eap

Last edited by **Rafael Ângelo dos Reis Campeão** 5 months ago

EAP: Architecture Specification and Prototype

> The Bulletin is a community-driven news platform that empowers users to submit, vote on, and discuss news stories. It offers personalized feeds based on user interests, enabling people to engage with relevant content while shaping its visibility through voting. With a reputation system that rewards quality contributions and moderation tools to maintain platform integrity, The Bulletin fosters a respectful and collaborative environment. Accessible on all devices, it aims to create a democratic space where users actively participate in curating and discussing the news.

A7: Web Resources Specification

> This artifact describes the web API to be developed for The Bulletin, detailing the required resources, their properties, and JSON responses. This API includes create, read, update, and delete operations (where applicable) for each resource.

1. Overview

Modules	Description
M01: Authentication	Web resources associated with user authentication and registration, with features such as login/logout and registration.
M02: User	Web resources associated with user information such as view own and others profile and edit own profile
M03: Posts	Web resources assosiated with posts and comments such as create edit and delete news, a detailed page for news items, top news feed and exact match search
M04: Administrator	Web resources associated with features required by admins

2. Permissions

Code	Name	Description
PUB	Public	Unauth user
USR	User	Auth User
BLK	Blocked	Blocked User
OWN	Owner	Owner of a post/comment
ADM	Administrator	Admin of the system

