Development of a Control Web Panel to Manage IoT Devices in the Real World.

Master Thesis

Jose Mir Huguet

Madrid, July 2018
This thesis is submitted to the ETSI Informáticos at Universidad Politécnica de Madrid in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering.

Master Thesis
Master Universitario en Ingeniería del Software – European Master in Software Engineering
Thesis Title: Development of a control web panel to manage IoT devices in the real world.
Thesis no: EMSE-2018-6
July 2018

Author: Jose Mir Huguet
EMSE
UPM

Supervisor:
Nelson Medinilla Martínez
Ph.D
Polytechnical University of Madrid
Lenguajes y sistemas informáticos
ETSI Informáticos
Universidad Politécnica de Madrid
Dedication

To my parents,
who have always being there supporting and pushing me specially during the hard times.

To Nelson,
because he mentored me in this project, but specially because of the closeness and kindness he provided me since the very first day.

To my professors and classmates,
That make the journey seem shorter.

To my friends,
For being there when in need, that have not been few times.

To all,
sincerely, thank you.
# Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of contents</td>
<td>4</td>
</tr>
<tr>
<td>Table of figures</td>
<td>8</td>
</tr>
<tr>
<td>Table of tables</td>
<td>10</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>12</td>
</tr>
<tr>
<td>1.1. Problem overview</td>
<td>12</td>
</tr>
<tr>
<td>1.2. Thesis objectives</td>
<td>12</td>
</tr>
<tr>
<td>1.3. Document structure</td>
<td>14</td>
</tr>
<tr>
<td>2. Project planification</td>
<td>15</td>
</tr>
<tr>
<td>2.1. Company structure</td>
<td>15</td>
</tr>
<tr>
<td>2.2. Tasks planification</td>
<td>16</td>
</tr>
<tr>
<td>2.3. Effort and timing</td>
<td>16</td>
</tr>
<tr>
<td>2.4. Budget and economic estimation</td>
<td>17</td>
</tr>
<tr>
<td>3. Project development</td>
<td>19</td>
</tr>
<tr>
<td>3.1. Sprint 0</td>
<td>20</td>
</tr>
<tr>
<td>3.1.1. Sprint plan</td>
<td>20</td>
</tr>
<tr>
<td>3.1.2. Application design</td>
<td>22</td>
</tr>
<tr>
<td>3.1.3. Implementation</td>
<td>28</td>
</tr>
<tr>
<td>3.1.4. Sprint review</td>
<td>28</td>
</tr>
<tr>
<td>3.2. Sprint 1</td>
<td>29</td>
</tr>
<tr>
<td>3.2.1. Sprint plan</td>
<td>29</td>
</tr>
<tr>
<td>3.2.2. Application design</td>
<td>31</td>
</tr>
<tr>
<td>3.2.3. Implementation</td>
<td>33</td>
</tr>
<tr>
<td>3.2.4. Sprint review</td>
<td>34</td>
</tr>
<tr>
<td>3.3. Sprint 2</td>
<td>35</td>
</tr>
<tr>
<td>3.3.1. Sprint plan</td>
<td>35</td>
</tr>
<tr>
<td>3.3.2. Application design</td>
<td>37</td>
</tr>
<tr>
<td>3.3.3. Implementation</td>
<td>37</td>
</tr>
<tr>
<td>3.3.4. Sprint review</td>
<td>37</td>
</tr>
<tr>
<td>3.4. Sprint 3</td>
<td>38</td>
</tr>
<tr>
<td>3.4.1. Sprint plan</td>
<td>38</td>
</tr>
<tr>
<td>3.4.2. Application design</td>
<td>39</td>
</tr>
<tr>
<td>3.4.3. Implementation</td>
<td>40</td>
</tr>
<tr>
<td>3.4.4. Sprint review</td>
<td>40</td>
</tr>
</tbody>
</table>
3.5. Sprint 4
   3.5.1. Sprint plan
   3.5.2. Application design
   3.5.3. Implementation
   3.5.4. Sprint review

3.6. Sprint 5
   3.6.1. Sprint plan
   3.6.2. Application design
   3.6.3. Implementation
   3.6.4. Sprint review

3.7. Sprint 6
   3.7.1. Sprint plan
   3.7.2. Application design
   3.7.3. Implementation
   3.7.4. Sprint review

3.8. Sprint 7
   3.8.1. Sprint plan
   3.8.2. Application design
   3.8.3. Implementation
   3.8.4. Sprint review

4. Project review
   4.1. Project status
   4.2. Software process analysis
      4.2.1. Things that were wrong
      4.2.2. Things that were good

5. Personal thoughts, conclusions and future work
   5.1. Personal thoughts
   5.2. Conclusions
   5.3. Future Work

6. Bibliography

Annex I - Technology used
   Technology used in the backend
   Technology used in the frontend
   Technology used for source code control
   Technology used in the server architecture
   Technology used for continuous Integration/Delivery/Deployment
   Technology used for planning and team communication
# Table of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High level product structure diagram.</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Company organizational diagram.</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>Effort of the project diagram.</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>Columns of the board</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>Architectural component model.</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>Two way ssl conceptual model.</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>Update device communication sequence diagram.</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>Django user management data model.</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>System data model found.</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>Screen flow diagram</td>
<td>28</td>
</tr>
<tr>
<td>11</td>
<td>System data model updated.</td>
<td>32</td>
</tr>
<tr>
<td>12</td>
<td>Module diagram of the system.</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>Screen flow diagram</td>
<td>34</td>
</tr>
<tr>
<td>14</td>
<td>Columns of the board</td>
<td>36</td>
</tr>
<tr>
<td>15</td>
<td>Screen flow diagram</td>
<td>41</td>
</tr>
<tr>
<td>16</td>
<td>Branch control system</td>
<td>44</td>
</tr>
<tr>
<td>17</td>
<td>Continuous Delivery workflow.</td>
<td>45</td>
</tr>
<tr>
<td>18</td>
<td>Docker image pull per environment</td>
<td>46</td>
</tr>
<tr>
<td>19</td>
<td>Component architectural schema.</td>
<td>50</td>
</tr>
<tr>
<td>20</td>
<td>Build request sequence diagram.</td>
<td>51</td>
</tr>
<tr>
<td>21</td>
<td>System data model</td>
<td>52</td>
</tr>
<tr>
<td>22</td>
<td>Remote configuration sequence diagram.</td>
<td>56</td>
</tr>
<tr>
<td>23</td>
<td>System data model</td>
<td>59</td>
</tr>
<tr>
<td>24</td>
<td>Screen flow diagram</td>
<td>60</td>
</tr>
</tbody>
</table>
Table of tables

Table of tables  
Table 1. Dedicated hours per month forecast.  
Table 2. Budget estimated to develop completely this project.
Confidentiality Clause

This master thesis contains confidential data of Barbara IoT S.L. This work may only be made available to the reviewers and authorized members of the board of examiners. Any publication and duplication of this master thesis - even in part - is prohibited. An inspection of this work by third parties requires the expressed permission of the author and the company Barbara IoT S.L.

Due to confidentiality concerns, in this document there are no images of the product, any class diagrams nor any data model concrete representation that belongs to the intellectual property of the company.

---

1. Introduction

In the first chapter it is described what is this thesis about. It contains a short problem overview, to put in context the reader, the thesis objectives and finally, how this document is structured.

1.1. Problem overview

This thesis has been produced inside the company where I started to work. The company is a start up which is trying to provide a solution to manage securely and professionally the lifecycle of any Internet of Things (from now on IoT) deployment. A fully encrypted operating system alongside with a control web panel compose a platform to manage the devices of the client. Providing a secure communication between the devices and a control panel, grant to the companies to monitor and to manage their devices remotely. A simple conceptual model is shown in the figure 1.

![High level product structure diagram.](image)

The basics of the system are decomposed in two main parts, the control panel, and the secure operating system. The clients only have to develop their applications and place them on the top. The communications and the operating system of the devices are fully encrypted, therefore the clients' applications running over them are securized too.

1.2. Thesis objectives

The goal of this thesis is to develop a solution of quality for a problem in a real life environment as it is a company, just as simple as that. Developing a project in a company, far from the academic world where I came from. Brings new constraints to the equation, such as short timings, tight budgets, to deal with work colleagues, to deal with clients, to be dependant of the changing business goals,... The result should be to deliver the best product possible with the resources given.
The objective is to create a control web panel to manage securely and professionally the lifecycle of any IoT deployment. To achieve this objective, the produced tool must have the following functionalities:

- **User management**
  Register and delete users, let users login and logout, manage user sessions, edit user information, change and recovery user's password,...

- **Device operations**
  Create and remove deployments, create and delete devices, assign and unassign devices to/from deployments, manage the information of the devices or the deployments, live telemetry, check historical data,...

- **Build images**
  As build a new image of the operating system needs so much resources, it will be handled by another server with several CPUs somewhere in the cloud. The control panel should be fully integrated with this other server API.

- **Remote Updates**
  From the control panel it will be possible to send the images built in the step before to the devices so they update when they receive the instruction.

- **Remote Configuration**
  Similar to the previous epic, the devices must be configured remotely from the control panel. The user has a section in the control panel where the configuration of the devices can be changed and applied.

- **Manage company managers**
  The manager of the company should be to create new managers, and assign permissions to each of them.

- **Alert System (for clients)**
  The system will warn the user when certain events occur. It could be some physical events, such as the device is physically manipulated. Telemetry events, such as when a device reaches certain CPU temperature level. Security events, such as when a device suddenly disappears,...

- **Logging, monitoring, reporting and internal alarm system**
  To have full control of our system, prevent or recover from possible issues, it is need to have all the information possible.

- **Backup and recovery system**
  As a security company aware of the threats of the internet, we need a disaster recovery system, it is mandatory for us.

- **Telemetry data management**
  Each device sends hundreds or thousands of reports per day to our datacenter, there is a physical need to handle this huge amount of data in the better way possible.

- **Full support any service in any cloud provider**
  Our business is splitted in different services in the cloud, but we need to be able to deploy any service in any cloud, to not be cloud provider dependant and to be able to
offer the maximum options for our clients. The cloud provider should not be ever a constraint.

- **API for clients**
  Most of our clients may have their own control panels for their solutions, so they may not want to use our panel in addition. All the information that we handle, must be available for them to be consumed and integrated in their own backends.

All these functionalities are demands from the business point of view that the control panel must satisfy. Behind any of this list, several requirements will appear, and many others may even be invisible for the business or client parts. But discover and implement these new requirements will be my task during the project development.

This thesis will recopilate the complete project lifecycle, documenting in each iteration, from the phase of obtaining the requirements of the system, to the phase of delivering and deploying the product in production. Despite it may be mentioned in the document, the validation and verification with users(clients) is out of the scope of this thesis, mainly because I was not in charge to perform these tests.

### 1.3. Document structure

This document is composed of 6 chapters, each one of them contains the following:

- **Introduction:** The first chapter is just a quick presentation of the work to be done. A brief overview of the project and an explanation about what is going to be performed in this project.
- **Project planification:** In the second chapter there is a put in context scene, trying to explain the company and the timings that they have. Also there is an economic estimation of the work done.
- **Project Development:** In this chapter it is explained how the project was built. It goes sprint after sprint explaining what was done (or not) in each one. The structure is the same for each sprint explained, where there are the sections: sprint plan, application design, Implementation and sprint review.
- **Retrospective, Future work and Conclusions:** After explaining what was done, in this chapter there is a retrospective comments, at a glance the most remarkable things I achieved. The pending work to do in the next months and the final conclusions with the lessons learned.
- **Bibliography:** Finally there is a chapter listing the resources used to write this thesis.
- **Annex I - Technology used:** This annex contains an explanation of the technology, methodology or other elements used in the project. In further chapters there is not any explanation of what any term is for nor how I use it. I think it was interesting just in case the reader is not familiar with any of the terms used.
2. Project planification

This chapter explains how the project is meant to be accomplished. It contains a short description of the project objectives, how the company is organized, who is taking each role and tasks, the temporalization of these tasks and an economic estimation of the project.

2.1. Company structure

The company is a startup, so there are some roles that don’t exist, and some persons who have several roles. The figure 2 contains the organization chart:

![Company organizational diagram.](image)

My role in the company is to implement the control web panel all by myself. I will only communicate with the product manager to get/manage the user stories and with the engineer
lead form the other team to organize and design the communications between the operating system and the control panel. The schema describes how it is meant to be, but after all, we are a small, agile and dynamic company, so this schema is not so rigorous. There are fluent communication between all the stakeholders anytime is needed.

