PLACEMENT
OPPORTUNITIES
WITH
ENTREPRENEURIAL
REACH

Co-funded by the
Erasmus+ Programme
of the European Union

Open Source Placement Catalogue

Documentation

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

POWER project – WP2 – Online platform development plan

Executive summary

This document describes the goals and basic use cases of the online platform; a comparison between the two possible implementation approaches and the technical staff's recommendation for developing the tools described in the WP2 (Work Package 2) of the POWER project. Also contains a brief plan of the data sources and structures, software products and components to use.

Relevant project goals

The main goal of the POWER project is to give universities a tool, to provide approved placement opportunities (referred to as "PO" or "POs" from here on) for students, thereby widening the services they are offering. It's also a great opportunity to improve relations with companies on the market, opening the possibilities to other means of cooperation.

The product of WP2 is a collection of ICT tools that provide the means to POs for universities that do not yet support this process.

Main actors and use cases

The following chapter describes the main actors and their most important use cases.

1. **Companies,** as the sources of placement opportunities:
   a. Must be able to post POs for students in a chosen Institution
   b. Must be notified about any changes concerning these POs (opportunity was accepted / rejected by Institution)
   c. Should be able to modify their POs before they're accepted by an institution
   d. Have to provide the necessary information, including a contact for POs

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2. **Institutions,** as reviewers of opportunities and power users of the ICT:
    a. Must be able to register administrative users for handling POs
    b. Must be able to list POs posted by companies for the Institution
    c. Must be able to accept / reject / reopen POs
        i. By accepting, the PO will become visible for all students in the given Institution
        ii. By rejecting, the PO will not become visible for students
        iii. By reopening, the PO will become editable for the company representative again and is removed from the active offers list. If the modified PO is accepted by an Institution representative, it will be available again.
    d. Should be provided a deployable software product / application, which is reusable as is after configuration and is customizable via development if intended (done by the Institution)

3. **Students**, as browser's and target audience of POs:
    a. Must be able to search / view POs accepted by their Institution
    b. Must be able to see the contact for the POs and use their own email clients to apply

Possible system architecture approaches

In order to provide the aforementioned functionality, three ICT architectural approaches are possible. Centralized, decentralized and hybrid, each with their own advantages, disadvantages and implications.

Centralized approach

This solution only contains one platform, which consists of a database and a website. All three actors use one website for all their use cases, maintenance is done by the developers.

Advantages

- Requires the least effort from Institutions
- All aspects of the development process and data are controlled by the developer

Disadvantages

- Requires the most expensive infrastructure out of the three on the EUF side, should be scalable
- Requires the most marketing to disseminate the results and make users aware of the site
- Right after the project finishes and users should claim it, it will look like an empty site, which reduces companies' trust
- Cannot be tailored to individual Institutions' needs, not even Institution branding, institutions have the least control
- Cannot be used embedded in Institution websites

Implications

- Comfortable, but the least benefitting for Institutions
- Risk of being abandoned after project ends (infrastructure maintenance will be ownerless)
- Will least likely induce company trust, since it's not tied to individual Institutions, will help Institution branding very little

Decentralized approach

In this approach, EUF and partners deliver a deployable software package and the necessary documentation to Institutions, which they maintain on their own infrastructure and can customize according to their needs.

Advantages

- No central platform, maintenance is done by Institutions after project ends, which could prevent abandonment without funding

- Customizable for each Institution's needs, branding and special functions are addable
- Can be developed further and tailored to special needs and customs

Disadvantages

- Requires the most technical knowledge and effort from Institutions
- Installing and maintaining requires technical expertise (IT staff and some IT knowledge)
- Custom development and / or CMS management knowledge is required to customize it
- Interinstitutional cooperation will not be part of the provided software package and requires significant effort to implement from the parties involved
- Will function as a separate system (website and database) in the owner Institution's IT ecosystem, making it troublesome to incorporate into existing websites

