

EAP: Architecture Specification and Prototype

The vision of this project is to create a social network for food enthusiasts, where they can discover, share, and engage with dining experiences worldwide. This platform will offer in-depth reviews, personalized recommendations, and community interactions, empowering food lovers to make informed dining choices and explore new culinary horizons.

A7: Web Resources Specification

This artifact defines the web API to be developed, detailing the necessary resources, their properties, and the structure of JSON responses. It includes Create, Read, Update, and Delete (CRUD) operations for each resource.

1. Modules

Modules	Description
M01: Authentication	Web resources for user authentication and registration, including functionalities like login, logout, and user sign-up.
M02: Profile	Web resources for accessing user information, including the homepage, profile page, and options to edit profile details.
M03: Posts	Web resources for managing posts, including features for creating, editing, deleting, and viewing posts.
M04: Search	Web resources related to search functionality, including the ability to search for users, posts, and comments with Exact Match Search or Full-Text Search.
M05: Administration	Web resources for administrators to manage user accounts, posts, and comments, enabling the review and search for inappropriate activity.

Table 87: Raffia resources overview.

2. Permissions

This section outlines the permissions discussed in the previous section to define the access conditions for resources.

Identifier	Name	Description
VST	Visitor	An unauthenticated user.
USR	User	An authenticated user.
OWN	Owner	Users who own information such as their profile, posts, and comments.
ADM	Administrator	Platform administrator.

Table 88: Raffia permissions.

