Private Public Partnership Project (PPP)
Large-scale Integrated Project (IP)

D11.5.2: Community Building and Engagement Platform

Project acronym: FI-WARE
Project full title: Future Internet Core Platform
Contract No.: 285248
Strategic Objective: FI.ICT-2011.1.7 Technology foundation: Future Internet Core Platform
Project Document Number: ICT-2011-FI-285248-WP11-D11.5.2
Project Document Date: 2014-06-11
Deliverable Type and Security: PU
Author: Futura Networks
1.1 Executive Summary

This report is the accompanying document for the second prototype of Campuse.ro, FI-WARE’s “Community Building and Engagement Platform”. The report briefly describes the main requirements, functionalities and characteristics of this second prototype, and gives clear ideas for the future steps in the development process of Campuse.ro. Special attention has been paid to skip the features available for the previous version and thus exclusively focus on those developed for this one, and to indicate how the platform has progressed during these months.

We contemplate three successive deployments of the Campuse.ro platform, having started with the first prototype in September 2013 and progressing into May and September 2014, when the final version will be ready.
1.2 About This Document
This report is the accompanying document for the second prototype of Campuse.ro. The document briefly describes the main requirements, functionalities, architecture and characteristics of such prototype.

1.3 Intended Audience
The document targets all types of stakeholders interested in the FI’s possibilities as a business and as communities’ catalyser, with special focus on developers, web entrepreneurs and SMEs.

1.4 Keyword list

1.5 Changes History

<table>
<thead>
<tr>
<th>Release</th>
<th>Major description</th>
<th>changes</th>
<th>Date</th>
<th>Editor</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.5</td>
<td>First draft of deliverable, submission to ATOS and TID</td>
<td>2014-06-11</td>
<td>Javier de Vicente, Sergio López, Imedi Hassisen, Asier Arranz, Eduardo Lorea, Esther Paniagua (FNE), Leonardo Santiago, Leonardo Moreno (FNE Colombia)</td>
<td></td>
</tr>
<tr>
<td>v0.51</td>
<td>Revision after WPL’s comments</td>
<td>2014-06-13</td>
<td>Javier de Vicente (FNE)</td>
<td></td>
</tr>
</tbody>
</table>
# 1.7 Table of Contents

1.1 Executive Summary ................................................................. 2
1.2 About This Document .................................................................. 3
1.3 Intended Audience ..................................................................... 3
1.4 Keyword list ........................................................................... 3
1.5 Changes History ........................................................................ 3
1.7 Table of Contents ...................................................................... 4
1.8 Table of Figures ........................................................................ 6
1.9 Index of Tables ......................................................................... 8
2 Introduction ................................................................................. 9
3 Requirements and Functionalities for the May 2014 Version .......... 10
  3.1 Organization Profiles ............................................................... 10
    3.1.1 Requirement .................................................................... 10
    3.1.2 Quick Description of the Functionality ............................... 10
    3.1.3 Associated Actions .......................................................... 14
  3.2 Challenges ............................................................................... 15
    3.2.1 Requirements ................................................................... 15
    3.2.2 Quick Description of the Functionality ............................... 17
    3.2.3 Associated Actions .......................................................... 18
  3.3 Talent Search ........................................................................... 24
    3.3.1 Requirement .................................................................... 24
    3.3.2 Quick Description of the Functionality ............................... 24
    3.3.3 Associated Actions .......................................................... 26
4 Architecture .................................................................................. 31
  4.1 Introduction .............................................................................. 31
  4.2 Significant Design Packages .................................................... 32
    4.2.1 Presentation Tier ............................................................... 32
    4.2.2 Business Tier ................................................................... 33
    4.2.3 Integration Tier ............................................................... 34
    4.2.4 Data Tier ........................................................................ 35
  4.3 Deployment Layer ..................................................................... 35
    4.3.1 Description of the Servers ............................................... 35
    4.3.2 Staging and Consolidation of the System ......................... 39
    4.3.3 Security .......................................................................... 40
    4.3.4 Internationalization ......................................................... 41

D11.5.2: Community Building and Engagement Platform 4
4.3.5 Logging and Error Tracking

4.4 Significant Libraries and Frameworks
4.4.1 Libraries and Frameworks
4.4.2 Development Tools
4.4.3 Java Packages

5 Quick User Guide

5.1 Organization User
5.1.1 Creating an Organization User Account
5.1.2 Managing your Profile

5.2 Challenges
5.2.1 Creating a Challenge
5.2.2 Managing the Challenge
5.2.3 Participating in a Challenge

5.3 Talent
5.3.1 Searching for Talent
5.3.2 Job Posting
5.3.3 Managing the Job Offers

6 Future Steps in the development of Campuse.ro
1.8 Table of Figures

Figure 1. Campuse.ro Logo ........................................................................................................ 9
Figure 2. Registration form process at Campusero ................................................................. 11
Figure 3. The process of creating a Challenge inside Campusero ......................................... 19
Figure 4. All the different functions for managing the Challenge ....................................... 21
Figure 5. Functions that an individual user can perform regarding challenges .................. 22
Figure 6. Accepting, favouriting and managing the Challenge (individual user) ............... 23
Figure 7. The process of posting a job offer (organization profiles) ..................................... 25
Figure 8. Functions available for individual users in the Talent module ............................... 26
Figure 9. Job opportunity form, as shown to organization profiles ...................................... 28
Figure 10. Jobs listings, as shown to individual users ............................................................ 29
Figure 11. Details of one job opportunity ............................................................................. 30
Figure 12. Campusero's N-tier architecture ........................................................................... 31
Figure 13. Campusero's software architecture ..................................................................... 32
Figure 14. Presentation Tier ................................................................................................... 33
Figure 15. Business Tier ......................................................................................................... 34
Figure 16. Integration Tier ...................................................................................................... 34
Figure 17. Campusero's deployment model .......................................................................... 35
Figure 18. Databases involved in the authorization system .................................................. 41
Figure 19. User guide: Log in ................................................................................................. 49
Figure 20. User guide: Join now ............................................................................................ 49
Figure 21. User guide: Register ............................................................................................. 50
Figure 22. User guide: The dashboard .................................................................................. 51
Figure 23. User guide: Profile form ...................................................................................... 52
Figure 24. User guide: Create .............................................................................................. 53
Figure 25. User guide: New challenge .................................................................................. 54
Figure 26. User guide: Contest form .................................................................................... 55
Figure 27. User guide: Description ....................................................................................... 56
Figure 28. User guide: manage the challenge ....................................................................... 57
Figure 29. User guide: challenges at the dashboard ............................................................... 57
Figure 30. User guide: managing evaluations ....................................................................... 58
Figure 31. User guide: Open challenges .............................................................................. 59
Figure 32. User guide: The challenges button ....................................................................... 59
Figure 33. User guide: Submission form .............................................................................. 61
Figure 34. User guide: Talent search .................................................................................... 62
Figure 35. User guide: Filtering the results ........................................................................... 62
Figure 36. User guide: Creating a job offer ........................................................................................................ 63
Figure 37. User guide: Subscribing to the job offer ...................................................................................... 64
Figure 38. User guide: Job offers .................................................................................................................. 65
Figure 39. User guide: Managing the job offers .............................................................................................. 66
Figure 40. User guide: Job offers at the dashboard ....................................................................................... 67
1.9 Index of Tables

Table 1. Fields, functions and their differences between profiles ........................................ 14
Table 2. Roles and differences between profiles .................................................................... 15
Table 3. Processes and profiles inside the Challenges module ............................................. 17
Table 4. Pages and views inside the Challenges module ....................................................... 18
Table 5. Processes within the Talent module and differences between profiles .................... 25
Table 6. Development Server’s details .................................................................................. 36
Table 7. Development Server’s services ................................................................................ 36
Table 8. Test Server’s details ................................................................................................. 36
Table 9. Test Server’s services ............................................................................................... 37
Table 10. Application Servers’ details ................................................................................... 37
Table 11. Available services ................................................................................................ 37
Table 12. Details of the Load-balancing server ..................................................................... 38
Table 13. Services running on the Load-balancing server .................................................... 38
Table 14. Auth Server’s details ............................................................................................. 38
Table 15. Services running on the Authorization Server ....................................................... 39
Table 16. Details of the Database Server ................................................................................ 39
Table 17. Components relevant for each stage ................................................................... 40
Table 18. Relationship between user and resource ............................................................... 41
Table 19. Libraries and frameworks (Backend) ................................................................... 45
Table 20. Libraries and frameworks (Frontend) .................................................................. 46
Table 21. Development tools ............................................................................................... 46
Table 22. Subpackages structure (Backend) ......................................................................... 48
Table 23. Subpackages structure (Frontend) ....................................................................... 48
2 Introduction

Ever since the first Campus Party in the late nineties, strong support has been given for the development of a community that, at the time of writing, has exceed 325,000 “campuseros”: fanatics of Internet, digital culture and experts in many branches of ICT knowledge. These individuals come from 130 different countries, from Spain to Mexico, and Germany to Brazil and many of them are attendees of the Campus Party events.