Implications

- Challenge of creating, customizing and enhancing the software package will likely lead to very limited number of Institutions leveraging the project WP2 outcome
- Maintaining the platforms for Institutions will require continuous effort and insufficient care could even cause security risks
- Regular maintenance, customization and mindful enhancement could lead to a refined and effective tool in individual institution's ecosystem
- Limited support capacity of the software provider might lead to dissatisfaction

Hybrid approach

Advantages

- Requires less effort from Institutions than the decentralized approach
- Some customization is available and easy, like individual branding
- Integration to existing Institution IT infrastructure is more flexible

- Can likely leverage other ICT resources already in development, beta or released stage, like the Institutions API, and external identity providers
- The most state of the art of the proposed solutions
- Payload is shared between different systems, making it lighter weight on main components
- With know-how and available development capacity, PO data from the POWER API can be leveraged in ways other than using the provided application, giving freedom to the Institution

Disadvantages

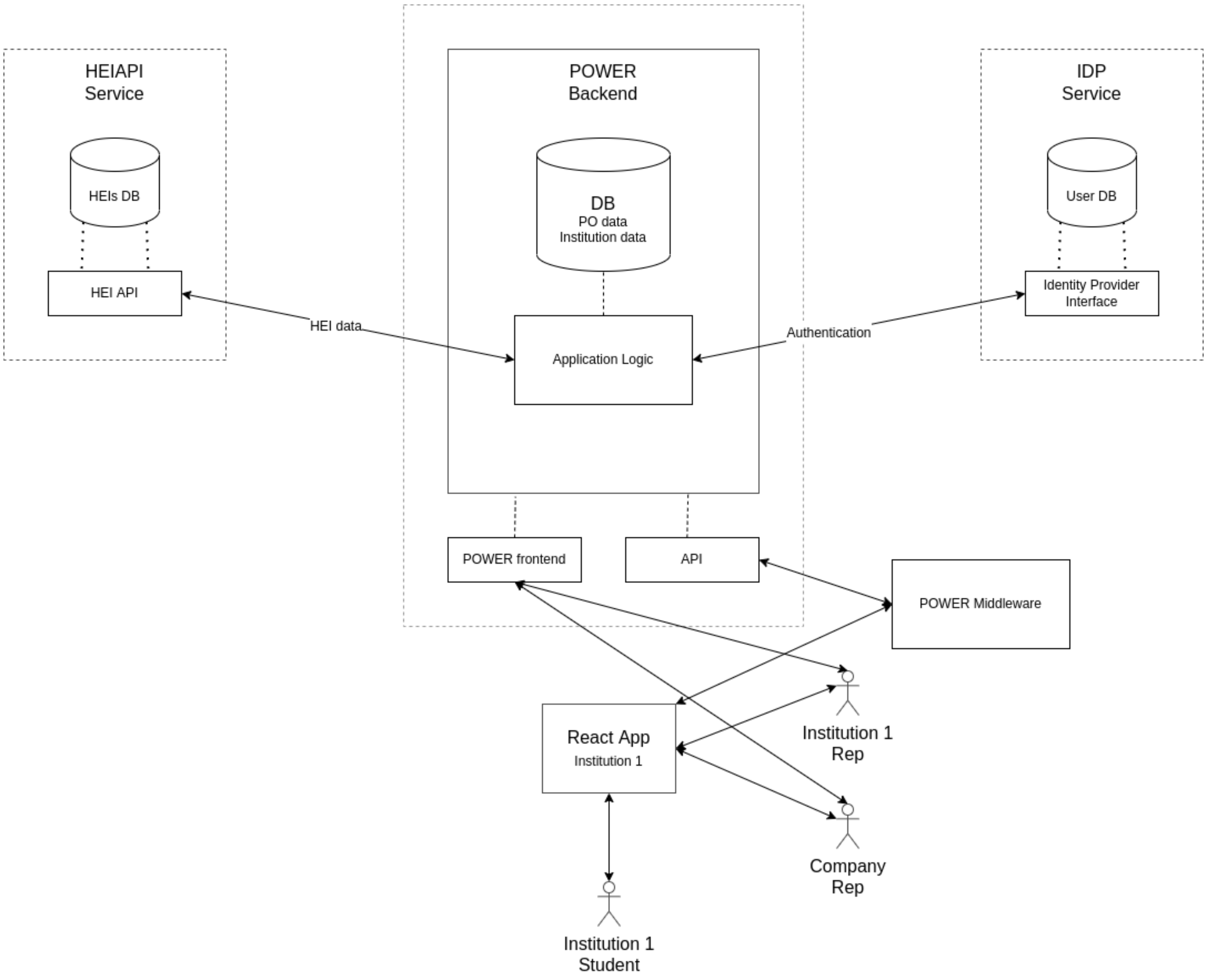
- Development has to be done on four environments (Drupal, React, Slim PHP and NodeJS), likely resulting in more man hours
- Customizing possibilities are limited compared to decentralized approach, but takes less effort from the Institution side
- Some maintenance work and support of software product falls on the development team, raises abandonment issues after project ends