3. OpenAPI Specification

> Section for The Bulletin OpenAPI specification

```yaml
openapi: 3.0.0
info:
  version: '2.0'
  title: 'The Bulletin API'
  description: 'The Bulletin Web Resources Specification (A7)'
servers:
  - url: http://localhost:8001/
externalDocs:
  description: Find more info here
  url: https://gitlab.up.pt/lbaw/lbaw2425/lbaw24104/-/wikis/eap
tags:
 - name: 'M01: Authentication'
 - name: 'M02: Users'
 - name: 'M03: Posts'
 - name: 'M04: Administration'
paths:
  /login:
    get:
      operationId: 'R101'
      summary: 'R101: Login Form'
      description: 'Provide login form Access: PUB'
      tags:
        - 'M01: Authentication'
      responses:
        '200':
          description: 'Ok. Show Log-in UI'
    post:
      operationId: R102
      summary: 'R102: Login Action'
      description: 'Process the login form submission. Access: PUB'
      tags:
        - 'M01: Authentication'
      requestBody:
        required: true
        content:
          application/x-www-form-urlenconded:
            schema:
              type: object
              properties:
                email:
                  type: string
                  format: email
                password:
                  type: string
                  format: password
              required:
                - email
                - password
      responses:
        '302':
          description: 'Redirect after processing login credentials'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  description: 'Login Successful. Redirect to homepage.'
                  value: '/home'
                302Error:
                  description: 'Failed authentication. Redirect to Login form.'
                  value: '/login'

  /logout:
    get:
      operationId: R103
      summary: 'R103: Logout Action'

      description: 'Logout current user. Access: USR,ADM'
      tags:
        - 'M01: Authentication'
      responses:
        '302':
          description: 'Redirect after processing logout.'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  description: 'Successful Logout. Redirect to login form.'
                  value: '/login'
  /register:
    get:
```

```yaml
      operationId: R104
      summary: 'R104: Register Form'
      description: 'Register a new user. Access: PUB'
      tags:
        - 'M01: Authentication'
      responses:
        '200':
          description: 'Ok. Show Sign-Up UI'
    post:
      operationId: R105
      summary: 'R105: Register Action'
      description: 'Processes the new user registration form submission. Access: PUB'
      tags:
      - 'M01: Authentication'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                username:
                  type: string
                email:
                  type: string
                  format: email
                password:
                  type: string
                  format: password
                confirm_password:
                  type: string
                  format: password
              required:
                - username
                - email
                - password
                - confirm_password
      responses:
        '302':
          description: 'Redirect after processing the new user information.'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  description: 'Successful authentication. Redirect to homepage.'
                  value: '/home'
                302Failure:
                  description: 'Failed authentication. Redirect to login form.'
                  value: '/login'
  /users/{id}:
    get:

      operationId: R201
      summary: 'R201: View user’s profile'
      description: 'Show the profile for a specific user. Access: USR, ADM'
      tags:
        - 'M02: Users'
      parameters:
        - in: path
          name: id
          schema:
            type: integer
          required: true
          description: 'The ID of the user whose profile is being viewed.'
      responses:
        '200':
          description: 'OK. Show the profile for the specified user.'
        '404':
          description: 'User not found.'
  /users/{id}/edit:
    get:
      operationId: R202
      summary: 'R202: View users edit profile form'
      description: 'Show the edit profile form for a specific user. Access: USR, ADM'
      tags:
        - 'M02: Users'
      parameters:
        - in: path
          name: id
          schema:
            type: integer
          required: true
          description: 'Id of profile to edit'
      responses:
        '200':
          description: 'OK. Show the form to edit profile'
        '403':
          description: 'No premission must be owner of the profile or admin'
  /api/users/{id}/edit:
    post:
      operationId: R203
      summary: 'R203: Edit profile'
      description: 'Updates the profile information of the specified user. Access: USR, ADM'
      tags:
        - 'M02: Users'
      parameters:
        - name: id
          in: path
          required: true
          schema:
            type: integer
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                username:
                  type: string
                  description: 'Updated username'
                email:
                  type: string
                  format: email
                  description: 'Updated email address'
                password:
                  type: string
                  format: password
                confirm_password:
                  type: string
                  format: password

                  description: 'Confirm updated password'
              required:
                - username
                - email
      responses:
        '200':
          description: 'Profile updated successfully.'
          content:
            application/json:
              schema:
                type: object
                properties:
                  username:
                    type: string
                  email:
                    type: string
        '400':
          description: 'Bad request. Invalid input.'
        '404':
          description: 'User not found.'
  /search:
    get:
      operationId: R204
      summary: 'R204: Exact Match Search'
      description: 'Search for username using exact match search. Access: ADM, USR'
      tags:
        - 'M02: Users'
      parameters:
        - in: query
          name: query
          description: 'Search string for exact match search in user username.'
          schema:
            type: string
          required: true
      responses:
        '302':
          description: 'Redirect after successfully matching a username.'
          headers:
            Location:
              schema:
                type: string
              examples:
                302Success:
                  value: '/users/{id}'
                  description: 'Redirect to the user profile.'
                302Failure:
                  value: '/search'
                  description: 'Failed. Redirect back.'
  /createPosts:
    get:
      operationId: R301
      summary: 'R301: Create News Item Form'
      description: 'Provide login form Access: USR'
      tags:
        - 'M03: Posts'
      responses:
        '200':
          description: 'Ok. Show Log-in UI'
  /api/createPosts:
    post:
      operationId: R302
      summary: 'R302: Create News action'
      description: 'Submit a new news item. Access: USR'
      tags:
        - 'M03: Posts'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:

              type: object
              properties:
                title:
                  type: string
                  description: 'Title of the news item'
                content:
```

```
                type: string
                description: 'Main content of the news item'
            required:
              - title
              - content
    responses:
      '302':
        description: 'Redirect after successfully creating the news item.'
        headers:
          Location:
            schema:
              type: string
            examples:
              302Success:
                value: '/post/{id}'
                description: 'Redirect to the newly created news item.'
              302Failure:
                value: '/posts/createPost'
                description: 'Failed. Redirect back.'
/post/{id}:
  get:
    operationId: R303
    summary: 'R303: View News Item'
    description: 'Retrieve detailed information about a specific news item. Access: PUB, USR'
    tags:
      - 'M03: Posts'
    parameters:
      - in: path
        name: id
        schema:
          type: integer
        required: true
        description: 'The ID of the news item to retrieve.'
    responses:
      '200':
        description: 'Success. Returns the details of the post.'
        content:
          application/json:
            schema:
              type: object
              properties:
                id:
                  type: integer
                title:
                  type: string
                content:
                  type: string
                created_at:
                  type: string
                  format: date-time
                updated_at:
                  type: string
                  format: date-time
                author:
                  type: string
                tags:
                  type: array
                  items:
                    type: string
      '404':
        description: 'News item not found.'
/post/edit/{id}:
  post:
    operationId: R304

    summary: 'R304: Edit Post action'
    description: 'Edit an existing post by the author. Access: USR (author only) ADM.'
```

```yaml
      tags:
        - 'M03: Posts'
      parameters:
        - in: path
          name: id
          schema:
            type: integer
          required: true
          description: 'The ID of the post to edit.'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                title:
                  type: string
                  description: 'Updated title of the post.'
                content:
                  type: string
                  description: 'Updated content of the post.'
                tags:
                  type: array
                  items:
                    type: string
                  description: 'Updated tags for the post.'
              required:
                - title
                - content
      responses:
        '200':
          description: 'Post updated successfully.'
          content:
            application/json:
              schema:
                type: object
                properties:
                  id:
                    type: integer
                  title:
                    type: string
                  content:
                    type: string
                  updated_at:
                    type: string
                    format: date-time
                  tags:
                    type: array
                    items:
                      type: string
        '403':
          description: 'Forbidden. You can only edit your own post.'
        '404':
          description: 'Post not found.'
  /deletePost/{id}:
    post:
      operationId: R305
      summary: 'R305: Delete Post action'
      description: 'Delete a post by the author. Access: USR (author only) ADM.'
      tags:
        - 'M03: Posts'
      parameters:
        - in: path
          name: id
          schema:
            type: integer
    required: true

    description: 'The ID of the post to delete.'
  responses:
   '200':
    description: 'Post deleted successfully.'
   '403':
    description: 'Forbidden. You can only delete your own post.'
   '404':
    description: 'Post not found.'
/:
 get:
  operationId: R306
  summary: 'R306: View all Posts'
  description: 'Retrieve posts and show them in the home page.  Access: PUB, USR'
  tags:
   - 'M03: Posts'
  parameters:
   - in: query
    name: query
    description: 'Search string for full-text search in post titles and content.'
    schema:
     type: string
    required: false
  responses:
   '200':
    description: 'Success. Returns the list of posts.'
    content:
     application/json:
      schema:
       type: array
       items:
        type: object
        properties:
         id:
          type: integer
         title:
          type: string
         content:
          type: string
         created_at:
          type: string
          format: date-time
         updated_at:
          type: string
          format: date-time
         author:
          type: string
         tags:
          type: array
          items:
           type: string
   '404':
    description: 'No posts found.'

/home:
 get:
  operationId: R307
  summary: 'R307: Home Page'
  description: 'Show the home page with the latest news items. Access: PUB, USR'
  tags:
   - 'M03: Posts'
  responses:
   '200':
    description: 'OK. Show the home page with the latest news items.'

/user/{id}/posts:
    get:
      operationId: R308
      summary: 'R308: View Users Posts'
      description: 'Show the page with a user news items. Access: USR'

      tags:
        - 'M03: Posts'
      parameters:
        - in: path
          name: id
          schema:
            type: integer
          required: true
      responses:
        '200':
          description: 'Success. Returns the list of posts in which the author has the same id as the owner of the prof
          content:
            application/json:
              schema:
                type: array
                items:
                  type: object
                  properties:
                    id:
                      type: integer
                    title:
                      type: string
                    content:
                      type: string
                    created_at:
                      type: string
                      format: date-time
                    updated_at:
                      type: string
                      format: date-time
                    author:
                      type: string
                    tags:
                      type: array
                      items:
                        type: string
        '404':
          description: 'No posts found.'


  /admin:
    get:
      operationId: R401
      summary: 'R401: Admin Dashboard'
      description: 'Show the Admin Dashboard. Access: ADM'
      tags:
        - 'M04: Administration'
      responses:
        '200':
          description: 'OK. Show the Admin Dashboard.'
  /admin/create:
    post:
      operationId: R402
      summary: 'R402: Create User'
      description: 'Allows a System Manager to create a new user account dynamically from the Admin Dashboard. Access:
      tags:
        - 'M04: Administration'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                username:
                  type: string
                  description: 'The username of the new user'
                email:
                  type: string

                  format: email
                  description: 'The email address of the new user'
                password:
                  type: string
                  format: password
                  description: 'The password for the new user'
                confirm_password:
                  type: string
                  format: password
                  description: 'Confirmation of the password for the new user'
              required:
                - username
                - email
                - password
                - confirm_password
      responses:
        '201':
          description: 'User successfully created.'
          content:
            application/json:
              schema:
                type: object
                properties:
                  success:
                    type: boolean
                    example: true
                  user_id:
                    type: integer
                  username:
                    type: string
                  email:
                    type: string
        '400':
          description: 'Validation error. Invalid input data.'
  /api/admin/edit/{id}:
    post:
      operationId: R403
      summary: 'R403: Edit User'
      description: 'Allows a System Manager to edit the profile information of a user. Access: ADM'
      tags:
        - 'M04: Administration'
      parameters:
        - in: path
          name: id
          schema:
            type: integer
          required: true
          description: 'The ID of the user to edit.'
      requestBody:
        required: true
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                username:
                  type: string
                  description: 'Updated username'
                email:
```

```
            type: string
            format: email
            description: 'Updated email address'
           password:
            type: string
            format: password
            description: 'Updated password'
           confirm_password:
            type: string
            format: password

            description: 'Confirm updated password'
          required:
           - username
           - email
      responses:
       '200':
        description: 'User profile updated successfully.'
        content:
         application/json:
          schema:
           type: object
           properties:
            username:
             type: string
            email:
             type: string
       '400':
        description: 'Bad request. Invalid input.'
       '404':
        description: 'User not found.'

...
```