3. OpenAPI Specification

You can acess the a7_openapi.yaml file here: [a7_openapi](#)

```
openapi: 3.0.0

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

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

externalDocs:
  description: Find more info here.
  url: http://example.com
```

```
tags:
  - name: 'M01: Authentication'
  - name: 'M02: Profile'
  - name: 'M03: Posts'
  - name: 'M04: Search'
  - name: 'M05: Administration'

paths:
  /login:
    get:
      tags:
        - 'M01: Authentication'
      summary: Display a login form
      operationId: 559d6d342a89e8abd4973c02942355a1
      responses:
        '200':
          description: Successful response
          content:
            application/json:
              schema:
                type: string
    post:
      tags:
        - 'M01: Authentication'
      summary: Handle an authentication attempt
      operationId: 2cd09128d362ec2f2862c827936f3549
      requestBody:
        required: true
        content:
          application/json:
            schema:
              properties:
                login:
                  type: string
                password:
                  type: string
                remember:
                  type: boolean
                nullable: true
              type: object
      responses:
        '200':
          description: Successful authentication
          content:
            application/json:
              schema:
                type: string
        '401':
          description: Unauthorized
        '422':
          description: Validation error
  /logout:
    post:
      tags:
        - 'M01: Authentication'
      summary: Log out the authenticated user
      operationId: fe1c1eca634ba2811908fec5e5183950
      responses:
        '200':
          description: Successful logout
          content:
            application/json:
              schema:
                type: string
  /register:
    get:
      tags:
```

```
- 'M01: Authentication'
summary: Show the sign up form
operationId: a9cb8c0a946f50565763929a13d438f6
responses:
  '200':
    description: Successful response
    content:
      application/json:
        schema:
          type: string
post:
  tags:
    - 'M01: Authentication'
  summary: Register a new user
  operationId: 6cb93c5db517a3e6e369c44ede7f22fd
  requestBody:
    required: true
    content:
      application/json:
        schema:
          properties:
            name:
              type: string
              maxLength: 250
            username:
              type: string
              maxLength: 250
            email:
              type: string
              format: email
              maxLength: 250
            password:
              type: string
              minLength: 3
            password_confirmation:
              type: string
              minLength: 3
            description:
              type: string
              maxLength: 250
              nullable: true
            image:
              type: string
              format: binary
              nullable: true
            role:
              type: string
              enum:
                - client
                - restaurant
            type_id:
              type: integer
              nullable: true
            capacity:
              type: integer
              nullable: true
          type: object
  responses:
    '201':
      description: User registered successfully
      content:
        application/json:
          schema:
            type: string
    '422':
      description: Validation error
'/post/{id}/comment':
```

```
post:
  tags:
    - 'M03: Posts'
  summary: Add a comment to a post
  operationId: 90fb1cad226efa82da1e068591e8cadc
  parameters:
    - name: id
      in: path
      required: true
      schema:
        type: integer
  requestBody:
    required: true
    content:
      application/json:
        schema:
          type: object
          properties:
            content:
              type: string
              maxLength: 255
  responses:
    '200':
      description: Comment added successfully
      content:
        application/json:
          schema:
            type: object
            properties:
              success:
                type: boolean
              comment:
                $ref: '#/components/schemas/Comment'
    '422':
      description: Validation error
'/post/{id}/comments':
  get:
    tags:
      - 'M03: Posts'
    summary: Load comments for a post
    operationId: dab0bb61f6efae2ae7321dc1713ba578
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
      - name: offset
        in: query
        required: false
        schema:
          type: integer
          default: 0
      - name: limit
        in: query
        required: false
        schema:
          type: integer
          default: 20
    responses:
      '200':
        description: Comments loaded successfully
        content:
          application/json:
            schema:
              type: array
              items:
```

```

        $ref: '#/components/schemas/Comment'
'/like-post/{id}':
  post:
    tags:
      - 'M03: Posts'
    summary: Like or unlike a post
    operationId: a01ad24c58542b26090b2dc243e7f4d1
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
    responses:
      '200':
        description: Successfully liked or unliked the post
        content:
          application/json:
            schema:
              type: object
              properties:
                success:
                  type: boolean
                liked:
                  type: boolean
                  nullable: true
                unliked:
                  type: boolean
                  nullable: true
      '404':
        description: Post not found
'/likes/{postId}':
  get:
    tags:
      - 'M03: Posts'
    summary: Get likes by post ID
    operationId: 47bc3262cca8954d783bfc5ab82f72bc
    parameters:
      - name: postId
        in: path
        required: true
        schema:
          type: integer
    responses:
      '200':
        description: Successfully retrieved likes
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/LikePost'
'/post/{id}':
  get:
    tags:
      - 'M03: Posts'
    summary: Show the post for a given id
    operationId: 820a95132e64f02c9c71b2cfc3cadb0b
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
    responses:
      '200':
        description: Successful response
```

```
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Post'
    '404':
      description: Post not found
delete:
  tags:
    - 'M03: Posts'
  summary: Delete a post
  operationId: cdf3a5c9a54637677e9d19e0bd933ab4
  parameters:
    - name: id
      in: path
      required: true
      schema:
        type: integer
  responses:
    '200':
      description: Post deleted successfully
      content:
        application/json:
          schema:
            type: object
            properties:
              success:
                type: string
    '404':
      description: Post not found
/posts:
  get:
    tags:
      - 'M03: Posts'
    summary: Show the feed of posts made by followed users and restaurants
    operationId: defdc3831ba536184a2f2353075ac97e
    responses:
      '200':
        description: Successful response
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/Post'
    '401':
      description: Unauthorized
/api/posts/create:
  post:
    tags:
      - 'M03: Posts'
    summary: Create a new post
    operationId: 8290ca9ff65b39e8d4f21af94b6b037e
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              content:
                type: string
              images:
                type: array
                items:
                  type: string
                  format: binary
              rating:
```

```

        type: integer
        nullable: true
      restaurant_id:
        type: integer
        nullable: true
    responses:
      '201':
        description: Post created successfully
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Post'
      '422':
        description: Validation error
'/posts/{id}':
  put:
    tags:
      - 'M03: Posts'
    summary: Update a post
    operationId: 06c81121178208730e11d2f6b655e78d
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
    requestBody:
      required: true
      content:
        application/json:
          schema:
            properties:
              content:
                type: string
              images:
                type: array
                items:
                  type: string
                  format: binary
              rating:
                type: integer
                nullable: true
            type: object
    responses:
      '200':
        description: Post updated successfully
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Post'
      '422':
        description: Validation error
'/profile/{id}':
  get:
    tags:
      - 'M02: Profile'
    summary: Show the profile for a given id
    operationId: 7896cd4c9914563bcd4562d735c3de27
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
    responses:
      '200':
        description: Successful response