The idea of Campuse.ro is in fact very recent. Campuse.ro, as a concept, was born in 2012 with the objective of delivering a digital platform for strengthening and fostering the network of knowledge that is already present in the communities. Making this knowledge continuously available – and not only during the Campus Party events – usable, and useful for other individuals or organizations is the goal of Campuse.ro.

By entering the FI-WARE consortium, a huge opportunity for extending this vision has been given to the European FI sphere. The opportunity of making it real and connecting it to the geek and the professional worlds. Our challenge now is to reach the developers, entrepreneurs, companies and researchers and engage them in the FI-WARE experience.

By delivering Campuse.ro, we expect our users to be able to get in touch with other individuals and organizations with technological interests alike while at the same time promoting contents, activities and talent-search for future tasks and developments on top of the FI-WARE architecture.

![Campus Logo](image)

Figure 1. Campuse.ro Logo

When all functionalities are in place, a great experimental environment for promotion, cooperation and business development will be ready, gathering thousands of strategic players (developers, creative minds, internet leaders, opinion leaders, bloggers, social media influencers, entrepreneurs, start-ups, big companies, public institutions, media, universities etc) that will contribute to create and support dynamic ecosystems around the FI-WARE results.

The current status of Campuse.ro can be checked at [http://campuse.ro/](http://campuse.ro/)

This report is the accompanying document for the second prototype of Campuse.ro. The report briefly describes the main requirements, functionalities, architecture and characteristics of this second prototype. The features already available for the previous version are not indicated; while we focus on those developed for this one and how the platform has progressed during these months. This includes the absence of the “Performance” chapter, which was present in the previous release in this series of reports, since there are no major differences in this regard between both versions and our intention is to keep the previous, present and future releases as an overarching but complementary set of reports about the entire platform. The new modules designed for this version are “Organization User”, “Challenges” and “Talent”.

D11.5.2: Community Building and Engagement Platform 9
3 Requirements and Functionalities for the May 2014 Version

In this chapter the requirements and functionalities developed for this version will be introduced to the reader. As it will be noted, the modules relative to this release are directly related to the previously established roadmap, and include two big groups of updates to the previous version: (1) non-individual profiles - for organizations, including companies, universities, governments – and (2) modules that allow these entities to relate with the already significant community of individuals, more specifically modules about Talent search and the execution of Challenges.

The ulterior objective of these functionalities is to set the basis for the activation of the platform. In other words, these are the first modules that allow individuals and organizations to interact in a productive way, stimulating innovation, creativity and jobs creation. In the next, third version, other functionalities in the same vein will be added, sophisticating the opportunities of interaction.

3.1 Organization Profiles

In its previous release, Campuse.ro published a User Profile functionality, that allowed users to register at the platform, manage their personal information, perform other basic processes such as creating or attending events, exchanging messages with other users and choosing the language in which contents should be presented (Spanish, Portuguese or English). However, there was no differentiation between types of users, and notably the required information in the registration form was only adapted for individuals, not for organizations.

Organizations are fundamental for the platform, since they are typically on the “demand side” of talent, as well as being the key promoters of events and challenges. It was necessary to define and to adopt a new profile model for these roles. With the new modules, organizations are able to perform some activities that are not allowed for individual users. Looking also into the future of the platform, both profiles (individual and organization) and the records for their activities within the platform are the basis for features to be added to the platform in the upcoming months, especially content recommendation and user reputation. Criteria associated with these features for each type of user should be different.

Considering the online social habits of the users of the platform, a modality of the organization profile dedicated to user communities was created – in other words, informal groups of users with similar interests, that may gather in order to perform activities typically associated with organizations even though they do not have any formal constituency.

3.1.1 Requirement

The requirements to be met were for the organizations (1) to have a public/private space, analogous but different from the previously released user profile dedicated to individuals, in which profile information about the organizations can be administered and used by the business modules; in this version of the platform, by the challenges module and the talent search module, in a specific role if compared to individual users; (2) to extend the ability of using the messaging module and the activities module, as well as the friendship functionality, that had been released in the previous version of the platform, for organization users; and (3) to allow individual users to declare themselves members of the organizations.

3.1.2 Quick Description of the Functionality

The platform allows the organization user to

1. Create;
The access to these actions starts by creating a profile. The link to perform this action is at the Campusero homepage. After registering, the organization user has access to a dashboard in which is possible to view, edit and delete the profile. The following diagram describes the flux of the organization profile:

![Diagram of the organization profile flux](image)

**Figure 2. Registration form process at Campusero**

An organization profile is constituted by the following data groups:

- Registration (including fields: username, email address and password), used to access the platform and to perform all activities in it;
- Identification (including fields: name of the organization, brief description, type of organization – public institution, government, company, associations/non-governmental/civil society, educational institution, user community, venture capitalist –, foundation date, website, detailed description, profile image);
- Contact (including fields: location/address, time zone, social media profiles, phone number, email address);
- Interests (including fields: area of activity, thematic tags), used to allow any user to find others with similar interests, and additionally to recommend content and to match\(^1\) individuals with organizations and vice versa, with potentially congruent interests talent-wise;
- Members (a notification will be sent to the organization, which will then be able to approve who belongs to it);

The table below compares our two types of user profiles: individuals and organizations.

<table>
<thead>
<tr>
<th>Data Group</th>
<th>Field or function</th>
<th>Individual User Profile</th>
<th>Organization User Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registration</td>
<td>Username</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

\(^1\) The content recommendation and the user matching features are expected to be added to the platform in the third release

D11.5.2: Community Building and Engagement Platform
<table>
<thead>
<tr>
<th>Identification</th>
<th>Password</th>
<th>Mandatory</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Identification</td>
<td>Mandatory</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Type of organization</td>
<td>Identification</td>
<td>Not Applicable</td>
<td>Mandatory</td>
</tr>
<tr>
<td>[Public Institution</td>
<td>Government</td>
<td>Company</td>
<td>Education</td>
</tr>
<tr>
<td>Date of Birth/Foundation</td>
<td>Identification</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Brief description/Bio</td>
<td>Identification</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Gender</td>
<td>Identification</td>
<td>Optional</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>[Male</td>
<td>Female</td>
<td>Other]</td>
<td>Identification</td>
</tr>
<tr>
<td>Description</td>
<td>Identification</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Profile Image/Picture</td>
<td>Identification</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Website</td>
<td>Identification</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Contact Location/Address</td>
<td>Identification</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Time Zone</td>
<td>Identification</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Social media profiles</td>
<td>Identification</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Phone number</td>
<td>Identification</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>Email address</td>
<td>Identification</td>
<td>(Already filled in)</td>
<td>(Already filled in)</td>
</tr>
<tr>
<td>Interests Area/s of activity</td>
<td>Identification</td>
<td>Optional</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Thematic tags</td>
<td>Optional [Treated as “interests” and “skills”]</td>
<td>Optional [treated as “interests” and “products/services”]</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Reputation</td>
<td>To be added in future release</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Experience</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position #1</td>
<td>Optional [Occupation</td>
<td>Position</td>
<td>Main Activities and Merits</td>
</tr>
<tr>
<td>Position #2</td>
<td>Optional [Occupation</td>
<td>Position</td>
<td>Main Activities and Merits</td>
</tr>
<tr>
<td>Position #3</td>
<td>Optional [Occupation</td>
<td>Position</td>
<td>Main Activities and Merits</td>
</tr>
<tr>
<td>Education</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Education #1</td>
<td>Optional [Title</td>
<td>School</td>
<td>Duration</td>
</tr>
<tr>
<td>Education #2</td>
<td>Optional [Title</td>
<td>School</td>
<td>Duration</td>
</tr>
<tr>
<td>Education #3</td>
<td>Optional [Title</td>
<td>School</td>
<td>Duration</td>
</tr>
</tbody>
</table>
Members | Members | Not Applicable | [The organization does not fill its list of members; individuals that declare themselves members of the organization are listed in the profile visualization]

| Member of | Optional | Not Applicable |

Table 1. Fields, functions and their differences between profiles

3.1.3 Associated Actions

After logging in to the platform using the correspondent access data, the users with organization profiles may perform basic activities common to any user in the platform:

1. to access the homepage and any public contents;
2. to see other users profiles;
3. to search contents;
4. to manage profile information;
5. to participate of activities;
6. to follow users.

