



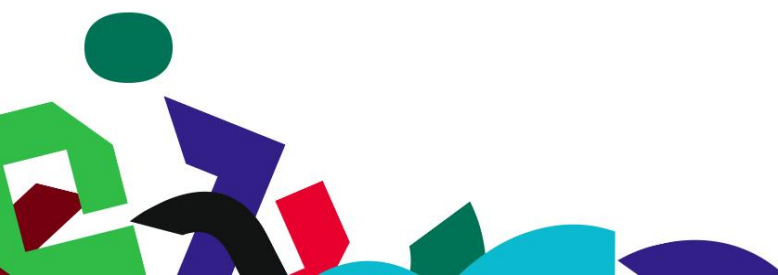
Deliverable 2.5 EPICommunity design

European Partnership for an Innovative Campus Unifying Regions
EPICUR Research Agenda
Shaping European Society in Transition

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016926.

Deliverable number	D7
Relative number in WP	D2.5
Deliverable title	EPICommunity design
Work package	WP2 EPICommunity as basis for strengthening human capital
Lead beneficiary	AUTh
Dissemination Level	PU
Type	R
Due date (in months)	June 2022 (M18)

The main objective of D2.5 is to design a non-commercial, open and inclusive registry of European researchers and the necessary platform for creating a European Social Network of researchers (EPICommunity) to support academic matchmaking and collaboration.



Author(s)		
Name	Organisation	E-mail
Thras Tsiatsos	AUTh	tsiatsos@office365.auth.gr

Status of deliverable		
//	By	Date
Submitted (Author(s))	Thras Tsiatsos	17.06.2022
Verified	PMT	23.06.2022
Approved	PMT SC	23.06.2022 08.07.2022

Change of history		
Version	Date	Description of change
01	09.04.2022	Use Cases proposal
02	20.04.2022	Use Cases final version
03	23.05.2022	Draft Functionality
04	17.06.2022	Final version for PMT
05	28.06.2022	Modifications in section 6 Conclusions and next steps

Contents

Contents	3
List of Figures	5
List of Tables	7
1 Introduction	10
1.1 Context.....	10
1.2 Objective of D2.5	10
1.3 Methodology	10
1.3.1 About the Use cases.....	11
1.3.2 About System Interface and System functionality.....	11
2 Researcher profile	11
2.1 State of the art of Social Networks.....	12
2.2 Profile characteristics common standards	14
2.3 Towards a social network designed for researchers by researchers	15
2.4 EPICommunity: Researcher's profile characteristics	16
3 Definition of EPICommunity actors.....	16
3.1 General Roles.....	16
3.2 Main entities.....	17
4 EPICommunity Use Cases.....	17
4.1 Access and Registration / Login.....	17
4.1.1 User Registration.....	18
4.1.2 User Login	18
4.2 Entities interaction.....	19
4.2.1 User to user	19
4.2.2 Recommended entities to the user by the platform	23
4.2.3 User interaction with platform entities	24
4.3 Profile Related Use Cases	25
4.3.1 Profile Management (Create / Read / Update / Delete)	25
4.3.2 Profile Import	28
4.3.3 Profile Export.....	29
4.4 Use Cases for the Entities Courses, Publications and Projects (Create / Read / Update / Delete) ..	29
4.4.1 Management of Courses, Publications and Projects.....	30
4.4.2 Search of Courses, Publications and Projects	32
4.5 Group Related Use Cases	33
4.5.1 Group Management (Create / Read / Update / Delete).....	33
4.5.2 Group Member Management.....	35
4.5.3 Group Enrolment.....	36
4.5.4 Leave Group	37

4.6	Badges Use Case (Create / Read / Delete)	38
5	EPICommunity Functionality	39
5.1	Access and Registration / Login	39
5.1.1	User Registration	39
5.1.2	Email Verification	41
5.1.3	Resend Email Verification	42
5.1.4	User Login	43
5.1.5	Password Reset	44
5.2	Profile Functionality	46
5.2.1	Create	46
5.2.2	Read	53
5.2.3	Update	64
5.2.4	Delete	66
5.2.5	Export	67
5.2.6	Sync	67
5.3	Entities Functionality	68
5.3.1	Create	68
5.3.2	Read	73
5.3.3	Update	79
5.3.4	Delete	81
5.3.5	Endorsement	82
5.3.6	Comment	82
5.3.7	Share	83
5.4	General Functionality	83
5.4.1	Dashboard	83
5.4.2	Notification	85
5.4.3	Send Contact Request	86
5.4.4	Private Message	86
5.4.5	Assessment	87
5.5	Groups Functionality	89
5.5.1	Create	89
5.5.2	Read	90
5.5.3	Update	95
5.5.4	Delete	96
5.5.5	Create Announcement	96
5.5.6	Enrolment	97
5.5.7	Leave Group	98
5.6	Search Functionality	98

5.6.1	Entities (Courses/Publications/Projects)	98
5.6.2	Users.....	99
5.7	Recommendations Functionality.....	100
5.7.1	Users.....	100
5.7.2	Entities (Courses/Publications/Projects)	101
5.7.3	Groups	102
6	Conclusion and next steps	103
7	References	104

List of Figures

Figure 1: WP2 tasks interconnection	10
Figure 2: Social networks grouped by domain (personal – professional – academic).....	13
Figure 3: Shared characteristics between social networks	14
Figure 4: Europass and EPIQAsses relation to the EpiCommunity profile characteristics	15
Figure 5: Access and Registration Diagram.....	18
Figure 6: Members' Entities Interactions Diagram.....	19
Figure 7: Subscribers' Search and Visit Profile Diagram.....	19
Figure 8: Diagram for the Profile Related Use Cases	25
Figure 9: Diagram for Management/Search of Courses, Publications, Positions and Projects by Members.	30
Figure 10: Diagram for Search of Courses, Publications, Positions and Projects by Subscribers.....	30
Figure 11: Group Management and Member Management Diagram.....	33
Figure 12: Leave Group Diagram	37
Figure 13: Diagram for View Badges Use Case	38
Figure 14: User Registration Screen	40
Figure 15: User Registration Screen with errors	40
Figure 16: Email Verification Screen	41
Figure 17: User Login Screen.....	43
Figure 18: User Login Screen with errors.....	43
Figure 19: Request Password Reset Link Screen.....	44
Figure 20: Password Reset Screen	45
Figure 21: Role Selection Screen	46
Figure 22: Profile Create – Research Information Screen.....	47
Figure 23: Profile Create – Personal Information Screen	48
Figure 24: Profile Create – Courses Publications, Projects, Positions Screen.....	49
Figure 25: Profile Create – Create Project Modal Screen	49
Figure 26: Profile Create – Courses, Publications, Positions , Projects Edit/Delete Options Screen.....	50

Figure 27: Profile Create – Skills Screen	52
Figure 28: Profile Read / General Information Screen	53
Figure 29: Profile Read / Courses Screen	55
Figure 30: Profile Read / Projects Screen	56
Figure 31: Profile Read / Publications Screen	57
Figure 32: Profile Read / Positions Screen	58
Figure 33: Profile Read / Skills Screen	59
Figure 34: Profile Read / Groups Screen.....	60
Figure 35: Profile Read / Badges Screen.....	61
Figure 36: Profile Read / Friends List Screen.....	62
Figure 37: Profile Read / Extra Options menu Screen	63
Figure 38: Profile Read / Contact Requests Screen	64
Figure 39: Profile Dropdown Options	65
Figure 40: Profile Update Screen	65
Figure 41: Profile Delete Screen	66
Figure 42: Profile Export Screen	67
Figure 43: Course Create Screen.....	68
Figure 44: Publication Create Screen.....	70
Figure 45: Position Create Screen	71
Figure 46: Project Create Screen	72
Figure 47: Course Read Screen	74
Figure 48: Publications Read Screen	76
Figure 49: Position Read Screen.....	77
Figure 50: Project Read Screen	78
Figure 51: Edit and Delete options Screen	79
Figure 52: Course Update Screen	80
Figure 53: Course Delete Screen	81
Figure 54: Dashboard Screen.....	84
Figure 55: Notifications Screen.....	85
Figure 56: Another Members' Profile Screen.....	86
Figure 57: Private Message Screen	87
Figure 58: Assessment Screen	88
Figure 59: Group Create Screen	89
Figure 60: Groups List Screen	90
Figure 61: Group General Information Screen.....	91
Figure 62: Group Members Screen.....	92

Figure 63: Group Wall Screen	93
Figure 64: Group Membership Requests Screen	94
Figure 65: Group Update Screen	95
Figure 66: Group Side Menu	96
Figure 67: Create Announcement Screen	97
Figure 68: Search Courses Screen	98
Figure 69: Recommended Users Screen	100
Figure 70: Recommended Publications Screen	101
Figure 71: Recommended Courses Screen	101
Figure 72: Recommended Groups Screen.....	102

List of Tables

Table 1: Template for Use Cases.....	11
Table 2: Template for Functionalities.....	11
Table 3: Overview of EPICommunity Roles and Rights per Component.....	17
Table 4: Use Case - User Registration	18
Table 5: Use Case - User Login	18
Table 6: Use Case - Search Profile	20
Table 7: Use Case - Visit Profile	21
Table 8: Use Case - Contact Request.....	22
Table 9: Use Case - Private Message	22
Table 10: Use Case - Evaluate User.....	23
Table 11: Use Case - Recommended entities to the user by the platform	23
Table 12: Use Case - User interaction with platform entities	24
Table 13: Use Case - Profile Management	25
Table 14: Use Case - Profile Import	28
Table 15: Use Case - Profile Export.....	29
Table 16: Use Case - Management of Courses, Publications and Projects	30
Table 17: Use Case - Search of the Courses, Publications and Projects	32
Table 18: Use Case - Group Management.....	33
Table 19: Use Case - Group Member Management	35
Table 20: Use Case - Group Enrolment.....	36
Table 21: Use Case – Leave Group	37
Table 22: Use Case - Badges	38
Table 23: Functionality – User Registration	40

Table 24: Functionality – Email Verification	41
Table 25: Functionality – Resend Email Verification	42
Table 26: Functionality – User Login.....	43
Table 27: Functionality – Password Reset	45
Table 28: Functionality – Role Selection	46
Table 29: Functionality – Profile Creation / Research Information.....	47
Table 30: Functionality – Profile Creation / Personal Information	48
Table 31: Functionality – Profile Creation / Courses Publications Projects.....	50
Table 32: Functionality – Profile Creation / Skills.....	52
Table 33: Functionality – Profile Creation / General Information	53
Table 34: Functionality – Profile Read / Courses	55
Table 35: Functionality – Profile Read / Projects	56
Table 36: Functionality – Profile Read / Publications	57
Table 37: Functionality – Profile Read / Positions	58
Table 38: Functionality – Profile Read / Skills	59
Table 39: Functionality – Profile Read / Groups	60
Table 40: Functionality – Profile Rad / Badges	61
Table 41: Functionality – Profile Read / Friends List	62
Table 42: Functionality – Profile Read /Contact Request	64
Table 43: Functionality – Profile Update Research and Personal Information.....	65
Table 44: Functionality – Profile Delete	66
Table 45: Functionality – Profile Export	67
Table 46: Functionality – Profile Sync.....	67
Table 47: Functionality – Course Create	68
Table 48: Functionality – Publication Create	70
Table 49: Functionality – Position Create.....	71
Table 50: Functionality – Project Create.....	73
Table 51: Functionality – Crouse Read	74
Table 52: Functionality – Publication Read Screen.....	76
Table 53: Functionality – Position Read.....	77
Table 54: Functionality – Project Read	78
Table 55: Functionality – Entities Update.....	80
Table 56: Functionality – Entities Update.....	81
Table 57: Functionality – Entities Endorsement	82
Table 58: Functionality – Entities Comment	82
Table 59: Functionality – Entities Share	83

Table 60: General Functionality – Dashboard	84
Table 61: General Functionality – Notification	85
Table 62: General Functionality – Send Contact Request	86
Table 63: General Functionality – Private Message	87
Table 64: General Functionality – Assessment	88
Table 65: Groups Functionality – Create.....	89
Table 66: Groups Functionality – Groups List.....	90
Table 67: Groups Functionality – Group General Information.....	92
Table 68: Groups Functionality – Group View Members.....	93
Table 69: Groups Functionality – Group Wall	93
Table 70: Groups Functionality – Manage Group Membership Requests.....	94
Table 71: Groups Functionality – Update Group	95
Table 72: Groups Functionality – Delete Group	96
Table 73: Groups Functionality – Create Announcement.....	97
Table 74: Groups Functionality – Group Enrolment	97
Table 75: Groups Functionality – Leave Group	98
Table 76: Search Functionality – Courses/Publications/Projects	99
Table 77: Search Functionality – Users	99
Table 78: Recommended Functionality – Users.....	100
Table 79: Recommended Functionality – Entities (Courses/Publications/Projects)	102
Table 80: Recommended Functionality – Groups	102
Table 81: EPICommunity Use Cases.....	103
Table 82: EPICommunity Functionalities.....	103

1 Introduction

1.1 Context

This deliverable is part of Task 2.3 EPICommunity Design (M13-M18) of Work package 2 “EPICommunity as basis for strengthening human capital”. This work package aims to enhance sustainable research careers in the context of the European Research Area and is broken down in four tasks:

- **Task 2.1.** It elaborated EPICUR’s researchers’ framework (EPIQAsses) and will combine this with a gamification framework, namely incentives and rewards (EPIGame)
- **Task 2.2.** It will elaborate blended mobility formats (EPIMove) for researchers and will produce policy recommendations to the EC on the needs of early career researchers in the context of blended mobility, with a view to the new instruments in the next Erasmus+ programme
- **Tasks 3+4.** Tasks 2.3 and 2.4 will be devoted to design and implement the EPICommunity prototype by exploiting (a) EPIQAsses and (b) EPIGame (see Figure 1)



Figure 1: WP2 tasks interconnection

The main steps of Task 2.3 are:

Main steps of this task are the following:

- (a) **Analysis of the state of the art of researchers’ social networks** (e.g., Research Gate, LinkedIn, Google Scholar, etc.) or platforms: This step will basically collect common design issues, user profiling issues, etc. in order to give input to the EPI-Community design.
- (b) **Design of European Alliance Researcher profile qualitative criteria for assessing skills:** this step will take into account the results of Task 2.1.
- (c) **Design of networking functionality among researchers** (for example joint projects, proposal writing, common research, data exchange, peer review, etc.). UML notation and mock-ups will be exploited for the design of the EPICommunity functionality.

1.2 Objective of D2.5

The main objective of D2.5 is “to **design** a non-commercial, open and inclusive registry of European researchers and the necessary platform for creating a European Social Network of researchers (EPICommunity) to support academic matchmaking and collaboration [...]”.

This deliverable will give input to deliverable D2.6 in order to “[...] further **develop** the EPICommunity (technically), to be open and accessible for members of other European University alliances and beyond. Like this, EPICUR will offer services to researchers across the globe who are able and willing to collaborate in order to solve global (EPI-) challenges”.

1.3 Methodology

This paragraph is briefly presenting the methodology for describing and deciding the EPICommunity Design. More specifically three workshops have been conducted with the EPICUR-Research WP2 experts and the participation of EPICUR ECR board members to define:

- The methodology and the templates for the Use Cases and the System Functionality (see paragraphs 1.3.1 and 1.3.2)

- The discussion and finalization of the Use Cases
- The discussion and the finalization of EPICommunity Functionality

1.3.1 About the Use cases

The section on Use Cases (**Section 4 - EPICommunity Use Cases**) describes the context of the system and the problem(s) that it solves. The aim is to introduce the system that is accessible to non-domain experts. The problem description enumerates the key features of with the system and how the system provides value to them. The focus of this section is on the features concerned with and communicating with the system, and on the roles of these features, not on the system itself.

Use cases have been adopted to extract the main components resulting (mainly) from the previous WP2 tasks:

We use these use cases in order to capture the intended behavior of the system without having to specify how that behavior is implemented. These use cases will provide a way for developers and users to come to a common understanding on the functionality of the intended system.

The use cases are divided into basic groups concerning the different end user target groups and entities. The agreed format for the use cases, as decided by the EPICUR-Research partners, is presented at Table 1.

Table 1: Template for Use Cases

Use Case	Use case identifier and reference number and modification history
Description	Goal to be achieved by use case
Actors	List of actors involved in use case
Assumptions	Conditions that must be true for use case to terminate successfully
Steps	Interactions between actors and system that are necessary to achieve goal
Variations	Any variations in the steps of a use case
Non-Functional	List any non-functional requirements that the use case must meet
Issues	List of issues that remain unresolved

1.3.2 About System Interface and System functionality

The section on system interface and System Functionality (Section 5) documents the services that the system provides in terms of responsibilities. Often the system interface may be organized into a set of sub-interfaces, each sub-interface corresponding to a distinct usage of the system, e.g., there may be specific interfaces for system configuration, for normal system use, and for system management. This document is mainly focused on the normal system use. Each system interface will be presented as a rapid prototype. Along with the system interfaces the specific system functionality are described using the template presented in Table 2:

Table 2: Template for Functionalities

Functionality Name	Name of the functionality
Relative Use Case(s)	Use Case(s) where the functionality could contribute
Actors	Users or system components that will use this functionality
Input Data	Data needed for the implementation
Description	Process that will be followed for the achievement of the functionality
Output Data	Expected result

2 Researcher profile

At EPICommunity, we aim to create a space where researchers can provide information for their career, research interests and aspirations moving forward with their research. We argue that a research profile is a combination of the different activities that the researcher becomes a part of while she is evolving.

These activities can include but are not limited to:

- Publications: one of the most integral activities for a researcher. Published work establishes the findings of the research and circulates them to the academic community through the peer review process.
- Courses: very often researchers are involved in teaching or supporting academic courses. This is also a very important activity since it adds the educational / teaching dimension to a research profile.
- Projects: researchers that are involved in research projects, especially European Union funded projects, are unlocking several different skills because of this involvement. Technical writing. Design of applied research-oriented solution. Social skills for cross-disciplinary, border agnostic, diverse communication, and work.
- Groups: depending on their research interests, researchers are often organized in groups, to discuss and exchange experiences. Open science resources like Arxiv, Zenodo and so forth have enabled this idea exchange by safeguarding the intellectual property of the original authors.

Such activities, as the ones presented above, make clear that a researcher's profile can combine traits and skills from different dimensions. If we add, on the top of that, the fact that a researcher can work with different research fields during her career we have a rich, complex and diverse set of information to present her fellow researchers.

Contemporary digital communities, however, have come to incorporate a set of similar features and, depending on their target audience to inherit specific styling in terms of visuals, language and actions that their members can perform. The most used such communities are the different social networks (i.e. Facebook, Instagram, Twitter). As the WWW evolves, social networks evolve as well. In the beginning we started with plain simple social networks like Facebook, LinkedIn, Google+, functioning mainly as an agenda of contacts online. Those social networks evolved to offer several different functionalities but alongside them other types of social networks started appearing.

- Microblogging: Twitter was the first microblogging platform to introduce a social network that allowed its users to post updates limited to a small number of characters.
- Academic: Platforms like Research Gate and Academia are similar to the classic social networks (like Facebook) but they are built to accommodate the research community, mainly exchanging information in the form of updates for publications and project involvement.
- Decentralized: The blockchain revolution made the concept of decentralization a trend. And while, peer-networks were being used for a long time for content sharing, WiFi sharing and so forth, this new era of decentralization also brought social networking platforms with it. In fact we are seeing social network alternatives emerging for almost every one of the most known social network platforms (i.e. PeerTube for YouTube, Mastodon for Twitter and so forth).
- Communication apps as social networks: Although they are not falling to the social network category, lately many communication applications have started incorporating social features. Viber, WhatsApp and Signal are offering a group feature where, the members of the group have a public profile, usually associated with little information like their name and their e-mail or phone, a profile picture and so forth. We are mentioning that last category of apps just to stress the fact that more and more applications are currently offering "social features" without being social networking platforms per se.
- Complementary apps for academics: These are applications that academic institutions use for internal management and project management. Universis.eu and OLAS by Divinus Soft are two such solutions.

Following we are going to present the state of the art regarding the social networking platforms, their different goals, features and audiences focusing on the characteristics of the profiles of the users of these platforms. After that we will move on with presenting our reasoning on choosing the set of characteristics that we believe that best serve in the context of the EPICommunity.

2.1 State of the art of Social Networks

In the image below we are presenting different social network applications grouped by domain related to the scope of usage. Personal, where we expect more casual, everyday content, professional, where we expect expert, professional content and key updates related to the user's career and academic, where we expect research and scientific specific content alongside key updates to the user's academic endeavours.

The State of Social Networks



Figure 2: Social networks grouped by domain (personal – professional – academic)

If we observe carefully the different social networks that fall to the aforementioned categories, we will find that there is an overlap in certain characteristics: Full name, profile image, phone number, address, website URL, e-mail address and so forth. These characteristics are serving the purpose of identifying the user of the platform and hence, are pretty much essential for every social network.

Similar characteristics, available in most of the social networks seem to be timeline related characteristics: Professional work, studies, certifications, courses, awards and so forth. For the professional related social networks (like LinkedIn) this is considered an essential feature since these social networks are also used as a source of candidate hires from recruiters and HR departments. For the more casual social networks, these characteristics usually serve the purpose of showcasing important highlights of a profile. Vanity related characteristics (i.e., likes, shares and so forth) are frequent used in social networks to enable user generated content.

At this point we would like to stress the fact that simpler social networks have managed to become relevant in different domains. For example, Twitter is being used from personal, professional and academic users to share relevant content. This is very important because it shows that the usage of a social network is not necessarily limited when the network itself offers limited characteristics.

For the EPICommunity, being a community targeting mainly members of the research and academic community, we will be basing our research on the pre-existing academic related social networking platforms. Therefore, academic usage becomes the dominant domain and our goal is to investigate whether other characteristics, found in social networks from other domains, would be valuable to EPICommunity. Since we have already discussed how some characteristics are shared between domains, the pool of characteristics is organized as follows.

The State of Social Networks



Figure 3: Shared characteristics between social networks

Academic usage social platforms, our dominant dimension is in the center and shares characteristics with the two neighbouring dimensions containing social networks of personal and professional usage respectively.

2.2 Profile characteristics common standards

Before we move on with the selection of EPICommunity's features we have to address an issue that becomes apparent from the aforementioned analysis. While the three domains share similar features when it comes to the user profile, there is no common standard between them. LinkedIn has its own way to describe a user's working experience, Facebook and ORCHID might use a slightly different way of doing that. The EPICommunity aims in becoming a reference point for the European researchers' community in terms of discovering talent, forming cooperations and enabling cross-disciplinary work. That is why we need to find a common standard to base the EPICommunity's profile to.

After analysing the state-of-the-art social networks available for researcher we came to the conclusion that the Europass CV schema¹ incorporates the majority of the characteristics that efficiently promote the skills of the researchers and enable the search functionality.

“Europass is a European Union (Directorate General for Education and Culture) initiative to increase transparency of qualification and mobility of citizens in Europe. It aims to make a person's skills and qualifications clearly understood throughout Europe (including the European Union, European Economic Area and EU candidate countries). The five Europass documents are the Curriculum Vitae, Language Passport, Europass Mobility, Certificate Supplement, and Diploma Supplement, sharing a common brand name and logo, Since 2012 individuals have been able to assemble all Europass documents in the European Skills Passport.” (source: Wikipedia)

We believe that Europass is the ideal candidate to solve our standardization problem regarding user profiles on EPICommunity because:

¹ <https://europa.eu/europass/en/create-europass-cv>

- It has been already used across the European Union and most of the Europeans, especially researchers actively participating in EU funded projects are familiar with it.
- It includes most of the key characteristics that we want to incorporate in EPICommunity.
- It offers portability allowing to export the data to a machine-readable format. This way the user will be able to import her Europass profile to EPICommunity easily.

The State of Social Networks

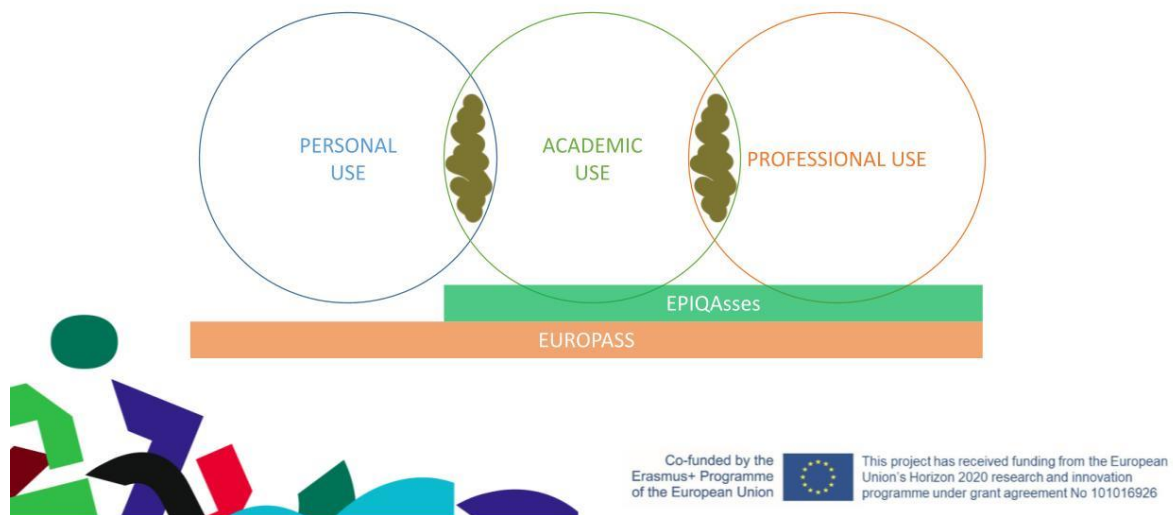


Figure 4: Europass and EPIQAsses relation to the EpiCommunity profile characteristics

In the image above we are presenting how the Europass can serve as the common layer between the characteristics on the different platforms. By using Europass, we add standardization to our EPICommunity's users profile.

Another element, worth mentioning is the EPIQAsses features that we full adopt (as described on the deliverable D2.1) to be used as qualitative criteria.

2.3 Towards a social network designed for researchers by researchers

We believe that EPICommunity can serve as a social network incorporating all those features that can foster research and collaboration rather than be “yet another social network”. Researchers' time is limited and valuable. Moreover, their work reflects on their publications and academic activities and this is why they are not very much interested in utilizing marketing tools and / or vanity metrics to promote their work. What they do need though are ways to easily find fellow researchers to collaborate, institutions and organizations to partner with on proposals writing or as contractors, teams or laboratories to join for post-doctoral research and so forth.

That is why we designed EPICommunity to serve as a registry, allowing for the easy discovery of high-quality research profiles. Like LinkedIn the members of EPICommunity are the competitive advantage of the platform. The selection of the aforementioned features on a researcher's profile gives our members the opportunity to effectively research future collaborators. On a second level our members can make use of community like features (like participating in groups). This way they can have a quick way of coordinating for a common goal (for example to prepare a project proposal). On the third level, the platform is trying to engage or kickstart such communications by recommending to its members material and contacts relevant to them.

2.4 EPICommunity: Researcher's profile characteristics

Based deliverable D2.3-EPIGame, we identify a set of characteristics that we consider key for the researcher's profile platform wise.

1. First name
2. Last name
3. Date of birth
4. Gender
5. Place of birth
6. University Name
7. University Faculty/Department
8. Skills
 - Communication/Interpersonal
 - Organizational
 - Digital
 - Language
9. Profile description/annotation (form of hashtags)
10. Publications
11. Projects
12. Work experience
13. Courses

The reasoning behind the selection of these characteristics is the following. Characteristics 1-5 are providing key information about the identity of the user mainly related to demographics. Characteristics 6-7 are related to the Institution with which the user is related. This is key information since many of the activities of the EU (and EPICUR Alliance activities) require groups by researchers of different academic institutions. Characteristics 8-9 provide work related information (skills and interests). Characteristics 10-13 are lists of elements directly linked with the activity of the user in all the different dimensions that we described in the beginning of 2.3.

3 Definition of EPICommunity actors

3.1 General Roles

The main roles are the following:

- Guest
- Subscriber
- Member
- Moderator
- Administrator

ADMINISTRATOR is responsible for each of the EPICommunity components.

The platform is divided in the following parts:

- The Public Space (Web site)
- The Community
- The Administration Panel

Each of those sites can be accessed from the central homepage. Each part has distinct roles. The public can access parts of the information on the platform as a GUEST.

SUBSCRIBERS are users that are signed in to the EPICommunity not as ECRs. Even though they are signed in users, they do not have their own profile entities, they cannot evaluate other users, neither join nor create a

group. The permissions that this role has, is to be able to use the search mechanism of the platform, view profiles and their entities, as well as view group and their information.

MEMBER users have increased rights in comparison to the GUEST and SUBSCRIBER users in the sense that they can interact with the platform instead of just consume content. Specifically for the GROUP entity a user can have the role of MODERATOR, if he is the creator of the GROUP.

The ADMINISTRATOR of all the components is one.

Table 3 presents and overview of EPICommunity Roles and Rights per Component

Table 3: Overview of EPICommunity Roles and Rights per Component

Component	Roles and Rights
Public Web Site	Roles and Rights for Public Web Site
Guest	Access the front page of the platform. Access general purpose informative pages. Access generic metrics of the platform (i.e., number of projects, number of groups, number of members).
Community	Roles and Rights for Community
Member / Moderator	Access to logged in users' information (full users' profiles and related entities). Profile and Entities search. Entities Interactions. Matchmaking & Recommendation capabilities. Peer review.
Community	Roles and Rights for Community
Subscriber	Access to logged in users' information (full users' profiles and related entities). Profile and Entities search.
Administration Panel	Roles and Rights for Administration Panel
Administrator	Technical Role. Management of overall Users, Entities, Groups Responsible for Data Consistency. Responsible for Bug Fixes. Responsible for analytics reports. Responsible for site reliability.

3.2 Main entities

The main entities as described in deliverable D2.3-EPICGame are the following:

- User
- Project
- Group [Open / Closed]
- Courses
- Publications
- Positions

4 EPICommunity Use Cases

4.1 Access and Registration / Login

The Figure 5 shows a diagram for the use cases of the Access and Registration / Login. The actual use cases are two and will be thoroughly analyzed in the corresponding paragraphs:

- User Registration

- User Login

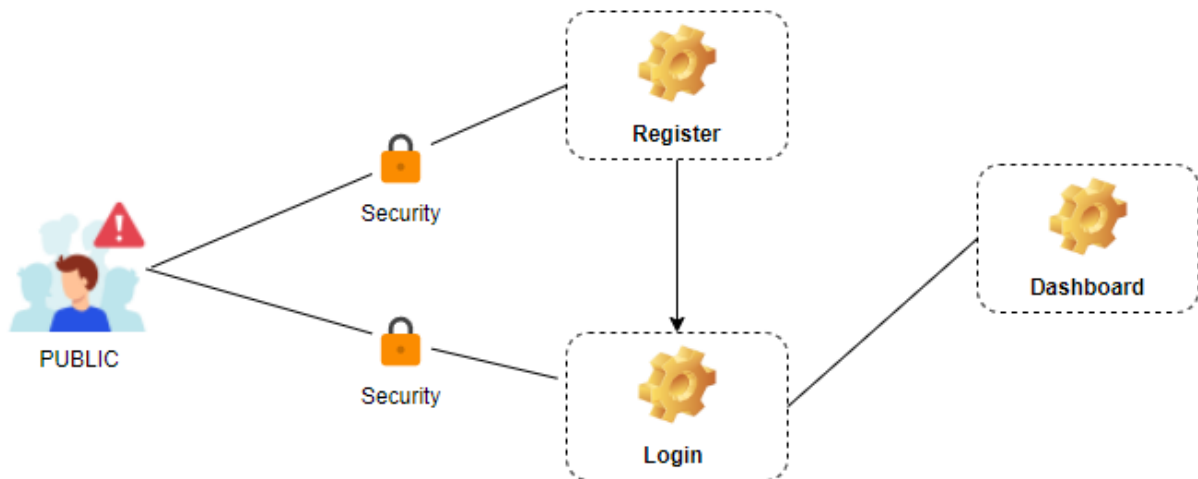


Figure 5: Access and Registration Diagram

4.1.1 User Registration

Table 4: Use Case - User Registration

Use Case	User Registration
<i>Description</i>	A mechanism to join EPICommunity and store personal information to the registry
<i>Actors</i>	PUBLIC
<i>Assumptions</i>	A PUBLIC user has accessed the EPICommunity site and wishes to create a profile in the EPICommunity and add his/her data on the systems registry-catalogue.
<i>Steps</i>	The user clicks on a registration button on the home page. This links to a page, which asks for information about the user. Required fields are: <ul style="list-style-type: none"> • First name • Last name • Email address • Chosen password (password is repeated to check for typing errors)
<i>Variations</i>	-
<i>Non- Functional</i>	The registration process must be quick - completed within seconds.
<i>Issues</i>	-

4.1.2 User Login

Table 5: Use Case - User Login

Use Case	User Login
<i>Description</i>	A mechanism to authenticate EPICommunity registered members in order to access the platform
<i>Actors</i>	Member / Subscriber
<i>Assumptions</i>	A registered user has accessed the EPICommunity site and wishes to login in his/her EPICommunity profile.
<i>Steps</i>	The user clicks on a login button on the home page. This links to a page, where the user must fill a form with his credentials. Required form fields are: <ul style="list-style-type: none"> • Email address

	<ul style="list-style-type: none"> • Password <p>After filling the correct information in the form, by pressing the Login In button, the user is redirected to the main view of the platform, and he is authenticated.</p>
<i>Variations</i>	-
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.2 Entities interaction

The Figure 6 depicts the Members' interactions with the EPICommunity platform entities while the Figure 7 shows the flow of how a Subscriber can Search and Visit profiles. The interactions are divided into three categories and be further analyzed in the respective paragraphs:

- User to user (Members, partially Subscribers)
- Recommended entities to the user by the platform (only Members)
- User interaction with platform entities (only Members)

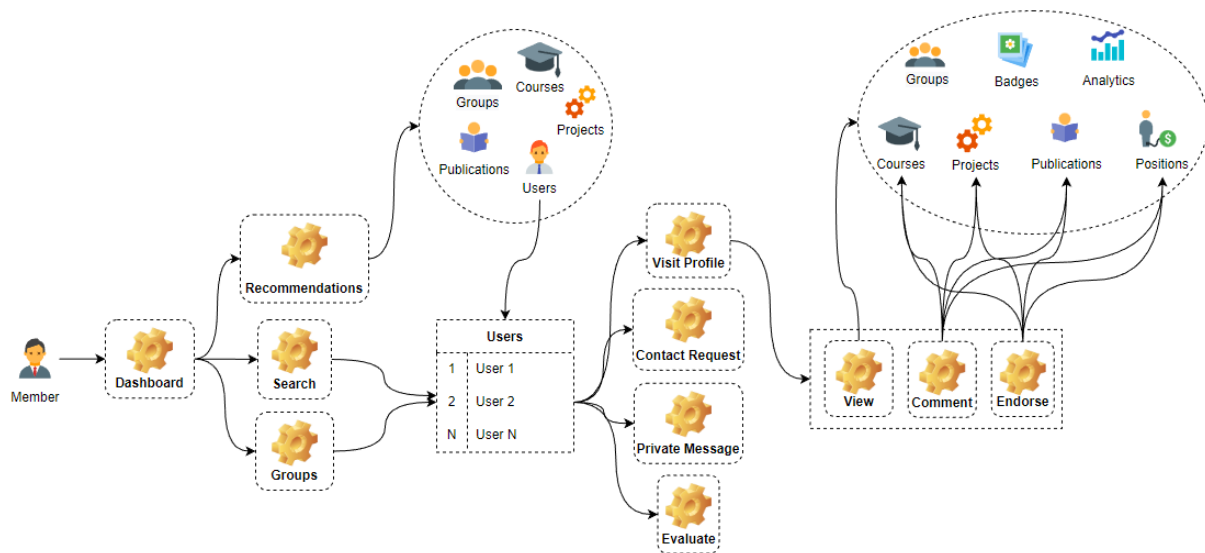


Figure 6: Members' Entities Interactions Diagram

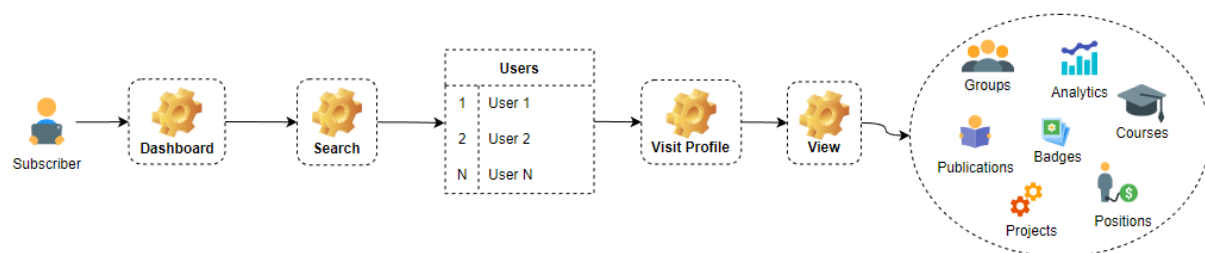


Figure 7: Subscribers' Search and Visit Profile Diagram

4.2.1 User to user

The users can interact with each other with the following ways:

- Search Profile
- Visit Profile
- Contact Request
- Private Message
- Evaluate User

4.2.1.1 Search Profile

Table 6: Use Case - Search Profile

Use Case	Entities Interaction / User to User / Search Profile
<i>Description</i>	This use case describes the ways that a user can utilise in order to find another EPICommunity Member.
<i>Actors</i>	Member, Subscriber
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform
<i>Steps</i>	<p>EPICommunity platform empowers the Members and Subscribers with a set of tools and different ways that grants them the ability to search for other Members within the platform. Those are:</p> <p>Search: The EPICommunity platform offers an advanced search mechanism that allows a user to perform complex queries in order find profiles that match his/her requirements. In order for a Member or Subscriber to use the Search, s/he could:</p> <ul style="list-style-type: none"> • Select the Search option from the sidebar menu • Select the entity option Users • Fill the corresponding fields • Send the query • View the results <p>Some of the available search options regarding the profile search are the following:</p> <ul style="list-style-type: none"> • First name (result contains the characters of the user input) • Last Name (result contains the characters of the user input) • Associated Tags • Publication name (result contains the characters of the user input) • Number of Publications • Project name (result contains the characters of the user input) • Number of Projects • Course name (result contains the characters of the user input) • Number of Courses • Number of Positions • Number of Friends • Number of assessments that the profile has received • Number of assessments made • Profile score • Assessment scores per role (Researcher, Innovation, Entrepreneur, Society) • Number of groups that s/he is member of • Top categories tags <p>Search bar: Using a search bar the user can find perform a quick search for another member that match his search query. The search bar input will be matched only against first names, last names and tags while the search results will only contain users.</p> <p>Groups: Utilizing the open groups, a user can find other EPICommunity users by reviewing the members that belong to a group.</p>
<i>Variations</i>	<p>An alternative way to search for a profile using the groups, is to view member of the groups that he/she is member of. By following:</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select Groups tab

	<ul style="list-style-type: none"> • In the new window, the user views a list containing all the groups that he is member • Selecting a group, the user can view the rest group members <p>Also, using the build in recommendations mechanism of the platform the user can find other member that the application suggests for contact.</p> <ul style="list-style-type: none"> • Choose Recommendations in the side bar • Choosing Connection from the options will display recommended Members for contact by the platform • Selecting a group, the user can view the rest group members <p>Note: The above variations are only available for the registered Members, while the Subscribers have neither access to recommendations nor groups.</p>
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.2.1.2 Visit Profile

Table 7: Use Case - Visit Profile

Use Case	Entities Interaction / User to user / Visit Profile
<i>Description</i>	A user can visit the profile of another user and view the information regarding the entities of that profile
<i>Actors</i>	Member / Subscriber
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform
<i>Steps</i>	<p>The steps to view another user's profile are the following:</p> <ul style="list-style-type: none"> • Find the EPICommunity member that the user is interested to view • Press on his/her name <p>After pressing on the user's name, the platform redirects to the profile page of that user. The information that a user will see by visiting another profile is the following:</p> <ul style="list-style-type: none"> • Number of friends • Number of assessments that the profile has received • Number of assessments that the user has made to other users • Profile score • Number of groups that s/he is member of • Top categories • Category impact • Profile strength • List of the last (e.g., 5) assessments • List of the most recent activities regarding the platform entities <p>Using a navigation bar, a user is also able to see the entities information of that profile. This includes the following info:</p> <ul style="list-style-type: none"> • Projects • Publications • Courses • Positions <p>The user can also select a specific instance of an entity in order to view the instance information and useful data like:</p> <ul style="list-style-type: none"> • Instance annotations (tags) • Number of views • Number of endorsements

	<ul style="list-style-type: none"> • Comments <p>In the entities' navigation bar, there is also the option to view the acquired Badges for that user. Explained in section 4.6.</p> <p>In order to find an EPICommunity member the user can utilise the ways described in 4.2.1.1</p>
<i>Variations</i>	-
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.2.1.3 Contact Request

Table 8: Use Case - Contact Request

Use Case	Entities Interaction / User to user / Contact Request
<i>Description</i>	A user can send a contact request to another user
<i>Actors</i>	Member
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform
<i>Steps</i>	<p>The steps that an authenticated user must follow in order to send a contact request to another user are:</p> <ul style="list-style-type: none"> • Visit the profile of the user (explained in 4.2.1.2) • Press the send contact button <p>Then, the platform forwards the request to the other user. When that user accepts the request, they both can view each other in the contacts list.</p>
<i>Variations</i>	In case of a rejected contact request, the user can send a new request.
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.2.1.4 Private Message

Table 9: Use Case - Private Message

Use Case	Entities Interaction / User to user / Private Message
<i>Description</i>	A user can use the messaging mechanism of the platform to send a private message to a contacted user
<i>Actors</i>	Member
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform and being in contact (friends) with the user that is going to receive the message
<i>Steps</i>	<p>The steps that an authenticated user must follow in order to send a message to a contact are:</p> <ul style="list-style-type: none"> • Visit the profile of the user (explained in 4.2.1.2) • Press the Send Message button • Fill the message information (subject, body) in a text form • Press "Send button" <p>The Send Message button is only visible if and only if the two users are already in contact (friends).</p>
<i>Variations</i>	<p>An alternate procedure to use the messaging functionality is through the contact list:</p> <ul style="list-style-type: none"> • Select the Profile option from the sidebar menu • Select the Contact List • Find the contact that is going to receive the message • Press the mail icon

	<ul style="list-style-type: none"> • Fill the message information (subject, body) • Press “Send button”
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.2.1.5 Evaluate User

Table 10: Use Case - Evaluate User

Use Case	Entities Interaction / User to user / Evaluate User
<i>Description</i>	A user can evaluate another user based on the EPIQAssess
<i>Actors</i>	User
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform
<i>Steps</i>	<p>The steps that an authenticated user must follow in order to evaluate another user are:</p> <ul style="list-style-type: none"> • Visit the profile of the user (explained in 4.2.1.2) • Press the Evaluate button <p>After pressing on the Evaluate button, the assessment process starts. The assessment has the form of a questionnaire. The rating is from 1 to 10. The user has to provide his rating and comments (qualitative assessment) for the user regarding the specified field</p> <p>The evaluation could be done in 4 aspects of a researcher based on EPIQAssess, namely:</p> <ul style="list-style-type: none"> • Researcher • Teacher • Entrepreneur • Society <p>Each role has 3 categories of criteria:</p> <ul style="list-style-type: none"> • Core criteria • Specific criteria • Personal Qualities <p>Depending on the category the number of questions per aspect varies from 3 to 6 questions.</p> <p>The evaluation will expire after 6 months.</p>
<i>Variations</i>	<p>The evaluation process is flexible. A user should be able to evaluate another one:</p> <ul style="list-style-type: none"> • only in a specific role • save the evaluation process to finish later
<i>Non- Functional</i>	-
<i>Issues</i>	The “Emerging Criteria” of EPIQAssess are not included in the assessment. This could be integrated in a future version of the EPICommunity if decided/needed.

4.2.2 Recommended entities to the user by the platform

Table 11: Use Case - Recommended entities to the user by the platform

Use Case	Entities Interaction / Recommended entities to the user by the platform
<i>Description</i>	The platform proposes entities to a user in the form of recommendations (i.e., other user profiles, publications, courses, groups)
<i>Actors</i>	Member, Platform

<i>Assumptions</i>	A registered Member that is authenticated (logged in) in the EPICommunity platform
<i>Steps</i>	<p>The steps that a user must follow in order to view the recommendations are the following:</p> <ul style="list-style-type: none"> • Choose Recommendations in the side bar • Choosing an entity from the options will display the recommendations of that choice. <p>The different options are:</p> <ul style="list-style-type: none"> • Projects • Groups • Publications • Courses • Connections (users) <p>The user can view the platforms recommendation regarding the entity of his choice. The platform sorts the recommendations in a descending order, from the most relevant to the least relevant.</p> <p>In order to create a recommendations list for a user, the EPICommunity platform considers two parameters. Those parameters are:</p> <ul style="list-style-type: none"> • Tags • Institution <p>The Tags parameter contributes 70% to the result, while the Institution parameter contributes 30%.</p>
<i>Variations</i>	-
<i>Non- Functional</i>	-
<i>Issues</i>	The recommendation algorithm could be revised in future versions by changing the input parameters (for example by considering “Alliance”, “Co-Authors”, etc.) and their weights.

4.2.3 User interaction with platform entities

Table 12: Use Case - User interaction with platform entities

Use Case	Entities Interaction / User interaction with platform entities / Actions
<i>Description</i>	<p>Users can interact with platform entities. The interactions include:</p> <ul style="list-style-type: none"> • Endorsement, which is a form of “like” functionality used in other social networks • Copy entity link, where the user can create a link that leads to that entity • Comment
<i>Actors</i>	Member
<i>Assumptions</i>	A registered member that is authenticated (logged in) in the EPICommunity platform
<i>Steps</i>	<p>To interact with an entity, an authenticated member user has to:</p> <ul style="list-style-type: none"> • Visit the profile of a user (explained in 4.2.1.2) • Visit the entity that he wants to interact with • Choose the type of interaction <p>In case of Endorsement, the user can press the Endorse button. The button changes colour in order to indicate that an entity in already endorsed. If the user wants to remove his/her endorsement, he/she presses again the same button, which will return to its normal state.</p>

	<p>In case of Comment, the user can view the previous comments of that entity and by using the input field and the Comment button can leave his/her own comment. A user can delete a comment if he/she wishes by:</p> <ul style="list-style-type: none"> • using the options button on the top right of a comment he already has posted • Select the delete option <p>The user cannot use the Endorsement for his own entities.</p>
<i>Variations</i>	-
<i>Non- Functional</i>	-
<i>Issues</i>	Consider adding an Edit option in the comment's options.

4.3 Profile Related Use Cases

The Figure 8 depicts the EPICommunity profile related use cases. The use cases are distinct in the following categories and will be thoroughly analyzed in the corresponding paragraphs:

- Profile Management (Create / Read / Update / Delete)
- Profile Import
- Profile Export

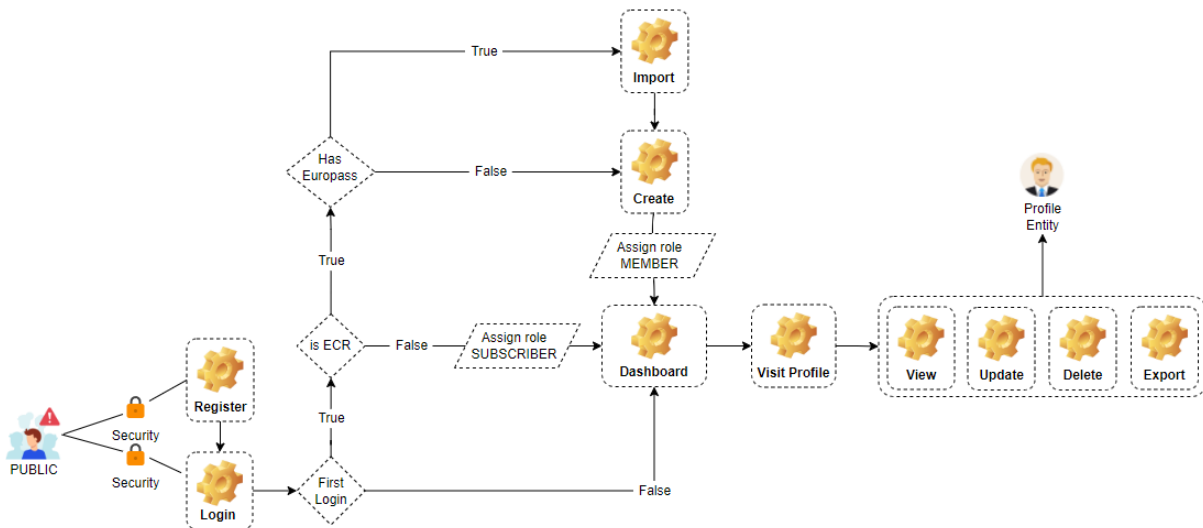


Figure 8: Diagram for the Profile Related Use Cases

4.3.1 Profile Management (Create / Read / Update / Delete)

Table 13: Use Case - Profile Management

Use Case	Profile Related Use Cases / Profile Management
<i>Description</i>	<p>This use case refers to the set of actions that allow a user to manage his/her own profile. The profile management actions are:</p> <ul style="list-style-type: none"> • Create: this management action refers to the fulfilment of the extended information regarding the user. While an email and a password are sufficient for a user to get registered in the EPICommunity, in order to use the platform in its extend and get the most of it, a user must provide additional information. • Read: allows a user to view the information of his/her own profile • Update: allows a user to update the information in his/her profile • Delete: allows a user to remove information from his/her profile or delete his/her account.
<i>Actors</i>	Member
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform

<i>Steps</i>	<p>The profile creation begins right after a successful registration. The new user will be prompt to provide additional information that is crucial for the functionality of the platform.</p> <p><u>Step 1</u> The mandatory information is:</p> <ul style="list-style-type: none"> • Institution / Organisation • Research level (R1-R4, as agreed by the EPICUR consortium) • Profile descriptors (in form of tags) • ORCID • Europass link <p>The following field is optional:</p> <ul style="list-style-type: none"> • Alliance <p><u>Step 2</u> In this next step, the platform requests from the user the following information:</p> <ul style="list-style-type: none"> • Date of birth • Place of birth • Contact Information • Address <p><u>Step 3</u> In the final step, the platform asks the user for information regarding the entities:</p> <ul style="list-style-type: none"> • Courses • Publications • Projects • Positions² <p>And the following lists containing his/her skills:</p> <ul style="list-style-type: none"> • Digital skills • Communication/Interpersonal skills • Organisational skills • Language skills <p>By finishing the above procedure, the role MEMBER is assigned to the user, so he/she can use the full capabilities of the EPICommunity platform and access the Dashboard.</p> <p>The Dashboard is the main view of the EPICommunity platform. Here the user will see useful information not only regarding his/her own profile, but also information regarding his/her friends' recent activity, as well as general platform news.</p> <p>The information regarding his/her own profile in Dashboard contains:</p> <ul style="list-style-type: none"> • Total Endorsements • Profile visits • Total Assessments • Total Profile Score <p>Also, s/he will be able to see:</p> <ul style="list-style-type: none"> • <u>Radar chart</u> displaying his assessment score in the 4 roles (Researcher, Teacher, Entrepreneur, Society) based on other EPICommunity members evaluations • <u>Line chart</u>: displaying his profile visits, endorsements and comments overtime • <u>Friends News</u>: A table displaying his/her friends' recent activity regarding the entities (e.g., X X added a new Publication)
--------------	---

² "Jobs" field in the Europass schema

	<ul style="list-style-type: none"> • <u>EPICommunity News</u>: Displaying information regarding the platform daily news in the form of timeline (e.g., X New Users Joined) <p>The users can read–view the profile information that they have inserted-shared in the EPICommunity. By following the steps below:</p> <ul style="list-style-type: none"> • Select the Profile option in the sidebar menu • Press the Profile Information button <p>The user can view his/her general information:</p> <ul style="list-style-type: none"> • Institution / Organisation • Research level (R1-R4, as agreed by the EPICUR consortium) • Profile descriptors (in form of tags) • Date of birth • Place of birth • Contact Information • Address • Alliance • Digital skills • Communication/Interpersonal skills • Organisational skills • Language skills <p>As well as the following:</p> <ul style="list-style-type: none"> • Friends List • Number of assessments that he/she has received • Number of assessments that he/she has made to other users • List of his assessments • Profile score • Group List • Top categories • Category impact • Profile strength • List of his/her most recent activities regarding the platform entities <p>In order to update the profile information, the user must:</p> <ul style="list-style-type: none"> • Navigate to the Profile Information • Select the Edit option <p>The information that he/she can update is:</p> <ul style="list-style-type: none"> • Institution / Organisation • Research level (R1-R4, as agreed by the EPICUR consortium) • Profile descriptors (in form of tags) • Date of birth • Place of birth • Contact Information • Address • Alliance • Digital skills • Communication/Interpersonal skills • Organisational skills • Language skills • ORCID <p>In order to delete the profile information, the user has to:</p> <ul style="list-style-type: none"> • Navigate to the Profile Information
--	--

	<ul style="list-style-type: none"> • Select the Delete option • Select Yes, in a pop-up window <p>The Delete option removes the following information from the user profile:</p> <ul style="list-style-type: none"> • Date of birth • Place of birth • Contact Information • Address • Hard skills • Soft skills • Alliance <p>Furthermore, in the delete function the member/subscriber will have an option to delete his/her account from the EPICommunity permanently.</p>
<i>Variations</i>	<p>From the three steps described in the profile management action create, only the first one is mandatory.</p> <p>A user can choose to continue to the EPICommunity platform as a SUBSCRIBER in the first step of the action create. By doing so, the whole create action is skipped, the role SUBSCRIBER is assigned to the user, and he/she continues to the EPICommunity platform with limited access. SUBSCRIBER:</p> <ul style="list-style-type: none"> • Has access only EPICommunity news to his/her Dashboard • Has no Recommendation option • Has no ability to interact with entities • Has the tag “Subscriber” in his/her profile • Has no ability to assess Members • Has no ability to create or join groups <p>Also, the available information for update or view by other users in his/her profile is:</p> <ul style="list-style-type: none"> • First name • Last name • Address • Organisation • Email
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.3.2 Profile Import

Table 14: Use Case - Profile Import

Use Case	Profile Related Use Cases / Profile Import
<i>Description</i>	Users can import existing profiles that follow the Europass profile schema. A convenient way to fill profile information regarding the entities Publication, Courses, Projects, Jobs and generic profile information like contact, address, skills and date of birth.
<i>Actors</i>	Member
<i>Assumptions</i>	The user is registered in the EPICommunity platform
<i>Steps</i>	<p>After a new user successfully registers to the EPICommunity platform and finishes the first mandatory step of the profile creation, he/she can choose to finish the profile creation either manually or import. The steps to import existing profile are:</p> <ul style="list-style-type: none"> • Press the import button • Provide a link containing the profile for import • Review the information that are going to be imported per entity

	<ul style="list-style-type: none"> • Press the Complete button.
<i>Variations</i>	The user can use either file or link for the import process.
<i>Non- Functional</i>	-
<i>Issues</i>	<p>Issues for this use case contain:</p> <ul style="list-style-type: none"> • Incompatible schema • Incompatible file extension • Expired link

4.3.3 Profile Export

Table 15: Use Case - Profile Export

Use Case	Profile Related Use Cases / Profile Export
<i>Description</i>	Users can export their profiles into a file that follows the Europass profile schema definition in a convenient way for profile reusability.
<i>Actors</i>	Member
<i>Assumptions</i>	The user is registered and authenticated in the EPICommunity platform
<i>Steps</i>	<p>In order to export the profile information, the authenticated has to:</p> <ul style="list-style-type: none"> • Select the Profile option in the sidebar menu • Press the Profile Information button • Select the export option <p>After selecting this option, a file is downloaded. The user can use this file to import or update his information to any platform that is compatible with the Europass profile schema.</p> <p>The user can select the export file extension between the two available export formats:</p> <ul style="list-style-type: none"> • JSON • XML
<i>Variations</i>	-
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.4 [Use Cases for the Entities Courses, Publications and Projects \(Create / Read / Update / Delete\)](#)

The Figure 9 depicts the use cases for the Management and Search of the entities Courses, Publications and Projects by a user with the role of a Member. It shows the set of actions that allow a user to manage the entities Courses, Publications and Projects as well as the ways he/she can find and view instances of those entities.

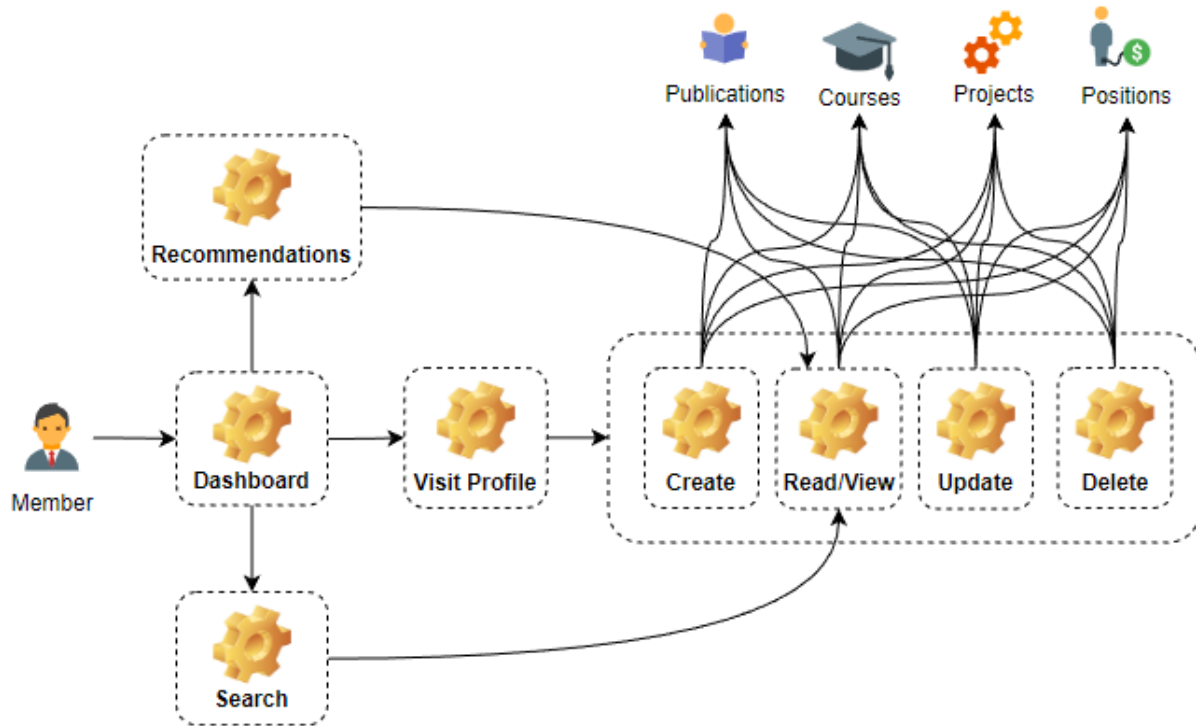


Figure 9: Diagram for Management/Search of Courses, Publications, Positions and Projects by Members

The Figure 10 demonstrates how users with Subscriber role can search for the entities Courses, Publications and Projects.

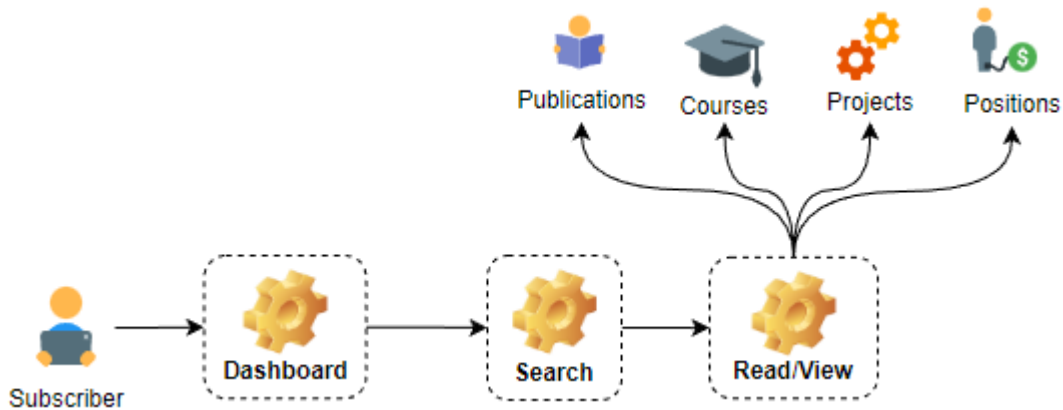


Figure 10: Diagram for Search of Courses, Publications, Positions and Projects by Subscribers

4.4.1 Management of Courses, Publications and Projects

Table 16: Use Case - Management of Courses, Publications and Projects

Use Case	Use Cases for the Entities Courses, Publications and Projects Courses / Management
Description	<p>This use case refers to the set of actions that allow a user to manage his/her own profile Courses, Publications and Projects. The management actions are:</p> <ul style="list-style-type: none"> • Create: allows a user to create new instance of either Course, Publication or Job entity in his/her profile • Read: allows a user to view his/her own records of each entity • Update: allows a user to update the information of an already registered instance

	<ul style="list-style-type: none"> • Delete: allows a user to delete instances of Courses, Publication or Projects from his/her profile
<i>Actors</i>	Member
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform
<i>Steps</i>	<p>In order to create a new instance of an entity, the authenticated user must follow the steps:</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select the category that corresponds to the entity that he/she wants to create • In the new window, select the option to create new entity instance • The new window contains the form for the user to fill with required information regarding new entity • Press create <p>Each entity has its own fields to be filled in the creation form. The fields below are common among all the entities:</p> <ul style="list-style-type: none"> • Title • Description <p>The Publication entity has the additional fields, and they are all mandatory:</p> <ul style="list-style-type: none"> • Year • URL (the link to the publication) • Reference • Co-authors <p>The Course and Project entities have also the fields, with the last two fields being optional:</p> <ul style="list-style-type: none"> • Start date • End date • Ongoing • Link to file or video (optional) • Project logo (optional image attachment of the project, accepted formats: PNG, JPG) <p>In case of Course creation an optional field “Co-Teachers” will be available. In case of Project creation, the user has also the option to create a Group for that Project. By enabling the checkbox labelled Create Group in Project creation form, a new private group will be created and will be attached to this Project using the Project ID field automatically. This new Group will act as an internal community for the project, where the moderator can invite new members.</p> <p>The user can read–view his/her registered entities instances to the EPICommunity platform. The steps are:</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select the option that matches the type of the entity that he/she wants to view • In the new window, the user views all the instances that he/her has created for that entity <p>By selecting an instance, the user will see the detailed information regarding the entity instance as well as:</p> <ul style="list-style-type: none"> • The number of Endorsements • Number of views • Comments • Annotations <p>To edit – update the information of an instance the user has to perform the following actions:</p>

	<ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select the option that matches the type of the entity that he/she wants to update • In the new window, the user views all the instances that he/her has created for that entity • Select the entity instance that is going to be updated • Select the update option • The new window contains a prefilled form with the data of the instance. The user can update the instance data using this form • Press update <p>Deleting an entity follows similar steps as the update procedure.</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select the option that matches with the type of the entity that he/she wants to delete • In the new window, the user views all the instances that he/her has created for that entity • Select the entity instance for deletion • Select the delete option • The pop-up window prompts the user that the delete action is irreversible and ask for his confirmation • Pressing Yes concludes the instance deletion
<i>Variations</i>	-
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.4.2 Search of Courses, Publications and Projects

Table 17: Use Case - Search of the Courses, Publications and Projects

Use Case	Use Cases for the Entities Courses, Publications and Projects / Search
<i>Description</i>	The EPICommunity platform empowers the Members and Subscribers with a tool that allows them to search for Courses, Publications and Projects within the community. This table refers to the search use case of the above entities.
<i>Actors</i>	Member / Subscriber
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform
<i>Steps</i>	<p>Members and Subscribers can utilise the advanced search mechanism that allows them to perform complex queries in order to find Courses, Publications and Projects that meet his/her requirements. To use the Search option:</p> <ul style="list-style-type: none"> • Selecting the Search option from the sidebar menu • Select one or more of the entities options (Courses, Publications, Projects) • Fill the corresponding fields • Send the query • View the results <p>Some of the available search options that are common among the Courses, Publications and Projects are the following:</p> <ul style="list-style-type: none"> • Name (result contains the characters of the user input) • Associated Tags (result contains the characters of the user input) • Number of Endorsements • Number of Comments
<i>Variations</i>	An alternative way to search for Courses, Projects and Publications is by using the build in recommendations mechanism of the platform. A Member can find Courses, Publications and Projects that suggested by the application through:

	<ul style="list-style-type: none"> Choose Recommendations in the side bar Choosing either Courses, Projects or Publications from the options will display the recommended instances of chosen entity <p>The above is only available for the registered Members, while the Subscribers have no access to recommendations.</p>
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.5 Group Related Use Cases

The Figure 11 depicts the diagram for the Group Management (Create / Read / Update / Delete), Group Member Management and Group Enrolment use cases. The actual use cases are further analyzed by the tables in each corresponding paragraph.

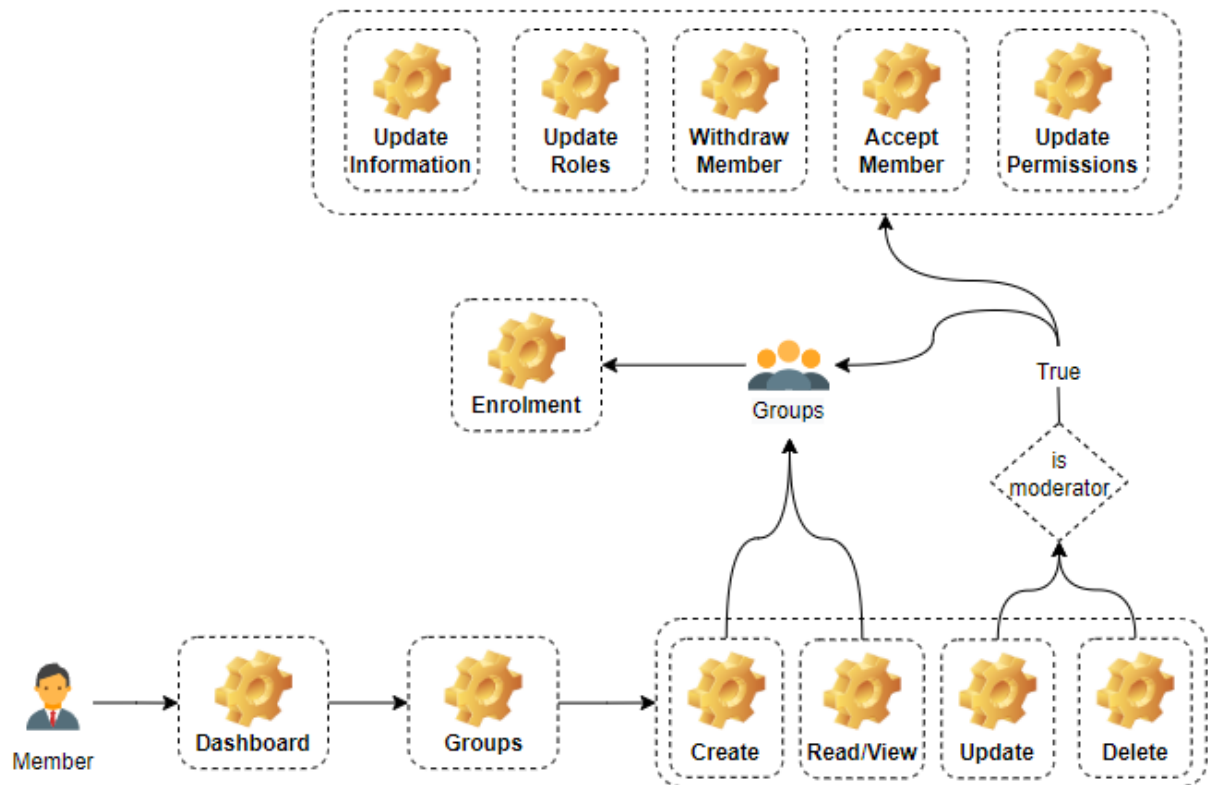


Figure 11: Group Management and Member Management Diagram

4.5.1 Group Management (Create / Read / Update / Delete)

Table 18: Use Case - Group Management

Use Case	Group Related Use Cases / Group Management
<i>Description</i>	<p>This use case refers the built-in group system of the EPICommunity platform. A user can use the following actions in order to manage the entity group:</p> <ul style="list-style-type: none"> Create: allows a user to create a new group Read: allows a user to view the information of a group Update: allows a user to update the information of a group Delete: allows a user to delete a group
<i>Actors</i>	MEMBER, Group Moderator
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform For the update and delete actions the user has to also be group moderator

<i>Steps</i>	<p>In order to create a new group, the authenticated user has to follow the steps:</p> <ul style="list-style-type: none"> • Select Groups option in the sidebar menu • Select the New Group option • Fill in the group information • Press Create <p>The mandatory group information consists of:</p> <ul style="list-style-type: none"> • Group name • Description • Group annotations (tags) • Group type (open/closed) <p>The optional group fields are:</p> <ul style="list-style-type: none"> • Meeting link • Project ID • EPICluster (i.e., to characterize the group as an EPICluster) • EPICradle (i.e., to characterize the group as an EPICradle) <p>The group types are:</p> <ul style="list-style-type: none"> • <u>Open</u>: the open groups do not require membership approval • <u>Closed</u>: the closed groups require for the group moderator to accept the group membership request <p>By completing the creation of the group, the user is assigned automatically as the user founder and the user moderator.</p> <p>A user can read – view the groups that he/she is a member by:</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select Groups tab • In the new window, the user views a list containing all the groups that he is member <p>By clicking on a group, the user can view the groups information like:</p> <ul style="list-style-type: none"> • Group tags • Group name • Description • Group members (if he/she is member or group is public) • Meeting Link (if he is member) • Group wall (if he/she is member) • Group statistics • Manage Group button (if he is group moderator) <p>As well as analytics like:</p> <ul style="list-style-type: none"> • Number of views • Number of Endorsements • Number of comments • Number of badges • List of badges held over time <p>The group wall has the form of a forum.</p> <p>A user can read – view a list of all the registered groups in EPICommunity platform despite the fact if he/she is a member of or not:</p> <ul style="list-style-type: none"> • Selecting the Groups option from the sidebar menu <p>To edit – update the information of group:</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select Groups tab
--------------	--

	<ul style="list-style-type: none"> • In the new window, the user views a list containing all the groups that he is member • Select the group that is going to be updated • Select the Edit option • The new window contains a prefilled form with the information of the group. The user can update the group data using this form • Press update <p>The option Edit in the group view, is only available if and only if the user is moderator of the group or has permissions for edit.</p> <p>Deleting a group follows similar steps as the update procedure.</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select Groups tab • In the new window, the user views a list containing all the groups that he is member • Select the group that is going to be deleted • Select the Delete option • The pop-up window prompts the user that the delete action is irreversible and ask for his confirmation • Pressing Yes concludes the instance deletion
<i>Variations</i>	-
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.5.2 Group Member Management

Table 19: Use Case - Group Member Management

Use Case	Group Member Management
<i>Description</i>	This use case describes how a group moderator can manage the users within a group
<i>Actors</i>	Group moderator
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform The user is group moderator, and the group has at least another one member
<i>Steps</i>	<p>To manage the member of a group the group moderator must:</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select Groups tab • In the new window, the user views a list containing all the groups that he is member • Select the group for user management • Select the Manage Group option • Select Manage Members • The new window contains a table with all the users in the group • Manage the group • Press Complete <p>The option Manage Members, like the Edit option in the group view, is only available if and only if the user is moderator of the group.</p> <p>The bullet “Manage the group” refers to the actions:</p> <ul style="list-style-type: none"> • Remove Member • Change permissions of Member • Change Role

	<p>If the moderator decides that a member can no longer be a part of that group, he/she can revoke memberships and remove users from the group using the Remove Member option. A moderator cannot remove another moderator from the group.</p> <p>The different permissions that a group moderator can give to the members are:</p> <ul style="list-style-type: none"> • Write to wall • Update information <p>The available roles are that a moderator can assign to a group member are:</p> <ul style="list-style-type: none"> • Moderator • Member <p>Another role that exists but it is not assignable is the Founder role. This role is immutable and can only be held by one user, the creator of the group. This role has not any special privileges, it is just an honorary title.</p> <p>The moderator role, is the most powerful role in a group, having all the permissions. A group moderator cannot degrade other moderators. Always must be at least one moderator. In case of multiple, a moderator can only degrade his own role, from Moderator to Member. In case of multiple group moderators, the group cannot be deleted, unless they consent.</p>
<i>Variations</i>	-
<i>Non-Functional</i>	-
<i>Issues</i>	-

4.5.3 Group Enrolment

Table 20: Use Case - Group Enrolment

Use Case	Group Related Use Cases / Group Enrolment
<i>Description</i>	This use case describes how a user becomes member of a group.
<i>Actors</i>	MEMBER, Group Moderator, Platform
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform The group moderator handles the request of the applicant
<i>Steps</i>	<p>There are two types of groups:</p> <ul style="list-style-type: none"> • Open • Closed <p>Which follow different approach to group membership. If the type of the group is open, the applicant automatically becomes member of the group upon application.</p> <p>On the other hand, to become member in a closed group, the applicant sends membership request. The EPICommunity platform pushes a notification to the group moderator and adds a new entry to the list of the pending membership requests of the group. If the moderator accepts the request, the user becomes member of the group. The applicant cannot send a new request while his previous one is still pending. If the group moderator declines the membership request or does not handle it within 7 days, the request is automatically cancelled, and the applicant can reapply.</p> <p>The procedure to apply for a group is:</p> <ul style="list-style-type: none"> • Selecting the Groups option from the sidebar menu • Select a group from the groups list • Press the Apply Now button

	<p>If the group is open, the user becomes a member. In case of closed group, in order the user to become member the group moderator must follow the steps:</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select Groups tab • In the new window, the user views a list containing all the groups that he/she is member • Select the group for user management • Select the Manage Group option • Select the Manage Requests option • The new window contains a table with all the incoming group membership applications • Each entry has 2 options, Accept – Reject • Selecting Accept, the applicant becomes member of the group <p>The default role for new members is Member.</p>
<i>Variations</i>	<p>An alternative procedure to apply for a group is by utilising the search mechanism to find the group first and then join. The steps are:</p> <ul style="list-style-type: none"> • Selecting the Search option from the sidebar menu • Make a Groups query • Select a group from the results • Press the Apply Now button
<i>Non- Functional</i>	-
<i>Issues</i>	-

4.5.4 Leave Group

The Figure 12: Leave Group Diagram depicts the diagram for the Leave Group use case, which is further elaborated in the Table 21 below.