```

```
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/User'
  '404':
    description: User not found
'/profile/{id}/follow':
  post:
    tags:
      - 'M02: Profile'
    summary: Follow a user
    operationId: d9fdbd1ecbecc48f6040b9eb93a49161
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
    responses:
      '200':
        description: Successfully followed the user
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/User'
      '404':
        description: User not found
'/profile/{id}/unfollow':
  post:
    tags:
      - 'M02: Profile'
    summary: Unfollow a user
    operationId: cd5c110ff64dfec1ff802ae6527ce52a
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
    responses:
      '200':
        description: Successfully unfollowed the user
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/User'
      '404':
        description: User not found
'/profile/{id}/edit':
  get:
    tags:
      - 'M02: Profile'
    summary: Edit the profile for a given id
    operationId: 13ba8d21152584bd477fbd7090d5cc59
    parameters:
      - name: id
        in: path
        required: true
        schema:
          type: integer
    responses:
      '200':
        description: Successful response
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/User'
```



```
  '404':
    description: User not found
/profile/update:
  post:
    tags:
      - 'M02: Profile'
    summary: Update the profile
    operationId: 64de3f5784fae6c1fc66d45de961f260
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              name:
                type: string
                maxLength: 255
              description:
                type: string
                nullable: true
              image:
                type: string
                format: binary
                nullable: true
    responses:
      '200':
        description: Profile updated successfully
        content:
          application/json:
            schema:
              type: object
              properties:
                success:
                  type: boolean
                user:
                  $ref: '#/components/schemas/User'
  '422':
    description: Validation error
/restaurants/search:
  get:
    tags:
      - 'M04: Search'
    summary: Search for restaurants
    operationId: 62bf429846a612797d8bc95bcc1b0026
    parameters:
      - name: q
        in: query
        required: false
        schema:
          type: string
    responses:
      '200':
        description: Successful response
        content:
          application/json:
            schema:
              type: array
              items:
                type: object
                properties:
                  id:
                    type: integer
                  name:
                    type: string
                  username:
                    type: string
```

```
        email:
          type: string
        image:
          type: string
          nullable: true
/search:
  get:
    tags:
      - 'M04: Search'
    summary: Handles the search functionality for users and posts
    operationId: 4724040b63b941e77f510ebfaf0d95a1
    parameters:
      - name: query
        in: query
        required: true
        schema:
          type: string
      - name: type
        in: query
        required: true
        schema:
          type: string
          enum:
            - users
            - posts
      - name: searchType
        in: query
        required: false
        schema:
          type: string
          enum:
            - full-text
            - exact-match
    responses:
      '200':
        description: Successful response
        content:
          application/json:
            schema:
              type: array
              items:
                type: object
      '422':
        description: Validation error
      '500':
        description: Internal server error
components:
  schemas:
    Comment:
      type: object
      properties:
        id:
          type: integer
        content:
          type: string
        post_id:
          type: integer
        user_id:
          type: integer
        datetime:
          type: string
          format: date-time
    LikePost:
      type: object
      properties:
        id:
          type: integer
```

```
  user_id:
    type: integer
  post_id:
    type: integer
  datetime:
    type: string
    format: date-time
Post:
  type: object
  properties:
    id:
      type: integer
    content:
      type: string
    user_id:
      type: integer
    images:
      type: array
      items:
        type: string
    datetime:
      type: string
      format: date-time
User:
  type: object
  properties:
    id:
      type: integer
    name:
      type: string
    username:
      type: string
    email:
      type: string
    description:
      type: string
    image:
      type: string
      nullable: true
Restaurant:
  type: object
  properties:
    id:
      type: integer
    name:
      type: string
    username:
      type: string
    email:
      type: string
    image:
      type: string
      nullable: true
    rating:
      type: integer
      format: date-time
    capacity:
      type: integer
      format: date-time
    type:
      type: string
      format: date-time
Client:
  type: object
  properties:
    id:
      type: integer
```

```
name:
  type: string
username:
  type: string
email:
  type: string
image:
  type: string
nullable: true
```

A8: Vertical prototype

The Vertical Prototype implements key features from the common and theme requirements, addressing high-priority user stories and validating the proposed architecture while providing foundational knowledge of the project technologies. Following recommendations, the implementation is based on the LBAW Framework code and covers all layers of the solution’s architecture. The prototype includes the implementation of page visualizations (such as homepage, profile, admin, and search), along with functionality for creating, editing, and removing posts, as well as displaying error and success messages.

1. Implemented Features

1.1. Implemented User Stories

The user stories implemented in the prototype are detailed in the table below.

