

# Architecture Specification and Prototype (EAP Component)

---

Databases and Web Applications Laboratory (LBAW)  
Bachelor in Informatics Engineering and Computation (L.EIC)

Sérgio Nunes  
Dept. Informatics Engineering  
FEUP · U.Porto

# Architecture Specification and Prototype (EAP) Component

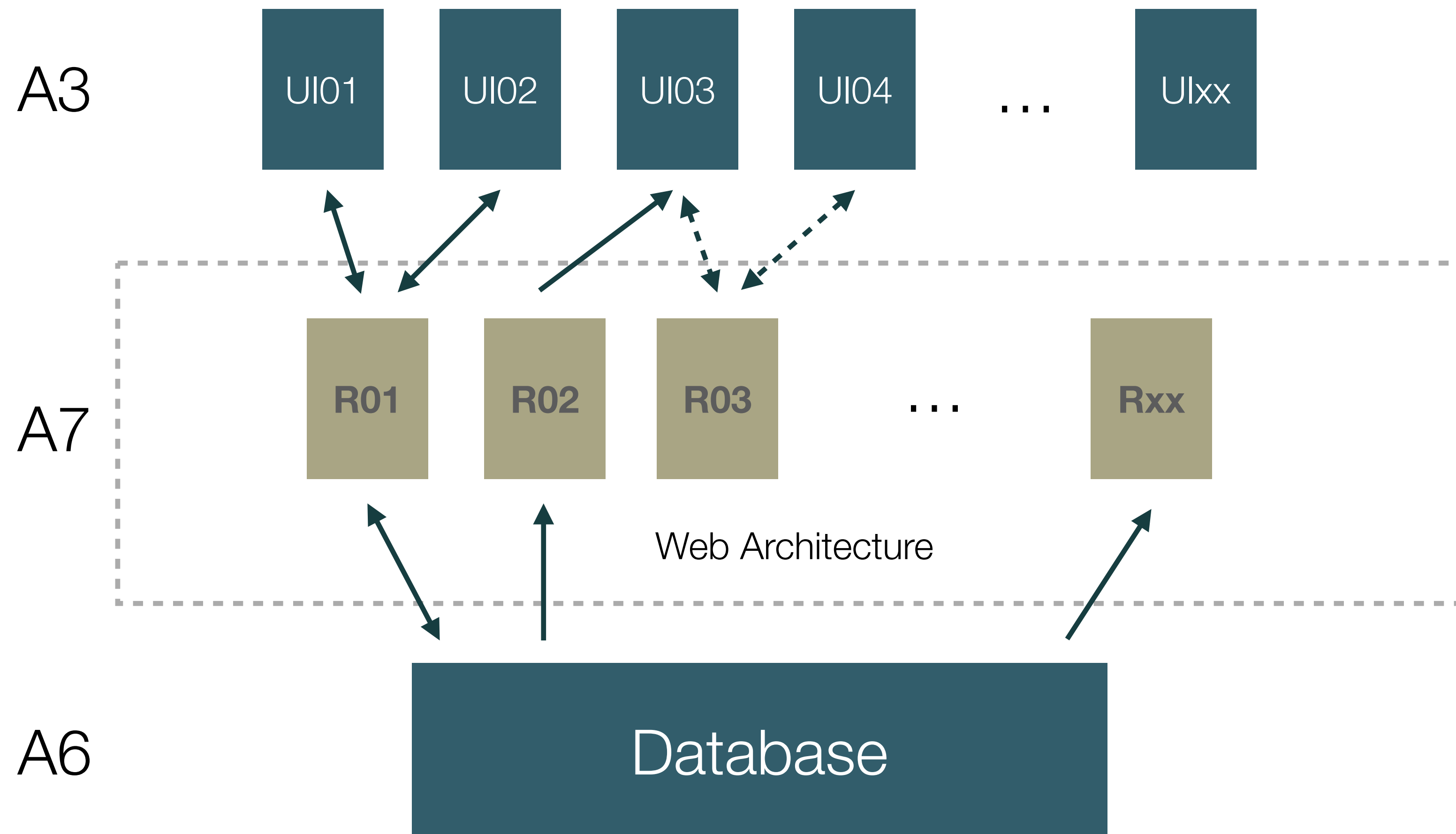
# EAP: Architecture Specification and Prototype