Figure 12: Leave Group Diagram

Table 21: Use Case – Leave Group

Use Case	Group Related Use Cases / Leave Group
<i>Description</i>	This use case describes how a user can leave a group.
<i>Actors</i>	MEMBER
<i>Assumptions</i>	A registered user that is authenticated (logged in) in the EPICommunity platform and wishes to quit from a group that he/she is member of.
<i>Steps</i>	<p>In either case of the open or closed group the steps that a user can follow to leave from a group are:</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select Groups tab • In the new window, the user views a list containing all the groups that he/she is member • Select the group • Select the Leave option • Confirm the leave pop-up message
<i>Variations</i>	-
<i>Non- Functional</i>	-

Issues	-
--------	---

4.6 Badges Use Case (Create / Read / Delete)

The Figure 13 demonstrates how a user with the Member role can view his own badges. The procedure of viewing other Members' badges by Members is shown in the Figure 6, while for the Subscribers to view other Members badges can be found in Figure 7.

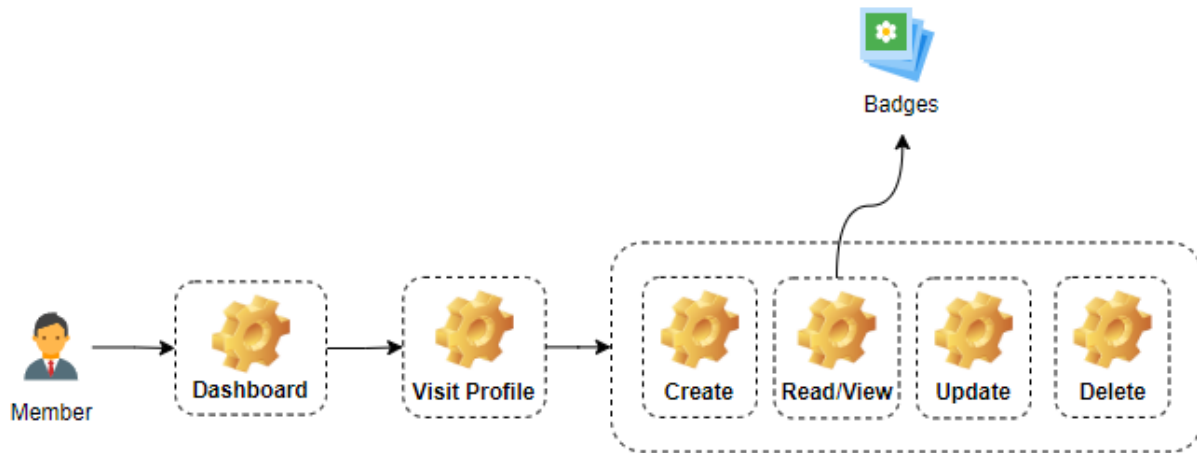


Figure 13: Diagram for View Badges Use Case

Table 22: Use Case - Badges

Use Case	Badges
<i>Description</i>	<p>This use case refers to the actions regarding the built-in badge system of the EPICommunity platform. These actions are:</p> <p>Create: The platform issues a new badge to a user or a group</p> <p>Read: A user can view his badges, other members' badges or groups' badges</p> <p>Delete: The platform decides if the conditions are not met to revoke a badge from a user or a group</p>
<i>Actors</i>	Member, Platform
<i>Assumptions</i>	<p>A registered user that is authenticated (logged in) in the EPICommunity platform.</p> <p>User that has at least one badge.</p> <p>Group that has at least one badge.</p>
<i>Steps</i>	<p>The badges that are considered are those with type Badge of Achievements, from the deliverable of EPIGame (Table 4 - Proposed Gamification Mechanics).</p> <p>For the platform to create a new badge, the conditions of the badge must be satisfied³. After the badge creation the badge is issued to the eligible user.</p> <p>Only the EPICommunity platform can create and issue badges.</p> <p>Users can view – read their earned badges through:</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select Badges tab • In the new window, the user views a list containing all the badges that he/she has acquired <p>Or they can view what badges other members of the EPICommunity have received by:</p> <ul style="list-style-type: none"> • Visit the profile • From the navigation bar select Badges option

³ Please refer to the deliverable D2.3 (EPIGame) for more information regarding the conditions of gaining the badges.

	<p>Also, users can view the Badges that a Group has earned by:</p> <p>If he/she is member of the group</p> <ul style="list-style-type: none"> • Navigate to his/her own profile using the sidebar menu • Select Groups tab • In the new window, the user views a list containing all the groups that he is member <p>If he/she is not member of the group:</p> <ul style="list-style-type: none"> • Selecting the Groups option from the sidebar menu • A list of all the registered groups in EPICommunity platform is displayed <p>In either case, by clicking on a specific group, the user can view the information of a particular group as well as the badges that this group has obtained.</p> <p>The deletion case is also a platform only action. Users cannot delete badges. If the conditions of a badge cease to be met, the platform revokes the issuance of that badge and will not be able to see it on his profile Badge list (same rules apply for group badges).</p>
<i>Variations</i>	-
<i>Non- Functional</i>	-
<i>Issues</i>	-

5 EPICommunity Functionality

This section describes the services that the system provides in terms of responsibilities. Each envisaged system service is described along with a wireframe in order to depict the user interface. The functionality of the system has been categorized as much as possible to each system component/element.

5.1 [Access and Registration / Login](#)

5.1.1 *User Registration*

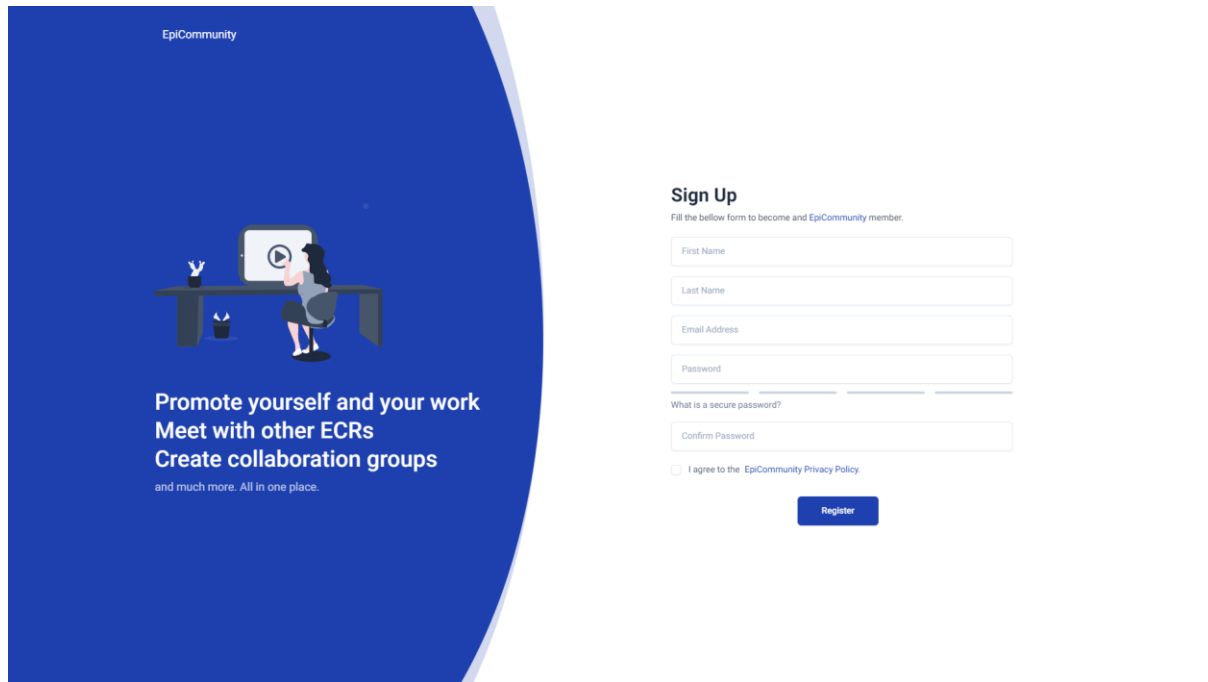


Figure 14: User Registration Screen

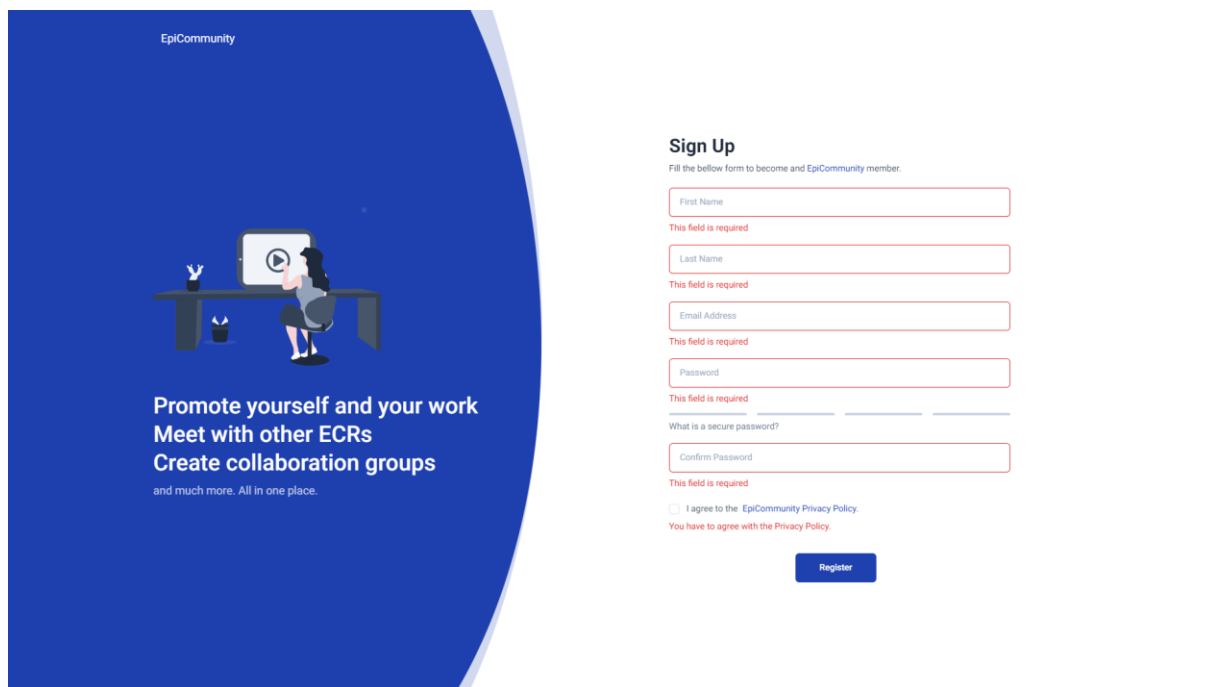


Figure 15: User Registration Screen with errors

Table 23: Functionality – User Registration

Functionality Name	User Registration
Relative Use Cases	User Registration Paragraphs: 4.1.1
Actors	PUBLIC
Input Data	<ul style="list-style-type: none"> First name

	<ul style="list-style-type: none"> • Last name • Email address • Chosen password • Confirm password (repeated password to check for typing errors)
<i>Description</i>	<p>It is assumed that the actor is an anonymous user who wishes to register on the system. Therefore, the actor fills in the form with his/her information as described above in the Input data.</p> <p>Providing the wrong or inaccurate information, the interface (Figure 14) informs the user about the error/s below and each corresponding field. Also, the border of the field turns red, indicating error. The image Figure 15 depicts how a registration form with errors will look like.</p> <p>The incorrect information could be:</p> <ul style="list-style-type: none"> • First name (at least 2 characters) • Last name (at least 2 characters) • Email address (already used, emails are unique in the EPICommunity platform) • Password (must contain at least 1 upper case, 1 lower case, 1 number and have minimum length of 8) • Confirm password (does not match the Password field) <p>All the above fields are mandatory. After providing his/her information and pressing the register button, the user is redirected to a new page for the email verification process (Paragraph 5.1.2).</p>
<i>Output Data</i>	<p>The system creates a new entry in the users table and marks it with unverified email. Also, a verification email is being send to the user.</p>
<i>Variations</i>	-

5.1.2 Email Verification

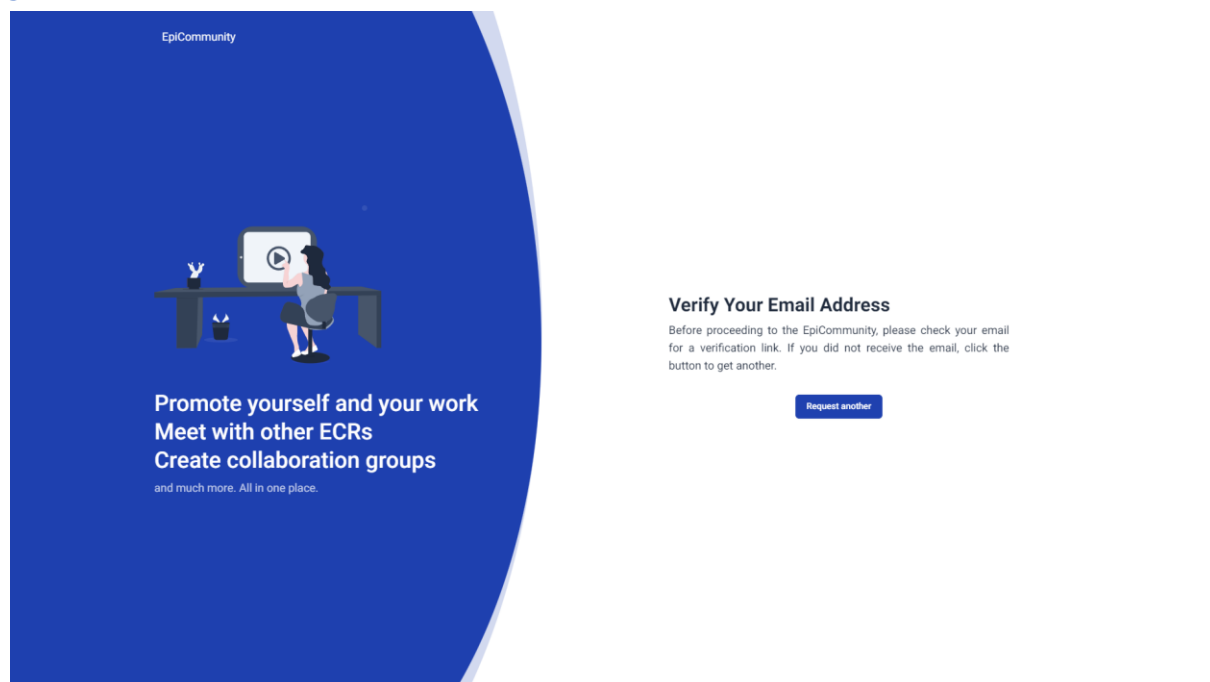


Figure 16: Email Verification Screen

Table 24: Functionality – Email Verification

Functionality Name	Email Verification
Relative Use Cases	User Registration, User Login Paragraphs: 4.1.1, 4.1.2
<i>Actors</i>	PUBLIC
<i>Input Data</i>	<ul style="list-style-type: none"> • Notification ID • Hash (SHA1 of the user email) • Expiration date and time
<i>Description</i>	<p>It is assumed that the actor is a user that has finished the registration process (paragraph 5.1.1) and wishes to log in to his/her account. After a successful log in, the system will redirect the user to a page that instructs her/him to verify the provided email address in order to continue.</p> <p>The steps that a user must follow to verify the email address for the EPICommunity, are:</p> <ul style="list-style-type: none"> • Find the email verification in the inbox • Press the “Verify now” button or visit the link under that button <p>The input data described above, are parameters encoded in the URL that is being send to the users’ email. The user does not provide, and/or not required to know/remember/hold the input information.</p> <p>The email verification is a onetime procedure, and it is mandatory for the user to continue to the usage of the platform. The user must finish the email verification process either during his/her registration or later during an another session right after his/her log in.</p>
<i>Output Data</i>	The system updates the data of the user in the database by setting the email verification date field and redirects the user to for the profile creation procedure.
<i>Variations</i>	

5.1.3 Resend Email Verification

Table 25: Functionality – Resend Email Verification

Functionality Name	Resend Email Verification
Relative Use Cases	User Registration, User Login Paragraphs: 4.1.1, 4.1.2
<i>Actors</i>	PUBLIC
<i>Input Data</i>	<ul style="list-style-type: none"> • Notification ID • Hash (SHA1 of the user email) • Expiration date and time
<i>Description</i>	<p>The verification link that the EPICommunity platform send the users’ email upon his/her registration (described in paragraph 5.1.2) expires after 60 minutes. If the link expires, the user can request a new one, with the rate limited to one verification link per hour.</p> <p>The steps that a user must follow to request a new verification link are:</p> <ul style="list-style-type: none"> • Press the “Resend Verification Link” button <p>As depicted in the Figure 16.</p> <p>After that, she/he must follow the email verification process (paragraph 5.1.2).</p>

<i>Output Data</i>	The system updates the database with the new verification link and its expiration date, for the user that has requested it.
<i>Variations</i>	

5.1.4 User Login

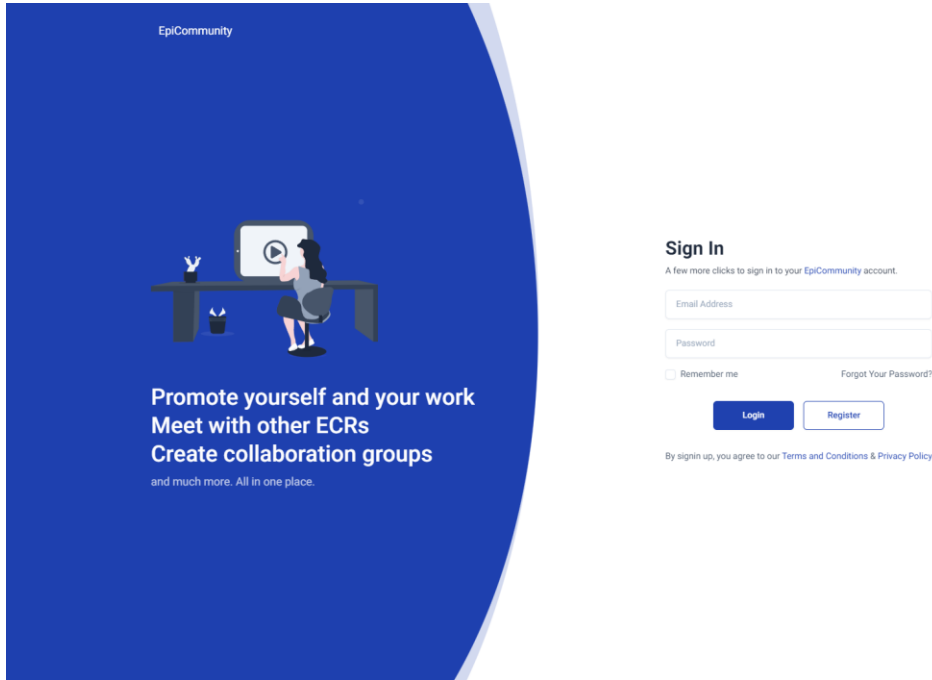


Figure 17: User Login Screen

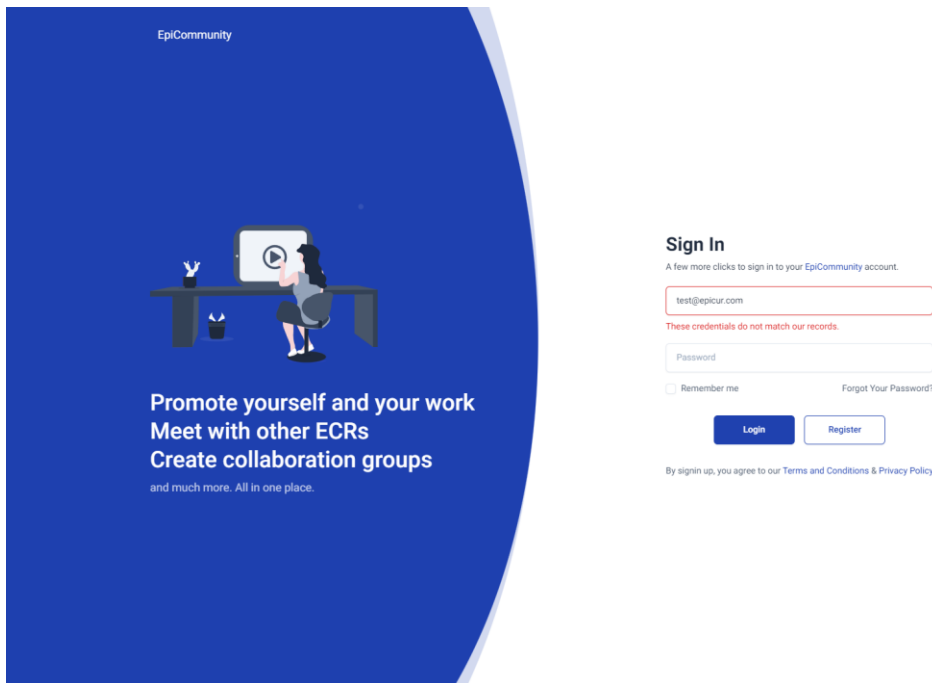


Figure 18: User Login Screen with errors

Table 26: Functionality – User Login

Functionality Name	User Login
Relative Use Cases	User Login Paragraphs: 4.1.2
Actors	PUBLIC
Input Data	<ul style="list-style-type: none"> • Email • Password
Description	<p>It is assumed that the actor is a registered user. The user has to fill the form shown in the Figure 17, providing the input data described above.</p> <p>Pressing the Login button, the information is sent to the server for validation. If the credentials does not match the EPICommunity records, the border of fields in the login form turn red, indicating error, and the user is notified regarding the invalid information, as shown in Figure 18.</p> <p>On the other hand, If the provided information is correct, the platform will check if the user has verified email. Being verified, means that the user will be automatically redirected to the Dashboard screen or role selection screen (Figure 21) (depending if this is the fresh account or not), while not being verified, the platform redirects to the verification process screen (paragraph 5.1.2).</p>
Output Data	The system updates the data of the user in the database by setting the last login field to the login datetime.
Variations	-

5.1.5 Password Reset

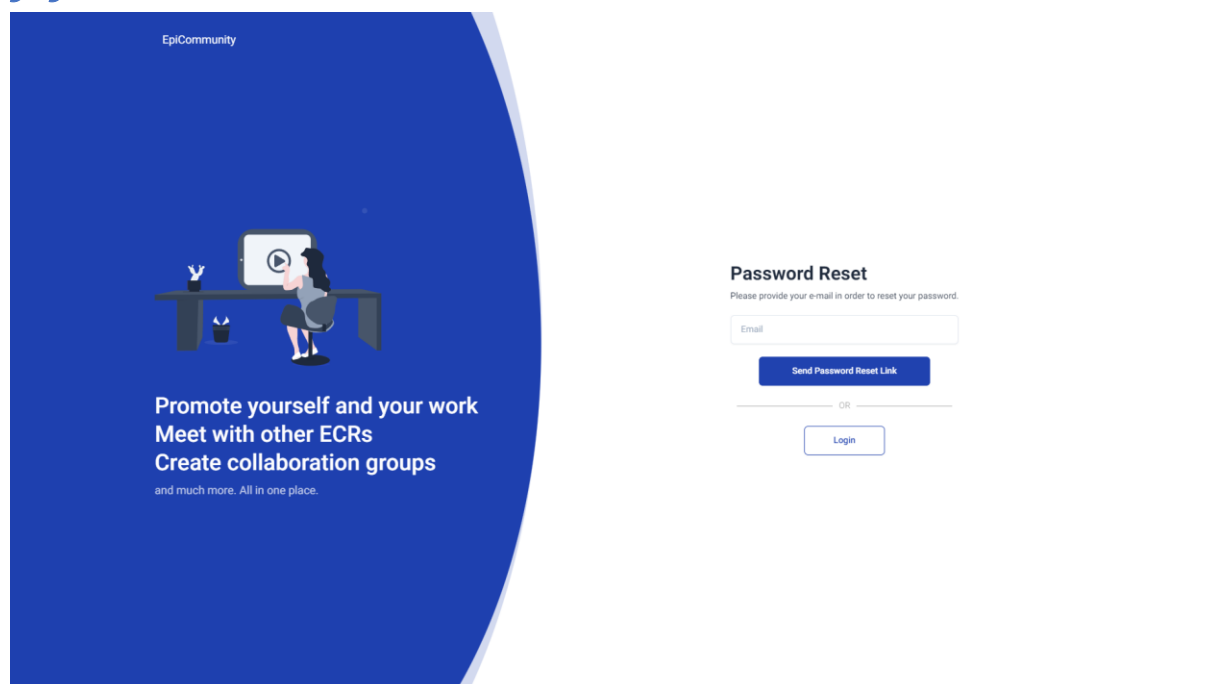


Figure 19: Request Password Reset Link Screen

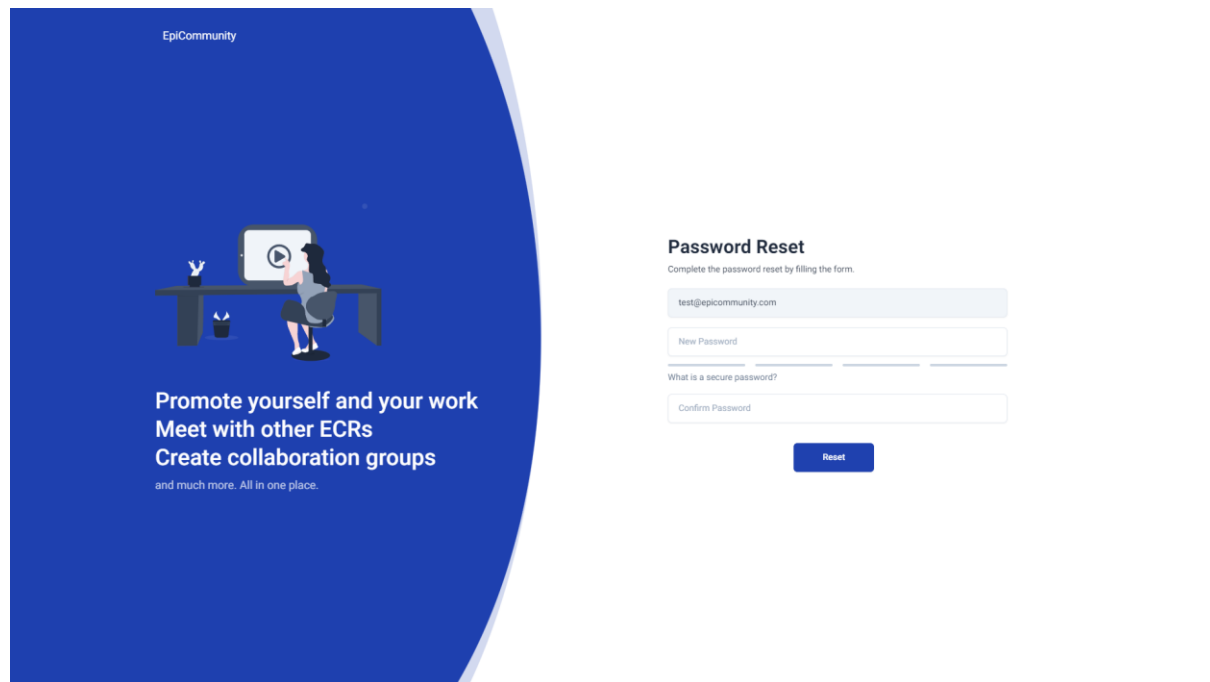


Figure 20: Password Reset Screen

Table 27: Functionality – Password Reset

Functionality Name	Password Reset
Relative Use Cases	User Login Paragraphs: 4.1.2
Actors	PUBLIC
Input Data	<ul style="list-style-type: none"> • Email • Password • Password Confirm
Description	<p>Assuming a user has forgotten the password of his/her account. Using the “Forgot Your Password” option from the Login screen (Figure 17), the user is redirected to the interface as shown in the Figure 19. After filling the email field and pressing the “Send Password Reset Link” button, the EPICommunity platform will check if the email address exists in the records, and if so, will send a mail to that address containing a password reset link.</p> <p>This link redirects the user to the screen Figure 20, where the user must create a new password and provide the password confirm field also. The password field must fulfil all the rules below:</p> <ul style="list-style-type: none"> • At least 1 capital letter • At least 1 number • At least 1 lowercase • Length >= 8 <p>The email field in this screen is not editable and is prefilled with the address that requested the password reset.</p> <p>The password reset link expires after 60 minutes. The user cannot request a password reset link if another one is already active.</p>

<i>Output Data</i>	The system updates the new password of the user in the database and redirects the user to the login page, with the message “Password successfully changed” being displayed.
<i>Variations</i>	-

5.2 Profile Functionality

5.2.1 *Create*

5.2.1.1 *Role Selection*

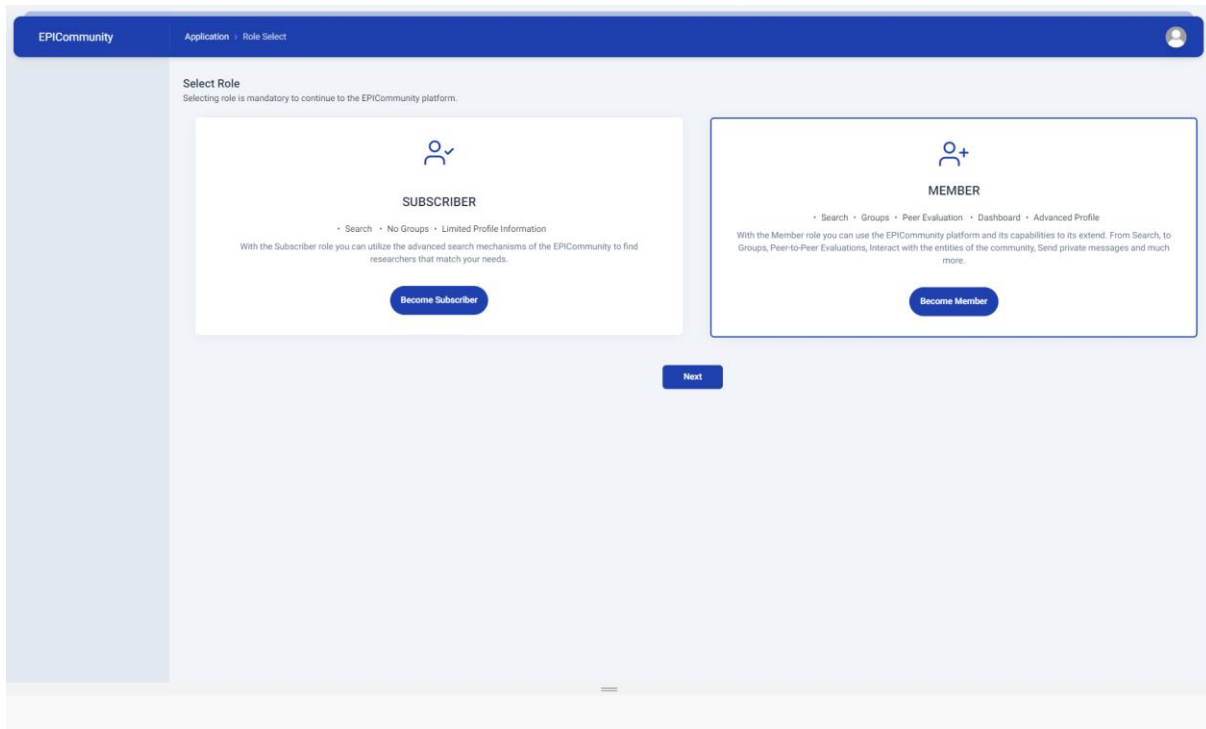


Figure 21: Role Selection Screen

Table 28: Functionality – Role Selection

Functionality Name	Role Selection
<i>Relative Use Cases</i>	User Login, Profile Management (Create) Paragraphs: 4.1.2, 4.3.1
<i>Actors</i>	MEMBER
<i>Input Data</i>	<ul style="list-style-type: none"> • User ID • Role
<i>Description</i>	<p>After the user has created a brand-new account and successfully verified the provided email address, he/she will be prompt to select a role as presented in Figure 21. Refer to paragraph 3.1 for the restrictions and permissions for each role. The interface also informs the user about the rights of each role.</p> <p>The role immutable. If a user wants to use the platform as a different role, he/she has to create a new account with a new email.</p>

<i>Output Data</i>	The system updates the data of the user in the database by setting the role field depending on the choice and based on that choice redirects the user to profile creation.
<i>Variations</i>	-

5.2.1.2 Research Information

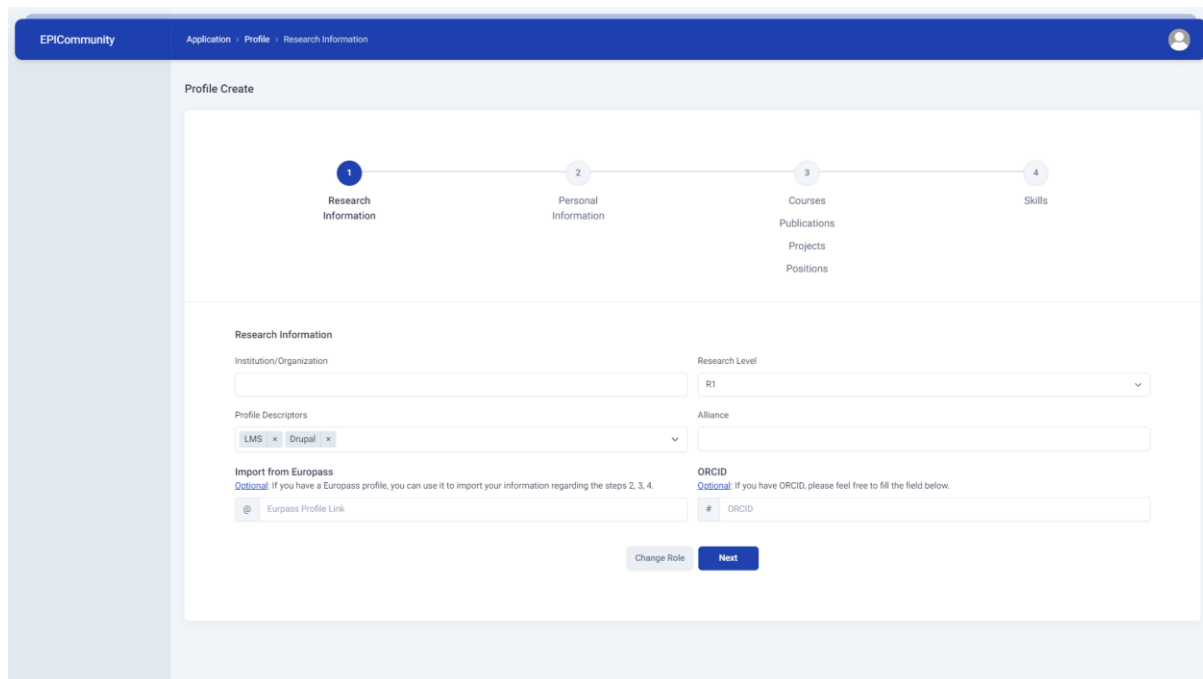


Figure 22: Profile Create – Research Information Screen

Table 29: Functionality – Profile Creation / Research Information

Functionality Name	Research Information
Relative Use Cases	Profile Management (Create) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • Institution / Organisation • Research level (R1-R4, as agreed by the EPICUR consortium) • Profile descriptors (in form of tags) • Alliance (optional) • ORCID (optional) • Europass Profile Link (optional)
Description	<p>If the user selects the Member role (as described in paragraph 5.2.1.1), he/she will be prompt to complete a four-step form providing additional and extended information in order to create his research profile in the EPICommunity. The first step regards the users’ research information as shown in the Figure 22. Refer to the above row regarding the input data.</p> <p>The Europass Profile Link, could be quite useful, in case the user already has a Europass profile. By filling this field, EPICommunity platform will pull the profile information and automatically update the information of the next steps. The user will still be able to update/edit the fetched Europass information.</p>
Output Data	The system updates the corresponding fields of the user in the database and unlocks the next step in the form.

