



# FoTRRIS

Fostering a Transition towards Responsible Research and Innovation Systems

# Design and specs of the co-RRI web-based platform

## Deliverable D2.1

**Authors:** Susana Bautista<sup>1</sup>, Juan Pavón<sup>1</sup>, Marlon Cárdenas<sup>1</sup>, Javier Arroyo<sup>1</sup>

<sup>1</sup> *Universidad Complutense de Madrid, Spain*

<http://fotrris-h2020.eu/>



This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 665906

## Document Information

Grant Agreement #: 665906

Project Title: Fostering a Transition towards Responsible Research and Innovation Systems

Project Acronym: FoTERRIS

Project Start Date: 01 October 2015

Related work package: WP 2

Related task(s): Task 2.1

Lead Organisation: LGI

Submission date: 2018/01/16

Dissemination Level: PU

## History

Date	Submitted by	Reviewed by	Version (Notes)
14/11/2016	Susana Bautista	Juan Pavón	Creation of the document
16/11/2016	Susana Bautista	Eve Dallamaggiore	Change of template
15/12/2016	Susana Bautista	Eve Dallamaggiore	Feedback of Eve
21/04/2017	Susana Bautista	Javier Arroyo	Review of the State of the art
27/06/2017	Susana Bautista	Juan Pavón	Review new version
06/07/2017	Susana Bautista	Juan Pavón	Updated with new functionality of the platform
25/09/2017	Susana Bautista	Javier Arroyo	Updated of the State of the art
11/10/2017	Susana Bautista	Juan Pavón	Updated with details of the architecture of the platform
24/10/2017	Juan Pavón		Reorganisation of the structure of the document and addition of new sections for the architecture and design
20/11/2017	Susana Bautista	Juan Pavón	Review diagrams of design
13/12/2017	Susana Bautista	Juan Pavón	Complete software specification following the new structure
18/12/2017	Susana Bautista, Juan Pavón		Review of specifications and elaboration of conclusions
20/12/2017	Susana Bautista, Juan Pavón	Jelena Mazaj, Eve Dallamaggiore	Overall review of minor comments
22/12/2017	Susana Bautista, Juan Pavón	Susana Bautista, Juan Pavón	Full review of spell and grammar check, format, and resolution of last comments.
16/01/2018	Susana Bautista, Juan Pavón	Nele D'Haese	Final version. Review of used terminology and abbreviations.

## Table of contents

List of Tables .....	4
List of Figures .....	5
List of Acronyms.....	7
About the FoTERRIS project.....	8
1 Introduction .....	9
1.1 Use and Design for co-RRI.....	9
1.2 Vision and Scope .....	10
2 Collaborative platforms for RRI.....	11
2.1 State of the art .....	11
2.2 Conclusions from the review of the state of the art .....	16
3 Analysis of the FoTERRIS web-based platform .....	18
3.1 Requirements for supporting FoTERRIS co-RRI methodology .....	18
3.2 Use Cases of the platform.....	18
4 Design of the FoTERRIS web-based platform.....	25
4.1 Objectives and Restrictions of the System Architecture .....	25
4.2 Specification of the System Architecture .....	25
5 Validation and experimentation of the platform.....	37
5.1 Evaluation.....	37
5.2 Accessibility Check .....	47
5.3 Gender Check.....	49
6 Distribution and sustainability of the platform.....	50
6.1 GitHub .....	50
6.2 Installation of the platform.....	50
7 Conclusions .....	53
8 Bibliography .....	54
Appendix 1: Database schema .....	56

## List of Tables

<b>Table 1: Priority of Use Cases.....</b>	<b>24</b>
<b>Table 2: Versions of the main external components .....</b>	<b>36</b>
<b>Table 3: Summary of the guidelines at levels A and AA of accessibility.....</b>	<b>48</b>

## List of Figures

Figure 1: Types of actors on the platform.....	19
Figure 2: Use case diagram for the User actor .....	20
Figure 3: Use case diagram for the Registered User actor .....	21
Figure 4: Use case Diagram for a Hub Manager .....	22
Figure 5: Platform Admin use case diagram.....	23
Figure 6: Class diagram with the main entities of the Model.....	26
Figure 7: Sequence diagrams for management of a project (creation, deletion, update).....	27
Figure 8: Sequence diagram for the assignment of a Registered User to a Project .....	28
Figure 9: Sequence diagram to Login on the platform.....	29
Figure 10: Sequence diagram for a Registered User to join to a Project .....	29
Figure 11: Activity diagram to access the platform .....	30
Figure 12: Activity diagram of a user to join to a project.....	31
Figure 13: Activity diagram of create project process .....	32
Figure 14: Activity diagram to assign a user to a project .....	33
Figure 15: Participation in a pad .....	34
Figure 16: Package diagram of the application .....	35
Figure 17: Deployment diagram of the application .....	36
Figure 18: Gender of the participants in the evaluation (Blue: man; Red: woman; Orange: don't reply) ..	38
Figure 19: Age of the participants (Blue: between 18-20; Red: 20-25; Orange: 25-30; Green: More than 30; Purple: more than 25).....	38
Figure 20: Level of study of the participants (Blue: High School; Red: University; Orange: Master; Green: PhD) .....	38
Figure 21: Data of the evaluation of Home page (menus, logos, information) (Likert scale: value 1: not easy to use; value 5: very easy to use) .....	39
Figure 22: Data of the evaluation of the Registration process (Likert scale: value 1: not easy to use; value 5: very easy to use) .....	39
Figure 23: Data of the evaluation of the Login process (Likert scale: value 1: not easy to use; value 5: very easy to use) .....	40
Figure 24: Data of the evaluation of the Show project functionality (Likert scale: value 1: not easy to use; value 5: very easy to use) .....	40
Figure 25: Profile of manager of a project (Blue: yes; Red: no) .....	41
Figure 26: Functionality to manage a user (Likert scale: value 1: not easy to use; value 5: very easy to use) .....	41
Figure 27: Functionality to manage a project (Likert scale: value 1: not easy to use; value 5: very easy to use).....	42
Figure 28: Functionality to manage the assignment of a project to users (Likert scale: value 1: not easy to use; value 5: very easy to use).....	42
Figure 29: Functionality to write on a pad (Likert scale: value 1: not easy to use; value 5: very easy to use) .....	43
Figure 30: Functionality to use the chat of the platform (Likert scale: value 1: not easy to use; value 5: very easy to use).....	43
Figure 31: Functionality to see the connected user at the same time in the project (Likert scale: value 1: not easy to use; value 5: very easy to use) .....	44

<b>Figure 32: Functionality to save versions automatically (Likert scale: value 1: not easy to use; value 5: very easy to use) .....</b>	<b>44</b>
<b>Figure 33: General use of the platform. Is it easy? (Likert scale: value 1: not easy to use; value 5: very easy to use).....</b>	<b>45</b>
<b>Figure 34: General view: Is the platform useful to manage a project? (Likert scale: value 1: not easy to use; value 5: very easy to use).....</b>	<b>45</b>
<b>Figure 35: General use: Does it improves the control of the project? (Likert scale: value 1: not easy to use; value 5: very easy to use).....</b>	<b>46</b>
<b>Figure 36: Example of accessibility checking.....</b>	<b>48</b>

## List of Acronyms

CC	Competence Cell
CRM	Customer Relationship Management
FOSS	Free/Open Source Software
FoTERRIS	Fostering a Transition Towards RRI Systems
MVC	Model-View-Controller
RRI	Responsible Research and Innovation
W3C	World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines

## About the FoTERRIS project

FoTERRIS develops and introduces new governance practices to foster Responsible Research and Innovation (RRI) policies and methods in Research and Innovation (R&I) systems.

FoTERRIS stresses that RRI is a collaborative activity from the very beginning. Therefore FoTERRIS adds the prefix 'co' to the acronym RRI. Important present-day challenges are of a global nature but manifest themselves in ways that are influenced by local conditions. Thus, FoTERRIS focusses on glocal challenges, i.e. local or regional manifestations of global challenges and on local opportunities for solving them.

FoTERRIS performs a transition experiment, i.e. an experiment to support the transformation of present-day research and innovation strategies into co-RRI-strategies. It designs, tests and validates the organisation, operation and funding of co-RRI competence cells. A competence cell is conceived as a small organisational unit, which functions as a local one-stop innovation platform that encourages various knowledge actors from science, policy, industry and civil society to co-design, -perform, and –monitor co-RRI-projects that are attuned to local manifestations of global sustainability challenges.

Since research and innovation systems and practices in EU member states and within different research performing organisations vary, FoTERRIS experiments the implementation of new governance practices in five member states. These five experiments are evaluated, validated and constitute the basis for FoTERRIS policy recommendations towards EU and member states policy makers so as to enforce co-RRI into the national and EU R&I systems. Training is dispensed to various stakeholders, so as to form them to establish other co-RRI competence cells.

**For more information see <http://www.fotrris-h2020.eu>**



Coordinator contact:

Nele D'Haese / Unit Sustainable Materials Management / VIT NV / Boeretang 200, 2400 MOL, Belgium.

t: +32/14 33 58 46 | e: [info@fotrris-h2020.eu](mailto:info@fotrris-h2020.eu) | w: [www.vito.be/english](http://www.vito.be/english)



# 1 Introduction

This document is mainly addressing software engineers and technicians who are interested in reusing or installing the FoTERRIS co-RRI platform. This is a web based platform that implements several functions, utilities and services to support co-RRI Competence Cells (CC). It embodies a process architecture, i.e. a well-structured workflow consisting of various phases that are supported by appropriate tools and methods, which guides the participation of innovation and knowledge actors through the process of co-designing RRI project concepts. This process architecture reflects the co-RRI concept (which is described in deliverable D1.2 [D12]). The platform facilitates the process via online dialogues between participating actors or stakeholders, and the co-creation of accessible and transparent documentation of the intermediary and final process results. The process, consisting of both online activities and real life workshops, will support actors to gradually design, together with other actors, a co-RRI project concept.

As a web-based service to foster co-RRI, the platform is meant to support stakeholders to address *glocal* challenges. With this purpose, it provides several services, to support implementation of the conceptual framework for co-RRI as defined in WP1. More specifically, the platform provides:

- Innovation services to facilitate interactions between stakeholders and to support knowledge actors to co-design RRI-projects. In order to realise co-projected visions of solutions to local manifestations of global societal challenges according to RRI methods and standards by following the co-RRI process architecture.
- Communication and dissemination of co-RRI activities and results.
- Storage of lessons learnt from past RRI projects, such as examples of co-RRI project concepts.

This co-RRI platform is based on state-of-the-art collaborative platforms, which have been reviewed in the project (see section 2). The general technical prerequisites of the platform, as stated in the project Technical Annex, are the following:

- Support for online communication (discuss, comment).
- Support for online collaboration (in order to reach a common solution).
- Provide some means of dissemination for the general public.
- Searchable storage of past projects.

## 1.1 Use and Design for co-RRI

The stakeholder feedback is considered essential for the success of the web platform, as we do not aim to build “yet another tool” that would not be used, but to fulfil the needs of the different stakeholders taking part in the co-RRI process. Due to the difficult nature of the task, the changing needs and the possibility of unforeseen difficulties in the development of the co-RRI platform, a rapid prototyping approach has been adopted (based on both *Lean design* and *Scrum agile* (Lines and Ambler, 2012) development methodologies). This conforms to standard practices in industrial engineering for innovative technologies, and has been already applied with success in other projects by UCM (for instance in the P2P Value<sup>1</sup>

---

<sup>1</sup> <https://p2pvalue.eu/>

project). An advantage of this approach is that it helps to ensure that the intended design and the implementation are coherent and evolve to fulfil users' needs.

Besides, the approach to development has been both incremental and participatory, with short development cycles to test the features of the co-RRI-platform. Early releases of working system have allowed involving the stakeholders from the start and assessing and evaluating the progress regularly. Preliminary testing has been performed prior to deployment of the first version and has been performed for each version in a laboratory environment.

The co-RRI platform, in order to fulfil its aims and facilitate its sustainability and widespread use, has been built on top of free/open source software (FOSS), together with open standards and open protocols to facilitate interoperability. Besides, its development process has been “open”, that is, with public repositories (GitHub: <https://github.com/Grasia/fotrrisweb> ), public tickets (reports and feature requests) and public documentation. This facilitates, for anyone, following the project development, reporting bugs or even collaborating with the developers in the evolution of the platform.

## ***1.2 Vision and Scope***

The goal of the FoTERRIS web based platform for co-RRI is to support users to work in a collaborative way, in real time, on the definition of a RRI project concept, following the process of the FoTERRIS methodology.

This means that the FoTERRIS web based platform features are derived from the FoTERRIS methodology. This methodology specifies a sequence of phases in the process of defining RRI projects. Each phase is implemented as a workshop with people representing different stakeholders, and coordinated by a co-RRI expert (a member of the competence cell that manages the process). The purpose of the platform is to support users to prepare each of these phases and associated workshops, to collect their results, and to facilitate summarising and making public the outcome of the process. This is done by providing tools for co-edition of documents, user interaction, and publishing, in an integrated framework.

## 2 Collaborative platforms for RRI

### 2.1 *State of the art*

There is no specific collaborative platform for RRI so far. One reason is that RRI project features cannot be managed using typical project management tools. A typical project management tool defines tasks, which are assigned to project members. The project members only interact to impute the number of hours dedicated to each task, so that the progress of the project can be measured in time and cost using Gantt-like tools. The project manager uses the tool to exploit this data and generate reports or re-plan. Redmine, PlanView and Microsoft Project are examples of project management tools. However, this kind of tools does are not well adapted for RRI projects, where the goal is not so well-defined and the tasks are usually not known in advance, or the time that they require cannot be easily estimated. Furthermore, in RRI, the path of the project usually changes through time and the inputs from different actors need to be taken into account. As a result, the work in a RRI project cannot be managed with conventional project management tools, but require an integration of tools that support communication, cooperation and coordination.

In this vein, there are many software tools and platforms that people and organisations use to cooperate and coordinate their work. The software they use depends on many factors including their computer skills, the kind of project they are working on (for example, when the ‘product’ they are creating is or can be digital, such as a digital document), their organisational structure (for example, whether it is a private company with a hierarchical structure or a commons-based peer production organisation) and the needs the software covers (communication, digital repository, planning, collaborative editing, etc.)

Usually, organisations do not rely on a single tool that satisfies all their needs, because such tool does not exist. So they combine several tools, which in some cases are not prepared to interact. With the advent of Internet, email became ubiquitous and many organisations use tools that were born as email clients and where more functionality was added, such as personal calendar, management of resources (e.g., meeting rooms), address books, or lists of personal tasks. Examples of these tools are Lotus Notes from IBM, Microsoft Outlook, Google Mail, etc. Other organisations use software platforms for keeping their own documents and media files in the cloud not only for backup, but also for collaboration. These software platforms usually offer folder and file syncing. Examples of these platforms are Dropbox and SugarSync. Some of these products even double as collaboration software, offering real-time document coediting and office suites; this is the case of Microsoft’s OneDrive with Office 365, and Google’s Drive with Docs. In addition, there are also dedicated tools for collaborative editing including pads such as Etherpad or wikis such as those from MediaWiki.

Besides these tools there is a myriad of tools that support collaboration in different forms. While these tools are used by some private companies, they are intended to support collaboration in a more flexible way including non-profits, cooperatives, commons-based peer production organisations (CBPPs) and project cooperation and coordination in general. In this sense, they are more suitable for RRI collaboration.

We have reviewed many of these tools, which are classified in several categories according to their main purpose. While the lists do not aim to be exhaustive (such a task is almost impossible as many of them appear, disappear or change each month), but to offer a good perspective of the current trends in

collaboration tools. This is useful to identify relevant functionality to support the co-RRI process requirements. Furthermore, as usually these collaboration tools can belong to more than one category, we have assigned them to the category that better represents them.

### 2.1.1 Matching people

**OurGoods** [OurGoods 2017]: Our Goods is a barter network for creative people that connects artists and designers and crafts people and activists with each other so they can trade skills and spaces to get independent projects done. OurGoods is a peer-to-peer online network that facilitates the barter of goods and services between artists. The site matches barter partners, provides accountability tools, and offers technical assistance resources to help artists complete their barter and their projects successfully. Registered users define their profile, list their needs and what they have, and what sort of projects they are involved in. Then, users can search for what they need and find people who can fill those needs, as well as what projects others might have that they could help with. When two persons find a potentially good match, they have a conversation, and decide between and how to rate the exchange of labour.

**Taproot Foundation** [Taproot 2017] The Taproot Foundation is a non-profit organisation that connects non-profits and other social change organisations with skilled volunteers through *pro bono* service. Taproot intends that organisations dedicated to social change have full access, through *pro bono* service, to the marketing, strategy, information, technology and human resources they need to be most effective. They pair non-profit staff with qualified, skilled volunteers for high-impact, high-quality *pro bono* service in a range of ways, from team-based, long-term projects to one-time, one-on-one consultations.

**Coliga** [Coliga 2017]: Coliga is a tool that enables communities of independent workers to share and receive job offers as a collective. Communities of freelancers can find, share and carry out work on their own terms, for which Coliga charges a 5% fee. Every member of Coliga who contributes to the completion of a job offer is rewarded financially. Such a tool makes possible that emerging communities could thrive within the traditional market economy while maintaining their flexible dynamics. By helping communities mediate their own jobs and allowing everyone in the network to earn a portion of transactions they facilitate, Coliga helps workers, and the communities that help them succeed, and keep more of the value they generate.

**Part-Up** [PartUp 2017]: Part-up provides organisations with a platform to match selected internal and external talent to project teams. Organisations that use Part-up let people decide via the app itself which projects and with whom they want to work with. Temporary teams are assembled and ready to start easily. This aims to empower organisations to increase their strength and capacity for innovation. A user can find a work that fits, choose interesting communities or organisations (*tribes*) to join, and receive personal suggestions for temporary teams (*part-ups*), where to contribute with their skills and talent.

**Taiga Tribe** [Taiga Tribe 2017]: Taiga Tribe is a tool to find collaborators for specific tasks. Taiga Tribe is an extension to Taiga (see below), which supports finding external help for your projects but also, as an independent tool, can be used to hire people to work on a specific task for any of your projects, not necessarily Taiga related. Typically, if you manage a project you post a User Story that you need people to develop as a gig on Taiga Tribe and hire interested people, with the required knowledge, to work on your gig. Once they finish the gig, you can pay them via PayPal.

