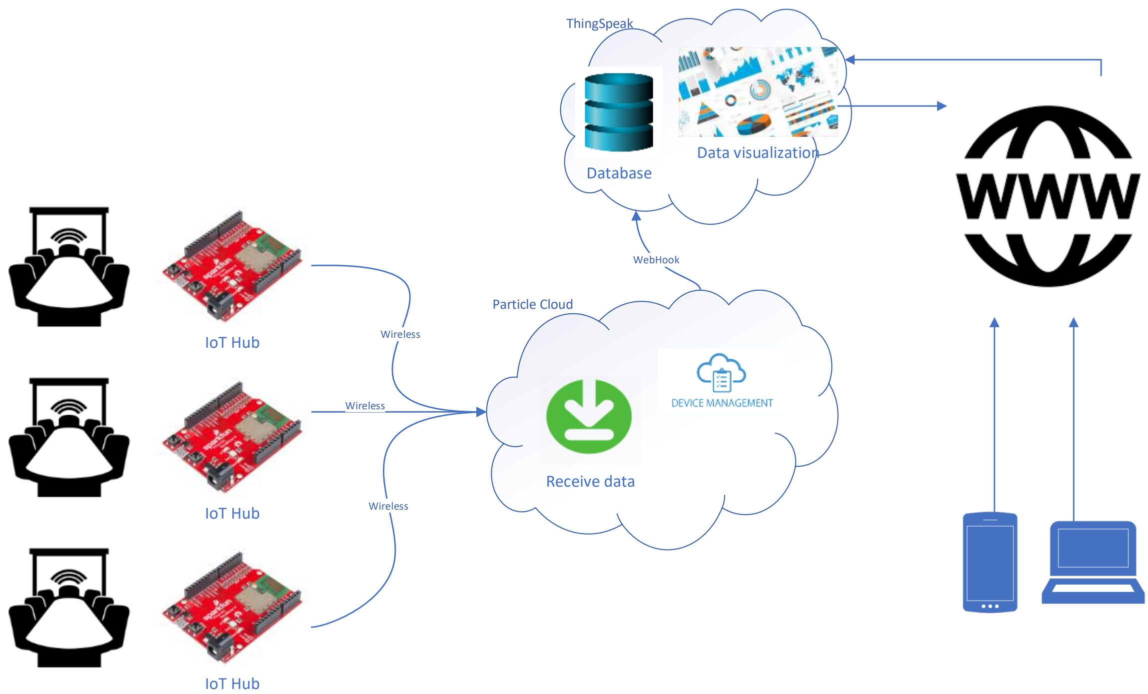

Figure 2 - Overview Photon Redboard only

This architecture is still modular and can be adapted to new installations very easily.

4.2 System Architecture

Our architecture is divided into the following parts: Board, Sensors, Cloud infrastructure, visualization. The goal is that each component could be exchanged by a more advanced / newer product if necessary. This will provide the highest flexibility to the customer. The below figure shows the different layers used:

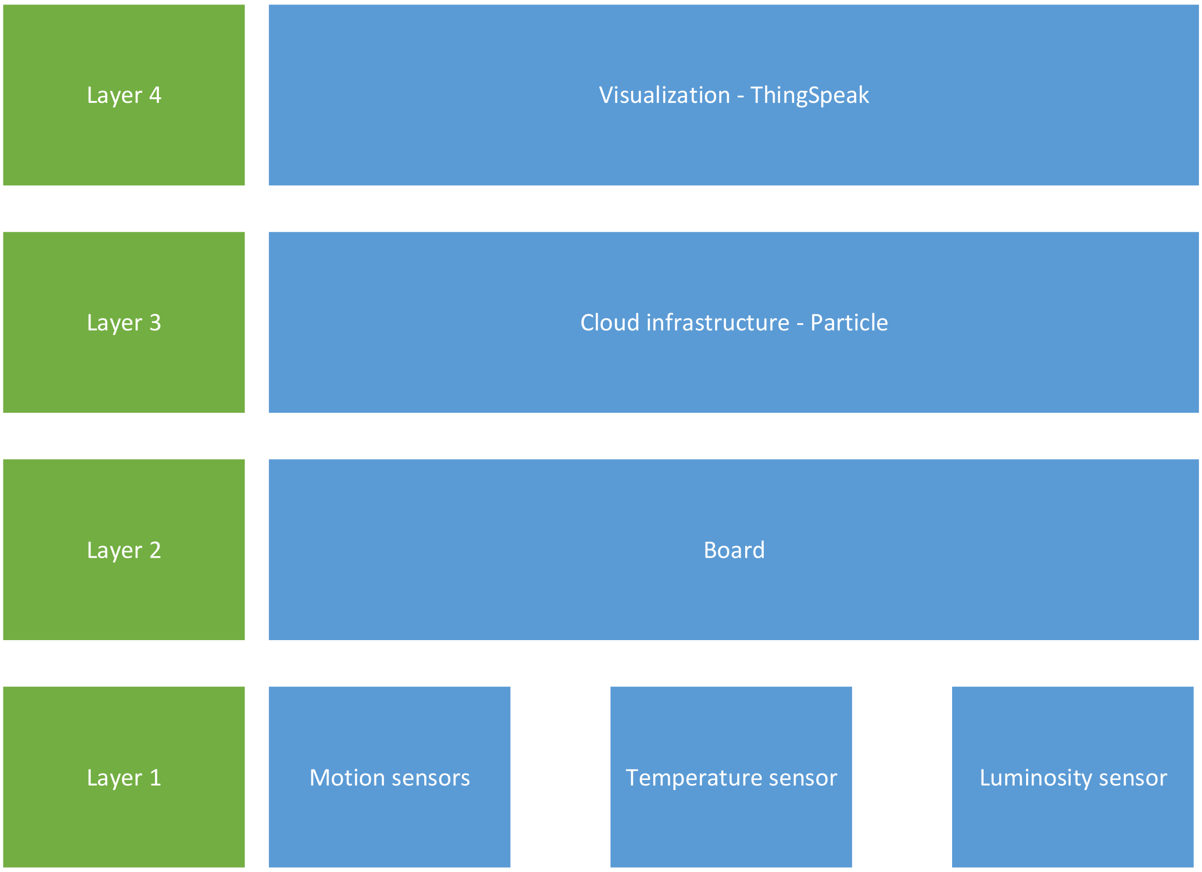

Figure 3 - System architecture layers

Unfortunately, using particle cloud has some dependencies as the infrastructure can only be used by a few boards. Nevertheless, more boards will be integrated (Raspberry Pi already integrated with Particle cloud). Particle cloud could be exchanged by any IoT service available (i.e. Microsoft IoT-Hub, AWS IoT with AWS Lambda etc.).

4.2.1 Layer 1 - Sensors

The system integrates sensors such as motion detector, temperature and luminosity sensors. They are directly connected to the RedBoard. The data it receives from the sensors will be published to the particle cloud for further processing. Each sensor generates a specific event. For instance, the temperature sensor will generate a “temperature” event.

4.2.2 Layer 2 - Board

Our solution works with a Sparkfun RedBoard and uses their cloud service called Particle. Events triggered by the attached sensors are sent to the particle cloud for further processing.

4.2.3 Layer 3 - Cloud infrastructure - Particle

Particle is a cloud platform for IoT devices. With the “Particle.publish()” function it is possible to send / push the information from the sensors directly to particle through a secure channel. Furthermore, Particle gives the ability to integrate with webhooks. This is needed as Particle cloud itself doesn't store the information received from the sensors.

4.2.4 Layer 4 - Visualization - ThingSpeak

To provide a visualization for the users, we're using webhooks on the particle cloud. The webhooks send the data based on the event further to ThingSpeak, which is used for data visualization. On ThingsSpeak it is easy to create a graph for the received events. An example of the graphs used can be found in chapter 5.