<i>Variations</i>	-
-------------------	---

5.2.1.3 Personal Information

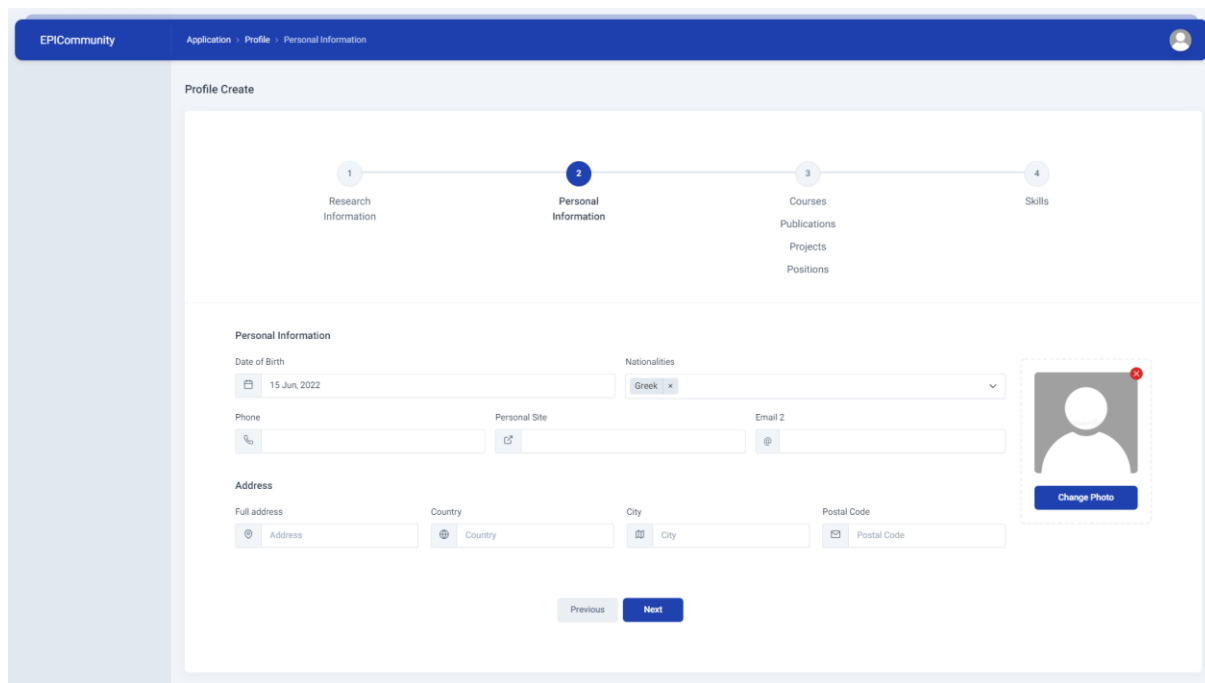


Figure 23: Profile Create – Personal Information Screen

Table 30: Functionality – Profile Creation / Personal Information

Functionality Name	Personal Information
Relative Use Cases	Profile Management (Create) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • Date of birth • Nationality • Phone (optional) • Personal Site (optional) • Email 2 (optional) • Address (all address fields are optional) • Profile photo (optional)
Description	<p>The second step regards the users' research information as shown in the Figure 23. Refer to the above row regarding the input data. Will most of the fields in this step are optional, by filling them, could help other users reach out to you more easily and increase your profile completion score.</p> <p>The information in this step could be completely imported from a Europass profile, by setting the Europass Profile Link field in the previous step.</p>
Output Data	The system updates the corresponding fields of the user in the database and unlocks the next step in the form.
Variations	-

5.2.1.4 Courses, Publications, Positions, Projects

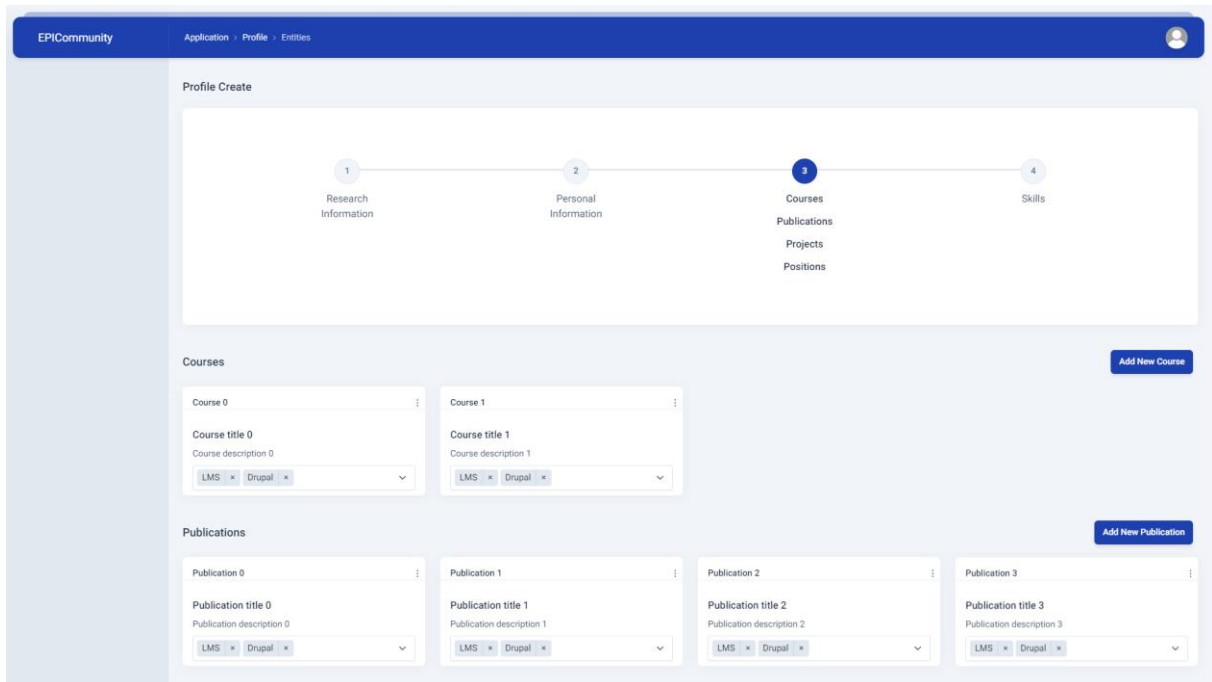


Figure 24: Profile Create – Courses Publications, Projects, Positions Screen

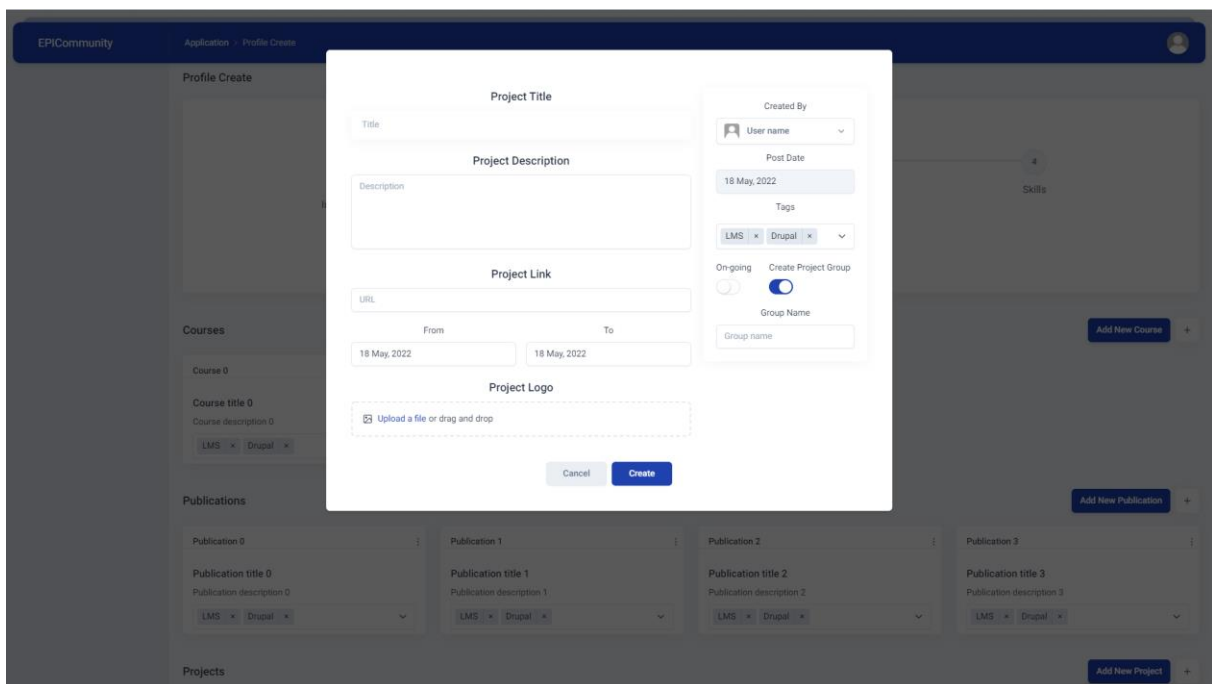


Figure 25: Profile Create – Create Project Modal Screen

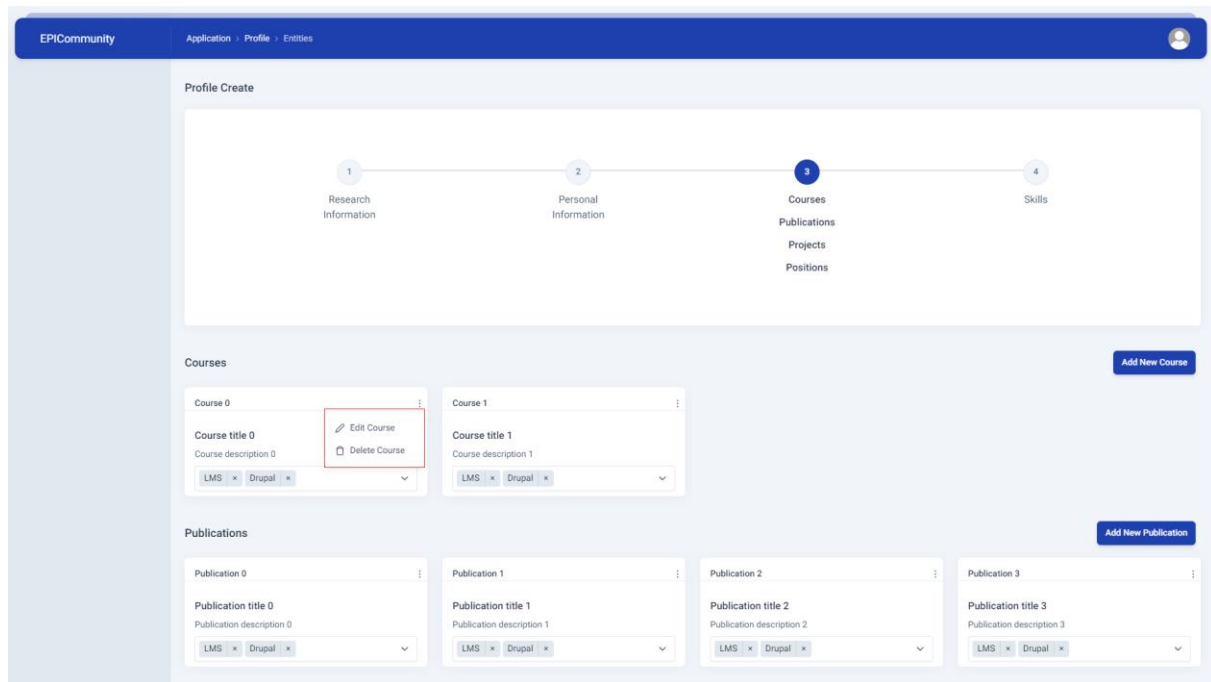


Figure 26: Profile Create – Courses, Publications, Positions , Projects Edit/Delete Options Screen

Table 31: Functionality – Profile Creation / Courses Publications Projects

Functionality Name	Courses Publications Projects
Relative Use Cases	Profile Management (Create) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	<p>Courses fields</p> <ul style="list-style-type: none"> • Title • Description • Host Institution • ECTS • Course Format • Language of instruction • Target Group • Cost • Contact person • Contact email • Learning Outcomes • From (date) • To (date) • Link (optional) • Logo (optional) • Tags • On-going • Co-Teacher(s) <p>Publications fields</p> <ul style="list-style-type: none"> • Title • Description

	<ul style="list-style-type: none"> • Year • Link • Logo (optional) • Tags • Published • Co-authors <p>Projects fields</p> <ul style="list-style-type: none"> • Title • Description • From (date) • To (date) • Link (optional) • Logo (optional) • Tags • On-going • Create Project Group (optional) • Group Name (optional) <p>Positions fields</p> <ul style="list-style-type: none"> • Occupation title • Description • Tags • Employer • City • Country • From (date) • To (date) • On-going • Organization website (optional) • Organization email (optional)
<p><i>Description</i></p>	<p>The third step regards the users' entities, specifically his/her projects, publications, and courses. The user is prompt to add his/her information regarding the mentioned platform entities. The screen Figure 24 depicts the user interface for this step having already added/created some entries. Refer to the above row regarding the corresponding input data for each entity.</p> <p>As shown in the Figure 24, by pressing the "Add New" button across each entity title, a modal window is appeared containing a form with the appropriate fields of the entity. By filling the form and pressing the "Create" button in the modal, a new instance of the chosen entity will be generated. The image Figure 25 shows an example of a form in a modal window regarding the Project entity.</p> <p>Having set the Europass profile link in the first step could be handy, especially for this step. Having the entities information automatically imported could save a lot of time, instead of adding them one by one from scratch. The user can always edit/delete any imported information. Using the 3 dots icons as shown in the Figure 26, the user either edit or delete a specific entity instance. Selecting the edit option will open a modal containing the appropriate form for the entity but being prefilled with the information of the instance, so the user can update the data according to his/her needs.</p>

<i>Output Data</i>	The system updates the corresponding fields of the user in the database and unlocks the next step in the form.
<i>Variations</i>	-

5.2.1.5 Skills

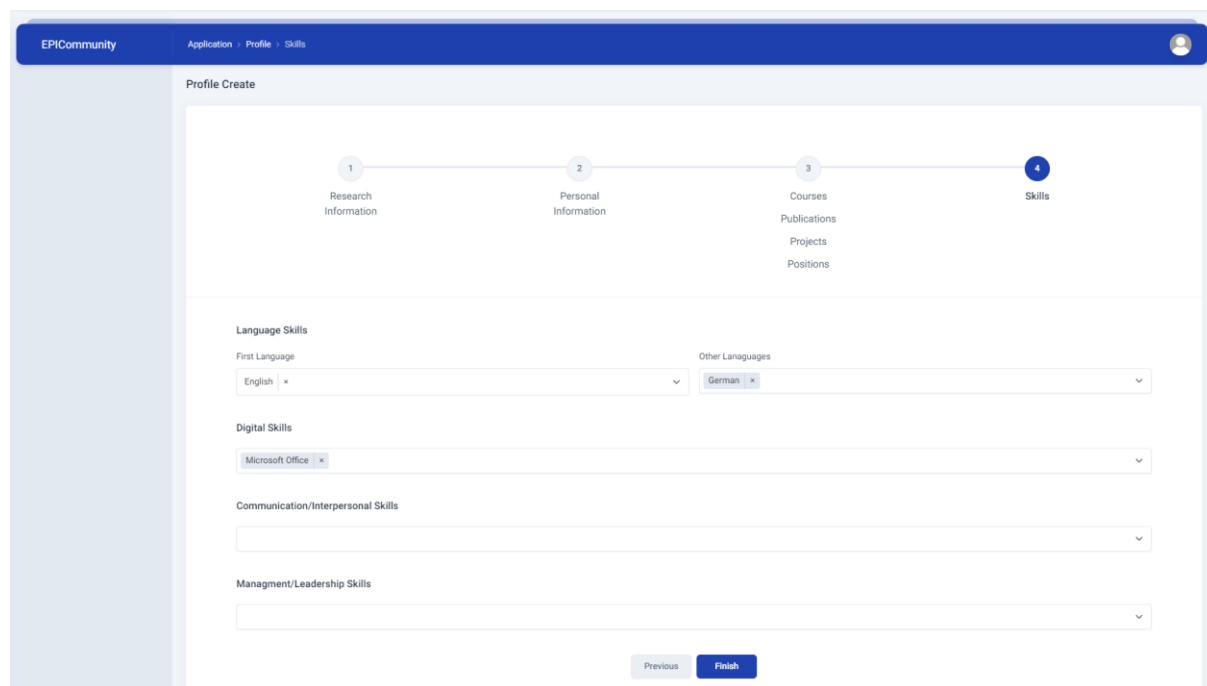


Figure 27: Profile Create – Skills Screen

Table 32: Functionality – Profile Creation / Skills

Functionality Name	Skills
Relative Use Cases	Profile Management (Create) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • First Language • Other Languages • Digital Skills • Communication/Interpersonal Skills • Management/Leadership skills
Description	<p>The final step regards the users' skills. Refer to the previous row for the input data. The Figure 27 depicts the user interface of the fourth step. Every field in this step is a dropdown menu with options. The Mother Tongue field is single select while the rest are multi select. If a skill is not in the dropdown options, the user can type the skill and create a new option on the spot.</p> <p>Skills are also importable from Europass.</p> <p>Pressing "Finish" the profile creation process completes.</p>
Output Data	The system updates the corresponding fields and creates a new profile in the database, associates the user's account with that profile, unlocks the Dashboard view for the user, as well as the side menu options.
Variations	-

5.2.2 Read

5.2.2.1 General Information

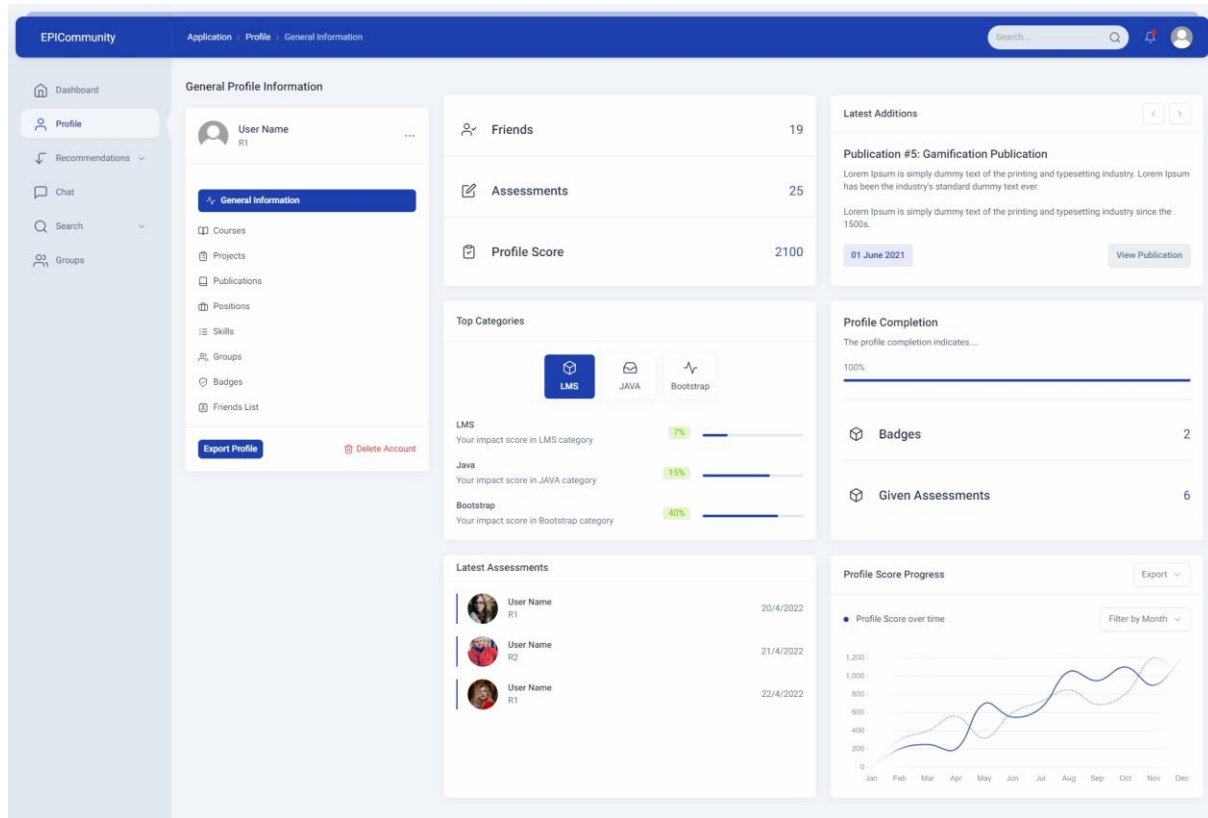


Figure 28: Profile Read / General Information Screen

Table 33: Functionality – Profile Creation / General Information

Functionality Name	General Information
Relative Use Cases	Profile Management (Read) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	
Description	<p>This is the landing page when visiting a profile or a user. The information that someone can view on this page is:</p> <ul style="list-style-type: none"> • Number of friends • Number of Assessments given by the user that owns the profile • Number of Assessments given to that profile by the community members • Profile Score • Top Categories • Profile Completion • Number of Badges • A table with information regarding the last 3 assessments given by the user that owns the profile

	<ul style="list-style-type: none"> • Profile score progression chart • Last 5 entities instances that the user has added <p>The Profile Score is the score of the profile based on the peer assessment functionality provided by the EPICommunity (paragraph 5.4.5). Based on the EPIQAssess, the formula for calculating the score is the following:</p> <p style="text-align: center;"><i>R_x</i> = the relevance number (based on the research level) <i>a</i> = number of assessments <i>c</i> = number of core criteria, <i>A_n</i> the <i>nth</i> core criteria <i>s</i> = number of specific criteria, <i>B_n</i> the <i>nth</i> specific criteria <i>p</i> = number of personal qualities criteria, <i>C_n</i> the <i>nth</i> personal criteria</p> $\text{Core Score} = \frac{\sum_{n=1}^c \left(\frac{\sum_{i=1}^a (A_{n,i} * R_x)}{\text{number of } A_n \text{ ratings}} \right)}{\text{non zero } A_n}$ $\text{Specific Score} = \frac{\sum_{n=1}^s \left(\frac{\sum_{i=1}^a (B_{n,i} * R_x)}{\text{number of } B_n \text{ ratings}} \right)}{\text{non zero } B_n}$ $\text{Personal Score} = \frac{\sum_{n=1}^p \left(\frac{\sum_{i=1}^a (C_{n,i} * R_x)}{\text{number of } C_n \text{ ratings}} \right)}{\text{non zero } C_n}$ <p>Finally, the profile score is:</p> $\text{Profile Score} = w_c * \text{Core Score} + w_s * \text{Specific Score} + w_p * \text{Personal Score}$ <p>The final values of the weights <i>w_c</i>, <i>w_s</i>, <i>w_p</i> will be decided after the platform testing. If:</p> $w_c + w_s + w_p > 1$ <p>then the Profile Score will be divided with the sum of these weights.</p> <p>The Top Categories is a list that shows the most popular tags of the profile. Also, users can see the impact score that the profile has on each tag. The impact score is a weighted sum and it is calculated with the formula:</p> <p>#endorses = the aggregation of endorsements for all the instances in the profile that contain the particular tag</p> <p>#comments = the aggregation of comments for all the instances in the profile that contain the particular tag</p> <p>#views = the aggregation of views for all the instances in the profile that contain the particular tag</p> $\text{Impact Score} = w_1 * \text{\#endorses} + w_2 * \text{\#comments} + w_3 * \text{\#views}$ <p>The final values of the weights <i>w₁</i>, <i>w₂</i>, <i>w₃</i> will be decided after the platform testing. If:</p> $w_1 + w_2 + w_3 > 1$ <p>then the Impact Score will be divided with the sum of these weights.</p> <p>The Profile Completion is an indicator that shows if the user has provided all the required information to the platform and also has added at least 1 instance in each entity of the EPICommunity.</p>
<i>Output Data</i>	-
<i>Variations</i>	-

5.2.2.2 Courses

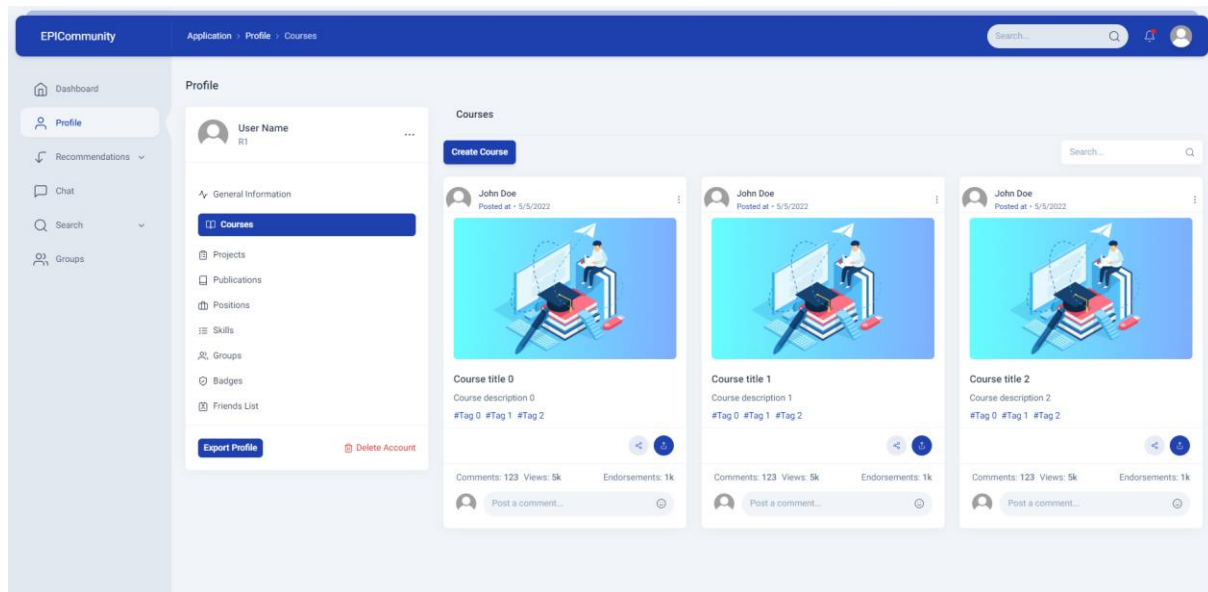


Figure 29: Profile Read / Courses Screen

Table 34: Functionality – Profile Read / Courses

Functionality Name	Courses
Relative Use Cases	Profile Management (Read) / Management of Courses, Publications, Positions and Projects (Courses) Paragraphs: 4.3.1, 4.4.1
Actors	MEMBER
Input Data	-
Description	<p>The screen Figure 29 depicts the list of the users’ courses. Each course is displayed as a “card” view, with the following information:</p> <ul style="list-style-type: none"> • Title • Description • Date posted on EPICommunity platform • Tags • Number of comments • Number of views • Number of endorsements <p>The user can use the share icon button to copy a link for that course, and the endorse button next to it, to endorse the specific course.</p> <p>The option of posting a new comment in a specific course is also provided as shown in the interface, at the bottom of each course view.</p> <p>The “Search” field in the top right corner of the interface can be used to filter the courses list. The field searches based on course title only.</p>
Output Data	In case the user decides to post a comment, the server creates a new comment entry in the database, associates the comment with the course and the user is alerted for the result. More regarding comments on entities in paragraph 5.3.6.
Variations	-

5.2.2.3 Projects

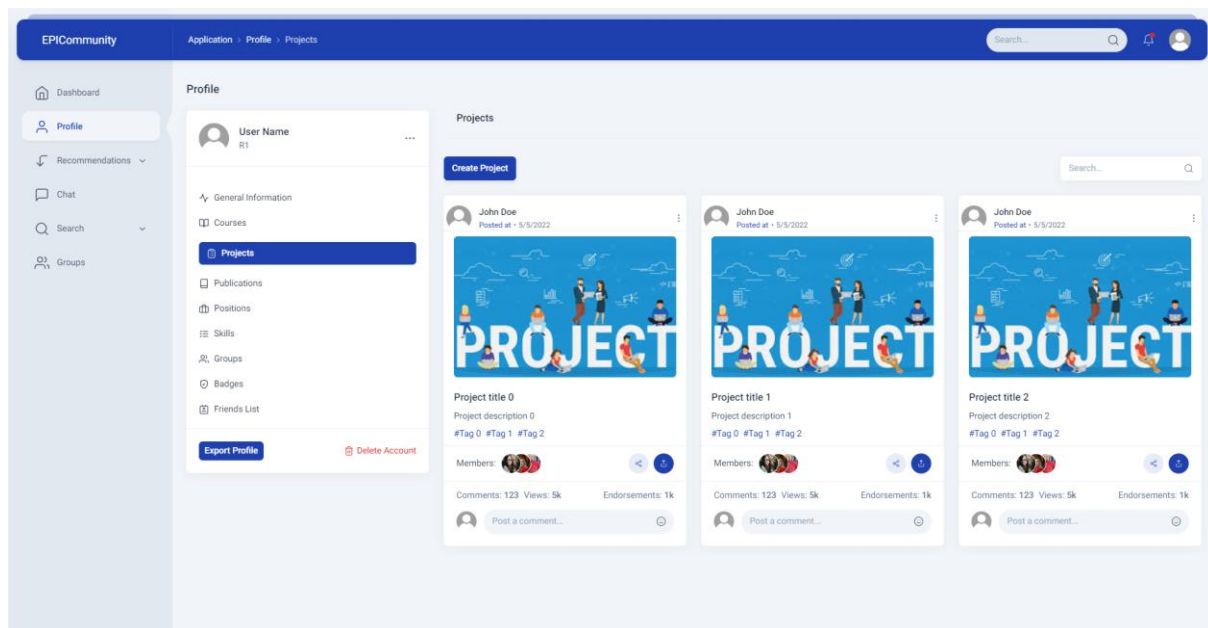


Figure 30: Profile Read / Projects Screen

Table 35: Functionality – Profile Read / Projects

Functionality Name	Projects
Relative Use Cases	Profile Management (Read) / Management of Courses, Publications, Positions and Projects (Projects) Paragraphs: 4.3.1, 4.4.1
Actors	MEMBER
Input Data	-
Description	<p>The screen Figure 30 depicts the list of the users' projects. Each project is displayed as a "card" view, with the following information:</p> <ul style="list-style-type: none"> • Title • Description • Date posted on EPICommunity platform • Tags • Members • Number of comments • Number of views • Number of endorsements <p>The Members information is only available if the project has a group attached to it. The Members show the group members in order descending order based on the member role and how long the user has been a member of that group project. The user can use the share icon button to copy a link for that project, and the endorse button next to it, to endorse the specific project. The option of posting a new comment in a specific project is also provided as shown in the interface, at the bottom of each course view. The "Search" field in the top right corner of the interface can be used to filter the projects list. The field searches based on project title only.</p>

<i>Output Data</i>	In case the user decides to post a comment, the server creates a new comment entry in the database, associates the comment with the project and the user is alerted for the result. More regarding comments on entities in paragraph 5.3.6.
<i>Variations</i>	-

5.2.2.4 Publications

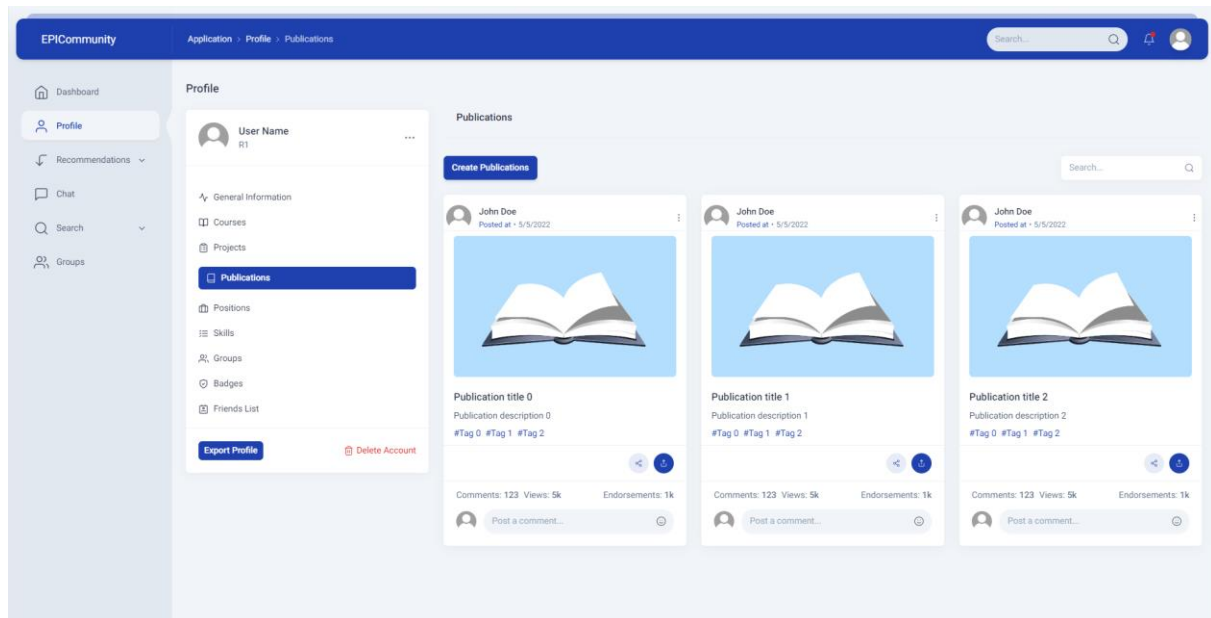


Figure 31: Profile Read / Publications Screen

Table 36: Functionality – Profile Read / Publications

Functionality Name	Publications
Relative Use Cases	Profile Management (Read) / Management of Courses, Publications, Positions and Projects (Publications) Paragraphs: 4.3.1, 4.4.1
Actors	MEMBER
Input Data	-
Description	<p>The screen Figure 31 depicts the list of the users’ publications. Each publication is displayed as a “card” view, with the following information:</p> <ul style="list-style-type: none"> • Title • Description • Date posted on EPICommunity platform • Tags • Number of comments • Number of views • Number of endorsements <p>The user can use the share icon button to copy a link for that publication, and the endorse button next to it, to endorse the specific publication. The option of posting a new comment in a specific publication is also provided as shown in the interface, at the bottom of each publication view.</p>

	The “Search” field in the top right corner of the interface can be used to filter the publications list. The field searches based on publication title only.
<i>Output Data</i>	In case the user decides to post a comment, the server creates a new comment entry in the database, associates the comment with the specific publication and the user is alerted for the result. More regarding comments on entities in paragraph 5.3.6.
<i>Variations</i>	-

5.2.2.5 Positions

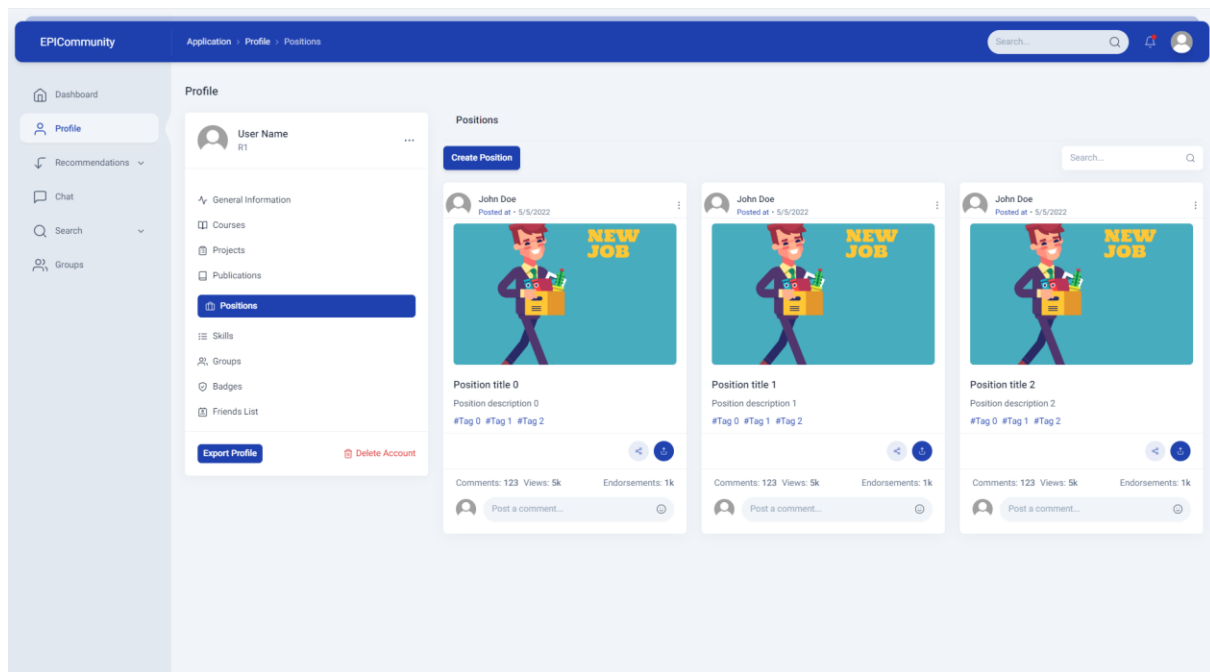


Figure 32: Profile Read / Positions Screen

Table 37: Functionality – Profile Read / Positions

Functionality Name	Positions
<i>Relative Use Cases</i>	Profile Management (Read) / Management of Courses, Publications, Positions and Projects (Positions) Paragraphs: 4.3.1, 4.4.1
<i>Actors</i>	MEMBER
<i>Input Data</i>	-
<i>Description</i>	<p>The screen Figure 32 depicts the list of the users’ job positions. Each position is displayed as a “card” view, with the following information:</p> <ul style="list-style-type: none"> • Title • Description • Date posted on EPICommunity platform • Tags • Number of comments • Number of views • Number of endorsements <p>The user can use the share icon button to copy a link for that position, and the endorse button next to it, to endorse the specific position.</p>