### 2.1.2 *Project creation and promotion*

**Hylo** [Hylo 2017]: Hylo is a collaborative creation platform. Hylo is built for communities-of-purpose, like co-working spaces, accelerators, and makerspaces; turning into vibrant resource-sharing networks. In Hylo users can post “Intentions” about what they would like to do or create, regardless it is a great vision or a small thing. In this way their friends and the communities they belong to can see their intentions, support them, offer suggestions and also help them out. Intentions can turn into Projects and be broken down in steps and Requests that need to be fulfilled to bring the project into reality. Thus, it creates a network of support through a “crowd-resourcing” process. And, if the Project reaches a stage where financial resources are required, Hylo allows to host rewards-crowdfunding campaigns, and Direct Public Offerings.

**Colony** [Colony 2017]: Colony is a social collaboration platform for distributed organisations. Users can start projects online and build a workforce to help make them a reality. The platform encourages users to compete to be the most skilled and productive workers by distributing crypto-equity in proportion to the value of each user’s contribution. Because it aligns the incentives of the workforce around being productive, it is possible for people anywhere in the world to work on a project without the need for hierarchical management. The platform automates the project management process, aggregating the collective intelligence of the workforce in suggesting tasks, making decisions, assigning tasks to the best candidates and providing feedback on people’s work. Every aspect of the platform employs game mechanics and behavioural design to provide a compelling experience which encourages repeat engagement.

**CoMakery** [CoMakery 2017]: Comakery helps early stage entrepreneurs to CoMake products. Users can join a project or start one. They contribute by coding, designing, creating content, marketing, exchanging ideas, etc. In doing so, they can earn shares of future revenue, while getting recognized for their skills and unlock new opportunities. In other words, Comakery is a tool to track and trade sweat equity, i.e., the hard work that someone does to create a product that helps to increase its value. Comakery tracks contributions and distributes awards in project tokens (like Bitcoin for projects), and provides support for simple licensing agreements. The platform includes Slack integration and an Ethereum blockchain accounting system.

**Teem** [Teem 2017]: Teem is an open source app developed specifically for online collaborative community teams developed in the European FP7 project P2Pvalue. It was originally conceived for commons-based peer production communities, but its features apply to any kind of open, online community. It is meant to increase participation along with sustainability, as its features were determined through intensive social research. This research revealed the main needs of the people in specific roles in communities – those roles as defined in the classical 1-9-90 rule [NJ06], including “core, occasional collaborators and users” – and what tools the people in these roles often find themselves lacking. Teem provides an informal project management area with tasks, where people can join to collaborate, a workspace with collaborative edition (like a Google doc) and group chat (like Whatsapp).

### 2.1.3 *Project management*

**Taiga** [Taiga 2017]: Taiga is a free and open-source project management system that can handle complex projects for startups, software developers, and other target teams. It tracks the progress of a project and users can choose either a Kanban or a Scrum template, which are software development methodologies but that can be applied to any kind of projects. Taiga works with the concept of User Stories, which in

software engineering allows to describe system features from an end-user perspective. Backlogs are shown as a running list of all features and User Stories added to the project. Taiga also integrates video conferencing functions and chats.

**Basecamp** [Basecamp 2017]: Basecamp is essentially a web-based project management tool. Basecamp's primary features are to-do lists, milestone management, forum-like messaging, file sharing, and time tracking. Each company has three elements: Company HQ (acronym for headquarters), teams, and projects. Inside each one of them, there are six tools: to-dos for tracking work, a message board for posting announcements and updates, a chat room for quick casual chats with the team, a schedule for posting deadlines and milestones, docs & files for organizing all the assets and notes everyone needs to do their work, and automatic check-ins to get insights from the team on a regular basis.

**OpenCollective** [OpenCollective 2017]: Open Collective is a crowdfunding platform for communities that can collect and manage money transparently without creating a legal entity. Because it operates in full transparency, it can safely be hosted by an existing legal entity (an organisation or an individual) that will in practice collect the money on behalf of the collective and report its activities in accordance to local legislations. OpenCollective is open source and aims to adapt the software to comply with local regulations.

**CoBudget** [CoBudget 2017]: CoBudget is an open source app developed by Enspiral. It makes possible to prepare collaborative and transparent budgets, by following a series of steps. Each month contributions to collective funds are published. Everyone can see who contributed to what and how much money is in the budget. Basic core expenses, for example, rental or utilities expenses, are collectively agreed on Loomio, then are subtracted and what's left is the discretionary budget. Each person or company retains the right to allocate their part of discretionary funds and anyone in the network can start a *bucket*, which is a proposal to do work that requires funding. They write up a proposal making their case for why the work they want to do will benefit everyone and why they are the right person to deliver the project. Everyone then considers the buckets and decides which ones to "fill" with their portion of the discretionary budget. If people collectively feel that a project will make a good use of resources, it will get funded. Funders can split up their allocations as they like, or put it all in one bucket. In aggregate, the result is a budget that reflects the collective priorities of the group, determined in proportion to real stakeholding and in the context of the big picture goals.

#### 2.1.4 Communication

**MetaMaps** [MetaMaps 2017]: Metamaps is an open source, collaborative concept mapping platform for visual sense-making by individuals and collectives. Maps are the central feature of the platform, where users can visualize and work with interconnected concepts and content. Maps are graph-based collections of *topics* and *synapses*, to map out ideas. Anyone can view public maps and topics on the platform, but it is needed to register a user account in order to create or edit maps. Maps make possible to make and discover connections between people and ideas. Maps can be reused in other maps as selected by users. With increasing usage, topics become part of a growing web of contextually linked semantic data, providing a powerful environment for various Collective Intelligence applications and insights.

**Loomio** [Loomio 2017]: Loomio is a free and open source tool for collaborative decision-making. Loomio is a tool for groups who want to collaborate democratically. It enables more transparency and inclusion, with



fewer meetings and emails. Unlike a survey or poll, it facilitates a process of constructive deliberation, synthesising solutions from diverse viewpoints. The top-level organisational structure in Loomio is the Group. A group is made up of members, granted permission to that group. Groups can be both public and private, allowing for privacy or openness where required. Within groups, members can create discussions on specific topics. During a discussion, members of the group can post comments and create proposals. Proposals solicit feedback from members on a specific proposition. Members can either agree, disagree, abstain, or block. Blocking is essentially a strong form of disagreement.

[Assembl](#) [Assembl 2017]: Assembl is an open source software platform from the European FP7 project Catalyst. It is intended to be a collective intelligence platform that can host complex and large online debates. The platform is designed according to a configurable approach (modules of creativity, votes, multi-criteria evaluation) which results in a formalized deliverable. Participants' contributions are free. They are then structured and organized so that no idea is lost. The organisation of continuous messages makes it possible to eliminate the noise and to focus the attention of the participants. Votes, when necessary, are only triggered at the end of the process so as not to prematurely disqualify singular ideas. Syntheses are produced regularly: they give a panoramic view of the discussion and ensure the dynamics of collective intelligence.

[Slack](#) [Slack 2017]: Slack is a team collaboration tool that offers many IRC-like features: persistent chat rooms (channels) organized by topic, as well as private groups and direct messaging. Slack is organized in teams, which are comprised of channels. Channels are chat rooms and can be public or private, they hold conversations between team members. They can be organized around departments, projects, etc. All content inside Slack is searchable, including files, conversations, and people. Slack integrates with a large number of third-party services and supports community-built integrations including Google Drive, Dropbox, and GitHub.

#### 2.1.5 *Building Communities*

[CiviCRM](#) [CiviCRM 2017] is a web-based, open source, internationalized suite of computer software for customer relationship management (CRM), more precisely for constituency relationship management, specifically designed for the needs of non-profit, non-governmental, and advocacy groups. CiviCRM makes possible to manage volunteers, activists, voters as well as more general sorts of business contacts such as employees, clients, or vendors. CiviCRM is also designed to manage contacts, memberships, cases, events, contributions by donors and accounting.

[NationBuilder](#) [NationBuilder 2017]: NationBuilder is a platform to create a website to “grow your community and lead them to action,” via social media. Political organisations and advocacy groups are using it to carry out their efforts. NationBuilder gained popularity since it was successfully used by Donald Trump and the Brexit movement in their campaigns of 2016 and Macron’s *En Marche* in 2017 (O’Brien 2017, July 14). It has tools for fundraising, coordination of events, creation of petition pages with goals, and member management. It also tracks the members using the email address from the profile in order to find any public social media profiles associated with that email address.

[Drutopia](#) [Drutopia 2017]: Drutopia emerges as an alternative to platforms such as NationBuilder. It aims to be a Drupal for the grassroots inspired by the principles of software freedom, community ownership and

intersectional politics. Amongst their core values, we can find the protection of personal information and privacy and freedom from surveillance, and the prioritization of value collaboration and cooperation above competition. As of today, Drutopia is only an initiative and there is no software associated to it.

[Mobilize.io](#) [Mobilize 2017]: Customizable “network relationship management” platform Mobilize is a leading social communication platform for helping group leaders to build strong relationships and communicate at scale with their network of partners or members. Mobilize is an all-in-one solution for email, sms, events and polls with a built-in database and analytics. Mobilize is used by businesses relying on large groups of external stakeholders, such as developer relations, on-demand workforces, brand ambassadors, startup accelerators, professional networks and marketplace sellers, including companies such as Google, Microsoft, Salesforce, Verizon and many other small and large companies.

[Mighty Networks](#) [Mighty Networks 2017]: Mighty Networks is a social network for deep interests. Anyone can create a Mighty Network for free about something they are interested in. It aims to gather people around a shared interest. Each Mighty Network uses proprietary algorithms and smart technology to introduce members by location, by topics they choose and by categories they define. With everything for an interest in one place, members can always get back to the content, conversations and members.

## 2.2 *Conclusions from the review of the state of the art*

This review shows that platform design is usually driven by the communities they intend to support. This fact affects the main features of each platform and the way they are implemented, such as the rewarding strategies, the organisational roles and structure, the way of defining goals, the project organisation, etc. Some of these features may be relevant for supporting the co-RRI methodology, and have been taken into account when defining FoTERRIS platform requirements (in the next section). From this review, we have identified the following as the most relevant and common features:

- **User roles.** Even if these platforms are not intended for conventional project management, some of them still differentiate the roles related with project manager and collaborator, e.g. Basecamp, Taiga tribe or Comakery. However, in some other platforms, roles are not predefined, and every user can either create a project or collaborate with it (e.g. Part-up). Some other platforms, such as, Mobilize, NationBuilder, CiviCRM, OpenCollective, are tools intended for people already in an organisation that want to coordinate supporters or collaborators or collect donations, but those are not users of the platform.
- **Content access.** In most of these of these platforms, such as Loomio, users need granted permission to access the content of the group or project. Some others follow an open approach with full access to all the functionality to anyone (e.g. Metamaps or Teem).
- **Project management.** Most of these platforms have a non-traditional concept of project management. Users can post tasks that need to be done in an informal way. In some cases, these tasks can be part of a project outside the platform (e.g. Taiga Tribe), while in others they are the project or product to be developed (e.g. Comakery). Others, such as Hylo, have a more comprehensive approach of project creation from the “intentions” to “project” and allow to break down projects into tasks and “offers” and “requests” of work. Finally, some, such as Basecamp or



Taiga, are more similar to traditional project management tools in a holistic sense, aiming to centralize communication and project work with lists of tasks, schedules, centralized repositories, chat rooms, etc.

- **Task assignment.** In these platforms, unlike in traditional project management tools, tasks rarely are directly assigned to people by the project leader, being Basecamp an exception. In many of them, such as OurGoods or Coliga, the person that creates the task waits for volunteers to do it and chooses the suitable one. Some others are more sophisticated, such as, Part-up, which facilitates the self-organisation and the creation of temporary teams to work into projects.
- **Rewarding mechanisms.** Some of these platforms include an economic compensation for the collaboration, e.g. Coliga and Taiga Tribe. But in many of them, collaboration is free and spontaneous. In some cases, it is even un-rewarded, e.g. Metamaps. However, some other platforms aim to stimulate collaboration by including rewarding mechanisms. Typically, they follow a meritocratic approach, for example, distributing crypto-equities in proportion to the value of each user's contribution, as is the case of Colony or CoMakery. In Colony, user's contributions are also a way to build a reputation, which is another rewarding mechanism. Other platforms, such as Part-up or Taiga Tribe, use direct rating to contributions or contributors as reputation mechanisms.
- **Communication.** Some of these platforms are specifically designed to facilitate communication. For example, Slack includes tools such as chat-rooms, private groups and direct messaging, or Loomio, which aims to facilitate debates and collaborative decision-making. Other platforms, such as NationBuilder or Mobilize, are "broadcasting" tools that use social media to build and communicate with a network of supporters. The project-oriented platforms also include some tools for communication. Typically, message boards or chat rooms associated to projects or tasks, as it is the case of Part-up or Taiga. CoMakery goes further and integrates Slack as communication tool. Basecamp is more ambitious than other project-oriented platforms, as it aims to streamline and centralize communications related with the projects inside the platform by means of message boards, chat rooms and direct communication related with the status of tasks.
- **Decision making.** In some platforms such as Loomio or Colony, decisions are made democratically. In some of them, e.g. Loomio or Assembl, all the votes count the same, while others use a proportional voting system, for example, in Colony the greater the reputation/expertise, the more influence.

### 3 Analysis of the FoTERRIS web-based platform

#### 3.1 Requirements for supporting FoTERRIS co-RRI methodology

As a web-based service to foster co-RRI, the FoTERRIS web platform is meant to support stakeholders to address *global* challenges. This has to be done following the principles of the co-RRI conceptual framework and the methodology to be applied in the co-RRI competence cells, as defined by the FoTERRIS project.

The main functional requirements were defined initially as follows:

1. R1: Innovation services to facilitate interactions between stakeholders and to support knowledge actors to co-design RRI-projects in order to realise co-projected visions of solutions to local manifestations of global societal challenges according to RRI methods and standards by following the co-RRI process architecture.
2. R2: Communication and dissemination of co-RRI activities and results.
3. R3: Storage of lessons learnt from past RRI projects, best cases examples and data on sustainability challenges.

Other services could be also investigated by taking into account feedback from the use of the platform.

Furthermore, the platform is compliant with web accessibility standards, in concrete level AA (<http://www.w3.org/WAI/>), and is responsive so it can be accessed from a variety of devices (computers, tablets, and smart phones).

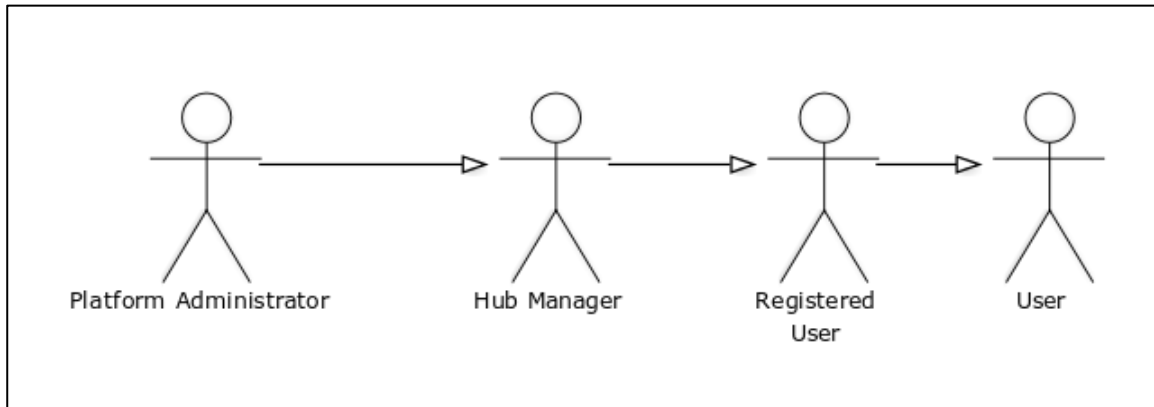
#### 3.2 Use Cases of the platform

##### 3.2.1 Actors

There are four main types of actors of the applications:

1. **Users:** everyone that accesses to the platform. They can explore the projects in the “Show Projects” section of the platform and decide to register in order to collaborate in any project.
2. **Registered Users:** Registered users with profile “User FoTERRIS”. They can co-work in the different projects which they have joined. They can edit the content of each collaborative document, called pad, in real time, at the same time as other users. In addition, they can use a chat tool for communicating with other users on the same document.
3. **Hub Manager:** The administrator of the users of a co-RRI Hub for a Competence Cell. A co-RRI Hub in the platform is the place for hosting projects that are promoted by a Competence Cell. A responsible of the Competence Cell will take the role of Hub Manager in order to create new projects for the hub and manage the users that will participate in those projects.
4. **Platform Administrator:** The user that manages all the functionality of the platform. This user can create new users, new hubs and new projects, and assign users to projects.

Figure 1 shows these types of users, and the fact that each category has the capabilities of the previous one, i.e., a Platform Administrator can perform all the functionality of a Hub Manager, who can perform all the functionality of a Registered User, and this one that of a simple User.



**Figure 1: Types of actors on the platform**

### 3.2.2 Use case diagrams

The next figures present the different use case diagrams for each actor of the platform, identified in the previous section.

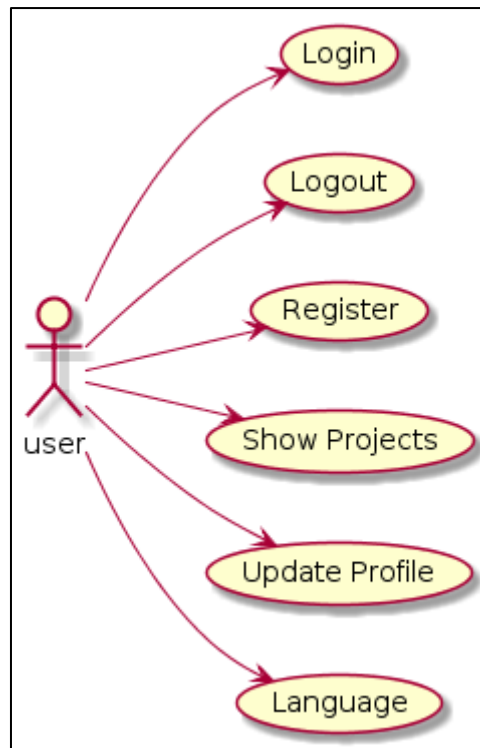
#### 3.2.2.1 End-user Actor

There are two kinds of end users: those who are registered in the platform (*Registered User*), who can participate in the co-creation process of RRI projects, and those who just can browse and see what projects exist (*User*).

