

2018

Meetings as a service in the cloud



Universitat  
Oberta  
de Catalunya

**Final degree project**

005.657 - FDP - Tools for Collaborative Work

**Marc Estévez Amén**

6/17/2018

Copyright © 2018 Marc Estévez Amén

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

# FINAL PROJECT REPORT

Project title:	<b>Meetings as a service in the cloud</b>
Author's name:	<b>Marc Estévez Amén</b>
Consultant's name:	<b>Fatos Xhafa</b>
Delivery date (mm/yyyy):	<b>06/2018</b>
Final's project area:	<b>Tools for Collaborative Work</b>
Degree:	<b>Bachelor's Degree in Computer Engineering</b>

## Project summary:

The project is motivated by the necessity of many emergent companies and organizations which want to take advantage from the benefits of working in group, without having the necessary knowledge and experience that it requires. So, the project objectives are:

- Increase efficiency of workgroup meetings
- Reduce administration time
- Improve workgroup evaluation
- Improve internal communication

In order to accomplish this commitment, it is proposed to implement a service-oriented system, 'Teamwork as a Service' (TaaS), which consists on a web application that gives some guidance during the processes of meeting preparation, execution, conclusion and evaluation.

To achieve this, it has been planned a project of less than six months of duration divided in several phases (project definition, working plan, planning and requirement analysis, project development and finally delivery) and carried out by a single person, assuming the risks that this involves. To do this, it is applied an iterative software life cycle methodology.

The resultant product consists on a J2EE application with several well differentiated layers that must be deployed in a web server to be accessed by the users via a web browser. The intention is the application can be used from most platforms and devices available in the market.

As it is seen above, the project has not the purpose of generating any economic profit, but its design allows generating benefits if desired.

Abstract:

Although team works are very profitable on projects development, bad management of working groups produces more troubles than benefits. One relevant tool for improving team works coordination are meetings, but they are a formal and complex way of communication, which require a certain knowledge and experience to implement it.

It has been implemented a service-oriented system, 'Teamwork as a Service' (TaaS), for helping on working group management. Specifically, a web application to facilitate the processes of meeting preparation, execution, conclusion and evaluation. This evidences two purposes: do a meeting giving their best performance and extract knowledge about how meetings work in order to improve future ones.

Key words:

meeting, teamwork, service, performance, enhancing, cloud, application, web

# INDEX

1.	Introduction .....	1
1.1.	Context and project justification.....	1
1.2.	Project objectives .....	2
1.3.	Methodology and approach.....	2
1.4.	Project planning .....	5
1.5.	Summary of obtained products .....	9
1.6.	Brief explanation of other report chapters .....	9
2.	The Meeting .....	11
2.1.	Meeting definition.....	11
2.2.	Meeting types .....	11
2.3.	Meeting structure .....	12
2.4.	Meeting preparation .....	13
2.5.	Meeting execution .....	14
2.6.	Meeting review .....	15
2.7.	Why meetings may be ineffective.....	15
3.	The Service .....	17
3.1.	Everything-as-a-Service.....	17
3.2.	Teamwork-as-a-Service (TaaS).....	17
3.3.	Meeting-as-a-Service.....	18
4.	Product analysis .....	19
4.1.	Product description.....	19
4.2.	Product functionalities .....	21
4.3.	Objectives definition .....	22
4.4.	Requirements analysis .....	23
4.5.	Roles .....	25
4.6.	Components distribution .....	25
4.7.	Use cases .....	26
4.8.	Legal and ethical considerations .....	28
5.	Product design.....	30
5.1.	Physical model design .....	30
5.2.	Software architecture design .....	32
5.3.	Application deployment.....	37
6.	The Application .....	38
6.1.	Sign in application .....	38
6.2.	Application Log in.....	38
6.3.	Preparing a meeting.....	39
6.4.	Leading a meeting .....	41
6.5.	Meeting conclusion .....	42
6.6.	Meeting review and rating.....	43
6.7.	Other functionalities .....	44
7.	Economic valuation .....	46
7.1.	Cost.....	46
7.2.	Profit.....	47
8.	Conclusions .....	48
8.1.	Lessons learned .....	48
8.2.	Objectives achievement.....	49
8.3.	Planification and methodology analysis.....	50
8.4.	Plans for the future .....	50
9.	Glossary .....	52

10.	Index of figures.....	53
11.	Index of tables.....	54
12.	Bibliography .....	55
13.	Annex.....	56
13.1.	Installation guide.....	56

# 1. Introduction

To begin with, it is presented the interest on carrying this project out and how is it proposed to achieve it. Hence, the following section gives a detailed description of the project, including motivations, project objectives, the applicable methodologies, a description of the roles and stakeholders which participate in the project and a tasks schedule with everything required to implement the project.

The intention of this section is to explain why this project is created for and how can it be achieved.

## 1.1. Context and project justification

Every day, new companies and organizations appear for boarding new ambitious and pushing projects. The most competitive projects are composed by people with different skills, so they put their abilities together to accomplish a certain objective. Many of those new companies are middle and small companies which want to take profit from the benefits of working in group, but not necessary have the necessary knowledge and techniques for working in group successfully.

In any case, even though working in group is very useful, because allows concentrate the efforts of many people in a single objective, it requires a good organization and coordination; otherwise, people lose motivation, their efforts turn into useless results and working in group becomes and athwart.

One important tool when working with groups are meetings. In a meeting, people gather together for sharing information and taking decisions. But meetings made by inexperienced people use to last too much or cannot aboard all the planned topics. For that reason, many people distrust meetings, considering them a waste of time; when, in fact, they are the key of success in projects working with groups.

To face this issue appears an interesting idea: In order to prevent companies from suffer the inconveniences of bad group coordination, is necessary to offer Meetings-as-a-service, within the paradigm of TeamWork-as-a-Service (Taas)\*.

Working in group allow putting different people abilities together to accomplish a certain objective. Getting good results implies good **coordination** and requires group **management skills**. In that direction, **meetings** provide group coordination.

\*TeamWork-as-a-Service (TaaS) is explained bellow on section in:  
3.2 Teamwork-as-a-Service (TaaS).

## 1.2. Project objectives

There are many concepts to consider when planning a meeting. Some of them can be difficult to understand and handle; or, merely, an organization don't want to spend resources on understanding them. This is what gives sense to this project so, the main goals of this project are:

- Increase efficiency of workgroup meetings
- Reduce administration time
- Improve workgroup evaluation
- Improve internal communication

In the first place, increasing the efficiency of workgroups meetings can be achieved by facilitating to users the necessary meetings know-how and providing some tools to supplant most of the required skills to perform successful meetings.

In the second place, reducing administration time can be resolved by centralizing meetings information and documentation and automating the generation of the documentation.

In the third place, improving workgroup evaluation can be tackled by gathering opinions from those people who participates in a meeting and by providing analysis tools of previous meetings.

Finally, improving internal communication can be solved by centralizing meeting related communications.

## 1.3. Methodology and approach

The objectives of the project can be reached applying different strategies and methodologies. The following lines presents which have been considered most suitable for this project and which one have finally applied.

### Strategies

On first place, using several existing products for managing meetings is the easiest way of front facing the problem. For example, a word processor can be used for preparing the meeting. Then, the resultant document can be delivered and confirmed using and email service. Next, the meeting can be followed and commented via a chat application or email. And finally, a word processor can be used again for generating the Minutes of the Meeting\*. In fact, this is what most organizations currently do. Despite the easy of applying this strategy, it presents some inconvenient: On one hand, the main handicap is that the information is scattered, which difficult getting global idea about the meeting necessary for evaluating its success. On the other hand, those tools do not give any guidance about

\* The **Minutes of the Meeting** is a formal document generated by the Meeting leader at the end of the meeting which collects everything remarkable in the meeting.

See: 2.6. Meeting review

Some examples of existing applications for setting up a Meeting are:

- **Word processors:** Microsoft Office Word, Google Docs, Open Office Word, etc.
- **Email services:** Gmail, Outlook, Yahoo, etc.
- **Chat applications:** Whatsapp, Hangouts, Slack, etc.

meeting development, so it is assumed that the people who prepare and attend the meeting know how to do it.

On second place, it could exist the possibility of adapting an already existing product for reaching the project objectives. To do so, on one hand, it is necessary to found existing products which supply somehow the requirements of the project and, on the other hand, those products must allow customizations. Taking these premises into consideration, is difficult to find such products; most products for managing meetings are proprietary, payment and very complex.

On third place, another option is developing a new product for managing meetings. This is the most flexible strategy but also the most difficult to implement. As a product created from the objectives of the project it could perfectly fit those objectives, but it also implies having less functionalities than other products created with a more general purpose.

Among the mentioned strategies, it has been decided to develop a new application for managing meetings. So, it has been accepted the challenge of creating an application from zero and has been assumed that the application will only cover the requirements of the project. In front of the existing applications that includes general purpose tools that can be used for setting up meetings, and those specific applications for managing meetings but considered very complex and payment; it is proposed to develop a simple application not only for gathering parameters received from the meeting participants but also guiding and giving advice to both meeting planner and meeting attendants in the whole processes of planning, execution and conclusion of a meeting.

So, the application consists on the following elements:

- A web portal where a user can create a meeting event, start and follow up on a meeting in progress and conclude the meeting and generate deliverable documentation.
- A user can also confirm attendance to a meeting and agree a date time, follow up on a meeting in progress and rate a meeting the user has attended.
- A user can see all the information about meetings he or she has attended or has been invited to attend.

It has been decided to develop a new application for managing meetings because, despite of being the most delicate strategy, is the one that fits better with project requirements.

Samples of products for managing meetings:

- **MeetingBooster:**  
<http://www.meetingbooster.com/>
- **Wire:** <https://go.wire.com>
- **Diligent:**  
<https://diligent.com>

## Implementation methodology

After deciding what to do, the next step is choosing how to do it. When speaking about software development it exists several methodologies very extended used. Some of the them are:

- Waterfall model
- Incremental or iterative life-cycle
- Agile software development

A very brief explanation of those methodologies could be that, on one hand, the waterfall model is a very linear model which allow little flexibility, that implies little margin for error on the initial phases of the project; on other hand, the incremental or iterative life-cycle allows more flexibility, introduced by repeating the development cycle; finally, the agile software development is the most flexible, allowing modifications of any phase of the development cycle at any time.

Taking this in consideration, while waterfall model is more convenient for very static and strongly defined projects, such as those applications which requirements are defined by law or by strong standards; the agile software development is more suitable for very changing applications, which is neither the case. So, the proposed methodology for this project is based on the iterative or incremental software lifecycle because despite of needing an accurate planification, it also allows changes when necessary.

As it is shown in Figure 1, it is defined by six phases: Planning, requirements, analysis and design, implementation, test and validation. Once a project has gone through all these phases it is called a cycle. After a cycle ends, it used to appear some issues or modifications required to satisfy the project requirements; so, a new cycle must start to implement the mentioned modifications. Cycles must repeat until the product reach the desired level of quality.

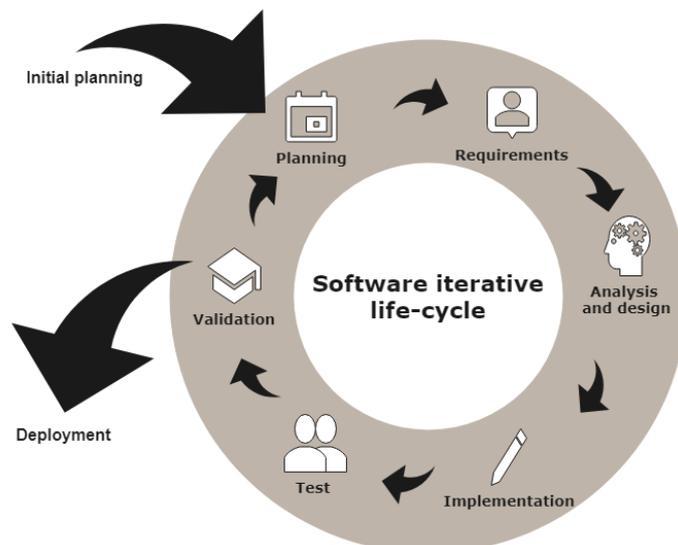


Figure 1: Software iterative life-cycle

## 1.4. Project planning

The following section includes the necessary planification for developing the product. It includes an identification of the stakeholders, who can affect the project development, a list of tasks necessary for carrying out the project and a time schedule, represented as a Gantt diagram.

### Stakeholders

A stakeholder is a member of the “groups without whose support the organization would cease to exist” [1]. So, as an academic project the list of stakeholders is quite short and mostly includes the student and the professors:

- The Universitat Obertat de Catalunya (UOC) is an academic organization that will give advice and guide during the project development. Furthermore, at the end of the project it will evaluate the project success. Those tasks will be performed by the figures of the professors Fatos Xhafa and Atanasi Daradoumis Haralabus.
- Marc Estévez Amén is a student at UOC and will has the responsibility of the whole project planning, development, testing, documentation and presentation.