2.2. Tasks planification

The tasks identified to be done in this project are three at a high level:

- **Initial analysis**, this was already done when I arrived to the company. It was carried out by the directors. They defined how the product should be at a extremely high level.
- **Project development**, the bigger task, it contains all the subtasks needed to develop the project. It will be an incremental iterative repetition of requirements analysis, design, implementation, testing, delivering and deploying.
- **Maintenance phase**, once the necessary functionality has been implemented and exposed to the clients, these new functionalities/issues identified are going to be developed or fixed.

2.3. Effort and timing

From business I had the constraint to finish the project before the end of the year, but if it was possible finish before, so it could be presented to the clients sooner, it would be much better. The plan is quite concrete and I estimated it to be ready by the summer, and add small functionalities that the company found necessary afterwards. Also fix the issues that it may appear. In the figure 3 it is visible how the effort is distributed.

![Figure 3. Effort of the project diagram.](image)

The red line in february represents when I entered in the project. After a few weeks, when the person who started the project transferred me all the knowledge, he left the company. So I started working “alone”. Not really alone physically, but alone working in this project. So far (end of june) the planification has not changed and the project aims to be finished on time.
2.4. Budget and economic estimation

First of all I have to say that I have a contract there and that I have a monthly payroll, but if I had to charge somebody for the project described in this document would be something similar to these estimations. This is not a real invoice, is more an exercise to learn to value my job. This is the estimation of hours per month according to the plan:

<table>
<thead>
<tr>
<th>Month</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>100</td>
</tr>
<tr>
<td>Feb</td>
<td>160</td>
</tr>
<tr>
<td>Mar</td>
<td>160</td>
</tr>
<tr>
<td>Apr</td>
<td>160</td>
</tr>
<tr>
<td>May</td>
<td>160</td>
</tr>
<tr>
<td>Jun</td>
<td>160</td>
</tr>
<tr>
<td>Jul</td>
<td>160</td>
</tr>
<tr>
<td>Aug</td>
<td>160</td>
</tr>
<tr>
<td>Sep</td>
<td>110</td>
</tr>
<tr>
<td>Oct</td>
<td>70</td>
</tr>
<tr>
<td>Nov</td>
<td>50</td>
</tr>
<tr>
<td>Dec</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 1. Dedicated hours per month forecast.

This would be the price per hour and the total cost of the project.

<table>
<thead>
<tr>
<th>Description</th>
<th>€/hour</th>
<th>#hours</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project kick off (January)</td>
<td>€19.00</td>
<td>100</td>
<td>€1,900.00</td>
</tr>
<tr>
<td>Project development (Feb - Aug)</td>
<td>€38.00</td>
<td>1120</td>
<td>€42,560.00</td>
</tr>
<tr>
<td>Project maintenance (Sep - Dec)</td>
<td>€76.00</td>
<td>250</td>
<td>€19,000.00</td>
</tr>
</tbody>
</table>

Table 2. Budget estimated to develop completely this project.
I did some research on pricing, looked for some freelancers located in Spain, and the price range was from 30 to 70 € approx. It is true that the price is for pure development, not for business understanding, nor requirements analysis, nor system integrations or deployments, nor validations or verifications,…. A freelancer with this skills and tasks would cost much more than that, but as I don’t have much experience carrying projects on my own, I don’t think I could charge much more than that.

The price was estimated for project development stage, but here in this case we have a project kick off and a project maintenance. As I personally I want to understand completely the problem before start working on it, I set the project kick off as half price because I think this time should be mainly meetings and investigation trying to answer the what and the how. The maintenance phase is not for bug fixing, it is more for bring small functionalities to the project. After releasing it to the clients, the company may detect some lack of functionalities or maybe the need to modify some others. As these tasks are planned be just temporal and short, as a professional I should reserve some time of each month to this, as I could not take any other full time jobs, I would charge twice the price of maintenance.
3. Project development

The third chapter of this document is dedicated to explain all the process of the software creation. It is going to review throughout the sprints, explaining in each one of them what, how and why things are done.

Tasks distribution
To make things clear, I would like to mention that I did everything described in this document excepting:
- Creation of MQTT server, it was created before I arrived.
- Creation of Azure Database, it was created before I arrived.
- Technology stack, it was decided before I arrived.
- Anything related to User Experience design. There is a UX expert in the company who decides everything related to this matter.
- QA process. Is assured by another person(Product Manager).

The product manager was in charge of write the user stories, but I am able to create them too. The requirements that are more strict from business are created, mainly, by the product manager. I mainly create the more technical stories.

The communication between the control panel and the devices (Operating System) is designed together with the guys from the other team.

Important considerations
This project has been developed in a start up, therefore it is needed to sell the product soon to not run out of funds, it is needed to fail as cheap as possible for the same reason, but above all, get quick feedback of the product. After all, this is a new concept, the idea needs to be validated in the real world beyond the previous market studies.

The business idea is not completely defined, so, from business, they need to get feedback soon. The priority is to bring as much functionality and visibility to the product possible, the sooner the better. That means that there are other tasks that also bring value to the product, which are not visible to the clients, that have to be left aside.
Methodology used

The methodology that I am going to use is an agile approximation. Due to the size of the development team (just me) it makes no sense to use strictly a methodology. I will do just the necessary in order to organize the work, but also to let all the stakeholders obtain quick feedback of the development status.

As long as the project is not a defined product with a fixed requirements, and we will define the product during its development, at the beginning I have defined some epic stories, which will be decomposed in several user stories. Anybody related to the project can submit new stories, tasks or bugs. Me together with the product manager will define the user stories to implement during each sprint at the beginning of it. The tool used to create these new stories is Atlassian Jira. The board used will have the columns detailed in the figure 4:

<table>
<thead>
<tr>
<th>TO DO</th>
<th>IN PROGRESS</th>
<th>READY</th>
<th>DONE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. Columns of the board.

All the tasks planned for the sprint are going to the TO DO column. When somebody starts to work with a card, it is moved to the IN PROGRESS column. When it is finished, the card is moved to READY. Then the QA team (also just the product owner) verifies and validates that the card is properly finished, and moves it to the DONE column.

3.1. Sprint 0

This sprint was not a proper sprint, I just landed in the company, in the middle of it. Some things were done, some didn’t, and there was a demo with a client in less than a week. Since the very first day I started working with the pilot.

3.1.1. Sprint plan

The team had prepared the following stories for the control panel:
- As an Admin I need the ability to create Manager users
- As an Admin I need the ability to create new Companies to group different deployments
- As a User I want to be able to list devices
- As a User I want to be able to see device details
- As an Admin I need the ability to list users
- As an Admin I need the ability to securely login
- As a Manager I need to create new Deployments to manage grouped devices
- As a Manager I need the ability to create devices
- As a Manager I need to be able to assign devices to specific deployments so we can group them
- As a Manager I want to send remote updates to my devices
- As a Company Manager I want to create Deployment Managers in my company
- As a Manager I need to be able to assign Users to a specific deployment
- As a User I want device details site to refresh automatically when information change
- As a Manager I need the ability to securely log in to the Web Panel

Python comes with an admin panel out of the box to make changes to the entities of the data model. Therefore, some of the tasks were already implemented at the moment 0. When I arrived to the company, some of the tasks were already implemented in the backend, but the front-end had only the basic html code, with no styles at all. My main task, for the moment, was to bring some css to the pages that were created. We had a demo with a client by the end of the sprint, I had to hurry a bit to improve the look and feel.
3.1.2. Application design

This is what I found when I joined the project. The only thing was not set yet was the server for the control panel application, but was on the scope of this sprint to do it. In this sprint I was only working on the UI.

System architecture

Since the very beginning I was sure that all the deployments should be in the cloud from scratch despite the company has their own datacenter at the office.

The system is currently using Microsoft Azure and Amazon web services as cloud providers for other services. I stated with these per convenience, but it is on the roadmap to support any cloud provider. In the figure 5 there is an schema of how the architecture is defined.

![Architectural component model](image)

Figure 5. Architectural component model.
The devices currently send their telemetry status to Azure Iothub through MQTT. There, the messages are received and processed by Data Stream Analytics tool which is configured to store all this data in a Microsoft SQL Server database. The control panel reads this information directly from this database.

In another instance of Amazon cloud, we have deployed our own MQTT broker. Who is in charge to handle all the communications between the control panel and the devices. This communication is doubly authenticated, the devices and panel verify the certificates from the MQTT server, and the MQTT server verifies the certificates from the devices and the control panel.

Communication design

The control panel has to communicate securely with the devices through MQTT protocol. Also with the build server, but in this case is using HTTPS. In any case, all connections are two way authenticated with certificates. The schema is the following, in figure 6:

Using this schema, we not only assure that the sender is who has to send the information but also that the receiver of the message is who is meant to receive it.

---

2 “IBM Knowledge Center - Two-way SSL Authentication”
The communication created up to this point is to update over-the-air the devices. Remote Update communication will occur every time that a user wants to send an update to a device. It will send the location of the image file and its checksum, the device will download it, verify the file and apply it. The communication is described in the sequence diagram of figure 7:

Figure 7. Update device communication sequence diagram.
Logical design of the application

The data model

In order to develop this project, they decided to use the django framework. As I am using a framework for the development, some parts are built for us. Here, in figure 9, there is the data model for the user management provided by django:

The framework provides built-in user secure authentication and user operations such as login, logout or registration. It also provides its own permission system for the users, as it can be appreciated in the previous figure. It lets to manage the permissions at user level, group of users level or object level.
The proposed data model of our system was the following, in figure 10:

![Diagram of the system data model](image)

This schema represents the data model of the system that was created. There is a Manager entity which is linked with the django user entity provided by the framework. This manager, is never used in the system though, the system uses the entities deployment manager and company manager to refer to the two roles that are defined now in the system. There is a company entity which can have a company manager (just one) and many deployments. A company manager also belongs to a company. A deployment has a deployment manager and can contain many devices. It can also have tags. The devices can have tags too, and properties describing themselves, and specific for each deployment and for each company.

Figure 9. system data model found.
Even though it is being used this schema, I noticed that it has some issues, it has some cyclic relationships and a mixture between classes and entities. A company manager and deployment manager “inherit” from the manager model, the same happens with the properties. But fixing this is out of the scope of this sprint, I will do it in the next sprint. The software is already working with the data loaded for the demo planned for the end of the sprint. All the efforts are going just to improve the look and feel.

User experience design
As I mentioned before, all the user experience was and is being designed by the UX expert. I do not do anything in this part. The design of the user interface, was agreed between the UX expert, the product manager and the directors of the company as business team (CMO + CTO). During several meetings they came up with all the functionalities that the product must have and how them should be placed in the webpage, afterwards the UI designer expert came up with a design of the templates. The result was some photoshop templates, that I should replicate in the real software. The designs contained many functionalities that were not developed yet, so the final product cannot look equal than the designs at this moment.

The screen flow diagram
So far the screens implemented have the navigation described in figure 11:

![Screen flow diagram](image)

Figure 10. Screen flow diagram.

For this sprint, the functionality is very limited. All the information of the system is introduced by the project manager, even including usernames and passwords. The user can login to the system and in the deployment page, depending on the role of the user can see more or less deployments listed. If the user is a company manager, all the deployments of the company will be listed, but if the user is a deployment manager instead, the system will only show the deployments that the user is assigned to. When a user clicks into a deployment, all the devices linked to that deployment are listed, and if the user clicks into a device, then all the information of that device is listed.

The real devices send all the health information to an Azure server, they send usage of CPU, RAM and Memory. When they have the sensors installed, they also send other information like temperature, humidity, GPS position,... All this information is shown in the device page, together with the software version and the tags of the device. The tags are just mock data, but all the other information is real, a process in the backend is consuming the Azure SQL Server database in order to obtain all the information about that device.
3.1.3. Implementation

Code
The stack of this pilot is built in django 2.0.2 using HTML5 and CSS3 for the web templates. It is used bootstrap and jquery to help with the presentation and the logic in the front. The data to the templates is bounded directly from Django to the template, and when the browser renders the view, it renders also the data.

Deployment
In order to make the code public available, it is served in Amazon Web Services. An instance in EC2 was bought and the code ran there after installing python 3 and the project dependencies.

3.1.4. Sprint review
At the end of the sprint, the team was quite satisfied with the results even though not all tasks could be implemented. But it was enough to show the basic functionality of the system to the client with the demo.