**Users:** general user who can only see the main page of the platform, where they can do:

- **Login:** to establish a session to work in one or several projects.
- **Logout:** when the work is finished. There is automatic logout when a certain amount of time has passed without activity by the user, just for security reasons.
- **Register:** fill in the form to register on the platform.
- **Show Projects:** browse the projects to see their description.
- **Update** the user's **profile**, which includes some personal data, such as name, institution, location, password, and a photo.
- **Language:** change the language of the platform.

Figure 2 presents the use case diagram for the *User* actor.



**Figure 2: Use case diagram for the User actor**

### Description of relevant use cases

#### Update Profile

This basic functionality allows a user to browse and edit the corresponding information in the platform.

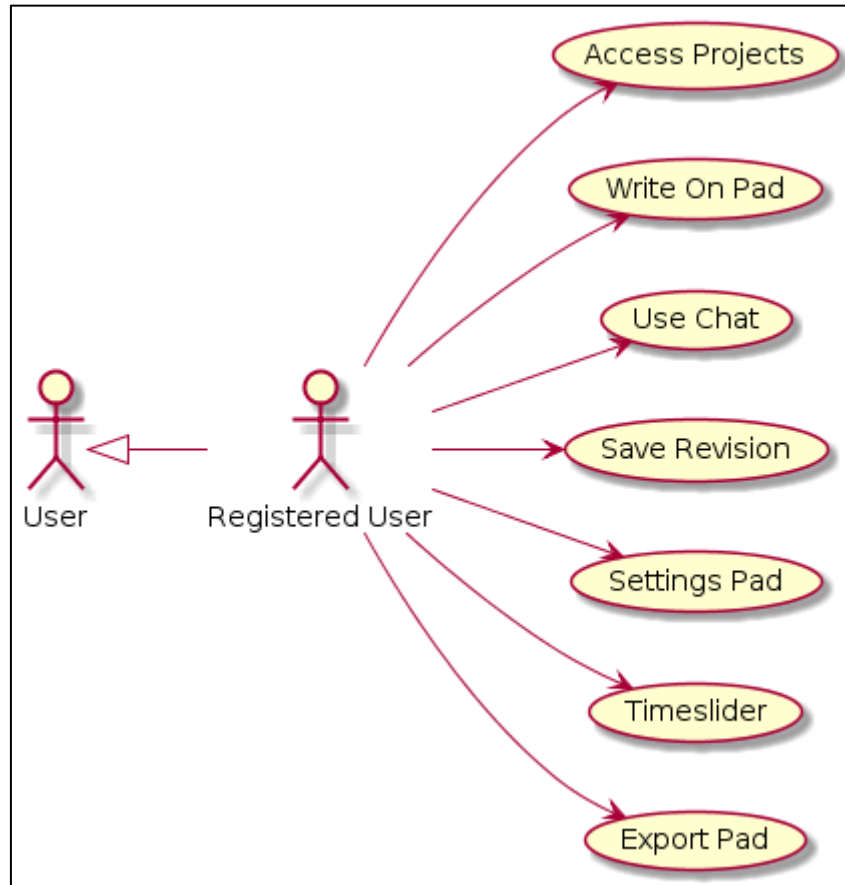
#### Show Projects

This functionality let user consult the information of each project created in the platform. In addition, a user can ask to join to a project and he can download its public abstract.

**Registered Users:** registered users can do more things, in addition of all from the User:

- **Access Projects:** access to the different projects which they have joined.
- **Write On Pad:** writing on the collaborative documents of the project, called pads.
- **Use Chat:** communicate with other users using the chat of the platform.
- **Save Revision:** save revision of the pad they are working on.
- **Settings Pad:** configure pad's setting.
- **Timeslider:** consult the time slider of the document.
- **Export Pad:** export to different file formats.

Figure 3 presents the registered user use case diagram.



**Figure 3: Use case diagram for the Registered User actor**

### Description of relevant use cases

#### Write On Pad

This functionality is one of the most important because it allows users to collaborate in the definition of the projects. The registered users can write on the different pads that can be defined for each project.

#### Use Chat

This functionality allows users to communicate using a chat of text. This chat is available to the registered users in each collaborative project created in the platform.

#### Save Revision

This functionality allows making backups with all the information that is generated for a Project. Because the pads are modified by different users, it is necessary to keep track of changes and the ability to recover in case of some fault or could loss of information.

#### Time slider

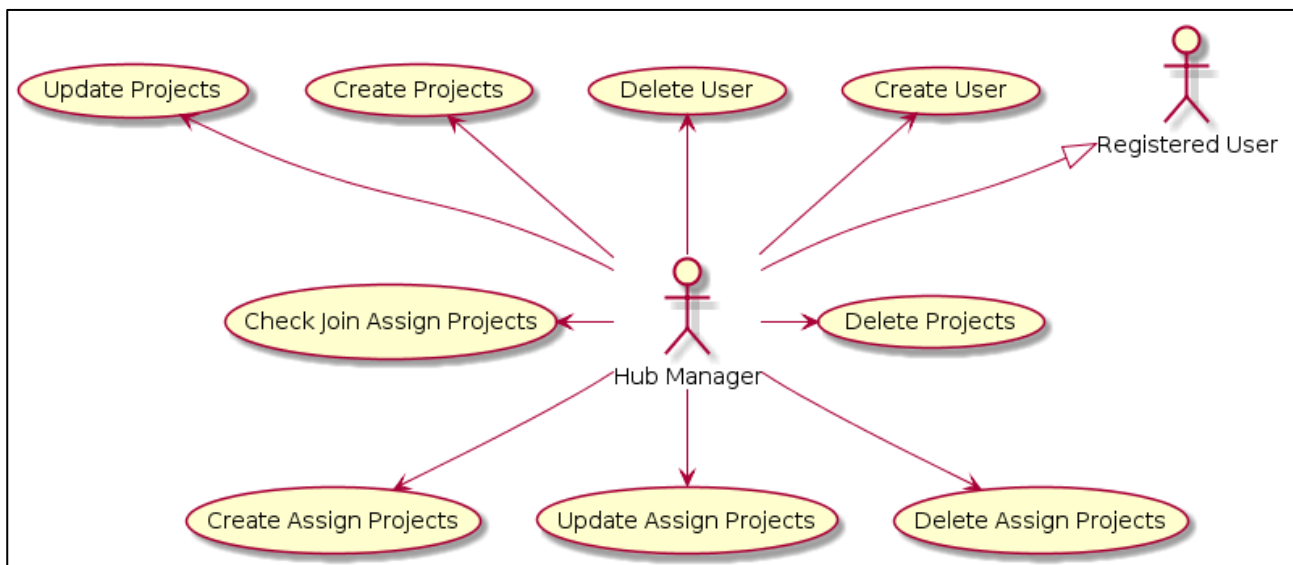
This functionality allows for each pad to keep track of all the modifications (text additions, deletions, etc.) that have been performed by the different users.

### 3.2.2.2 Hub Manager Actor

The Hub Manager can use the same functions of the registered users, in addition to manage the users and the projects of the corresponding hub:

- **Create User:** create a new user on the platform.
- **Delete User:** delete a user from the platform.
- **Create Projects:** create a new project with the number of pads needed.
- **Update Projects:** modify and update the information related to a project.
- **Delete Projects:** delete a project, only if they are the manager.
- **Create Assign Projects:** assign a project to a user.
- **Update Assign Projects:** modify and update the information of the assignment of a project to a user.
- **Delete Assign Projects:** delete the assignment.
- **Check Join Assign Projects:** test the user pending the assignment from their projects.

A project is created by the User Admin, who will assign one project manager for it. One project manager can be responsible of several projects, and can be a normal registered user for other projects. Figure 4 presents the use case diagram for a Hub Manager.



**Figure 4: Use case Diagram for a Hub Manager**

#### Description of relevant use cases

##### User Administration: create and delete user

This functionality provides control on the roles and permissions of different registered users of the system. It is possible to create and delete users from the platform.

Project Administration: create, update and delete projects

This functionality allows controlling the different projects created in the system. Using this functionality different contents can be created in order to work the different groups in a collaborative way.

All projects belonging to the different hubs and using the platform are presented with this option. Each one can be updated or deleted. In addition, new projects can be created.

Assign Projects Administration: create, update and delete assigned projects

This functionality controls the access of the registered users to the different projects.

### 3.2.2.3 Platform Admin actor

The Platform Admin can perform all the functions of the Hub manager. In addition, the Platform Admin has the control of the hubs on the platform. Figure 5 presents the Platform Admin use case diagram.

- **Create Hubs:** create a new hub on the platform.
- **Update Hubs:** modify and update the information related to a hub.
- **Delete Hubs:** delete a hub from the platform.

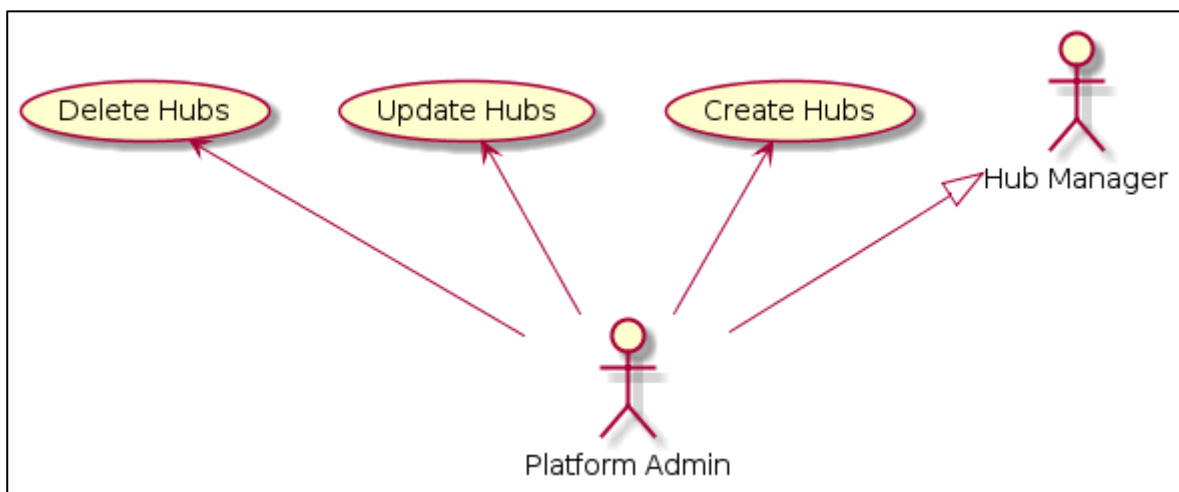


Figure 5: Use case Diagram for a Platform Admin

### Description of relevant use cases

Hubs Administration: delete, update and create hubs

Basic functionality to control all co-RRRI hubs created in the system. Each hub contains different collaborative projects created in the platform.

#### 3.2.2.4 Use Case Priority

The next table presents the priority of the different use cases that have been defined for the platform.

Use Case	Business Priority	Technical Priority
User Administration	High	High
Project Administration	High	Medium
Assign Projects	High	High
Hubs Administration	High	Medium
Use Chat	High	High
Save Revision	High	High
Write on Pad	High	High
Login	High	High
Logout	High	High
Register	High	
Settings Pad	Medium	Low
Timeslider	Medium	Low
Import/Export Pad	Medium	Low
Show Projects	Medium	Low
Update profile	Medium	Low
Language	Low	Low

**Table 1: Priority of Use Cases**



## 4 Design of the FoTERRIS web-based platform

### 4.1 Objectives and Restrictions of the System Architecture

Objective:

- Comply with the specifications and requirements established in the description of the work packages WP2 *Design of a multi-actor experiment*, Task 2.1 *Design of a co-RRI-web based platform* and Task 2.2 *Implementation and deployment of co-RRI platform*.

Restrictions:

- The architecture should be flexible to allow the incorporation of new functionality in the future.
- The platform should support online communication among users (discuss, comment).
- The platform should support online collaboration among users (in order to reach a common solution).
- The system should provide some means of dissemination for the public.
- The system must allow storage and consultation of past projects.
- The platform must be free/open source (FOSS), which implies to use FOSS technologies to implement it, together with open standards and open protocols to facilitate interoperability.

### 4.2 Specification of the System Architecture

The architecture of the application is presented as a set of views, which are common in software engineering [Kruchten 1995]. These views are:

- **Use case View.** It consists of the main use cases and scenarios, which show how users interact with the system. These are described with UML use case diagrams. (See Section 3.2)
- **Logical View.** It describes the functionality of the system and the objects of the domain. This is described with UML class diagrams and sequence diagrams. (See Section 4.2.1)
- **Process View.** It provides the dynamic view of the system, i.e., how the different processes components that build the system interact. This is described with UML activity diagrams. (See Section 4.2.2)
- **Implementation View.** It describes how software components are built and structured. These are described with UML package diagrams. (See Section 4.2.3)
- **Deployment View.** It describes how the software components of the system are deployed in different computer nodes in the network, and how they can be configured. This is described with UML deployment diagrams. (See Section 4.2.4)

#### 4.2.1 Logical View

The architecture follows the well-known Model-View-Controller (MVC) pattern. The objects of the model are persistent, so they are stored in a database, which is described with all details in Appendix 1. Figure 6 presents the class diagram of the application model.

The main purpose of the application is the *creation of collaborative projects* on the platform. There is a class to represent projects (see Figure 6), with several attributes like name, hub, location, description,

duration, number of pads, status, keywords and the image related to the project. In addition, a project has a list of keywords, a *repository*, at least one collaborative document or pad called *abstract* and a number of pads. This project belongs to a specific hub which has a manager. This manager is a registered user. Another important functionality of the application is *assign projects* to users. These users have to be registered users to collaborate in a project.

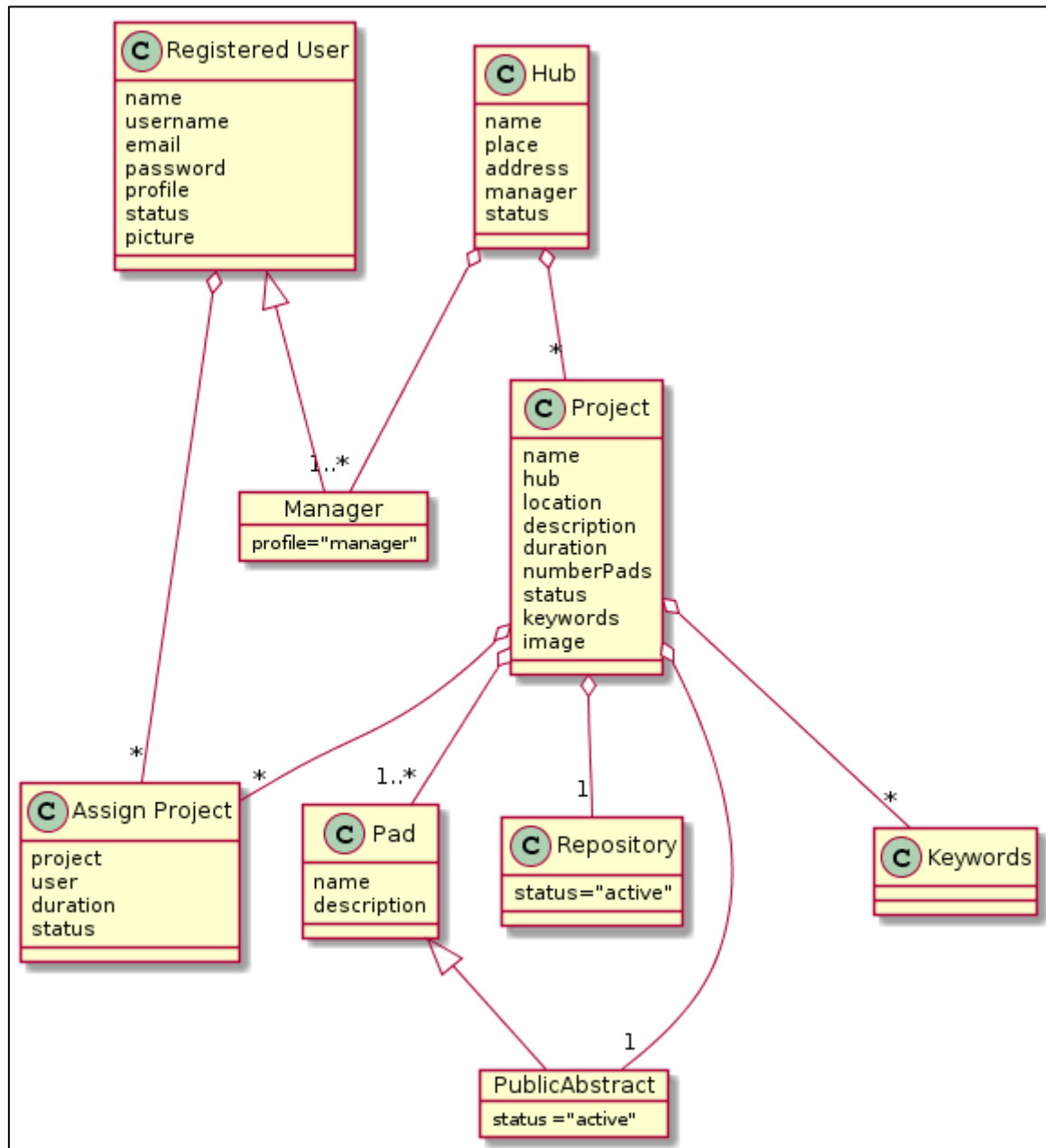
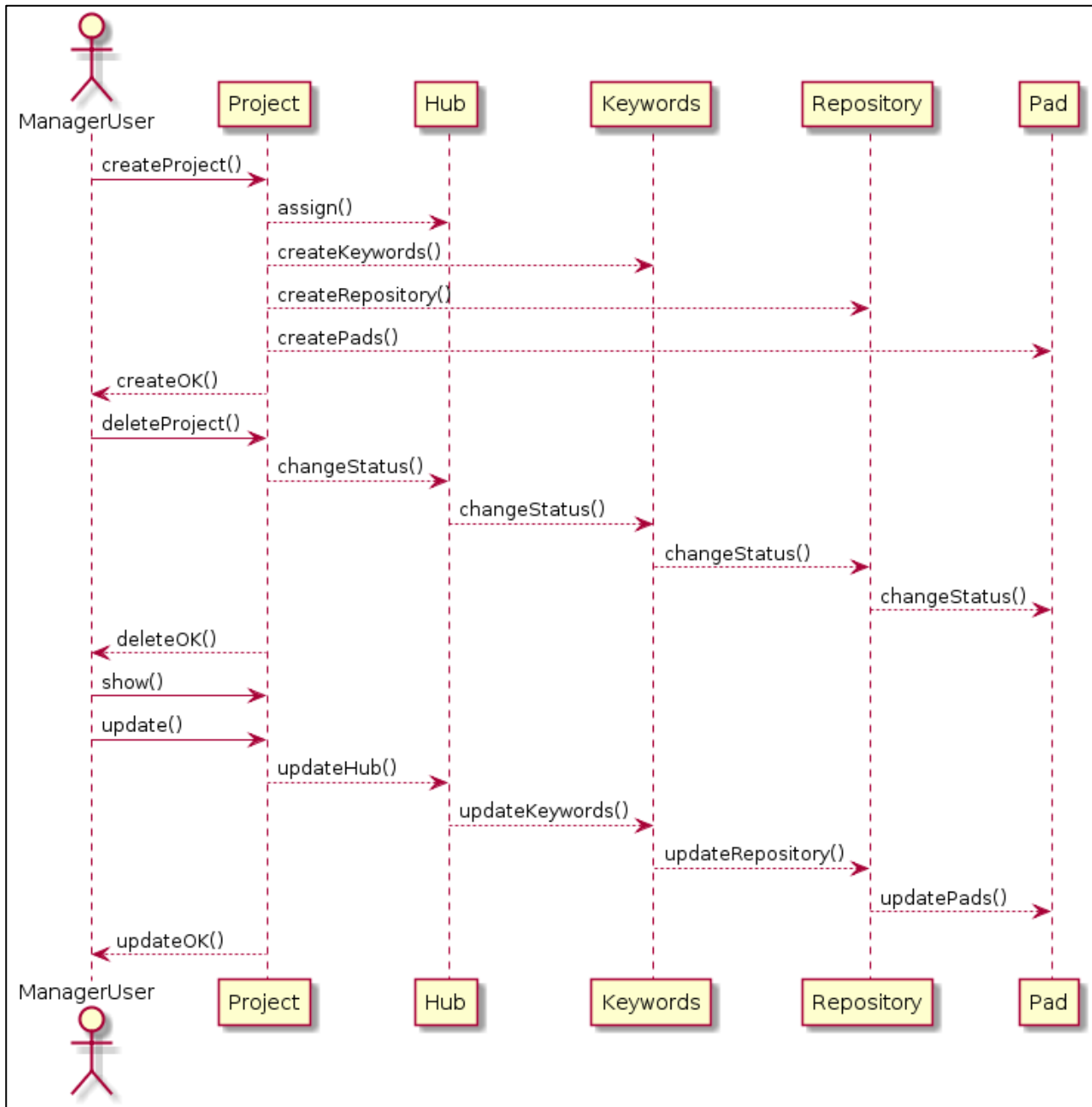