4.3 Software Architecture Layers & Modules

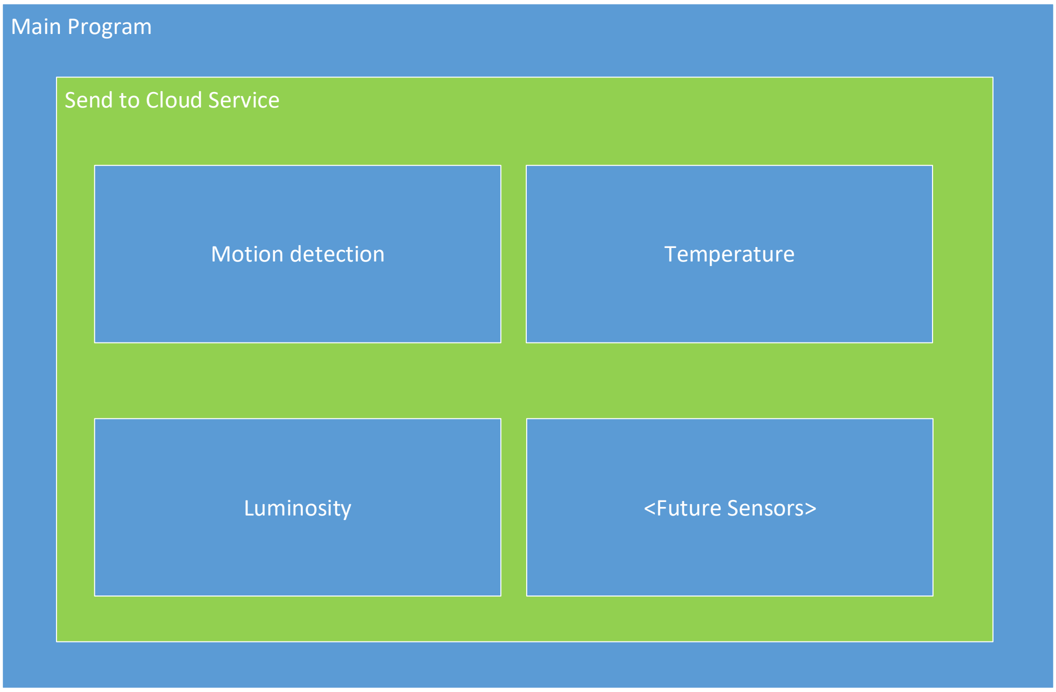

Figure 4 - Software Architecture

The system is kept simple and adjustable for further sensors. This is important as other companies might have an additional use case for the C-Room system (air quality sensors etc.). Additionally, each sensor module could be replaced by an advanced version of the sensor easily. All that data is then sent to the Particle cloud with the interface "Particle.publish()". This is the green part shown in the figure above. It is not clear what kind of additional interfaces "Particle.publish()" is using.

The main program handles all other requests and might publish the data locally to a server if requested.

Main Application – Public Interfaces	
Interface	**Description**
Provides	
Motion detection	Receives the motion sensor data
Temperature	Reads the temperature sensor data
Luminosity	Reads the luminosity sensor data
Uses	
Particle.publish()	Used for transmitting the data from the IoT device to the Particle cloud

Table 2 - Main Application Public Interfaces

4.4 System Implementation / Functional Software Architecture

With help of the Particle cloud, the written code could be flashed very easily to the Photon Redboard. The code will then be executed until the Redboard is switched off or new code will be flashed on to the board.

The data of the temperature and luminosity will be fetched every minute (60'000 milliseconds) by the main program. It uses the interface Particle.publish() to push the data to the Particle cloud. Special about the C-Room implementation is that two motion sensors are used for a higher accuracy. This is needed for detecting incoming and outgoing people. Let's assume that people are incoming. Then motion sensor 1 will be triggered earlier than motion sensor 2. This information will be counted as a decrease of the room availability. Vice versa, if people are outgoing, motion sensor 2 will be triggered earlier than motion sensor 1. This information will be counted as an increase of the room availability. This approach could not be accomplished. The issues are described in chapter 5.1.