User Story reference	Name	Priority	Description
U01	View a public timeline	High	As a user, I want to view a public timeline, so that I can explore the posts from other users
U02	View public user profiles	High	As a user, I want to be able to view public user profiles, so that I can see specific users' posts
U03	Search Profiles	High	As a user, I want to be able to search for user profiles by their tag, so that I may view their profiles and posts
U04	Exact Match Search	High	As a user, I want to be able to do Exact Match Search, so that I can find what I am looking for easily
U05	Full-text Search	High	As a user, I want to be able to do Full-text Search, so that I can find what I am looking for easily
V01	Sign In	High	As a visitor, I want to sign in to my account, so that I can access personalized features and interact fully with the platform
V02	Sign Up	High	As a visitor, I want to sign up for a new account, so that I can become a member of the platform
A01	View personalized timeline	High	As an authenticated user, I want to be able to view a personalized timeline, so that I can stay engaged
A02	View public user profiles	High	As an authenticated user, I want to be able to view public user profiles, so that I can explore other user's posts
A03	View own profile	High	As an authenticated user, I want to be able to see my profile, so that I can see the content other users find when they visit my personal page
A04	Edit profile	High	As an authenticated user, I want to be able to edit my profile, so that I can keep it up-to-date and appealing
AC01	Follow Users	High	As an authenticated client, I want to be able to follow other users, so that I can stay in touch and see their posts
AC02	Create a review	High	As an authenticated client, I want to be able to create a review, so that I can share my experiences and rate restaurants
AC03	View followed users' feed	High	As an authenticated client, I want to be able to view the feed of users I follow, so that I can see what they are eating and their reviews
AC04	See network posts	High	As an authenticated client, I want to be able to see posts from my network, so that I can engage and make more informed decisions

User Story reference	Name	Priority	Description
PA01	Edit post	High	As a post author, I want to be able to edit my post, so that I can change something that was wrong
PA02	Delete post	High	As a post author, I want to be able to delete my post, so that I can remove it from the platform
ADMIN01	Administer User Accounts	High	As an administrator, I want to administer all the users' accounts, posts, comments, so that I can review and search for any inappropriate activity

Table 88: User stories implemented in the prototype.

1.2. Implemented Web Resources

Module M01: Authentication

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 Action	POST /register

Table 89: Authentication implementation.

Module M02: Profile

Web Resource Reference	URL
R201: View user profile	GET /profile/{id}
R202: Follow a user	POST /profile/{id}/follow
R203: Unollow a user	POST /profile/{id}/unfollow
R204: Edit the user profile	GET/profile/{id}/edit
R205: Update the user profile	GET/profile/update

Table 90: Users implementation.

Module M03: Posts

Web Resource Reference	URL
R301: Create a new post	POST /api/posts/create
R302: Update a post	PUT /posts/{id}
R303: Delete a post	DELETE /posts/{id}
R304: Homepage	GET /posts
R305: Show a specific post	GET /posts/{id}
R306: Get likes by post id	GET /likes/{postId}
R307: Like a post	POST /like-post/{id}
R308: Load comments of a post	GET /post/{id}/comments
R309: Add a comment to a post	POST /post/{id}/comment

Table 91: Posts implementation.

Module M04: Search

Web Resource Reference	URL
R401: View user search page	GET /search
R402: Search users	GET /search
R403: Search posts	GET /search
R404: Search comments	GET /search
R405: Search restaurants	GET /restaurants/search

Table 92: Search implementation.

Module M05: Administration

Web Resource Reference	URL
R501: View profile page	GET /profile/{id}
R502: Search all posts in homepage	GET /posts/{id}

Table 92: Administration implementation.

2. Prototype

2.1. Instructions

```
# Login to GitLab's Container Registry
docker login gitlab.up.pt:5050

# Pull the Docker image
docker pull gitlab.up.pt:5050/lbaw/lbaw2425/lbaw2432

# Run the Docker container
docker run -d --name lbaw2432 -p 8001:80 gitlab.up.pt:5050/lbaw/lbaw2425/lbaw2432

# Access the application at http://localhost:8001
```

2.2. Credentials

- Client Admin: nelson/nelson
- Restaurant Admin: george_gourmet/password7
- Client: dteixeira/teixeira10
- Restaurant: chefcharlie/password3

To go to the prototype source code follow this link: [Prototype](#)

Revision history

Changes made to the first submission:

1. Added a table to the database to replace the enum.
2. Column added to users table:'remember_token'.
3. Changed passwords using encryption method bcrypt.

Editor: Sara Azevedo

GROUP2432, 23/11/2024

- Paulo Fidalgo, [up201806603@fe.up.pt](#)
- Wagner Pedrosa, [up201908556@fe.up.pt](#)
- Sara Azevedo, [up202006902@fe.up.pt](#)
- Nelson Neto, [up202108117@fe.up.pt](#)