Django framework provides an administration panel where all the database entities can be managed easily. That tool allows to the administrators to read, create, update and delete any data from the database. For that reason, the following tasks were already implemented from the beginning:
  ● As an Admin I need the ability to create Manager users
  ● As an Admin I need the ability to create new Companies to group different deployments
  ● As an Admin I need the ability to list users
  ● As an Admin I need the ability to securely login

This (half) sprint, were finished the following tasks:
  ● As a User I want to be able to list devices
  ● As a User I want to be able to see device details

These tasks weren’t done:
  ● As a Manager I need to create new Deployments to manage grouped devices
  ● As a Manager I need the ability to create devices
  ● As a Manager I need to be able to assign devices to specific deployments so we can group them
  ● As a Manager I want to send remote updates to my devices
  ● As a Company Manager I want to create Deployment Managers in my company
  ● As a Manager I need to be able to assign Users to an specific deployment
- As a User I want device details site to refresh automatically when information change
- As a Manager I need the ability to securely log in to the Web Panel

The not finished cards, will be moved to the backlog so they can be introduced in the next sprint. Despite of it may seem that not many stories were implemented, the truth is that there were many tasks that weren’t listed as a task in the sprint but were done to, many of them appeared at the moment of being implemented.

The demo with the client was also useful to get new requirements and needs for future users. It was also used as a first validation of the product with a real customer. In this short sprint, all the code was already done, it was just to apply css styles mainly. There weren’t much more problems rather than understanding a bit what the framework was doing in the dark.

3.2. Sprint 1

After the demo with the client, in this sprint we were more focused on improve our product and solve some issues that we had identified. Therefore this new set of tasks was agreed to be implemented:

3.2.1. Sprint plan

Here there are the tasks for this sprint ordered by priority:
- As a Manager I need to create new Deployments to manage grouped devices
- As a Manager I need the ability to create devices
- As a developer I want to review all relationships in the backend module to optimise queries and user experience
- As a developer I want to review the rules, permissions and ownership validations in order to enhance the security of the system
- As an Admin I need the ability to securely login
- As a Manager I need to be able to assign devices to specific deployments so we can group them
- As a Manager I want to send remote updates to my devices
- As a Manager I need to be able to assign Company Managers to an specific deployment
- As a Manager I need the ability to securely log in to the system
- Abuse Stories Panel; Sprint 1
- As a PO I want Web Panel integrated with Azure backend
- As a Manager I want to be able to configure iptables of my devices remotely using the web Panel
- As a Real Client CTO I want the web Panel to have a change log section indicating the changes in the latest version released
• As a Real Client Project Manager I want vehicles plate indicated in Web Panel so I can identify them quickly
• As a Real Client Project Manager I want an extra column in deployment visualization indicating UICCID of the SIM
• As a developer and as a Security Manager, I want to have a development-integration server in the Office so we can deploy releases in a safe environment, but also test them for security breaches from the Internet
• Remote updates messages sent at page loading when listing the devices of a deployment
• As System Admin I would like to be able to manage my remote updates parameters in Panel Admin Area so I have more control over deployments
• As deployment manager I want to add properties to the devices in order to identify specific attributes of each device
• As a Deployment Manager I want to have an alert icon next to the corresponding device in the deployment list when an alert is launched
• When a just created user try to create the first deployment, Panel report an error
• As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc
• As a PO I want to access the relevant sections using shortcuts in the left side menu
• As a User (of any type) I want an option to change my password
• As a user (of any type) I want to have a section where I can edit my user information
• As a PO I want a flow diagram describing Panel functionalities
• As a Deployment Manager I want to see the Alerts on each device coming from real data

This sprint contains much more cards than the previous sprint but the goal is not to finish all of them now. This is the first try with a real sprint for me and the company so we try to keep the expectancy not much high.
3.2.2. Application design

Here I started already to change things that I didn’t like much how were done.

Logical design of the application

One of the main tasks of the Sprint was to update the data model and remove all the issues found previously. Here appears the new data model proposed in figure 12:

![System data model updated.](image)

Nothing changed regarding user Authentication, I am still using the default user entity provided by the framework. But now there is a common manager entity for all the roles and is it possible to assign roles to each manager. This data model is based on a company as owner of everything.

A company may have one or several managers, one or several deployments, one or several devices and one or several properties. Then all this entities have relationships among them. The information of the entities didn’t change much from the previous model, but relationships and its cardinality did. In the new data model it was also removed this mixture with class models and inheritance, now the data model is independent from its code implementation.
The module diagram
In order to keep consistency across the design, I tried to follow a similar approach to the one
given by default by the framework for the default views. In figure 8 there is a module diagram of
the system:

![Module diagram of the system.](image)

This is the general schema of the system. The top layer is provided by the framework with the
default classes for the default views. The extra functionality that I added, I tried to keep similar
to the framework style. Each new functionality that I needed, I just added new modules to
perform the new operations. I didn’t painted all the modules, because I think it doesn’t bring
much having here listed all of them, there are over 50 in total.

User experience design
No new screens were designed by the UI expert for this sprint, all the new functionalities were
designed by me. In the middle of the sprint, there was a meeting with the UI expert where all
these new functionalities were explained to her and what was already done shown, so she could
design all the views of the platform for the next sprint.
The screen flow diagram
In the figure 13 there is a diagram of the navigation for the screens created so far.

![Screen Flow Diagram](image)

Figure 13. Screen flow diagram.

The diagram now has a few more screens. I added two screens to create deployments and devices. These views have access control, only certain users can reach them. The new view for user management allows company managers to assign and unassign deployment managers to deployments. The view of device management allows users with the proper privileges to assign devices to deployments. The new functionality to add properties to devices is encapsulated in the views devices on create device.

3.2.3. Implementation

Code
In this spring I went a bit deeper in django framework as my knowledge of it was improving. But nothing really affected. The code just gets a bit simpler because I was using the proper functions now.

Deployment
Nothing changed here. But we agreed with the product manager to deploy what I had developed and tested every week, friday morning preferably.
3.2.4. Sprint review

This sprint was quite good, at the middle of the sprint we had to rearrange the priority of the tasks because I realized that some of them had cross-dependencies. As a consequence, some tasks are started, but couldn’t be finished completely. These are the tasks that could be done:

- As a Manager I need to create new Deployments to manage grouped devices
- As a Manager I need the ability to create devices
- As a developer I want to review all relationships in the backend module to optimize queries and user experience
- As a developer I want to review the rules, permissions and ownership validations in order to enhance the security of the system

The rest of the tasks, even though some of the were started, couldn’t be finished. Here is the list:

- As an Admin I need the ability to securely login
- As a Manager I need to be able to assign devices to specific deployments so we can group them
- As a Manager I want to send remote updates to my devices
- As a Manager I need to be able to assign Company Managers to an specific deployment
- As a Manager I need the ability to securely log in to the system
- Abuse Stories Panel; Sprint 1
- As deployment manager I want to add properties to the devices in order to identify specific attributes of each device
- As a PO I want Web Panel integrated with Azure backend
- As a Security Manager I want to be able to configure iptables of my devices remotely using the web Panel
- As a Real Client CTO I want the web Panel to have a change log section indicating the changes in the latest version released
- As a Real Client Project Manager I want vehicles plate indicated in Web Panel so I can identify them quickly
- As a Real Client Project Manager I want an extra column in deployment visualization indicating UICCID of the SIM
- As a developer and as a Security Manager, I want to have a development-integration server in the Office so we can deploy releases in a safe environment, but also test them for security breaches from the Internet
- Remote update messages sent at page loading when listing the devices of a deployment
- As a System Admin I would like to be able to manage my remote update parameters in Panel Admin Area so I have more control over deployments
- As a Deployment Manager I want to have an alert icon next to the corresponding device in the deployment list when an alert is launched
- When a just created user try to create the first deployment, Panel report an error
As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc
As a PO I want to access the relevant sections using shortcuts in the left side menu
As a User (of any type) I want an option to change my password
As a user (of any type) I want to have a section where I can edit my user information
As a PO I want a flow diagram describing Panel functionalities
As a Deployment Manager I want to see the Alerts on each device coming from real data

I proposed to the product manager to involve more the UX expert in the development and the design of the screens during the sprint, not just at the beginning or at the end of it. The main goal of it is to not redo or directly throw away parts of the software as it happened in this sprint.

### 3.3. Sprint 2

After realizing and reporting the problem with the design of the templates (UI), I proposed to use the board shown in figure 14:

<table>
<thead>
<tr>
<th>TO DO</th>
<th>IN PROGRESS</th>
<th>UI READY</th>
<th>READY</th>
<th>QA DONE</th>
<th>DEPLOYED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 14. *Columns of the board.*

The proposal was accepted. The changes are mainly two columns, UI READY and DEPLOYED. Now after starting and analyzing a task, if it requires any UX elements, the task will be assigned to the UX expert in order to generate the proper designs. Once she finishes, the task moves to UI READY and I take it and implement it. This will avoid the efforts done placing the elements of the UI without being designed.

After a card passes the QA acceptance, I will take it again and deploy it to production environment, once is done, the card will be moved to DEPLOYED column. This will give the necessary visibility of what has been deployed or not.

### 3.3.1. Sprint plan

The goal of this sprint was to try to close all the stories from the previous sprint. Some of them were rearranged depending on the priority and just few bugs were added. Here is the list of stories for this sprint:
● As an Admin I need the ability to securely login
● As a Manager I need to be able to assign devices to specific deployments so we can group them
● As a Manager I want to send remote updates to my devices
● As a Manager I need to be able to assign Users to an specific deployment
● As a Manager I need the ability to securely log in to the Web Panel
● As a Client Project Manager I want vehicles plate indicated in Web Panel so I can identify them quickly
● As a Client Project Manager I want an extra column in deployment visualization indicating UICCID of the SIM in the box
● As a deployment manager I want to add properties to the devices in order to identify specific attributes of each device
● When a just created user try to create the first deployment, Panel report an error
● As a PO I want to access the relevant sections using shortcuts in the left side menu
● As a User (of any type) I want an option to change my password
● As a user (of any type) I want to have a section where I can edit my user information
● As a PO I want to change the label "Devices" in the left side menu and put "Manage Devices" instead
● Abuse Stories Panel; Sprint 1
● As a PO I want Web Panel integrated with Azure backend
● As a Security Manager I want to be able to configure iptables of my devices remotely using Web Panel
● As a Client CTO I want Web Panel to have a change log section indicating the changes in the latest version released
● As a developer and as a Security Manager, I want to have a development-integration server in the Office so we can deploy releases in a safe environment, but also test them for security breaches from the Internet
● Remote update messages sent at page loading when listing the devices of a deployment
● As a Web Admin I would like to be able to manage my Remote update parameters in Panel Admin Area so I have more control over deployments
● As a Deployment Manager I want to have an alert icon next to the corresponding device in the deployment list when an alert is launched
● As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc
● As a PO I want a flow diagram describing Panel functionalities
● As a Deployment Manager I want to see the Alerts on each device coming from real data
● When accessing the home page while logged in as an Admin
3.3.2. Application design

There were no changes affecting the design.

3.3.3. Implementation

Testing

There is a security team in the company which is preparing some security tests to the platform, but couldn’t run them yet. If they run them straight to our servers in AWS, Amazon may detect them as an attack, and block the communications. They are trying to contact with AWS to check if they can perform these tests. If they cannot do it to our production server, they will run the tests against the server at the office in our intranet.