Following table illustrates how the temperature and luminosity sensors are connected:

FROM PIN (sensor)	TO PIN (Redboard)
GND	GND
VIN	3.3V
SCK	A5 (SCL)
SDI	A4 (SDA)

Table 3 - temp. and lum. sensor connectivity

Following table illustrates how the motion sensors are connected:

FROM PIN (sensor)	TO PIN (Redboard)
GND	GND
VIN	V-USB
SCK	D0 / D2

Table 4 - motion sensor connectivity

Below picture shows how the sensors are physically attached to the Redboard.

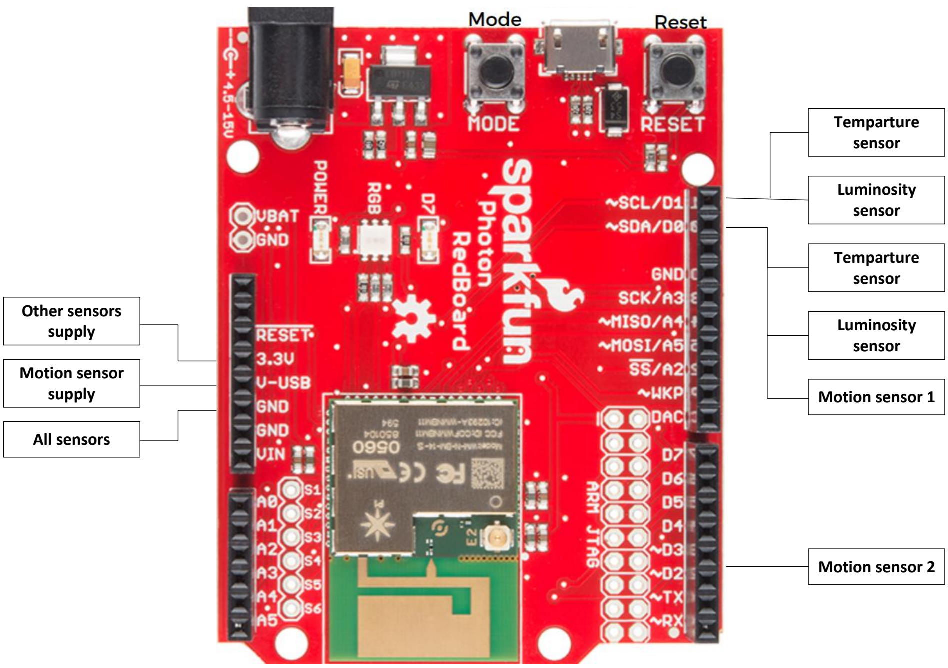

Figure 5 - sensor physical connectivity

5 Evaluation/Experiments/Results/Discussion

For testing the functionality, a split approach has been used. First, we tested if the sensors are providing the right information locally. For that, we printed out the data on the console line. This enabled fast feedback and troubleshooting possibilities. Data of temperature and luminosity sensors were manipulated manually (finger on temperature sensor, torch for luminosity sensor). After the data was validated, it was sent to Particle cloud for cloud integration. The data was compared with what was available on the console line. The last test included the communication between Particle cloud and ThingSpeak. The data received on ThingSpeak was compared with the data on Particle cloud. If they were identical, we agreed on a working solution.

Below illustration shows the temperature course during the time. Currently, there are many temperature events. For further usage, temperature should be captured every 1 minute.

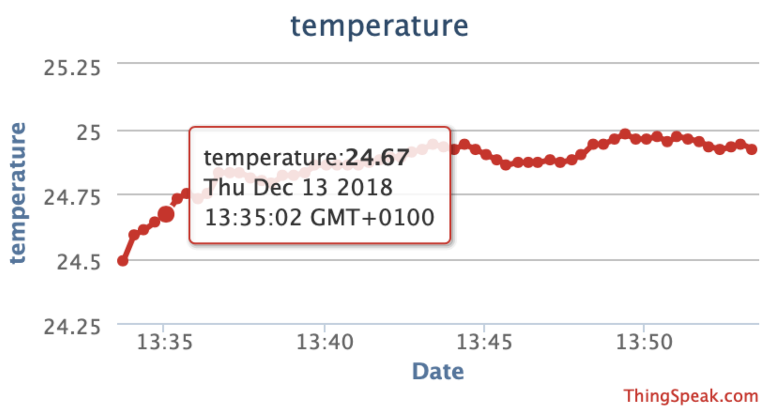

Figure 6 - Temperature Chart ThingSpeak

Below illustration shows the luminosity course during the time. Currently, there are many luminosity events. For further usage, luminosity should be captured every 1 minute (same as temperature events).