### Organization chart

The graph in Figure 2 shows a visual representation of the project organization chart:

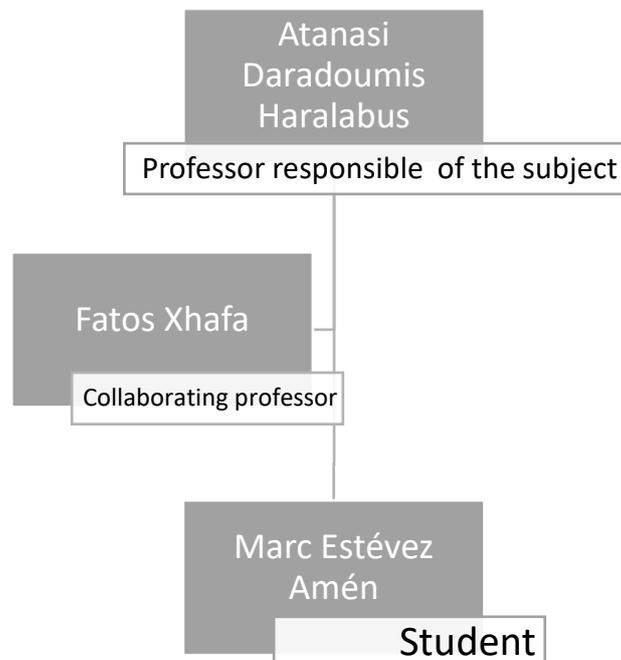


Figure 2: Project stakeholders chart

## Collaborators

Apart from the stakeholders it has been achieved some collaboration with external organizations for the tasks of testing and evaluating the product:

- The software development company Deister S.A., and its ICT department has offered as volunteer for performing a trial of the product in its latest versions to evaluate its functionalities.
- A consultant department of RICOH SPAIN IT SERVICES SL has also offered as volunteer for performing a trial of the product in its latest versions to evaluate its functionalities.

Deister S.A. is an IT company specialiced in developing software applications for bussines logic.  
<http://www.deister.net>

RICOH SPAIN IT SERVICES SL is a department of RICOH for optimizing the IT structure of a company.  
<https://www.ricoh.es/soluciones-negocios/todos-servicios/it-infraestructure-services/servicio-de-soporte-de-TI-corporate.html>

## Tasks table

Without considering those organizations necessary for providing specific services such as software or hosting, this project doesn't involve coordination with third party organizations. So, it is possible to set up the project development plan considering only the student schedule.

The Table 1 shows a rough guess of a schedule with the main deliveries and important dates. Some dates might slightly change after finishing the requirement gathering and project design, but it is a good approximation of the final plan. Despite this, considering the project is within the context of an academic work, the delivering deadlines are immutable.

Table 1: Project tasks schedule

Task Name	Work	Start	Finish	Priority
<b>Project definition</b>	<b>14 hours</b>	<b>Wed 21/02/18</b>	<b>Mon 26/02/18</b>	<b>500</b>
Course plan reading	2 hours	Wed 21/02/18	Wed 21/02/18	500
Classroom contact meeting	2 hours	Thu 22/02/18	Fri 23/02/18	200
Project proposals	8 hours	Fri 23/02/18	Sun 25/02/18	600
Project acceptance	2 hours	Sun 25/02/18	Sun 25/02/18	800
<b>Working plan</b>	<b>30 hours</b>	<b>Tue 27/02/18</b>	<b>Sun 11/03/18</b>	<b>500</b>
Objective definition	4 hours	Tue 27/02/18	Wed 28/02/18	500
Establish ethical and legal frame	2 hours	Thu 01/03/18	Fri 02/03/18	500
Estimation of project expenses	2 hours	Fri 02/03/18	Sat 03/03/18	500
General risks analysis	2 hours	Sat 03/03/18	Sat 03/03/18	400
Tasks list definition	4 hours	Sat 03/03/18	Sun 04/03/18	600
Gantt diagram implementation	10 hours	Sun 04/03/18	Wed 07/03/18	600
PAC1 preparation and delivery	6 hours	Thu 08/03/18	Sat 10/03/18	800
<b>Planning and requirement analysis</b>	<b>39 hours</b>	<b>Mon 12/03/18</b>	<b>Thu 29/03/18</b>	<b>500</b>
Project planification	8 hours	Mon 12/03/18	Fri 16/03/18	500
Requirements analysis	8 hours	Sat 17/03/18	Sun 18/03/18	500
Project analysis and design	8 hours	Sun 18/03/18	Wed 21/03/18	500
Detailed risks analysis	4 hours	Wed 21/03/18	Fri 23/03/18	400
Working plan evaluation (Phase 1)	3 hours	Sat 24/03/18	Sat 24/03/18	300
PAC2 Preparation and delivery	8 hours	Sat 24/03/18	Sun 25/03/18	800
<b>Project development</b>	<b>122 hours</b>	<b>Fri 30/03/18</b>	<b>Sun 13/05/18</b>	<b>500</b>

<i>Set up database</i>	8 hours	Fri 30/03/18	Sun 01/04/18	500
<i>Set up application server</i>	8 hours	Sun 01/04/18	Tue 03/04/18	500
<i>Beans implementation</i>	24 hours	Wed 04/04/18	Fri 13/04/18	500
<i>Backend processes implementation</i>	30 hours	Fri 13/04/18	Sun 22/04/18	500
<i>Front end implementation</i>	30 hours	Sun 22/04/18	Fri 04/05/18	500
<i>Testing</i>	16 hours	Fri 04/05/18	Tue 08/05/18	400
<i>Working plan evaluation (Phase 2)</i>	3 hours	Wed 09/05/18	Thu 10/05/18	300
<i>PAC3 Preparation and delivery</i>	3 hours	Fri 11/05/18	Sat 12/05/18	800
<b>Final delivery</b>	<b>91 hours</b>	<b>Mon 14/05/18</b>	<b>Sun 17/06/18</b>	<b>500</b>
<i>Requirements audit</i>	4 hours	Tue 15/05/18	Wed 16/05/18	300
<i>Deploy product in a hosting</i>	8 hours	Thu 17/05/18	Sat 19/05/18	500
<i>Pilot test</i>	4 hours	Sat 19/05/18	Sun 20/05/18	300
<i>Write manuals</i>	24 hours	Sun 20/05/18	Mon 28/05/18	500
<i>Project's report</i>	24 hours	Tue 29/05/18	Wed 06/06/18	500
<i>Project's presentation</i>	16 hours	Thu 07/06/18	Mon 11/06/18	500
<i>Working plan evaluation (Phase 2)</i>	3 hours	Tue 12/06/18	Wed 13/06/18	300
<i>PAC4 preparation and delivery</i>	8 hours	Wed 13/06/18	Sat 16/06/18	800
<b>Virtual defense</b>	<b>4 hours</b>	<b>Mon 18/06/18</b>	<b>Sun 24/06/18</b>	<b>500</b>
<i>Virtual presentation</i>	2 hours	Tue 19/06/18	Tue 19/06/18	800
<i>Answer requests</i>	2 hours	Wed 20/06/18	Wed 20/06/18	500

### Project summary

On one hand, it is remarkable that there's only one person responsible of doing the whole project, the student. This fact makes the project planification very sensitive to unexpected situations, such as illness, accidents, family or working matters, etc. Moreover, the mentioned person has a full-time job that occupies the major part of the week; for that reason, the planning puts the major of the work load at the weekends, especially at deliveries deadlines. In consequence, unexpected situations must be palliated by redistributing the work load during the week.

On the other hand, priorities have been defined by taking 500 as the average priority value. This value is defined for those general tasks necessary for the project development. The priorities below 500 are defined for those tasks not necessary for the project success, but they can noticeably improve the project. Finally, priorities above 500 are defined for the essential tasks, that is, those tasks that highly compromise the project success, generally those priorities are assigned to deliveries.

Table 2: Project schedule summary

CONCEPT	AMOUNT
<b>START</b>	Wed 21/02/2018
<b>FINISH</b>	Sun 24/06/2018
<b>DURATION</b>	495 days
<b>WORK</b>	300 hours
<b>COST</b>	0 €



Figure 3: Gantt diagram for project planning

## 1.5. Summary of obtained products

The project target is to build a single product, consisting on a web application providing Meeting-as-a-Service\*. That is, an environment where a user can create a meeting event, start and follow up on a meeting in progress and conclude the meeting and generate deliverable documentation.

\* Meeting-as-a-Service is explained in further sections:  
3.3 Meeting-as-a-Service

The product will offer a distribution of the mentioned application and specific instructions for installing it into an application server and join it to a database. It does not provide any server or database in the cloud by itself. This model allows each organization to have full control about their network and database management.

Even though, a functional distribution of the application will be deployed in a cloud service for enabling a demonstration environment that can be accessed by the professors, evaluators or anyone interested in seeing and testing the product as a service; as it would be, in a real corporative environment.

## 1.6. Brief explanation of other report chapters

Until now, it has been explained a global view of the project, the motivations of carrying this project out and how is it proposed to achieve it. The following sections are focused on the product; that is, giving a detailed description of the product and what is composed of. It has been proposed to give the mentioned product description through the following sections:

- The meeting
- The service
- Product analysis
- Product design
- The application

In the first place, **the meeting** section tries to expose everything relevant about what a meeting is and why is it relevant for the project. The section deals with the common parts of a meeting and what are they strengths and weaknesses.

In the second place, the product consists on providing Meeting-as-a-service, **the service** section introduces what this paradigm is and how does it fit within the general concepts of Everything-as-a-service and Teamwork-as-a-service.

In the third place, **product analysis** section shows a description of the application, a requirements analysis, the relationship between the users and the application and ethical and legal considerations.

In the fourth place, the **product design** section gives a complete description of the application design, with the intention of offering a deep knowledge about how this kind as application works and how could it be implemented.

Finally, **the application** section shows the main functionalities of the application and a glance of the application behaviour through some screen shoots.

## 2. The Meeting

One of the main objectives of the project is to provide some help on meeting organization. To achieve this objective, it has been necessary to gather and study many information about what a meeting is and what are best practices when preparing meetings. This section exposes all this knowledge, which is considered the product foundations.

### 2.1. Meeting definition

Organization projects used to involve many people working together for the same purpose. For that reason, communication between project participants is a determinant fact of success in projects working with groups. Communication within a project occurs in many ways: face to face chat, videoconference, telephone conference call, e-mail, document sharing, etc. One of these ways of communication are meetings.

In a **meeting**, two or more people come together to discuss one or more topics, often in a formal setting.

#### Further reading:

See the most widely used manual for govern a meeting of a diverse range of organizations in the book:

*General Henry M. Robert. (2011). Robert's Rules of Order Newly Revised 11th. Florida: Da Capo Press, A Member of the Perseus Books Group. [14]*

### 2.2. Meeting types

Depending on the meeting intention, they can be classified in different types. It exists many types of meetings, the product has focused on the following:

- Inform
- Consult
- Solve problems
- Make decisions

#### Informing Meetings

These are the most common meetings where one person, generally the one who convokes the meeting, has relevant information to communicate all those present. Such meetings tend to be formal, and their objective is to spread the information and how to use it for the best.

#### Consulting Meetings

These are meetings to discuss a specific policy or innovation, and for gathering the point of view, feelings or observations about this policy or ideas from all those present. Some examples could be:

- Review a current policy
- State its deficiencies
- Suggest change
- Stress the advantages of such change
- Admit any weaknesses
- Invite comments

### Problem solving meeting

On the meetings the person who convokes describes a problem as much clear as possible, so the people in the meeting can generate ideas, offer advice and reach conclusion for solving the problem. The members should be selected according to their specific knowledge, their experience or the interest on the problem.

### Decision making meetings

These are meetings to decide which are necessary actions to be taken to carry out a project or a task. They tend to follow an established method or procedure:

- Description of the problem
- Analysis of the problem
- Draw out ideas
- Decide which is best
- Reach conclusions

## 2.3.Meeting structure

While other kind of communication are more informal and improvised, meetings general involves certain planning and organization.

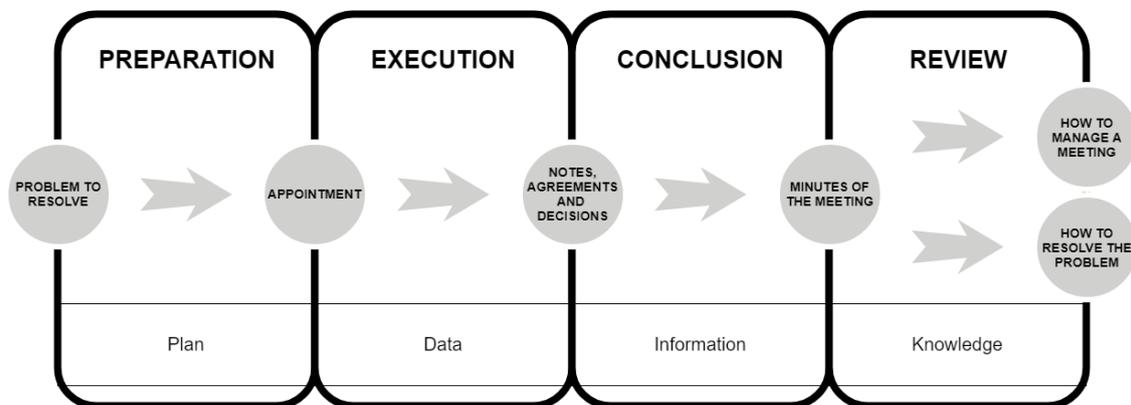


Figure 4: Structure of a meeting

Figure 4 shows a scheme of the common meeting structure. As it can be noticed, a meeting is divided in four phases:

- Preparation
- Execution
- Conclusion
- Review

Each phase implies some tasks, that are explained in Table 3.