	<p>The option of posting a new comment in a specific position is also provided as shown in the interface, at the bottom of each position view.</p> <p>The “Search” field in the top right corner of the interface can be used to filter the positions list. The field searches based on position title only.</p>
<i>Output Data</i>	<p>In case the user decides to post a comment, the server creates a new comment entry in the database, associates the comment with the specific position and the user is alerted for the result. More regarding comments on entities in paragraph 5.3.6.</p>
<i>Variations</i>	-

5.2.2.6 Skills

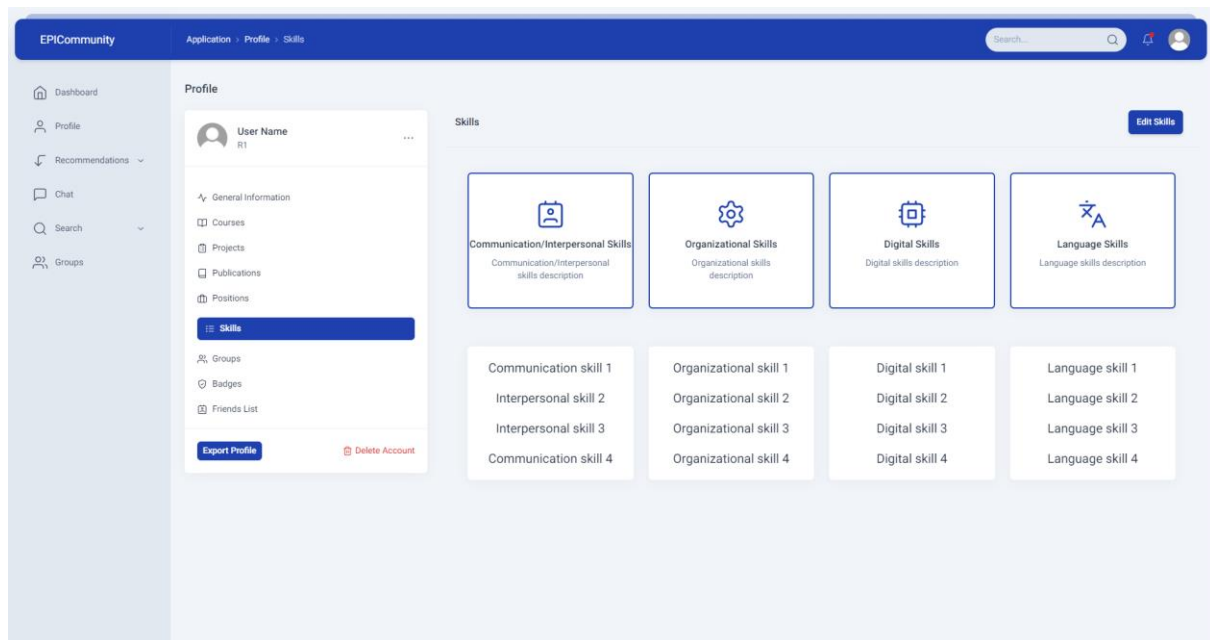


Figure 33: Profile Read / Skills Screen

Table 38: Functionality – Profile Read / Skills

Functionality Name	Skills
Relative Use Cases	Profile Management Paragraphs: 4.3.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • Skill category • Skill
Description	<p>The screen Figure 33 depicts the list of the users’ skills that he/she has provided during the profile creation. The skills are divided into the following categories:</p> <ul style="list-style-type: none"> • Digital skills • Communication/Interpersonal skills • Organisational skills • Language skills <p>If the user owns the profile, the button “Edit Skills” is visible. Pressing that button, allows the user to add new skills or remove existing.</p>

<i>Output Data</i>	<p>If the user decides to add a new skill, the system takes the category and the newly added skill, stores into the database the data and redirects the user back to screen with the list of the skills (Figure 33).</p> <p>If the user decides to discard a skill, the system removes the skill entry from the database and redirects the user back to screen with the list of the skills (Figure 33).</p>
<i>Variations</i>	-

5.2.2.7 Groups

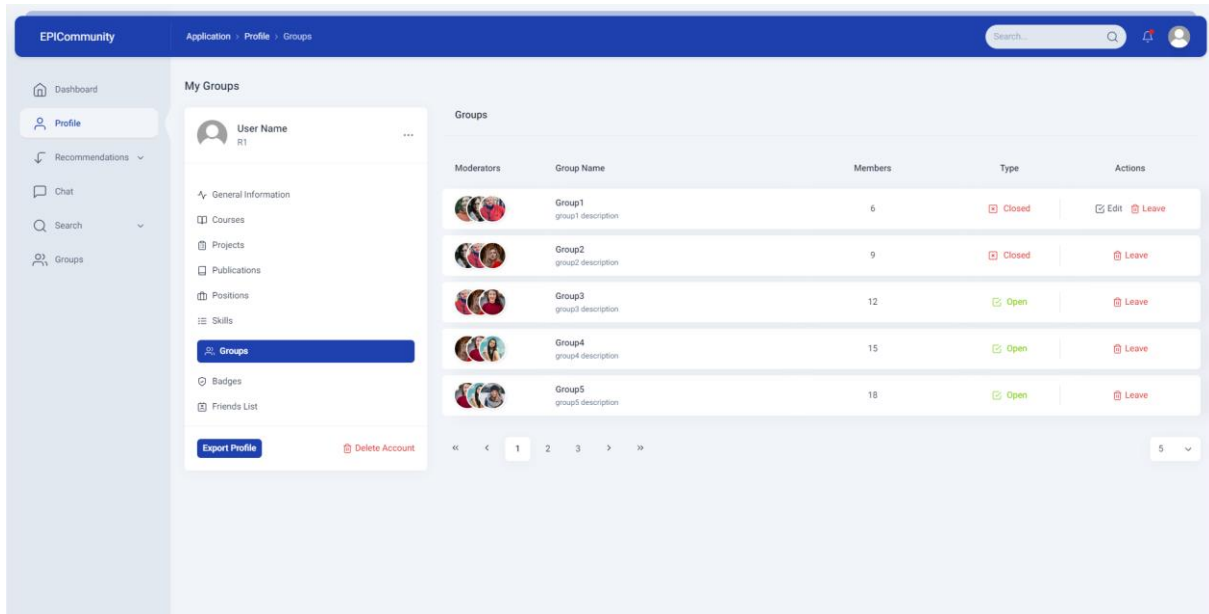


Figure 34: Profile Read / Groups Screen

Table 39: Functionality – Profile Read / Groups

Functionality Name	Groups
<i>Relative Use Cases</i>	Group Management (Read) Paragraphs: 4.5.1
<i>Actors</i>	MEMBER
<i>Input Data</i>	-
<i>Description</i>	<p>The screen Figure 34 depicts the list of the users' groups that he/she is member of, in a table form. Each entry of the table represents an enrolled group, with the following information:</p> <ul style="list-style-type: none"> • Title • Description • Moderators • Tags • Number of members • Type of group (closed, open) • A list of actions <p>The actions list contains the “Leave” option, and in case if the user has the moderator role of the group the “Edit” action is also displayed for that particular group, as indicated in the user interface.</p>

<i>Output Data</i>	-
<i>Variations</i>	-

5.2.2.8 Badges

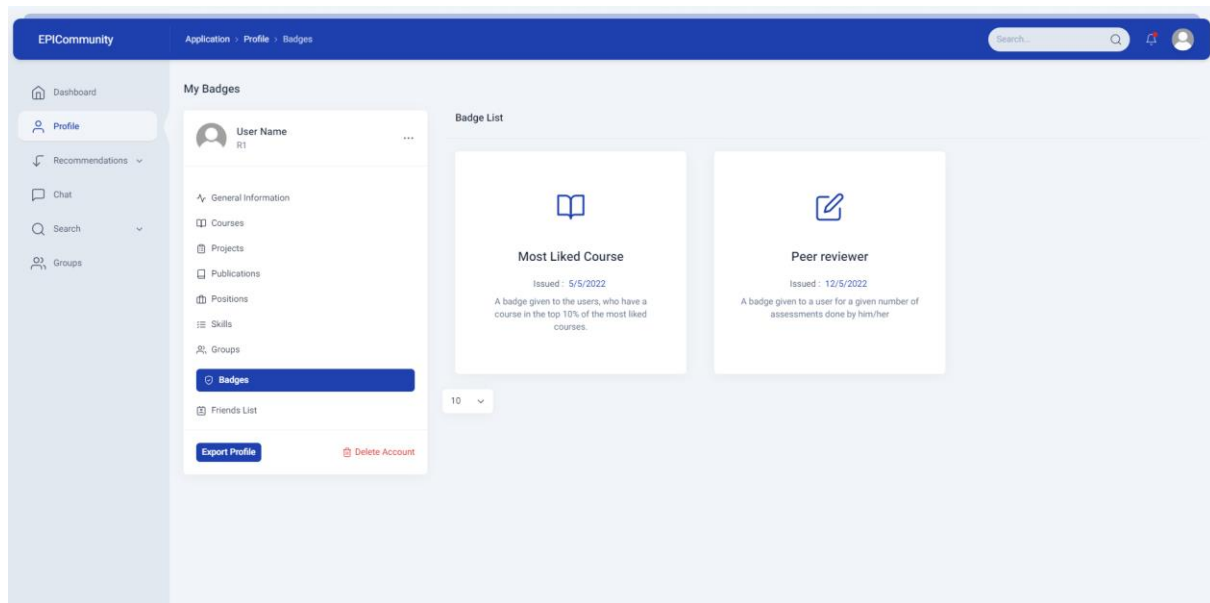


Figure 35: Profile Read / Badges Screen

Table 40: Functionality – Profile Rad / Badges

Functionality Name	Badges
Relative Use Cases	Badges Use Case (Read) Paragraphs: 4.6
Actors	MEMBER
Input Data	-
Description	The screen Figure 35 depicts the list of the users' acquired badges. Each badge is shown in a card view. Each card contains the following information: <ul style="list-style-type: none"> • Icon • Title • Date Issued • Description
Output Data	-
Variations	-

5.2.2.9 Friends List

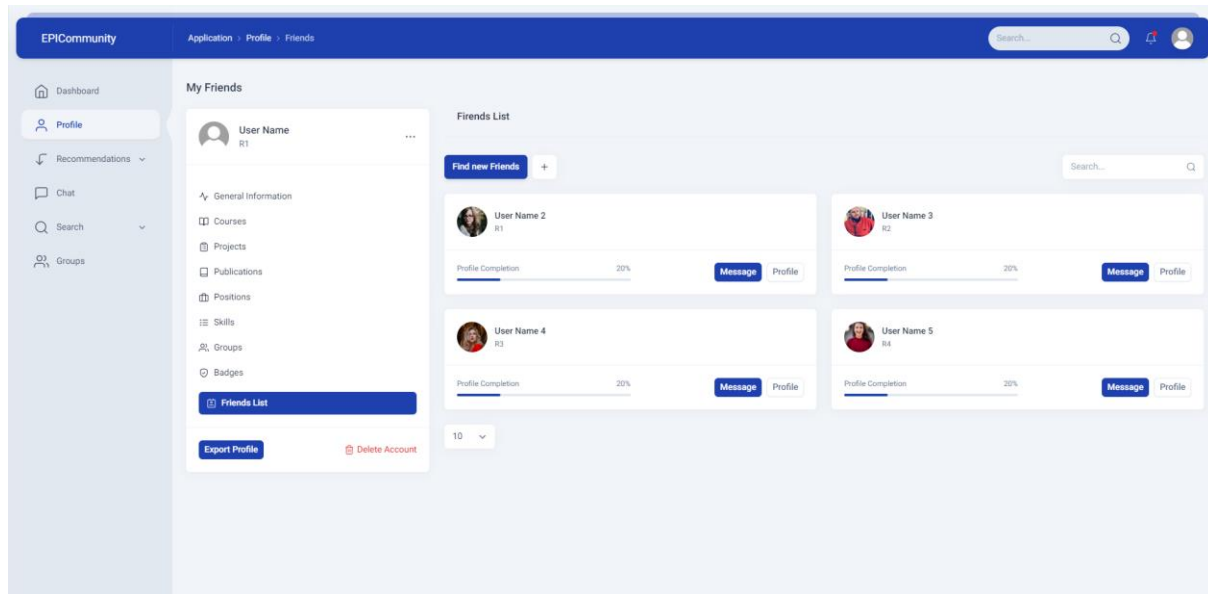


Figure 36: Profile Read / Friends List Screen

Table 41: Functionality – Profile Read / Friends List

Functionality Name	Friends List
Relative Use Cases	Profile Management (Read) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	-
Description	<p>The screen Figure 36 depicts the list of the users' contacts. Each contact is shown in a card view. Each card contains the following information:</p> <ul style="list-style-type: none"> • First and Last name • Research Level • Profile Completion • Message button • Profile button <p>Using the "Profile" button the user can visit the profile of a contact. Using the "Message" button the user opens the history of the messages with a contact.</p> <p>The "Search" field in the top right corner of the interface can be used to filter the user list. The field searches based on first and last name only.</p>
Output Data	-
Variations	-

5.2.2.10 Contact Request

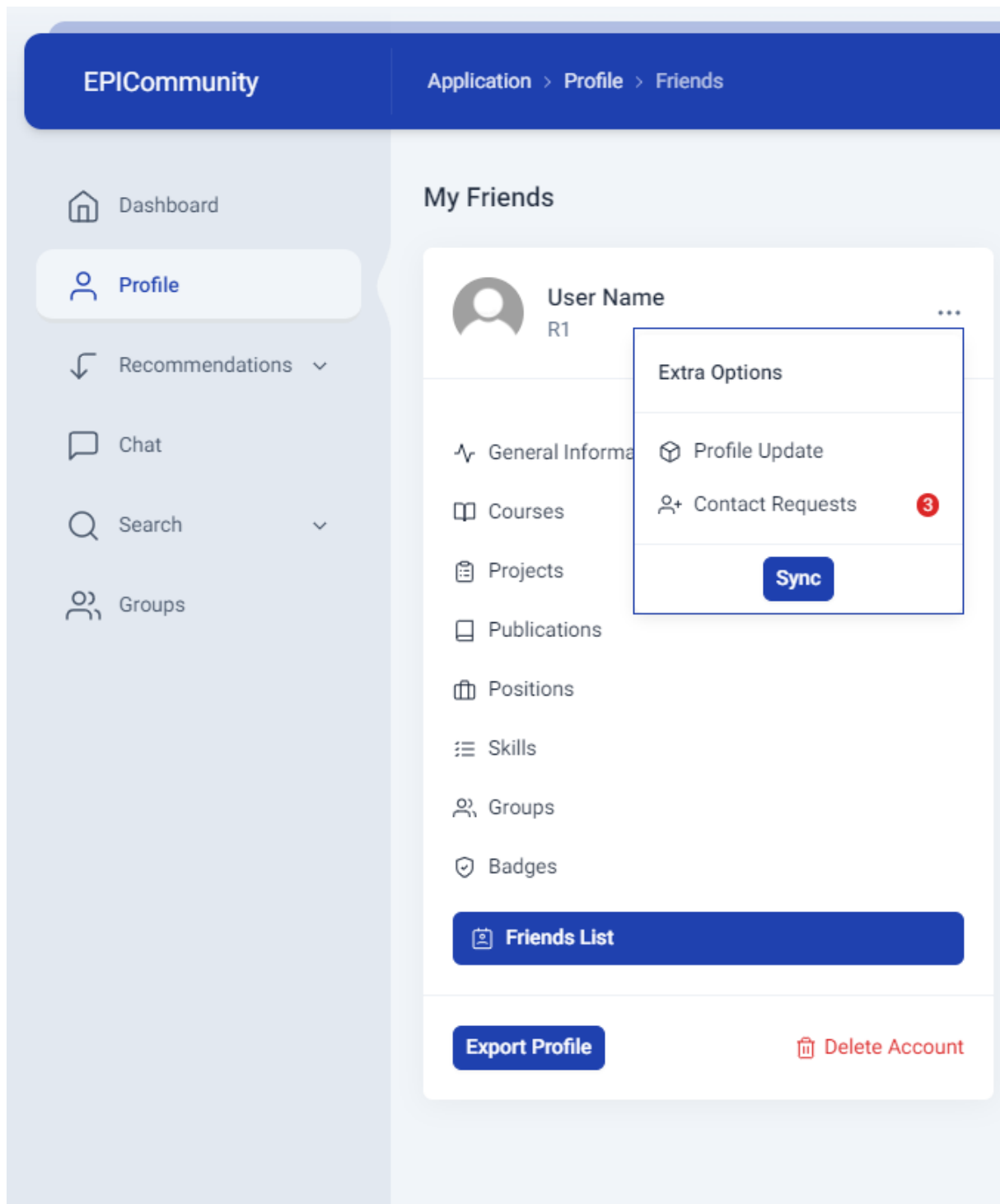


Figure 37: Profile Read / Extra Options menu Screen

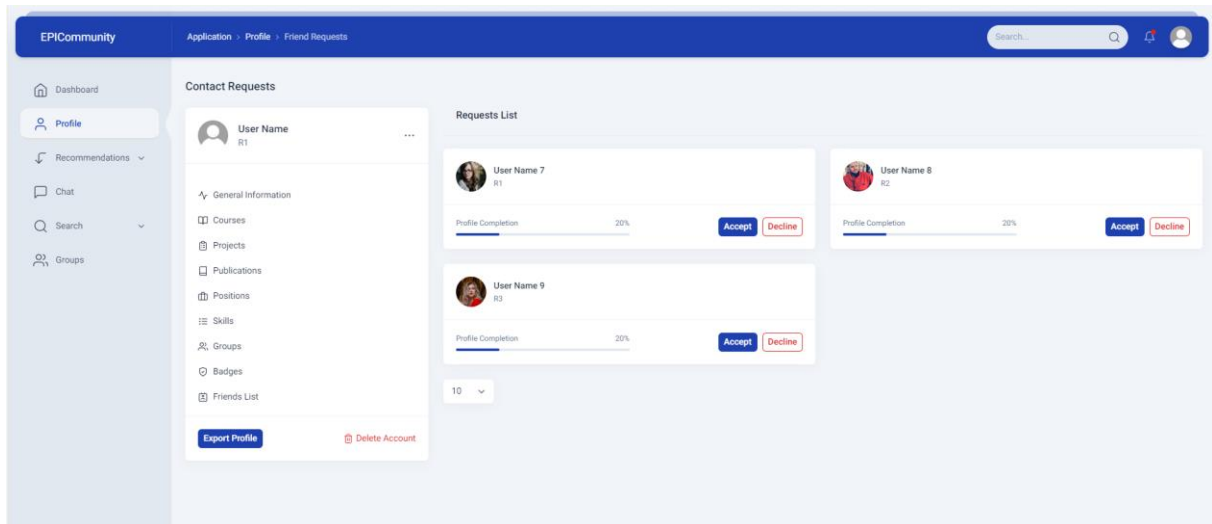


Figure 38: Profile Read / Contact Requests Screen

Table 42: Functionality – Profile Read /Contact Request

Functionality Name	Contact Request
Relative Use Cases	Profile Management (Read) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • User ID (the current user) • User ID (the user that send the request) • Accept/Decline
Description	Across the users' name there are three dots. Those dots, when clicked show a menu, Extra Options. The screen Figure 37 depicts the user interface. The user can choose the option Contact Requests, to view the incoming requests. The screen Figure 38, displays the list view of the requests. The user can accept or decline a request from another EPICCommunity member, by pressing the respective buttons.
Output Data	If a contact request gets accepted, the system creates a new entry in the database, that links the user (with user ID) with the other user (user ID, sender) and unlocked the option of messaging between the two.
Variations	-

5.2.3 Update

5.2.3.1 Research and Personal Information

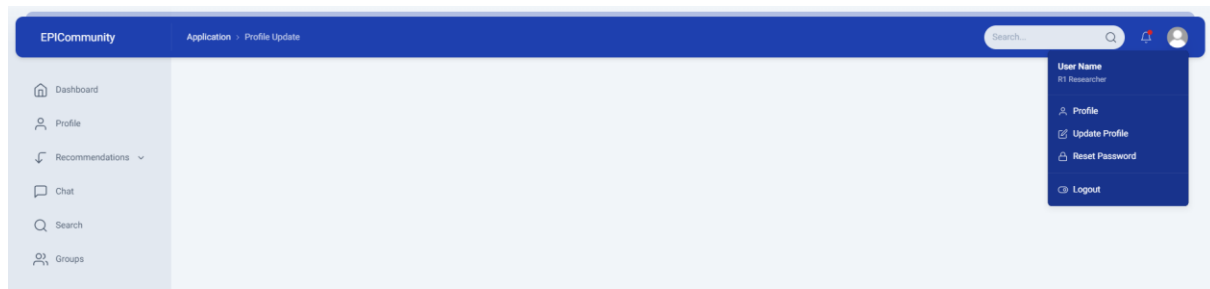


Figure 39: Profile Dropdown Options

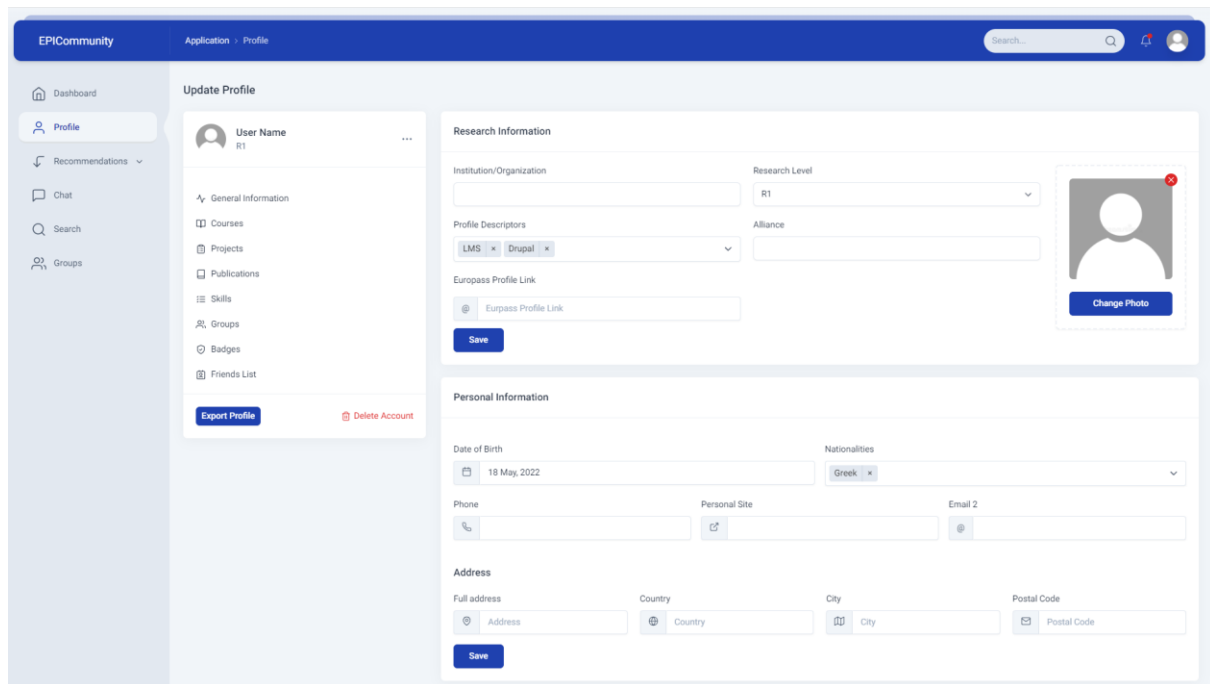


Figure 40: Profile Update Screen

Table 43: Functionality – Profile Update Research and Personal Information

Functionality Name	Profile Update Research and Personal Information
Relative Use Cases	Profile Management (Update) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • Institution / Organisation • Research level (R1-R4, as agreed by the EPICUR consortium) • Profile descriptors • Alliance • Europass Profile Link (optional) • Date of birth • Nationality • Phone • Personal Site • Email 2 • Address (all address fields are optional) • Profile photo

<i>Description</i>	The user can update the information that he/she provided during the profile creation procedure. In order to view the profile update interface, the user has to select the “Update Profile” option in the dropdown menu which appears by clicking in the rounded profile image on the top right corner as shown in the Figure 39. The Figure 40 shows the user interface for in the update functionality. The user can edit the info he/she provided in the first and second step in the profile creation multistep form (paragraphs : 5.2.1.2, 5.2.1.3).
<i>Output Data</i>	The system updates the corresponding fields and creates a new profile in the database, associates the user’s account with that profile, unlocks the Dashboard view for the user, as well as the side menu options.
<i>Variations</i>	-

5.2.4 Delete

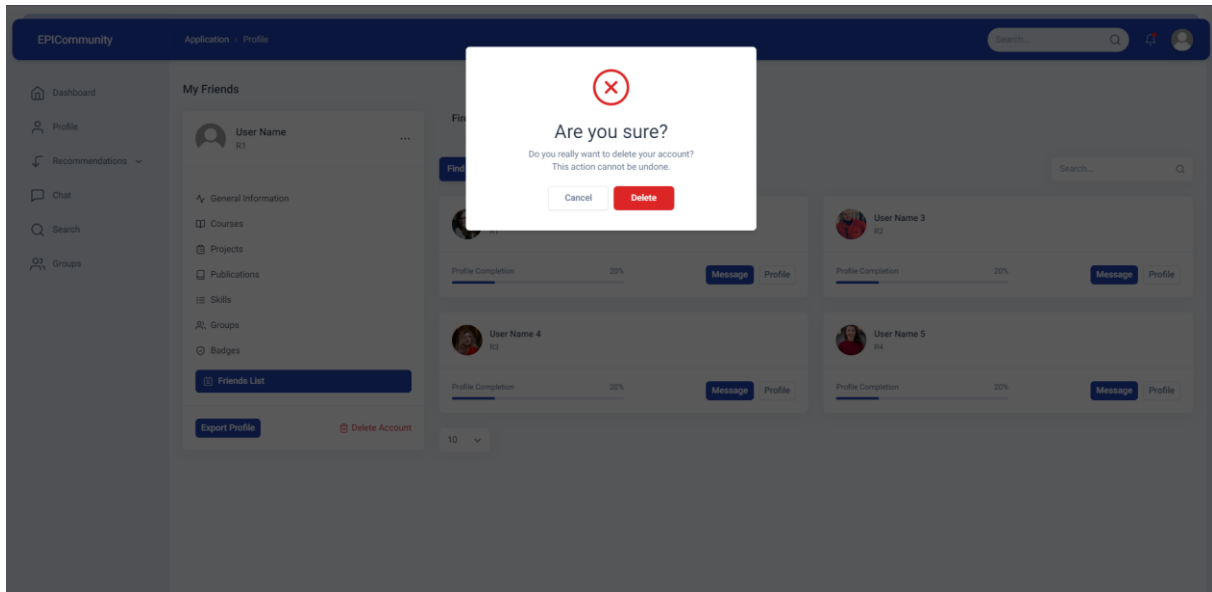


Figure 41: Profile Delete Screen

Table 44: Functionality – Profile Delete

Functionality Name	Profile Delete
<i>Relative Use Cases</i>	Profile Management (Create) Paragraphs: 4.3.1
<i>Actors</i>	MEMBER
<i>Input Data</i>	-
<i>Description</i>	The user can opt to delete his account using the “Delete Account” option in the profile (Figure 28). A popup window with a confirmation message is displayed as shown in the Figure 41. Selecting the “Delete” option in the modal the account with all its information are purged from the EPICommunity platform.
<i>Output Data</i>	The system deletes all the user data, entities information and interactions associated with the user profile, from the database and redirects. Finally the user is redirected to the Landing Page
<i>Variations</i>	-

5.2.5 Export

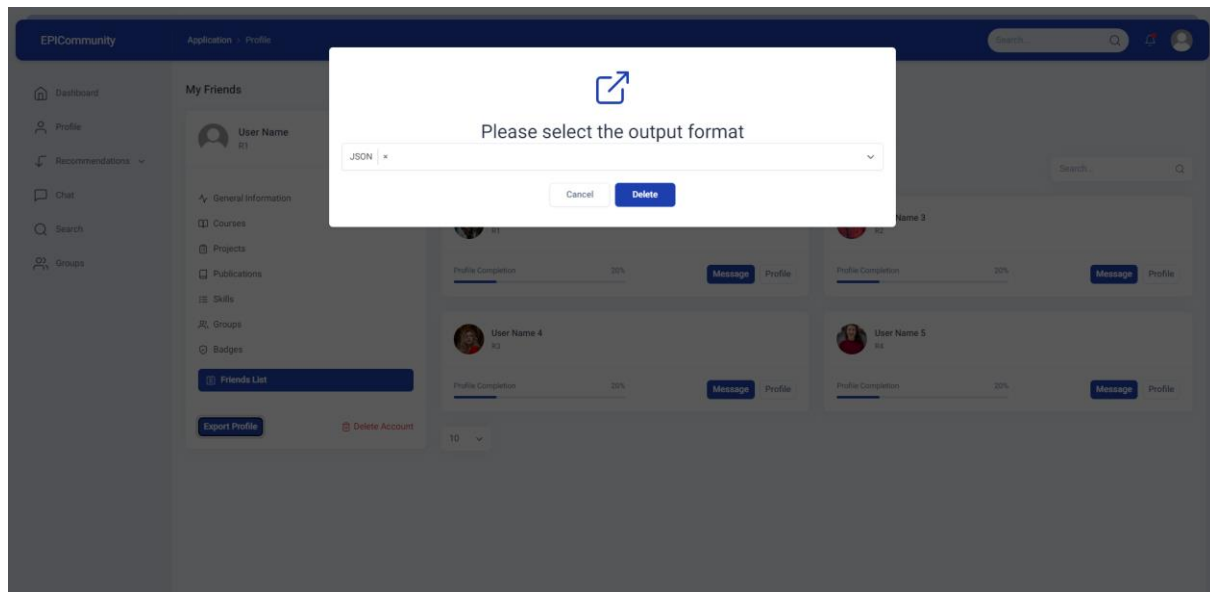


Figure 42: Profile Export Screen

Table 45: Functionality – Profile Export

Functionality Name	Profile Export
Relative Use Cases	Profile Export Paragraphs: 4.3.3
Actors	MEMBER
Input Data	-
Description	<p>The user can export his profile information from the EPICommunity using the “Export Profile” option as shown in the Figure 28. Pressing that button a modal window is displayed. The modal contains a form field, which allows the user to select the format of the output file. The options are:</p> <ul style="list-style-type: none"> • XML • JSON <p>The screen Figure 42 in the shows the modal interface.</p> <p>The export option is given to user in case he/she wants to import his/her profile information in another platform (eg Europass) that supports the same profile schema and definition.</p>
Output Data	The system generates a file based on the selected format containing the profile data of the user as well as the entities information associated with that profile. The result file gets downloaded by the user browser after the file generation is done.
Variations	-

5.2.6 Sync

Table 46: Functionality – Profile Sync

Functionality Name	Profile Sync
---------------------------	---------------------

Relative Cases	Use	Profile Management (Update) Paragraphs: 4.3.1
<i>Actors</i>		MEMBER
<i>Input Data</i>		Europass profile
<i>Description</i>		In case the user has provided a Europass Profile Link, he/she can use the “Sync” option to sync the profile data of the EPICCommunity platform with the Europass. This option is available for the user to maintain one profile among the two platforms. In other words, for example the user can add a new publication in his/her Europass profile and use the sync option to import this new publication instance into the EPICCommunity profile. The sync option is available in a dropdown menu that is visible when the user clicks the three dots across his name, as shown in the Figure 37.
<i>Output Data</i>		The system after parsing the Europass profile in the background, tracks the changes, and if there are any updates the user profile in the database according to the differences found. Finally, a notification is sent back to the user containing the new additions.
<i>Variations</i>		-

5.3 [Entities Functionality](#)

5.3.1 *Create*

5.3.1.1 *Course*

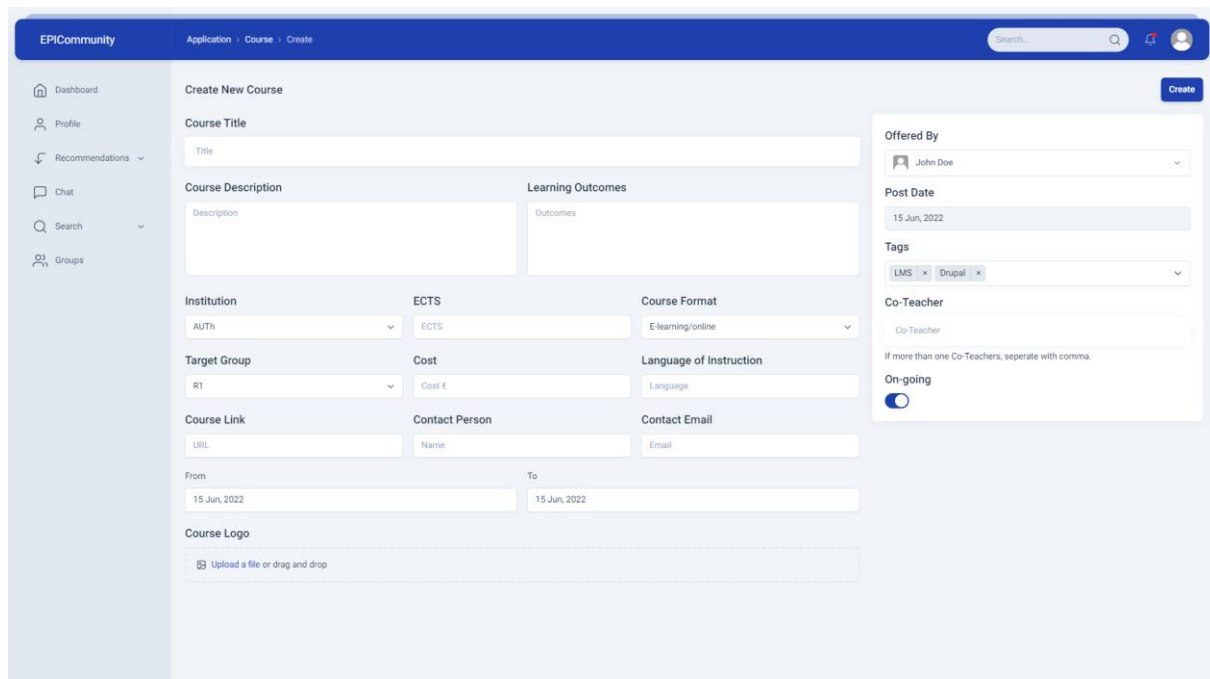


Figure 43: Course Create Screen

Table 47: Functionality – Course Create

Functionality Name	Course Create
Relative Use Cases	Management of Course, Publications, Positions and Projects Paragraphs: 4.4.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • Title • Description • Host Institution • ECTS • Course Format • Language of instruction • Target Group • Cost • Contact person • Contact email • Learning Outcomes • From (date) • To (date) • Link (optional) • Logo (optional) • Tags • On-going • Co-Teacher(s)
Description	<p>The user can use the “Create Course” button as shown in the screen Figure 29, in order to view the creation form of a new course. The form is shown in the screen Figure 43. The input data can be reviewed in the above row. The user interface also shows the fields “Offered By” and “Post Date”. Those two fields are not editable, with the corresponding values being the first/last name of the user and the current date. The creation form is validated with the rules:</p> <ul style="list-style-type: none"> • All the non-optional fields must have a value • The title must be at least 3 letters • The description must be at least 8 characters long • The To field cannot be chronologically before the From field • At least one tag • The logo must one of the following extensions [.jpg, .png, .jpeg] • The link field must a valid URL • Email fields must be valid email addresses • Cost and ECTS fields must be numbers <p>The On-going field disabled the “To” date field and sets its value to “Present”. If the user does not select a logo for the new course, EPICommunity will assign the default course logo. If an error is occurred, the invalid field/s get a red border and the error message is displayed below each field/s.</p>
Output Data	The system in case of a valid input data, creates a new course instance in the database with the values, associates that course with the user and finally redirects the user back to the page with his/her courses (Figure 29). In case of invalid, the errors are being returned and shown as alerts.
Variations	-