3.3.4. Sprint review

In this sprint some long and complex tasks were completed, but not all of them. This is the list of stories completed:

- As an Admin I need the ability to securely login
- As a Manager I need to be able to assign devices to specific deployments so we can group them
- As a Manager I want to send remote updates to my devices
- As a Manager I need to be able to assign Users to an specific deployment
- As a Manager I need the ability to securely log in Web Panel
- As a Client Project Manager I want vehicles plate indicated in Web Panel so I can identify them quickly
- As a Client Project Manager I want an extra column in deployment visualization indicating UICCID of the SIM in the box
- As a deployment manager I want to add properties to the devices in order to identify specific attributes of each device
- When a just created user try to create the first deployment, Panel report an error
- As a PO I want to access the relevant sections using shortcuts in the left side menu
- As a User (of any type) I want an option to change my password
- As a user (of any type) I want to have a section where I can edit my user information
- As a PO I want to change the label "Devices" in the left side menu and put "Manage Devices" instead

These stories could not been completed in time at the end of the sprint:

- Abuse Stories Panel; Sprint 1
- As a PO I want Web Panel integrated with Azure backend
- As a Security Manager I want to be able to configure iptables of my devices remotely using Web Panel
• As Client CTO I want Web Panel to have a change log section indicating the changes in the latest version released
• As a developer and as a Security Manager, I want to have a development-integration server in the Office so we can deploy releases in a safe environment, but also test them for security breaches from the Internet
• Remote update messages sent at page loading when listing the devices of a deployment
• As a System Admin I would like to be able to manage my remote update parameters in Panel Admin Area so I have more control over deployments
• As a Deployment Manager I want to have an alert icon next to the corresponding device in the deployment list when an alert is launched
• As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc
• As a PO I want a flow diagram describing Panel functionalities
• As a Deployment Manager I want to see the Alerts on each device coming from real data
• When accessing the home page while logged in as an Admin

Even having to deal with long tasks, the team is satisfied with the output. The UX specialist is still preparing designs for some parts of the platform, in nexts sprints we will decide when and how we introduce this designs to the system. Now I am implementing and designing the UI up to my criteria, not putting much effort due to the fact that it may change completely.

3.4. Sprint 3

3.4.1. Sprint plan

In this sprint, as usual lately, the goal is to close the started stories adding some issues. As the pilot is more close to the first release with clients, issues have more priority than functionalities.
• As a System Admin I would like to be able to manage my remote update parameters in Panel Admin Area so I have more control over deployments
• Remove Home link in Navigation Drawer as it is not including any content
• When accessing the home page while logged in as an Admin
• As a PO I want a versioning proposal and release plan so we can organize testing and development
• As a PO I want Web Panel integrated with Azure backend
• As a PO I want Web Panel integrated with AWS backend
• As a Security Manager I want to be able to configure iptables of my devices remotely using Web Panel
• Remote update messages sent at page loading when listing the devices of a deployment
• As a Deployment Manager I want to have an alert icon next to the corresponding device in the deployment list when an alert is launched
As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc. Story

As a PO I want a flow diagram describing Panel functionalities

Reset Password system is not working

As a Company Manager I want to be able to remove old deployments

UX when changing fields in the deployment table is confusing

A property shouldn't be created if there are missing fields

As a Deployment Manager I want to be able to remove properties if I don't need them anymore

Reset Password flow should have the same look and feel as the rest of the site

We need to fix, security vulnerabilities discovered during Panel testing

As a Deployment Manager I want to be able to remove old devices

As a developer I want to have a second Azure connection string which will be used only by devices to send the telemetry in order to grant the proper privileges to the devices

Crash when trying to configure devices being Company or Deployment Manager

As a manager I want to sync devices from Azure that already exist so I can reuse them directly in case that I was using them before.

As a manager I want to receive some error message when I create a device with an used device id so I know why the page is not doing anything

In order to try to be more accurate planning the sprints, not many stories were added to this sprint, so just close the open ones and solve some bugs.

3.4.2. Application design

Logical design of the application

There are not relevant changes in the design, just few and small changes. The type of some fields, and the creation of a couple of them. But I am seeing that the system design has some issues, and that the system needs a major restructuring again to meet the new requirements. It is not going to be implemented in the next sprint, but I am going to start thinking about the solution.

User experience design

Even though I do not have to come up with design decision, as the designs are not prepared yet, I am doing it when I need some UI elements. Up to this point of the development, and because it is not done by the UX expert, the UI is done as fast as possible, thinking always that all the invested hours may go to the trash.
3.4.3. Implementation
Nothing relevant in this sprint.

3.4.4. Sprint review

Many things were done, but none was completed. The QA assurer couldn’t test due to some technical problems. They lost access to the development machine. As a consequence, none of the task was complete because none of them could be reviewed. In the next sprint the tasks will be reviewed and hopefully deployed into the production environment.

3.5. Sprint 4

3.5.1. Sprint plan
The first and most urgent task for the beginning of this sprint was to have the development server up again, and then start with the tasks of the sprint. The task planned were the following:

- As a Security Manager I want to be able to configure iptables of my devices remotely using web Panel
- As an Admin I would like to be able to manage my remote update parameters in Panel Admin Area so I have more control over deployments
- A property shouldn't be created if there are missing fields
- As a Deployment Manager I want to be able to remove properties if I don't need them anymore
- Remove Home link in Navigation Drawer as it is not including any content
- When accessing the home page while logged in as an Admin
- As a Deployment Manager I want to be able to remove old devices
- As a manager I want to receive some error message when I create a device with an used device id so I know why the page is not doing anything
- As a developer I want CD for dev and prod environment so I don't worry anymore
- As a developer I want to rename the estimated column to in "progress/estimated"
- As a PO I want web Panel integrated with Azure backend
- As a PO I want web Panel integrated with AWS backend
- Remote update messages sent at page loading when listing the devices of a deployment
- As a Deployment Manager I want to have an alert icon next to the corresponding device in the deployment list when an alert is launched
- As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc
- As a PO I want a flow diagram describing Panel functionalities for Deployments, Devices and Users
- As a Company Manager I want to be able to remove old deployment
- UX when changing fields in the deployment table is confusing
- Reset Password flow should have the same look and feel as the rest of the site
- We need to fix, security vulnerabilities discovered during Panel testing
- As a Deployment Manager I want to be able to remove old tags
- As a developer I want to have a second Azure connection string which will be used only by devices to send the telemetry in order to grant the proper privileges to the devices
- Crash when trying to configure devices being Deployment Manager
- As a manager I want to list the tags and properties in device details view in order to see which tags and properties a device has
- As a manager I want to sync devices from Azure that already exist so I can reuse them directly in case that I was using them before.
- As a developer I want to create an Azure device in a iotHub when reassign devices so I don't have non sync devices in my deployments
- As a PO I want to investigate and design a Panel Implementation based on containers
- As a PO I want the ability to include tags and associate them with devices
- As system admin I see an error when trying to configure devices
- Some columns should be placed at right in the list of devices per deployment
- As a manager I want to add properties to the devices from device detail view
- As a manager I want to add tags to the devices from device detail view
- As a manager I want to remove properties
The tasks prepared are meant to close the main panel functionality. In the next sprints there will be tasks that involve communication with the devices, but first stabilize and finish the control panel.

3.5.2. Application design

Logical design of the application

The design of the application didn’t change up to this point. Only a couple of fields of the data model changed their types. Nothing relevant to be mentioned.

User experience design

With the complete implementation of tags and properties, all the functionality designed in the last revision will be covered. Even though the styles designed by the UI expert aren’t applied 100% yet, due to some discordination, the mock design used in the application, help to reveal design issues, so the UX expert can come up with better designs.

From now on, it is true that most of the functionality remaining it is not going to be visible for the user, but if it involves any user action, it is very probably that the UI will be affected, and even the data model.
3.5.3. Implementation

Code

As I introduced continuous delivery in the project, I come up with a branching protocol to keep the code organized. So I do not upload things that I don’t want, or that are not tested or not even completely developed yet. In figure 15 there is a schema of how it works:

I defined 4 main branches in our source control system:

- **Master**: It contains only the code that it is going to be released in production. I use tags to identify the versions that I have deployed.
- **Release**: It is a branch between dev and master, where we prepare, if necessary, the last tweaks before deploy the code in production. Generally is just the change log, but it may be anything to make the code production ready.
- **Dev**: This is the development branch, this is where I merge our changes under development branches and where I test the code. The QA runs their tests also over this branch, in addition to production.
- **##-user_story**: I create a new branch per card in the board. I keep the card number and a small description of the task. I develop each feature or fix an issue here and then I merge to the dev branch.

Figure 16. Branch control system.
CI/CD
The delivery process has been automated using Docker containers. As Bitbucket didn't offer a registry for Docker images, I moved the project to Gitlab. Now after each commit in every branch, a new container is generated, tagging it according to its branch. Afterwards, the image is pushed to the images repository. The workflow of the delivery is described in figure 16:

![Diagram](image)

Figure 17. Continuous Delivery workflow.

After each commit, in Gitlab I have created a pipeline that takes the code and creates a new Docker image. Depending on the branch it uses a tag or another. When it is the master branch it uses special parameters for production, such as remove the debug mode, remove extra logs or increase the number of threads in the application server.

Once the images are created, they are hosted in the image repository provided by Gitlab ready to be pulled from any of our machines (Development, Integration or Production).

Testing
The life of the QA tester was more easy, bringing the last changes to the integration machine. There is a dedicated server running the development brach, so all the features can be tested straight away after they are developed.
Deployment

A script in production and development servers pulls the Docker images generated and runs them. Each server pulls its image, stops the running container, deletes the current image, downloads the new image and runs it. In figure 17 it is detailed how the images are pulled in each environment:

![Diagram of Docker image pull per environment]

Only if there are changes in the data model, a manual action to import these changes is need. The data from the database is persistent in each server, even though I update the data model, or the database version, or whatever, the data of the database it is going to be outdated, needs to be fixed.
3.5.4. Sprint review

The deployments in containers and its automation, took most of the time of this sprint, much more than expected, but the benefits are worthy. Even though several tasks were finalized, many of them came from the previous sprint.

The following tasks were completed this sprint:

- As a Security Manager I want to be able to configure iptables of my devices remotely using web Panel
- As a web Admin I would like to be able to manage my remote update parameters in Panel Admin Area so I have more control over deployments
- A property shouldn't be created if there are missing fields
- As a Deployment Manager I want to be able to remove properties if I don't need them anymore
- Remove Home link in Navigation Drawer as it is not including any content
- When accessing the home page while logged in as an Admin
- As a Deployment Manager I want to be able to remove old devices
- As a manager I want to receive some error message when I create a device with an used device id so I know why the page is not doing anything
- As a developer I want CD for dev and prod environment so I don't worry anymore
- As a developer I want to rename the estimated column to in "progress/estimated"
- As a PO I want web Panel integrated with Azure backend
- As a Company Manager I want to be able to remove old deployments
- UX when changing fields in the deployment table is confusing
- Reset Password flow should have the same look and feel as the rest of the site
- As a PO I want to investigate and design a Panel Implementation based on containers
- As a PO I want the ability to include tags and associate them with devices
- As a system admin I see an error when trying to configure devices
- Some columns should be placed at right in the list of devices per deployment
- As a manager I want to list the tags and properties in device details view in order to see which tags and properties a device has
- AS a Deployment Manager I want to be able to remove old tags
- Remote update messages sent at page loading when listing the devices of a deployment
- As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc
- As a developer I want to have a second Azure connection string which will be used only by devices to send the telemetry in order to grant the proper privileges to the devices
- As a manager I want to sync devices from Azure that already exist so I can reuse them directly in case that I was using them before.
- As a developer I want to create an Azure device in a iothub when reassign devices so I don't have non sync devices in my deployments
- Crash when trying to configure devices being Deployment Manager
These tasks could not be done:

- As a PO I want web Panel integrated with AWS backend
- As a Deployment Manager I want to have an alert icon next to the corresponding device in the deployment list when an alert is launched
- As a PO I want a flow diagram describing Panel functionalities for Deployments, Devices and Users
- We need to fix, security vulnerabilities discovered during Panel testing
- As a manager I want to add properties to the devices from device detail view
- As a manager I want to add tags to the devices from device detail view
- As a manager I want to remove properties

Not all the tasks could be done due to the overrun integrating containers in the system. Issues and key tasks will be moved to the next sprints, the other tasks will go to the backlog to be done in further sprints.