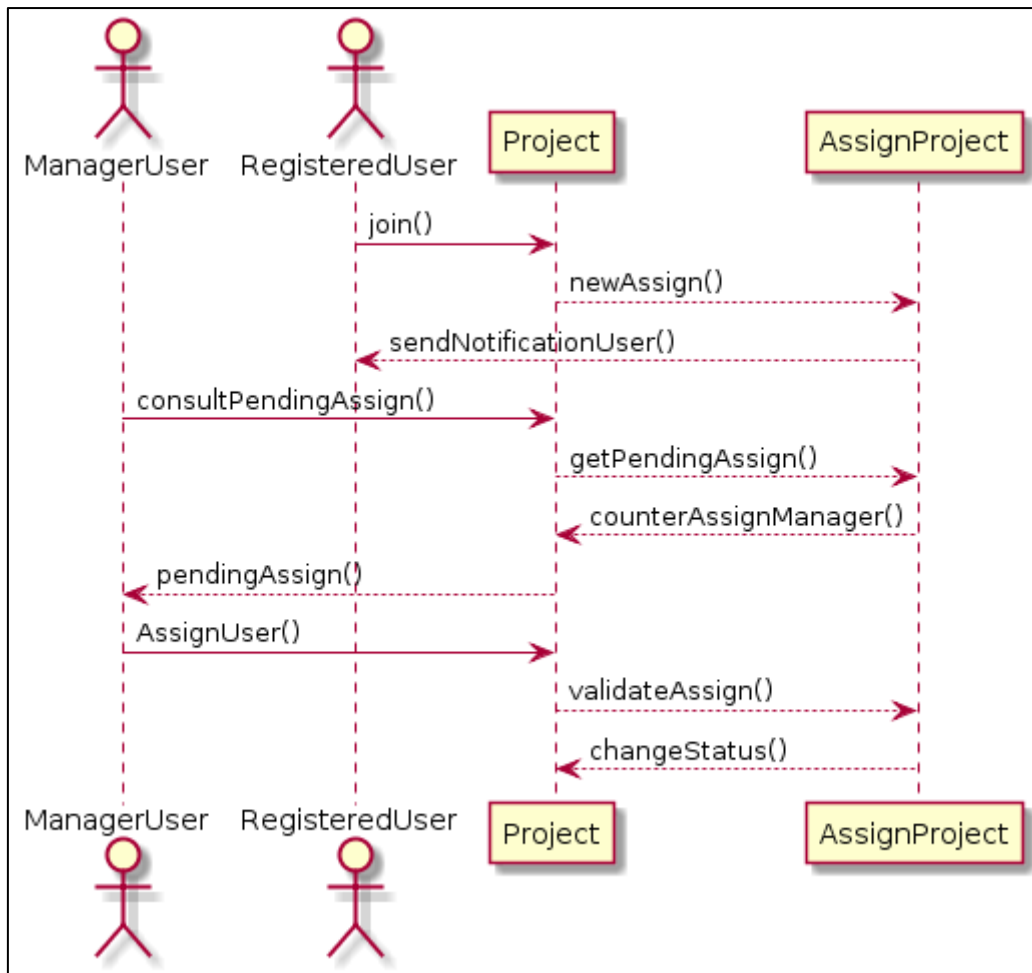
Figure 6: Class diagram with the main entities of the Model

In addition, there are some classes to access databases, *FluentPDO* (PDO is an acronym for PHP Data Objects) like *BaseQuery*, *CommonQuery*, *InsertQuery*, *DeleteQuery*, *SelectQuery* or *UpdateQuery*. Other classes define the controller elements like *PerfilController*, *ProjectController*, *UserController* or *AssignController*. Of course, for the view there are different PHP files that generate the different windows that are presented to the users of the platform.

Figure 7 shows the sequence diagram for the creation of a project is presented by the manager user. And Figure 8 presents the assignment of a user to a project.



**Figure 7: Sequence diagrams for management of a project (creation, deletion, update)**



**Figure 8: Sequence diagram for the assignment of a Registered User to a Project**

The next diagrams show different functionalities of the system and the objects of the domain. Figure 9 presents the specific sequence diagram to login on the platform. Figure 10 shows the sequence diagram of a Registered User to join to a Project.

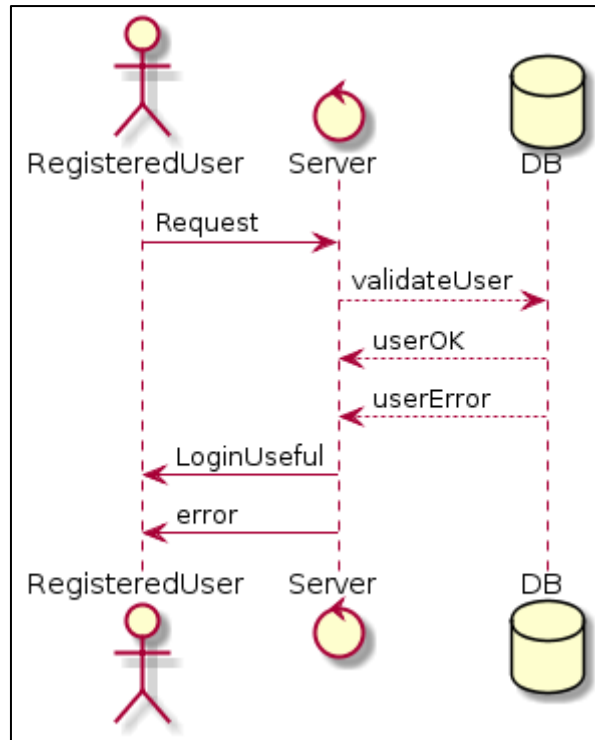


Figure 9: Sequence diagram to Login on the platform

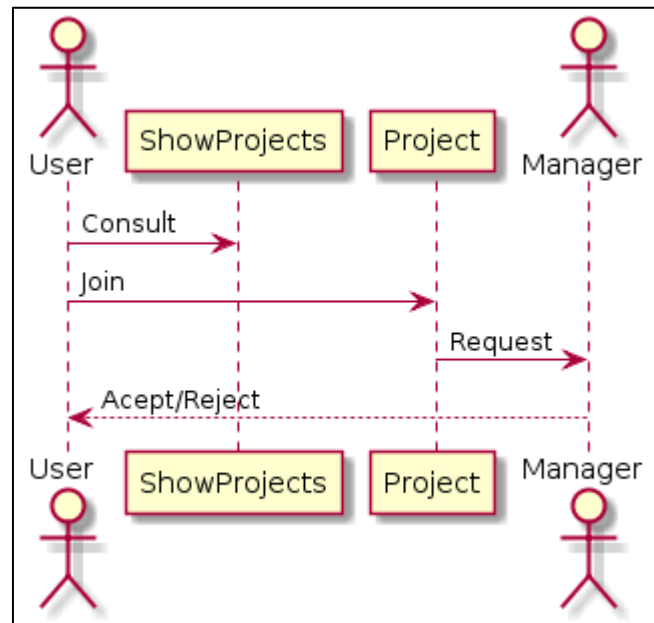


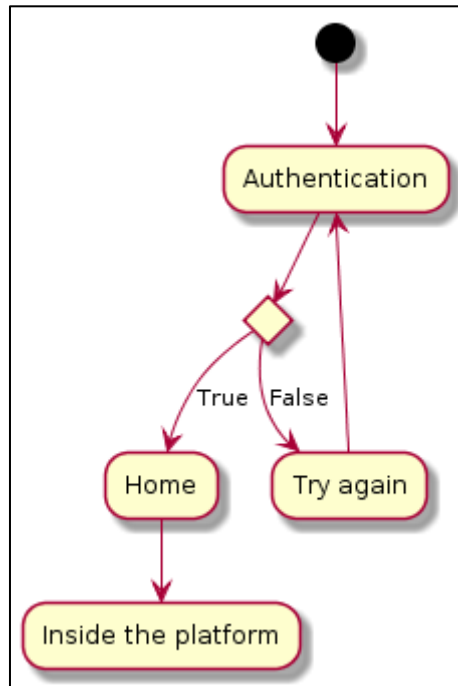
Figure 10: Sequence diagram for a Registered User to join to a Project

#### 4.2.2 Process View

This section presents the dynamics of the main activities in the system, i.e., how the different process components that build the system interact. This is described with UML activity diagrams.

#### 4.2.2.1 Login of registered user

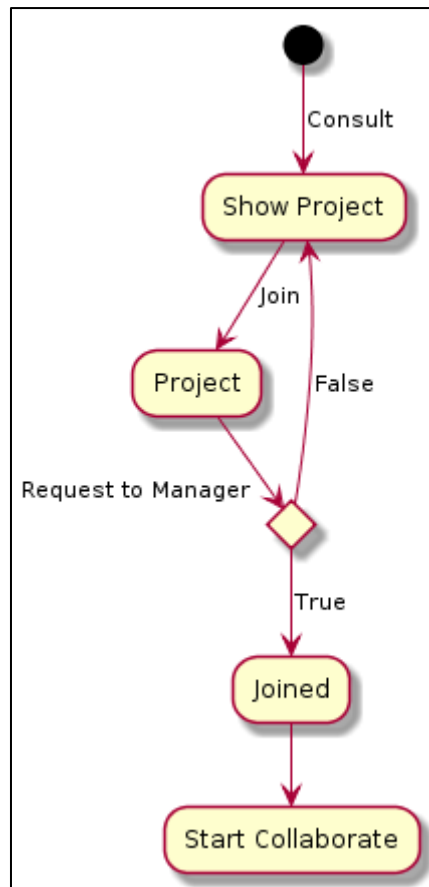
The diagram in Figure 11 shows the activity diagram of the interaction between a registered user and the process to validate the access to the system. The user sends the information to the system, which checks the user name, password and role of the user. If the information is correct, then a valid session is generated in the platform to allow the interaction of the user with the system. Figure 8 presents the sequence diagram for the login process on the platform.



**Figure 11: Activity diagram to access the platform**

#### 4.2.2.2 Join a project

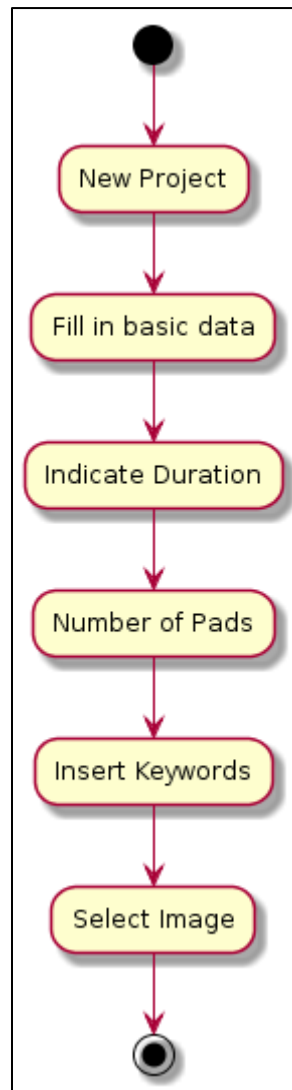
Figure 12 shows the activity diagram of the interaction between a Registered User and the process to join to a project. The user consults the information presented in “*show project*”, then the user chooses the project to join and collaborate. The manager receives the request and decides to accept it or not. Figure 10 presents the sequence diagram for this process on the platform.



**Figure 12: Activity diagram of a user to join to a project**

#### 4.2.2.3 Create a project

Figure 13 shows the activity diagram for the creation of a project. The Manager user or the Admin user can create a project. This requires them to fill in the basic data of a project, indicate the duration of the project, create the pads, insert the keywords, and select the image for the project.

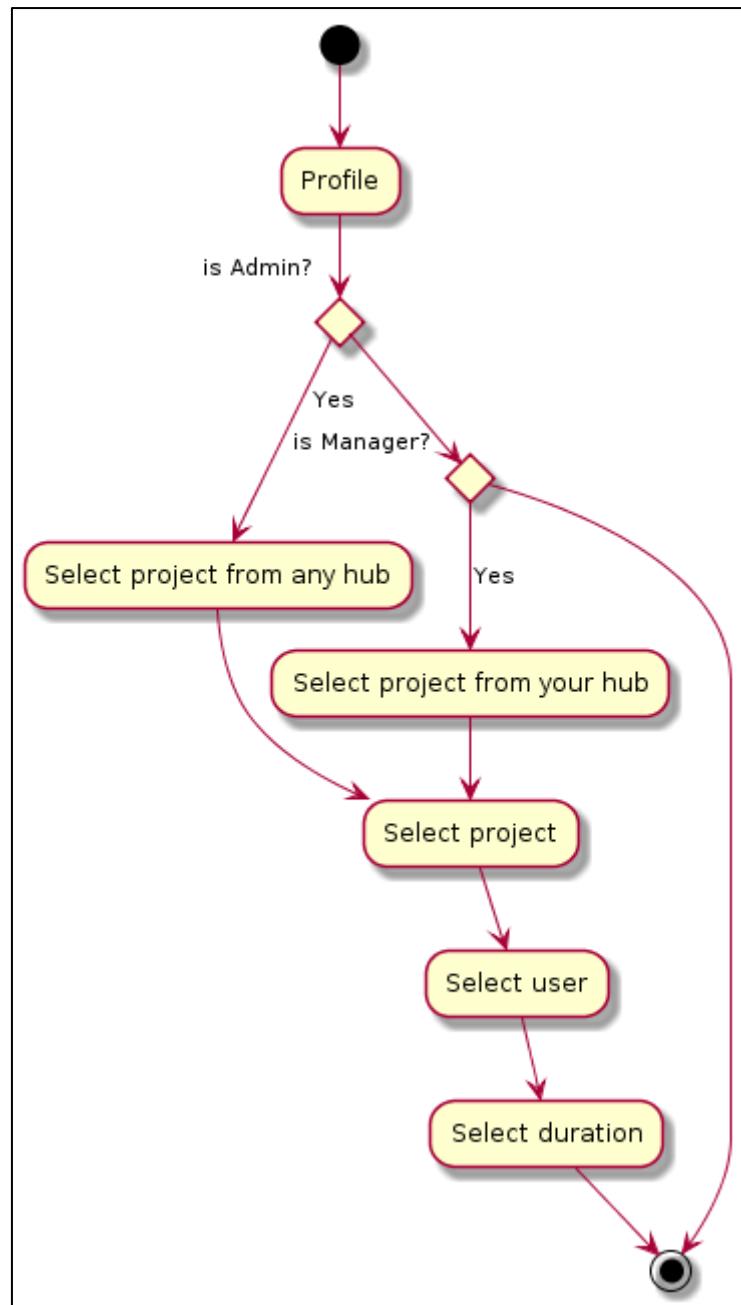


**Figure 13: Activity diagram of create project process**

#### 4.2.2.4 *Assign a project to a user*

Figure 14 shows the activity diagram for the assignment of a project to a user. Depending on the profile of the user a project and a user are selected to carry out the assignment.