Implications

- Has the largest chance of being used by Institutions, since customization is possible, deployment requires less effort and knowledge than decentralized approach
- Institutions with the required knowledge and allocatable resources could develop their own software solutions, using the data provided by the backend

After comparing the three possible approaches, the hybrid architecture will be used, since it has the most advantages and requires less compromises than the other two models.

Network diagram



POWER frontend (using Drupal)

POWER Frontend is a Drupal site, which provides user interfaces 1 and 2 described below. Is separated from the React App, see details of the latter below.

User and password credentials for the site:
Username: test_user
Password: liL%tWZG0p42

Site URL: https://power.uni-foundation.eu

POWER backend

The POs manipulated (created, accepted, modified) via UIs 1 and 2 are exposed via REST endpoints by Drupal. The React app is connecting to these API endpoints through the middleware, also delivered as a project deliverable.

Authentication from the React App

Each Institution is going to have their own API user, that is created by the maintainers. These API users are given an API key, that has to be added to a middleware configuration file.

It has to be identifiable which Institution an API user and an Institution reviewer is affiliated with, therefore all user entities have a field value that contains their Institution's SCHAC code.

Authentication is done with an API key that's assigned to the API user on creation.

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Authorization

Authorization is done by the backend and happens based on two factors (indirectly three).

SCHAC code value saved in the user entity's corresponding field

- The PO endpoints are Drupal View REST exports
- The filters on these API endpoints are dynamically set, according to the SCHAC code saved with the user (and API user), thereby identifying which Institution the user represents

Authenticated role for the user in Drupal

- The API users role controls overall access to the PO endpoints
- Technically the permission required to access the endpoints is `'view any webform submission'`
- Giving this permission to users is done giving them the "API user" role

POWER middleware

For the convenience of Institutions, the middleware is provided in two technology stacks. Both pieces of software are provided on Github, a common version handling platform used by developers. If an Institution wants to join the project, their github accounts will be granted access to these repositories.

- Link to the SlimPHP based middleware
- Link to the NodeJS based middleware

Access to the middleware endpoints should be restricted via server rules, these endpoints should not be accessible by the public, only by the react app.

Note: In the setup provided for testing purposes, the middleware endpoints are not restricted, in order to be testable.

Middleware API Endpoints

Public POs endpoint

Exposes all the POs, that are public, meaning, they are not directed at a specific Institution, but are available to all. Only vacant and public positions are shown here.

Request details

URI: https://slim.power.uni-foundation.eu/power-middleware/rest/public-pos

Method: GET

Params: None

Response example:

```
[
    {
     "nid": "33",
     "field_company_name": "European Company Ltd.",
     "field_company_website": "https://eucompany.eu",
     "field_contact_first_name": "Emma",
     "field_contact_last_name": "Petite",
     "field_contact_email": "emma.petite@eucompany.eu",
     "field_contact_phone_no": "+33 6 12 34 56 78",
     "field_job_country": "FR",
     "field_job_city": "Paris",
     "field_job_application_deadline": "2022-09-30",
     "field_job_category": "Business and Finance",
     "field_job_title": "Accountant trainee",
     "field_job_duration": "6 month",
     "field_job_description": "<p>Trainee accountant job for a multinational, multicultural company.</p>\n",
     "field_job_commitment": "Full time",
     "field_job_compensation": "Full compensation",
     "field_target_picker": "Public",
     "field_target_institution": "",
```