---

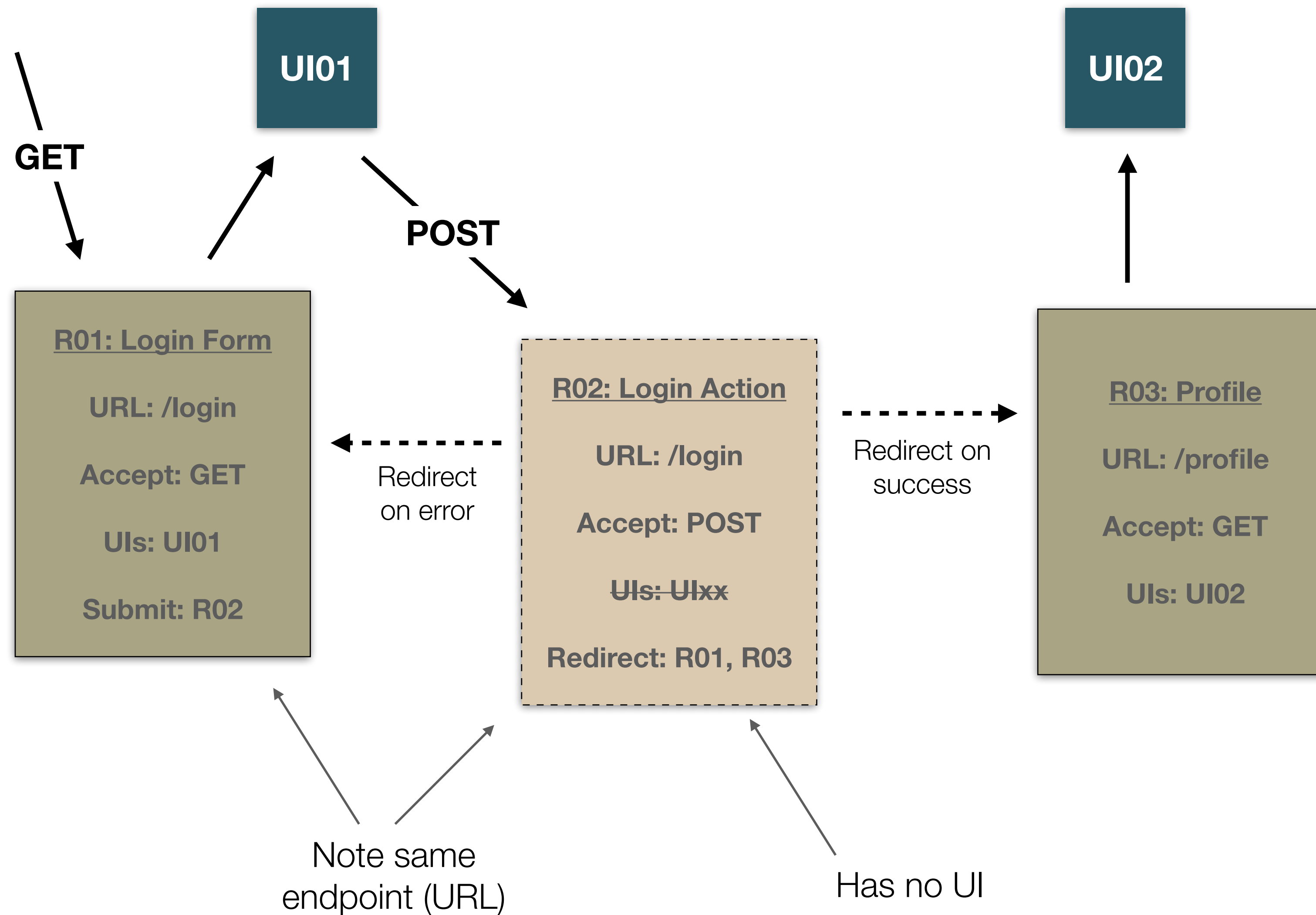
- Decide on the web architecture to support the previously defined views (A3).
- What is going to be the mapping between views and requests.
- Two (extreme) examples:
  - In a document-oriented web site, each view corresponds to a resource, i.e. we have each resource producing each view (1-1 mapping).
  - In single page web application (SPA), only one document is output by the server, i.e. we have 1 resource producing N views (1-N mapping).