The messaging system and the activities module, as well as the friendship functionality, functionalities that were already released in the previous version of the platform, are managed from the organization user dashboard. Dashboard items associated with the mentioned functionalities are:

- Recent Activity: Notifications from the system about recent activity carried out by the organization (e.g. Conversations or activities the organization is managing or takes part of, other users that the organization interacts with, events the organization is promoting)
- Contents: A summary of the activities (meetings, workshops, talks) which are linked to the organization
- Conversations: Messages that the organization is delivering or receiving
- Contacts: A list of followers and followed individuals.

Furthermore, as previously described, organization profiles play a specific role in the business modules within the platform. Typically, an organization is the promoter of innovation challenges, and asks individuals to submit proposals; is the hirer of talents, and expects individuals to provide useful information about their professional profiles as to be considered candidates to job opportunities; and is the one requiring services, asking for specific duties to be performed by individuals as providers. The following table compares roles performed by individual profiles and organization profiles in each module:

<table>
<thead>
<tr>
<th>Profile type</th>
<th>Business Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual User</td>
<td>Participant</td>
</tr>
<tr>
<td></td>
<td>Candidate</td>
</tr>
</tbody>
</table>

D11.5.2: Community Building and Engagement Platform
Table 2. Roles and differences between profiles

The specific activities performed by organizations and individual users in each business module will be detailed in the descriptions of the modules.

3.2 Challenges

Campusero’s challenges are based in the cooperation among professionals and any other. It extends innovation from the companies’ R&D departments to the Internet sphere, where almost anyone from all over the world can have a new - or old - idea that incorporates value to the company’s business, and can play a key role in solving problems of organizations by taking part in innovative projects.

This module’s objective is to help organizations reach the person that has the solution for their specific needs. This in return also feeds back progress of the own ecosystem, and encourages others to take part in the other available challenges.

Hard work has been done with this aim in mind, creating specific tools that allow the organization users in Campusero to launch challenges within the community and to find projects from excellent developers, designers, analysts, entrepreneurs and combinations of the above. The module enables direct interaction between individual users and organizations too.

The full process of launching and resolving a Challenge is briefly described in these lines. The reader will find more details in the following chapters, in any case.

Step 1) The process starts with the organization user - who is already registered in Campusero - filling in a preliminary form with basic information about the contest.

Step 2) Then, an administrator of Campusero contacts the organization user to make an agreement. Once the agreement has been reached, the organization user completes a more comprehensive form with all the details of the challenge, including the legal terms, criteria and others.

Step 3) Afterwards, the challenge is published and remains visible to everyone that visits Campusero.

Step 4) In case a user wants to participate, he/she must be registered. Then he/she is entitled to click on the challenge, apply and finally upload his/her project on time.

Step 5) Resolution: The organization user fixes the date for announcing the winner/s and selects them once the decision is taken. Then, the winner/s will be notified through the standard notification system plus receiving a personal email.

Step 6) Prize: The organization will contact the winner/s directly in order to deliver the prize/s. If that does not happen, the winner/s will have the right to ask for arbitration of an administrator of Campusero (administrator user).

3.2.1 Requirements

To reach its objectives, the platform should

(1a) Allow users to propose Open Innovation Challenges, enable proper dissemination of the challenges, allow users to send proposals, allow creators of the challenge to select the winner or winners [and finalists, given the case] and to officially close the challenge;

(1b) Allow the inclusion of a specific link for “Challenges” in the website’s main menu, in which all the current challenges are to be shown; and also feature all the on-going challenges in the
website’s homepage, ordered by closing date. This must be visible in three different languages: Spanish, English and Portuguese.

(2) Enable seamless interaction among three sorts of users: (a) organization users, as potential proposers of the Open Innovation Challenges; (b) individual users, as potential providers of solutions (proposals) to such challenges; and (c) administrators, representatives of the Campusero platform, as catalysts of the process. All these users should be previously registered in the Campusero platform and have permissions to “Create”;

Let us now conclude this sub-chapter about requirements by differentiating among all the types of users and which activities they should be able to carry out:

(3) Organization profiles should be able

1. to create Open Innovation Challenges, by filling in a form containing all the necessary information;
2. to manage the Open Innovation Challenges they have created, which means accessing a dashboard that allows them
   b1. to update relevant information about the challenge;
   b2. to publish news/announcements about the challenge;
   b3 to send messages to individual users both in bulk (i.e., for all the participants that submitted proposals or chose the challenge as favourite) and individually;
   b4. to access statistics for the challenges: views, favourites; number of users registered (participants) and proposals submitted;
   b5. to see a list of proposals (ideas) submitted by individual users and to download the materials they sent; and
   b6. to set the challenge as open, closed, or awarded.
3. to include information about eligibility in the form (not eligible candidates would be excluded from the distribution list of the challenge)
4. to publish it in up to three different languages (English, Spanish or Portuguese); and
5. to share the Challenge in social media by a permalink and an optimized Facebook.

(4) Individual user profiles should be able

a. to access information about Open Innovation Challenges;
b. to apply to one or several Open Innovation Challenges, by setting themselves as applicants in challenges;
c. to submit a proposal by filling in a template (that would be provided by Futura Networks S.L. or by the organization responsible for the Challenge) and optionally sending additional attached files (PDF; PPT, DOC, XLS, or image formats);
d. to mark challenges as favourites; these favourite challenges should appear in the user’s dashboard;
g. to communicate with the organization through private messages
h. to receive an official certificate in case of being chosen as finalist and/or winner

(5) Administrators should be able

a. to receive notifications when an Open Innovation Challenge has been created by an organization;
b. to contact the organization and make necessary arrangements before the challenge is published;
c. to authorize the publication of the Challenge, once it is approved;
3.2.2 Quick Description of the Functionality

The requirements listed above were translated into processes that will be described in the subsequent paragraphs. The following table shows which kinds of user profiles are related to each process:

<table>
<thead>
<tr>
<th>Process</th>
<th>Organization User</th>
<th>Individual User</th>
<th>Administrators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge creation</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Challenge managing</td>
<td>Yes</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Challenge viewing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Applying/Proposal submission</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Communication</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3. Processes and profiles inside the Challenges module

Here, the following table lists the pages/views related to the Challenge module:

<table>
<thead>
<tr>
<th>Page/view</th>
<th>Contents</th>
<th>Possible Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenges page – Organization user view</td>
<td>List/Preview of current active challenges; &quot;Create New Challenge&quot; link - Global amount of the money delivered to users – winners and finalists – until the date (this amount will be automatically updated)</td>
<td>Create Challenge; see details of challenges</td>
</tr>
<tr>
<td>Challenges page – Individual user view</td>
<td>List/Preview of current active challenges organized by country and date of closing; Global amount of the money delivered to users – winners and finalists – until the date (this amount will be automatically updated)</td>
<td>See details of challenges</td>
</tr>
</tbody>
</table>
Table 4. Pages and views inside the Challenges module

| Challenges dashboard – Organization user view | List/Preview of user’s created challenges, including status | Manage challenges. Create a new challenge |
| Challenges dashboard – user view | List/Preview of user’s favourite/applying challenges | Un-favourite; Submit proposals |
| Challenges dashboard – administrator view | List/Preview of challenges created by organization users | Approve challenges; Manage challenges Create a new challenge |
| Challenge page (details) – individual user view | All the challenge information and files; link to set as favourite; link to submit proposals | Set as favourite; Submit proposal |
| Challenge page (details) – individual user -participant-view | All the challenge information and files; link to set as favourite; link to update the proposal submitted | Set as favourite; Update the proposal submitted |

3.2.3 Associated Actions
As indicated above, the following sub-chapters sum up each process with regards to the Campusero challenges, from creation to communication, thus describing all the functionalities available for the users.

3.2.3.1 Creating a Challenge
This process is only available for organization user profiles, and is catalysed by administrators. After accessing the Open Innovation Challenges page in the Campusero platform, organization users can find a link in which they are able to create a challenge. Also, they can have access to it directly form anywhere in Campuse.ro through the “Create” button (“Crear”). If an individual user tries to create a Challenge, an automatic message asks him/her to register as an organization user.

To complete the process, the organization needs to fill in two forms: one (1) very basic preliminary form, and then (2) another more comprehensive form, that will be the final contest form.

The user is firstly presented with the preliminary form. It contains the fields below:

- Aim
- Brief description
- Target audience
- Budget
- Prize/s
Future Internet Core Platform

- Name of representative in charge of the challenge
- Contact info (Telephone and email)

After the form is filled in and submitted, an Administrator at Campusero receives the request and contacts the organization to provide information about different packages and pricing and to make the necessary arrangements.