3.6. Sprint 5

3.6.1. Sprint plan

The goal of this sprint was pretty clear, finish the unfinished stories from the previous sprint and integrate Build Images and Remote Updates. The build system must call the Jenkins API which will build the images, and receive a message when it is done. The remote updates were working before, but all the values were "hardcoded" to the request. Now it is time to make it work properly. These are the tasks prepared for this sprint:

- As a Manager I want to send remote updates to my devices
- As a PO I want web Panel integrated with Azure backend
- Remote update messages sent at page loading when listing the devices of a deployment
- As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc
- As a PO I want a flow diagram describing Panel functionalities for Deployments, Devices and Users
- As a Company Manager I want to be able to remove old deployments
- UX when changing fields in the deployment table is confusing
- Reset Password flow should have the same look and feel as the rest of the site
- We need to fix, security vulnerabilities discovered during Panel testing
- As a developer I want to have a second Azure connection string which will be used only by devices to send the telemetry in order to grant the proper privileges to the devices
- As a manager I want to list the tags and properties in device details view in order to see which tags and properties a device has
- As a PO I want to investigate and design a Panel Implementation based on containers
● As a system admin I see an error when trying to configure devices
● Some columns should be placed at right in the list of devices per deployment
● As a system admin I want styles back to admin panel of django
● As a developer I want to create an Integration environment
● As a PO I want Panel user to be warned just before applying a new configuration to make her to confirm if she wants the config to be applied
● AS a Deployment Manager I want to be able to remove old tags
● Crash when trying to configure devices being Deployment Manager
● As a manager I want to sync devices from Azure that already exist so I can reuse them directly in case that I was using them before.
● As a developer I want to create an Azure device in a iotHub when reassign devices so I don't have non sync devices in my deployments
● As a PO I want the ability to include tags and associate them with devices
● As a manager I want to add properties to the devices from device detail view
● As a manager I want to add tags to the devices from device detail view
● As a manager I want to remove properties
● As a Security Manager I need the ability to apply full remote updates from web Panel to one device
● As a Security Manager I need the ability to apply full remot updates from web Panel to several devices at the same time
● As a Security Manager I want web Panel to ask me for confirmation before applying any remote update
● As a PO I need Panel to check device version in order to determine if the remote update has been applied
● As a Dev I need Jenkins to inform Panel about when and where the images have been created/stored
● As a developer I want to securize all function endpoints for obvious reasons
● As a developer I want to clean up all the Azure connection stuff from UI and model
● Reset Password system is not working
● As a developer I want to inform the device os version available in order to inform the user which version of os is going to build
3.6.2. Application design

System architecture

The new Jenkins server was added to the system diagram. It is the server to build new OS images, it is another EC2 instance with several CPUs. It runs a Jenkins server and has exposed the Jenkins API. The control panel communicates with this API. The new architectural component schema is described in figure 18:

![Component architectural schema](image)

Figure 19. Component architectural schema.
Communication design

A new communication is introduced in this sprint, is the communication with the Build OS images server. This message will be used to send a compilation request to the build server. First I need to generate a token to operate with the Jenkins API and set it as a header in each request. Once the job run, it will send a message to the control panel whether with an ok or with an error message. The sequence diagram to perform the request is detailed in figure 19:

![Sequence Diagram](image)

**Figure 20. Build request sequence diagram.**
Logical design of the application

In this sprint I added a new entity Build. This entity will store information about the builds of each deployment. A deployment will generate an image for its devices, which will be sent to them afterwards in order to update them. In this entity, it is stored basic information about the image such as url location to download it, size, timestamp and the version number. There is a token also stored, it is used to identify the messages in a secure way coming from the build server. In figure 20 there is the new data model created:

![System data model diagram](image)

Figure 21. System data model.

There are also minor changes in the diagram regarding the data types. At some point, some of them were meant to have a type that afterwards turned out that it was more suitable in another type.

User experience design

Just a small section was added to the deployment view with the build images button and information. Not substantial changes so far.
3.6.3. Implementation

Testing
I had to set up an integration server for testing purposes. Until now, there only was the production server and a development server. The development server was in our office behind the firewall, so it was impossible to expose it to the internet. Now I needed to communications going between our server and the build server, so I need a machine exposed to the internet.

3.6.4. Sprint review
In this sprint many issues were completed. It is true that many of them were already started from the previous sprint. These are the tasks completed:

- As a Manager I want to send remote updates to my devices
- As a PO I want web Panel integrated with Azure backend
- Remote update messages sent at page loading when listing the devices of a deployment
- As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc
- As a PO I want a flow diagram describing Panel functionalities for Deployments, Devices and USers
- As a Company Manager I want to be able to remove old deployments
- UX when changing fields in the deployment table is confusing
- Reset Password flow should have the same look and feel as the rest of the site
- We need to fix, security vulnerabilities discovered during Panel testing
- As a developer I want to have a second Azure connection string which will be used only by devices to send the telemetry in order to grant the proper privileges to the devices
- As a manager I want to list the tags and properties in device details view in order to see which tags and properties a device has
- As a PO I want to investigate and design a Panel Implementation based on containers
- As system admin I see an error when trying to configure devices
- Some columns should be placed at right in the list of devices per deployment
- As system admin I want styles back to admin panel of django
- Rearrange Board and relate resolutions with issue status
- As a developer I want to create an Integration environment
- As a developer I want to clean up all the Azure connection stuff from UI and model
- Reset Password system is not working
- As a manager I want to sync devices from Azure that already exist so I can reuse them directly in case that I was using them before.
- As a developer I want to create an Azure device in a iothub when reassign devices so I don't have non sync devices in my deployments
- As a PO I want the ability to include tags and associate them with devices
• As a manager I want to add properties to the devices from device detail view
• As a manager I want to add tags to the devices from device detail view
• As a manager I want to remove properties
• AS a Deployment Manager I want to be able to remove old tags
• Crash when trying to configure devices being Deployment Manager

These tasks could not be delivered. Many were done, but could not be tested by the QA assurer by the end of the sprint:
• As a PO I want Panel user to be warned just before applying a new configuration to make her to confirm if she wants the config to be applied
• As a Security Manager I need the ability to apply full remote updates from web Panel to one device
• As a Security Manager I need the ability to apply full remote updates from web Panel to several devices at the same time
• As a Security Manager I want web Panel to ask me for confirmation before applying any remote update
• As a PO I need Panel to check device version in order to determine if the remote update has been applied
• As a Dev I need Jenkins to inform Panel about when and where the images have been created/stored
• As a developer I want to securize all function endpoints for obvious reasons
• As a developer I want to inform the device os version available in order to inform the user which version of os is going to build

In this sprint, I needed to set up a new integration server so the devices and the build server could communicate with the web panel server. Our development server used by the QA assurer is inside our intranet and it cannot be reached from outside. The team rented a virtual machine server in a minor cloud provider looking for low price instead of high performance.
3.7.  Sprint 6

3.7.1.  Sprint plan

In this sprint there is a mixture between new functionalities and issues found in the system. The goal is to fix the issues first and bring the new functionalities if possible. I do that in this way, just in case a client sees the control panel.

- Delete device button not working properly
- As a developer I want to configure Network Interfaces remotely
- As a security manager I need the ability to apply updates from control panel to several devices at the same time
- As a Security Manager I want control panel to ask me for confirmation before applying any update
- As a Security Manager I need the ability to apply full Updates from control panel to just one device
- As a PO I need Panel to check device version in order to determine if the update has been applied
- As a Dev I need Jenkins Build api to inform Control Panel about when and where the images have been created/stored
- As a developer I want to inform the OS version available in order to inform the user which version of OS is going to build
- Not possible to assign devices to an specific deployments (error 500)
- As a manager I want to select a device type per deployment
- As a developer I want to set up an integration environment machine exposed on the internet
- As a Company Manager I need the ability to generate images from control Panel
- As a Security Manager I want Control Panel to inform me about the progress of the update
- As a PO I want Privacy Policy (GDPR) to be linked in the home page
- the button to erase properties and tags of a device does not work
- As a manager I want to edit tags and properties from device detail view so I can make changes easy and quickly
- Device unassign button from a deployment does not work
- Typo in the error message when creating a device with a used Device ID
- There an issue displaying the GPS coords, and as a consequence the position can not be showed in a map
- As a PO I want to apply new styles in the login page
- As a developer and as a Security Manager, I want to have a development-integration server at the Office so we can deploy releases in a safe environment, but also test them for security breaches from the Internet
- As a Deployment Manager I want a mechanism to assign the built image to a device or group of devices, defining beforehand the type of device.
- As a Security Manager, I want the Company Manager to be able to manage Deployment Manager permissions.
- As a Security Manager I want the admin section of the panel to be accessed by a different url from default (..../admin)

3.7.2. Application design

Communication design

There is a new communication to configure the devices remotely, currently just for the network interfaces. This message will be sent every time that a user enters in the edit configuration option of the panel. First it will request the configuration to the device and verify its integrity against the configuration saved in the control panel database. If it has been compromised, it will alert the user. The second part will happen, if the user modifies the configuration and wants to apply it to the device, the configuration to be applied will be send to the device. The communication design is available in the sequence diagram of figure 21:
3.7.3. Implementation
Nothing changed from the previous sprint.

3.7.4. Sprint review
This sprint was one of the most productive so far. Almost all the tasks on the scope of this sprint were finished, and many of them were quite complex. Only these two were not completed:

- Delete device button not working properly
- As a developer I want to configure Network Interfaces remotely

The remote configuration of the network interfaces task was started, but not finished. This card should have been splitted in few cards because it is on the border of being an epic task. The list of tasks done is the following:

- As a security manager I need the ability to apply updates from control panel to several devices at the same time
- As a Security Manager I want Control Panel to ask me for confirmation before applying any update
- As a Security Manager I need the ability to apply full Updates from Control Panel to just one device
- As a PO I need Panel to check device version in order to determine if the update has been applied
- As a Dev I need Jenkins Build api to inform Control Panel about when and where the images have been created/stored
- As a developer I want to inform the OS version available in order to inform the user which version of OS is going to build
- Not possible to assign devices to an specific deployments (error 500)
- As a manager I want to select a device type per deployment
- As a developer I want to set up an integration environment machine exposed on the internet
- As a Company Manager I need the ability to generate images from control Panel
- As a Security Manager I want Control Panel to inform me about the progress of the update
- As a PO I want Privacy Policy (GDPR) to be linked in the home page
- the button to erase properties and tags of a device does not work
- As a manager I want to edit tags and properties from device detail view so I can make changes easy and quickly
- Device unassign button from a deployment does not work
- Typo in the error message when creating a device with a used Device ID
- There an issue displaying the GPS coords, and as a consequence the position can not be showed in a map
- As a PO I want to apply new styles in the login page
- As a developer and as a Security Manager, I want to have a development-integration server at the Office so we can deploy releases in a safe environment, but also test them for security breaches from the Internet
- As a Deployment Manager I want a mechanism to assign the built image to a device or group of devices, defining beforehand the type of device.
- As a Security Manager, I want the Company Manager to be able to manage Deployment Manager permissions
- As a Security Manager I want the admin section of the panel to be accessed by a different url from default (..../admin)

The pace is going good up to this point, but more key functionalities for the business should be introduced soon.

### 3.8. Sprint 7

#### 3.8.1. Sprint plan

The goal of this sprint is to finish the remote network configuration, this is part of a key functionality of the system, together with the remote updates of the devices implemented in last sprint, business and marketing departments want to start to expose the control panel to the clients, so they can start having feedback. To do so, after developing the functionality, the other tasks are to fix issues. There are some errors that appear from time to time, I need to purge all of them. I have them more or less identified, most of them produce due to inconsistency between databases across environments, but anyway I need to track them down and fix them.

- Delete device button not working properly
- As a developer I want to configure Network Interfaces remotely
- Improve the user experience of the new functionalities
- Eliminate database errors when moving environments due to data model inconsistencies.
3.8.2. Application design

Logical design of the application

I introduced to our data model the entity `config` to save all the device configurations. I will store all configurations as a historical, so a user can reload them. The new data model is shown in figure 22:

![Data model diagram](image)

Figure 23. System data model.
User experience design

After the new functionality is implemented, this is the complete screen flow design, there is just one screen added for device configuration. The updated navigation diagram is displayed in figure 23:

3.8.3. Implementation

Nothing changed from last sprint. At this point, our implementation process including coding, continuous integration and deliver, testing and deployment, is quite mature already. It’s been a while since I changed anything here, and unless I need to add something, not much is going to change until the end of the project.

3.8.4. Sprint review

This sprint was great, it was a coincidence that a “consolidation” sprint, as we called it, was at the same time that the end of this thesis. The business side was quite happy with the results of the product, not only just regarding the control panel, but also the operating system.

