eap

Last edited by **Gonçalo Cunha Marques** 2 months ago

EAP: Architecture Specification and Prototype

Planora is a flexible and efficient project management platform that empowers individuals and teams to organize, track, and complete their projects with ease. Designed with user-friendly navigation and a responsive interface, it enables seamless collaboration and task management across all devices.

A7: Web Resources Specification

This artifact documents the architecture of the Planora web application, outlining the resource catalog, their properties, and the format of the JSON responses. The specification adheres to the OpenAPI standard and is presented in YAML format.

Additionally, the document describes the CRUD (Create, Read, Update, Delete) operations for all resources implemented in the vertical prototype.

1. Overview

This section offers an overview of the Planora web application to be developed, highlighting the key modules and providing a brief description of their functionality.

Identifier	Description
M01: Authentication and Profile	Resources assigned to authentication of individual profile. Including : edit, create, login/logout, register, view.
M02: Projects	Web resources associated with Projects and their tasks. Features : create a project , projects, view personal projects , adding a user to a project.
M03: Tasks	Web resources associated with tasks of a project. Features : creating task for a project, edit tasks that the user have created, view tasks of own project, deleting tasks.
M04: Search	Web resources associated with searching users, tasks and projects. Features: exact match search, search suggestions.
M05: Admin	Web resources related to user management, if the user is an admin. Search users and edit their profiles.

2. Permissions

The permissions within the modules are designed to manage and limit access to resources, ensuring appropriate authorization and conditional usage based on user roles and privileges.

Identifier	Name	Description
PUB	Public	Users without privileges
USR	User	Authenticated users
ADM	Administrator	System administrators
OWN	Owner	User that are owner of a project
PRM	Project Member	Users that are part of a project
PRC	Project Coordinator	Project Members that can create task