Figure 3. The process of creating a Challenge inside Campusero
During the negotiation between the organization and the administrator, the Challenge request will appear in the organization’s challenges dashboard with the status “Approval pending”. Nevertheless, the organization may continue the process of challenge creation. The only action that cannot be performed before approval is the publication of the Challenge.

After submitting the “I need...” form, the organization user is led to the “Contest form”. It contains all the necessary information to launch the contest, and more specifically the following fields:

- Banner: Insert their own image at the heading of the page
- Title
- Target audience: location, skills
- Category
- Type
- Description of the challenge, containing the following info:
  - What do you expect as a final result?
    - Extension?
    - In what format?
    - What technical features must have?
    - Which other additional characteristics will be considered?
- Evaluation criteria (e.g. impact, potential, innovation, creativity, utility, feasibility, presentation, design...)
- What will be the evaluation process? Methodology and schedule (This info could be selected as private)
- Who will be the jury? (This info could be selected as private)
- Legal terms: Futura will provide templates in three languages (Spanish, English and Portuguese)
- Prizes: organization will have the chance to select the currency
- Dates:
  - Opening date
  - Closing date
  - Announcing date
  - Awards Ceremony? (Y/N + Place)
- Video: the organization user may explain the challenge also through a video (by inserting an iframe/embedded code or a link to an external video-platform)
- Mandatory presentation: a template for the participants will be provided by default by Futura but organizers of the challenges may replace it by their own template.
- Contact support: the organization user will have to provide an email address for some person who can provide technical support to the participants

After filling in and submitting the “Contest form”, the organization user’s browser will be sent to a “Thank You” page informing that the administrator will soon get in touch to make a deal and then publish the challenge. The challenge will be available for publishing only after necessary arrangements between the organization and the administrator have been concluded.

The administrator, meanwhile, contacts the organization user and negotiates the necessary terms. After the negotiation is positively concluded, he/she sets the challenge as “approved” in the administrator dashboard. The new status is visible for both the administrator and the organization (previously it was “Pending Approval”). This way the organization can advance in the management process. Nevertheless, no challenge could be published if one or more fields marked with an asterisk “*” is/are incomplete. An automatic notification message asking the organization to fill in the missing information would immediately appear when trying to publish the challenge.

### 3.2.3.2 Managing a Challenge

After the challenge is created and approved, the organization may perform any other action related to the challenge’s managing process without the need of the administrator’s intervention.
Nevertheless, the administrator is able to perform any of the managing actions on behalf of the organization upon request.

All the following processes are performed through the challenge dashboard (organization/administrator view). In the dashboard, the challenge(s) created by the organization are listed, with their current statuses. During the life of the contest, the main aspects of the challenge that need the organization’s intervention are:

A) “Status”: should be set
   - Draft
   - Published and open for proposals;
   - Closed for proposals; or
   - Awarded: In this case, the organization will have to select the winner/s and the finalist/s, with the chance to upload one or more videos of the awards ceremony and the awarded projects

B) “Projects”: a space where the organization user can access and directly download the proposals submitted for the challenge

C) “Share”: to share the Challenge in social media by a permalink and an optimized Facebook

D) “Notifications”: This field allows the user to manage all the alerts that he/she will receive in his/her email account, e.g. when a campusero has accepted the challenge, when someone has submitted a proposal, internal messages received from candidates, reminders for deadlines and others (closure, evaluation period, awarding etc.). By default, all these alerts will be shown as notifications when the organizer signs in to Campusero

E) “Recent Activity”: Notifications from the system about recent activity carried out by the candidates (e.g. “someone has submitted a new proposal”) ²

![Figure 4. All the different functions for managing the Challenge](image)

3.2.3.3 **Viewing a Challenge**

Previews of active challenges are shown both in the homepage of the Campusero platform and in the Challenges page. In the later case, challenges will be organized by country and date of closure. By clicking in the previews, both individual users and organization users are able to see the challenges details in the challenge page (details).

² D and E to be included in future releases
Additionally, individual users are able to mark challenges as favourites and to apply in a challenge (that means, to say that the user has the intention of submitting a proposal and to receive notifications about). Both favourite challenges and challenges in which the user is applicant appear in the individual user challenges dashboard, as a reminder for him/her to submit a proposal within the deadline.

Figure 5. Functions that an individual user can perform regarding challenges

3.2.3.4 Applying, Proposal Submission and My Challenges

The proposal submission process is carried out by individual users or by organizations. This sets users as applicants in a challenge, and does so through a changing button at the end of each challenge’s page that firstly says “Accept the challenge”, secondly – once accepted – says “Upload your proposal” and finally – when submitted – says “Update your proposal. After setting himself/herself as applicant in a challenge, the challenge appears in the user’s challenges dashboard, hereafter referred as “My challenges”.

- The “My challenges” dashboard shows all the challenges in which the Campusero is participating and those marked as favourites, including the following fields:
  - The challenge’s title
  - A short description of the challenge (first paragraph of the “Description of the challenge”)
  - A button with the status:
    - Open
    - Closed: the submission phase has finished and the candidate did not submit any proposal
    - Awaiting: the submission phase has finished and the candidate submitted his/her proposal but the winners have not been announced yet
    - Failed: the winner/s have been announced and the contestant has not been selected as one of those.
    - Awarded: when the user has been one of the winners/finalists. In this case, there will be a button called “Certificate” that will be automatically be sent to an administrator of Campuse.ro (administrator user) to ask for a document to certify
the prize

- A link to the challenge’s page where the candidate can see all the information and submit his/her proposal

- “Templates”: a template provided by Futura Networks S.L. and/or the organization, which must be completed by the user

- “Notifications”: this field allows the user to manage the alerts that will be received in his/her email account: challenge’s deadlines, internal messages received from the organizations, information about the contests, and others. By default, all this alerts will be shown as notifications when the organization user makes “Sign in” on campuse.ro

- “Recent Activity”: notifications from the system about recent activity carried out by an organization (e.g. “some company has posted a new challenge”) ³

---

³ Same as previous comment

---

After applying to one challenge, the user should submit a proposal before its deadline to complete the participation process. For doing so, he/she should access the specific challenge and click on the button “Upload your proposal”. Then, the user will have to fill in the template provided by Futura Networks S.L. - or by the organization - and attach other files (PDF; PPT, DOC, XLS, or image formats). If wishing to update the proposal, the user could do so by clicking on the button “Update your proposal”, but only during the submission period.
3.3 Talent Search

As the platform now features both individual user profiles and organization profiles, with different roles, it is possible to develop modules in which both can interact with specific objectives. It is expected that the interaction results in the creation of jobs, provision of services and innovative solutions.

The first module that enhances this productive interaction is the Talent module. In its first release, defined hereafter, its main requirement is to allow organizations to search and contact individual users that meet their human resources needs, through relevant parameters. Additionally, organizations can also create job offers for which individual users can offer themselves as candidates.

3.3.1 Requirement

The requirement to be met in this first release was (1) for organizations to be able to search within the Campusero user base with relevant parameters for individuals that meet their human resources needs; (2) to show the results of the search in a friendly and useful way; (3) to allow organizations to contact selected individuals. Additionally, (4) organizations should be able to publish job offers by filling a form; (5) the job offer should be visible for individual users who, finally, (6) should be able to apply as candidates.

3.3.2 Quick Description of the Functionality

In the Campuse.ro platform main menu, a link leads the user to the Talent Search module. For organizations, in their “hirer” roles, there are two possible actions:

- to search for talents, or, in other words, individual users whose professional characteristics meet the organization’s human resources needs;
- to publish a job opportunity, which will be visible for the community, and will allow individual users to declare themselves candidates.

The following table describes all the processes that take place in the Talent Search module, and relates it to each user:

<table>
<thead>
<tr>
<th>Process</th>
<th>Organization User</th>
<th>Individual User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Talent Search</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Job Creation</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Job managing</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Job viewing</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Application</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Communication</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 5. Processes within the Talent module and differences between profiles

The following diagram shows the flux for organizations in the module:

![Diagram showing the process of posting a job offer for organizations](image)

**Figure 7. The process of posting a job offer (organization profiles)**

Similarly, the following diagram shows the flux for individual users in the module:
3.3.3 Associated Actions

3.3.3.1 Process of Searching for Talent

To search for talents, a form is displayed for the organization user to fill it in with relevant criteria. The available parameters are:

- Skills/Interests, determined by tags in the user’s profile
- Geography (Country/Province/City)
- Experience, measured in years
- Age

After filling in the form with the desired parameters and hitting the “search” button, the system returns a list of individual users that match the criteria. The organization can then

- Order the results according to the different parameters
- Click on a profile to see its details
- Send a message to selected users
- Recalibrate search parameters to further filter results

The talent search is not visible for individual users, in other words, they don’t know if their profiles appeared in search results. New functionalities will be added to allow users to evaluate their
performance in talent searches in the future.