Table 3: Description of the phases of a meeting

STAGE	SUBJECT	DESCRIPTION
<b>PREPARATION</b>	Description	Define the meeting reason, a summary and where it will take place.
	List Topics	Define a list of topics that will be discussed in the meeting ordered by priority and defining a duration forecast.
	Documentation	Attach all those documents, pictures, graphs, videos, etc. that might be used in the meeting.
	List participants	List of people that might attend the meeting.
	Schedule agreement	Define one or more date times available for the meeting in order to agree a date for the meeting.
<b>EXECUTION</b>	Roll call	List all the people who finally attended the meeting.
	Topics control	Indicates each topic starting and ending date times.
	Provide feedback	Allow adding annotations during the meeting with relevant information provided during the meeting such as decision making, delivered tasks etc.
<b>CONCLUSION</b>	Agreements	Write a text with the achievement agreements discussed during the meeting.
	Pending	Write a text with the pending issues to be discussed in further meetings.
	Minutes of the Meeting generation	Generate the Minutes of the Meeting document and send it to all participants.
<b>REVIEW</b>	Meeting review	Distribute and review the Minutes of the Meeting
	Meeting evaluation	Allow each participant to rate the meeting including comments about the meeting development.

## 2.4. Meeting preparation

The inherent formality of a meeting involves some planification. As it happens on any tasks, the quality of planification will majority determine the success of the task, the meeting in this case. For that reason, is very important to pay as much attention as possible on the meeting planning.

### Is the meeting appropriate?

A meeting involves certain amount of time and resources, both for the person who convokes and for those who attend it. So, unnecessary meetings are a waste of time and resources, and generates frustration and lowers the motivation to participate in future meetings. For that reason, before convoking a meeting the leader must be complete sure that the meeting is necessary, or if exists an easier way of communication to solve the situation.

## What kind of meeting

As it is said before, exists many classifications of meetings which proportionate some rules and techniques for a better planning and execution of the meeting. The bare fact of deciding what type of meeting is required makes the leader taking important decisions about requirements of the meeting. Such as, how many people should attend the meeting, what profiles does it require, how many topics it should deal with, etc.

## Who will attend the meeting?

Attendants selection is another important decision to take when preparing the meeting. It largely depends on the type of meeting, the size of the organization or department and the magnitude of the meeting. In a nutshell, it must be determined the parts interested in the topic, or those who can proportionate a worthy knowledge, and invite a significative representation of this parts. Depending on the volume of the organization or departments it could one person, or the whole staff.

## Agenda

The meeting agenda includes the most relevant information of the meeting. As more precise the information of the agenda is, the better informed the invited person will be, so the meeting will be most profitable. Some essential content for a well-written agenda are:

- Give the time and place of the meeting.
- List the topics to be covered, indicating who will introduce them.
- Have any relevant papers attached.
- Give the time the meeting will close.

## 2.5.Meeting execution

The most important when executing a meeting is the fact that the leader must conduct the meeting. It can be summarized as:

- Set the meeting pace
- Introduce the topics
- Ensure that all the attendants give and receive the information as expected from them.

On one hand, the firmness when conduction the meeting highly depends on its formality and the leader's strength within the organization structure. Anyway, in one way or another, is mandatory the leader keep its role to reach the meeting success.

On the other hand, is important that the meeting takes place in a climate of serenity and implication among all those present. To keep the complicity of the attendants it might be helpful to

keep all necessary resources handy by the time of the meeting execution, such as documents, presentations, graphics, etc.

## 2.6. Meeting review

After the meeting ends, it still some pending tasks to do.

In first place, the meeting execution might be animated and spirited, so most of the accounted elements during the meeting may have not been annotated properly or could have been ignored. For that reason, after the meeting ends is vital to review these annotations; and add or rewrite everything necessary. The mentioned review must be done by the meeting leader after the meeting ends, but not much later, so the meeting memories and feelings must be still fresh.

In second place, the meeting leader must take everything accounted in the meetings: comments, decisions taken, feelings, etc. and written down in a document called 'Minutes of the Meeting'.

Finally, the people invited to the meeting have to review the Minutes of the Meeting and communicate their point of view about the meeting to its leader as far as possible.

### **Minutes of the Meeting (MoM)**

The Minutes of the Meeting is a formal document describing the most relevant events and information of the meeting. It must contain information on both planning and execution:

- Title of the meeting
- Date
- Duration
- Attendants
- Topics discussed at the meeting
- Comments made
- Decisions taken
- Agreements
- Pending issues

## 2.7. Why meetings may be ineffective

Considering all the stuff mentioned on this block, there are many reasons why the meeting may be ineffective. There are some of the main facts:

- The meeting is unnecessary.
- The meeting lacks a clarity of purpose.
- Inappropriate style of leadership. For example, the leader dominates and closes down or disregards other contributions.

- The leader exercises little control and allows one or two members to dominate the proceedings; so, the meeting revolves around discussion of trivial issues.
- The meeting is too large thereby decreasing people attention and complicity.
- Decisions emerge that are not truly representative.
- Problems are talked about rather than finding solutions.
- Decisions are delayed or not acted upon.
- No clear-cut decisions are made.
- Minutes are inaccurate or seen as being manipulated by the leader for his/her own purposes.
- The wrong people are present, thus preventing the meeting proceeding effectively. Those present have to refer back to another person and are therefore unable to comment effectively.

Consequently, the best way for having an effective meeting is a good preparation and a capable leader.

## 3. The Service

The main purpose of this project is to offer meetings management as a service, known as Meeting-as-a-Service. For a better understanding of what does it means, is necessary to mention the paradigms of Everything-as-a-Service and Teamwork-as-a-Service. These concepts are all related with cloud computing and creating applications capable of giving a service to the users. All those concepts are exposed in detail on this section, with the objective of giving a clear idea of what offering a meeting as a service means.

### 3.1. Everything-as-a-Service

Everything-as-a-Service (EaaS, XaaS or \*aaS) is a subset of cloud computing. The basic idea is to offer a service over the web or similar networks through an application. Some key characteristics of EaaS are:

- Low barriers to entry is a common method of offerings, with services typically being available to or targeting consumers and small businesses.
- Little or no capital expenditure as infrastructure is owned by the provider.
- Massive scalability is also common, though this is not an absolute requirement and many of the offerings have yet to achieve large scale.
- Multitenancy enables resources (and costs) to be shared among many users.
- Device independence enables users to access systems regardless of what device they are using (e.g. PC, mobile, etc.).
- Location independence allows users remote access to systems.

The paradigm started with the basic software as a service (SaaS), with cloud providers that offered individual software applications. But soon started to appear other related services such infrastructure as a service (IaaS) or communications as a service (CaaS). With so many kinds of resources proliferating within the model of cloud computing, the developer's community started to reference them as everything as a service (EaaS). The name itself holds the ambitious idea that most of actual real-life services could fit in this term, therefore could be computerized and offered in the cloud.

Among all those, an engaging service to be included was teamwork as a service.

### 3.2. Teamwork-as-a-Service (TaaS)

Teamwork might be described as the process of working collaborative with a group of people in order to achieve a goal.

It is a crucial part in various ambits: work, academical, corporative, etc. For a teamwork to work well all components should use their better skills and provide constructive feedback, despite their personal conflicts and disagreements.

With the aim of getting the better teamwork profit, exists diverse techniques or methodologies, which can be classified as:

- Team developing
- Communication
- Group management
- Delegation
- Motivation

Therefore, Teamwork-as-a-Service (TaaS) includes all services applicable to the methodologies mentioned before. The most relevant teamwork technique for this project are meetings.

### 3.3.Meeting-as-a-Service

One way of communication in teamwork is by setting a meeting. But, as it has been mentioned before\*, planning and executing a meeting could become very complex task. Therefore, meeting as a service provides some tools over cloud computing for helping on setting meetings.

In first place, Everything-as-a-Service (EaaS) references a diverse set of services offered over cloud computing. In second place, Teamwork-as-a-Service (TaaS) is a subset of EaaS which includes only those services related with team works. Finally, Meeting-as-a-Service is a subset of TaaS including those services related to meetings. The objective if this project is developing a web application providing Meeting as a Service.

#### Review:

\* Review section about 2.3 Meeting structure.

## 4. Product analysis

The current section gives a deeply description of the application and its functionalities. After reading this section, the reader will have a clear idea about what the application consists on and what a user can do with the application.

### 4.1. Product description

The project consists on developing an application providing the necessary environment and tools for managing team work meetings. According to this, the product will consist on an application providing meetings management as a service, within the paradigm Teamwork-as-a-Service (TaaS)\*.

**Review:**

\* See 3.2. Teamwork-as-a-Service (TaaS) section.

The application consists on a repository of code including both application server and the application itself, ready to be connected to a database and be deployed as a cloud service as it can be seen in Figure 5.

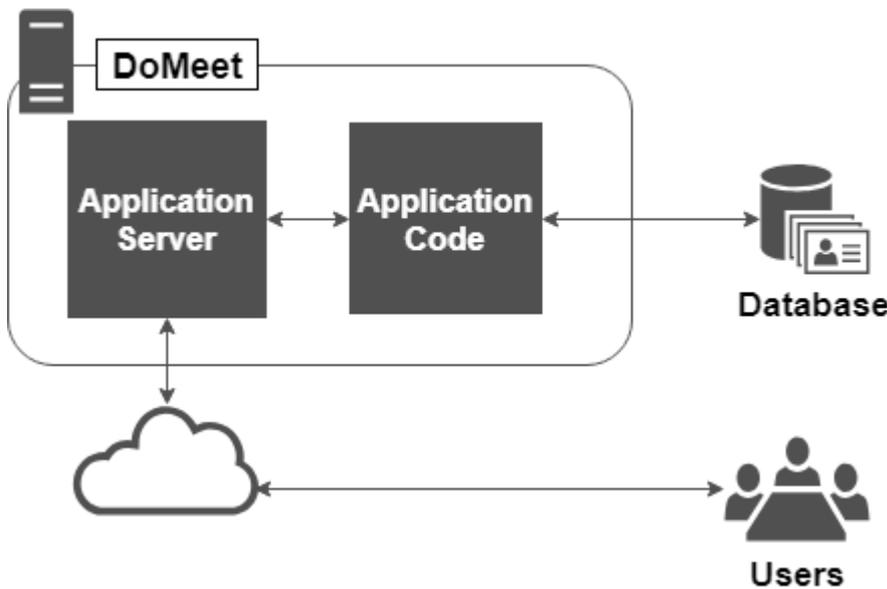


Figure 5: DoMeet architecture

#### Product name

It has been decided to call the application DoMeet. This name reflects the main target of the project, that is a meeting, and its main purpose, do a meeting.

#### Product vision

The product is aimed to personal or corporative groups with the need of gathering together to discuss one or more subjects and taking decisions about them. It embraces and unlimited scope of teamwork groups: companies, students, neighbours, etc.

The limitations of the product are that requires an electronic device for access the meeting information, mainly for the planification phase.

### **Product opportunities**

It exists many ways of scheduling meetings, but very few applications for leading a meeting and generating its documentation and existing applications are payment and very complex. Consequently, those must be the most remarkable characteristics of the product.

Besides, nowadays users access applications from many diverse devices and operative systems: PC, MAC, smartphones, tablets, Android, IOS, Microsoft, etc. For that reason, it would be very profitable for the product would be accessible from the most of those devices.

The most remarkable of the project is, without a doubt, the capability of generating the meeting documentation and analysing meetings success through the analysis of previous meetings, considering duration, subjects, rates, etc.

### **Product critical points**

In summary, the product most profitable opportunities, or critical points, are:

- Easy to use
- Generating a PDF with the Minutes of the Meeting which can be send to all the meeting participants.
- Allow participants to rate the meeting and giving comments.
- Having a good reporting of previous meetings to analyse the success or failure of meetings.

### **How to compare the product with other products in the market**

Similar products on the market are focused on schedule, but many little give any guidance on how a meeting should be planned and executed. That must be the differentiator factor.

Furthermore, the capability of generating the Minutes of the Meeting automatically is also an attractive functionality that all meetings leader would have.

Considering all the information given until now, the developing product can be described as:

For personal or corporative group teams who wish to manage successful meetings, **DoMeet** is a cloud service that gives guidance on meetings setting up, scheduling and executing. Contrary to scheduling tools, such as Doodle or Google suite, this product offers guidance on meeting definition, allows generating documentation and analyze meetings success.

## 4.2. Product functionalities

The lines below try to explain in detail the functionalities of the application, considering the mentioned objectives, needs and opportunities of the project. These functionalities are divided in blocks, depending of the meeting stage.

### Planning and preparation

The quality of the meeting planification determines in great measure its success. Besides, making a good planification requires a certain knowledge and methodologies. For that reason, the application functionalities for this phase consist of a set of tools for helping in meeting creation and giving advice on how to create it.

The application provides the following functionalities for the meeting planning:

- Meeting creation
- Meeting schedule
- Definition of meeting general description and topics
- Convene and invite to the meeting
- Mailing to meeting attendees with the meeting details
- Guidance on meeting preparation:
  - Notifications about meeting standard requirements: topics definition, duration, etc.
  - Guidance deciding about what type of meeting suits the best for a certain project Execution
  - Help on meeting preparation depending the chosen type
  - Check attendees schedule issues

### Execution

After planned and scheduled a meeting properly, the next determining stage is the meeting execution. The most striking obstacles during meeting executions are mainly related to time management and topics prosecution.