3. OpenAPI Specification

Section that includes the API specification in OpenAPI (YAML) for the Planora prototype. OpenAPI in repository:

```yaml
openapi: 3.0.0

info:
  version: '1.0'
  title: 'Planora'
  description: 'Web Resources Specification (A7) for Planora'

servers:
  - url: http://localhost:8000
```

```
    description: Localhost server

tags:
  - name: 'M01: Authentication and Profile'
  - name: 'M02: Projects'
  - name: 'M03: Tasks'
  - name: 'M04: Search'
  - name: 'M05: Admin'

paths:

  /login:
    get:
      operationId: R101
      summary: 'Show Login Form'
      description: 'Returns the login form for users to authenticate. PUB'
      tags:
        - 'M01: Authentication and Profile'
      responses:
        '200':
          description: 'Successful response. Displays the login form.'
    post:
      operationId: R102
      summary: 'Authenticate User'
      description: 'Processes the login form submission and authenticates the user. Redirects on success or failure. PU
      tags:
        - 'M01: Authentication and Profile'

      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                email:
                  type: string
                password:
                  type: string
              required:
                - email
                - password

      responses:
        '302':
          description: 'Redirect after processing the login credentials.'
          headers:
            Location:
              schema:
                type: string
                example:
                  302Success:
                    description: 'Successful authentication. Redirect to user dashboard.'
                    value: '/projects'
                  302Error:
                    description: 'Failed authentication. Redirect to login form.'
                    value: '/login'

  /logout:

    post:
      operationId: R103
      summary: 'Logout Action'
      description: 'Logs out the currently authenticated user and invalidates the session. Redirects to the login page.
      tags:
        - 'M01: Authentication and Profile'
      responses:
        '302':
          description: 'Redirect after processing logout.'
          headers:
            Location:
              schema:
                type: string
                example:
                  302Success:
                    description: 'Successful logout. Redirect to login form.'
                    value: '/login'
                  302Error:
                    description: 'Logout failed. Redirect to projects page.'
                    value: '/projects'

  /register:
    get:
      operationId: R104
      summary: 'Show Register Form'
      description: 'Returns the new user registration form. No authentication required. Access: PUB'
      tags:
        - 'M01: Authentication and Profile'
      responses:
        '200':
          description: 'Successful response. Displays the sign-up form.'

    post:
      operationId: R105
      summary: 'Register User'
      description: 'Processes the new user registration form submission, creates a new user, and logs them in. Access: 
      tags:
        - 'M01: Authentication and Profile'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                username:
                  type: string
                email:
                  type: string
                password:
                  type: string
                password-confirm:
                  type: string
              required:
                - username
                - email
                - password
                - password-confirm
      responses:
        '302':
          description: 'Redirect after successful registration and login.'
          headers:
            Location:
              schema:
                type: string
                example:
                  302Success:
                    description: 'Successful registration. Redirect to user dashboard.'
                    value: '/projects'

  /profile/{id}:
    get:
      operationId: R106
      summary: 'View User Profile'
      description: 'Display the profile of a specific user. Access: USR, ADM'
      tags:
        - 'M01: Authentication and Profile'
      parameters:
        - name: id
          in: path
          required: true
          description: 'The ID of the user whose profile to view.'
          schema:
            type: integer
      responses:
        '200':
          description: 'User profile displayed successfully.'
        '404':
          description: 'User profile not found.'

  /profile/{id}/edit:
    get:
      operationId: R107
      summary: 'Edit User Profile Form'
      description: 'Displays the form for editing the authenticated user  profile. Access: USR, ADM'
      tags:
        - 'M01: Authentication and Profile'
      parameters:
        - name: id
          in: path
          required: true
          description: 'The ID of the user whose profile to edit.'
          schema:
            type: integer
      responses:
        '200':
          description: 'Edit profile form displayed successfully.'
        '404':
          description: 'User not found.'

  /profile:
    post:
      operationId: R108
      summary: 'Update User Profile'
      description: 'Updates the authenticated user profile with new data (username, email, password, biography, and pro
      tags:
        - 'M01: Authentication and Profile'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                username:
                  type: string
                email:
                  type: string
                biography:
                  type: string
                password:
                  type: string
                password_confirmation:
                  type: string
                profile_picture:
                  type: string
                  format: binary
              required:
                - username
                - email
      responses:
```

```yaml
        '302':
          description: 'Redirect after updating the profile.'
          headers:
            Location:
              schema:
                type: string
                example:
                  302Success:
                    description: 'Profile successfully updated. Redirect to the user profile page.'
                    value: '/profile/{id}'
                  302Error:
                    description: 'Error Updating Profile.'
                    value: '/profile/{id}'

  /projects:
    get:
      operationId: R201
      summary: 'View Projects'
      description: 'Display the projects of a specific user. Access: USR, ADM, OWN, PRM, PRC'
      tags:
        - 'M02: Projects'
      responses:
        '200':
          description: 'Projects page displayed successfully.'

  /projects/create:
    get:
      operationId: R202
      summary: 'Show Create Project Form'
      description: 'Display the create project form. Access: USR, ADM, OWN, PRM, PRC'
      tags:
        - 'M02: Projects'
      responses:
        '200':
          description: 'Create Project Form displayed successfully.'
    post:
      operationId: R203
      summary: 'Create a New Project'
      description: 'Create a new project. Access: USR, ADM, OWN, PRM, PRC'
      tags:
        - 'M02: Projects'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                projectname:
                  type: string
                description:
                  type: string
              required:
                - projectname
                - description
      responses:
        '200':
          description: 'New Project created successfully.'

  /projects/{id}:
    get:
      operationId: R204
      summary: 'Show Project Page'
      description: 'Display the create project form. Access: USR, ADM, OWN, PRM, PRC'
      tags:
        - 'M02: Projects'
      parameters:
        - name: id
          in: path
          required: true
          description: 'The ID of the project to view.'
          schema:
            type: integer
      responses:
        '200':
          description: 'Project page displayed successfully.'

  /projects/{projectId}/add-member:
    post:
      operationId: R205
      summary: 'Add a New Member to the Project'
      description: 'Adds a new member to the project. Access: ADM, OWN, PRC'
      tags:
        - 'M02: Projects'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                user_id:
                  type: integer
              required:
                - user_id
      responses:
        '200':
          description: 'A new member was added to the project successfully.'
        '500':
          description: 'Failed adding a new member to the project.'

  /tasks/{id}:
    get:
      operationId: R301
      summary: 'View Task'
      description: 'Displays the details of a specific task, including its status. Access: ADM, OWN, PRM, PRC'
      tags:
        - 'M03: Tasks'
      parameters:
        - name: id
          in: path
          required: true
          description: 'The ID of the task to view.'
          schema:
            type: integer
      responses:
        '200':
          description: 'Task details displayed successfully.'
        '500':
          description: 'Failed to show task.'
    put:
      operationId: R303
      summary: 'Update Task'
      description: 'Updates the details of an existing task. Access: ADM, OWN, PRM, PRC'
      tags:
        - 'M03: Tasks'
      parameters:
        - name: id
          in: path
          required: true
          description: 'The ID of the task to update.'
          schema:
            type: integer
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                task_title:
                  type: string
                task_description:
                  type: string
                deadline:
                  type: string
                  format: date
                task_status_id:
                  type: integer
                  description: 'The status ID of the task.'
                task_priority:
                  type: string
                  enum: [ Low, Medium, High ]
              required:
                - task_title
                - task_status_id
      responses:
        '200':
          description: 'Task updated successfully.'
        '500':
          description: 'Failed to update task. Please try again.'
    delete:
      operationId: R304
      summary: 'Delete Task'
      description: 'Deletes a specific task. Access: ADM, OWN, PRM, PRC'
      tags:
        - 'M03: Tasks'
      parameters:
        - name: id
          in: path
          required: true
          description: 'The ID of the task to delete.'
          schema:
            type: integer
      responses:
        '200':
          description: 'Task deleted successfully.'
        '500':
          description: 'Failed to delete task. Please try again.'

  /tasks/{project}:
    post:
      operationId: R302
      summary: 'Create Task'
      description: 'Creates a new task for a specific project. Access: ADM, OWN, PRM, PRC'
      tags:
        - 'M03: Tasks'
      parameters:
        - name: project
          in: path
          required: true
          description: 'The ID of the project to create the task under.'
          schema:
            type: integer
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                task_title:
                  type: string
                  description: 'Title of the task.'
                task_description:
                  type: string
                  description: 'Description of the task.'
                task_status_id:
                  type: integer
                  description: 'ID of the task status.'
              required:
                - task_title
                - task_status_id
      responses:
        '200':
          description: 'New task created successfully.'

  /search:
    get:
      operationId: R401
      summary: 'Search for different entities'
      description: 'Search for users, projects and tasks. Access: USR, ADM, OWN, PRM, PRC'
      tags:
        - 'M04: Search'
      parameters:
        - name: query
          in: path
          required: true
          description: 'The query to perform in the search.'
          schema:
            type: string
      responses:
        '400':
          description: 'No query provided.'
        '200':
          description: 'Search results displayed successfully.'
        '500':
          description: 'Search Error.'

  /users/search:
    get:
      operationId: R402
      summary: 'Search users'
      description: 'Search for users by username and email. Access: USR, ADM, OWN, PRM, PRC'
      tags:
        - 'M04: Search'
      parameters:
        - name: project_id
          in: path
          required: false
          description: 'The id of the project to search members.'
          schema:
            type: integer
        - name: user
          in: path
          required: true
          description: 'The user to search for.'
          schema:
            type: string
      responses:
        '200':
          description: 'Search results displayed successfully.'
        '500':
          description: 'Failed to load user list.'

  /admin:
    get:
      operationId: R501
      summary: 'Search users'
      description: 'Search for users by username and email. Access: ADM'
      tags:
        - 'M05: Admin'
      parameters:
        - name: project_id
          in: path
          required: false
          description: 'The id of the project to search members.'
          schema:
            type: integer
        - name: user
          in: path
          required: true
          description: 'The user to search for.'
          schema:
            type: string
      responses:
        '200':
          description: 'Search results displayed successfully.'
        '500':
          description: 'Failed to load user list.'
```