A8: Vertical prototype

> Brief presentation of the artifact goals.

1. Implemented Features

1.1. Implemented User Stories

User Story reference	Name	Priority	Responsible	Description
US01	Register	High	Rafael Campeão	As a visitor, I want to register, so that I can authenticate myself into the system.
US02	Login	High	Rafael Campeão	As a visitor, I want to be able to login, so that I can enter my account and have access privileged information.
US03	Top News Feed	High	Luana Lima	As a user, I want to see the top news feed, so that I can remain up to date with popular topics.
US04	See News Item	High	Rafael Campeão	As a user, I want to see a news item, so that I can read the entire article.
US05	Exact Match Search	High	Miguel Neri	As a user, I want to search for news articles using an exact phrase, so that I can quickly find content that matches my specific query without unrelated results.
US06	Full-Text Search	High	Miguel Neri	As a user, I want to perform a full-text search across all articles, so that I can find content that contains the keywords or phrases I'm looking for within the body of the text.
US19	Logout	High	Luana Lima	As a reader, I want to logout, so that i can leave my account.

User Story reference	Name	Priority	Responsible	Description
US20	Profile Page	High	João Cordeiro	As a reader, I want to have a profile page, so that my information can be displayed.
US21	Edit Profile	High	João Cordeiro	As a reader, I want to be able to edit my profile, so that I can update my profile.
US22	See Other Profiles	High	João Cordeiro	As a reader, I want to see other users profile, so that I can know more about them.
US23	News Item Details	High	Rafael Campeão	As a reader, I want to see a news item details, so that I can see its author, the publication date and the votes.
US24	Other Users News Feeds	High	Luana Lima	As a reader, I want to view other users news feeds, so I can check what they have been writing about.
US51	Create News Item	High	Rafael Campeão	As an author, I want to create a news item, so that I can share my news.
US52	Edit my news	High	Rafael Campeão	As an author, I want to edit the news I post, so that I can correct errors and keep the news up to date.
US53	Delete my news	High	Rafael Campeão	As an author, I want to be able to delete the news I post, so that in case I notice on time that I made a mistake.
US57	Administer Accounts	High	João Cordeiro	As a System Manager, I want to administer user accounts, so that I can create, view and edit them.

1.2. Implemented Web Resources

> Module M01: Module Authentification

Web Resource Reference	URL
R101: Login Form	GET /login
R102: Login Action	POST /login
R103: Logout Action	POST /logout
R104: Register Form	GET /register
R105: Register Form	POST /register

> Module M02: Module User

Web Resource Reference	URL
R201: Show user profile	GET /users/{id}
R102: Show edit profile form	GET /users/{id}/edit
R203: Edit profile action	POST /api/users/{id}/edit
R204: Search for username	GET /search

> Module M03: Module Posts

Web Resource Reference	URL
R301: Post creation Form	GET /createPosts
R302: Post creation action	POST /api/createPosts
R303: Show post	GET /post/{id}

Web Resource Reference	URL
R304: Edit post action	POST /post/edit/{id}
R305: Delete post action	DELETE /deletePost{id}
R306: Show posts	GET /
R307: Home page	GET /home
R308: Show users posts	GET /user/{id}/posts

> Module M04: Module Admin

Web Resource Reference	URL
R401: Show admin dashboard	GET /admin
R402: Create new user action	POST /admin/create
R403: Edit user profile action	POST /api/admin/edit/{id}

2. Prototype

In this prototype, our primary focus was on implementing the core functionalities of the project. While we didn’t prioritize the visual design, the interface is structured enough to provide a clear sense of the overall layout and ensure easy navigation throughout the website.

Start the docker image: `docker run -d --name lbaw24104 -p 8001:80 gitlab.up.pt:5050/lbaw/lbaw2425/lbaw24104`

Credentials:

- admin user: admin@test.com | password
- regular user: t4@test.com | password

The code is available at https://gitlab.up.pt/lbaw/lbaw2425/lbaw24104

Revision history

Changes made to the first submission:

GROUP104, 27/11/2024

- João Cordeiro, up202205682@up.pt
- Luana Lima, up202205682@up.pt
- Miguel Neri, up202006475@up.pt
- Rafael Campeão, up202207553@up.pt

Comments