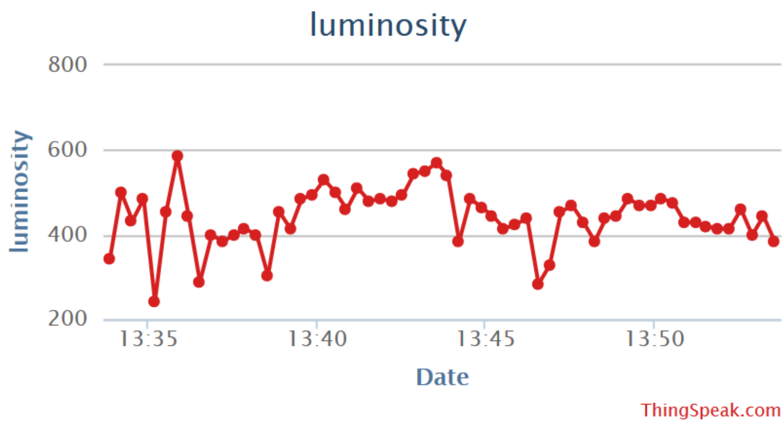

Figure 7 - Luminosity Chart ThingSpeak

Below illustration shows the number of available seats in a room. Those events are nearly real-time.

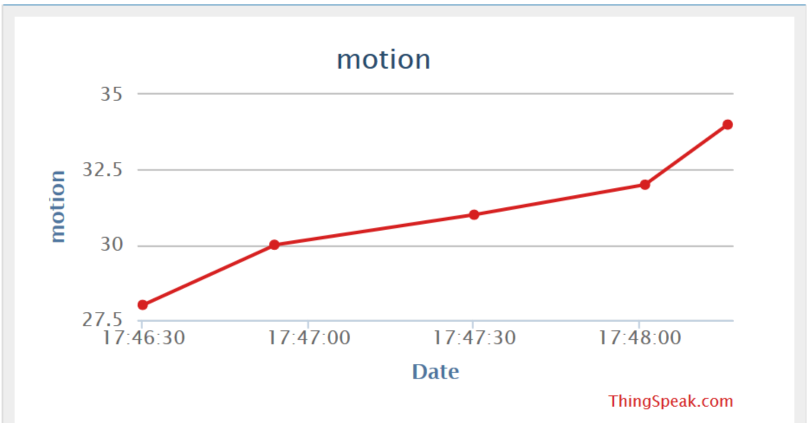

Figure 8 - Motion Chart ThingSpeak

We've been able to gather some first experiences with several types of sensors. Consequently, we found out which sensors would be applicable for a productive system as for example the motion sensors which aren't as accurate as they would need to be. One of the main problems of those sensors is their spread detecting motions. For an improved version (V2.0) of the C-Rooms system, we would take instead of motion sensors infrared sensors which don't tend to spread (see chapter 5.1 for detailed description of issues detected with motion sensor).

One major problem in IoT is the stability of the system which we discovered during our project, too. This finding does not rely on the hardware but there are issues on cloud services such as Microsoft solutions or Particle which we used or at least tried to use in our project. Several times we had lot of troubles to connect our devices to the particle cloud. During the development multiple connection issues occurred. Those could be resolved by a reset of the device – nevertheless, this was time consuming and we didn't progress as initially planned.