The application provides the following functionalities for the meeting execution:

- Meeting start/end
- Topics prosecution
- Taking notes
- Annotation decisions and agreements

- Guidance on meeting execution
  - Time control notifications
  - Advice on agreements and decisions

### **Meeting conclusion**

After the meeting ends, the leader must review all decisions and notes taken during the meeting. Finally, those annotations must be sent to all invited people in form a document called Minutes of the Meeting.

The application provides the following functionalities for the meeting conclusion:

- Add and rewrite annotations taken during the meeting
- Generating the Minutes of the Meeting
- Mailing the Minutes of the Meeting

### **Review**

Once the meeting has been closed and all the documentation has been generated, the people involved in the meeting are able to review those documentation for extracting the necessary knowledge to resolve the objective of the meeting. Besides, they can get extraordinary knowledge about how to plan and execute future meetings.

The application provides the following functionalities for the meeting review:

- Show information of previous meetings
- To rate attended meetings
- Add comments about the meeting
- Download the Minutes of the Meeting

### **Other functionalities**

Apart from those functionalities directly related to meetings phases the application includes some extra functionalities due to is a cloud application:

- Meetings database store
- Centralized communication

## **4.3. Objectives definition**

Considering the project description and the roles definition given above, the following content is a collection of the technical objectives of the product:

- User account management.
- Guided creation and scheduling of meetings.
- Users mailing.
- Guided meeting execution.
- Generation of a meeting documentation.
- Meeting rating.
- Past meetings analysis and review.

## 4.4. Requirements analysis

From the list of objectives defined before, it has been built a list of requirements, as it is shown in Table 4. Furthermore, it has been assigned to each requirement a priority and a cost. The main factors that have considered for deciding the priority of each requirement are the remarkable capabilities that distinguishes the product from other existing products, which are: guidance on meeting management, documentation generating, meeting rating and previous meetings analysis.

Apart from this, it is necessary to remark that, on one hand, the priority has been established from 0 to 10, where 0 is lowest priority and 10 the highest; on the other hand, the developing cost has been defined in a range between 0 and 5, where 0 is low cost and 5 is a high cost.

Table 4: Product requirements

<i>ID</i>	<i>Description</i>	<i>Priority</i>	<i>Cost</i>	<i>Proposal</i>
R001	As an unregistered user, I want to register as a user.	10	3	<b>Accepted</b>
R002	As a user, I want to log in into the application.	10	3	<b>Accepted</b>
R003	As a user, I want to log out the application.	8	2	<b>Accepted</b>
R004	As a user, I want to modify my personal data.	6	2	<b>Accepted</b>
R005	As a user, I want to recover my password if I did not remember it.	2	3	Delayed
R006	As a user, I want to complain if I found any comment offensive or inappropriate.	2	4	Delayed
R007	As a meeting leader, I want to create a new meeting. That is, define the main attributes of the meeting: title, description, date, duration, subjects, documentation.	10	4	<b>Accepted</b>
R008	As a meeting leader, I want to update a meeting data.	7	3	<b>Accepted</b>
R009	As a meeting leader, I want to cancel a meeting that has not been realized.	5	3	<b>Accepted</b>
R010	As a meeting leader, I want to invite other users to a meeting I have previously created.	8	3	<b>Accepted</b>
R011	As a meeting leader, I want to know if the invited users have received and have read the invitation.	3	2	Delayed
R012	As a meeting leader, I want to propose one or more available dates for the meeting. In the first case, the meeting date will be immutable; in the second case it will begin an agreement process for determining the meeting date.	2	4	Delayed
R013	As a meeting leader, I want to define periodical meetings within a range of time.	2	4	Delayed
R014	As a meeting leader, I want to create a meeting as a copy of a previous one.	2	2	Delayed
R015	As a meeting leader I want to send to all invited users the final date of the meeting and ask for confirmation. In case of the meeting doesn't allow invited users choosing a date, the invitation message will be enough.	5	3	Delayed

R016	As a meeting leader, I want to send any meeting modification or cancellation to all invited users.	5	4	Delayed
R017	As a meeting leader, I want to know if exists any issues between the selected dates and the invited users working hours. That is, if I have invited users to a meeting out of their working hours.	6	4	<b>Accepted</b>
R018	As a meeting leader, I want to define how much time each subject should approximately last.	7	2	<b>Accepted</b>
R019	As a meeting leader, I want to prioritize the subjects within a meeting.	7	2	<b>Accepted</b>
R020	As a meeting leader, I want to know if I have schedule too many, or too less, subjects for the meeting. According to its duration and the defined standards.	6	3	<b>Accepted</b>
R021	As a meeting attendance, I want to receive an invitation to a meeting and know if they are some available dates.	6	3	<b>Accepted</b>
R022	As a meeting attendance, if exists the possibility of choosing a date for the meeting I would response with the ones that suits me the best.	2	4	Delayed
R023	As a meeting attendance, I want to accept or decline an invitation to a meeting and adding a comment if is necessary.	5	3	<b>Accepted</b>
R024	As meeting leader, I want to lead a meeting during its execution. That is, start a meeting, indicate subjects progression and conclude the meeting.	5	5	<b>Accepted</b>
R025	As a meeting attendance, I want to join a meeting the moment it starts.	4	2	<b>Accepted</b>
R026	As a user participating in the meeting I want to post messages with relevant information during the meeting.	3	5	Delayed
R027	As a meeting leader, I want to receive a state of the progression of the meeting during its execution.	7	4	<b>Accepted</b>
R028	As a meeting leader, I want to add a list of the decisions taken within a meeting.	5	3	<b>Accepted</b>
R029	As a meeting leader, I want to add a list of agreements reached within a meeting.	5	3	<b>Accepted</b>
R030	As a meeting leader, I want to add a list of pending issues noticed in a meeting.	5	3	<b>Accepted</b>
R031	As a meeting leader, I want to add some comments or notes to a meeting.	5	3	<b>Accepted</b>
R032	As a meeting leader, I want to automatically generate the Minutes of the Meeting document.	9	4	<b>Accepted</b>
R033	As a meeting leader, I want to send the Minutes of the Meeting document to all the users invited to the meeting.	7	3	<b>Accepted</b>
R034	As a meeting attendance, I want to rate a meeting I have attended and some comments.	9	4	<b>Accepted</b>
R035	As a user I want to see the information about all the meetings I have participated, or I have been invited to participate.	7	3	<b>Accepted</b>
R036	As a user I want to see the list with the most relevant information (number of participant, duration, rating, etc.) of all the meetings I have participated, or I have been invited to participate.	7	3	<b>Accepted</b>

R037	As a meeting leader, I want to see the meeting rate and comments.	8	3	Accepted
R038	As a user, I want to synchronize meeting schedules with Google Calendar.	3	3	Delayed

As it can be seen in Table 4, requirements that have been considered Accepted or Delayed of being included in the current project. To accept or deny each requirement has been considered its priority, the cost and the available resources for the project. Those requirements not Accepted are not completely discarded, they might be implemented if there would be spare resources at the end of the project or in farther projects.

### 4.5. Roles

First, it is important to explain the application user roles. That is, what users are expected to use DoMeet application and for what purpose. So, the application distinguishes between the following roles:

- An **unregistered user** is any person who access the application but not have the necessary credentials for log in into the application.
- A **registered user** or **user** is any person with the right to access the application.  
 A user has not specific roles for the application in general, but they have differentiated roles for a specific meeting:
  - A **meeting leader** is the user who organize a meeting.
  - A **meeting attendance** is a user that has been invited to attend a meeting.

### 4.6. Components distribution

To reach a better classification of functionalities and a better organization of the final application, it has been decided to split the application in components:

- The **SystemAdministration** component deals with administration functionalities, such us user management, complaints and data maintenance.
- The **MeetingAdministration** component embraces the functions of creating a new meeting instance, meeting scheduling configuration.
- The **MeetingExecution** component covers all functionalities during the meeting execution: timing, attendance, participation, etc. It also includes generating the documentation.
- The **MeetingAnalysis** components includes the reporting and analysis functions.

## 4.7. Use cases

For a better approximation of the requirements to the final functionality, the selection of Accepted requirements have turned into use cases (UC) as it can be seen in Table 5.

Table 5: Use cases

ID	DESCRIPTION	REQUIREMENTS
UC_1	User registration	R001
UC_2	User login	R002
UC_3	User logout	R003
UC_4	Update personal data	R004
UC_5	Create new meeting	R007
UC_6	Update meeting data	R008
UC_7	Cancel a meeting	R009
UC_8	Invite users	R010
UC_9	Check working hours overlapping	R017
UC_10	Handle meeting subjects	R018, R019
UC_11	Check subjects fit	R020
UC_12	Meeting invitation mailing	R021
UC_13	Response to meeting invitation	R023
UC_14	Start a meeting	R024
UC_15	Start a subject	R024
UC_16	Stop meeting	R024
UC_17	Join a meeting	R025
UC_18	Show meeting progression	R027
UC_19	Fill decision taken list	R028
UC_20	Fill agreements reached list	R029
UC_21	Fill pending issues list	R030
UC_22	Fill meeting comments	R031
UC_23	Generate the Minutes of the Meeting	R032
UC_24	Sent Minutes of the Meeting	R033
UC_25	Rate a meeting	R034
UC_26	Show a list of meetings	R035
UC_27	Show meetings information	R036, R037

### Use cases diagram

The Figure 6 shows the use cases mentioned before divided by components and associated to the application user roles:

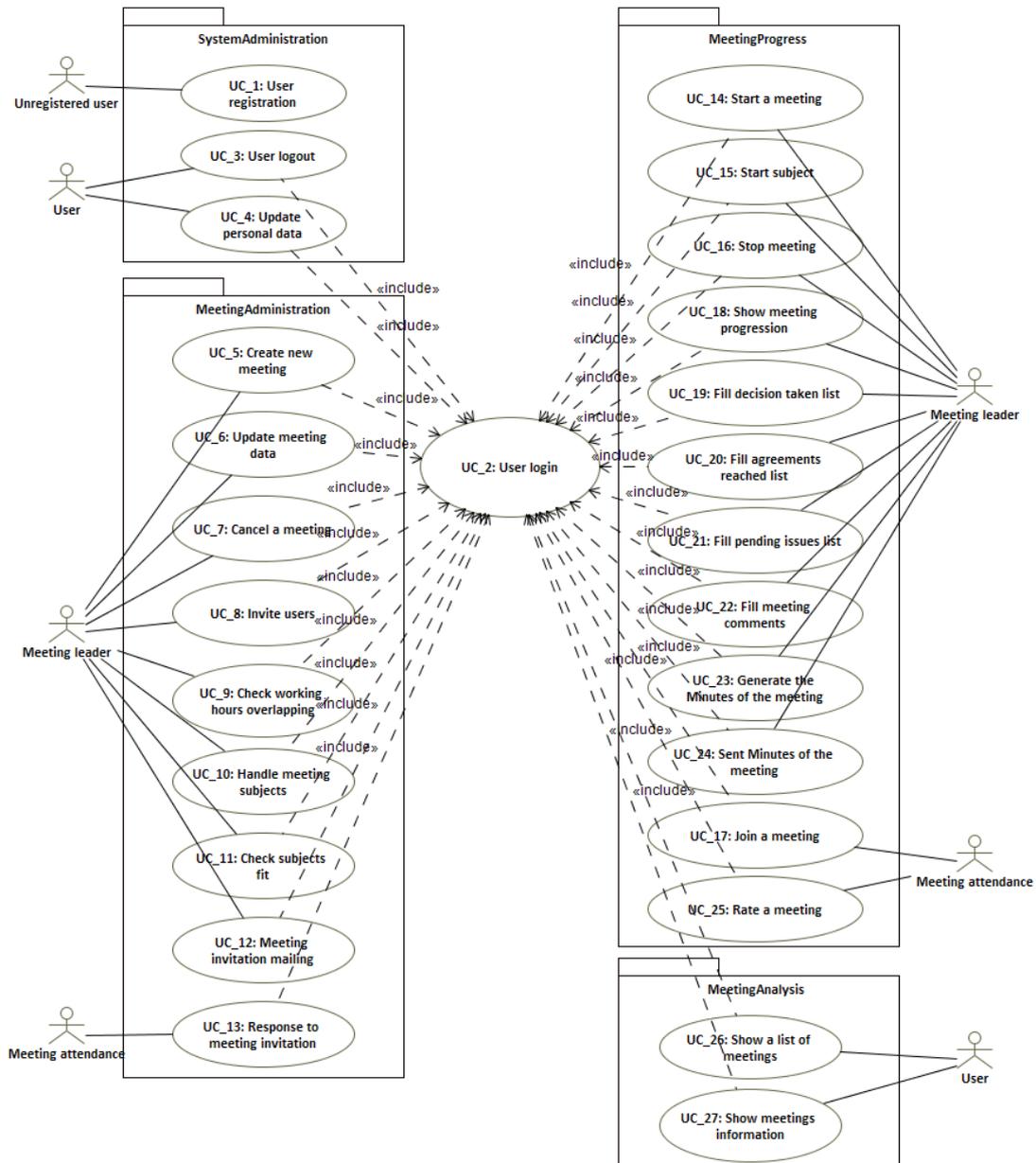


Figure 6: DoMeet use cases

## 4.8. Legal and ethical considerations

The DoMeet is offered as an application distribution ready to be put into an external server and join it with a database. The entity or person who prepare the mentioned environment and download and set up DoMeet is called “Administrator”. On the other hand, any person who access the application will be called a “User”.