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
    "field_job_responsibilities": "<ul><li>Work alongside senior accountants to keep company finances transparent</li>\n<li>Create reports for Project Managers</li>\n</ul>",
    "field_applicant_required_skills": "<ul><li>Great communication skills</li>\n<li>Great assessment and problem solving skills</li>\n<li>Bachelor's degree in Finance or Accounting</li>\n</ul>",
    "field_how_to_apply": "<p>Send CV and cover letter to emma.petite@eucompany.eu.</p>\n",
    "field_placement_status": "Vacant"
  }

]
```

Institution POs endpoint

Exposes all POs directed to an institution. According to the API users SCHAC code stored in the backend database, this endpoint dynamically sets the filtering of the POs so only the ones belonging to an Institution are available.

Request details

URI: https://slim.power.uni-foundation.eu/power-middleware/rest/institution-pos

Method: GET

Params: None

Response example

```
[
   {
    "nid": "32",
    "field_company_name": "European Company Ltd.",
    "field_company_website": "https://eucompany.eu",
    "field_contact_first_name": "Johann",
    "field_contact_last_name": "Müller",
    "field_contact_email": "johann.muller@eucompany.eu",
    "field_contact_phone_no": "+49 1512 3456789",
    "field_job_country": "DE",
    "field_job_city": "München",
    "field_job_application_deadline": "2022-09-30",
    "field_job_category": "Education and Training",
    "field_job_title": "Coach Trainee",
    "field_job_duration": "6 month",
    "field_job_description": "<p>Coaching of Company personnel in Assertive communication, negotiation</p>\n",
    "field_job_commitment": "Full time",
    "field_job_compensation": "Limited compensation",
    "field_target_picker": "One Institution",
    "field_target_institution": "Eotvos Lorand Tudomanyegyetem",
    "field_job_responsibilities": "<ul><li>Visit trainings held by senior coaches</li>\n<li>Learn coaching methods</li>\n<li>Eventually organize trainings individually</li>\n</ul>",
    "field_applicant_required_skills": "<ul><li>Great communication skills</li>\n<li>Fluent in English and German</li>\n<li>Structured problem solver</li>\n<li>Bachelor's degree in Psychology</li>\n</ul>",
    "field_how_to_apply": "<p>Send your CV and cover letter to johann.muller@eucompany.eu</p>\n",
    "field_placement_status": "Vacant"
  }
]
```

React App

The React app should be embedded into Institutions' own websites. This requires limited IT work on the Institution side.

Just like in the middleware's case, the github repository will be made available for Institutions requesting to use the POWER platform.

The React Application set up for testing purposes can be found on the following link: https://app.power.uni-foundation.eu/

Technical requirements for the React App

- Has to be embeddable into Institution websites
- With proper React knowledge, the Institution is free to adjust anything in the App

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

User interfaces

1. **POWER administration user interface**

   Part of the POWER frontend. Contains the full functionality of the Drupal administration interface.

2. **POWER non-administrator user interface**

   Part of the POWER frontend. Contains the functionality:

   - used by company representatives (post POs, optionally review and modify ones posted by them)
   - used by Institution representatives to handle PO related functionality

3. **Institution specific React app**

   The React app. Contains the functionality:

   - used by students to browse, search POs
   - can also be used by institution representatives to view accepted POs if they have access to the university website

Users, actions, permissions

This section defines the user types and the (user) interfaces they are going to connect to.

User types

1. **Administrators** (EUF and / or partners)
   a. Application manager(s): Contact person(s) for handling data and everyday issues, that doesn't require the technical staff's intervention

   b. System administrators, developers: Supports the application manager in issues that require technical work

2. **Company representatives**
   a. HR representative**:** Managers of posted POs
3. **Institution representatives**
   a. Reviewer, approver: Has approval rights for POs posed for the Institution
4. **Students:** Can browse POs for the attended Institution and contact companies via email

Permissions and actions

1. **System administration**: Full control, can change data structure and data , functions and operations.
   - Owned by: 1/b