5.3.1.2 Publication

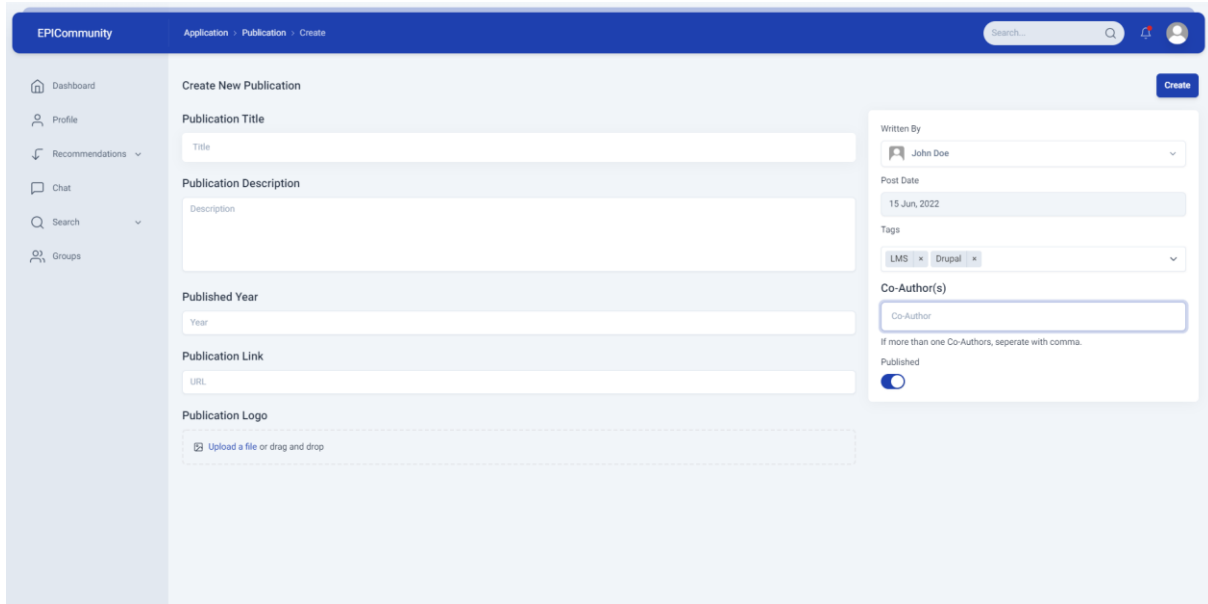


Figure 44: Publication Create Screen

Table 48: Functionality – Publication Create

Functionality Name	Publication Create
Relative Use Cases	Management of Course, Publications, Positions and Projects Paragraphs: 4.4.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • Title • Description • Co-authors • Link (optional) • Logo (optional) • Year • Tags • Co-Authors • Published
Description	<p>The user can use the “Create Publication” button as shown in the screen Figure 31, in order to view the creation form of a new publication. The form is shown in the screen Figure 44. The input data can be reviewed in the above row. The user interface also shows the fields “Written By” and “Post Date”. Those two fields are not editable, with the corresponding values being the first/last name of the user and the current date. The creation form is validated with the rules:</p> <ul style="list-style-type: none"> • All the non-optional fields must have a value • The title must be at least 3 letters • The description must be at least 8 characters long • At least one tag • The chosen logo file must one of the following extensions [.jpg, .png, .jpeg] • The link field must a valid URL

	<p>If the user does not select a logo for the new publication, EPICommunity will assign the default publication logo.</p> <p>If an error is occurred, the invalid field/s get a red border and the error message is displayed below each field/s.</p>
<i>Output Data</i>	<p>The system in case of a valid input data, creates a new publication instance in the database with the form values, associates that publication with the user and finally redirects the user back to the page with his/her publications (Figure 31). In case of invalid data, the errors are being returned and shown as alerts.</p>
<i>Variations</i>	-

5.3.1.3 Position

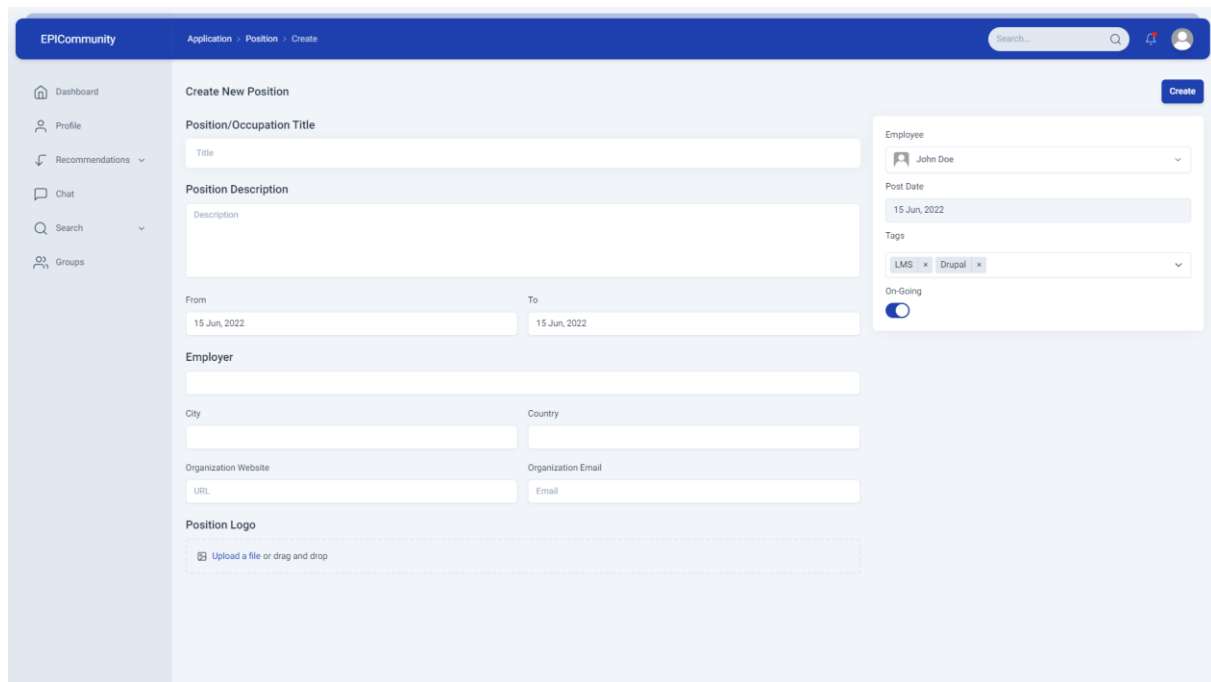


Figure 45: Position Create Screen

Table 49: Functionality – Position Create

Functionality Name		Position Create
Relative Cases	Use	Management of Course, Publications, Positions and Projects Paragraphs: 4.4.1
Actors		MEMBER
Input Data		<ul style="list-style-type: none"> • Occupation/Position title • Description • Tags • Employer • City • Country • From (date) • To (date) • On-going • Logo (optional) • Organization website (optional)

	<ul style="list-style-type: none"> • Organization email (optional)
<i>Description</i>	<p>The user can use the “Create Position” button as shown in the screen Figure 32, in order to view the creation form of a new position. The form is shown in the screen Figure 45. The input data can be reviewed in the above row. The user interface also shows the fields “Employee” and “Post Date”. Those two fields are not editable, with the corresponding values being the first/last name of the user and the current date. The creation form is validated with the rules:</p> <ul style="list-style-type: none"> • All the non-optional fields must have a value • The title must be at least 3 letters • The description must be at least 8 characters long • The To field cannot be chronologically before the From field • At least one tag • The logo must one of the following extensions [.jpg, .png, .jpeg] • The Organization website field must a valid URL • Organization email must be a valid email address <p>The On-going field disabled the “To” date field and sets its value to “Present”. If the user does not select a logo for the new position, EPICommunity will assign the default position logo. If an error is occurred, the invalid field/s get a red border and the error message is displayed below each field/s.</p>
<i>Output Data</i>	<p>The system in case of a valid input data, creates a new position instance in the database with the form values, associates that position with the user and finally redirects the user back to the page with his/her positions (Figure 32). In case of invalid data, the errors are being returned and shown as alerts.</p>
<i>Variations</i>	-

5.3.1.4 Project

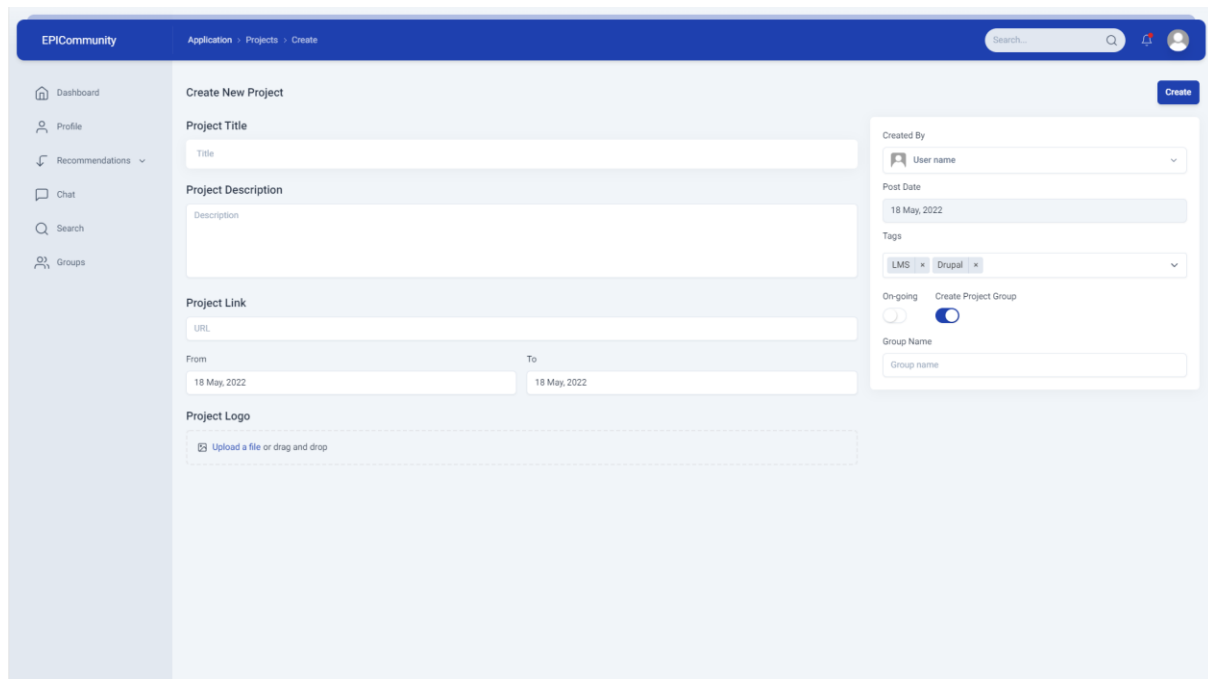


Figure 46: Project Create Screen

Table 50: Functionality – Project Create

Functionality Name	Project Create
Relative Use Cases	Management of Course, Publications, Positions and Projects Paragraphs: 4.4.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • Title • Description • Link (optional) • From (start date) • To (end date) • Logo (optional) • Tags • On-going • Create Group • Group Name
Description	<p>The user can use the “Create Project” button as shown in the screen Figure 30, in order to view the creation form of a new project. The form is shown in the screen Figure 44. The input data can be reviewed in the above row. The user interface also shows the fields “Created By” and “Post Date”. Those two fields are not editable, with the corresponding values being the first/last name of the user and the current date. The creation form is validated with the rules:</p> <ul style="list-style-type: none"> • All the non-optional fields must have a value • The title must be at least 3 letters • The description must be at least 8 characters long • The To field cannot be chronologically before the From field • At least one tag • The logo must one of the following extensions [.jpg. .png. .jpeg] • The link field must a valid URL <p>The On-going field disabled the “To” date field and sets its value to “Present”. If the user wants to create a group associated with that project, he/she must toggle the field “Create Group”. By doing so, the field “Group Name” is unlocked and a name for the group is suggested by the platform. The suggested title follows the pattern: ‘Group for’ + Project title. The user can edit the group name as he/she pleases. If the user does not select a logo for the new project, EPICommunity will assign the default project logo.</p> <p>If an error is occurred, the invalid field/s get a red border and the error message is displayed below each field/s.</p>
Output Data	The system in case of a valid input data, creates a new publication instance in the database with the form values, associates that publication with the user and finally redirects the user back to the page with his/her publications (Figure 30). In case of invalid data, the errors are being returned and shown as alerts.
Variations	-

5.3.2 Read

5.3.2.1 Course

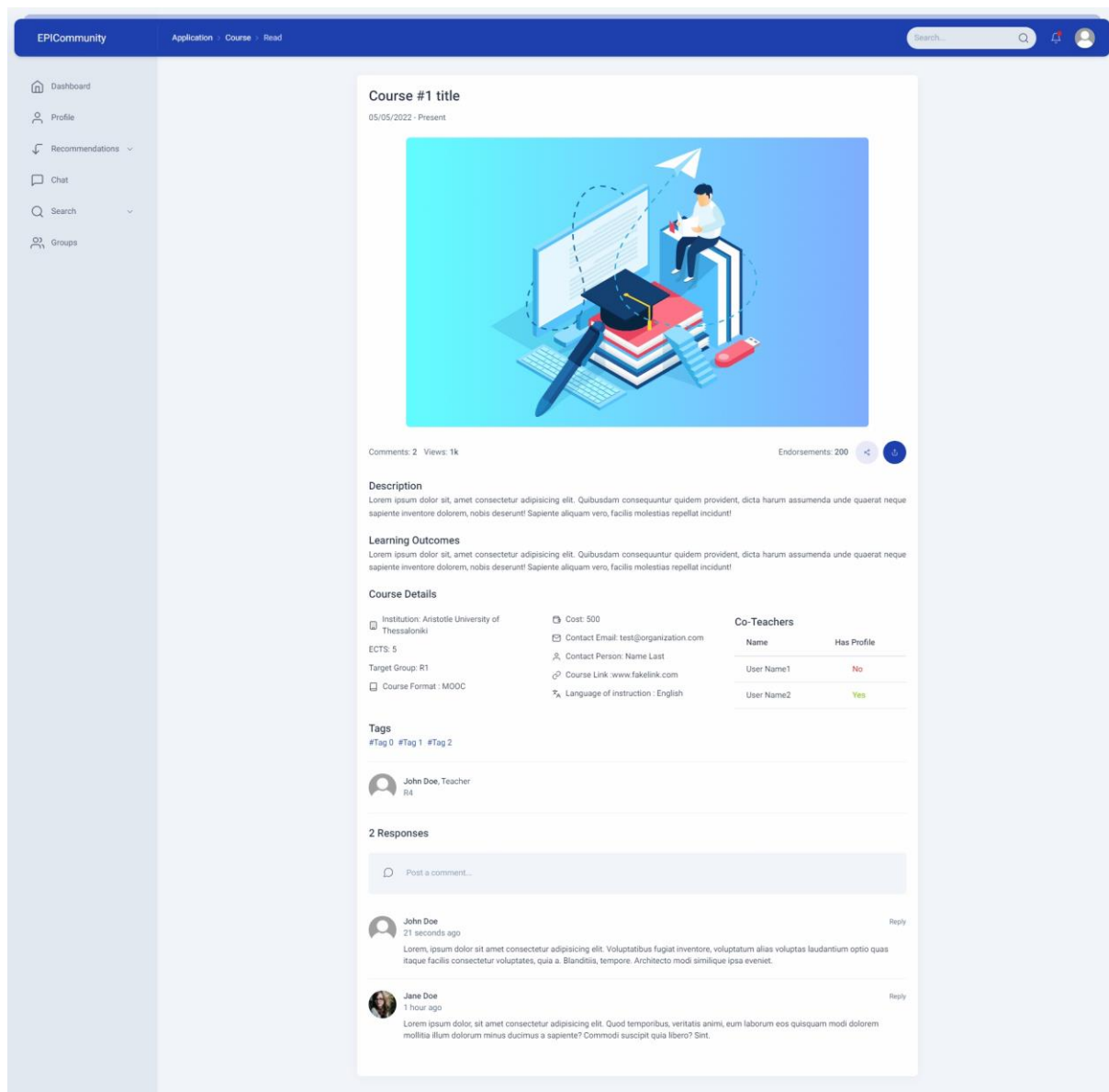


Figure 47: Course Read Screen

Table 51: Functionality – Crouse Read

Functionality Name	Course Read
Relative Use Cases	Management of Course, Publications, Positions and Projects Paragraphs: 4.4.1
Actors	MEMBER
Input Data	-
Description	By clicking on the title of a course instance, the user is redirected to a new screen showing the extended course information, like a post. The information that a user can see visiting a course is: <ul style="list-style-type: none"> • Title • Duration • Description • Tags

	<ul style="list-style-type: none"> • Name of the user that offers the course • Number of views • Number of endorsements • Number of comments • The list of the comments • Co-Teacher(s) • Host Institution • ECTS • Course Format • Language of instruction • Target Group • Cost • Contact person • Contact email • Learning Outcomes
<i>Output Data</i>	-
<i>Variations</i>	-

5.3.2.2 Publication

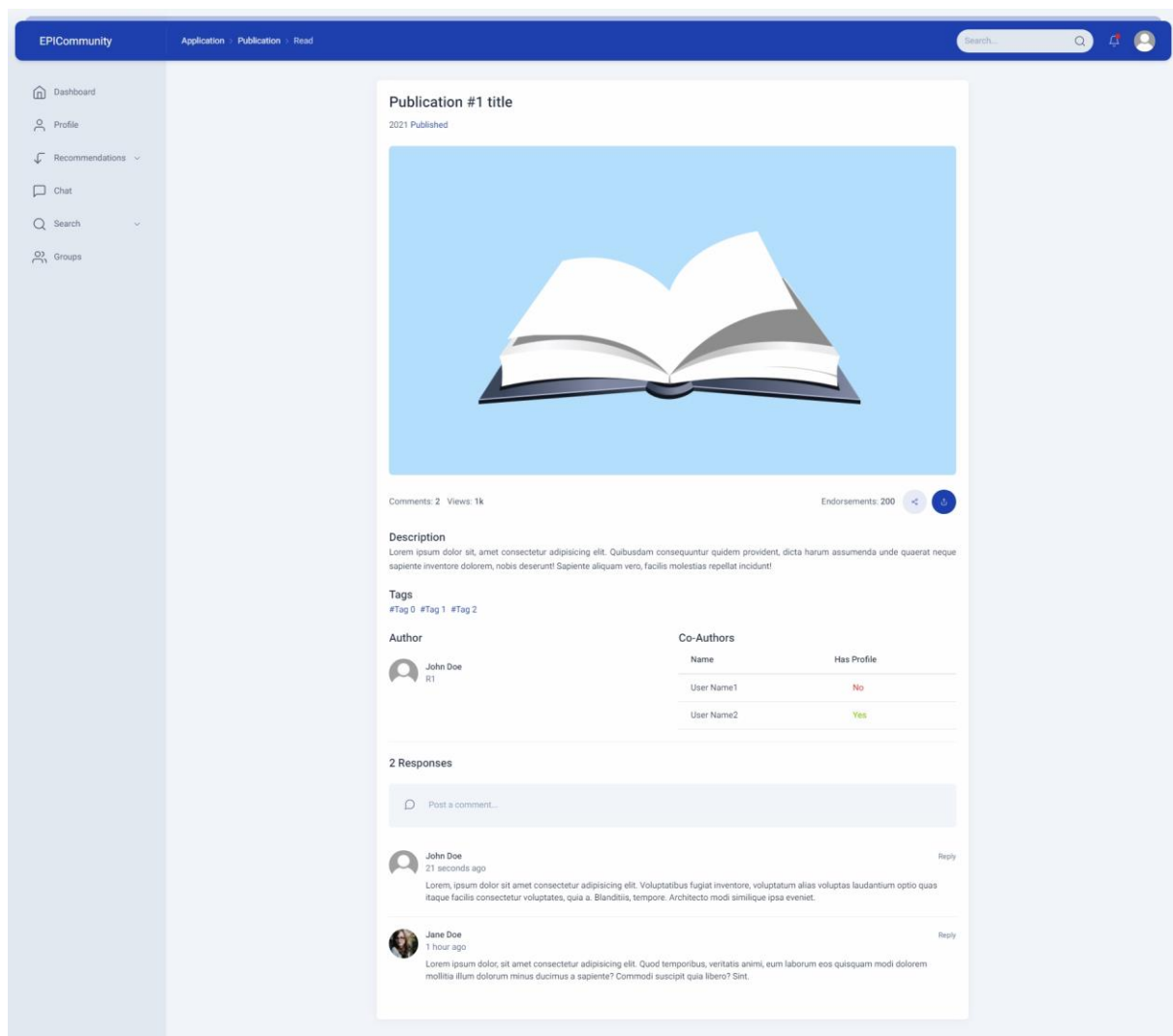


Figure 48: Publications Read Screen

Table 52: Functionality – Publication Read Screen

Functionality Name	Publication Read
Relative Use Cases	Management of Course, Publications, Positions and Projects Paragraphs: 4.4.1
Actors	MEMBER
Input Data	-
Description	<p>By clicking on the title of a publication instance, the user is redirected to a new screen showing the extended information of the publication, like a post. The information that a user can see visiting a publication is:</p> <ul style="list-style-type: none"> • Title • Published year • Publication status (published – unpublished) • Description • Co-authors • Tags • Name of the publisher • Number of views • Number of endorsements • Number of comments • The list of the comments
Output Data	-
Variations	-

5.3.2.3 Position

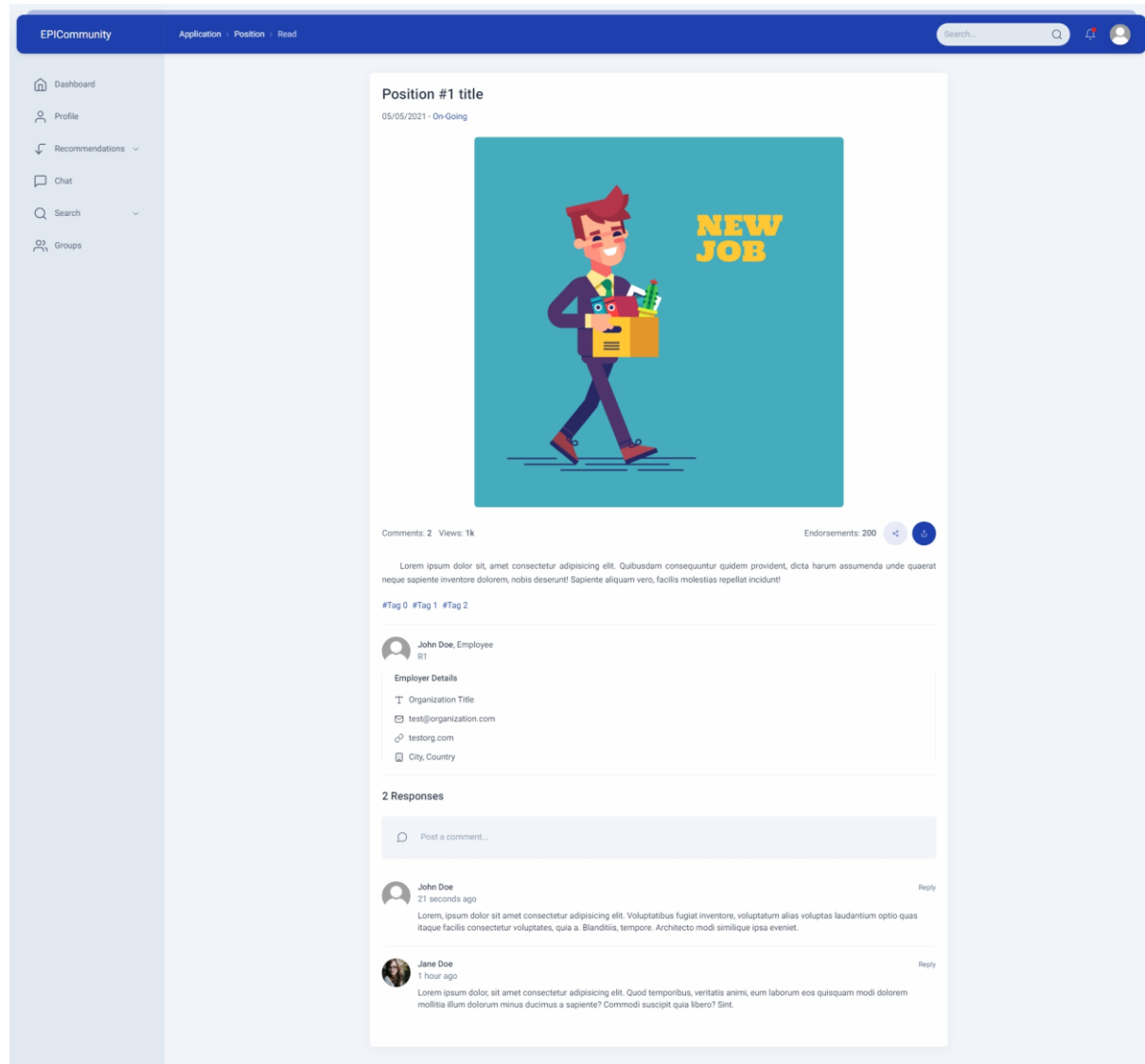


Figure 49: Position Read Screen

Table 53: Functionality – Position Read

Functionality Name	Position Read
Relative Use Cases	Management of Course, Publications, Positions and Projects Paragraphs: 4.4.1
Actors	MEMBER
Input Data	-
Description	By clicking on the title of a position instance, the user is redirected to a new screen showing the extended information of the positions, like a post. The information that a user can see visiting a position is: <ul style="list-style-type: none"> • Occupation title • Description • Tags • Employer

	<ul style="list-style-type: none"> • City • Country • From (date) • To (date) • On-going • Organization website • Organization Email
<i>Output Data</i>	-
<i>Variations</i>	-

5.3.2.4 Project

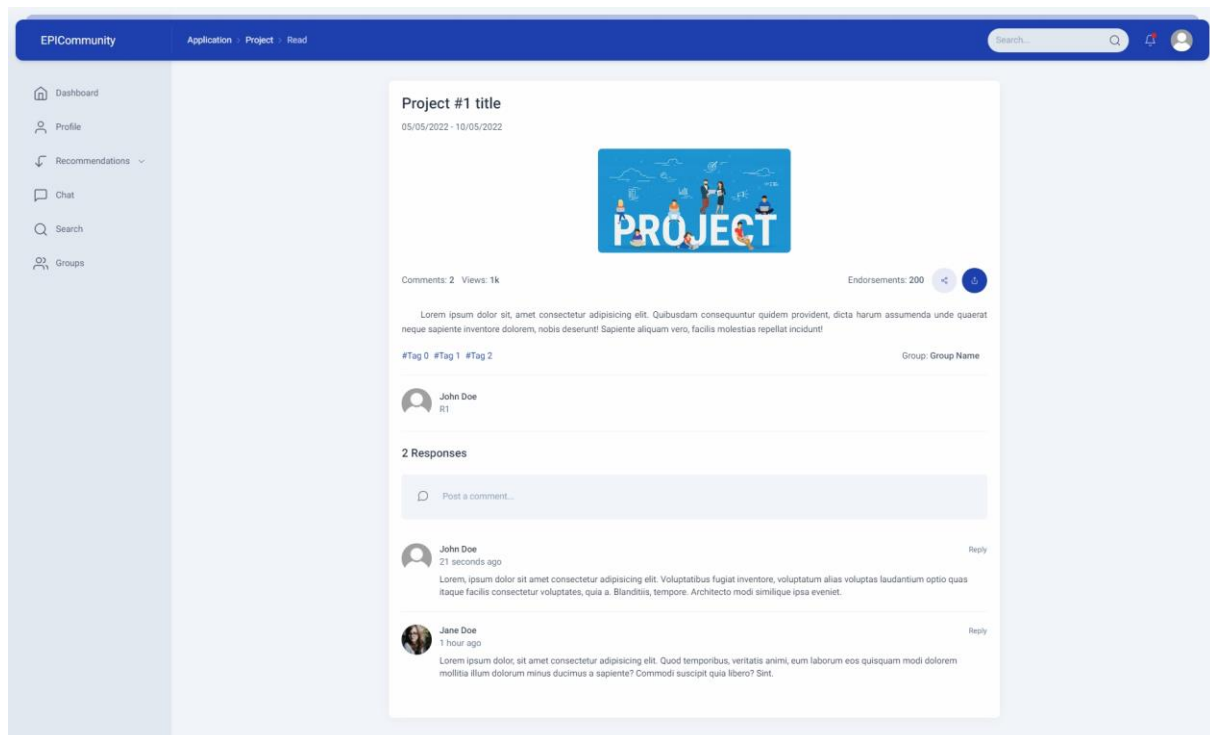


Figure 50: Project Read Screen

Table 54: Functionality – Project Read

Functionality Name	Project Read
Relative Use Cases	Management of Course, Publications, Positions and Projects Paragraphs: 4.4.1
Actors	MEMBER
Input Data	-
Description	<p>By clicking on the title of a project instance, the user is redirected to a new screen showing the extended information of the project, like a post. The information that a user can see visiting a project is:</p> <ul style="list-style-type: none"> • Title • Duration • Description

	<ul style="list-style-type: none"> • Name of the user that participated • Tags • Project group name (if exists) • Number of views • Number of endorsements • Number of comments • The list of the comments
<i>Output Data</i>	-
<i>Variations</i>	-

5.3.3 Update

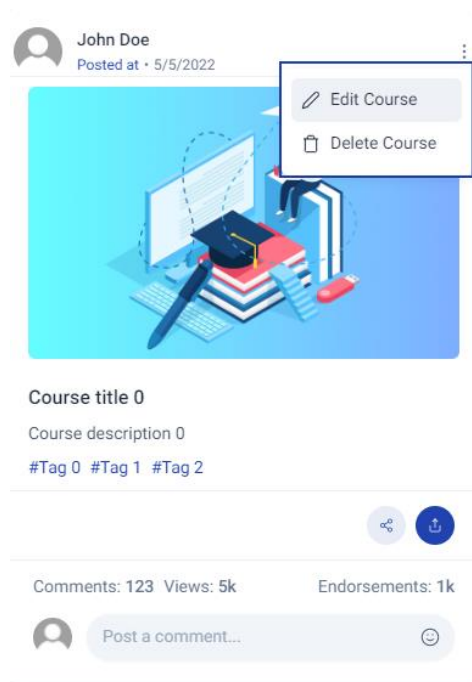


Figure 51: Edit and Delete options Screen

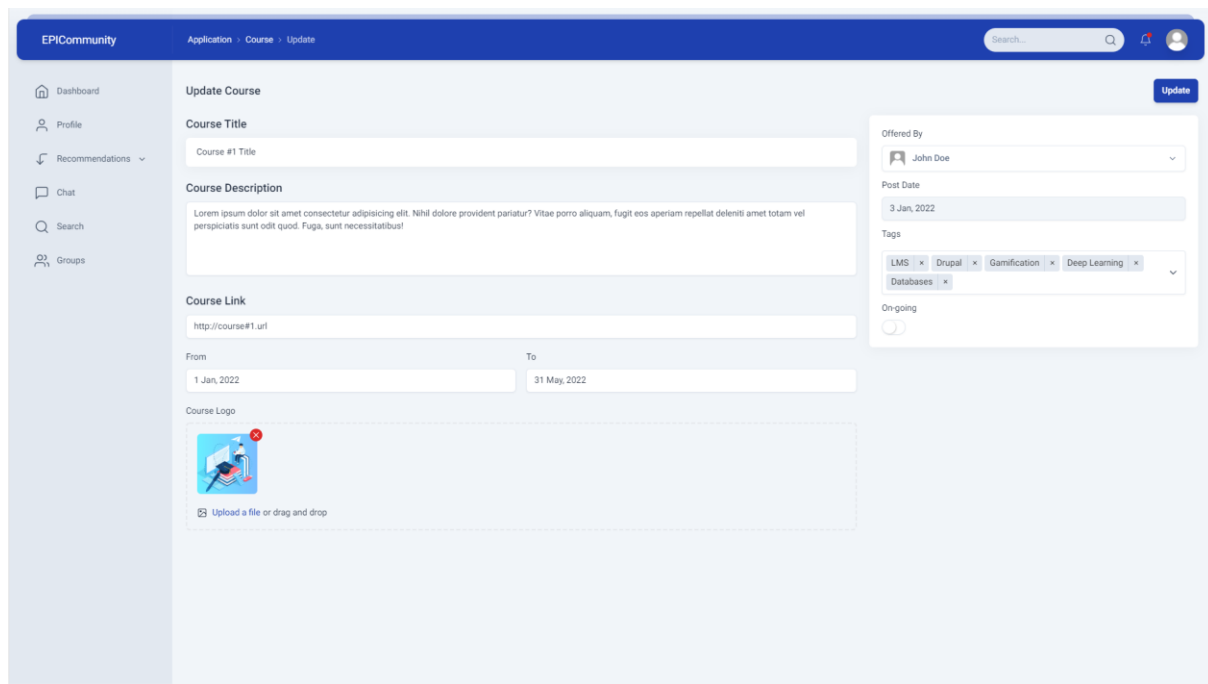


Figure 52: Course Update Screen

Table 55: Functionality – Entities Update

Functionality Name	Entities Update
Relative Use Cases	Management of Courses, Publication and Projects (Update) Paragraphs: 4.4.1
Actors	MEMBER
Input Data	<p>Common updatable fields among all entities:</p> <ul style="list-style-type: none"> Title Description Link (optional) Logo (optional) Tags <p>Project specific update fields</p> <ul style="list-style-type: none"> From (start date) To (end date) Create Group Group name <p>Course specific update fields:</p> <ul style="list-style-type: none"> From (start date) To (end date) On-going <p>On-going Publication specific update fields:</p> <ul style="list-style-type: none"> Published Year
Description	The update procedure of is common among the entities Project, Course and Publication. The creator of the instance -the user-, after visiting his/her profile, selects an entity from the menu, in order to view the list with entity instances. Every

	instance has three dots on the top right corner. Those three dots, on click, reveal the instance “Edit” and “Delete” options, as shown in the screen Figure 51. Pressing the “Edit” option, the user is redirected to a prefilled create form (for the entity create forms see paragraphs 5.3.1.1, 5.3.1.2, 5.3.1.3), based on the entity of the instance. The interface in Figure 52 displays an example of the update procedure of a course instance. Publication and Project entities follow the same logic. The corresponding updatable fields for each entity are being elaborated in the row Input Data. The validation of the fields for each entity has been analyzed in the paragraphs 5.3.1.1, 5.3.1.2, 5.3.1.3.
<i>Output Data</i>	The system after validating the input data, finds the to-be-updated entity in the database and sets its information based on the input data.
<i>Variations</i>	-

5.3.4 Delete

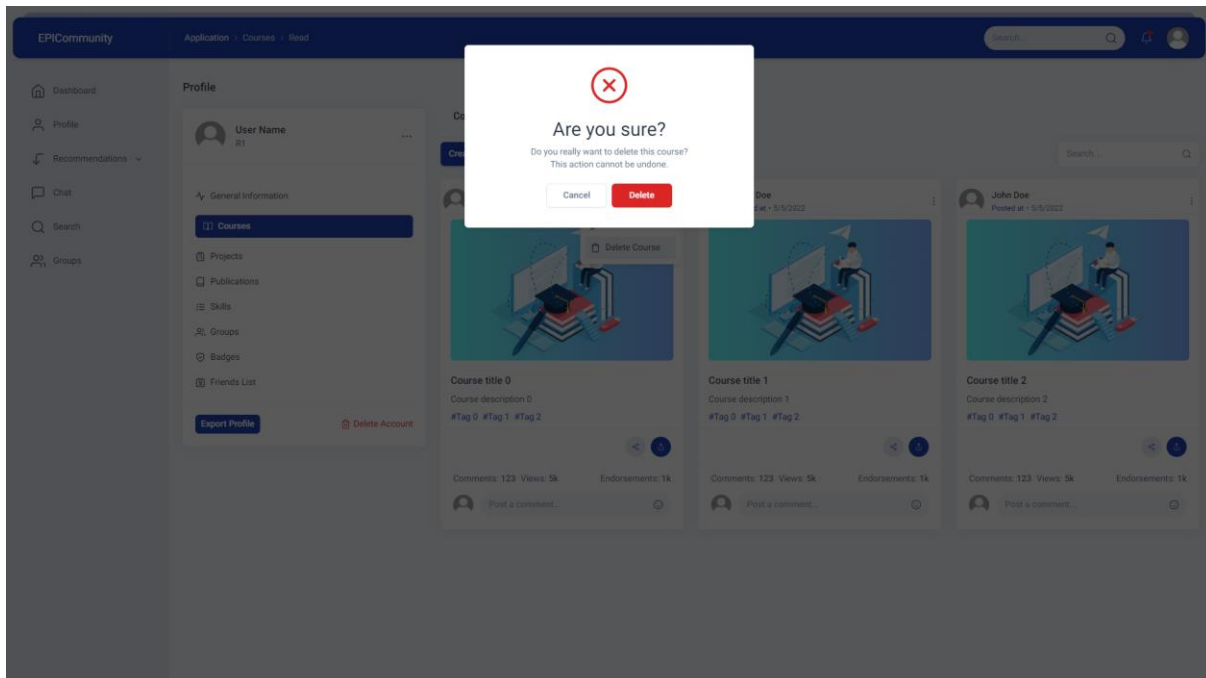


Figure 53: Course Delete Screen

Table 56: Functionality – Entities Update

Functionality Name	Entities Update
Relative Use Cases	Management of Courses, Publication and Projects (Update) Paragraphs: 4.4.1
Actors	MEMBER
Input Data	-
Description	The delete procedure of is common among the entities Project, Course and Publication. The steps are very similar to the Update procedure. Selecting the “Delete” option from the menu revealed by the three dots (explained in paragraph 5.3.3), a popup window is show asking for confirmation. If the user presses the