A big deal for us was to find the right solution. While you'll already get a lot of information in the internet it seems that often there aren't any solutions provided through the seller of the sensors or any other hardware so far. This leads that you must use some libraries, which have been written once, published on a website - as for example github - and won't be maintained anymore actively. That showed us that it would be very hard to update and provide security updates for a productive system as it would mean a lot of work. Therefore, it's not very surprising, that lots of IoT devices are insecure.

5.1 Motion sensor issues

During the project we found out that the motion sensors used are not appropriate for our use case. At the time the problem was discovered it was already too late to adjust the solution respectively, other sensors could not be purchased in this short time.

The problem occurred as the motion sensors are not intended to be used for precise motion detection. They are more likely to be used for broad detection (turning on lights in a room, activate a surveillance camera etc.). They tend to spread the signal quite broad. Additionally, we think that the sensitivity is too high which will trigger motions very fast. We would need a less sensitive detection as this might result in false positives.

In an improved version of the C-Room system, the team would use more precise sensors like an infrared barrier sensor. The project approach to detect incoming and outgoing people could be kept.

6 Application

The C-Room system can be operated in different use cases. Either schools or companies can use the system to optimize the existing capacity of available rooms. To get the most out of it, the rooms should be used as “open work spaces” – C-Rooms will then help users to find rooms with enough capacity. The user can monitor the room capacity and head then to a room with enough places.

More and more employees don’t have fix working places in their company. Therefore, each morning they must look for a free place. For this purpose, our solution would be able to find a room which fits best for each employee to work as efficient as possible.

Each eco system in a company uses a separate cloud account to guarantee confidentiality. This is important as no one should be able to modify the webhooks or events sent except the responsible of the C-Rooms system in the company itself.

As a prime example, rooms at Suurstoffi 12 - HSLU Rotkreuz could be equipped with the C-Rooms system. Students can look for free rooms on a webpage which shows the rooms capacity as well as the current temperature and luminosity in that room.

7 Conclusion

With the C-Room system, people will be able to go to a room which is not crowded and fits their needs best (temperature, luminosity, amount of people in the room).

Future work will include an app for iOS and Android, so users don’t have to check the website and can open the app easily. Furthermore, it would be possible to inform the user in advance which would be the best room to work when he’s reaching his working place. Additionally, in cooperation with a reservation system, users could pre-reserve an empty place in a (pre-defined) room as it might occur that only one place is free, and two users are heading to that place at the same time. Additionally, users should be able to select favourite rooms and will get notified when the availability changes. This part must be clarified with legal as the data might be critical. Finally, C-Rooms could propose the best room for a user depending on the current location and the predictions.

8 Contributions / Acknowledgments

Name	Contribution
Tim Bolzern	Controller to cloud communication, documentation
Pascal Keusch	User frontend, documentation
Raphael Husistein	Sensor to controller communication, documentation
Fabio Francioni	Documentation, System Architecture, Backend Architecture

Table 5 - Contributions

9 Major Milestones & Deliverables

The below table shows the project activities. Milestones are marked in blue.

Activity	State
Project brain storming	Done
Verify project idea	Done
finalize project idea and submit project proposal	Done
Learn about sensors and controllers / boards	Done
create backend architecture	Done
create frontend architecture	Done
hands on with sensors	Done
get data from temperature sensor	Done
get data from luminosity sensor	Done
get data from motion sensor	Done
publish data to the cloud	Done
visualize data to the user	Done
Finalize documentation and submit on gitlab	Done

Table 6 - Project activities

9.1 Team and Roles

The C-Rooms team consists of four qualified members with different backgrounds. Tim Bolzern and Pascal Keusch are most focused on app development whereas Raphael Husistein and Fabio Francioni have their focus on infrastructure as well as security. The work was divided to meet the area of expertise of each team member. Nevertheless, a strong skill exchange was focused to get the expertise in other fields, too.

TEAM	PROJECT WORK PACKAGES	OWNER
TEAM C-Rooms	System Architecture	Fabio Francioni
	Sensor to controller communication	Raphael Husistein
	User frontend	Pascal Keusch
	Controller to cloud communication	Tim Bolzern
	Documentation	Fabio Francioni

Table 7 - Work Packages

9.2 Project Planning, Timelines, Milestones & Deliverables

Below illustration should provide the reader an overview how the project was organized during the given time. The project started by finding a project idea and the submission of the project proposal.

