

# Registration site webapp

## Summary

The goal is to develop a web application that supports the maintenance of a registry of users. The following registration workflow has to be supported:

- A user submits their initial registration by using one of the registration pages of the application
- A user receives notification emails about changes regarding their registration status, including the initial registration request
- A user must be able to login to the system either while their registration or modifications are pending, or when approved, and check/modify their data
- the admin user checks the data the user submitted or modified, and decides on how to proceed:
  - Reject the application (the decision is final, no further actions are possible)
  - Approve the application/modification (the user becomes a registered user)
  - Asks for modification/supplement of user data (application is still pending)

## Tasks

- Create a state diagram of the user registration workflow
- Implement the frontend
  1. Create home page with links to *Login*, *Registration as visitor*, and *Registration as client* pages.
  2. Create registration page for two different types of users:
    - **Visitor:** attributes are
      - A. ORCID id (optional, if user fill in this field, then other field values are to be filled in from ORCID)
      - B. e-mail address
      - C. name
      - D. phone number
    - **Client:** attributes are
      - A. e-mail address
      - B. name
      - C. phone number
      - D. company name
      - E. country of origin
  3. Create Login page.
  4. Create page for users:
    - Overview information with user attributes and registration status.
    - Profile viewer and updater pages, including password change.
  5. Create admin pages for the admin user:
    - Overview page of users filtered by registration status.
    - Admin page where individual users are handled according to the workflow. The admin user has to be able to write a comment to any user attribute. These comments must be included in the emails sent to the user.
- Implement the backend

1. Define a database schema for *attributes* of user registrations (e.g. e-mail address, name), *user types*, and *information of registration process* (registration status, modification requests).
2. Create workflow for the registration process
3. Create a user notification subsystem which sends e-mail notification when registration state is changed.
4. Non-admin users must not have access to information of any other user.

## Constraints/preferences

- Required technologies/services:
  - Python 3.x, Django, Bootstrap
  - ORCID OAuth API (<https://info.orcid.org/documentation/api-tutorials/api-tutorial-get-and-authenticated-orcid-id/>)
- Preferred technologies/services:
  - Docker
  - MailDev server ( <https://hub.docker.com/r/maildev/maildev> )
- Linux operating system is preferred both as a development and target environment.

## Background information

- Any online or offline documentation can be used.
- We provide and encourage continuous consultation during the assignment. Please feel free to contact us at [srd.soft-eng@eli-alps.hu](mailto:srd.soft-eng@eli-alps.hu) at any stage of the development.