### **Security Policy**

As the application does not provide any server service, but they must be provided and handled by the Administrator. For that reason, it is his/her responsibility to take care about network security, system scalability, etc.

Compatible application servers must allow secure HTTP requests (HTTPS). So, data encryption can be implemented on application communications.

### **Private Policy**

As the database is not embedded in the application, but they must be provided and handled by the Administrator. For that reason, it is his/her responsibility to take care of the application data: backups, confidentiality, etc.

Furthermore, the confidentiality level of the application information will extremely depend on each installation (kind of data, country, etc.). For that reason, is responsibility of the Administrator to guarantee the rights of its users and protect the database and network communications according to the legislation of the country where it operates.

It has bee

The information DoMeet receives in their different services is processed with maximum confidentiality and security. Furthermore, such information is used exclusively by DoMeet for the purposes associated with each service and will not be disclosed to any third parties unless DoMeet is obliged to do so for it being public information.

### **Intellectual property**

Users may solely and exclusively use DoMeet for their personal and private use. It is forbidden to use it for commercial purposes or in illegal activities.

As an open source application, any person will have access to the un-obfuscated code for reading or even modifying it.

All the information stored in the database is responsible of the Administrator, so it is also his/her responsibility to guarantee that this information not break any copyright law.

## 5. Product design

Until now it has been explained that the project consists on developing a product in form of a web application for providing Meeting-as-a-Service. The following section tries to explain how to implement the mentioned application. That is, all the technical information: requirements, data model design, architecture, etc. necessary for developing the application.

After reading this block, the reader will obtain the necessary knowledge for developing a kind of DoMeet application by his/her own means. Anyway, the intention of this block is not giving a recipe of how to implement a meeting-as-a-service application but offer a deep knowledge about how this kind as application works.

### 5.1. Physical model design

Practically, any application requires data handling. The complexity of administrate the application data depends on the amount and nature of this data. So, the administration of data can be from a simple task, handled by an unexperienced person, to a much more complex project, which involves a group of experts or even entire organizations. This subsection describes what data the application needs, and how to manage and organize this data.

#### **Data-management type**

It exists many ways for handling data, but the major determining factors are:

- Persistence
- Integrity
- Confidentiality
- Model structure

In terms on persistence, the user need to have access to all the data generated by itself or his/her colleagues; those data recently generated and those generated in the past.

Referring to data integrity, every data generated by the user or his/her colleagues must be available by the time its generated. On the contrary, it could create confusion among the users, so the application will not accomplish its purpose.

In relation with data confidentiality, it highly depends on the nature of the organization which uses the application. For that reason, it has been decided to delegate the management of the database to the person or organization who installs the application and not to the application itself.

Finally, referring to the data model structure it has been decided to use the entity-relationship model. It is not the most optimum model in terms of space and processing time, but is the easiest to understand, so to create and manipulate. This simplification is motivated by the fact that it allows the designer to organize the data model as it is in real life; which makes it the most extended used. The little number of entities and the simplicity of the model makes this model most suitable for the application, versus non-relational models.

### Entity-relationship model

The mentioned simplicity and real-life like model proportioned by the entity-relationship model is clear in the UML diagram in Figure 7, which shows the complete data model of the application DoMeet.

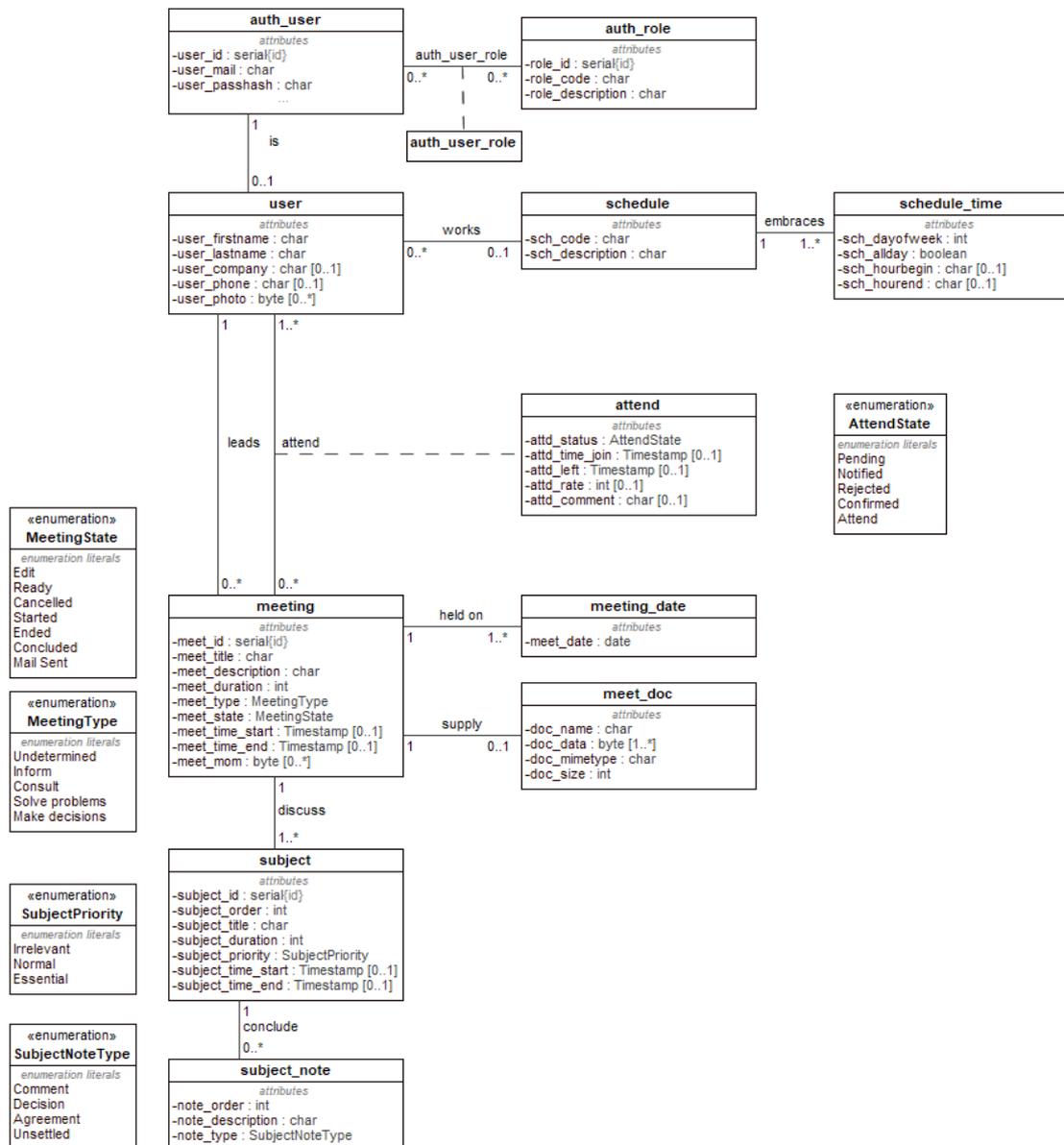


Figure 7: DoMeet data model

## Database-management system

The defined data requirements lead to choose a database-management system (DBMS) for storing and handling the data. It could be decided to use other types of storage systems such a file system, a unique file, a cloud storage, etc. but a DBMS is a computer-software application with a vast number of functionalities for data manipulation, querying and communication with external applications.

It exists many available DBMS's applications, the one selected to be used with DoMeet application is PostgreSQL database. The motivations for choosing this one among all the available are exposed in the following subsection.

## 5.2. Software architecture design

Another important factor when developing an application is defining the software architecture. That is, the selection of software elements for implementing the application and the relations among them.

Among all the architecture definition, the selection of the application layers is the most determinant, because it conditions the whole application. First, it has been decided to divide DoMeet application in the following tiers or layers:

- Enterprise information systems tier
- Business logic tier
- Presentation logic tier
- Client tier

The graph of Figure 8 shows the technologies used by DoMeet application and the communication between layers. A brief explanation of each tier and what technologies has been decided to use on each is presented below.

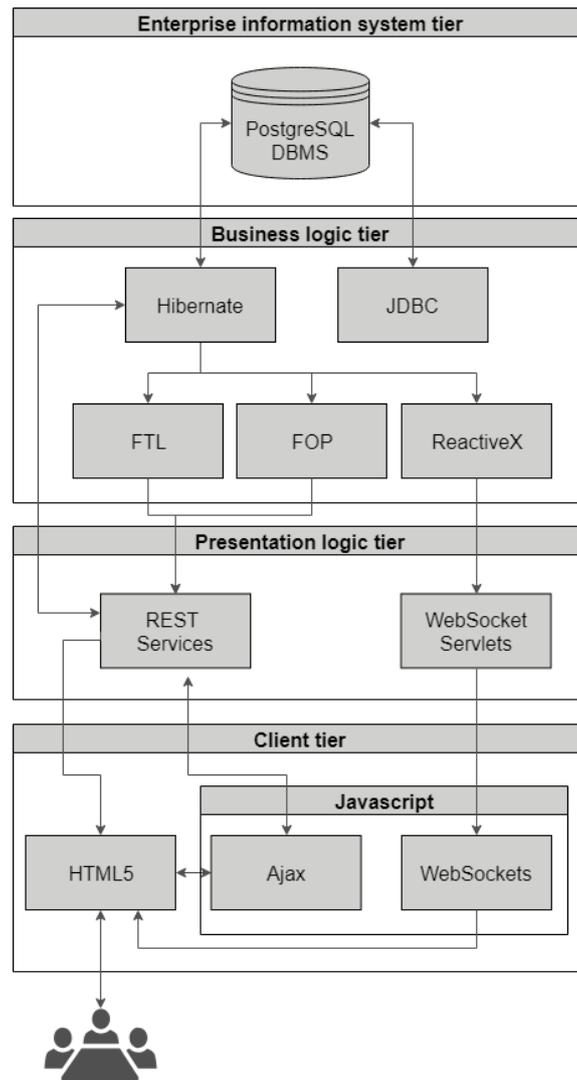


Figure 8: DoMeet tier architecture

### Enterprise information systems tier

The current tier encompasses all enterprise backend resources, such as databases, BackOffice/legacy systems and ERP implementations. In the case of DoMeet application only consists on the database-management service DBMS.

As it is previously said, DoMeet application database-management service has the following requirements:

- Application data requirements:
  - Data persistence
  - Data integrity
  - Relational database
- Architecture requirement:
  - Must has available Drivers to connect it to a Java application
  - Must be compatible with hibernate and JPA
- Project specific requirements:
  - Must be free

**Review:**

See 5.1 Physical model design section.

In spite of those requirements, it still existing many valid database-management services: MySQL, PostgreSQL, SQLite, OpenLink, etc. Anyway, it has been selected PostgreSQL among all of them. There is not a compelling reason for choosing it, but it is relatively simple to install and set up, have a very good documentation and accomplishes the requirements of the project.

### Business logic tier

This tier handles processes for manipulating enterprise information and handles communication with the presentation tier. It involves some technologies which are remarkable to mention:

- **JDBC:** It is used for database configuration and status check.
- **Hibernate:** is used for enterprise information manipulation and memory storage.
- **FreeMaker templates (FTL):** is used for generating output code from a data source via templates, such as HTML pages or PDFs. It has been used FTL instead of JSP because is much more versatile and can be used for either generate an HTML page, and Email or a FOP.
- **Apache FOP:** is used for generating PDF documents from the Enterprise information.
- **ReactiveX:** is used to observe application changes and notify them to the presentation logic tier to achieve a reactive environment.

**JDBC**

<http://www.oracle.com/technetwork/java/javase/jdbc/index.html>

**Hibernate [15]**

<http://hibernate.org/>

**FreeMaker Java Api [5]**

<https://freemarker.apache.org/>

**Apache FOP [6]**

<https://xmlgraphics.apache.org/fop/>

**ReactiveX [4]**

<http://reactivex.io/>

## Presentation logic tier

The presentation logic tier handles requests and responses between the client and business logic layers, that is, the communications between the clients and the server.

This part is mainly composed by the web server, a software element that can serve contents to the World Wide Web (www). It exists many available web servers, although all of them have as an objective serving content to the www. They differentiate from each other in many aspects; the most relevant are the programming language they use, and how are deployed and run. For the current project the web server must implement the following requirements:

- Application requirements:
  - Use **Java** as programming language and implement **J2EE**.
- Architecture requirement:
  - Implement **restful services**, which allows on demand requests from the clients to the server using the HTTP and REST protocols.
  - **WebSocket servlets**, which allow establishing bidirectional communication sockets between the clients and the server using the HTTP protocol.
- Project specific requirements:
  - Must be free

It still existing many web servers which accomplish these requirements: Apache Tomcat, Glassfish, Jetty, Oracle HTTP Server, etc. Among all of them it has been chosen Jetty web server because, a part of accomplishing the requirements of the project, is up to date with the latest technologies. Furthermore, common application servers initialize the Java application, whilst with Jetty is the Java application which starts the application server. This factor gives to the application more control over the server, which is pretty profitable.

## Client tier

Finally, client tier is defined by the user interface. That is, a software which enables the necessary elements to present the information to the user and collect the user interactions: text, images, graphics, buttons, forms, etc.