	“Delete” button, the instance is removed. The user interface of a course deletion is shown in the Figure 53.
<i>Output Data</i>	The system deletes the entity instance from the database and redirects the user to the list of the other instances.
<i>Variations</i>	-

5.3.5 Endorsement

Table 57: Functionality – Entities Endorsement

Functionality Name	Entities Endorsement
Relative Use Cases	User interaction with platform entities (Endorsement) Paragraphs: 4.2.3
<i>Actors</i>	MEMBER
<i>Input Data</i>	<ul style="list-style-type: none"> • Endorsed • Unendorsed • Entity ID • User ID
<i>Description</i>	The endorsement of an instance is similar between all the entities. Viewing an instance (paragraphs 5.3.2.1, 5.3.2.2, 5.3.2.3), the Endorse button (the rounded blue) located next to Share button. By pressing it the user endorses that particular instance. If a user has already liked an instance, by pressing the Endorse button again, his/her endorsement is removed.
<i>Output Data</i>	The system creates a new entry in the database, where the user (user ID) is linked with an entity (entity ID) through endorse. In case of unendorsed, this link is removed.
<i>Variations</i>	The Endorse button can be found in various screens: <ul style="list-style-type: none"> • 5.2.2.2 • 5.2.2.3 • 5.2.2.4

5.3.6 Comment

Table 58: Functionality – Entities Comment

Functionality Name	Entities Comment
Relative Use Cases	User interaction with platform entities (Comment) Paragraphs: 4.2.3
<i>Actors</i>	MEMBER
<i>Input Data</i>	<ul style="list-style-type: none"> • Comment text • Instance ID • User ID • Timestamp
<i>Description</i>	The comment of an instance is similar between all the entities. Viewing an instance (paragraphs 5.3.2.1, 5.3.2.2, 5.3.2.3), at the bottom there is a comment section. The

	user, through the text area with placeholder “Post a comment...” can create a new comment for that instance.
<i>Output Data</i>	The system creates a new entry in the database, where the user (with user ID) comments (with comment text) an instance of an entity (with instance ID) on that date (with Timestamp). The new comment is added on top of the comment list in the user interface.
<i>Variations</i>	The comment text area can be found also in screens: <ul style="list-style-type: none"> • 5.2.2.2 • 5.2.2.3 • 5.2.2.4

5.3.7 Share

Table 59: Functionality – Entities Share

Functionality Name	Entities Share
Relative Use Cases	User interaction with platform entities (Share) Paragraphs: 4.2.3
<i>Actors</i>	MEMBER
<i>Input Data</i>	<ul style="list-style-type: none"> • Instance ID
<i>Description</i>	The share of an instance is similar between all the entities. Viewing an instance (paragraphs 5.3.2.1, 5.3.2.2, 5.3.2.3), next to the Endorse button, is the Share button. Pressing that button, a URL that leads to that instance is copied to the user’s clipboard.
<i>Output Data</i>	The system gets the instance ID and generates a URL.
<i>Variations</i>	The share text area can be found also in screens: <ul style="list-style-type: none"> • 5.2.2.2 • 5.2.2.3 • 5.2.2.4

5.4 General Functionality

5.4.1 Dashboard

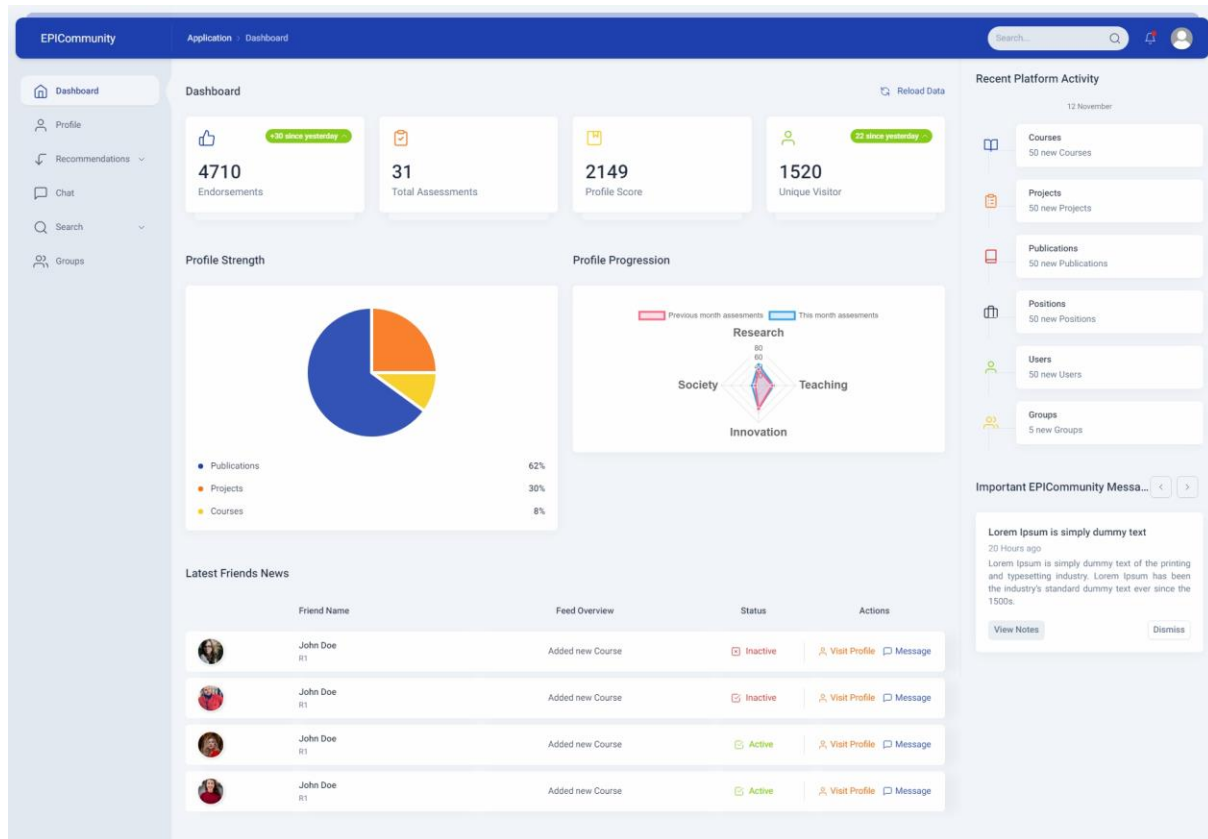


Figure 54: Dashboard Screen

Table 60: General Functionality – Dashboard

Functionality Name	Dashboard
Relative Use Cases	Profile Management (Read) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> Email Password
Description	<p>The Dashboard is the main screen of the EPICCommunity members. The screen Figure 54 shows user interface. In the top of the Dashboard are displayed 4 cards containing some generic analytics of the user’s profile. Specifically:</p> <ul style="list-style-type: none"> Number of Endorsements, plus the 24 hours change How many Assessments the profile has received The Profile Score The number of the unique visitors, plus the 24-hour change (views) <p>Below the 4 analytics cards, there are two charts.</p> <ul style="list-style-type: none"> Pie chart, displaying the percentage of each EPICCommunity entity, based on the users’ posts. Radar chart, displaying the score in each role based on the peer assessments. <p>Following the charts, the screen contains a table with the friend’s new feed. Each row contains:</p> <ul style="list-style-type: none"> Friend image

	<ul style="list-style-type: none"> • Friend name • Feed overview (eg Add new Course) • Friend Status (Active – Inactive) • Actions (Visit Profile, Message) <p>The right part of the Dashboard is dedicated to EPICommunity platform general news and messages.</p> <p>The news contain the number of instances created on each entity the past 24 hours, as well as how many new users join the platform and how many new groups are created. Below are the announcements area. The announcements are special EPICommunity messages, send by the community itself or the system administrator.</p>
<i>Output Data</i>	-
<i>Variations</i>	-

5.4.2 Notification

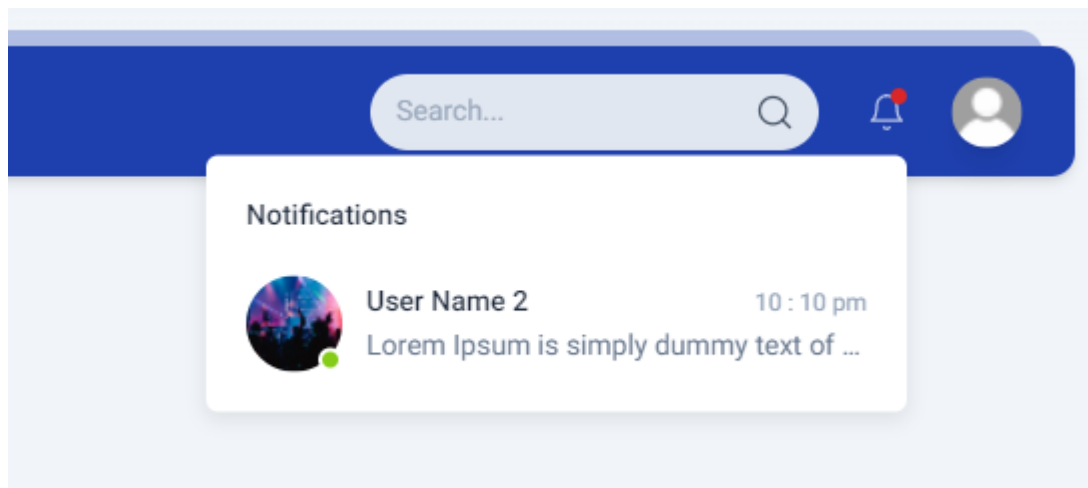


Figure 55: Notifications Screen

Table 61: General Functionality – Notification

Functionality Name	Notification
Relative Use Cases	Profile Management (Read) Paragraphs: 4.3.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • Email • Password
Description	<p>This functionality aims to inform the user about regarding some interactions that he might want to know. The notifications are pushed to the user by the system when an important even is triggered. Events that trigger a pushed notification are:</p> <ul style="list-style-type: none"> • New Message • New friend request • New comment on an instance (publication, course, project)

	The notifications are on the top right corner next to the user icon, as the screen Figure 55 displays. Notifications that have been read, are automatically dismissed after 3 days.
<i>Output Data</i>	All notifications are stored in the database until they expire.
<i>Variations</i>	

5.4.3 Send Contact Request

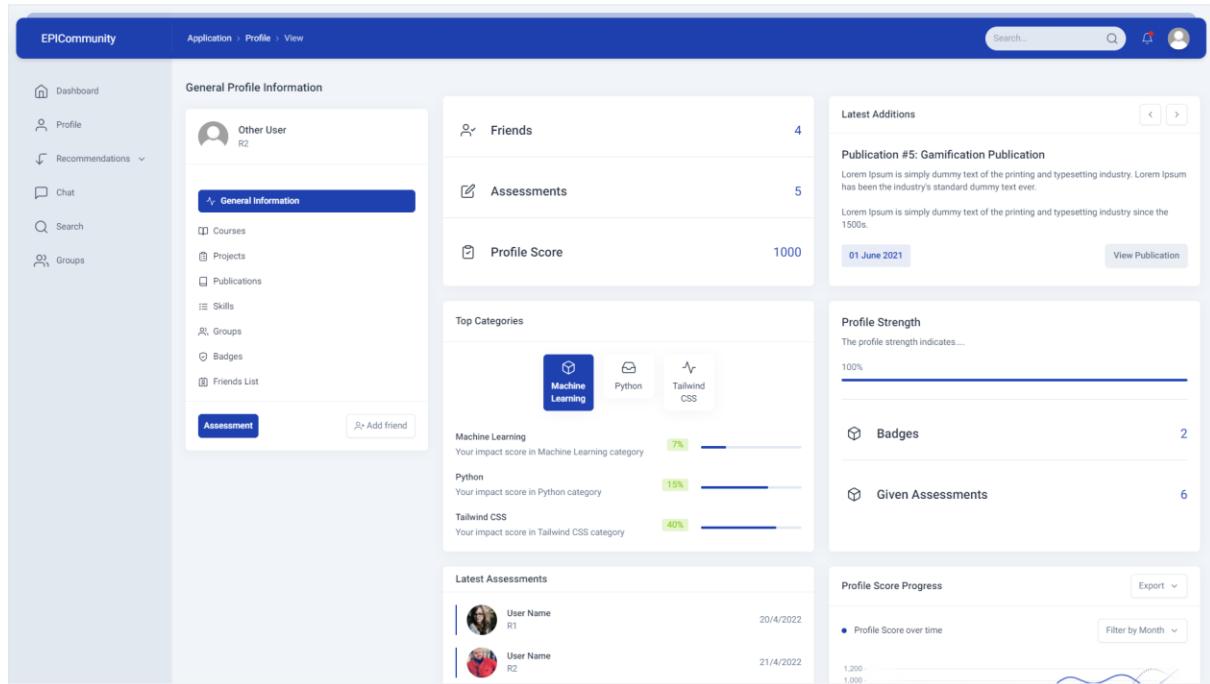


Figure 56: Another Members’ Profile Screen

Table 62: General Functionality – Send Contact Request

Functionality Name	Send Contact Request
Relative Use Cases	Entities Interaction / User to User / Contact Request Paragraphs: 4.2.1.3
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> User ID User ID
Description	Visiting another users’ profile, the user can send a contact request using the button “Add friend” as displayed in the right side of the Figure 56.
Output Data	The system sends a notification to the user regarding the incoming contact request.
Variations	-

5.4.4 Private Message

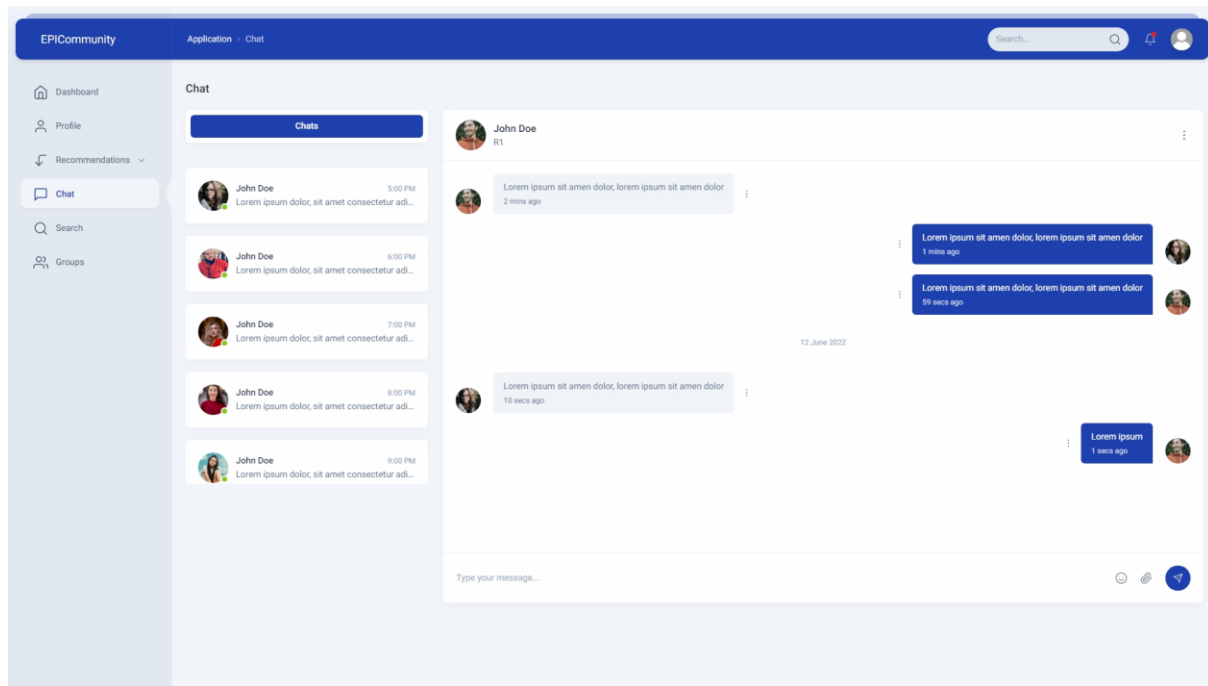


Figure 57: Private Message Screen

Table 63: General Functionality – Private Message

Functionality Name	Private Message
Relative Use Cases	Entities Interaction / User to User / Private Message Paragraphs: 4.2.1.4
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • User ID • Message • User ID
Description	The private message functionality allows the user to chat with each other. Using the “Chat” option of the sidebar the user can send a private message to a contact. The screen Figure 57 displays the user interface.
Output Data	The system takes the message send by a user1 (with user ID), adds the message to the user2 (with user ID) inbox and pushes a notification to inform the user2 for the incoming message. All messages are stored in the database.
Variations	There are various options to send a new message to another user. Those can be found in the form of button “Message”, in the following interfaces: <ul style="list-style-type: none"> • Figure 36 • Figure 54

5.4.5 Assessment

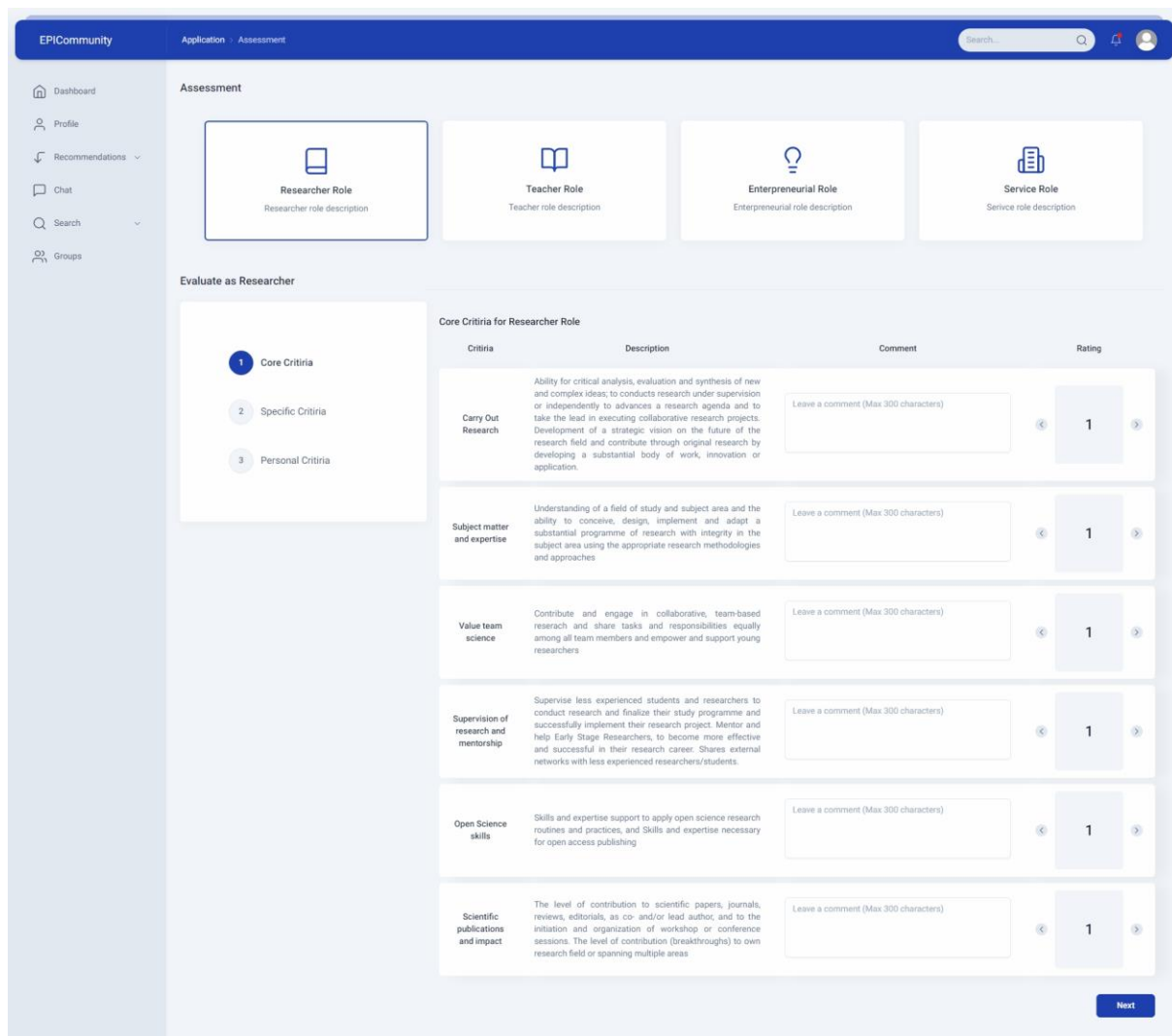


Figure 58: Assessment Screen

Table 64: General Functionality – Assessment

Functionality Name	Assessment
Relative Use Cases	Entities Interaction / User to User / Evaluate User Paragraphs: 4.2.1.5
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • User ID • User ID • Ratings of Core criteria for Researcher Role • Ratings of Specific criteria for Researcher Role • Ratings of Personal criteria for Researcher Role • Ratings of Core criteria for Teacher Role • Ratings of Specific criteria for Teacher Role • Ratings of Personal criteria for Teacher Role • Ratings of Core criteria for Entrepreneurial Role • Ratings of Specific criteria for Entrepreneurial Role • Ratings of Personal criteria for Entrepreneurial Role • Ratings of Core criteria for Service Role

	<ul style="list-style-type: none"> • Ratings of Specific criteria for Service Role • Ratings of Personal criteria for Service Role
<i>Description</i>	<p>The peer assessment is a unique functionality of the EPICommunity. The evaluation procedure has fully adopted the EPIQAssess roles and criteria. A member can choose to evaluate another member, only in a specific role, but has to complete all the criteria for that role (core, specific and personal). Each criteria has a list of different aspects regarding that criteria. The user can choose not to submit a rating for an aspect. The rating is between 1= lowest and 5 highest.</p> <p>The interface Figure 58 shows the rating of member, for the based criteria of the researcher role. By clicking next, the user is moved to the specific criteria. A “Previous” button will also be available to check the rating on the previous criteria if needed.</p> <p>The evaluation of a member expires after 6 months.</p>
<i>Output Data</i>	The system will receive the rating of the member and store it in the database.
<i>Variations</i>	-

5.5 Groups Functionality

5.5.1 *Create*

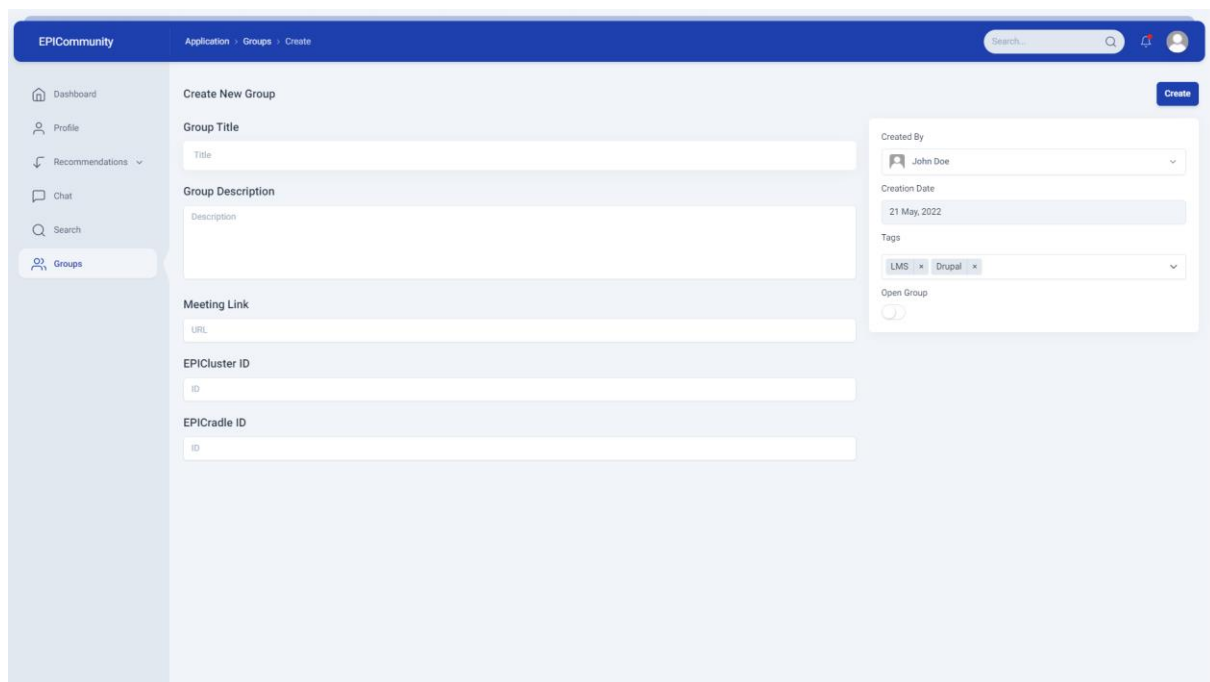


Figure 59: Group Create Screen

Table 65: Groups Functionality – Create

Functionality Name	Create Group	
Relative Cases	Use	Group Management (Create) Paragraphs: 4.5.1
<i>Actors</i>		MEMBER

<i>Input Data</i>	<ul style="list-style-type: none"> • Group name • Description • Group annotations (tags) • Group type (open/closed) <p>The optional group fields are:</p> <ul style="list-style-type: none"> • Meeting link • Project ID • EPICluster (i.e., to characterize the group as an EPICluster) • EPICradle (i.e., to characterize the group as an EPICradle)
<i>Description</i>	<p>The screen Figure 59 displays the user interface regarding the creation of a new group. The above row contains the information regarding the fields. The submitted data must follow the rules:</p> <ul style="list-style-type: none"> • Group name should be unique • Description should be at least 10 characters long • At least 1 tag
<i>Output Data</i>	<p>The system validates the data, generates a group ID and stores the data on the database.</p>
<i>Variations</i>	-

5.5.2 Read

5.5.2.1 List

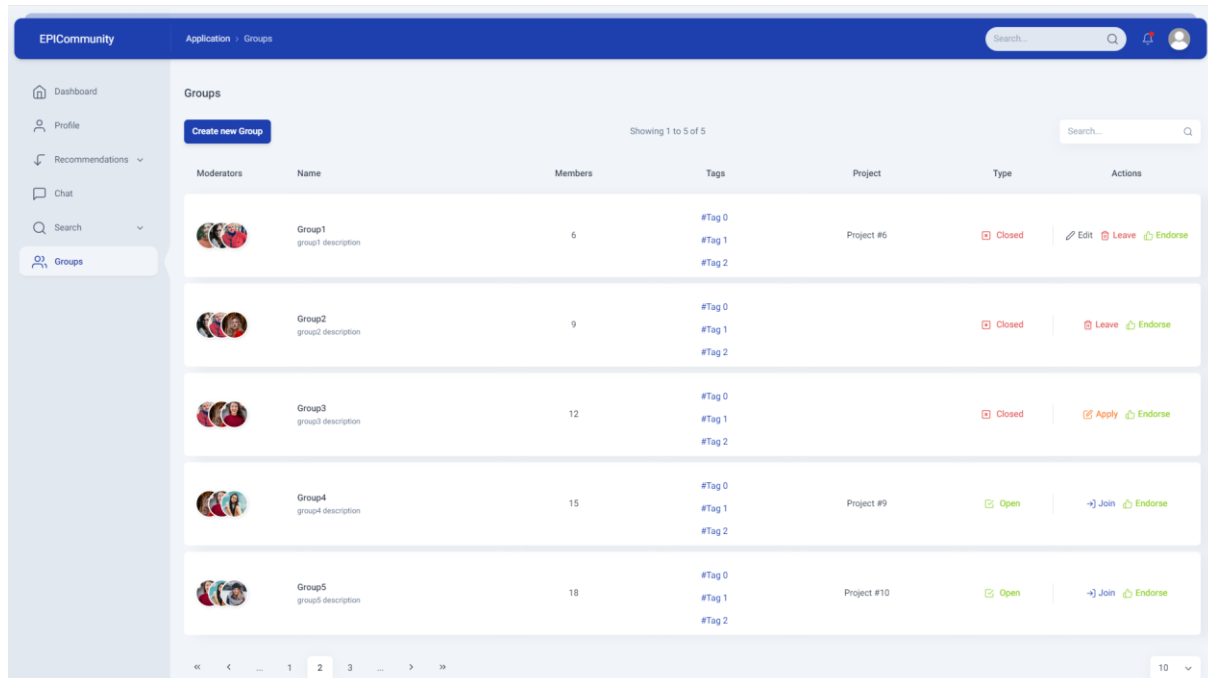


Figure 60: Groups List Screen

Table 66: Groups Functionality – Groups List

Functionality Name	Groups List
Relative Use Cases	Group Management (Read) Paragraphs: 4.5.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> Email Password
Description	<p>Using the “Groups” option of the sidebar a member can view all the groups of the EPICommunity in a form of table. Each row of the table contains information regarding a group instance. The information is:</p> <ul style="list-style-type: none"> List of moderators Group name Group description Number of members Tags If the group is attached to a project instance The type of the group (closed, open) Actions <p>The “Actions” column in dynamic based on, if the user is already member of the group and if the user is moderator of the group. In such cases, the options “Edit” and “Leave” are displayed. On the other hand, the option “Join” is displayed if the group is open, or the option “Apply” if the group is closed. Finally, the action “Endorse” is displayed in case the user have not endorsed the group previously. The user interface can be found in the Figure 60.</p>
Output Data	-
Variations	-

5.5.2.2 General Information

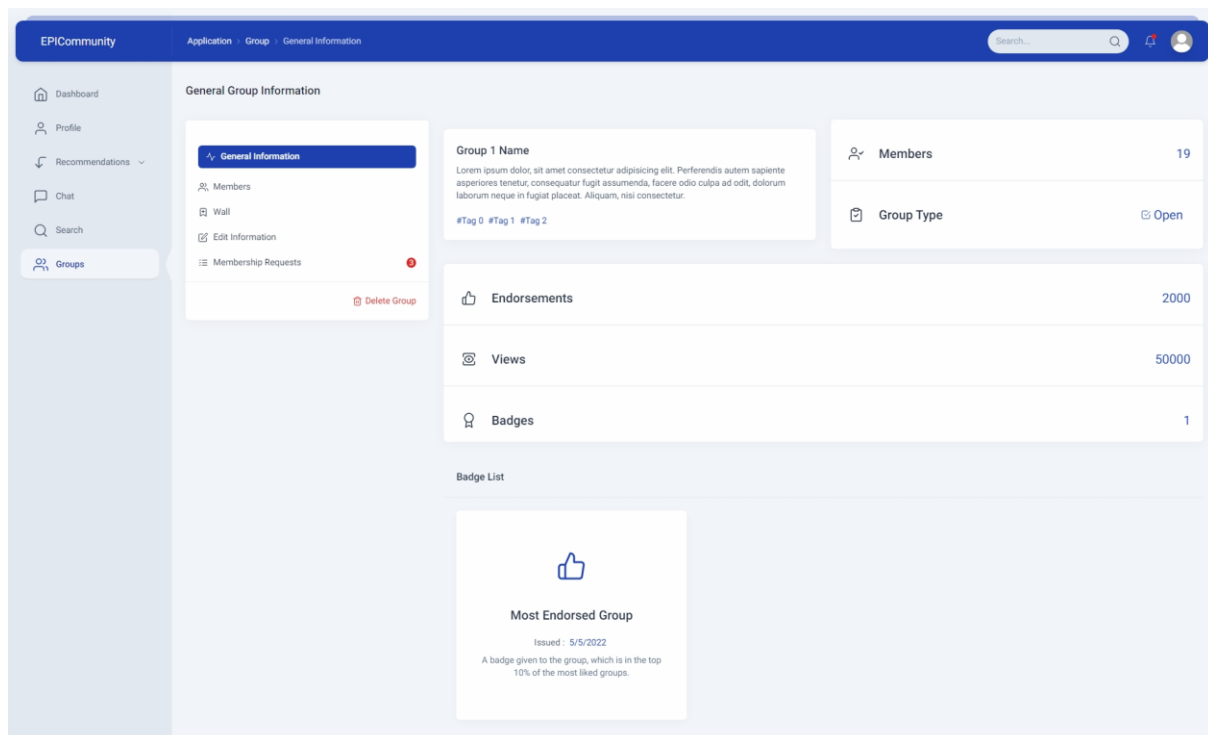


Figure 61: Group General Information Screen

Table 67: Groups Functionality – Group General Information

Functionality Name	Group General Information
Relative Use Cases	Group Management (Read) Paragraphs: 4.5.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> Email Password
Description	<p>Clicking on a group from the list Figure 60, the member is redirected in the general information screen, Figure 61. Here one can see the following information:</p> <ul style="list-style-type: none"> Group name Group description Group tags Number of members Group type Number of endorsements Number of views Number of badges List of badges <p>In the left side of the interface, there is a menu. The options of that menu are displayed according to the group type (open), the user role in the group (moderator) and the user permissions (view wall, edit information).</p>
Output Data	-
Variations	-

5.5.2.3 Members

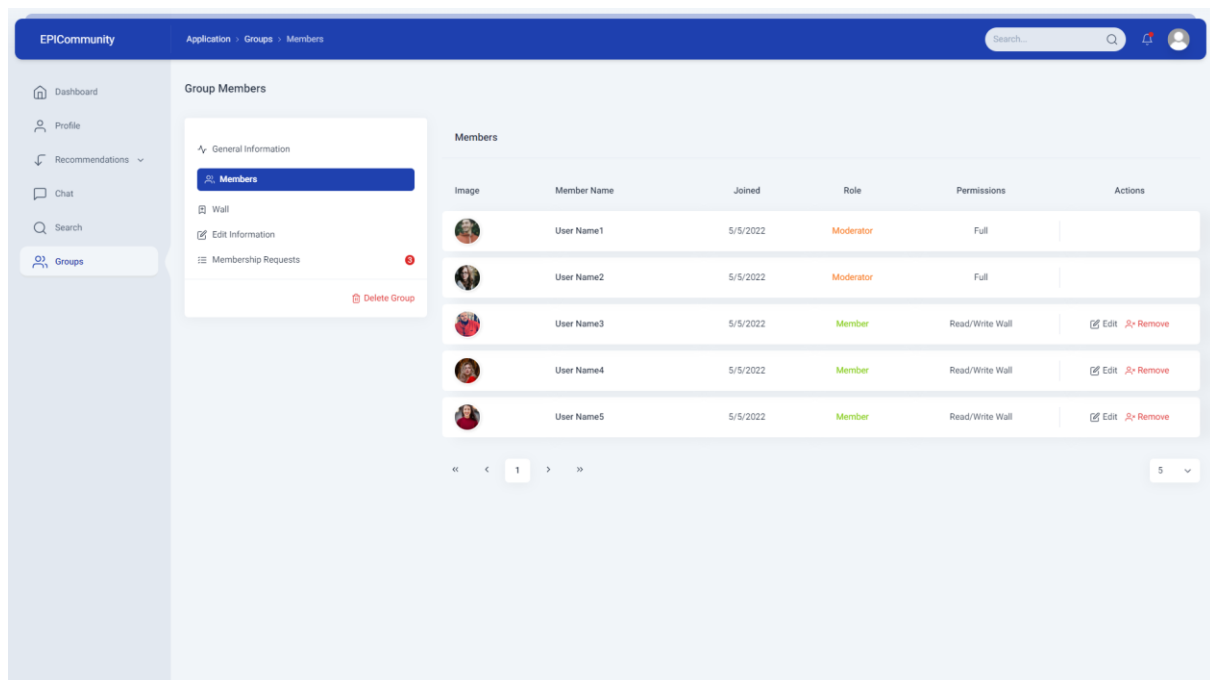


Figure 62: Group Members Screen

Table 68: Groups Functionality – Group View Members

Functionality Name	Group View Members
Relative Use Cases	Group Management (Read) Paragraphs: 4.5.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> Email Password
Description	<p>The interface in the Figure 62 shows the members of a group in a table form. Each table row consists of:</p> <ul style="list-style-type: none"> Member image Member name Date join Role Permissions Actions <p>The actions field contains the options to remove a member from the group, and the option to edit the members' permissions and role.</p>
Output Data	-
Variations	-