# A7: Web Architecture

---



# A7: View Action Pattern



# A7: Web Architecture

# A7: Web Architecture

---

- Elements of the A7 artifact:
  - ... presents an **overview of the web resources to implement**.
  - ... establishes the **conditions of access to resources (permissions)**.
  - ... defines a **catalogue of web resources**, including:
    - expected URL;
    - HTTP methods and parameters supported;
    - possible responses;
    - for resources with a user interface, a reference to the UI page (see A3 sitemap) is included.

# A7: MediaLibrary - Overview

---

## 1. Overview

*An overview of the web application to implement is presented in this section, where the modules are identified and briefly described. The web resources associated with each module are detailed in the individual documentation of each module inside the OpenAPI specification.*

<b>M01: Authentication and Individual Profile</b>	Web resources associated with user authentication and individual profile management. Includes the following system features: login/logout, registration, credential recovery, view and edit personal profile information.
<b>M02: Works</b>	Web resources associated with library items. Includes the following system features: items list and search, view and edit item details, and delete items.
<b>M03: Reviews and Wishlist</b>	Web resources associated with reviews and wishlist. Includes the following system features: add review, list reviews and delete reviews; add items to wish list and remove items from the wish list.
<b>M04: Loans</b>	Web resources associated with loans. Includes the following system features: view loans, add loans and edit loans.
<b>M05: User Administration and Static pages</b>	Web resources associated with user management, specifically: view and search users, delete or block user accounts, view and change user information, and view system access details for each user. Web resources with static content are associated with this module: dashboard, about, contact, services and faq.



# A7: MediaLibrary - Permissions

---

## 2. Permissions

*This section defines the permissions used in the modules to establish the conditions of access to resources.*

<b>PUB</b>	Public	Users without privileges
<b>USR</b>	User	Authenticated users
<b>OWN</b>	Owner	User that are owners of the information (e.g. own profile, own items)
<b>ADM</b>	Administrator	System administrators

# A7: MediaLibrary - OpenAPI Specification

---

## 3. OpenAPI Specification

*This section includes the complete API specification in OpenAPI (YAML).*

*Additionally there is a link to the OpenAPI YAML file in the group's repository. The filename should include 'openapi' to active GitLab's OpenAPI viewer.*

```
openapi: 3.0.0

info:
  version: '1.0'
  title: 'LBAW MediaLibrary Web API'
  description: 'Web Resources Specification (A7) for MediaLibrary'

servers:
- url: http://lbaw.fe.up.pt
  description: Production server

externalDocs:
  description: Find more info here.
  url: https://web.fe.up.pt/~ssn/wiki/teach/lbaw/medialib/a07

tags:
```

## A8: Vertical Prototype

# A8: Vertical Prototype

---

- The A8 artifact:
  - corresponds to the implementation of the high priority user stories (features marked with asterisks);
  - is used to validate the architecture presented;
  - also to gain familiarity with the technologies used in the project.
- It must:
  - be based on the LBAW Framework and
  - include work on all layers of the architecture: user interface, business logic and data access.
- The implemented user stories must be specific of the group's project.  
Adapting the provided authentication system is required but not considered part of those US.
- Implemented user stories must include at least a form, an action, an AJAX request, search, and update to the database.

## A8: Vertical Prototype

---

- LBAW framework stack: PostgreSQL and Laravel.
  - Laravel setup on the local machine;
  - PostgreSQL and pgadmin as Docker containers (same);
- Deployment using GitLab's Container Registry.
- More details next lecture.