It exists many options to implement user interfaces: Text-based, Command line, Touchscreens, etc. Depending on the used technologies and the type of application one or other option fits better. In that case, DoMeet application requires a graphical environment, which allow presenting and collecting information in an intuitive way; besides, the interface must be available from the most devices and platforms.

Taking these facts in consideration, it has been decided to use a web user interface (WUI), that accept input and provide output by generating web pages viewed by the user through a web browser. The point of web browsers is that a unique implementation of web pages can run over different devices and platforms. Even though, web technologies evolve very fast and web browsers rapidly becomes obsolete. The following table is not an exhaustive list of compatible browsers, but the main ones:

Chrome	Firefox	Edge	IE	Safari
Supported (Recomended)	Supported	Supported	11+	9+

Web browsers are defined by three technologies or languages:

- **HTML** to define the content of web pages
- **CSS** to specify the layout of the pages
- **Javascript** to program the behaviour of web pages

The purpose of web browsers is to read HTML documents and display them, so HTML is the used language to build web pages. **HTML5** is an evolution of HTML which increases user interaction and provide new functionalities.

**CSS** is a language that describes the style of an HTML document so, how HTML elements should be displayed. An important directive given by CSS is: One HTML Page – Multiple Styles! It stands for the fact that a same HTML Page can have completely different style just by changing the CSS. This enforces the tendency of split the style programming from the functionalities.

**Javascript** is used for programming the behaviour of the page. It enables uncountable possibilities and functionalities which are outside the scope of this article. The most significant functionalities of Javascript used in DoMeet application are modifying the HTML elements and communication with the Presentation tier. On one hand, to modify HTML elements DoMeet application have used the frameworks **Vue** and **Vuetify**, which provides many components, views and layouts for creating a web application. On the other hand, to communicate with the presentation tier Javascript provides **Ajax** and **WebSockets**. The former allows communication via HTTP requests and responses, whilst the latter allows stablishing a persistence socket between the two layers.

**Vue** [11]  
<https://vuejs.org/>

**Vuetify** [12]  
<https://www.skillsyouneed.com/ips/meetings.html>

### 5.3.Application deployment

Another important aspect about the DoMeet application software architecture is how to deploy the application. Application deployment is the way the source code and configuration turns into an executable application. The main aspects of DoMeet deployment are:

- Create and verify enterprise database
- Compile Java classes
- Compile Javascript application
- Run web server

To perform the deployment tasks it is used Gradle, an build automation tool. Thanks to gradle all the required tasks mentioned before can be executed with a single command.

**Gradle** [9]  
<https://gradle.org/>

## 6. The Application

The following section shows the application functionalities and main tricks. The description below does not pretend to be an exhaustive manual with the description of every single functionality but exposing the most interesting functionalities of the application. As the application is designed to be very intuitive and user friendly, it is recommended to explore and test it to get a deeply idea of what it offers.

### 6.1. Sign in application

The first step the user must do when accessing the application is to create a user. To do so, the initial page supplies a link to the register form, where the user can provide an email account, as user ID, and a password, as it can be seen in Figure 9.

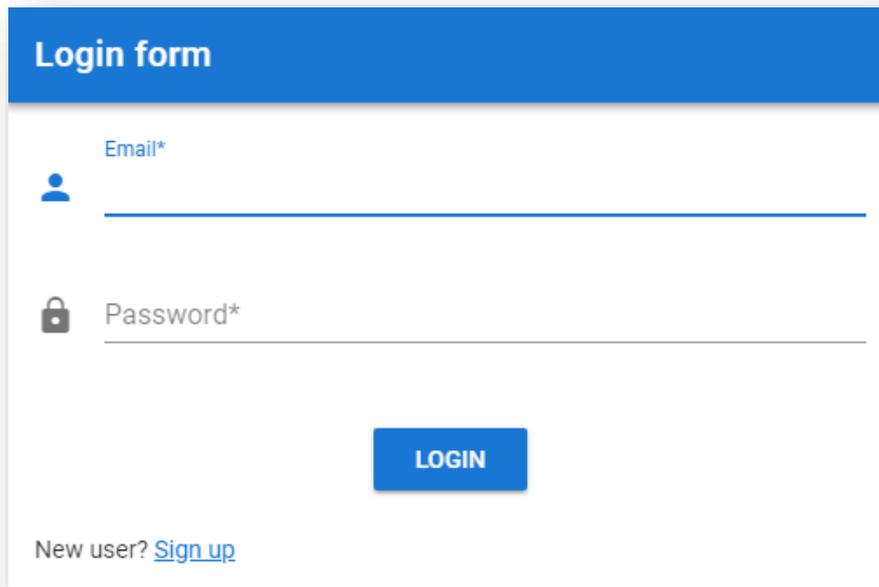
terms and conditions'." data-bbox="158 350 790 716"/>

The screenshot shows a mobile application interface for signing up. At the top, there is a blue header bar with a white back arrow on the left and the text "Sign Up" in white. Below the header, the form consists of three input fields stacked vertically. The first field is labeled "Email\*" and has a blue person icon to its left. The second field is labeled "Password\*" and has a grey lock icon to its left. The third field is labeled "Confirm password\*" and also has a grey lock icon to its left. Below these fields is a blue button with the text "SIGN UP" in white. At the bottom of the form, there is a line of text: "By signing up you agree to the [terms and conditions](#)".

Figure 9: Screenshot of DoMeet sign up form

### 6.2. Application Log in

Once having a user, it is possible to log in the application. As it is shown in Figure 10, just by accessing the initial page and putting the user ID (email) and password.



The screenshot shows a login form with a blue header containing the text "Login form". Below the header, there are two input fields. The first field is labeled "Email\*" and has a person icon to its left. The second field is labeled "Password\*" and has a lock icon to its left. Below these fields is a blue button with the text "LOGIN". At the bottom left of the form, there is a link that says "New user? Sign up".

Figure 10: Screenshot of DoMeet login form

### 6.3. Preparing a meeting

As a meeting leader, the first step when using the application is creating a meeting. After creating a meeting, the application provides a series of fields actions to define the meeting.

Adding attendees to a meeting is performed via a searching field that allows searching by user name, surname, user email or company. The performed search executes a fuzzy query, so the typed references accepts a margin of error for which the application will present the correct users.

Apart from the defined fields and actions, it is very important to remark the notifications panel, as it can be seen in Figure 11, at the right of the figure. This panel provides specific advice about the outgoing meeting. Advices are shown in form of cards, which have a different colour depending on their relevance.

Figure 11: Screenshot of DoMeet meeting register form

### Receiving meeting notifications

As meeting attendees, the invites will receive an email with some information about the meeting and an URL to accept or decline the invitation. Even they accept or not, the attendees will have the meeting available inside the application, so they can consult its evolution and conclusions.

## 6.4. Leading a meeting

When the meeting date arrives, the leader of the meeting can start the meeting. After the meeting starts, the attendees can join the meeting by itself, or the leader can add them to the meeting. As the meeting is composed by one or more topics, the meeting leader go by the different topics at the desired pace.

For each topic, the leader can annotate comments, agreements, decisions taken and pending issue. At the end of the meeting the leader can rewrite or add whatever necessary. After all the topics are done, the meeting leader must end the meeting.

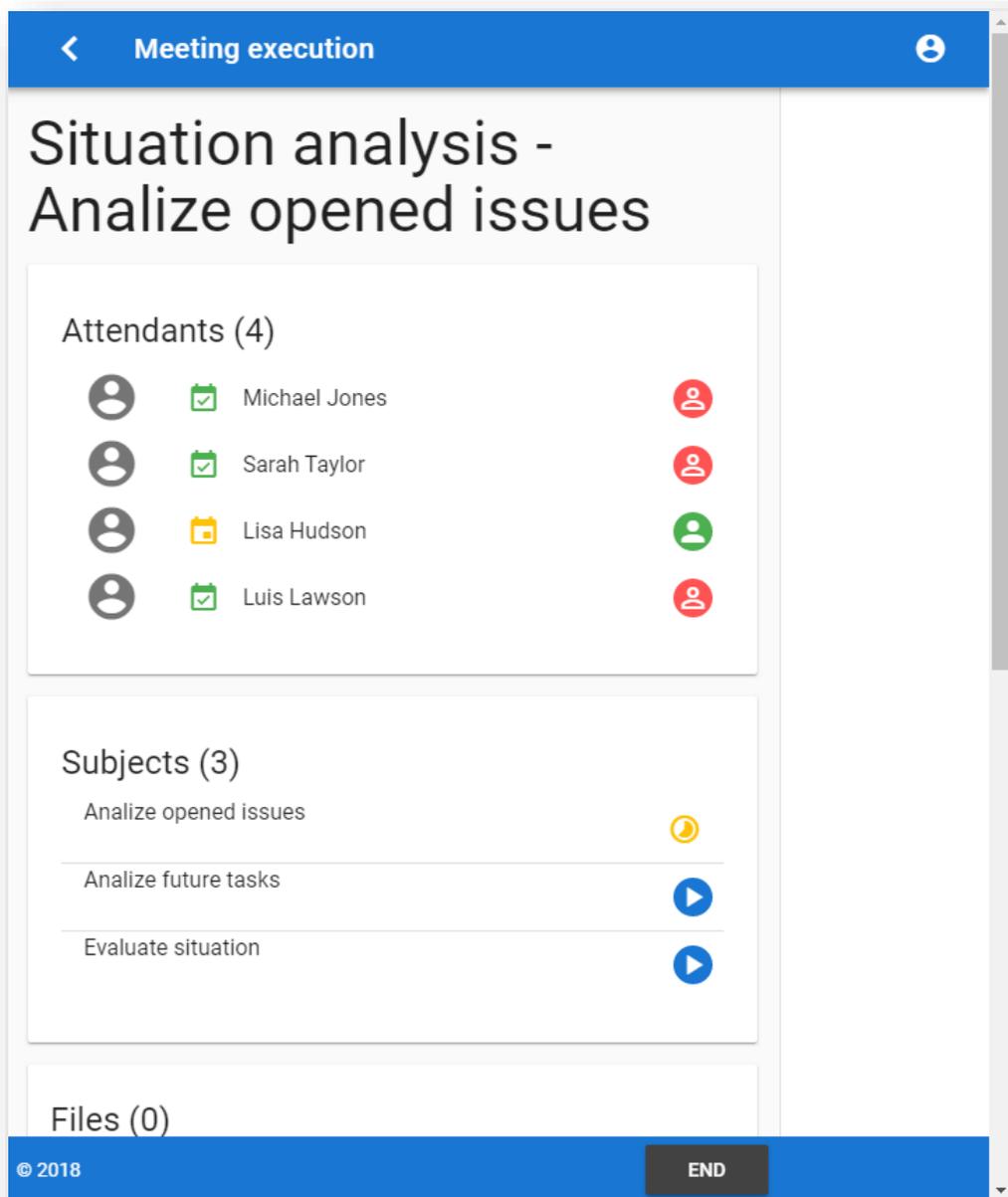


Figure 12: Screenshot of DoMeet meeting execution

## Attending a meeting

As an attendee, the user can join to a started meeting and visualize the progress of the topics and the annotations make it.

### 6.5. Meeting conclusion

After the meeting ends, the meeting leader can review everything annotated during the meeting and remove, add or rewrite whatever needed. When the leader decides that everything is ok, can conclude the meeting and send an email to the invited user with the Minutes of the Meeting in PDF format, automatically generated from the meeting data and annotations.

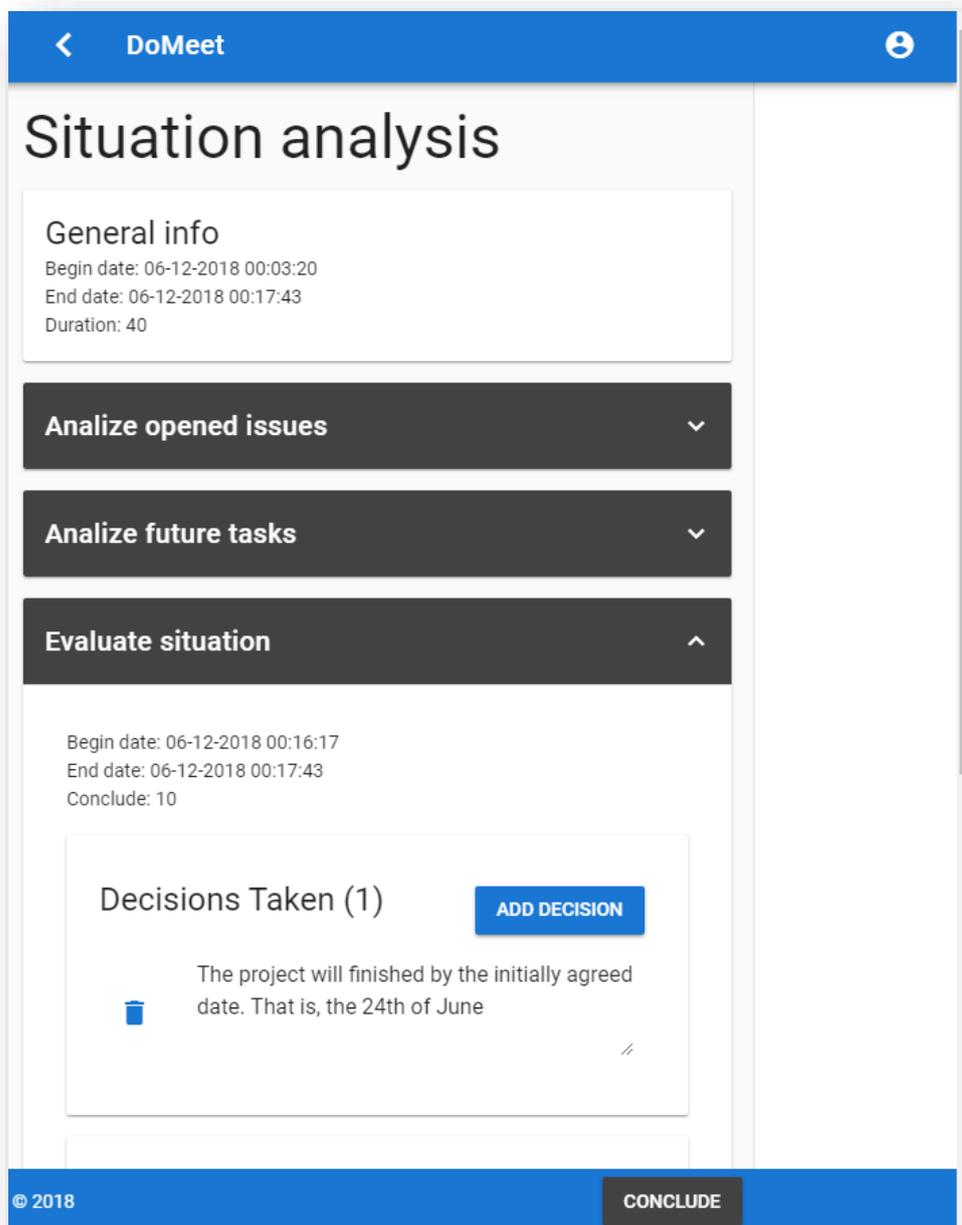


Figure 13: Screenshot of DoMeet meeting conclusion

## 6.6. Meeting review and rating

Both meeting leader and attendees will have all their meetings available to consult at any time. Apart from consult the meeting information through the application they can download the Minutes of the Meeting.