   - Actions: All, but only interferes with data in case of an error
2. **Data administration**: Permission to see, modify, change approval status, delete POs for all the Institutions. Can see company data related to any PO
   - Owned by: 1/a
   - Actions: Can modify PO data and manage website through the user interface if necessary
   - User interfaces: POWER frontend, React app
3. **Post PO**: Can post POs and browse POs posted

   Owned by: 2/a, 3/a

   User interfaces: POWER frontend, React app
4. **Review POs**: Can accept / reject / forward (?) POs posted

   Owned by: 3/a

   User interfaces: POWER frontend, React app (?)
5. **View POs**: Can view and search in POs posted for a specific Institution

   Owned by: 3/a, 3/b, 4

   User interfaces: React app
6. **Modify POs:** Can change PO data posted by the company representative and complete a variety of actions.

Owned by: 2/a

User interfaces: POWER frontend. Company representatives are provided with links to complete the following actions:

- Can change data entered when posting PO
- Can delete the posted PO
- Can mark the PO as taken (meaning the opportunity is no longer available)

Authentication

1. Institution users are authenticated with an IdP service or have an account registered directly on the site.
2. Company representatives are not authenticated, however checks are performed, to prevent fraud. Tests include:
    - Cross checking IP with location of PO
    - Checking contact email to avoid common free mailing providers (gmail, yahoo, hotmail)

Data

Types of data by content

Company data

Source: POs submitted by company reps

Storage type: Webform submissions

Institution data

Source: Fetched from the HEIAPI

Storage type: Necessary data is stored in the POWER backend's database

Placement Opportunity data

Source: Submitted by company reps through webforms

Storage type:

- Before approval: Webforms submissions
- After approval: Saved to entity or custom content type in POWER backend database

Fields (attributes):

Automatically filled:

- Id (auto)
- Created (auto)
- title (text)
- Posted_on (auto)

Company filled:

- Company information:
  - Company name (text)
  - Company website (url)
- Contact information:
  - First name (text)
  - Last name (text)
  - Contact phone (validated telephone number)
  - Contact email (email)
- Publicity information
  - Opportunity visibility: Public or directed to an Institution (list: Public / To specific Institution)
  - For Institution: If Specific to an Institution, picklist of Institutions registered to the site is shown (list)
- Job information:

  - Job category (list)
  - Job title (text)
  - Application deadline (date)
  - Country (list)
  - City (text)
  - Duration in months (number)
  - Expandable (Yes / No)
  - Job description (text area with editor)
  - Responsibilities (text area with editor)
  - Commitment (list with an option for free typed data)
  - Compensation (list with an option for free typed data)
  - Job requirements – knowledge, skills, talents, personality (text area with editor)
- Applicant information:
  - Required skills (text area with editor)
  - How to apply (text area with editor)

User data

Source:

- Registration on the POWER site for Institution personnel. Requires a check and confirmation by POWER staff.
- Possibility is open to use an IdP later. Could be the IdP maintained by EUF or MyAcademicID.

Storage type: Stored in POWER backend database

Fields (standard OpenID claims stored)

- Default user fields (Username, Email)
- Institution SCHAC code from registration or potentially IdP
- Personal SCHAC from registration or potentially from IdP

Use cases

For actor: Company representative

UC_CR_01

Use Case 1	Post PO
Actor	Company representative
Description	A company rep adds a job / traineeship opportunity to the POWER database
Preconditions	None
Postconditions	- User gets redirected to a confirmation page with the edit and view links to the PO posted
Basic flow	- Company rep goes to the website's post PO page - Enters PO data - Submits the form - PO is checked for signs of fraud, if suspicious, gets flagged
Termination outcome (basic)	- PO is checked for signs of fraud (if suspicious, gets flagged) - PO is saved as a webform submission - Company rep gets notified via the contact email provided, that the PO has been posted - Institution rep gets notified, that a PO has been posted for the Institution

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Alternative flow 1	PO data entered is invalid
Termination outcome 1	- User is returned to the form - Form shows missing or invalid data entries to be corrected