Even though the integration with the operating system was not easy at the beginning, we (me and the os teams) learnt how to communicate soon, and before the end of the tasks I developed an internal cross-team communication “framework”. This framework consist mainly in a Google Drive Sheet with all the “requirements” of the communications, all the possible actions, all the possible reactions, all the accepted parameters, etc everybody could access and edit this document. So whenever somebody finds an issue, or changes anything, it uploads immediately there for all. The second part consists in a Glip group with all the developers involved, and when somebody changes anything in the document, it must notify there.
4. Project review

4.1. Project status

After describing what has been done in each sprint, in this chapter it is going to summarize which stories has been done per initial objectives:

User management:
- As an Admin I need the ability to list users
- As an Admin I need the ability to securely login
- As a Manager I need the ability to securely log in to the Web Panel
- As a User (of any type) I want an option to change my password
- As a user (of any type) I want to have a section where I can edit my user information
- As a PO I want to apply new styles in the login page

Device operations:
- As a User I want to be able to list devices
- As a User I want to be able to see device details
- As an Admin I need the ability to create new Companies to group different deployments
- As a Manager I need to create new Deployments to manage grouped devices
- As a Manager I need the ability to create devices
- As a Manager I need to be able to assign devices to specific deployments so we can group them
- As a User I want device details site to refresh automatically when information change
- As deployment manager I want to add properties to the devices in order to identify specific attributes of each device
- As a Real Client Project Manager I want vehicles plate indicated in Web Panel so I can identify them quickly
- As a Real Client Project Manager I want an extra column in deployment visualization indicating UICCID of the SIM
- As a Deployment Manager I want to have historical plots of device health data: temperature, CPU etc
- As a Deployment Manager I want to be able to remove properties if I don't need them anymore
- As a Deployment Manager I want to be able to remove old devices
- As a manager I want to receive some error message when I create a device with an used device id so I know why the page is not doing anything
- As a Company Manager I want to be able to remove old deployments
- As a PO I want the ability to include tags and associate them with devices
● As a manager I want to list the tags and properties in device details view in order to see which tags and properties a device has
● As a Deployment Manager I want to be able to remove old tags
● As a manager I want to sync devices from Azure that already exist so I can reuse them directly in case that I was using them before.
● As a developer I want to create an Azure device in a iothub when reassign devices so I don't have non sync devices in my deployments
● As a manager I want to add properties to the devices from device detail view
● As a manager I want to add tags to the devices from device detail view
● As a manager I want to remove properties
● As a manager I want to select a device type per deployment
● As a manager I want to edit tags and properties from device detail view so I can make changes easy and quickly

Build images:
● As a Dev I need Jenkins to inform Panel about when and where the images have been created/stored
● As a developer I want to inform the device os version available in order to inform the user which version of os is going to build
● As a Company Manager I need the ability to generate images from control Panel
● As a Deployment Manager I want a mechanism to assign the built image to a device or group of devices, defining beforehand the type of device.

Remote Updates:
● As a Manager I want to send remote updates to my devices
● As a System Admin I would like to be able to manage my remote update parameters in Panel Admin Area so I have more control over deployments
● As a Security Manager I need the ability to apply full remote updates from web Panel to one device
● As a Security Manager I want web Panel to ask me for confirmation before applying any remote update
● As a PO I need Panel to check device version in order to determine if the remote update has been applied
● As a security manager I need the ability to apply updates from control panel to several devices at the same time
● As a Security Manager I need the ability to apply full remote updates from web Panel to several devices at the same time
● As a Security Manager I want Control Panel to inform me about the progress of the update
Remote Configuration:
- As a Security Manager I want to be able to configure iptables of my devices remotely using web Panel
- As a developer I want to configure Network Interfaces remotely
- As a PO I want Panel user to be warned just before applying a new configuration to make her to confirm if she wants the config to be applied

Manage company managers:
- As an Admin I need the ability to create Manager users
- As a Company Manager I want to create Deployment Managers in my company
- As a Manager I need to be able to assign Users to an specific deployment
- As a Manager I need to be able to assign Company Managers to an specific deployment
- As a Security Manager, I want the Company Manager to be able to manage Deployment Manager permissions

Alert System (for clients):
- As a Deployment Manager I want to have an alert icon next to the corresponding device in the deployment list when an alert is launched
- As a Deployment Manager I want to see the Alerts on each device coming from real data

Logging, monitoring, reporting and internal alarm system:
None

Backup and recovery system:
None

Telemetry data management:
None

Full support any service in any cloud provider:
- As a PO I want web Panel integrated with AWS backend
- As a PO I want Web Panel integrated with Azure backend
- As a Developer I want to investigate and design a Panel Implementation based on containers so deployments become easier
- As a developer I want CD for dev and prod environment so I don't worry *anymore about deployments
API for clients:
None

Development Issues:
- As a developer I want to review all relationships in the backend module to optimise queries and user experience
- As a developer I want to review the rules, permissions and ownership validations in order to enhance the security of the system
- As a developer and as a Security Manager, I want to have a development-integration server in the Office so we can deploy releases in a safe environment, but also test them for security breaches from the Internet
- As a developer I want to rename the estimated column to in "progress/estimated"
- As a developer I want to have a second Azure connection string which will be used only by devices to send the telemetry in order to grant the proper privileges to the devices
- We need to fix, security vulnerabilities discovered during Panel testing
- As a developer I want to create an Integration environment so I can run tests there
- As a developer I want to clean up all the Azure connection stuff from UI and data model
- As a developer I want to securize all function endpoints for obvious reasons
- As a developer I want to set up an integration environment machine exposed to the internet

Bugs:
- Remote update messages sent at page loading when listing the devices of a deployment
- When a just created user try to create the first deployment, Panel report an error
- When accessing the home page while logged in as an Admin
- A property shouldn't be created if there are missing fields
- UX when changing fields in the deployment table is confusing
- Reset Password flow should have the same look and feel as the rest of the site
- As a system admin I see an error when trying to configure devices
- Some columns should be placed at right in the list of devices per deployment
- Crash when trying to configure devices being Deployment Manager
- Eliminate database errors when moving environments due to data model inconsistencies.
- Improve the user experience of the new functionality (Build updates)
- Delete device button not working properly
- As system admin I want styles back to admin panel of django Reset Password system is not working
- Not possible to assign devices to an specific deployments (error 500)
- The button to erase properties and tags of a device does not work
- Typo in the error message when creating a device with a used Device ID
- There an issue displaying the GPS coords, and as a consequence the position can not be showed in a map

Other issues:
- Abuse Stories Panel
- As a PO I want a flow diagram describing Panel functionalities
- As a Real Client CTO I want the web Panel to have a change log section indicating the changes in the latest version released
- As a PO I want to access the relevant sections using shortcuts in the left side menu
- As a PO I want to remove Home link in Navigation Drawer as it is not including any content
- As a PO I want a flow diagram describing Panel functionalities for Deployments, Devices and Users
- As a PO I need to Rearrange Board and relate resolutions with issue status
- As a PO I want Privacy Policy (GDPR) to be linked in the home page
- As a Security Manager I want the admin section of the panel to be accessed by a different url from default (..../admin)

Each one of these user stories has a different level of effort and sub-functionalities. The goals with more user stories are not necessarily more complex nor extend than any other. As it can be appreciated, there are some goals that do not have any user story, it means that no action has been done regarding that objective.

4.2. Software process analysis

In this section of the document I am going the analyze how went the software creation process, the things I liked and the things I did not.

4.2.1. Things that were wrong

The first thing I have to mention, is that I find that is not the best way how user stories are created. I feel that are not clear enough, and above all they are not atomic enough. There are issues with tasks of 5 minutes, and some others that need over 2 weeks of work. Which turns out in a story that is not useful at all, it does not describe properly the task. It is true that the product manager writes long descriptions inside, and also the “definition of done”, which contains the elements to consider a task finished, and defines how it will be tested. But obviously in big tasks this way of doing things it just doesn’t work.

This procedure works now just because we are a small company, with just one developer (doing the control panel). And also because we have meetings very often, and we talk a lot about the work to be done. But if just few developers more join the company, this way working wouldn’t be
possible anymore, because logistically, it won’t be possible to meet all the stakeholders always that often.

I am not happy either with the fact that I didn’t introduced unit testing to the development process. I have to mention that personally I consider unit testing a must, but not mandatory. In my previous work experience, unit testing only arouse a few percentage of the errors, and only during integration tests or user acceptance tests all the other problems appeared. If I would have more resources (people) I would have done unit testing for sure, but as long as I had to handle the priorities, I skipped unit testing and only did end to end tests (integration).

Another thing that I don’t like at all, is that I didn’t automate these end to end tests. After implementing each user story, and if it was the “definition of done” of the task defined, I prepared a tests list to make sure that the functionality was working properly. I tried to guarantee, at least, branch coverage of the functionalities. But after all, I am a human, and doing manual and repetitive tasks is not one of the things that I am good at. That is why I need to automate this part. It would be much better if a program passes all the tests for me. Also because in that way, the program could pass tests to the whole system, not just the feature that is being developed.

I would like to mention too that at some points during the development of the project, I felt that I was losing the real focus of the project goals, and I my efforts changed directions towards issues not much relevant for the business. Whenever I had this feeling, I consulted to the product manager and the business guys to get instant feedback. Sometimes it was the correct direction, but some others, I had to change the route. This is due to the constant changes of the requirements, and the fact that after all, we are all humans and took wrong decisions to solve a problem. But I am glad that thanks to agile methodologies, all these problems were identified early and we were able to correct them soon.

The last thing I would like to mention, is that I am not enjoying working alone. It is true that I can do whatever I want, whenever I want, but this freedom does not compensate all the negative effects. Just being one person, to take care of all the process is a bit overwhelming. And it is not because I don’t know how to do the things, or I am afraid to fail, is because I am not productive if I have to deal with hundreds of different things every day. I have lost complete days just going from a meeting to another. Then, the development is not as fast as I would like, in just a day, fix an issue in the production environment, implement a new feature in development, support the QA running tests in integration,... just opening and closing tools, change environments and change the mind state and so on, the day is gone, and the development peace is below the one I would like most of these days.

In addition to this, I am convinced that regarding development teams, 1 + 1 is not just 2, is 2+. And not only because the load can be distributed in a better way, also because you have somebody to talk to and spread the thoughts and problems. Many times happened to me in the past, that while I was explaining to somebody a problem I had, I just realize which was the
solution, and this happened to me very often. Also when making decisions, is always good to have different points of view, even when they are wrong. Working alone, also makes impossible to use techniques or practices such as pair programming or doing code reviews.

4.2.2. Things that were good

It has been 4 months now since I started this project. I had used python few times before and just to write small scripts, not ever a big system. But regardless this fact, honestly, I thought I would have more parts done, but also with less level of detail. In the academic field, where I came from, I usually tend to demonstrate the concept rather than the detail, so the most of the times, I just skipped that part. But in the real world, the detail is as much important as the concept and as a consequence, the developments are not as fast as I thought at the beginning. In spite of the inconveniences found, I think I did a good job in the overall. When I started, there were a couple or three basic html files, no styles, no functionality, no nothing... and now there is a fully working system, it still lacks some functionalities, but at the end is a valid product.

Another observation I have is how all this moved from a local machine to the cloud. At the very beginning I had the software just running locally in our machines, but soon I needed to make it available for the clients. Since the first moment, and despite there was a datacenter inside our office, I had clear that the production server must be “somewhere” in the cloud. After a couple of releases I discovered also that I had to improve our release process, and I introduced Docker containers to it, also the automatic generation of the images in the repository code server. After a while, I discovered that I needed a dev and integration environments, and thanks to the current deployment process, it was super easy to do it. In less than 30 min for each environment, I had the new servers up and running.

Through this journey, the product, the requirements and even the people changed quite a lot, starting by myself. The product and the requirements changed for business reasons, but the people and the interactions changed a lot too. We all rely in the communication as the main tool. Since the beginning, the communication was not bad, but now before even doing an step, we try to exchange points of views and try to see how we would solve the problem together. We didn’t start to do that because it is “cool”, it is just because it works.

To make things really go better, there was a barrier that we all had to cross: the confidence. After all, software development is just another human activity that needs social relationships. It is just like any other social activity but within a company frame. Just like the company, we all are new there, we didn’t know each other, and there was a huge difference when we were knowing more each other and start getting along. It was curious to see the impact to the meetings or communications. The directors of the company were aware of this fact, and they prepare events quite often outside the office to improve these relationships.
5. Personal thoughts, conclusions and future work

5.1. Personal thoughts

One of the things that usually companies tend to ignore is the integration of the new members. Many companies just prepare a landing for the new members to be productive as soon as possible for the company, but many times they just forget about the human behind. I started being productive since the very first day, but also since the very first day I felt that I was part of something. The horizontality of the company regarding communications surprised me, since the beginning, the directors of the company shared their vision of the business, explained the route they want to take and also the reality of the current day. Good news and bad news are communicated to all the members of the company. The motto is “we win and lose together”, so as a member of a team, I know in every moment how the company is doing, nothing is secret. This great atmosphere soon creates quickly a link between the new members and the rest of the company, and immediately start the feeling of “I need to contribute in this”.