5.5.2.4 Wall

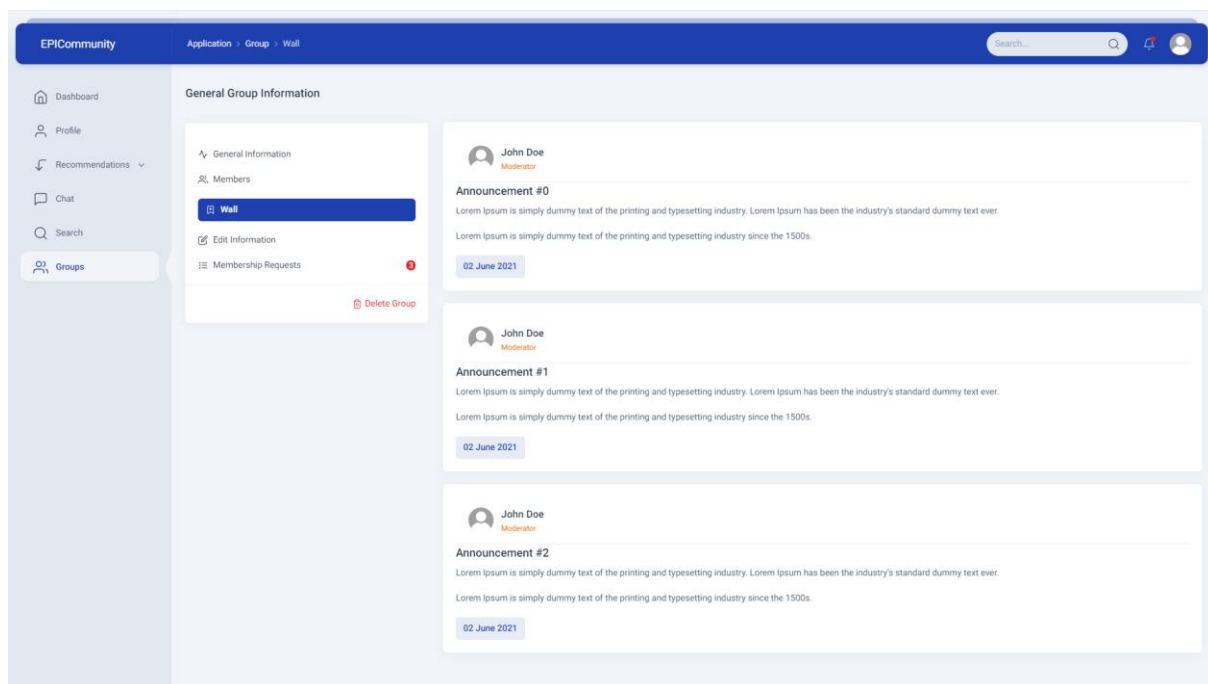


Figure 63: Group Wall Screen

Table 69: Groups Functionality – Group Wall

Functionality Name	Group Wall
---------------------------	-------------------

Relative Cases	Use	Group Management (Read) Paragraphs: 4.5.1
<i>Actors</i>		MEMBER
<i>Input Data</i>		<ul style="list-style-type: none"> Email Password
<i>Description</i>		The Wall of a group is a place where the moderators can post announcements. The screen in Figure 63 shows the interface. Members of the group, in case of a closed group, or anyone in case of an open group, can view those announcements.
<i>Output Data</i>		-
<i>Variations</i>		-

5.5.2.5 Membership Requests

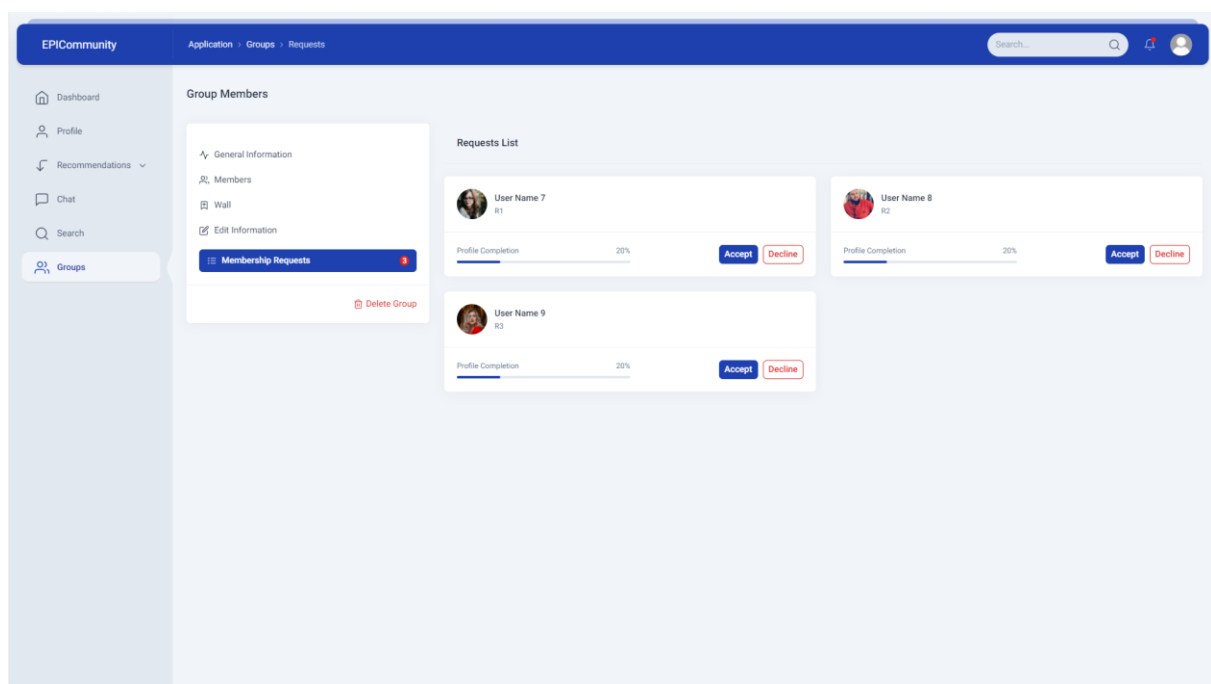


Figure 64: Group Membership Requests Screen

Table 70: Groups Functionality – Manage Group Membership Requests

Functionality Name	Manage Group Membership Requests	
Relative Cases	Use	Group Member Management Paragraphs: 4.5.2
<i>Actors</i>		MEMBER
<i>Input Data</i>		<ul style="list-style-type: none"> User ID Group ID
<i>Description</i>		When a group is closed, the users apply to join on that group. The moderators can view pending membership requests as shown in the screen Figure 64. Moderators can accept a request or decline a request.
<i>Output Data</i>		The system if a membership request gets accepted, adds the applicant (with user ID) to the members of the group (with group ID). If the request get declined the

	system deletes the pending request, and unlocks the option for the user to reapply for membership to the group.
<i>Variations</i>	-

5.5.3 Update

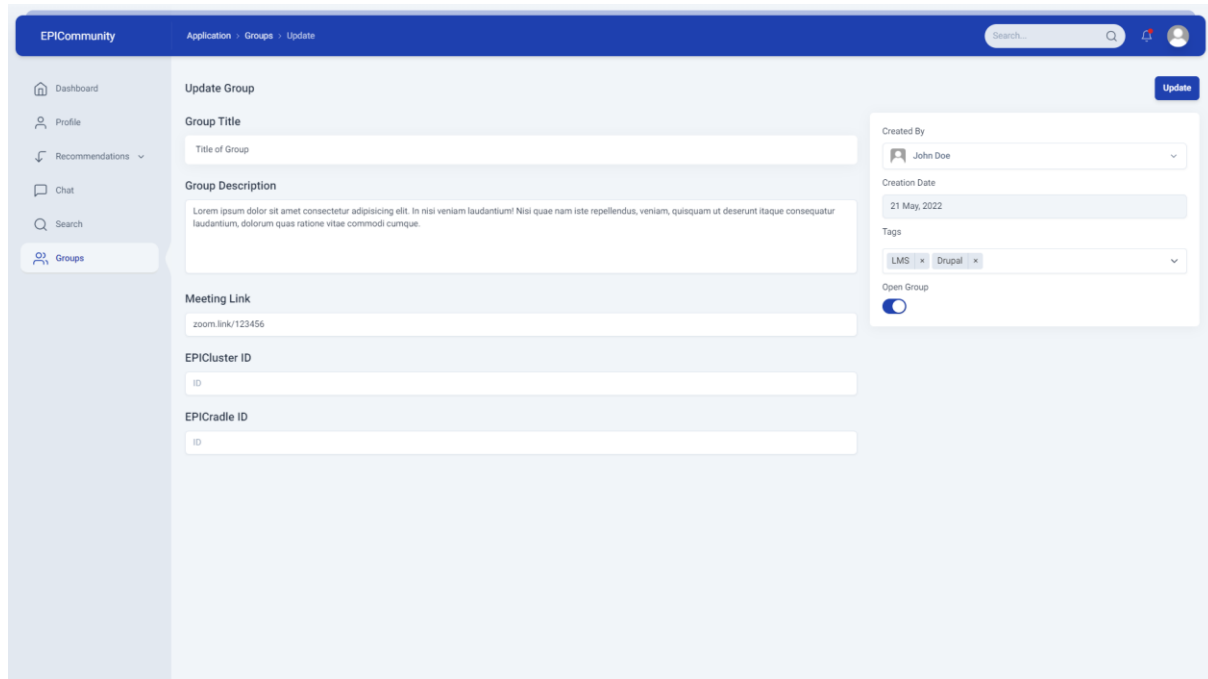


Figure 65: Group Update Screen

Table 71: Groups Functionality – Update Group

Functionality Name	Update Group
Relative Use Cases	Group Management (Update) Paragraphs: 4.5.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> Group ID Group name Description Group annotations (tags) Group type (open/closed) <p>The optional group fields are:</p> <ul style="list-style-type: none"> Meeting link Project ID EPICcluster (i.e., to characterize the group as an EPICcluster) EPICradle (i.e., to characterize the group as an EPICradle)
Description	Group moderators or users with edit permission can update the information of the group. Using the same form as in create group functionality, but prefilled with the groups data, the allowed group members can change the data of the group. The

	“Edit” button can be found in the list of the groups that the user is already member, shown in interface Figure 34.
<i>Output Data</i>	The system takes the update information based on the user input, and updates the group (with group ID) in the database.
<i>Variations</i>	The “Edit” button can also be found in the list of all the platform groups, as shown in the Figure 60.

5.5.4 Delete

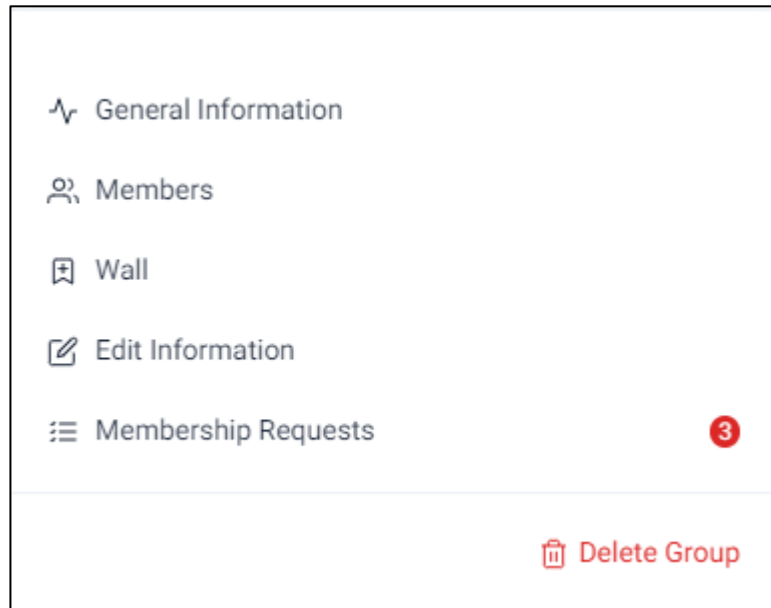


Figure 66: Group Side Menu

Table 72: Groups Functionality – Delete Group

Functionality Name	Delete Group	
Relative Use Cases	Use	Group Management (Delete) Paragraphs: 4.5.1
Actors		MEMBER
Input Data		<ul style="list-style-type: none"> Group ID
Description		As described in the relative use case, a group can be deleted. The circumstances are the same as in the use case. The moderator of a group can delete the group from the EPICommunity using the “Delete Group” option as shown in the Figure 66.
Output Data		The system deletes the group (with group ID) and all the associated data with it.
Variations		

5.5.5 Create Announcement

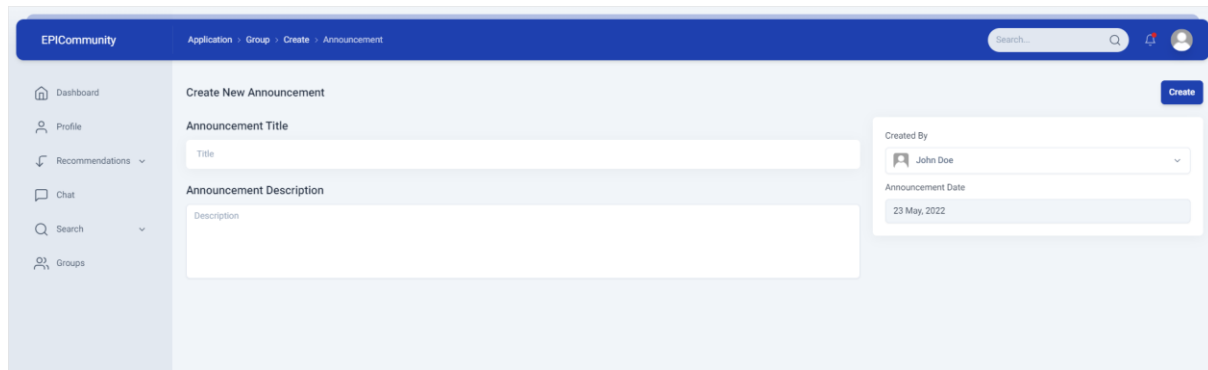


Figure 67: Create Announcement Screen

Table 73: Groups Functionality – Create Announcement

Functionality Name	Create Announcement
Relative Use Cases	Group Management Paragraphs: 4.5.1
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • User ID • Group ID • Announcement Title • Announcement Description • Date
Description	<p>Group moderators can create group announcements, that are posted in the group wall. The group announcement a title and a description. The screen Figure 67 shows the creation form of a new announcement. The validation rules are:</p> <ul style="list-style-type: none"> • Title must be at least 4 characters long • Description must be at least 8 characters long <p>The date field and the user that creates the announcement are disabled and autocompleted by the platform.</p>
Output Data	The system creates a new announcement for the group and persists the announcement information in the database.
Variations	-

5.5.6 Enrolment

Table 74: Groups Functionality – Group Enrolment

Functionality Name	Group Enrolment
Relative Use Cases	Group Enrolment Paragraphs: 4.5.3
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> • User ID • Group ID
Description	A user can become a member of the group using the “Join” option, in the “Actions” column, in case of an open group, as shown in the Figure 60. In that case he/she will

	automatically become the member of the group. If the group is closed the user can use the option “Apply” to send a membership request to the moderators of the group.
<i>Output Data</i>	If the group is open the user clicks the “Join”, the system add the user (with user ID) to the member of the group (with group ID) and stores to the database. If the group is closed and the user clicks “Apply”, a new membership request is created for the group (with group ID) and stored in the database.
<i>Variations</i>	-

5.5.7 Leave Group

Table 75: Groups Functionality – Leave Group

Functionality Name	Leave Group
Relative Use Cases	Leave Group Paragraphs: 4.5.4
<i>Actors</i>	MEMBER
<i>Input Data</i>	<ul style="list-style-type: none"> • User ID • Group ID
<i>Description</i>	A member of a group can leave the group using the “Leave” option as shown in the Figure 34.
<i>Output Data</i>	The system removes the user (with user ID) from the members of the group (with group ID).
<i>Variations</i>	The “Leave” option can also be found in the Figure 60.

5.6 Search Functionality

5.6.1 Entities (Courses/Publications/Projects)

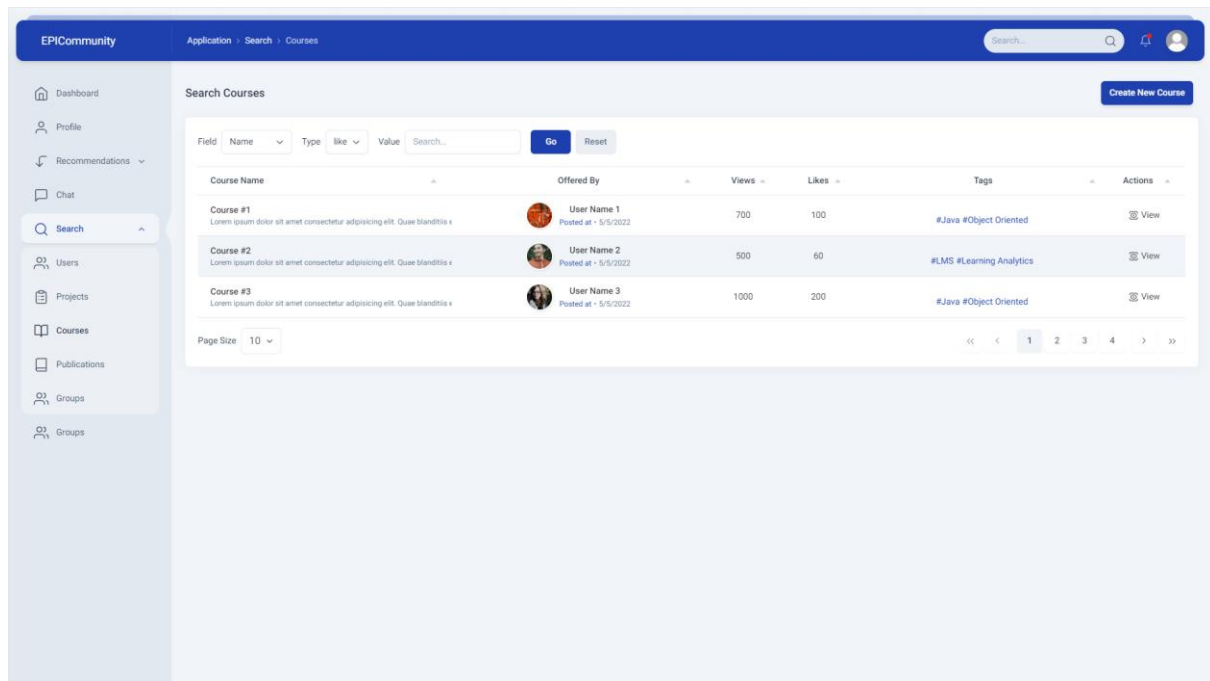


Figure 68: Search Courses Screen

Table 76: Search Functionality – Courses/Publications/Projects

Functionality Name	Search Courses/Publications/Projects	
Relative Use Cases	Use	Search of Courses, Publications, Positions and Projects Paragraphs: 4.4.2
Actors		MEMBER
Input Data		<p>Common:</p> <ul style="list-style-type: none"> • Name (result contains the characters of the user input) • Associated Tags (result contains the characters of the user input) • Number of Endorsements • Number of Comments <p>Courses:</p> <ul style="list-style-type: none"> • On-going <p>Publications:</p> <ul style="list-style-type: none"> • Year <p>Projects:</p> <ul style="list-style-type: none"> • On-going
Description		Members and Subscribers can search for instances of entities in the EPICommunity platform. The search criteria can be found in the above row. The screen Figure 68 displays the user interface for the search of courses. The interface is similar for the entities.
Output Data		The matched instances are returned.
Variations		-

5.6.2 Users

Table 77: Search Functionality – Users

Functionality Name	Search Users	
Relative Use Cases	Use	Search Users Paragraphs: 4.2.1.1
Actors		MEMBER
Input Data		<ul style="list-style-type: none"> • First name (result contains the characters of the user input) • Last Name (result contains the characters of the user input) • Top categories tags • Publication name (result contains the characters of the user input) • Number of Publications • Project name (result contains the characters of the user input) • Number of Projects • Course name (result contains the characters of the user input) • Number of Courses • Number of Friends • Number of assessments that the profile has received • Number of assessments made • Profile score

	<ul style="list-style-type: none"> Assessment scores per role (Researcher, Innovation, Entrepreneur, Society) Number of groups that s/he is member of
<i>Description</i>	The Users search mechanism offers a set of extensive fields, in order for a very specific query to be made. The fields can be found in the above row. Members and Subscribers can both use the Users search, but members can also send a contact request.
<i>Output Data</i>	The users that match the query fields.
<i>Variations</i>	-

5.7 Recommendations Functionality

The Members of the EPICommunity also have the Recommendations option available on their side bar.

5.7.1 Users

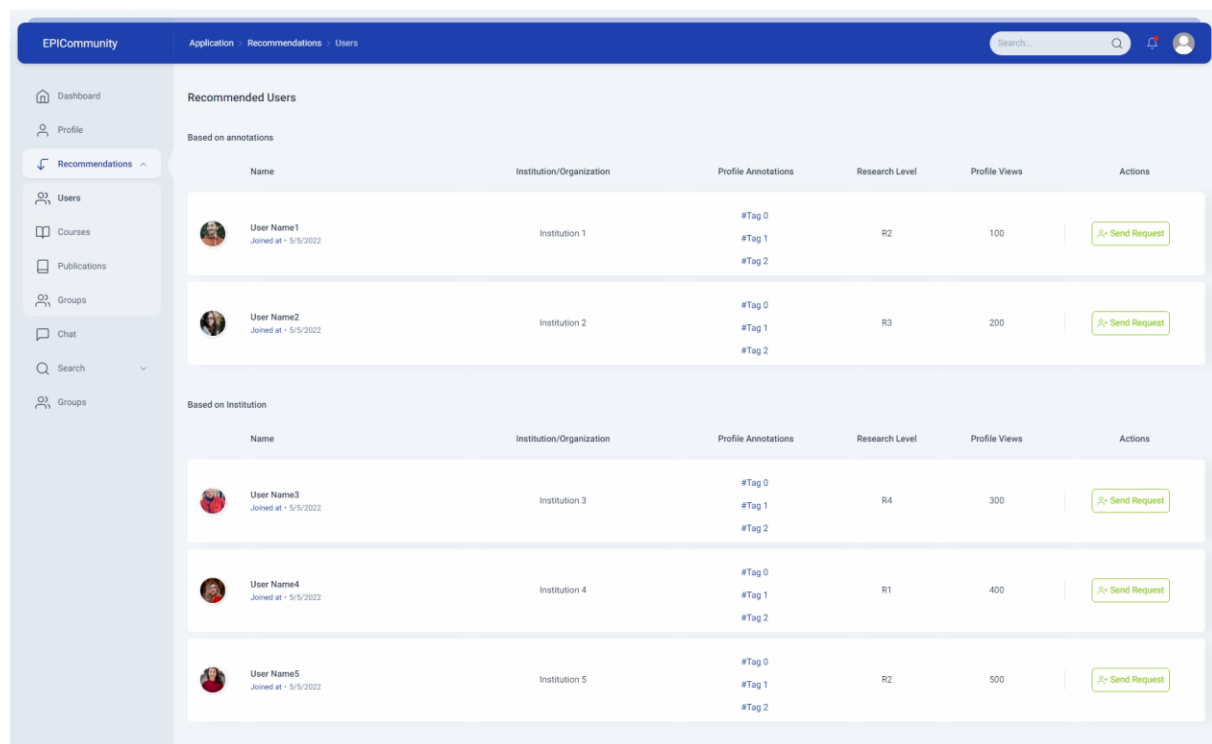


Figure 69: Recommended Users Screen

Table 78: Recommended Functionality – Users

Functionality Name	Recommended Users
Relative Use Cases	Recommended entities to the user by the platform Paragraphs: 4.2.2
<i>Actors</i>	MEMBER
<i>Input Data</i>	User ID User Annotations User Institution
<i>Description</i>	The EPICommunity recommends users for contact to a member. Based on the profile annotations and the Institution, the platform gives a list to the user for

	potential new contacts. The screen Figure 69 shows the user interface, where the two lists of recommended users are displayed (based on tags and based on institution).
<i>Output Data</i>	The platform returns two lists of recommended users.
<i>Variations</i>	-

5.7.2 Entities (Courses/Publications/Projects)

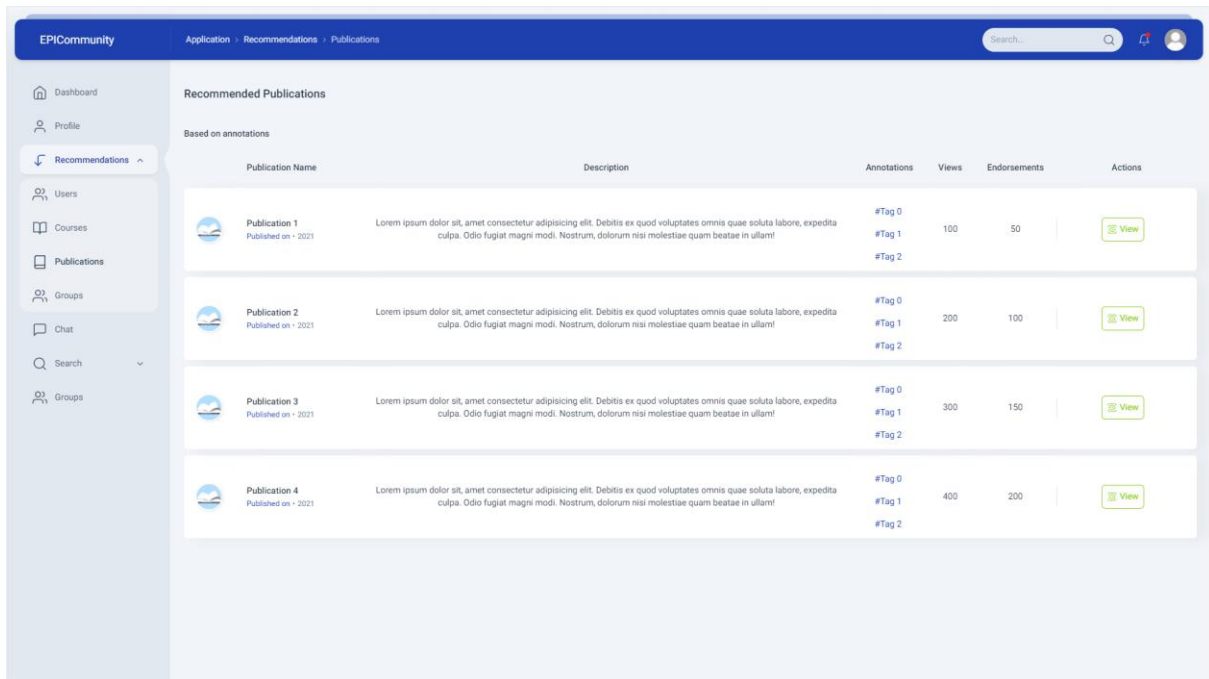


Figure 70: Recommended Publications Screen

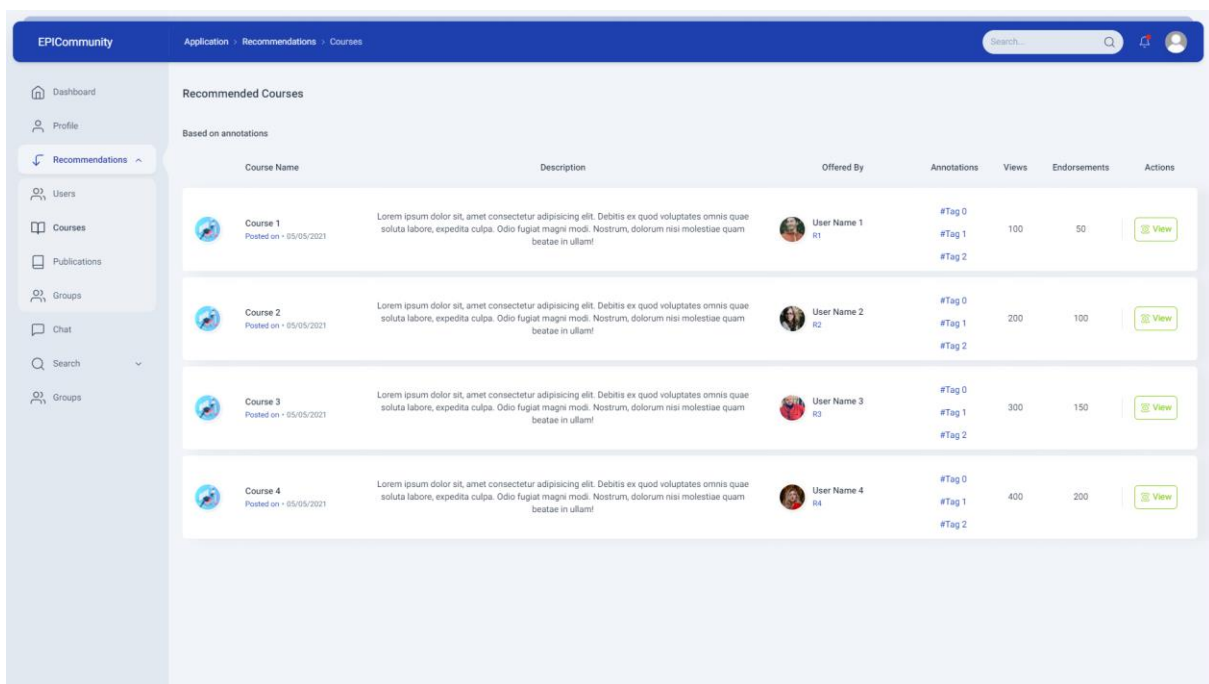


Figure 71: Recommended Courses Screen

Table 79: Recommended Functionality – Entities (Courses/Publications/Projects)

Functionality Name	Recommended Entities (Courses/Publications/Projects)
Relative Use Cases	Recommended entities to the user by the platform Paragraphs: 4.2.2
Actors	MEMBER
Input Data	<ul style="list-style-type: none"> User ID User Annotations User Institution
Description	The EPICommunity recommends instances of entities to a member. Based on the profile annotations, the platform gives a list to the user for interesting instances. The screens Figure 70, Figure 71 shows the user interfaces of recommended courses and publications.
Output Data	List of instances.
Variations	-

5.7.3 Groups

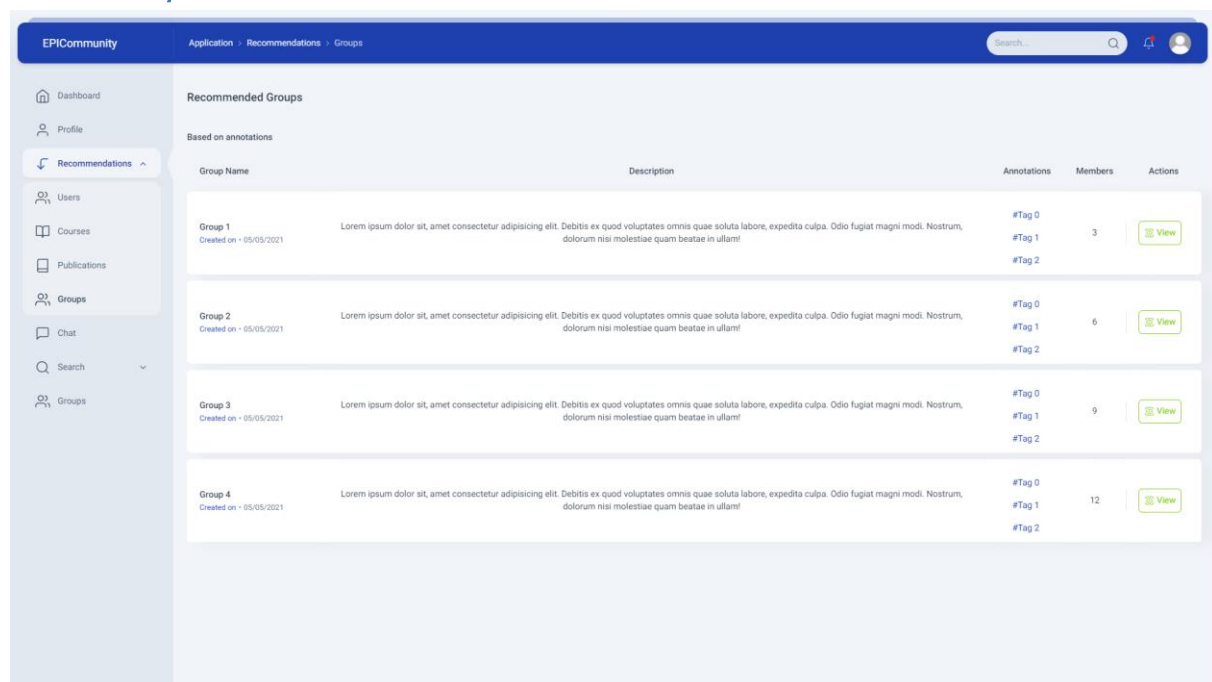


Figure 72: Recommended Groups Screen

Table 80: Recommended Functionality – Groups

Functionality Name	Recommended Groups
Relative Use Cases	Recommended entities to the user by the platform Paragraphs: 4.2.2
Actors	MEMBER

<i>Input Data</i>	-
<i>Description</i>	The EPICommunity recommends groups to a member. Based on the profile annotations, the platform gives a list to the user for interesting groups that he/she might want to become a member of. The screen Figure 72 , shows the user interface of recommended groups.
<i>Output Data</i>	The list of recommended groups.
<i>Variations</i>	-

6 Conclusion and next steps

This deliverable has presented the functionality of the EPICommunity to be developed in the next project period.

EPICommunity is a community designed by researchers for researchers!

The deliverable elaborated in collaboration with all EPICUR partners and ECR board representatives in a stepwise approach taking into account the results of the previous tasks of WP2. More specifically:

- The researcher profile has been adopted by EPIGame (Deliverable D2.2) and after the results of a state of the art of other social networks profiles
- The peer-assessment functionality is in line with the EPIQAssess – EPICUR qualitative researchers' assessment framework (Deliverable D2.1)

The main rationale of the EPICommunity design is to serve as a social network incorporating all the necessary features that can foster research and collaboration rather than be “yet another social network”. To this direction 6 groups of 16 Use Cases (see Table 81) and 7 groups of 35 functionalities has been elaborated (see Table 82).

Table 81: EPICommunity Use Cases

Paragraph	Use Case group	Number of Use Cases
4.1	Access and Registration / Login	2
4.2	Entities interaction	3
4.3	Profile Related Use Cases	3
4.4	Use Cases for the Entities Courses, Publications and Projects (Create / Read / Update / Delete)	2
4.5	Group Related Use Cases	4
4.6	Badges Use Case (Create / Read / Delete)	2

Table 82: EPICommunity Functionalities

Paragraph	Functionalities group	Number of Functionalities
5.1	Access and Registration / Login	5
5.2	Profile Functionality	6
5.3	Entities Functionality	7

Paragraph	Functionalities group	Number of Functionalities
5.4	General Functionality	5
5.5	Groups Functionality	7
5.6	Search Functionality	2
5.7	Recommendations Functionality	3

The next steps are the following:

- Continuation of the activities of Task 2.4 EPICommunity Prototype Implementation (M16-M36). Main steps of this task are the following:
 - Creation of the database and the registry of the EPICommunity: This step will create the structure of the main component of the EPICommunity, namely its database, according to the European Alliance Researcher profile, designed in Task 2.3. During this step, the Privacy Policy⁴ and “Data Protection Impact Assessment” (DPIA)⁵ will be drafted by AUTH. Relevant experts (e.g., Data Protection Officers and/or Chief Information Security Officers) will be consulted on these documents. This consultation could be organised in the context of a wider GDPR Working Group meeting, provided that a general meeting concerning GDPR issues of all EPICUR-Research WPs will be organised.
 - Development of the user interface and the functionality of the EPICommunity: This step concerns the implementation of the user interface, the creation/ customization of each module of the EPICommunity for development the alpha version of EPICommunity Prototype (M24). The alpha version of EPICommunity Prototype will be accompanied by the final version of Privacy Policy and “Data Protection Impact Assessment” (DPIA).
 - Organization of the evaluation of the alpha version of EPICommunity Prototype with the support of ECR board members.
 - Deployment of the EPICommunity platform to creates groups for the EPICluster members.
 - Continuous support and fixes of the EPICommunity to develop the final EPICommunity Prototype (M36).
- Presentation in Conferences/infodays of the EPICommunity

7 References

EPICUR-Research Deliverable D2.1 EPIQAssess – EPICUR qualitative researchers’ assessment framework

EPICUR-Research Deliverable D2.2 EPIGame – Researchers gamification framework

⁴ “Privacy Policy” document: This document is being published as a part of any application that is going to collect user’s personal data and informs them about the kind of processing, the requirements of using such user interfaces etc. “Privacy Policy” will be publicly available as a link in the EPICommunity landing page for anyone who wants to read it or as a part of an application component

⁵ “Data Protection Impact Assessment” (DPIA). DPIA is an internal document of the service provider. DPIA is going to be approved by the DPO of the hosted institution and contains information about a process designed to identify risks arising out of the processing of personal data and to minimise these risks as far and as early as possible.