A sample of the Minutes of the Meeting in PDF format can be seen in Figure 15.

Besides, the meeting attendees can give their opinion about the meeting executing by letting a comment and a numeric valuation. This rating and comments, along with meeting metadata, can help on internal evaluation to take conclusions about meetings executions and help on improving future meetings.

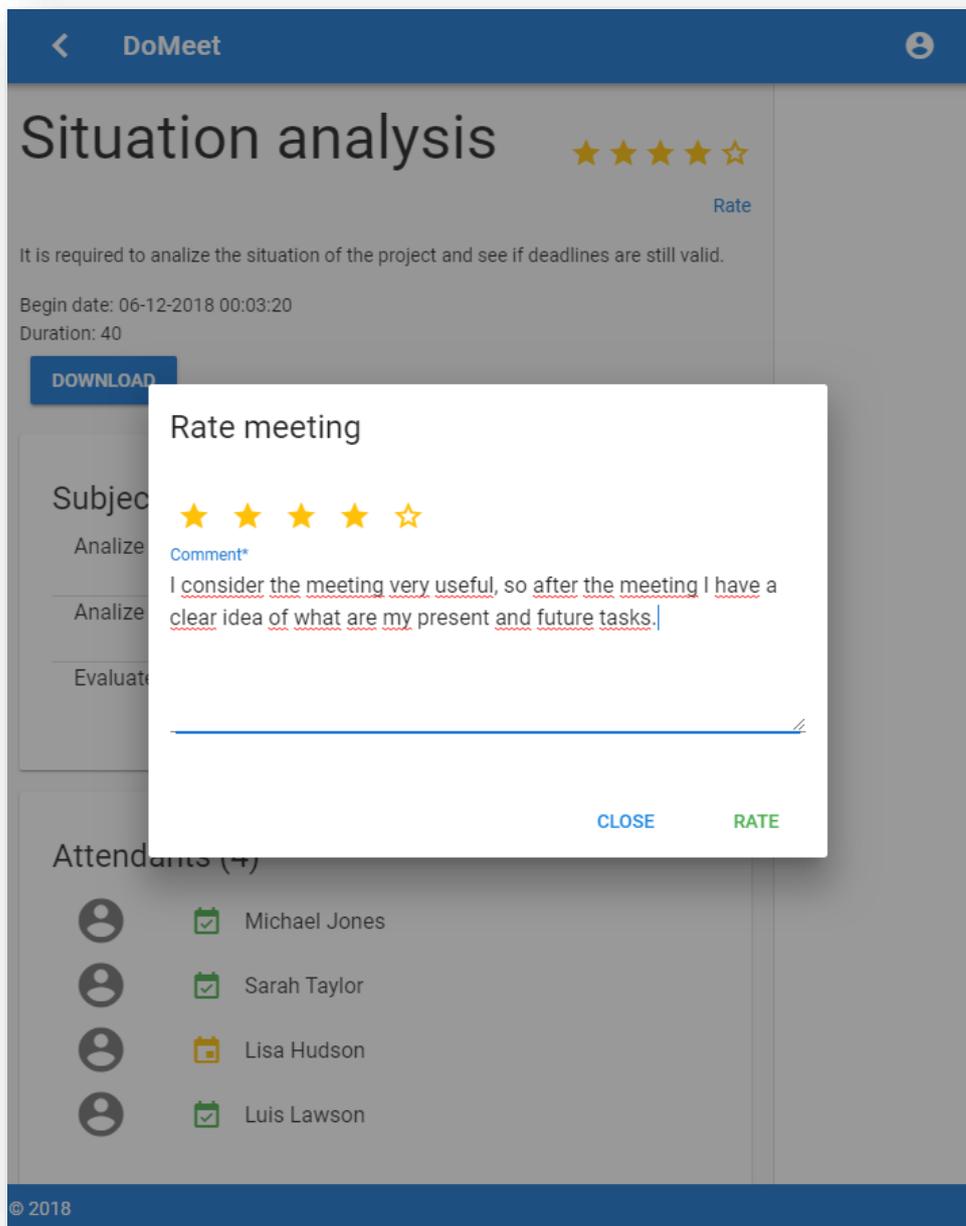


Figure 14: Screenshot of DoMeet meeting rating

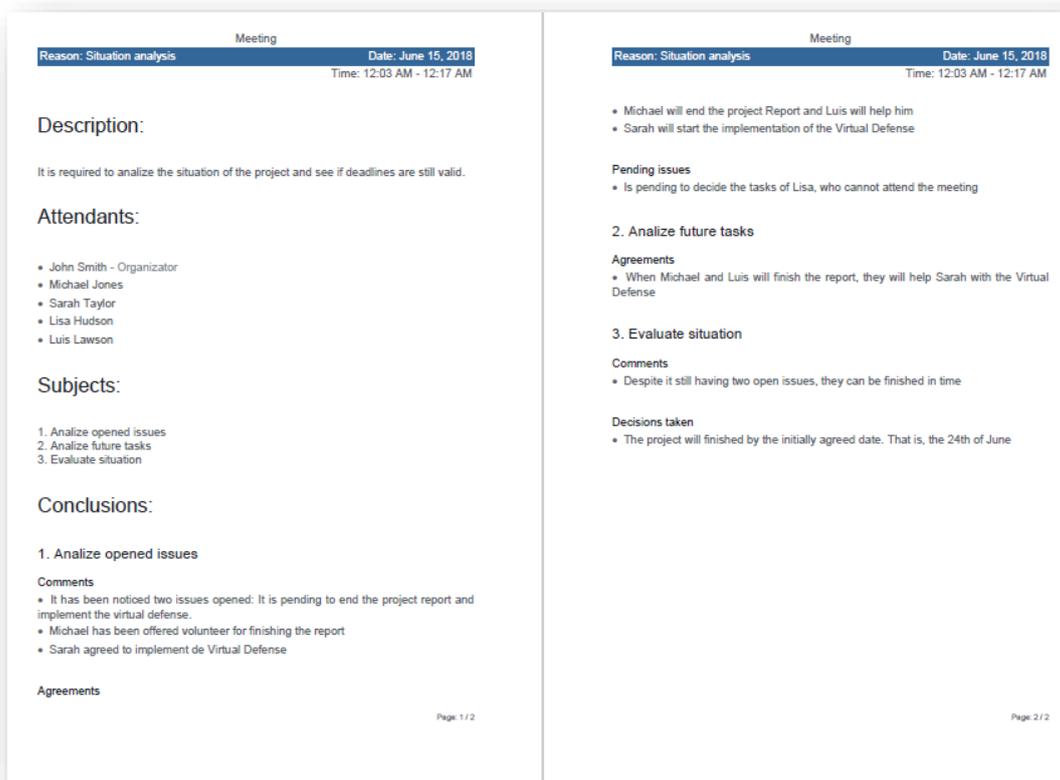


Figure 15: Screenshot of the Minutes of the Meeting PDF

## 6.7. Other functionalities

It has been exposed the main functionalities of DoMeet application; that is, those functionalities that makes possible Meeting-as-a-Service. But it exists some other internal properties or functionalities that makes the application extremely awesome and are worthy to comment.

WebSocket servlets, which allow stablishing bidirectional communication sockets between the clients and the server using the HTTP protocol.

### Automatic data refresh

As DoMeet application is conceived to be executed by team works, is very important that every member of the team has an updated view of the meeting data as soon as this data changes.

ReactiveX: is used to observe application changes and notify them to the presentation logic tier to achieve a reactive environment.

Thanks to the use of WebSockets and Java ReactiveX, each client browser stablishes a persistent communication with the server that notify and refresh meeting changes immediately.

### Multilanguage

Another important functionality is that the application is multilanguage. As the application is executed via a web browser, it takes the language of the operative system as application language. By now, it is only implemented in Catalan, English and Spanish. In case of the user language is not any of the mentioned before, the application will use the English language.

**Personal data**

Finally, it is worthy to say that the users can introduce some personal data, such as company name, telephone number or their picture in order to improve their identification within the application.

Finally, it is very recommended to try the application under the following URL:

<http://ec2-52-47-183-22.eu-west-3.compute.amazonaws.com:8081>

## 7. Economic valuation

After describing the project, is interesting to comment what benefits can generate the resulting product. Despite the project has not been implemented to generate benefits directly, as the product has been designed as an Open Source application, it will be seen that the derived services the product offers are suitable to be used for getting profit.

Furthermore, this section gathers all the human and material costs of implementing the project. As an academical project those cost has been at kept near to zero, essentially by using Student licences and Open Source applications.

### 7.1. Cost

Regarding to economic expenses, it must be considered that the project is an academic work or, in case it finally arrives to the real-world, it will do as an open source application. For that reasons the cost of the project budget must be very low or zero.

On one hand, the project is developed by a single person, the student, which won't receive any economic reward for the job. Furthermore, the role of accepting and validating the project will be performed by the professor or the collaborating professor, which won't neither receive extra payment for this task. Taking this into account, the Table 6 shows the summary of the human resources costs.

Table 6: Resources costs

<i>Resource type</i>	<i>Name</i>	<i>Cost/hour</i>	<i>Description</i>
<i>Developer</i>	Marc Estévez	0.00 €	The person in charge of carrying out with the whole project development. Including analysis, implementation, testing and documentation.
<i>Collaborating professor</i>	Fatos Xhafa	0.00 €	His tasks are validating and accepting the project and giving technical or executing advice.
<i>Professor responsible of the subject</i>	Atanasi Daradoumis Haralabus	0.00 €	His tasks are validating and accepting the project and giving technical or executing advice.

On the other hand, it must be considered to use open source products or use student licenses for the required software and services to minimize the expenses of project development and maintaining as it is shown in Table 7.

Table 7: Project and maintaining expenses

<i>Resource type</i>	<i>Name</i>	<i>Cost</i>	<i>License</i>	<i>Description</i>
<i>Programming IDE</i>	IntelliJ	0.00 €	Student license	Software used for writing the code.
<i>Version control system (VCS)</i>	Git hub	0.00 €	Public repository	Service use for storing the programming code in the cloud; which provides backup and version control.
<i>Database</i>	PostgreSQL	0.00 €	GPL	Relational database for storing the application information.
<i>Hosting</i>	Pending to decide	0.00 €	???	Service for making the application accessible in the internet.
<i>Project manager</i>	Microsoft Project	0.00 €	Student license	Software useful for project planification, including task definition and resources,
<i>Programming language</i>	Java	0.00 €	BCL	General purpose programming language used for writing the application code.
<i>Build automation tool</i>	Gradle	0.00 €	Apache License 2.0	Software used for handling application libraries and deploying process.
<i>Server application</i>	Jetty	0.00 €	Apache License 2.0 and Eclipse Public License 1.0	Application server which enables web applications with REST and WebSockets capabilities.
<i>Video editor</i>	Camtasia	0.00 €	Trial version	Software for create and editing videos.
<i>Document editor</i>	Microsoft Word	0.00 €	Student license	Software for create and edition documents.

## 7.2. Profit

The project has been developed with cost zero, because it is not intended to generate any economic profit. Anyway, the application is ready to be deployed in a public web server; so, anyone who wanted to offer meeting-as-service just have to configure a web server, a database and deploy the application in it. As DoMeet application is OpenSource, they could charge to their users for the service and get profit from the difference between how much they charge per user and the costs derived of maintaining the server.

## 8. Conclusions

The following lines try to summarize the thoughts and feelings about the project. What level of achievement has reached, what lessons has been learned, which have been its weaknesses and strengths, and what does the project let pending to do.