Likewise, since the very beginning, I was empowered to make important decisions regarding my part of the product. I had the power to change anything that was done before without being disputed, just explaining the motives was enough. This also makes you feel the responsibility and the weight of the duty.

It also brought me the opportunity to see agile methodologies in pure state. The definition of the product was (and still a bit) not defined completely. We see week after week our progress and after we discuss the direction that we want to take together with the people form business. This was quite shocking to me. I worked for almost 5 years for a big bank using CMMI, which is exactly the opposite. Everything was perfectly defined before anything started. The requirements analysis document was a contract between companies, the software must do exactly what was specified there, nothing else, nothing less. You know perfectly what you had to do until the moment you finished the project.

Here the requirements change weekly if not daily, and it’s something “normal”. It shocked me a bit at the beginning, but soon realized that this is the way it works and after a while I got use to it. Nothing is set in stone, everything can change, even the most structural things. This fact required an additional skill that was dealing with the uncertainty, and not just in business perspective, also in the technological side. Together with the business requirements, it also changes the technical requirements, and several times I don’t have a clue how even to start to implement things.

Another interesting skill learnt is that as a software architect, due to the constant changes of requirements, I am forced to do better designs of the system. Whenever I design a new functionality, since the beginning, I already think about how I can change and how it can be
extended. At the beginning, my short sight designs, made me throw away some modules and start over again when the requirements changed. After a while, I just changed the way I think when designing.

To sum up, this experience was a huge change in the way I use to work coming from a more traditional way of doing things. But at the end I think that was not real at all, the world changes dynamically and we have to adapt to it every day, why we could not do the same with software? I really learnt how to work in an agile environment, learning also that the ‘unknown’ is just another thing to take into account, not something to be afraid of.
5.2. Conclusions

This is the list of the project objectives defined from the beginning, that the control panel has to implement at some point. Here is reviewed the state of them at this moment:

- **User management:** DONE
  Almost all the issues related to this point were solved using the tools provided by the Django framework. No major problems during the development, beyond learning how to use the framework.

- **Device operations:** DONE
  Despite new actions may be required in the future, all the functionality defined up to this point has been implemented. All these tasks, may not be easy to implement, but all of them are precisely defined in terms of functionality.

- **Build images:** DONE
  The complexity of this tasks was mainly at servers configuration level. The security of the connections and files had to be guaranteed, and this is always a complex task. Regarding the code was quite easy due to the fact that the Jenkins API consumed was well documented.

- **Remote Updates:** DONE
  Although this group of tasks may seem complex, the major complexity was in the devices operating system. The control panel just had to send a message and serve the updates generated in the previous step to the devices.

- **Remote Configuration:** DONE (FOR NOW)
  This task is one of these that may not ever end. So far only the configuration of IP tables and network interfaces has been implemented. I know for sure that many other configurations will be implemented, but until this moment, the implementation meets the current requirements defined. This task is by far the more complex due to the nature of the problem. The solutions are designed ad-hoc to each configuration to be done.

- **Manage company managers:** DONE
  Like the operations for devices, these are a set of functionalities to manage users within a client organization. Just as the others, these requirements were always quite concrete.

- **Alert System (for clients):** PARTIALLY DONE
  I say partially done because the communication channel between devices and control panel has been implemented, but not the alerts themselves. I have implemented the tool, but what should the system alert for is not defined yet. Whenever I know exactly what should I warn the user for, this task has not major issues to be implemented.
• Logging, monitoring, reporting and internal alarm system: **NOT DONE**
  Not started yet.

• Backup and recovery system: **NOT DONE**
  Pending to be done.

• Telemetry data management: **NOT DONE**
  Even though no actions were taken, the problem has been analyzed. I know exactly which new tables need to be created, how to distribute the data among them, how to access the information efficiently,… But due to the current volume of data that the system is handling, this has not been a priority yet, so it has not been implemented.

• Full support any service in any cloud provider: **PARTIALLY DONE**
  The task is marked as partially done because the control panel has been encapsulated in Docker container images to guarantee the compatibility with any location. So, in theory it should not arise any issue, but the reality is that this has not been tested yet. And even beyond that, the major task here is to learn how to use the different clouds. Despite all the providers offer almost the same products, each one of them has their own branding and a different way of using the tools. Again, so far was not a major priority, so it was not done.

• API for clients: **NOT DONE**
  Nothing done here yet.

Despite there are tasks not finished and many not even not even started, I have to remember that the project is still going. I aimed to be finished by the end of august, which I am pretty sure that will not be a problem to finish it on time.

The first thing I have to say that we all, as a company, are still learning how to do the things. The software creation process is still being improved. It is true that at this point it has certain level of maturity, but still far from being perfect. There are some areas that need to be improved.

Starting at an organizational level, the process of task definition was not correct. Despite it is working now, it is not valid. If the company grows just a bit, it is going to be impossible to manage the work properly. Furthermore, during the development of the project, I felt at different moments that I was not developing the correct features. This is a clear symptom that the task/user story definition process and its prioritization needs to be revised.

Even though there may be some problems at organizational level, I have to say that the way we are using agile, works quite good. After each iteration it is easy to identify the problems arised and react to them. Due to the early stage of the business that we are, these feedback loops are something critical to grant us that we are going in the correct direction.
Another thing that could have been done in a better way is the testing. Unit testing is not present, only end to end tests were done, but they were done manually. There is a huge necessity, as the project grows, to bring automated tests to the system. Manual tests are not enough to guarantee that the system is fault free. It would be much more professional to run all the tests for the whole system after any change, that would be the only way we can assure that the change did not break anything and that the system works perfectly.

The context and the nature of the problem added different constraints to the solution proposed. First of all, the system has to have the quality enough to convince the users to use the product, and secondly, as the product is not completely defined yet, it must be adapted quick to the new requirements. The first point made me dedicate a big percentage of the effort in polish all the details, specially regarding the user experience, because it is what the users see. But also I had to put big efforts respecting the security, and this is more “invisible” for the user. The second point, in the other hand, forced me to prepare better designs of the product, in order to be easily adapted to the requirement changes. In the same way, the deployment process had to be fast and reliable, so the only solution possible was the automation of the deployment process.

Working alone in this project, made me realize of the huge amount of different tasks that must be performed during the whole lifecycle, and how difficult is to handle all these tasks with just two hands. The lack of personal in the project, lead to a fall of the productivity. I spent days completely just going from a meeting to another, but also performing tasks that are so different. Just the time needed to open and close tools, change environments, read and write documentation or change the focus and the mind state, consumes a huge time of the day.

The creation of a good team is crucial for the future of any project of software development. And I do not refer as a good team a set of good individuals, I mean a good team as a whole. The skills and abilities of the individuals will just improve the results, but if the team does not get along, will end up in a total failure. A good team is a team with an excellent ability to communicate, and also to rely in each other and being confident enough to raise hands when something is not done properly. Developing a software project is just another social activity, obviously with a big difference, as it is being inside a company. But all the interpersonal relationships are exactly the same as any other activity.

I have seen how the communication and the interactions between all the members of the company have improved progressively as we all were knowing more each other. It was curious to see how this fact directly affected on how productive the meetings and the communications became, and as a consequence, the overall productivity of the company also improved.
5.3. Future Work

The most obvious pending task is to finish the application. There are still some epics pending to be implemented:

- Alert System for clients
- Log, Monitoring, reporting and internal alarm system
- Backup and recovery System
- Full support any service in any cloud provider
- Expose API for clients

Some are not much complicated nor long to be implemented, such as backup and recovery system, but I have to do it at some point. Some others may have some more technical difficulties, like expose an api for the clients. Even though I am using a framework, this is always a complicated task to do, specially regarding security considerations.

Another point I would like to introduce is the automated tests, so that after each commit, I can automatically make sure that the code is going to work, or get a report in case of failure. Currently I am doing manual evaluation running the test suite prepared manually and analyzing the results by myself.

After I finish the project, with some time on my pocket, I would like to decompose the project in small projects and extract them as independent modules, introducing microservices to the system. I am convinced that the device operations, the telemetry reports, the building images of the OS or the user management could work independently as separate modules in different servers. The demand of them and the resources needed are quite different between them, so they may need to scale up at different paces.

In order to improve the deployment infrastructure, and as long as containers are already used in the project, I would like to use some tool for container management, such as kubernetes. Manage our servers, using containers in a professional and, above all, automatic way. Once the costs of the cloud reach certain threshold, intelligent infrastructure management can reduce, in a huge percentage, the cloud bill at the end of the month.
6. Bibliography

In order to write this thesis only online resources were used. Here is the complete list of all the resources used:


3 - "Python (programming language) - Wikipedia", https://en.wikipedia.org/wiki/Python_(programming_language), accessed May 29th 2018


29 - “Introduction to pipelines and jobs | GitLab”, https://docs.gitlab.com/ee/ci/pipelines.html , accessed May 30th 2018
30 - “GitLab Container Registry | GitLab”, https://about.gitlab.com/2016/05/23/gitlab-container-registry , accessed May 30th 2018
31 - “Agile software development - Wikipedia”,

32 - “Scrum (software development) - Wikipedia”,


Annex I - Technology used

In this section, it is described/explained all the technology, concepts and products used. Every time any of this terms is mentioned in any of the chapters, the reader can come here and check what is it about. It is not going to explain how to use the element or how I have used, it is just going to describe what is it for. All the descriptions are extracted from internet, mainly from wikipedia.

Technology used in the backend

Python 3

Python is an open source interpreted high-level programming language for general-purpose programming. Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python 3.0 was developed with the same philosophy as in prior versions. However, as Python had accumulated new and redundant ways to program the same task, Python 3.0 had an emphasis on removing duplicative constructs and modules, in keeping with "There should be one—and preferably only one—obvious way to do it".

Pip 3

Pip is a package management system used to install and manage software packages written in Python. Many packages can be found in the default source for packages and their dependencies - Python Package Index (PyPI). Pip is a recursive acronym that can stand for either "Pip Installs Packages" or "Pip Installs Python". Alternatively, pip stands for "preferred installer program".

We use pip to install the dependencies of our project

---

3 “Python (programming language) - Wikipedia”, https://en.wikipedia.org/wiki/Python_(programming_language), accessed May 29th 2018
Django 2\textsuperscript{5}
Django is a free and open-source web framework, written in Python, which follows the model-view-template (MVT) architectural pattern. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

SQLite 3\textsuperscript{6}
SQLite is a relational database management system contained in a C programming library. In contrast to many other database management systems, SQLite is not a client-server database engine. Rather, it is embedded into the end program. SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

Microsoft SQL Server\textsuperscript{7}
Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications-which may run either on the same computer or on another computer across a network (including the Internet).

\textsuperscript{7} “Microsoft SQL Server - Wikipedia”, https://en.wikipedia.org/wiki/Microsoft_SQL_Server , accessed May 29th 2018
Technology used in the frontend

HTML5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard. HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.

Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new <video>, <audio> and <canvas> elements were added, and support for scalable vector graphics (SVG) content and MathML for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as <main>, <section>, <article>, <header>, <footer>, <aside>, <nav> and <figure>, are added. New attributes are introduced, some elements and attributes have been removed, and others such as <a>, <cite> and <menu> have been changed, redefined or standardized.

CSS3

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Unlike CSS 2, which is a large single specification defining various features, CSS 3 is divided into several separate documents called "modules". Each module adds new capabilities or extends features defined in CSS 2, preserving backward compatibility.

---

Javascript\textsuperscript{10}

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

JQuery\textsuperscript{11}

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

\textsuperscript{10} “JavaScript - Wikipedia”, \url{https://en.wikipedia.org/wiki/JavaScript} , accessed May 29th 2018
\textsuperscript{11} “jQuery - Wikipedia”, \url{https://en.wikipedia.org/wiki/JQuery} , accessed May 29th 2018
Bootstrap

Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML (and CSS) based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap 3 supports the latest versions of the Google Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE8 and the latest Firefox Extended Support Release (ESR).

Since 2.0, Bootstrap supports responsive web design. This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone).

Starting with version 3.0, Bootstrap adopted a mobile-first design philosophy, emphasizing responsive design by default.

---

12 “Bootstrap (front-end framework) - Wikipedia”,
https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework), accessed May 29th 2018
Technology used for source code control

VS Code\(^{13}\)
Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences. It is free and open-source, although the official download is under a proprietary license.

Git\(^{14}\)
Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision control system it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

As with most other distributed version control systems, and unlike most client-server systems, every Git directory on every computer is a full-fledged repository with complete history and full version tracking abilities, independent of network access or a central server. Git is free and open source software distributed under the terms of the GNU General Public License version 2.