3.3.3.2 *Job Opportunities*

In order to publish a job opportunity, the organization clicks on the correspondent button in the main menu, and is leaded to a form that needs to be filled with relevant data about the position. The fields that should be filled are:

- Name of the Position
- Description
- Knowledge/Professional Area
- Required Education/Experience/Skills
- Salary and Benefits
- Start date / End date
- Geography (Country/Province/City)

3.3.3.3 *Viewing and Applying for Offers*

Starting from the set start date, the offer is shown to the public (1) among other contents in the homepage of the platform, and (2) upon request in the list of available job opportunities. If an individual user gets interested in a job offer, he/she is able to

- click it to see the details, and
- apply to it, which results in a message sent to the organization with the information about the candidacy.

3.3.3.4 *Managing Offers*

After having been created, the job offer will be listed also in an administration panel for the organization, which is able to edit and delete the offer before the end date, if needed. After the end date, the offer is automatically unpublished.

These are some screenshots of the platform:
Figure 9. Job opportunity form, as shown to organization profiles
Figure 10. Jobs listings, as shown to individual users
Figure 11. Details of one job opportunity
4 Architecture

4.1 Introduction

Campusero is a Java web system using a standard N-tier architecture, with a presentation tier, business tier, integration tier and data tier. The presentation tier contains all the visible JSP and HTML, and its mission is to handle all inputs from the inner tiers and to deliver outputs to the user. The business tier handles the business logic and provides the abstraction layer to access the database. The integration tier links internal and external services from third parties. The data tier consists of the database and provides the persistence required for Campusero.

In the following diagram we can schematically see the architecture of the nested layers of Campusero. Note the differences from the previous release. This chapter focuses on the internal architecture of the system which, according to the diagram, corresponds to the middleware layer (center) and integration layer (left).

Recent work in the architecture of Campusero, as we can read in previous chapters, has been focused on the extent of user profiles, module of challenges, module of talent search. The addition of new types of users and the extension of its functions are implemented in the same module of the original business logic of Campusero (User bean). A new module is added for each new functionality regarding “Challenges” (Challenge Bean) and “Talent search” (Talent Bean). The rest of the architecture remains unchanged from the previous version of Campusero. The core of the architecture is maintained in the current conceptual version up to the completeness of the requirements for future phases.

The following diagram shows a quick overview of all the basic subsystems of Campusero and gives a basic overview of the complete system.
4.2 Significant Design Packages

4.2.1 Presentation Tier

The presentation tier implements a standard Model-View-Controller design pattern developed using Spring MVC framework. Spring MVC is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mappings, view resolution, locale and theme resolution as well as support for upload files. The default handler is a very simple Controller interface, just offering a ModelAndView handleRequest method. The controllers invoke the business tier through the use of business delegates, which allow the decoupling between the presentation tier and the business tier, acting as a client-side business abstraction. These business delegates interact using JNOs (Java Native Objects) as transfer objects.

To facilitate horizontal scalability and reduce the complexity, the business delegates use service locators in order to connect to the business tier. The business tier can be locally or remotely deployed.

As front-end, JSP components are used as a primary interface.
4.2.2 Business Tier

The business tier was developed using session stateless Enterprise Java Beans 3. The business logic is exposed through Session Facades, which then are used by the aforementioned presentation tier through the business delegates.

The services exposed by the session facades are coarse-grain and invoke other services exposed by the Application Services that are fine-grain. The session facade services can invoke several services of the Application Services.

The Application Services allow data persistency and are able to convert between Model objects and Entity objects with JPA annotations. This conversion between Model and Entity ensures that data persistency can be performed without affecting the higher-level layers.
4.2.3 Integration Tier

The Integration Tier allows Campusero to use internal services as well as external ones from third parties. Each service is composed of an interface and its specific implementation; these service components are Java interfaces and Java classes that can actually be used by any Java client.
4.2.4 Data Tier

The data layer consists of a PostgreSQL database and a stored procedure contained within. The data layer provides persistence for the system and all communications through SQL queries, stored procedures and views.

4.3 Deployment Layer

The deployment of Campusero is done using the AWS infrastructure, which is able to offer a professional solution for concurrent connections and significantly improves system performance and response times.

In this section the following items are presented: the structure of the platform architecture, the necessary hardware and the distribution of the components on the individual systems.

The architecture of the platform is shown below:

![Campusero's deployment model](image)

**Figure 17. Campusero’s deployment model**

4.3.1 Description of the Servers

4.3.1.1 Development Server

The development environment of Campusero consists of a Virtual Machine with all the necessary components to code and deploy the platform. This VM is called the development server and it is also used to execute unit tests, security tests and other sorts of tests.

<table>
<thead>
<tr>
<th>Location</th>
<th>Developer’s machine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 6. Development Server’s details

The basic services running on this server are listed below.

<table>
<thead>
<tr>
<th>Service</th>
<th>Package/version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application server</td>
<td>Oracle Glassfish 3.1.2</td>
</tr>
<tr>
<td>Search service</td>
<td>ElasticSearch 0.90.2</td>
</tr>
<tr>
<td>Auth Service</td>
<td>Play 2.1.1</td>
</tr>
<tr>
<td>Database service</td>
<td>Postgres 9.2.4</td>
</tr>
</tbody>
</table>

### Table 7. Development Server’s services

#### 4.3.1.2 Test Server

The test server is a clone of the global infrastructure of Campusero concentrated in a single server. This server is used for testing the quality control of the development and the quality of the internal logic of the system.

<table>
<thead>
<tr>
<th>Location</th>
<th>Amazon WS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Amazon Linux AMI 2013.3</td>
</tr>
<tr>
<td>Java version</td>
<td>OpenJDK 1.6.0_24</td>
</tr>
</tbody>
</table>

### Table 8. Test Server’s details

The basic services running on this server are listed below.
### Application Servers

The application servers contain the main code of Campusero and are responsible for the execution of all the processes and services. Campusero uses three replicates of the same server to handle user traffic. The third server is a backup server that runs only during the Campus Party event, when traffic on the platform increases significantly.

<table>
<thead>
<tr>
<th>Location</th>
<th>Amazon WS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Amazon Linux AMI 2013.3</td>
</tr>
<tr>
<td>Java version</td>
<td>Java(TM) SE 1.6.0_45</td>
</tr>
</tbody>
</table>

The basic services running on this server are listed below.

<table>
<thead>
<tr>
<th>Service</th>
<th>Package/version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application server</td>
<td>Oracle Glassfish 3.1.2</td>
</tr>
</tbody>
</table>

### Load-balancing Server

The load balancer server is in charge of distributing the user traffic between the application servers.
servers.

<table>
<thead>
<tr>
<th>Location</th>
<th>Amazon WS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Amazon Linux AMI 2013.3</td>
</tr>
</tbody>
</table>

Table 12. Details of the Load-balancing server

The basic services running on this server are listed below.

<table>
<thead>
<tr>
<th>Service</th>
<th>Package/version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load balancer server</td>
<td>HA-Proxy 1.4.22</td>
</tr>
</tbody>
</table>

Table 13. Services running on the Load-balancing server

4.3.1.5 **Search & Authorization Server**

This server includes two different processes: user authorization and resource searching. On one hand, the server stores information related to the access to the system. On the other, the server is responsible for managing incoming search requests to avoid overloading on the PostgreSQL database.

<table>
<thead>
<tr>
<th>Location</th>
<th>Amazon WS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Amazon Linux AMI 2013.3</td>
</tr>
<tr>
<td>Java version</td>
<td>Java(TM) SE 1.6.0_45</td>
</tr>
</tbody>
</table>

Table 14. Auth Server's details

The basic services running on this server are listed below.

<table>
<thead>
<tr>
<th>Service</th>
<th>Package/version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search server</td>
<td>ElasticSearch 0.90.2</td>
</tr>
<tr>
<td>Web server (for OpenAM service)</td>
<td>Apache Tomcat 6.0.35</td>
</tr>
</tbody>
</table>
4.3.1.6 **Database Server**
The database server is responsible for managing the database of Campusero. Here, the users data, the resources and all the information presented in the platform are stored.