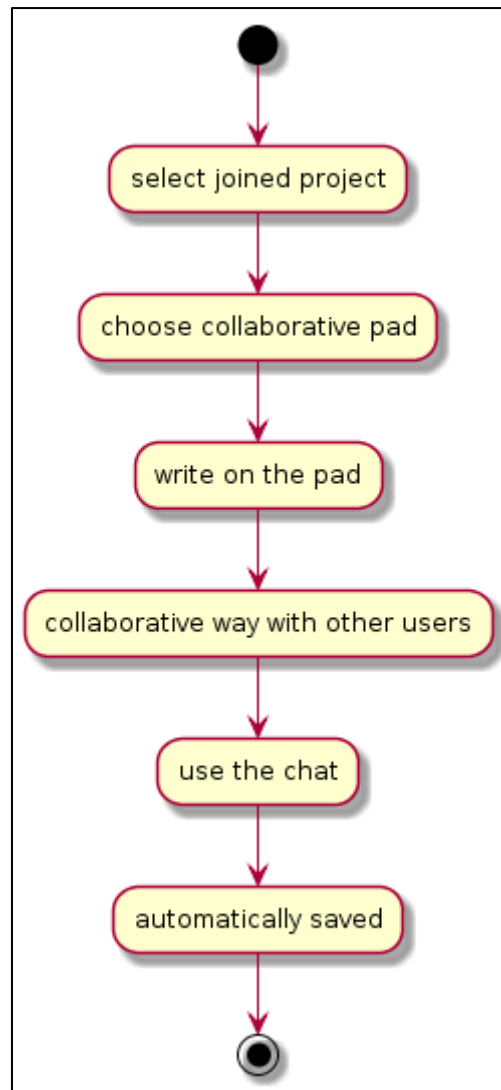




**Figure 14: Activity diagram to assign a user to a project**

#### 4.2.2.5 Participation in a Pad

Figure 15 shows the activity diagram for the participation of a user in a project. The registered users select a project where they were joined. They choose the collaborative pad where they are going to participate. They can write on the pad, collaborate with other users through the chat from the project to speak with other registered users connected at the same time. These changes are saved automatically on the platform.

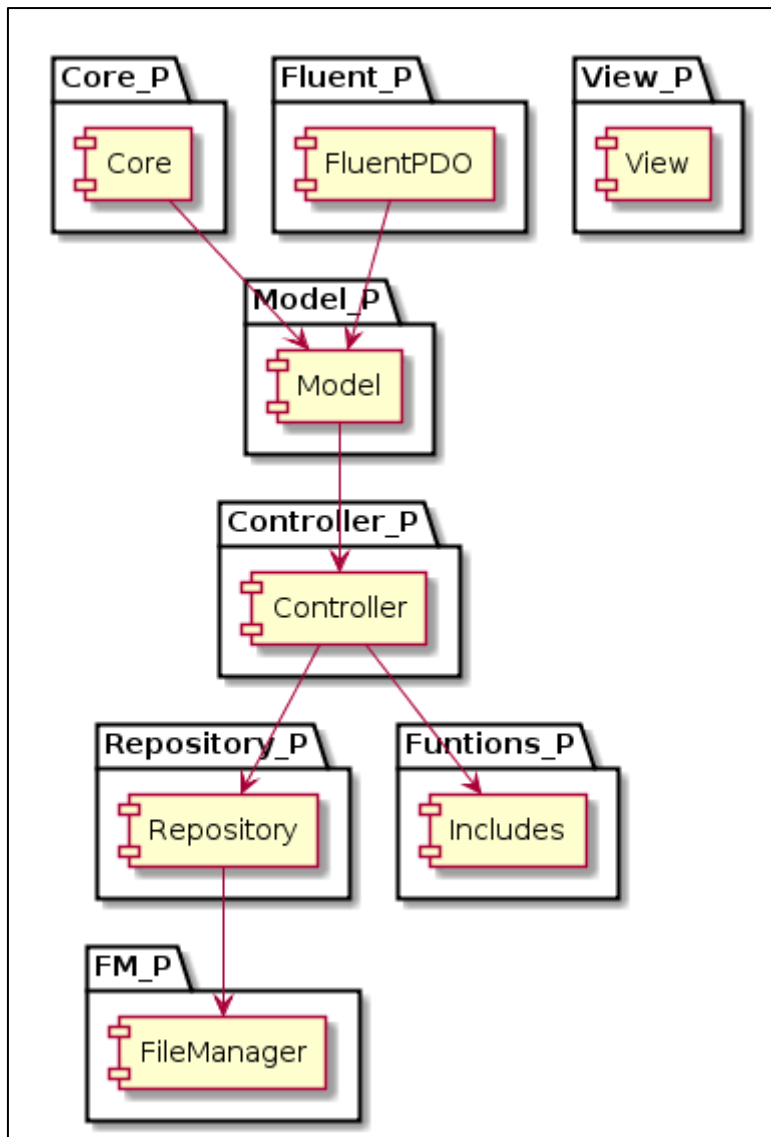


**Figure 15: Participation in a pad**

#### 4.2.3 Implementation View

This view describes how software components are built and structured. These are described with UML package diagrams.

Figure 16 presents the Package diagram of the application of the web platform. Different packages represent the software components that are used or implemented to build the platform. The *Core* package and the *FluentPDO* (PHP Data Objects to access databases) provide basic classes for implementing the model and its persistency, respectively. The *Model* package uses them in order to implement the classes of the objects that represent the different entities that are managed in the platform (which are described in section 4.2.1 and shown in Figure 6. There is also a package *View* that implements all visual components of the user interface (i.e., the generation of HTML pages). The *Controller* package has the components that process the inputs from the users, in order to consult or modify the elements of the model. There are two more packages: *Repository*, which implements the functionality of a common repository to collect files that are shared in a project, and *File manager*, with the general functions for accessing the file system, which is used by the *Repository* classes.



**Figure 16: Package diagram of the application**

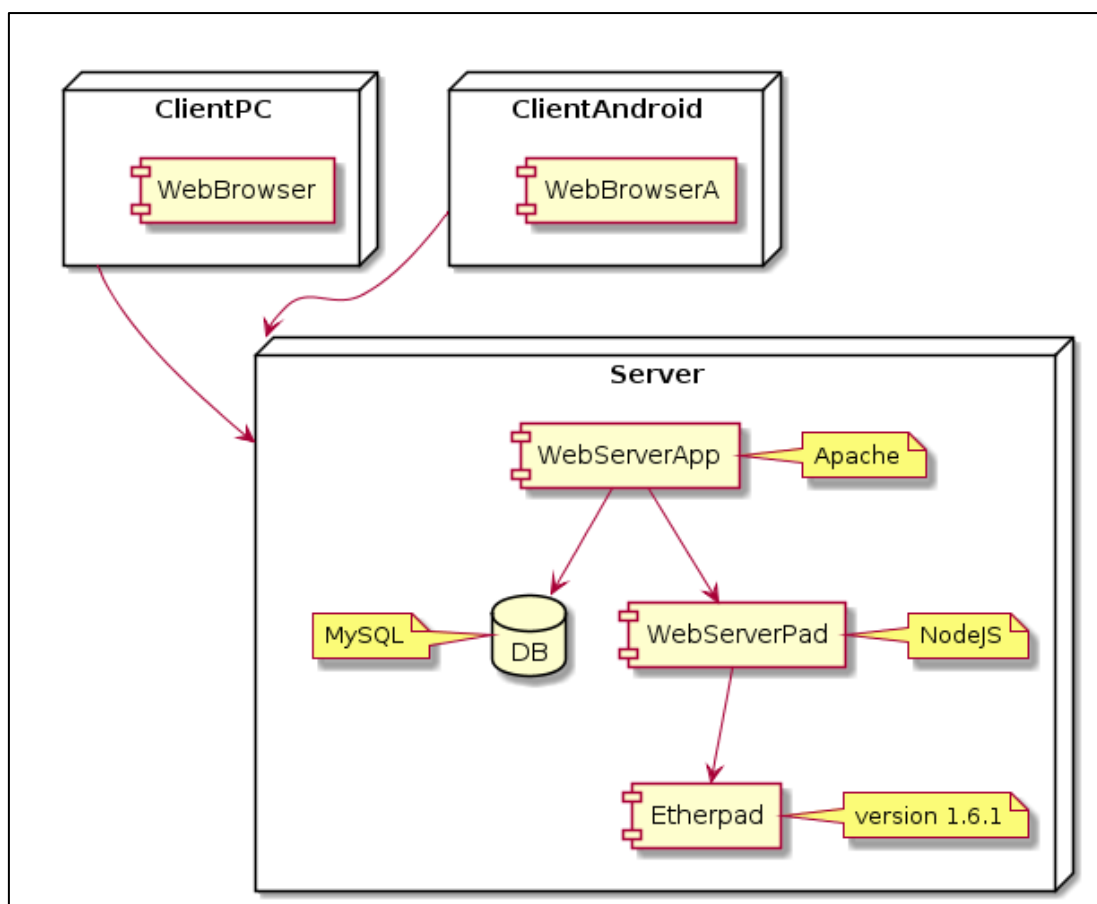
#### 4.2.4 Deployment View

The Deployment View describes how the software components of the system are deployed in different nodes in the network, and how they can be configured. This is described with UML deployment diagrams.

Figure 17 presents the deployment diagram of the web platform. A server hosts the application and provides access to different clients, which are standard web browsers from a computer or from a smartphone.

The server contains several components:

- An Apache web server with the main application logic component.
- A MySQL database server, for data persistence.
- A NodeJS server to access the component that implements the pads, which is currently Etherpad package.



**Figure 17: Deployment diagram of the application**

#### 4.2.5 Versions of the different components

The next table shows the versions of the main external components that have been used to build the platform.

Component	Version	Link
HTML, CSS	5, 3	<a href="https://www.w3.org/TR/html5/">https://www.w3.org/TR/html5/</a>
PHP 5	5.6.28	<a href="http://php.net/ChangeLog-5.php#5.6.28">http://php.net/ChangeLog-5.php#5.6.28</a>
JQuery	1.11.0	<a href="http://api.jquery.com/">http://api.jquery.com/</a>
Etherpad	1.6.1	<a href="http://etherpad.org/">http://etherpad.org/</a>
MySQL	5.6	<a href="http://dev.mysql.com/doc/refman/5.6/en/">http://dev.mysql.com/doc/refman/5.6/en/</a>
Bootstrap	3.1.1	<a href="http://getbootstrap.com/">http://getbootstrap.com/</a>

**Table 2: Versions of the main external components**

## 5 Validation and experimentation of the platform

The platform has been used to support the transitions experiments of Work Package 3 (WP3). Different topics have been considered for each transition experiment.

Each project concept has been developed using the FoTERRIS methodology through a process that involves the performance of three workshops. The platform has supported the members of the competence cell in the preparation of the workshops and tracking their development for the following activities:

- Specify the goal of the workshops.
- Manage the list of participants to each workshop, their contact data, to confirm their availability and attendance, etc.
- Organise logistic things like reservation of meeting room, prepare material for the dynamics, etc.
- Define dynamics before the workshops.
- Taking note of the dynamics of each workshop and their results.
- Review and discuss the results and conclusions after the workshops

In addition, the platform has been validated in a Master course at UCM. A group of around 35 students has been using the platform to carry out the definition of different research projects. Small work groups, of four or five persons, have been co-working on a collaborative way to achieve their objectives. Using the platform, they could edit at the same time the documents, they were able to speak using the chat tool, and they had control on the progress of the project.

### 5.1 Evaluation

The platform has been evaluated by the students from the Master course at UCM. We has launched a questionnaire to collect their opinions during their use of the platform. We have collected answers to evaluate the different functionalities of the platform from 16 students.

First, the home page and the register, login and show projects pages. Their opinion was satisfactory because they consider that the platform's interface is good, and that it is easy to register and login. They also found useful the show projects option because there they can see all projects on the platform and they can access public information. Next figures present the graphics from the general data of the participants of the questionnaire carried out in the evaluation.

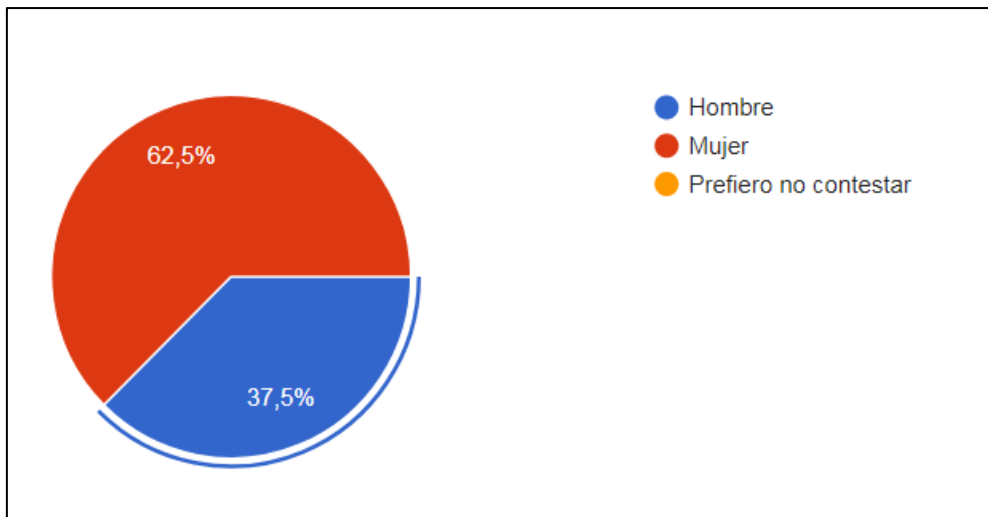


Figure 18: Gender of the participants in the evaluation (Blue: man; Red: woman; Orange: don't reply)

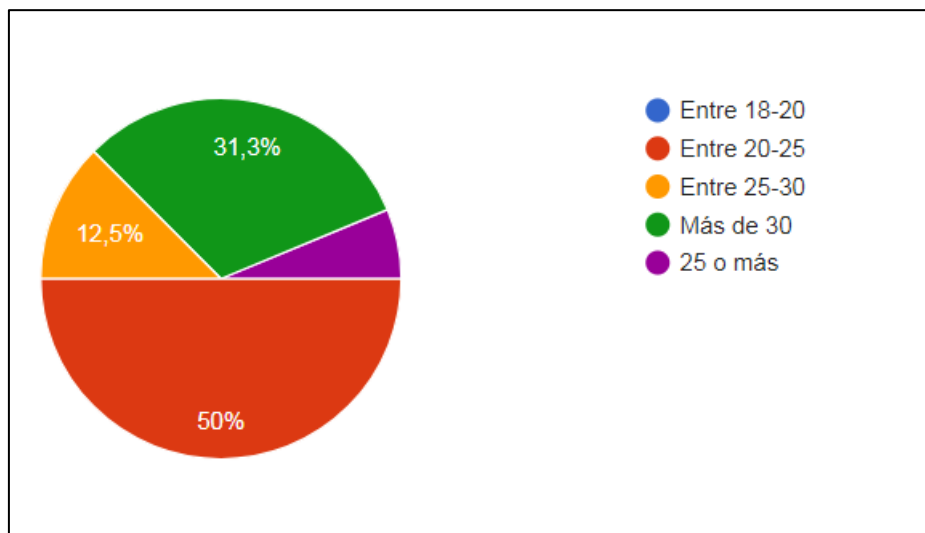


Figure 19: Age of the participants (Blue: between 18-20; Red: 20-25; Orange: 25-30; Green: More than 30; Purple: more than 25)