Alternative flow 2	Company rep cancels posting PO (quits or cancels before submitting)
Termination outcome 2	PO doesn't get submitted

UC_CR_02

Use Case 2	Delete posted PO
Actor	Company representative
Description	A company rep removes an already submitted PO
Preconditions	- PO was submitted before - User has the notification email sent when PO was submitted
Basic flow	- User clicks the delete PO link in the confirmation email - Confirms PO removal
Termination outcome (basic)	- PO is removed from the submissions - PO is removed from the active opportunities list
Alternative flow 1	PO has already been accepted by Institution
Termination outcome 1	- PO is removed from the accepted POs

	- Company rep and Institution rep is notified via email, that the PO has been removed

UC_CR_03

Use Case 3	Mark PO as taken
Actor	Company representative
Description	The company rep wants to flag the PO as taken, because someone was hired
Preconditions	- PO was posted - Company rep has the notification email about the PO being posted or accepted
Postconditions	
Basic flow	- Company rep opens the notification email it received when the PO was accepted by an Institution - Clicks the link provided to mark the PO as no longer available
Termination outcome (basic)	The PO is marked as no longer available and is displayed accordingly on the sites
Alternative flow 1	- Company rep doesn't have the PO acceptance notification email - Requests the email to be sent again
Termination outcome 1	Acceptance email is sent again

UC_CR_04

Use Case 4	Modify PO data
Actor	Company representative
Description	The company rep wants to modify the PO posted
Preconditions	- PO was posted - Company rep has the notification email about the PO - PO was not accepted by an Institution representative yet
Postconditions	
Basic flow	- Company rep opens the notification email received - Clicks the link provided to edit the PO - Edits PO data
Termination outcome (basic)	PO data is changed
Alternative flow 1	- Company rep doesn't have the PO acceptance notification email - Requests the email to be sent again
Termination outcome 1	Acceptance email is sent again

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

For actor: Institution representative

UC_IR_01a

Use Case 1a	User registration (non-IdP user)
Actor	Institution representative
Description	User wants to register
Preconditions	User is not registered yet
Postconditions	User can log in
Basic flow	- Actor requests to have a registered user as advised on the front page and the help page of the POWER site - POWER staff reviews if they're eligible for registration
Termination outcome (basic)	- User is registered - User data is inserted into the database
Alternative flow 1	User data is not legitimate
Termination outcome 1	User registration is rejected by POWER staff

UC_IR_01b

Use Case 1b	User registration (via EUF IdP)
Actor	Institution representative

Description	User logs in with EUF IdP service for the first time
Preconditions	User is not registered yet
Postconditions	User can log in
Basic flow	- Actor initiates login with the EUF IdP service - POWER staff reviews registration
Termination outcome (basic)	- User gets registered - User data is inserted into the database
Alternative flow 1	IdP service not available or fails to authenticate
Termination outcome 1	User gets an alert containing the problem

UC_IR_02a

Use Case 2a	User login (non-IdP)
Actor	Institution representative
Description	User accesses the login page and wants to log in
Preconditions	- User is not logged in
Postconditions	
Basic flow	- User goes to login page - Enters credentials - Clicks login
Termination outcome	- User is logged in

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

(basic)	- User gets directed to it's profile or to the page it wanted to reach originally
Alternative flow 1	Wrong credentials were entered
Termination outcome 1	User gets warning, that username / password is not correct

UC_IR_02b

Use Case 2b	User login (using IdP)
Actor	Institution representative
Description	User accesses the login page and wants to log in via EUF IdP service
Preconditions	- User is not logged in
Postconditions	
Basic flow	- User goes to login page - Clicks login via EUF IdP
Termination outcome (basic)	- User is logged in - User gets directed to it's profile or to the page it wanted to reach originally
Alternative flow 1	User is not registered yet - User gets registered via IdP (=UC_IR_01b)
Termination outcome 1	UC_IR_01b starts

UC_IR_03

Use Case 3	View submitted, but non-accepted POs
Actor	Institution representative
Description	User views submitted PO(s) for its own organization