<table>
<thead>
<tr>
<th>Location</th>
<th>Amazon WS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Amazon Linux AMI 2013.3</td>
</tr>
<tr>
<td>Database</td>
<td>Postgre 9.2.4</td>
</tr>
</tbody>
</table>

4.3.2 **Staging and Consolidation of the System**
The staging system includes the following stages:

- **Development Environment.** Set up like a virtual machine, the development environment is the same for all the developers and it is installed in the machine of all the members of the development team. The updates of the development environment are controlled via GIT. Unit tests are executed and tested in this stage.

- **Test environment.** This stage is used for testing new functionalities in debug mode. It is used both to test the quality and performance of development and to check whether the design fulfils the requirements for new functionalities.

- **Production.** Contains the final version of Campusero. The version deployed in this stage has been previously validated in the Test environment.

For the development and test environments, reduced copies of the production environment have been set up, without the complexity of the load balancer.

<table>
<thead>
<tr>
<th>Component</th>
<th>Development</th>
<th>Test</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development server</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Test server</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Application Server 1</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Application Server 2</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Table 17. Components relevant for each stage

<table>
<thead>
<tr>
<th>Application Server 3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Load balancer Server</td>
<td>X</td>
</tr>
<tr>
<td>Search and Auth Server</td>
<td>X</td>
</tr>
<tr>
<td>Database Server</td>
<td>X</td>
</tr>
</tbody>
</table>

4.3.3 Security

The OWASP ESAPI is a web application security-control library that manages security issues in the whole platform, providing authentication and authorization capabilities. The former relates to resolving whether or not a caller is who they claim to be. Authorization, on the other hand, relates to determining whether or not an authenticated caller is permitted to perform a given operation. Beyond this, OWASP ESAPI defines a security API that covers all the security controls that a typical enterprise web application or web service project might need. There are about 120 methods across all the different security controls, organized into a simple and intuitive set of interfaces.

4.3.3.1 Authentication

The user must enter the username and password to access the platform. In order to improve the quality of the authentication system, the development team has integrated OpenAM too. OpenAM provides open source Authentication, Authorization, Entitlement and Federation software. The goal of this integration is to simplify the implementation of transparent single sign-on (SSO) as a security component in a network infrastructure.

4.3.3.2 Authorization

Campusero handles authorization using user roles and access control at the data level, accepting or denying access to different areas and contents. In addition, the client displays only the appropriate data and menu items.

As we can see in the next figure, the authorization system depends on four variables:

- **User role**: admin, editor, auditor or campusero.
- **Type of requested permission**: participate (P), visualize (V), modify (M), change state (C)
- **Type of relationship between user and resource**: follow, like, owner, admin, children, has, sponsor.
- **Publication status**: draft (D), published (P), suspended (S), deleted (E).
The following table shows how the permissions for each profile, according to their role in the system, have an effect on the resources.

| Roles      | D | P | S | E | D | P | S | E | D | P | S | E | D | P | S | E | D | P | S | E | D | P | S | E | D | P | S | E | D | P | S | E |
| None       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| CampusUser |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Auditor    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Editor     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Admin      | O | A | F | N | H | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C | C |
| Relations  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Table 18. Relationship between user and resource

4.3.4 Internationalization

The internalization and location system is based on properties files that contain the strings in the different languages, identified by a unique key. This key is called from the views and replaced by the associated translation, according to the language of the user’s session. Spring MVC is in charge of extracting the location and language from the session, controlling language cookies and updating its values if the user requests so (for example when the user selects a new language via the website’s interface). Spring MVC, in combination with the language interceptor, which intercepts the incoming request from the client and identifies the language, and assigns the new values to the session if the user has intentionally changed the language or if no language was defined in the profile.

Management of cookies for location and internationalization applies only to registered users with an active session. In case of unregistered users, the system is only able to identify the language of the web browser and nothing else.
4.3.5 Logging and Error Tracking

Logging is simply the process of writing log messages during execution time. Campusero uses a standard Java Logging Library with its default configuration.

Additionally to the standard Java logging library, Campusero implements a servlet that is invoked by the application server as soon as a `java.lang.Exception` is intercepted. The servlet renders an error view showing the identification code of the error and a brief description. This code could be provided by the users to the development team for identifying problems in the platform and enable a better customer support.

The logging system of Campusero allows us to report and persist error and warning messages as well as info messages (e.g. runtime statistics) so that the messages can later be retrieved and analysed.

Current log levels used in Campusero are:

- **Info**: Interesting runtime events.
- **Warning**: Unexpected conditions which could result in errors.
- **Error**: Runtime errors or unexpected conditions.

4.4 Significant Libraries and Frameworks

As a key motto when designing the architecture of Campusero, the code developed for the platform uses libraries and frameworks released under Open Source Licenses, which warranty future compatibility with any other open source products used in its development.

4.4.1 Libraries and Frameworks

Campusero uses multiple different libraries to achieve efficiently the fulfilment of the requirements. The use of Open Source libraries and frameworks is one of our priorities as well as part of our software production philosophy. That allows us to go on the path of the "Open Innovation", possible only if open tools are used in the development of Campusero.

From all the frameworks used in Campusero is necessary to remark the use of Spring instead of our original framework: Java Server Faces. Spring is much more flexible and modern as JSF and allows us to manipulate unconventional graphical interfaces (understanding for conventional those based on CRUD tasks). Spring also allows us to reduce the development time and enrich the platform interface.

Apache Maven was selected to reduce the complexity of the tedious process of maintaining the dependencies between the packages of Campusero. Maven replaced Ant, which was the initial tool used by the development team.

Behind other tools there are less decisions. EJB is part of the project because of its robustness, extensive support and reduced learning curve. PostgreSQL JDBC Driver is the standard driver to connect to a Postgres database. Furthermore, PostgreSQL Database was selected because is the strongest open source relational database in the world with a wide and clear documentation. Complementing PostgreSQL, ElasticSearch was introduced to reduce the load of the searching process on Postgres, besides being more agile in searches, as it is a server storage and distributed search tool.

Find below a list of the architecturally significant components as used in Campusero.
## 4.4.1.1 Backend

<table>
<thead>
<tr>
<th>Library</th>
<th>Ver.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring</td>
<td>3.2</td>
<td>The Spring Application Framework provides both a Dependency Injection framework and web application framework. Dependency injection promotes loosely coupled and highly pluggable development architecture. The web model-view-controller (MVC) architecture supports a clearly defined and well-understood model for developing web applications. The benefits include ease of extensions and maintainability. <a href="http://projects.spring.io/spring-framework/">http://projects.spring.io/spring-framework/</a></td>
</tr>
<tr>
<td>Apache Maven</td>
<td>3.0</td>
<td>Maven is a build automation tool. Software developers can use Maven to specify how to build software and to describe the dependencies of the software using an XML file called “pom.xml”. Maven download dependencies from the specified remote repositories to a local machine automatically. <a href="http://maven.apache.org/">http://maven.apache.org/</a></td>
</tr>
<tr>
<td>EJB</td>
<td>3.1</td>
<td>Enterprise JavaBeans (EJB) is a technology used in Java platform, typically in Java Enterprise Edition (Java EE). EJB runs in the server that uses the client/server architecture. It provides the advantage of the centralized architecture that software developers can make changes only on the server side, rather than on the clients. Furthermore, it provides features; e.g., session management, security, and transaction. <a href="https://jcp.org/aboutJava/communityprocess/final/jsr318/">https://jcp.org/aboutJava/communityprocess/final/jsr318/</a></td>
</tr>
<tr>
<td>PostgreSQL JDBC Driver</td>
<td>8.4</td>
<td>PostgreSQL JDBC Driver provides a standard set of Java interfaces to SQL-compliant databases. <a href="http://jdbc.postgresql.org/">http://jdbc.postgresql.org/</a></td>
</tr>
<tr>
<td>DSpace GeoIP</td>
<td>1.2.3</td>
<td>Provides geographical information of the user based on the IP. <a href="https://github.com/DSpace/dspace-geoip">https://github.com/DSpace/dspace-geoip</a></td>
</tr>
<tr>
<td>ElasticSearch</td>
<td>0.90.2</td>
<td>ElasticSearch is an open source distributed full-text search engine based on Apache Solr. It provides elasticity capability; i.e., developers can add or remove nodes as needed at runtime. The instance of ElasticSearch is called a “node” and a group of nodes is called a “cluster”. ElasticSearch supports partition indices (sharing indices) e.g., multiple indices by constructing an index with the same “type”. <a href="http://www.elasticsearch.org">http://www.elasticsearch.org</a></td>
</tr>
<tr>
<td>Name</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Google Guava</td>
<td>14.0.1</td>
<td>Guava is a Java project from Google that provides Java libraries; e.g.,</td>
</tr>
<tr>
<td>Jersey</td>
<td>1.17.1</td>
<td>Jersey is an open source framework that provides RESTful web services</td>
</tr>
<tr>
<td>Jackson</td>
<td>1.9.1</td>
<td>Jackson is an open source framework used to process data in Java platform;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>e.g., parsing, and data encoding (Avro, CSV, XML, Smile, YAML, and CBOR).</td>
</tr>
<tr>
<td>EHCache</td>
<td>2.5.0</td>
<td>EHCache is an open source Java distributed caching. It provides flexibility</td>
</tr>
<tr>
<td>Jericho</td>
<td>3.3</td>
<td>Jericho is an open source Java library used to analyse and parse HTML</td>
</tr>
<tr>
<td></td>
<td></td>
<td>application security control library..</td>
</tr>
<tr>
<td>Amazon Java SDK</td>
<td>1.3.2 7</td>
<td>AWS SDK for Java provides a Java API for AWS infrastructure services. It</td>
</tr>
<tr>
<td></td>
<td></td>
<td>is also used to store images in the storage services of Amazon and to send</td>
</tr>
<tr>
<td></td>
<td></td>
<td>emails.</td>
</tr>
<tr>
<td>RestFB</td>
<td>1.6.1</td>
<td>Simple and flexible Facebook Graph API and Old REST API client written in</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Java. Not used anymore.</td>
</tr>
</tbody>
</table>
## Libraries and frameworks (Backend)