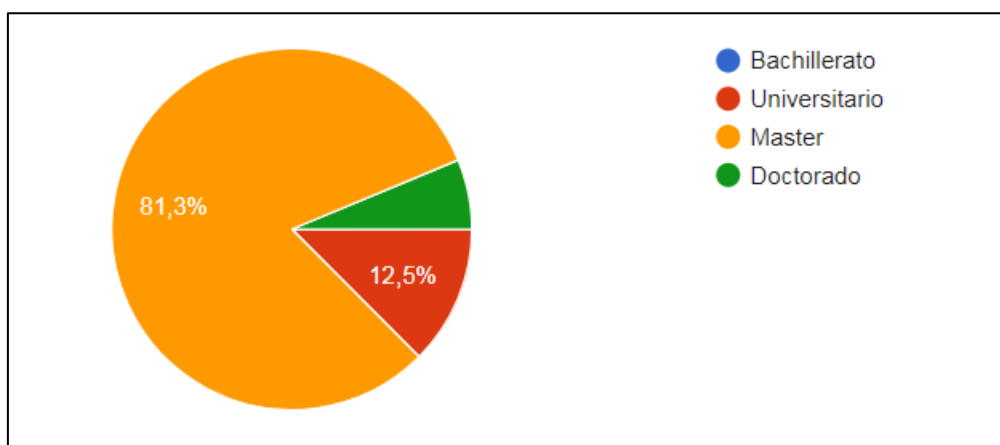
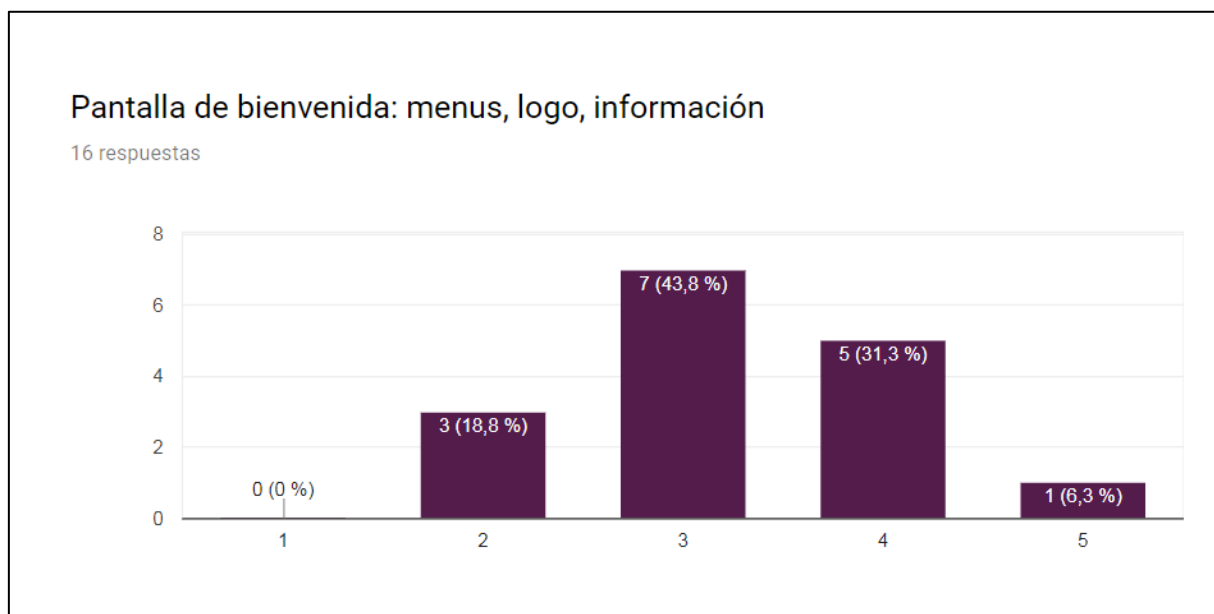
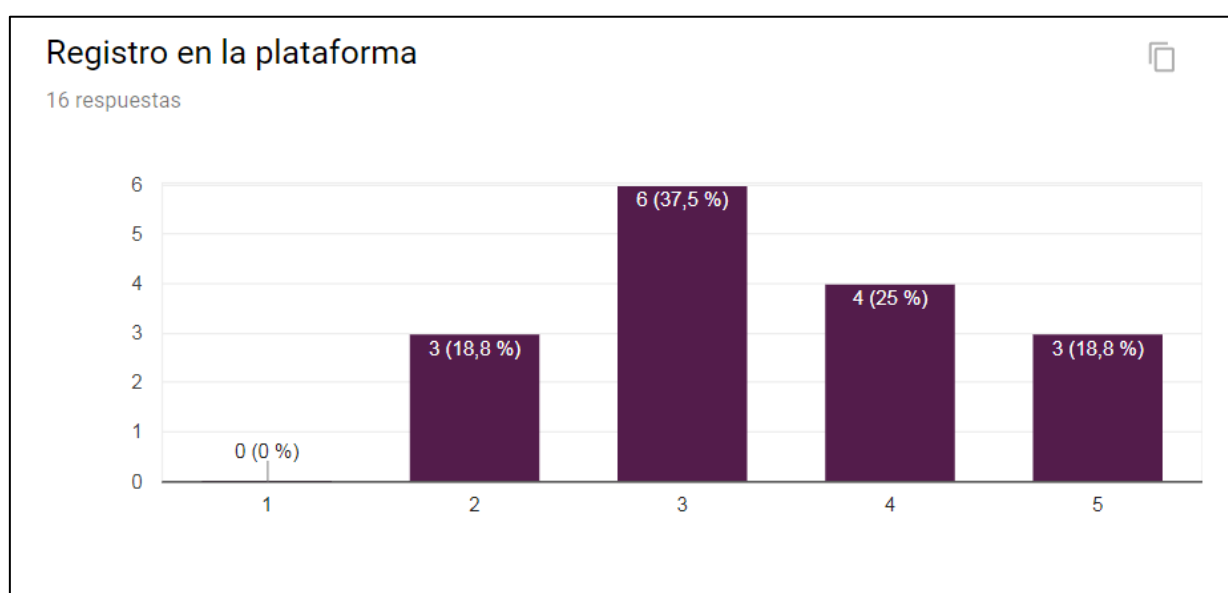


Figure 20: Level of study of the participants (Blue: High School; Red: University; Orange: Master; Green: PhD)

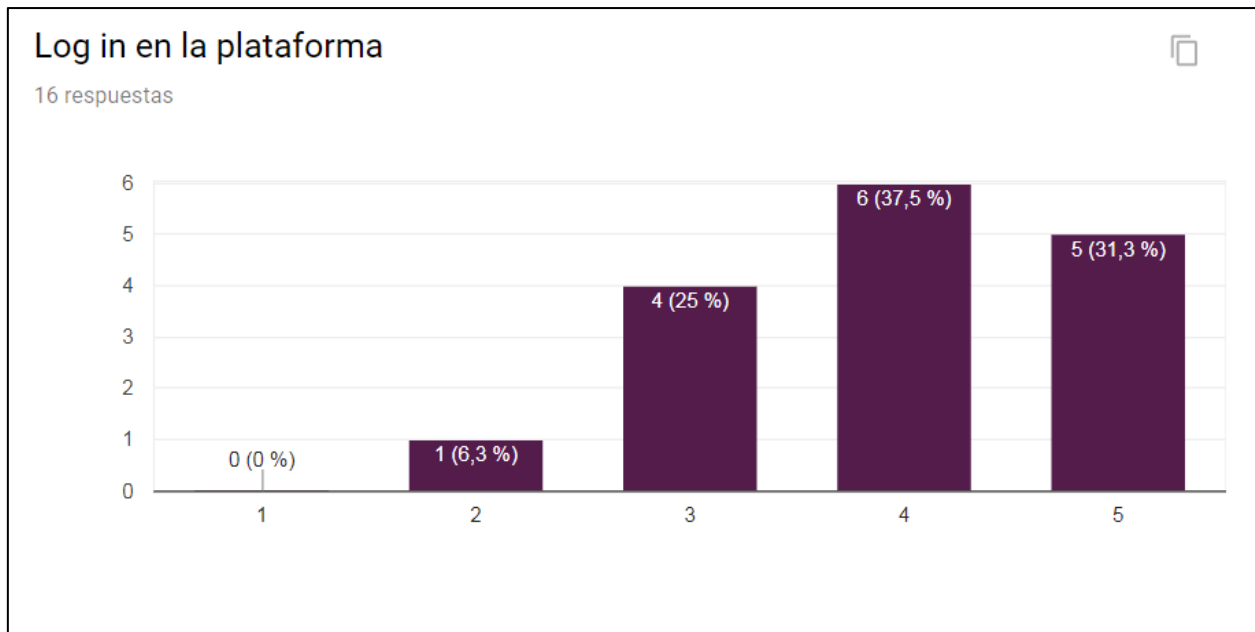
Next figures present the results of the evaluation of the platform's functionality. A Likert scale of 5 values is used to evaluate the functionality. The scale is from value 1: "not easy to use" to value 5: "very easy to use".



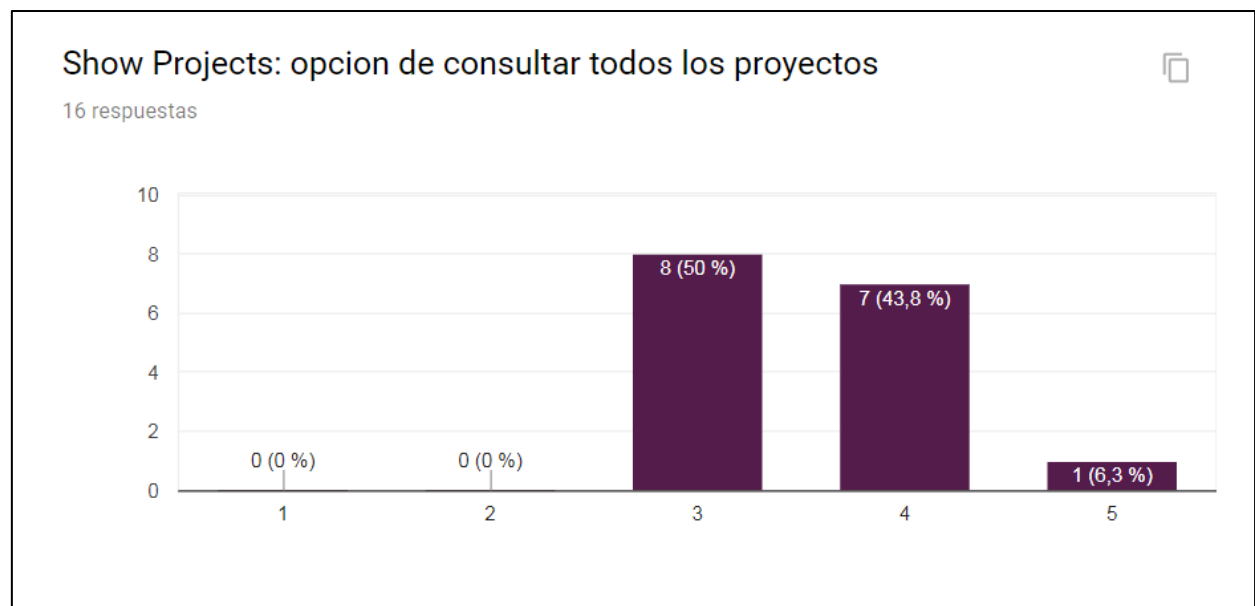
**Figure 21: Data of the evaluation of Home page (menus, logos, information) (Likert scale: value 1: not easy to use; value 5: very easy to use)**



**Figure 22: Data of the evaluation of the Registration process (Likert scale: value 1: not easy to use; value 5: very easy to use)**



**Figure 23: Data of the evaluation of the Login process (Likert scale: value 1: not easy to use; value 5: very easy to use)**



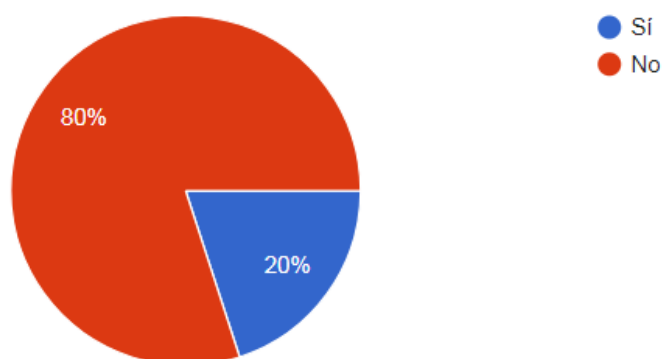
**Figure 24: Data of the evaluation of the Show project functionality (Likert scale: value 1: not easy to use; value 5: very easy to use)**

The **managers** of the projects have been asked about the use of the platform for user and project management: control of users, management of the project, definition of pads of the project, and assignment of users to the project. 20% of the students were manager of projects, and their feeling of the use of this profile was positive, in the sense that they could use this functionality without problems. The next figures present the results for the managers.



## Tu perfil en la plataforma ha sido de Manager (IP) de algún proyecto

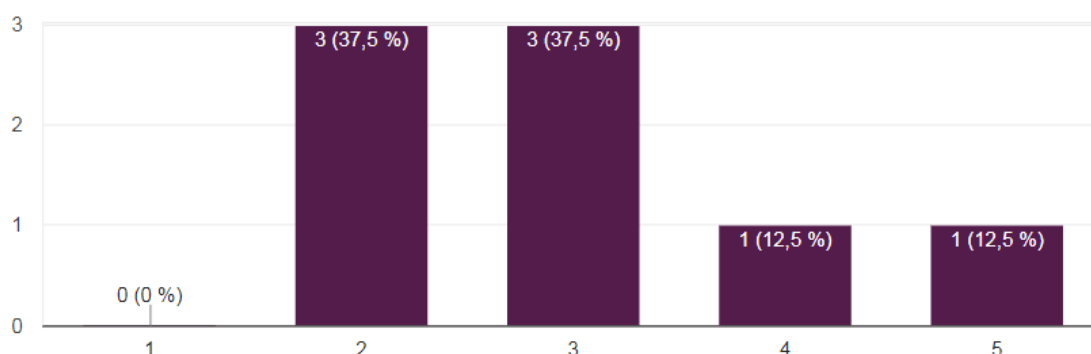
15 respuestas



**Figure 25: Profile of manager of a project (Blue: yes; Red: no)**

## Gestionar un usuario (crear, modificar)

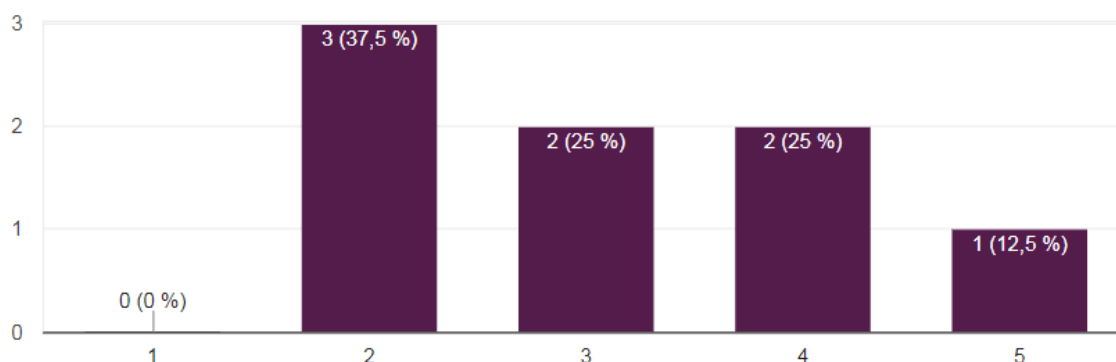
8 respuestas



**Figure 26: Functionality to manage users (Likert scale: value 1: not easy to use; value 5: very easy to use)**

### Gestionar un proyecto (crear, modificar)

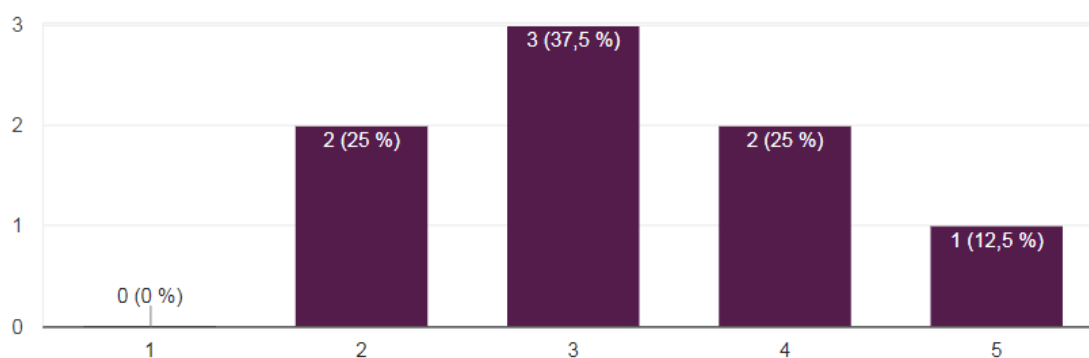
8 respuestas



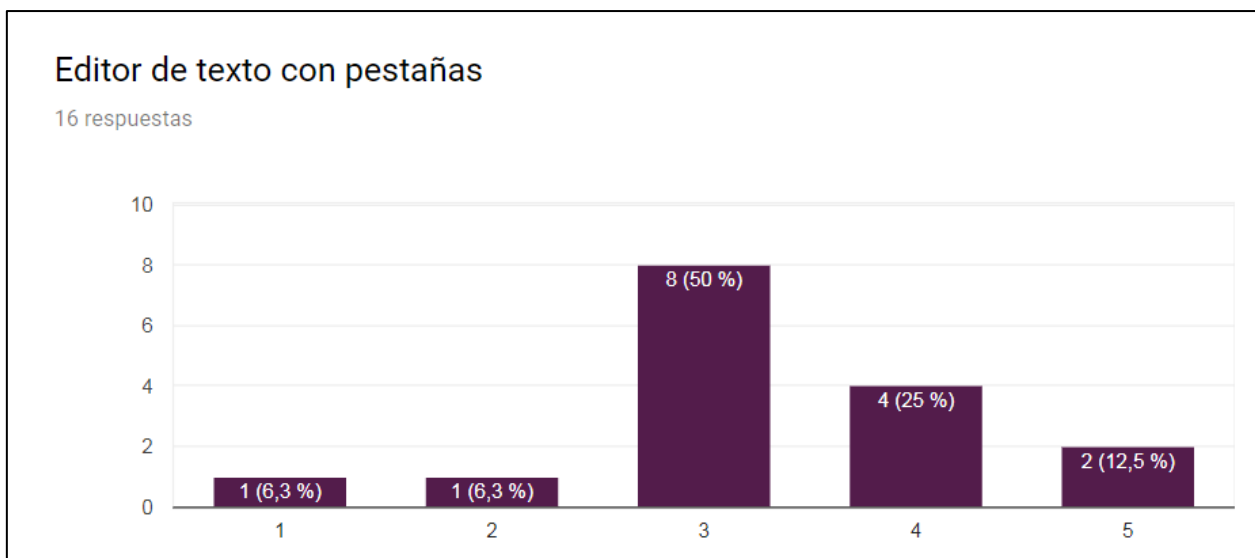
**Figure 27: Functionality to manage a project (Likert scale: value 1: not easy to use; value 5: very easy to use)**