Preconditions	- User is logged in - At least one non-reviewed PO exists for the Institution
Postconditions	
Basic flow	- User requests to see non-reviewed POs
Termination outcome (basic)	- Gets a searchable list of non-reviewed POs for its Institution
Alternative flow 1	User is not logged in
Termination outcome 1	Gets redirected to the login page

UC_IR_04

Use Case 4	Change review state of submitted POs
Actor	Institution representative
Description	User wants to accept / reject a submitted PO
Preconditions	- User is logged in - At least one non-reviewed PO exists for the Institution

Postconditions	
Basic flow	- Requests to see non-reviewed POs - Opens the details of one non-reviewed PO - Chooses to accept / reject PO
Termination outcome (basic)	Review state of the chosen PO is changed to: - Accepted - PO will be saved as custom content - PO will be visible for designated students - PO will be locked, no further data modification will be available - Rejected - PO submission will be flagged as rejected PO poster gets notified of the review state change

UC_IR_05

Use Case 5	Reopen PO
Actor	Institution representative
Description	User wants to reopen a PO for editing and redo review
Preconditions	- User is logged in - The PO was reviewed and accepted previously and is now locked for editing
Postconditions	

Basic flow	- User navigates to the POs list - Selects the accepted and locked PO and unlocks it
Termination outcome (basic)	PO is: - Removed from the accepted opportunities list - Is open for modification by the company representative or the institution representative - PO's review status is set from accepted to pending PO poster gets notified of the review state change

UC_IR_06

Use Case 6	Add Institution specific data / comments to PO
Actor	Institution representative
Description	Institution rep wants to add comments or additional information to the PO
Preconditions	- User is logged in - At least one reviewed PO exists for the Institution
Postconditions	
Basic flow	- Requests to see reviewed POs - Opens the details of one reviewed PO - Adds Institution comments and special info on form - Saves the PO

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Termination outcome (basic)	- Institution comments are saved and will be visible to students after PO is accepted

For actor: Student of an Institution

These use cases are going to be available in the React App, that is running on Institutions' websites

UC_ST_01

Use Case 1	View PO entries
Actor	Student of an Institution
Description	A student wants to see the full list of POs accepted by their Institution
Preconditions	Student is visiting the university website's PO page
Postconditions	
Basic flow	- Student requests to see the POs posted for its organization
Termination outcome (basic)	- Student gets a list of POs - When a PO is chosen, gets a detailed view
Alternative flow 1	No PO is posted for the Institution
Termination outcome 1	App shows a message, stating the fact

Code: **UC_ST_01a**

Use Case 1a	Filter listed PO entries
Actor	Student of an Institution
Description	A student wants to see the filtered list of POs accepted by their Institution
Preconditions	Student is on the PO list page
Postconditions	
Basic flow	- Student requests to see the POs posted for its organization - Student picks a field to filter for and enters a value ()
Termination outcome (basic)	- Student gets a filtered list of POs - When a PO is chosen, gets a detailed view
Alternative flow 1	There's no PO posted that matches the filter criteria
Termination outcome 1	App shows a message, stating the fact

Code: **UC_ST_02**

Use Case 2	Get a detailed view of PO data
Actor	Student of an Institution
Description	A student wants to see all the details of a PO

Preconditions	Student has access to university website and POWER app is embedded
Postconditions	
Basic flow	- Student requests to see the POs posted for its organization - Student picks a PO and clicks details
Termination outcome (basic)	- Student gets the details of a PO on the UI

Code: **UC_ST_03**

Use Case 3	Apply for a PO
Actor	Student of an Institution
Description	A student wants to apply for a posted job / traineeship
Preconditions	Student has an email address
Postconditions	
Basic flow	- Student requests to see the POs posted for its organization - Student picks a PO and clicks apply - Student writes an email for the company rep's email address given in the posting using its own email client

Termination outcome (basic)	- Company rep gets the email sent by the student

Disclaimer: This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.