<table>
<thead>
<tr>
<th>Library</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google API Client</td>
<td>1.6.0</td>
<td>Java client library for accessing Google APIs using JSON and OAuth 2.0. Not used anymore.</td>
</tr>
<tr>
<td>LinkedIn-J</td>
<td>1.0.41</td>
<td>Unofficial version of LinkedIn API. Not used anymore.</td>
</tr>
<tr>
<td>Json4S</td>
<td>3.2.2</td>
<td>JSON object manipulation library used to work with data in web services implemented on Scala.</td>
</tr>
<tr>
<td>Typesafe Slick</td>
<td>1.0.1</td>
<td>Slick is a database query and access library for Scala. It allows working with stored data through Scala collections and gives you full control over when a database access happens and which data is transferred.</td>
</tr>
<tr>
<td>Scala Test</td>
<td>1.9.2</td>
<td>Testing library for Scala.</td>
</tr>
<tr>
<td>OpenAM</td>
<td>10.1</td>
<td>OpenAM provides open source Authentication, Authorization, Entitlement and Federation software.</td>
</tr>
</tbody>
</table>

### Table 19. Libraries and frameworks (Backend)

#### 4.4.1.2 Frontend

jQuery and AngularJS are both Open Source projects with a fairly sophisticated philosophy in terms of software development. AngularJS is one of the most recommended libraries for multi page web applications, providing rich interfaces, robust data handling, agile development and easy maintenance.

Twitter Bootstrap is one of the most powerful tools that a web designer can use these days. It is now the most popular CSS library in GitHub.

<table>
<thead>
<tr>
<th>Library</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter Bootstrap</td>
<td>2.3.2</td>
<td>Bootstrap is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.</td>
</tr>
</tbody>
</table>
Future Internet Core Platform

Table 20. Libraries and frameworks (Frontend)

<table>
<thead>
<tr>
<th>Library</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Font Awesome</td>
<td>3.2.1</td>
<td>Icon library for Bootstrap</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="http://fortawesome.github.io/Font-Awesome/3.2.1/">http://fortawesome.github.io/Font-Awesome/3.2.1/</a></td>
</tr>
<tr>
<td>JQuery</td>
<td>Several</td>
<td>Cross-platform JavaScript library designed to simplify the client-side</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scripting of HTML.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="http://jquery.com/">http://jquery.com/</a></td>
</tr>
<tr>
<td>JQuery UI</td>
<td>Several</td>
<td>jQuery UI is a curated set of user interface interactions, effects, widgets,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and themes built on top of the jQuery JavaScript Library.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="http://jqueryui.com/">http://jqueryui.com/</a></td>
</tr>
<tr>
<td>AngularJS</td>
<td>1.0.7</td>
<td>AngularJS is an open-source web application framework, maintained by Google</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and community, that assists with creating single-page applications,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>one-page web applications that only require HTML, CSS, and JavaScript</td>
</tr>
<tr>
<td></td>
<td></td>
<td>on the client side.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="https://angularjs.org/">https://angularjs.org/</a></td>
</tr>
</tbody>
</table>

4.4.2 Development Tools
These are the development tools currently being used by the Campusero development team:

Table 21. Development tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netbeans</td>
<td>HTML, CSS, JavaScript, Java y Scala IDE ver 7.4+</td>
</tr>
<tr>
<td>Sublime</td>
<td>General Purpose IDE</td>
</tr>
<tr>
<td>GIT</td>
<td>Control Version Manager</td>
</tr>
<tr>
<td>Grunt</td>
<td>Javascript Task Runner, based on NodeJs used for automating front-end and</td>
</tr>
<tr>
<td></td>
<td>JavaScript workflow tasks.</td>
</tr>
</tbody>
</table>
4.4.3 Java Packages

4.4.3.1 Backend Subpackages Structure
This package holds all classes related to the services layers, not directly connected to the GUI presentation. It contains all the classes that would be frontend (GUI framework) agnostic.

<table>
<thead>
<tr>
<th>Package</th>
<th>Subpackage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractEjb</td>
<td>org.campusparty.social.ejb.*</td>
<td>EJBs interface container package(s)</td>
</tr>
<tr>
<td></td>
<td>org.campusparty.social.model.*</td>
<td>Models container package(s) (Data Transfer Objects)</td>
</tr>
<tr>
<td>SocialEjb</td>
<td>org.campusparty.social.ejb.*</td>
<td>EJBs implementation container package(s)</td>
</tr>
<tr>
<td></td>
<td>org.campusparty.social.model.*</td>
<td>Entities container package(s) (Data Transfer Objects)</td>
</tr>
<tr>
<td></td>
<td>org.campusparty.social.persistence.*</td>
<td>Abstractions and implementations of persistence beans (Data Access Object)</td>
</tr>
<tr>
<td>Social-api</td>
<td>org.campusparty.social.api.services</td>
<td>Implementations of business logic and data access</td>
</tr>
<tr>
<td>(Campusero’s private API implemented using Scala)</td>
<td>org.campusparty.social.api.model.*</td>
<td>Models (Data Transfer Objects)</td>
</tr>
<tr>
<td></td>
<td>org.campusparty.social.api.entity.*</td>
<td>Entities (Data Transfer Objects)</td>
</tr>
<tr>
<td></td>
<td>org.campusparty.social.api.rest</td>
<td>Web services</td>
</tr>
</tbody>
</table>
### AbstractService *
Abstraction of the logic associated with connection to external services (that which is outside the core Campuse.ro)

### ServiceImpl *
Implementation of the logic associated with connection to external services (that which is outside the core Campuse.ro)

### NotificationClient *
Notification client (reader)

### CpUtil *
Utilities of general use.

### Identity-api *
Implementation of an authentication system for development and testing.

<table>
<thead>
<tr>
<th>Package</th>
<th>Subpackage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social</td>
<td>org.campusparty.social.web.form.*</td>
<td>Package (s) container (s) of the representations of user forms.</td>
</tr>
<tr>
<td></td>
<td>org.campusparty.social.web.controller.*</td>
<td>Package (s) container (s) of the view controllers.</td>
</tr>
<tr>
<td></td>
<td>org.campusparty.social.web.filter</td>
<td>Package container filters applied to the client’s requests.</td>
</tr>
</tbody>
</table>

**Table 22. Subpackages structure (Backend)**

### 4.4.3.2 **Frontend Subpackages Structure**
This package holds all classes directly connected to the GUI presentation.

<table>
<thead>
<tr>
<th>Package</th>
<th>Subpackage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social</td>
<td>org.campusparty.social.web.form.*</td>
<td>Package (s) container (s) of the representations of user forms.</td>
</tr>
<tr>
<td></td>
<td>org.campusparty.social.web.controller.*</td>
<td>Package (s) container (s) of the view controllers.</td>
</tr>
<tr>
<td></td>
<td>org.campusparty.social.web.filter</td>
<td>Package container filters applied to the client’s requests.</td>
</tr>
</tbody>
</table>

**Table 23. Subpackages structure (Frontend)**
5 Quick User Guide

This is a very quick guide to the newly deployed functionalities of the Campusero platform. It is aimed at organization users, who are our main focus with this release, and to individual users in the case of some of the functionalities in the Challenges module. This user guide is written in second person. Each subchapter 5.x is dedicated to a new module, as developed for this release, and each subchapter 5.x.y is dedicated to a given new functionality inside such module. The new modules are “Organization User”, “Challenges” and “Talent”.