### Gestionar Asignación de usuarios a proyectos (crear, modificar)

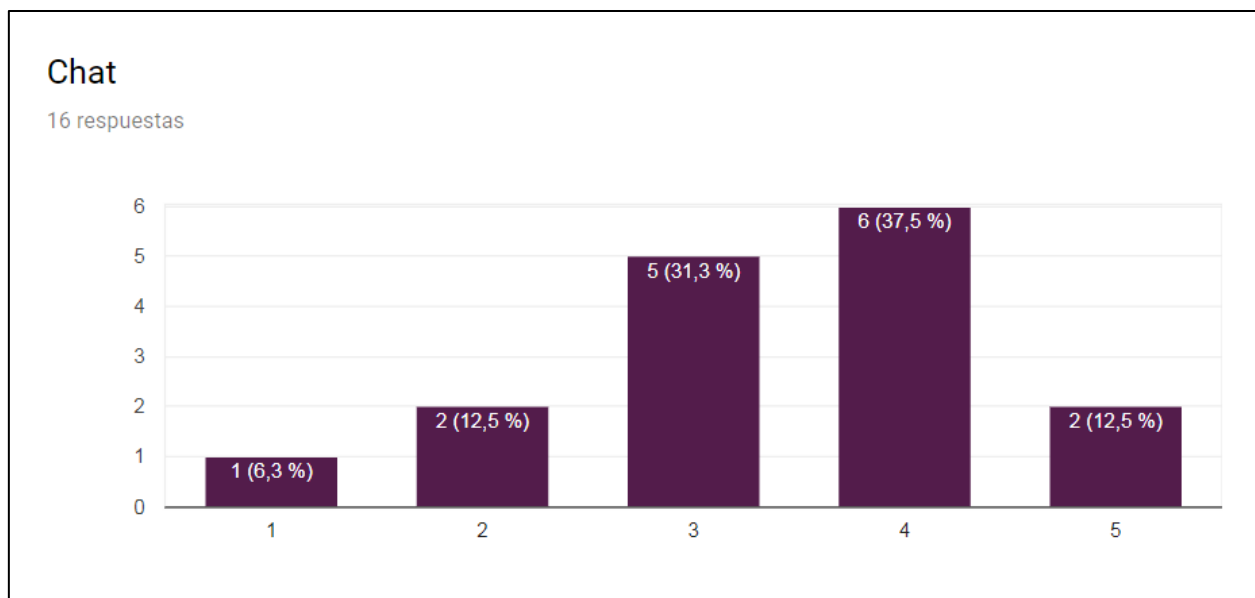
8 respuestas



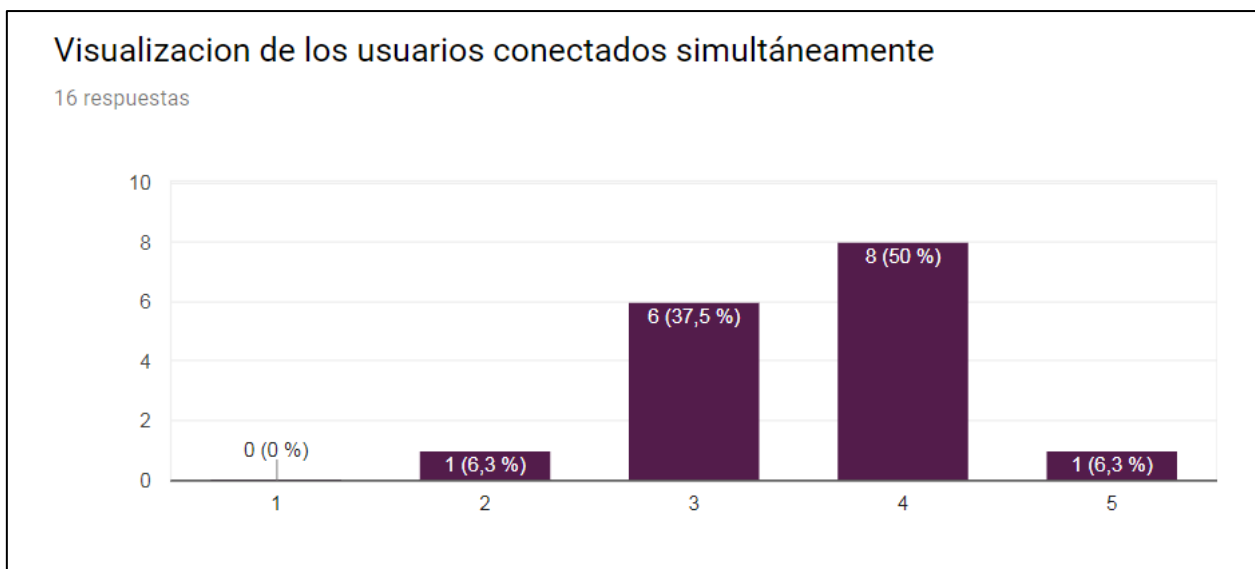
**Figure 28: Functionality to manage the assignment of a project to users (Likert scale: value 1: not easy to use; value 5: very easy to use)**



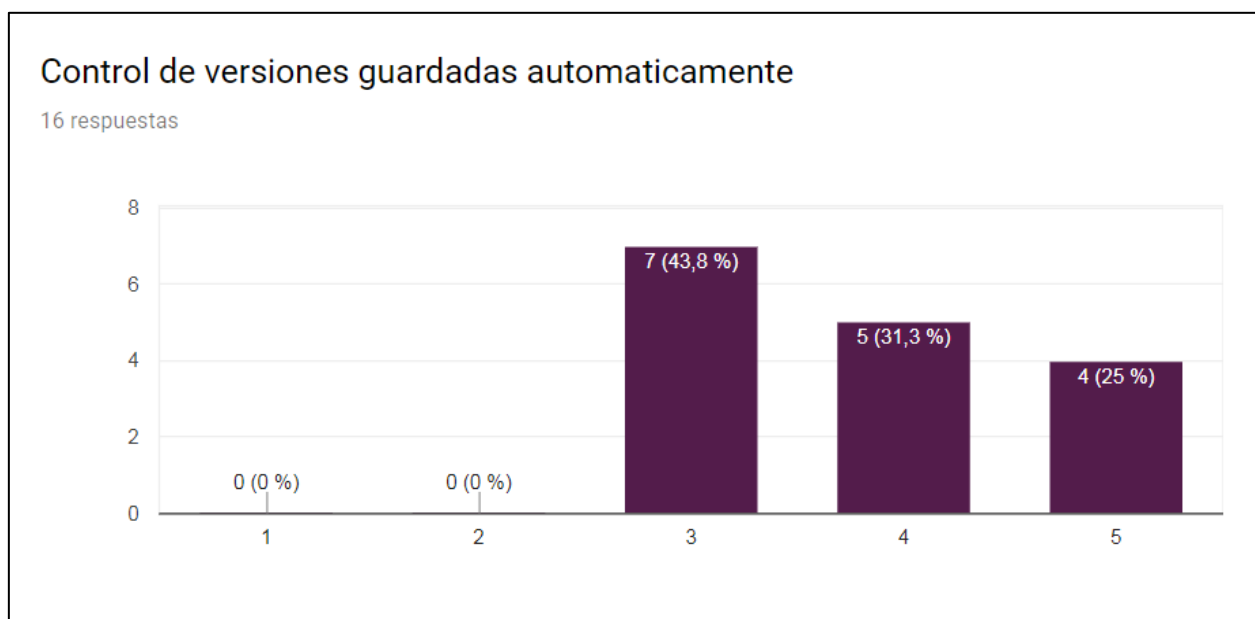
**Figure 29: Functionality to write on a pad (Likert scale: value 1: not easy to use; value 5: very easy to use)**



**Figure 30: Functionality to use the chat of the platform (Likert scale: value 1: not easy to use; value 5: very easy to use)**



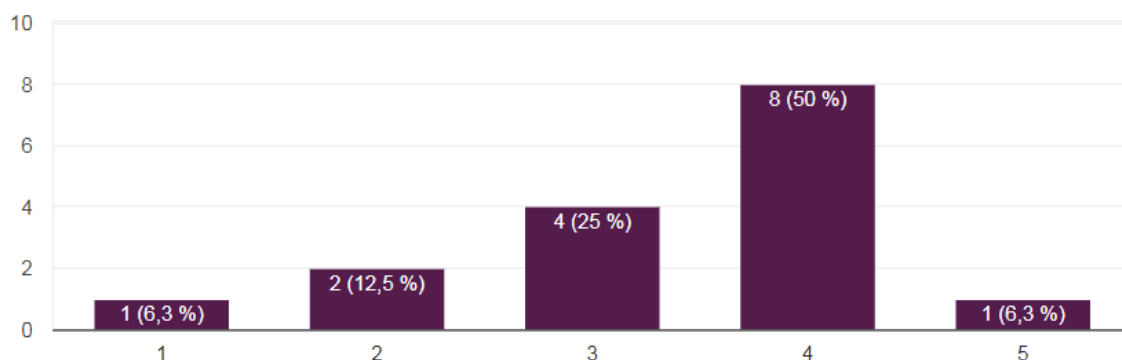
**Figure 31: Functionality to see the connected users at a given time in the project (Likert scale: value 1: not easy to use; value 5: very easy to use)**



**Figure 32: Functionality to save versions automatically (Likert scale: value 1: not easy to use; value 5: very easy to use)**

### ¿Es fácil usar la plataforma?

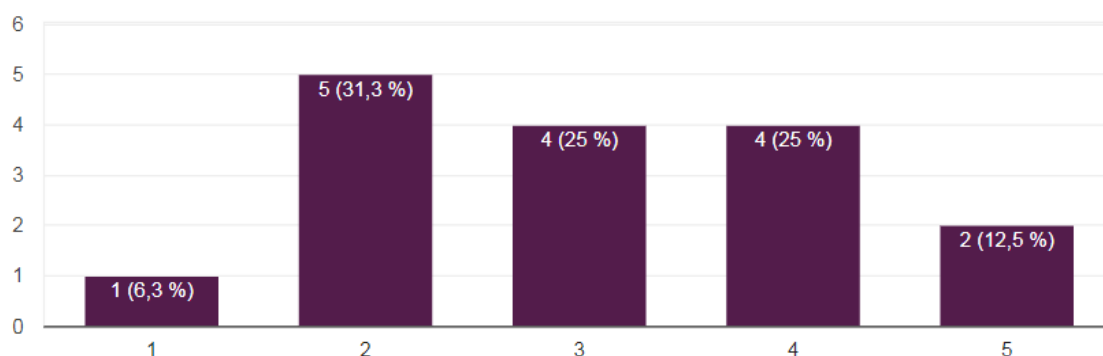
16 respuestas



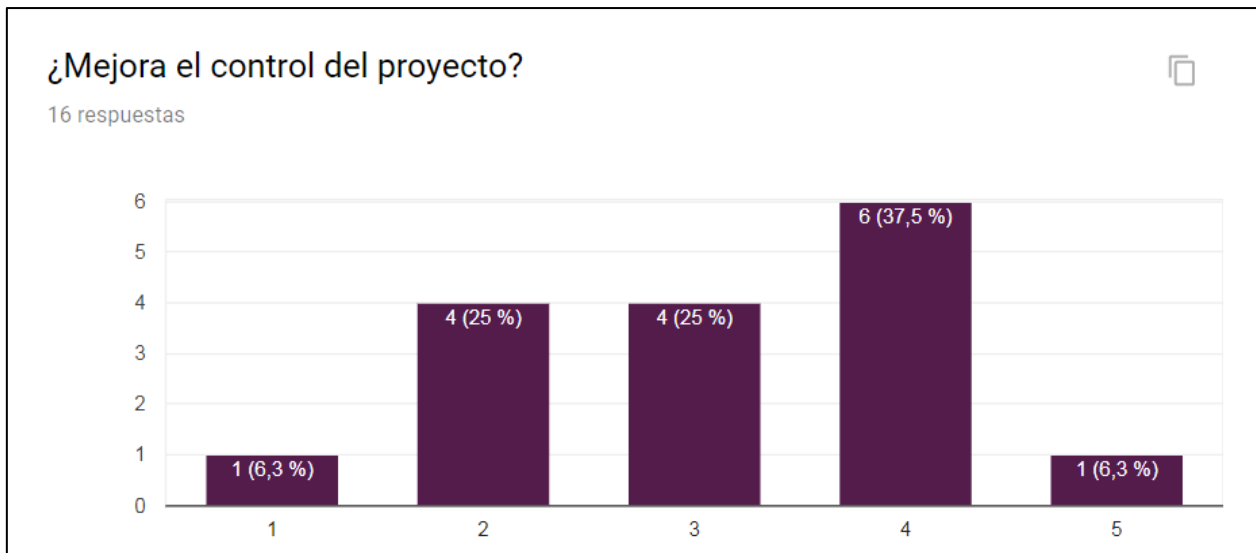
**Figure 33: General use of the platform. Is it easy?** (Likert scale: value 1: not easy to use; value 5: very easy to use)

### ¿Es útil para gestionar el control del proyecto?

16 respuestas



**Figure 34: General view: Is the platform useful to manage a project?** (Likert scale: value 1: not usefulness; value 5: very usefulness)



**Figure 35: General use: Does it improves the control of the project? (Likert scale: value 1: not easy to use; value 5: very easy to use)**

In summary, analysing the use of the platform during this course of the Master, more than 30% of the users evaluated the experience as very positive or positive. The students could access to projects and work together in a collaborative way. They appreciated the use of this collaborative online tool because it improves communication, centralises the work and organises tasks and responsibilities.

In addition, the platform was evaluated by each member of the competence cell at UCM. The platform was used before, during and after each workshop of the different transition experiments. The FoTERRIS web-platform supported the coordination of the preparatory information, the invitations, the budget and the contents. This has been considered useful to control the list of participants and different aspects of logistics (like material) and organize presentations and dynamics. During the workshops the use of platform was satisfactory because it supported to the competence cell members taking notes at the same time to compile in a collaborative way the results of each workshop. After the workshop these notes were reviewed by the experts and the competence cell members to generate the report and the public information of the project.

## 5.2 Accessibility Check

A validation for web accessibility has been carried out on the web platform to check whether it is compliant with the Level AA Conformance to Web Content Accessibility Guidelines 2.0 (WCAG 2.0) of W3C<sup>2</sup>. These guidelines explain how to make web content accessible to people with disabilities. Level AA means that the site deals with the biggest and most common barriers for disabled users. In order to check that the FoTERRIS web platform satisfies the criteria defined by the W3C guidelines, the tool AChecker<sup>3</sup> has been used. This tool checks single HTML pages for conformance with accessibility standards to ensure the content can be accessed by everyone.

AChecker is used to evaluate HTML content for accessibility problems by entering the location of a web page, uploading an HTML file, or by pasting the complete HTML source code from a Web page. It produces a report of all accessibility problems for the selected guidelines. More specifically, AChecker identifies 3 types of problems:

- **Known problems:** These are problems that have been identified with certainty as accessibility barriers. They require a modification of the page to fix them.
- **Likely problems:** These are problems that have been identified as probable barriers, but require a human to make a decision. They will likely need to modify the corresponding page to fix these problems.
- **Potential problems:** These are problems that AChecker cannot identify, that require a human decision. They may require a modification of the page, but in many cases it is just needed to confirm that the problem described is not present.

This tool gives an accessibility review with comments to improve the design and content of the page. The next list presents some examples of changes that have been carried out:

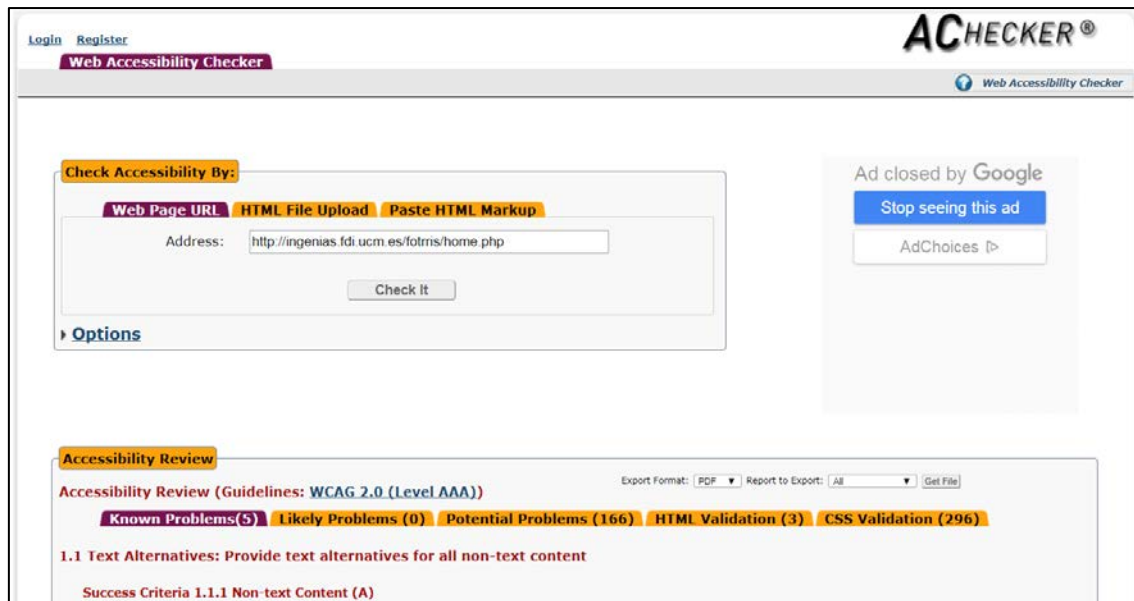
- *Text Alternatives:* Provide text alternatives for any non-text content (icons, figures, etc.)
- *Distinguishable:* Make it easier for users to see and hear content including separating foreground from background.
- *Navigable:* Provide ways to help users navigate, find content, and determine where they are.
- *Adaptable:* Create contents that can be presented in different ways (for example, simpler layout) without losing information or structure.
- *Keyboard Accessible:* Make all functionality available from a keyboard.
- *Predictable:* Make Web pages appear and operate in predictable ways.
- *Readable:* Make text content readable and understandable.

Figure 36 presents an example of the tool's interface to check accessibility.

---

<sup>2</sup> <https://www.w3.org/TR/WCAG20/>

<sup>3</sup> <https://achecker.ca/checker/index.php>



**Figure 36: Example of accessibility checking**

In the current version, the web site has been validated to pass Level AA, which means that the Web page satisfies all the Level A, the minimum level of conformance, as well as the Level AA Success Criteria that are specified in WCAG 2.0.

Table 3 shows some issues that have been identified with the Achecker tool and a summary of the correction process on the Web page.

Guideline	Summary of corrective actions
Non-text Content	Provide text alternatives for non-text content
Captions (Pre-recorded)	Provide captions for videos with audio
Info and Relationships	Logical structure
Meaningful Sequence	Present content in a meaningful order
Use of Colour	Don't use presentation that relies solely on colour
Keyboard	Accessible by keyboard only
Page Titled	Use helpful and clear page titles
Headings and Labels	Use clear headings and labels
Consistent Navigation	Use menus consistently
Error Prevention (Legal, Financial, Data)	Reduce the risk of input errors for sensitive data

**Table 3: Summary of issues to comply with conformance levels for accessibility A and AA**



### **5.3 Gender Check**

A gender check (Thaler 2010) of the web site has been carried out during the design and implementation of the web platform. The IFZ team, FoTERRIS partner from Austria, performed it to improve the final version of the web platform.

The gender check has reviewed that the pages of the web site are gender reflexive and contain no gender biases:

- Pictures of the web site. This refers to those generic of the platform. It does not include the pictures of each project, because these are uploaded by the users when creating new projects.
- Phrases and language, in terms of gender reflexivity, i.e., avoiding stereotypes, avoiding hetero-normative or hegemonic male positions.
- The assignment of functionality and roles does not depend on gender.