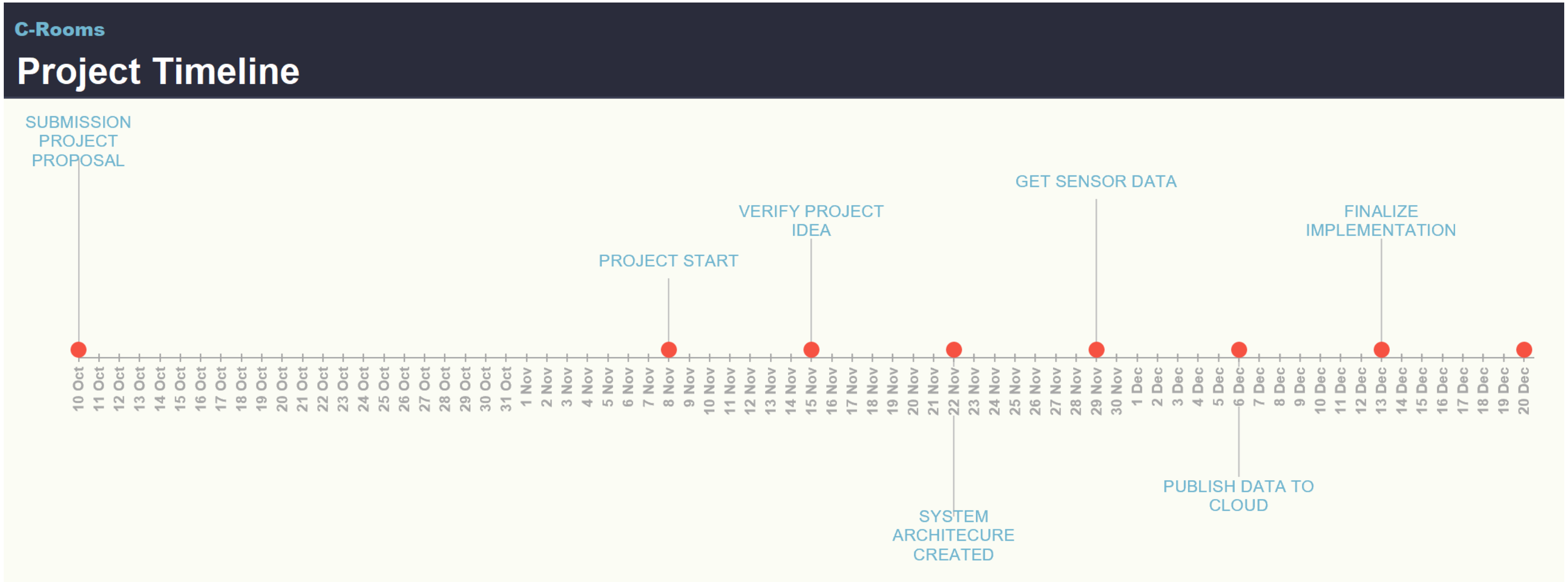

Figure 9 - Project Timeline

10 References / Bibliography

10.1 List of Tables

10.2 List of Figures

10.3 Bibliography

Akkaya, K., Guvenc, I., Aygun, R., Pala, N., & Kadri, A. (2015). IoT-based occupancy monitoring techniques for energy-efficient smart buildings. In *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)* (pp. 58–63). Piscataway, NJ: IEEE. https://doi.org/10.1109/WCNCW.2015.7122529

Highfive.com. (2014). 3 Hacks to create more conference room capacity - Highfive. Retrieved from https://highfive.com/blog/3-hacks-to-create-more-conference-room-capacity

Tafelrunde, z. z. (2018). Studenten der Uni Luzern finden mit Handy freien Platz. Retrieved from https://www.zentralschweizertafelrunde.ch/news/news-aus-der-region-_-archiv/studenten-der-uni-luzern-finden-mit-handy-freien-platz.html