BitBucket\(^{15}\)
Bitbucket is a web-based version control repository hosting service owned by Atlassian, for source code and development projects that use either Mercurial (since launch) or Git (since October 2011) revision control systems. Bitbucket offers both commercial plans and free accounts. It offers free accounts with an unlimited number of private repositories (which can have up to five users in the case of free accounts) as of September 2010. Bitbucket integrates with other Atlassian software like Jira, HipChat, Confluence and Bamboo.
It is similar to GitHub, which primarily uses Git. Bitbucket has traditionally marketed its services to professional developers with private proprietary software code, especially since being acquired by Atlassian in 2010.

Gitlab\textsuperscript{16}

GitLab is a web-based Git-repository manager with wiki and issue-tracking features, using an open-source license, developed by GitLab Inc. As of December 2016, the company has 150 team members and more than 1400 open-source contributors. It is used by organisations such as IBM, Sony, Jülich Research Center, NASA, Alibaba, Invincia, O’Reilly Media, Leibniz-Rechenzentrum (LRZ), CERN, European XFEL, and SpaceX.

\textsuperscript{16}“GitLab - Wikipedia”, https://en.wikipedia.org/wiki/GitLab, accessed May 29th 2018
Technology used in the server architecture

Docker\textsuperscript{17}

Docker is a computer program that performs operating-system-level virtualization also known as containerization. Docker is primarily developed for Linux, where it uses the resource isolation features of the Linux kernel such as cgroups and kernel namespaces, and a union-capable file system such as OverlayFS and others to allow independent "containers" to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines (VMs). The Linux kernel's support for namespaces mostly isolates an application's view of the operating environment, including process trees, network, user IDs and mounted file systems, while the kernel's cgroups provide resource limiting for memory and CPU.

Apache HTTP Server\textsuperscript{18}

The Apache HTTP Server, colloquially called Apache, is a free and open-source cross-platform web server, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

Originally based on the NCSA HTTPd server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the World Wide Web, quickly overtaking NCSA HTTPd as the dominant HTTP server, and has remained most popular since April 1996. In 2009, it became the first web server software to serve more than 100 million websites. As of July 2016, it was estimated to serve 46% of all active websites and 43% of the top million websites.

TLS\textsuperscript{19}

Transport Layer Security (TLS) - and its predecessor, Secure Sockets Layer (SSL), which is now deprecated by the Internet Engineering Task Force (IETF) - are cryptographic protocols that provide communications security over a computer network. Several versions of the protocols find widespread use in applications such as web browsing, email, instant messaging, and voice over IP (VoIP). Websites are able to use TLS to secure all communications between their servers and web browsers.

\textsuperscript{17} "Docker (software) - Wikipedia", https://en.wikipedia.org/wiki/Docker_(software) , accessed May 30th 2018
The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. When secured by TLS, connections between a client (e.g., a web browser) and a server (e.g., wikipedia.org) have one or more of the following properties:

- The connection is private (or secure) because symmetric cryptography is used to encrypt the data transmitted.
- The identity of the communicating parties can be authenticated using public-key cryptography.
- The connection is reliable because each message transmitted includes a message integrity check using a message authentication code to prevent undetected loss or alteration of the data during transmission.

In addition to the properties above, careful configuration of TLS can provide additional privacy-related properties such as forward secrecy, ensuring that any future disclosure of encryption keys cannot be used to decrypt any TLS communications recorded in the past.

**Ubuntu**

Ubuntu is a free and open source operating system and Linux distribution based on Debian. Ubuntu is offered in three official editions: Ubuntu Desktop[9] for personal computers, Ubuntu Server for servers and the cloud, and Ubuntu Core for Internet of things devices. New releases of Ubuntu occur every six months, while long-term support (LTS) releases occur every two years. Ubuntu is the most popular operating system for the cloud, and is the reference operating system for OpenStack.

**Virtual Machine**

A virtual machine (VM) is an emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination. There are different kinds of virtual machines, each with different functions:

- System virtual machines (also termed full virtualization VMs) provide a substitute for a real machine.
- Process virtual machines are designed to execute computer programs in a platform-independent environment.

---

Some virtual machines, such as QEMU, are designed to also emulate different architectures and allow execution of software applications and operating systems written for another CPU or architecture. Operating-system-level virtualization allows the resources of a computer to be partitioned via the kernel’s support for multiple isolated user space instances, which are usually called containers and may look and feel like real machines to the end users.

Amazon Web Services\textsuperscript{22}

Amazon Web Services (AWS) is a subsidiary of Amazon.com that provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis. The technology allows subscribers to have at their disposal a full-fledged virtual cluster of computers, available all the time, through the Internet. AWS's version of virtual computers have most of the attributes of a real computer including hardware (CPU(s) & GPU(s) for processing, local/_RAM memory, hard-disk/SSD storage); a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, CRM, etc. Each AWS system also virtualizes its console I/O (keyboard, display, and mouse), allowing AWS subscribers to connect to their AWS system using a modern browser. The browser acts as a window into the virtual computer, letting subscribers log-in, configure and use their virtual systems just as they would a real physical computer. They can choose to deploy their AWS systems to provide internet-based services for their own and their customers' benefit.

The AWS technology is implemented at server farms throughout the world, and maintained by the Amazon subsidiary. Fees are based on a combination of usage, the hardware/ OS/software/networking features chosen by the subscriber, required availability, redundancy, security, and service options. Based on what the subscriber needs and pays for, they can reserve a single virtual AWS computer, a cluster of virtual computers, a physical (real) computer dedicated for their exclusive use, or even a cluster of dedicated physical computers. As part of the subscription agreement, Amazon manages, upgrades, and provides industry-standard security to each subscriber's system. AWS operates from many global geographical regions including 6 in North America.

EC2\textsuperscript{23}

Amazon Elastic Compute Cloud (EC2) forms a central part of Amazon.com's cloud-computing platform, Amazon Web Services (AWS), by allowing users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a web service through which a user can boot an Amazon Machine Image (AMI) to configure a virtual machine, which Amazon calls an "instance", containing any software desired.

\textsuperscript{22} “Amazon Web Services - Wikipedia”, https://en.wikipedia.org/wiki/Amazon_Web_Services , accessed May 30th 2018

A user can create, launch, and terminate server-instances as needed, paying by the second for active servers – hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy.

S324

Amazon S3 (Simple Storage Service) is a cloud computing web service offered by Amazon Web Services (AWS). Amazon S3 provides object storage through web services interfaces (REST, SOAP, and BitTorrent). Amazon launched S3 on its fifth publicly-available web service[citation needed], in the United States in March 2006[3] and in Europe in November 2007.

Amazon says that S3 uses the same scalable storage infrastructure that Amazon.com uses to run its own global e-commerce network. Amazon S3 is described as one of the wonders of the internet age and is the most profitable division under the entire Amazon company.

Azure25

Microsoft Azure (formerly Windows Azure) is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through a global network of Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems.

Azure iothub26

IoT Hub is a managed service, hosted in the cloud, that acts as a central message hub for bi-directional communication between your IoT application and the devices it manages. You can use Azure IoT Hub to build IoT solutions with reliable and secure communications between millions of IoT devices and a cloud-hosted solution backend. You can connect virtually any device to IoT Hub.

IoT Hub supports communications both from the device to the cloud and from the cloud to the device. IoT Hub supports multiple messaging patterns such as device-to-cloud telemetry, file upload from devices, and request-reply methods to control your devices from the cloud. IoT Hub monitoring helps you maintain the health of your solution by tracking events such as device creation, device failures, and device connections.

Azure stream analytics2728

Microsoft Azure Stream Analytics is a serverless scalable complex event processing engine by Microsoft that enables users to develop and run real-time analytics on multiple streams of data from sources such as devices, sensors, web sites, social media, and other applications. Users can set up alerts to detect anomalies, predict trends, trigger necessary workflows when certain conditions are observed, and make data available to other downstream applications and services for presentation, archiving, or further analysis.

Technology used for continuous Integration/Delivery/Deployment

Gitlab Pipelines\textsuperscript{29}

Gitlab pipelines enables continuous delivery for any deployment. The purpose of the deployment pipeline has three components: visibility, feedback, and continually deploy. It executes a chain of jobs and validations to guarantee the integrity of the software delivered.

Gitlab Image Registry\textsuperscript{30}

Gitlab Registry is a repository for Docker images. Docker clients connect to registries to download ("pull") images for use or upload ("push") images that they have built. Registries can be public or private.

\textsuperscript{29} "Introduction to pipelines and jobs | GitLab", \url{https://docs.gitlab.com/ee/ci/pipelines.html}, accessed May 30th 2018

\textsuperscript{30} “GitLab Container Registry | GitLab”\url{https://about.gitlab.com/2016/05/23/gitlab-container-registry}, accessed May 30th 2018
Technology used for planning and team communication

**Agile**\(^{31}\)

Agile software development describes an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s). It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.

The term agile (sometimes written Agile) was popularized, in this context, by the Manifesto for Agile Software Development. The values and principles espoused in this manifesto were derived from and underpin a broad range of software development frameworks, including Scrum and Kanban.

There is significant anecdotal evidence that adopting agile practices and values improves the agility of software professionals, teams and organizations; however, some empirical studies have found no scientific evidence.

**Scrum**\(^{32}\)

Scrum is an agile framework for managing work with an emphasis on software development. It is designed for teams of three to nine developers who break their work into actions that can be completed within timeboxed iterations, called sprints (30 days or less, most commonly two weeks) and track progress and re-plan in 15-minute stand-up meetings, called daily scrums.

**Kanban**\(^{33}\)

Kanban is a lean method to manage and improve work across human systems. This approach aims to manage work by balancing the demands with available capacity, and improving the handling of system level bottlenecks.

Work items are visualized to give participants a view of progress and process, from start to finish usually via a Kanban board. Work is pulled as capacity permits, rather than work being pushed into the process when requested.

---


In knowledge work and software development, this provides a visual process management system which aids decision-making about what, when and how much to produce. Although the underlying Kanban method originated in lean manufacturing (inspired by the Toyota Production System) it is used mainly for software development and technology related work. However Kanban can be applied to any area of work, and it can even be combined with other methods or frameworks such as Scrum.

Scrumban\textsuperscript{34}

Scrumban is an Agile management methodology describing hybrids of Scrum and Kanban and was originally designed as a way to transition from Scrum to Kanban. Today, Scrumban is a management framework that emerges when teams employ Scrum as their chosen way of working and use the Kanban Method as a lens through which to view, understand and continuously improve how they work.

Google Mail\textsuperscript{35}

Gmail is a free, advertising-supported email service developed by Google. Users can access Gmail on the web and using third-party programs that synchronize email content through POP or IMAP protocols.

Google Calendar\textsuperscript{36}

Google Calendar is a time-management and scheduling calendar service developed by Google. It allows users to create and edit events. Reminders can be enabled for events, with options available for type and time. Event locations can also be added, and other users can be invited to events.

Google Drive\textsuperscript{37}

Google Drive is a file storage and synchronization service developed by Google. Launched on April 24, 2012, Google Drive allows users to store files on their servers, synchronize files across devices, and share files. In addition to a website, Google Drive offers apps with offline capabilities for Windows and macOS computers, and Android and iOS smartphones and tablets. Google Drive encompasses Google Docs, Sheets and Slides, an office suite that

\textsuperscript{35} “Gmail - Wikipedia”, https://en.wikipedia.org/wiki/Gmail, accessed May 30th 2018
permits collaborative editing of documents, spreadsheets, presentations, drawings, forms, and more. Files created and edited through the office suite are saved in Google Drive.

Slack

Slack is a cloud-based set of proprietary team collaboration tools and services, founded by Stewart Butterfield. Slack began as an internal tool used by their company, Tiny Speck, in the development of Glitch, a now defunct online game. The name is an acronym for "Searchable Log of All Conversation and Knowledge".

Telegram

Telegram is a cloud-based instant messaging and voice over IP service developed by Telegram Messenger LLP, a privately held company registered in London, United Kingdom, founded by the Russian entrepreneur Pavel Durov. Telegram client apps are available for Android, iOS, Windows Phone, Windows NT, macOS and Linux. Users can send messages and exchange photos, videos, stickers, audio and files of any type.

Telegram's client-side code is open-source software but the source code for recent versions is not always immediately published, whereas its server-side code is closed-source and proprietary.

---