5.1 Organization User

5.1.1 Creating an Organization User Account

Whether you represent a public institution, a government, a company, a non-governmental/civil society, an educational institution, a user community or a venture capitalist, you can visit Campusero and easily register as an “organization user”, and be then ready to use our tools and interact with other users.

Once at Campusero’s homepage, click on the “Login” link.

A new screen will prompt, choose the “Join now” option.

If you are indeed representing an organization, choose the “Organization” option. Remember that registering as an organization gives you access to the talent search and the ability to propose challenges.
Then you will be asked to input your data. You should choose what sort of organization you are representing and then provide the following details:

- Name of your organization and a brief description,
- What type of organization you represent,
- Your website and social media profiles,
- Your contact info, and
- Your interests

It is highly recommended that you select and fill in your interests and your area of activity: this will help us to put you in contact with the right people and to recommend relevant contents to you in the future.

After filling in the boxes, save your changes and follow the onscreen instructions and you are done. You are now ready to manage your profile information and access all of Campusero’s tools.

5.1.2 Managing your Profile

Once logged in, go to the main menu and click on your profile image to access your profile dashboard. This is what you will find:
This is the place where you can
Future Internet Core Platform

- see, edit or delete your profile,
- find your conversations with other users,
- see who is following you and who you follow, and
- see the activities you engaged in.

For you to edit your profile, you should click the “edit” button. You will be leaded to the profile form. This is how the form looks like:

![Profile form](Figure 23. User guide: Profile form)
5.2 Challenges

These are the key steps and procedures for managing the whole lifecycle of the Campusero challenges.

5.2.1 Creating a Challenge

Any organization that is registered in Campusero can easily launch a challenge.

1. Wherever you are within www.campuse.ro, if logged-in, click on the “Create” button, and a pop up window will appear, containing several options. Choose “Challenges”.

2. A preliminary form will appear (see next figure). Its aim is to give an overview of what we need from you. You will be kindly asked to fill in all the fields (title, description, budget, estimated budget and prize/s plus contact details). Shortly, a representative of Campusero will get in touch to propose you the product that best fits your needs.
3. Once you have filled in the preliminary form, you will be led to the “Contest form” (see next figure), in which all the details of the challenge should be specified and customized. (e.g. by uploading your own challenge’s image, among others).

Additionally, two templates will be provided: (1) one is to be filled in by all the participants, to ensure they follow the same formal and visual procedures for all proposals and (2) then there is a second template for covering the Legal Basis of the challenge, which you will be able to adapt and modify as well. At this point, you can either save the challenge as a draft or launch it and start receiving ideas right away. For the latter, the administrator user of Campusero should have granted you permission after a joint agreement between both parties, and you must have completed all the required fields, as marked with an asterisk. Then you can click on the button “Publish” at the end of the form.
Figure 26. User guide: Contest form
4. Congratulations: the challenge will be shown in the Challenge page (see next figure), regardless which status you have chosen (“Save as a draft” or “Publish”). If for any reason you need to change anything, you will have the chance to edit the challenge from your organization user dashboard, even after it has been published.

Figure 27. User guide: Description
5.2.2 Managing the Challenge

When a challenge has already been created and is open to ideas, individual users and organizations can upload their proposals. In order to manage your challenge, you should be logged-in and click on your profile image at the main menu.

![Figure 28. User guide: manage the challenge](image)

You will be led to your user dashboard. This is the place where you will find your challenges. Click the challenge to manage all the options and edit it if needed.

![Figure 29. User guide: challenges at the dashboard](image)

From this moment on, you, as the organization user responsible of the challenge, will also be able to download the ideas submitted by other users and see how many of them are following the challenge and/or applying, as well as check which their profiles are (see next figure).

Once the challenge has been closed for further submissions, you will have to evaluate all the ideas and select the corresponding winner/s and/or finalist/s. This can be made by selecting the specific option (“Winner” or Finalist) at the top left menu and then choosing the appropriate order directly on each candidate’s information row (pop-up on the right). For instance, if you have selected “Winner” and “1”, this means that this user has the pole position and will be awarded with the first prize in the challenge; if you select “Winner” and “2”, this means that the user is the second-
position awardee; if you select “Finalist” and “1”, this means that the user has been well ranked but has no allocated prize.

Figure 30. User guide: managing evaluations

5.2.3 Participating in a Challenge

Any individual user or organization user previously registered in Campuse.ro can apply to a challenge.

All challenges open to applications appear at Campusero’s homepage, among other contents. However, it is easier to see all of them by visiting the Challenges page (see next figure), by clicking on the “Challenges” button in the main menu.
As indicated, you can access it from anywhere within www.campuse.ro, by clicking the “Challenges” button at the top left menu:

Once here, you must complete the following steps to participate and submit your proposal in one or more challenges:

1. Click on the corresponding challenge
2. Then, you will be led to the Challenge’s details page

3. Download the Legal Basis with all the instructions and the template to present your idea, proposal or project. It is mandatory to present your idea in the provided template and respecting the Legal Basis.

4. Click on the “Participate” button at the top-right menu. You will be led to the challenge’s submission form (see next figure). If you had not already downloaded the Legal Basis and the template to submit your idea, you will once more have the chance to do so. The challenges you have accepted will appear in your user dashboard in case you want to submit your proposal in the future.

5. Now, please upload the template completed with your idea and any additional files that you consider important (either requested or not). These files should be compressed into a .ZIP-format file and not exceed 4 MB.

6. Click on the “Upload proposal” button. Wait for the confirmation message and good luck.
5.3 Talent

Use the Talent Search tool to find appropriate professionals for your human resources needs. You will be able to browse through Campusero’s individual user base and find relevant candidates. Also, you can publish job opportunities and wait for candidates to show.

5.3.1 Searching for Talent

To start the search, click the “Talent Search” button in the main menu.
You will be guided to a page in which you can see individual users as registered in Campuse.ro and you will also be able to refine your filtering criteria, by calibrating:

1. Skills/Interests
2. Geography
3. Experience
4. Age

The image below represents the screen you will find:
Once you access to the search results, you can click on the user profiles in which you are interested and contact the users through the messaging system.

5.3.2 Job Posting

To create a job offer, click on the “Create” button and choose the appropriate option. You will be led to the following form:

![Create a job offer](image)

Figure 36. User guide: Creating a job offer

You should fill in the form with the information for the job offer and click on the “publish” button. This is how your job offer will appear to other users:
Once you have published the job offer, it will be available for individual users in both the Campuse.ro main page and through the “Jobs” link in the main menu. Individual users will be able to apply for the offer. When they do, you will receive a message with their application. This is how job offers appear for individual users in the “Jobs” menu:
5.3.3 Managing the Job Offers

The job offers you have created will appear in your user dashboard. This is the place where you can edit the offer’s information or delete it if you have already selected someone for the position.

This step comes when a job offer has been created and is open to ideas, so individual users and organizations can upload their proposals. In order to manage your job offers, you must be logged-in and then click on your profile image in the main menu.
You will be guided to your user dashboard. This is the place where you will be able to check the job offers that you have already posted. Click on the specific challenge to manage its details and download the applications already made.
Figure 40. User guide: Job offers at the dashboard
6 Future Steps in the development of Campuse.ro

The enterprises and public and private organizations are major players in the European FI ecosystem, and so we have created specific tools for profiling organizations: everyone benefits from the entry of companies and organizations in a tight relationship within the community of Campuse.ro. Accordingly, new permissions and different ways for managing activities have been created for these new profiles.

Beyond this, two new modules have been created: “Talent” and “Challenges”.

In the case of Talent, the tool we have created allows companies and entrepreneurs to efficiently find experts in different areas of knowledge, to develop a specific task or to fill a job vacancy.

In the case of Challenges, they now enable direct interaction between the campuseros and organizations seeking to find on-the-spot solutions and innovative ideas using special contests.

The immediate future steps in the evolution of Campuse.ro, that will see the light in the next release, will focus on:

1) Continuing the work on the backend systems that deliver recommendations of content based on the interest of each user. This work is already taking place, and all the relevant modules previously developed have been designed with this in mind.

2) Further encouraging the use of Campusero through awards, reputation and recognitions for the use of FI-WARE technologies (also called Gamification).

These improvements will be part of the next version of Campuse.ro, to be released in September 2014 as part of the EU-funded FI-WARE project.