## 6 Distribution and sustainability of the platform

The FoTERRIS web-based platform will be supported by the FoTERRIS project during its lifetime at <http://ingenias.fdi.ucm.es/fotrris/home.php>. After the project, it will be supported and maintained by the UCM-GRASIA research group in this server to facilitate the co-creation of RRI projects by interested groups. Initially the UCM group has resources to support it during at least two years after the project. Later on, the use and evolution of the platform will determine how to maintain it.

On the other side, the software has been published as open source in GitHub (see section 6.1), so other groups and interested people can install their own instance of the platform, as well as collaborate to improve the software. Its success will depend on the ability to create an active community around this open source project. This approach will also allow the open community to improve the platform and adapt it to future needs or restrictions.

### 6.1 GitHub

The FoTERRIS web based platform is delivered as open source in the GitHub repository of the UCM-GRASIA research group:

- Web Module: web application (*fotrrisweb*)
  - Repository at <https://github.com/grasia/fotrrisweb.git>
- Collaborative Module: collaborative content server (*fotrrisserver*).
  - Repository at <https://github.com/grasia/fotrrisserver.git>

### 6.2 Installation of the platform

It is possible to carry out the deployment of the platform in two ways:

- **A manual installation** that would involve the installation of all the dependencies and libraries of the web application, on a machine whose recommended operating system is Ubuntu Server 16.
- **An installation using docker compose**, which will allow the creation of Linux containers to build the right environment for the implementation of the application.

For both installations, it is recommended to use a machine with operating system **Ubuntu 16.xx** or higher.

#### 6.2.1 Manual Installation

To install the platform step by step, there is a file with the instructions to follow in the script */fotrrisweb/install/install\_script\_manuel.sh*

In this script, all the instructions for dependencies' installation (useful to the commissioning of the platform) are collated.

##### a) Installation Database

It will be necessary to create the database farm. To do this, open a terminal and access the created account of mysql (administrator or root). The password will be the one that was designated during the installation.

```
$ mysql -u root -p
```

```
$ create database database name;
$ show databases;
$ pwd
$ source /home/yourdirectory/fotrrisweb/db/db_backup_XXXX.sql
$ show tables;
$ exit;
```

Note: XXXX corresponds to the most current version of the created backup copy of the application.

b) Installation *FotrrisWeb*

When all the necessary dependencies have been installed, copy the contents of the "FotrrisWeb" repository to the path "/var/www/html/.." so that Apache can find the corresponding sources.

c) Installation *FotrrisServer*

FotrrisServer uses NodeJs as a web server. After downloading the repository and placing it in a route that you consider appropriate, proceed to raise the server with the following instructions:

```
$ cd fotrrisserver
$ nohup bin/run.sh &
```

### 6.2.2 *Installation using Docker compose*

To deploy the application with Docker compose, it is necessary to perform the following steps:

1. Install docker-compose on the machine where the application will host.
2. Download the source code of the official repositories of the project (section 6.2.3) and place them in the directories where it has been considered appropriate.
3. Edit the file *fotrrisweb/install/docker-compose.yml* on the next line and add the path where the fotrrisweb source code has been located.

```
tomcat:
```

```
...
```

```
volumes:
```

```
- / your_directory / fotrrisweb: / var / www / html
```

It is possible to modify the map of external ports: internal ports of the proposed configuration:

```
tomcat:
```

```
ports:
```

```
- 2401: 80
```

4. Modify in the same way the path where the *fotrrisserver* source code will be located.

```
node:
```

```
...
```

```
volumes:
```

```
- / your_directory / fotrrisserver: / srv / fotrrisserver
```

### 6.2.3 Official Repository

Download the project's official repository for both *fotrrisweb* and *fotrrisserver*:

- <https://github.com/grasia/fotrrisweb.git>
- <https://github.com/grasia/fotrrisserver.git>

Copy the contents of the *fotrrisweb* directory into the path `/var/www/html/` of the machine with the Apache service. For example:

```
$ cp -r fotrrisweb/* /var/www/html/
```

Start the Apache server if it is not active.

Start the *fotrrisserver* server

```
$ cd fotrrisserver
```

```
$ nohup bin/run.sh &
```

Once the installation is finished and the server is started, the administration has to create some hub manager, who will be responsible of creating new projects. The deliverable D2.2 [D22] provides a description of how to start using the FoTTRIS web based platform for the different types of users.

## 7 Conclusions

This document has presented the technical information related to the co-RRI web-based platform of the FoTERRIS project (another document, Deliverable 2.2: *User Manual of FoTERRIS co-RRI web Platform*, provides the guidelines for using the platform). It targets mainly software engineers and technicians who are interested in reusing or installing the platform. As such, the structure of chapters 3 and 4 follows a typical software engineering documentation structure and use standard artefacts (UML). Chapter 6 explains how to install the platform by any interested competence cell, to manage on their own.

The development process has been iterative based on the evolution of the project over time. First step was identified and discussed a specification of the requirements of the application. Later, these requirements were adapted in different phases of the project to improve the functionality of the application and cover the needs of the different cases of study. Deliverable D2.2 "*User Manual of co-RRI web platform FoTERRIS*" presents the user interfaces in detail.

Other parts of the document address a more general public. In concrete, chapter 2 provides an extensive review of the state of the art on collaborative platforms, which has been used as a starting point before the design and implementation of the FoTERRIS platform. This has been useful in the identification of the functionality of the platform (chapter 3) and in the identification of some software components that have been integrated (e.g., Etherpad).

Chapter 5 has presented the evaluation of the platform. This has been performed with two groups of users. Initially with students of a MS course where the tool has been extensively used, and with the members of the UCM competence cell. The results show that the usability of the platform is high (more than 30% of the answers are positive or very positive), and the functionality has been considered sufficient for supporting the transition experiments (this has been tested in two transition experiments, one on the subject of Women and disability, and other on Refugees). Some improvements have been identified from these experiences. For instance, in the future it is possible to consider using a tool to recognise the voice and try to use on the platform, helping to collaborate with some users with special needs, or new functionality to look for experts with specific interests or capabilities.

Two aspects have been considered with care in the design of the platform: accessibility and gender reflexivity. The first has been tested against the accessibility standards of W3C, and the platform complies with AA level. It is also responsive, as it has been implemented with a responsive layout. Although there are no tools for gender reflexivity checking, the expertise in this subject of the IFZ team has been used to make a manual review of this aspect.

Finally, some considerations on the sustainability and distribution of the platform are presented. This implies the support for installation of the platform by different organisations. This has been accomplished by providing a docker implementation, which is easy to deploy and run on most operating systems, without too much concern on system dependencies. Furthermore, the software is distributed as open source in a GitHub repository, so it can be adapted to specific needs. This also facilitates the collaboration of the open source community in the evolution of the platform.

## 8 Bibliography

- [Kruchten 1995] Kruchten, Philippe (1995, November). [Architectural Blueprints — The “4+1” View Model of Software Architecture](#). IEEE Software 12 (6), pp. 42-50.
- [D12] Deliverable 1.2: Conceptual Framework for co-RRI. FoTERRIS project (05/12/2016)
- [D22] Deliverable 2.2: User Manual of FoTERRIS co-RRI web Platform. FoTERRIS project. (18/12/2017)
- [Nielsen 2006] Nielsen, Jakob (2006) "The 90-9-1 Rule for Participation Inequality in Social Media and Online Communities". <http://www.nngroup.com/articles/participation-inequality/> (Accessed 19/12/2017)
- [Assembl 2017] Assembl. [assembl.bluenove.com/](http://assembl.bluenove.com/). (Accessed 19/12/2017)
- [Basecamp] Basecamp. [basecamp.com/](http://basecamp.com/). (Accessed 19/12/2017)
- [CiviCRM] CiviCRM. [civicrm.org/](http://civicrm.org/). (Accessed 19/12/2017)
- [Coliga] Coliga. <http://www.coliga.co/> (Accessed 19/12/2017)
- [Colony] Colony. <http://www.colony.io/> (Accessed 19/12/2017)
- [Cobudget] Cobudget. <http://cobudget.co/>. (Accessed 19/12/2017)
- [Comakery] Comakery. [www.comakery.com/](http://www.comakery.com/). (Accessed 19/12/2017)
- [Drutopia] Drutopia. [drutopia.org/](http://drutopia.org/). (Accessed 19/12/2017)
- [Hylo] Hylo. [www.hylo.com/](http://www.hylo.com/). (Accessed 19/12/2017)
- [LinesetAI2012] Lines, M., Ambler, S.W., (2012). Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. IBM Press. pp. 54. ISBN 978-0-13-281013-5.
- [Loomio] Loomio. [www.loomio.org/](http://www.loomio.org/). (Accessed 19/12/2017)
- [Metamaps] Metamaps. [metamaps.cc/](http://metamaps.cc/). (Accessed 19/12/2017)
- [Mighty Networks] Mighty Networks. [mightynetworks.com/](http://mightynetworks.com/). (Accessed 19/12/2017)
- [Mobilize] Mobilize. [mobilize.io/](http://mobilize.io/). (Accessed 19/12/2017)
- [NationBuilder] NationBuilder. [nationbuilder.com/](http://nationbuilder.com/). (Accessed 19/12/2017)
- [Obrien2017] O'Brien, C. (2017, July 14). How NationBuilder's platform steered Macron's En Marche, Trump, and Brexit campaigns to victory. [Web log post]. Retrieved September 27, 2017, from <http://venturebeat.com>. (Accessed 19/12/2017)
- [OpenCollective] OpenCollective. [opencollective.com/](http://opencollective.com/). (Accessed 19/12/2017)
- [OurGoods] OurGoods. [www.ourgoods.org/](http://www.ourgoods.org/). (Accessed 19/12/2017)
- [Part-Up] Part-Up. [part-up.com/](http://part-up.com/). (Accessed 19/12/2017)
- [Slack] Slack. [slack.com/](http://slack.com/). (Accessed 19/12/2017)
- [Taiga] Taiga. [taiga.io/](http://taiga.io/). (Accessed 19/12/2017)

[Taiga tribe] Taiga tribe. <https://tribe.taiga.io/> (Accessed 19/12/2017)

[Taproot] Taproot. [www.taprootfoundation.org/](http://www.taprootfoundation.org/). (Accessed 19/12/2017)

[Teem] Teem. [teem.works/](http://teem.works/). (Accessed 19/12/2017)

## Appendix 1: Database schema

The schema of the database is **fotrrisdb** and consists of the following tables, which are described in the following sections:

- arena
- arena\_manager
- assigned\_project\_user
- group\_pad
- keywords
- keywords\_project
- login\_attempts
- perfil
- project
- store
- user

### Table arena

Table arena stores all information related to the hubs created.

**Table 4: Table arena**

Column	Type	Default	Links to	Comment
id	Number 11			Id arena
name	Varchar 100			Name of arena
place	Varchar 100			Place of arena
address	Varchar 100			Address of arena
responsible	Varchar 100			Responsible or Arena
id_user	Number 11			

### Table arena\_manager

Table arena\_manager stores the relation between the hubs and the managers who will control these hubs.

**Table 5: Table arena\_manager**

Column	Type	Default	Links to	Comment
id_arena	Number 11		arena.id	Id arena for arena
id_user	Number 11		user.id	Id user for arena

### Table assigned\_project\_user

Table assigned\_project\_user keeps the information associated with the projects assigned to each user of the application.

**Table 6: Table assigned\_project\_user**

Column	Type	Default	Links to	Comment
id	Number 11			Id assigned project
id_project	Number 11		project.id	Id project assigned
id_user	Number 11		user.id	Id user assigned
date_from	Date			Date from assigned project
date_to	Date			Date to assigned project



## Table group\_pad

Table group\_pad contains the information related to the pads of each project.

**Table 7: Table group\_pad**

Column	Type	Default	Links to	Comment
id	Number 11			Id group pad
pad_name	Varchar 50			Pad name
filename_config	Varchar 50			Filename settings
id_project	Number 11		project.id	Id project for pad
pad_name_view	Varchar 150			Pad name for view
description	Varchar 300			Description for pad

## Table keywords

Table keywords stores the set of keywords belongs to each project.

**Table 8: Table keywords**

Column	Type	Default	Links to	Comment
id	Number 11			Id keyword
keyword	Varchar 50			Description keyword

## Table keywords\_project

Table keywords\_project keeps the relation between the keywords and each project.

**Table 9: Table keywords\_project**

Column	Type	Default	Links to	Comment
id_keyword	Number 11		keyword.id	Id keyword
id_project	Number 11		project.id	Id project

## Table login\_attempts

Table login\_attempts stores the information of the number of times to log in the application by one user.

**Table 10: Table login\_attempts**

Column	Type	Default	Links to	Comment
id_user	Number 11		user.id	Id user login
time	Varchar 30			Time to login

## Table perfil

Table perfil keeps the information related to the profiles used in the application to manage the access control and restriction of permissions.

**Table 11: Table perfil**

Column	Type	Default	Links to	Comment
id	Number 11			Id profile
name	Varchar 50			Name of profile
type	Number 11			Type of profile

## Table project

Table project stores the information of each project of the application.

**Table 12: Table project**

Column	Type	Default	Links to	Comment
<b>id</b>	Number 11			Id project
<b>name</b>	Varchar 100			Name of project
<b>id_arena</b>	Number 11		arena.id	Id arena
<b>ubication</b>	Varchar 150			Location of project
<b>date_from</b>	Date			Date from project
<b>date_to</b>	Date			Date to project
<b>desc_project</b>	Varchar 300			Description of project
<b>num_pad</b>	Number 11			Number of pad for project
<b>images</b>	Varchar 100			Style image for project

## Table store

Table store keeps the information of each pad of Etherpad from each project of the application.

**Table 13: Table store**

Column	Type	Default	Links to	Comment
<b>key</b>	Varchar 100			Key document for collaborations
<b>value</b>	LongText			Document for collaborations

## Table user

Table user stores the information of each user of the application.

**Table 14: Table user**

Column	Type	Default	Links to	Comment
<b>id</b>	Number 11			Id of user
<b>name</b>	Varchar 100			Name user
<b>username</b>	Varchar 30			Username for user
<b>email</b>	Varchar 50			Email of user
<b>password</b>	Char 128			Password of user (HASH)
<b>id_perfil</b>	Number 11			Id profile
<b>salt</b>	Char 128			Salt of control security
<b>image</b>	Mediumblob			Image user in the web
<b>image_type</b>	Varchar 50			Image type
<b>location</b>	Varchar 70			Location user
<b>institution</b>	Varchar 70			Institution or Organization of user

## Relational Schema

Figure A.1 shows the schema of the relational database used to the application.

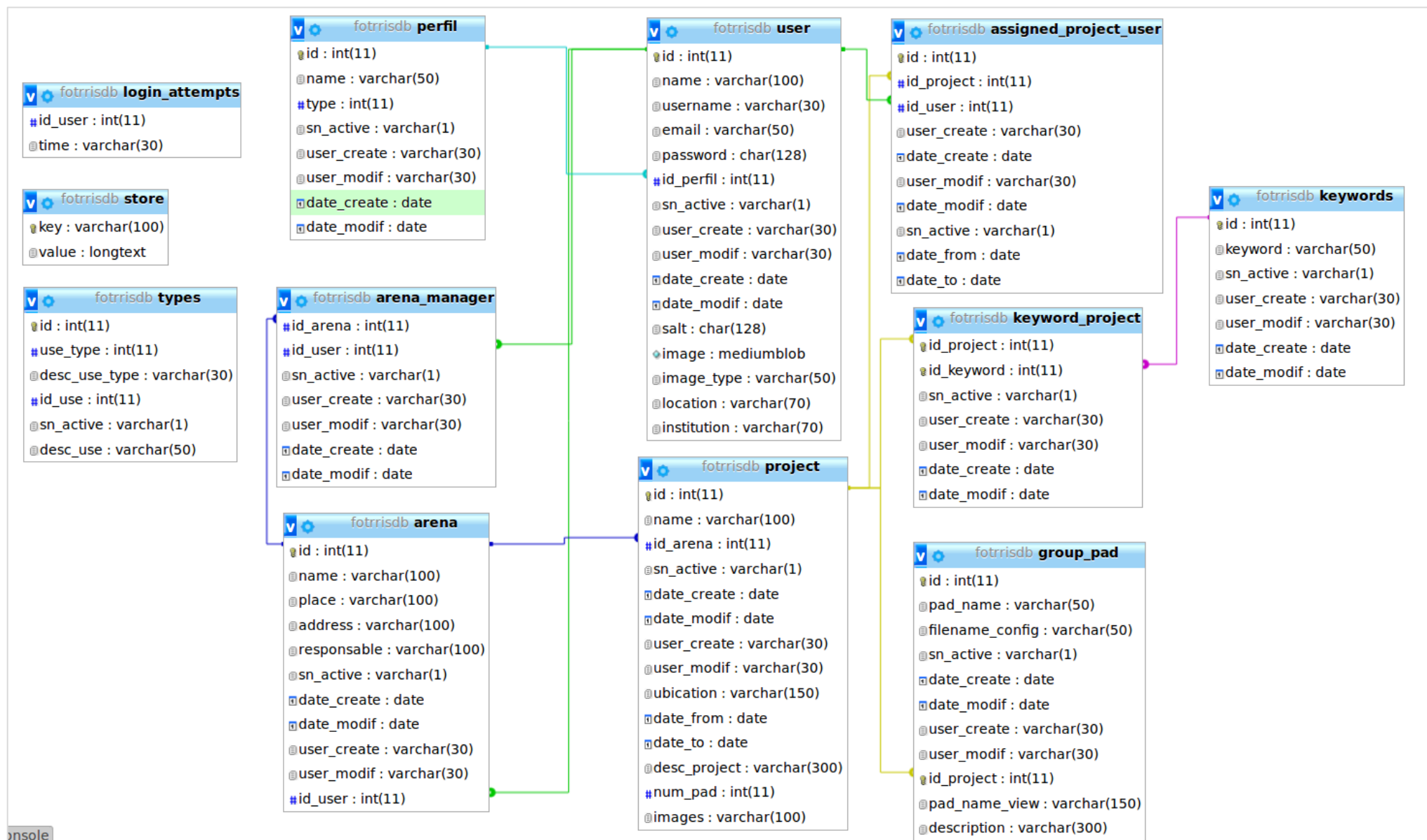


Figure A.1. Schema of the database