### 8.1. Lessons learned

As the project consisted in developing a web application for implementing meeting-as-a-service, it presents four differentiated parts: planification, development, test and documentation.

In first place, planification is quite laborious, it requires complex work of analysis, control and anticipation but it also the most predictable. So, it can be done step by step, trying to consider everything relevant for the project but with little fear of overdoing the deadlines. The greatest difficulty in this phase have been to predict and schedule the project subtasks, due to most of those tasks are very difficult to estimate how much time requires carrying them out.

In second place, development phase is the antithesis of planification. Due to, while developing an application which involves many technologies, such as DoMeet, constantly appear issues, crisis and questions, and with each one the deadline approaches more and more. Despite of having a good planification, what helps on identifying the major risks and allows giving them special attention, unexpected problems can delay the planed schedule. For the current project, the most remarkable technical issues during development has been: Managing beans with Hibernate, HTTP Error handling and character encoding.

In third place, testing is less conflictive than development, as long as the planification reflected properly the requirements of the product. In that phase, it is revised that the application does not produce errors, is user friendly and accomplish the initial expectances.

In fourth place, the documentation phase has two purposes: write a description of the product and analyse the development and results of the project. As it occurs with the planification phase, this is a very predictable stage and the major worry is to verify if the project has succeeded, as it is exposed on the following subsection.

## 8.2. Objectives achievement

The proposed objectives and main goals of the project were:

- Increase efficiency of workgroup meetings
- Reduce administration time
- Improve workgroup evaluation
- Improve internal communication

On one hand, increasing the efficiency of workgroups meetings have been tried to achieve by giving the application the capacity of guiding the meeting leader in the processes of meeting preparation and execution. Specifically, DoMeet application provides guidance for choosing a specific meeting type, prevents from introducing incoherencies on meeting preparation and gives some advices about meetings how to. Besides, it provides a linear review of topics during the meeting execution, it also gives advices during this phase and enables user-friendly forms for collecting meeting inputs. These tools cover the objective in question in great measure; anyway, more and better advices could be provided in order to increase workgroups efficiency. For it, the application source code has been designed to allow adding more advices with easy.

On the other hand, reducing administration time has been resolved by centralizing meetings information in a single application and automatize the generation of the documentation. So, the application tries to provide all necessary functionalities for setting up a meeting: meeting definition, meeting notifications, gather documentation, collect meeting inputs and generate and store the Minutes of the Meeting. With these functionalities, the objective is also covered in great measure, although the objective could be significantly improved by providing a mechanism for agree the date of the meeting and allow Minutes of Meeting customization, such as adding a company logo or name.

Improve workgroup evaluation has been carried out by providing a meeting voting system, with the capacity of letting comments. The intention of this rating is giving the leader a feedback about the meeting. With both ratings and statistics generated from meetings (duration, number of topics, attendees, etc.) it is possible to extract worthy conclusions about what are the best way of doing meetings. Despite of this information allow extracting valuable knowledge, it would be awesome that the application would *cook* the information and would present the conclusions automatically in form of text or graphs.

Finally, the goal of improving internal communication has been the least resolved. The major communication through the application is in one direction, from the leader to the attendees, even though the rating system allow a certain level of

### Review:

Review the initially defined project objectives on: 1.2. Project objectives.

communication backward. The reason for giving less importance to this goal is conceiving a meeting to be performed face to face; so, the responsibility of annotating everything commented within the meeting is on the leader or him/her secretary. It may be an old fashion way, but it still the most extended nowadays and the model proposed to cover by this project.

### 8.3. Planification and methodology analysis

Regarding to the planned schedule everything has gone very similar as expected. The main troubles occurred during the development phase, where some important application functionalities were not ready by the defined deadline, and they were finished a week after. Even the application was functional, it was not give advices during the meeting creation and the Minutes of the Meeting had a poor style. Once these issues were overcome, the project keep on without troubles. The key for minimizing possible planification troubles was setting the major risk tasks to very early phases: configuring web server and database, deciding and test models with Hibernate and setting the client-side framework up.

As an example of applying the iterative software life-cycle methodology, it can be noticed that the defined data model initially contemplated meeting comments attached directly to the meeting, but in the validation phase with the consultant department of RICOH SPAIN IT SERVICES SL, it figured out that it would be better to attach the comments per topic. So, it was required a new lifecycle for adding this change.

Despite of applying the mentioned life-cycle iterations for major changes, the development of DoMeet has required three entire life-cycles, small features and bug fixes have been included bypassing the methodology. As an example, is worthy to mention adding the meeting state Mail Sent or adding a block on the home page for the drafts meetings.

### 8.4. Plans for the future

The project has been tackled as a full application for providing meetings-as-a-service. The concept is very wide, but because of the limitations of time and resources only a few functionalities have been able to be implemented among all possible. So, the main improvement that can be provided in short term is adding some more functionalities. The most remarkable are: meeting date agreement, intensify the role of meeting attendees and allow customization of the Minutes of the Meeting.

One hand, with the current implementation the user needs to used external ways of communication for agree a date for the

#### Review:

Review project methodology on:  
1.3. Methodology and approach

meeting. The intention is to implement some tools during the meeting preparation phase for providing this functionality within DoMeet application.

On the other hand, the application gives many functionalities for the meeting leader and very little for the attendees. The intention in future versions of the application is giving more importance to the role of attendee. This can be achieved by implementing a chat during the meeting execution phase, which allows to attendees giving their opinions and participating during the meeting, even if they are not attending the meeting in situ.

Finally, it would be great to allow some customization in the generated Minutes of the Meeting. That is, adding the company name and logo, choosing the font, etc. Even these functions has not much impact in the meetings by itself, some companies requires to include such kind of information into their internal documents by protocol, so the fact of not having them can make some companies not using the application.

## 9. Glossary

- \*aaS, 17
- Agenda, 14
- Agile software development, 4
- Ajax**, 36
- attendance**, 25
- Business logic tier, 32
- Client tier, 32
- Conclusion, 13
- Confidentiality, 30
- Consulting Meetings, 11
- CSS**, 36
- Database-management system, 32
- DBMS, 32
- Decision making meetings, 12
- DoMeet, 19
- EaaS, 17
- Enterprise information systems tier, 32
- Entity-relationship model, 31
- Everything-as-a-Service, 17
- Execution, 13
- FOP**, 34
- FreeMaker**, 34
- FTL**, 34
- Gantt diagram, 8
- gradle, 37
- Hibernate**, 34
- HTML**, 36
- Informing Meetings, 11
- Integrity, 30
- iterative life-cycle, 4
- J2EE.**, 35
- Java**, 35
- Javascript**, 36
- JDBC**, 34
- leader**, 25
- Meeting, 11
- Meeting-as-a-Service, 18
- Minutes of the Meeting, 15
- Model structure, 30
- MoM, 15
- OpenSource, 47
- Persistence, 30
- PostgreSQL, 32
- Preparation, 13
- Presentation logic tier, 32
- Problem solving meeting, 12
- Project tasks, 6
- ReactiveX**, 34, 44
- Requirements, 23
- restful**, 35
- Review, 13
- Roles, 25
- stakeholder, 5
- TaaS, 1, 17, 19
- Teamwork-as-a-Service, 1, 17, 19
- Use case, 26
- Vue**, 36
- Vuetify**, 36
- Waterfall model, 4
- WebSocket**, 35, 44
- WebSockets**, 36
- XaaS, 17

## 10. Index of figures

Figure 1: Software iterative life-cycle .....	4
Figure 2: Project stakeholders chart .....	5
Figure 3: Gantt diagram for project planning .....	8
Figure 4: Structure of a meeting .....	12
Figure 5: DoMeet architecture.....	19
Figure 6: DoMeet use cases .....	27
Figure 7: DoMeet data model .....	31
Figure 8: DoMeet tier architecture .....	33
Figure 9: Screenshot of DoMeet sign up form .....	38
Figure 10: Screenshot of DoMeet login form.....	39
Figure 11: Screenshot of DoMeet meeting register form.....	40
Figure 12: Screenshot of DoMeet meeting execution .....	41
Figure 13: Screenshot of DoMeet meeting conclusion.....	42
Figure 14: Screenshot of DoMeet meeting rating.....	43
Figure 15: Screenshot of the Minutes of the Meeting PDF.....	44

## 11. Index of tables

Table 1: Project tasks schedule	6
Table 2: Project schedule summary	7
Table 3: Description of the phases of a meeting	13
Table 4: Product requirements	23
Table 5: Use cases	26
Table 6: Resources costs	46
Table 7: Project and maintaining expenses	47

## 12. Bibliography

- [1] R. E. Freeman and D. L. Reed, *Stockholders and Stakeholders: A new perspective on Corporate Governance*, California: California Management Review, 1983, pp. 88-106.
- [2] Wikipedia, "Everything as a service," 23 September 2017. [Online]. Available: [https://simple.wikipedia.org/w/index.php?title=Everything\\_as\\_a\\_service&oldid=5799801..](https://simple.wikipedia.org/w/index.php?title=Everything_as_a_service&oldid=5799801..) [Accessed 27 March 2018].
- [3] Wikipedia contributors, "Minutes," 11 May 2018. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Minutes&oldid=840653946>. [Accessed 26 May 2018].
- [4] Java ReactiveX, "An API for asynchronous programming with observable streams," [Online]. Available: <http://reactivex.io/>. [Accessed 5 May 2018].
- [5] Apache Software Foundation, "FreeMaker Java Api," [Online]. Available: <https://freemarker.apache.org/>. [Accessed 13 May 2018].
- [6] The Apache™ FOP Project, "Apache FOP," [Online]. Available: <https://xmlgraphics.apache.org/fop/>. [Accessed 13 May 2018].
- [7] Oracle Technology Network, "Java SE Technologies - Database (JDBC)," [Online]. Available: <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>. [Accessed 13 May 2018].
- [8] The PostgreSQL Global Development Group, "POSTGRESQL: THE WORLD'S MOST ADVANCED OPEN SOURCE RELATIONAL DATABASE.," [Online]. Available: <https://www.postgresql.org/>. [Accessed 13 May 2018].
- [9] Gradle Inc, "Gradle Tutorials and Guides," 2018. [Online]. Available: <https://gradle.org/guides/>. [Accessed 13 May 2018].
- [10] The Eclipse Foundation, "Jetty," [Online]. Available: <https://www.eclipse.org/jetty/>. [Accessed 13 May 2018].
- [11] E. You, "Vue, The Progressive JavaScript Framework," [Online]. Available: <https://vuejs.org/>. [Accessed 13 May 2018].
- [12] Vuetify, LLC, "Vuetify," [Online]. Available: <https://vuejs.org/> from: <https://vuetifyjs.com/>. [Accessed 13 May 2018].
- [13] "Planning and Structuring Effective Meetings. Skills You Need," [Online]. Available: <https://www.skillsyouneed.com/ips/meetings.html>. [Accessed 1 May 2018].
- [14] G. H. M. Robert., *Robert's Rules of Order Newly Revised 11th*, Florida: Da Capo Press, A Member of the Perseus Books Group, 2011.
- [15] J. O. T. & F. R. Elliot, *Harnessing Hibernate*, Sebastopol: O'REILLY, 2011.

## 13. Annex

### 13.1. Installation guide

#### Requirements

DoMeet server can be installed on any operating system (Linux, Windows, Mac OS X,) since it is a multi-platform application. It is required to have some applications installed in order to run the server.

- Java 8 JDK (also called Java 1.8)
- NPM (v.5.6.0 or greater)
- PostgreSQL (v.9.6 or greater)

#### Get the source code

The application code can be get from:

<https://github.com/mestevez/domeet-server>

#### Configuration files

The application mainly needs to configure two items:

- Database config
- SMTP config

On one hand, the application requires connecting to a PostgreSQL database, so some database configuration is required. By default, it connects to a postgresQL server at localhost listening by the port 5432. And connects with the administrative user postgres with password 12345.

On the other hand, the application is intended to send emails, so it requires some STMP configuration to execute properly.

All this configuration can be defined in the **configuration file** *conf/domeet.properties*:

```
dbms_host = localhost
dbms_port = 5432
dbms_name = domeet
dbms_user = postgres
dbms_password = 12345
smtp_host = smtp.gmail.com
smtp_port = 587
smtp_user = domeet.uoc@gmail.com
smtp_password = 0q4prpb3n0xt4cs
http_port = 8081
```

#### Java 8 JDK

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

#### NPM

<https://www.npmjs.com/get-npm>

#### PostgreSQL [8]

<https://www.postgresql.org/>

For test purposes you can leave the SMTP configuration as default, so the domeet.uoc@gmail.com account has been created with this purpose.

### Start the application

After defined the mentioned configuration, open a terminal and go to the root directory of the project, and then type:

- On Linux or Mac OS X type:

```
./gradlew appRun
```

- On Windows type:

```
gradlew.bat appRun
```

After a few minutes, it should return something like:

```
<=====--> 91% EXECUTING [6m 7s]
> :appRun
  IDLE
```

Once the server is installed you can access it with the URL: <http://localhost:8081>

It is remarkable that gradle has imported the necessary libraries for the project and has created the database into the postgres server.

The application code contains a **README.MD** file with a more detailed and up to date information about application requirements and installation.

## 13.2. Important URL's

Source code distribution:

<https://github.com/mestevez/domeet-server>

Sample of deployed application:

<http://ec2-52-47-183-22.eu-west-3.compute.amazonaws.com:8081>