A8: Vertical prototype

The Vertical Prototype includes the implementation of high-priority user stories. It features pages for viewing, adding, editing, and deleting data (project, user, task), with access controls defined for different roles (Public, Admin, Owner, User, Project Member, Project Coordinator).

1. Implemented Features

1.1. Implemented User Stories

The table below outlines the user stories implemented in the prototype.

User Story reference	Name	Priority	Responsible	Description
US01	Sign In	High	Rui	As a visitor, I want to be able to log into the system, so I can access restricted features and information.
US02	Sign Up	High	Rui	As a visitor, I want to create an account in the system, so that I can log in.
US03	Logout	High	Rui	As a user, I want to log out of the system, so I can end my session and protect my account.
US04	View Profile	High	Diogo	As a user, I want to view my profile, so that I can see my personal information and account details.
US05	Edit Profile	High	Diogo	As a user, I want to edit my profile, so that I can update my personal information and account details.
US06	Exact Match Search	High	Gonçalo	As a user, I want to perform an exact match search, so that I can find specific information that matches my query.
US07	Full-text Search	High	Gonçalo	As a user, I want to perform a full-text search, so that I can search for relevant information within all content.
US08	Create Project	High	Lucas	As a user, I want to create a new project, so that I can initiate and organize my tasks and collaborate with my team.
US09	View My Projects	High	Lucas	As a user, I want to view my projects, so that I can keep track of my progress and manage my tasks.
US29	Create Task	High	Lucas	As a project member, I want to create a new task, so that I can assign specific actions within a project and manage my workload.
US30	Manage Tasks	High	Lucas	As a project member, I want to manage tasks by setting priorities, adding labels, and defining due dates, so that I can organize my workload and ensure timely completion of my projects.

User Story reference	Name	Priority	Responsible	Description
US31	View Task Details	High	Diogo	As a project member, I want to view task details, so that I can understand the specific requirements, due dates, and assigned users for each task.
US32	Complete an Assigned Task	High	Diogo	As a project member, I want to mark an assigned task as complete, so that I can track my progress and indicate to my team that the task has been finished.
US33	Search Tasks	High	Gonçalo	As a project member, I want to search for tasks, so that I can find specific tasks based on keywords, due dates, or assigned users.
US44	Add User to Project	High	Rui	As a project coordinator, I want to add a user to a project, so that I can collaborate with them on tasks and leverage their skills and expertise.
US53	Administer User Accounts	High	Gonçalo	As an administrator, I want to search, view, edit, and create user accounts, so that I can manage users and ensure proper system access and functionality.

1.2. Implemented Web Resources

The following section describes the web resources implemented in the prototype.

> Module M01: Authentication and Profile

Web Resource Reference	URL
R101: Login Form	GET/login
R102: Login Action	POST/login
R103: Logout Action	GET/logout
R104: Register Form	GET/register
R105: Register Action	POST/register
R106: View Profile	GET/profile/{id}
R107: Edit Profile Form	GET/profile/{id}/edit
R108: Profile Update	POST/profile

> Module M02: Projects

Web Resource Reference	URL
R201: Projects View	GET/projects
R202: Project Create Form	GET/projects/create
R203: Project Create Action	POST/projects/create
R204: Project View	GET/projects/{id}
R205: Project Add Member	POST/projects/{projectId}/add-member

> Module M03: Tasks

Web Resource Reference	URL
R301: Task View	GET/tasks/{id}
R302: Task Create	POST/tasks/{project}
R303: Task Update	PUT/tasks/{id}

Web Resource Reference	URL
R304: Task Delete	DELETE/tasks/{id}

> Module M04: Search

Web Resource Reference	URL
R401: Search	GET/search
R402: User Search	GET/users/search

> Module M05: Admin

Web Resource Reference	URL
R501: Admin View	GET/admin

2. Prototype

To run our prototype, you can use a Docker image we uploaded to Gitlab.

```
docker run -d --name lbaw24135 -p 8001:80 \
gitlab.up.pt:5050/lbaw/lbaw2425/lbaw24135:latest
```

Credentials as email, password:
admin: up202205295@up.pt, 1234
regular: up202208297@up.pt, 1234

GROUP24135, 28/11/2024

- Diogo Ferreira, up202205295@up.pt
- Gonçalo Marques, up202206205@up.pt **(editor)**
- Lucas Greco, up202208296@up.pt
- Rui Cruz, up202208011@up.pt