

EAP: Architecture Specification and Prototype

The Popcorn Post aims to be the go-to collaborative news platform where cinema and entertainment enthusiasts actively engage, share, and explore their passion while fostering a trusted and vibrant community.

A7: Web Resources Specification

This artefact outlines the blueprint for our web application, detailing the resources, their properties, and the expected JSON responses. It follows the OpenAPI standard using YAML. The documentation includes the CRUD (create, read, update, delete) operations for each user story implemented in the Vertical Prototype.

1. Overview

An overview of the web application is presented in this section. The web application is divided into modules, which are described in the following subsections. The web resources associated with each module and the features to be implemented in the vertical prototype are detailed in the individual documentation of each module.

Module	Description
M01: Authentication and Individual Profile	Web resources associated with user authentication and individual profile management. This includes features such as login and logout, user registration, account recovery, view personal profile information and edit that information.
M02: Content Items	Web resources associated with all content items (i.e., news articles and comments). This includes features such as creating, viewing, editing, deleting and reporting both news articles and comments.
M03: Votes and Tags	Web resources associated with the voting system and the association between tags and news articles. This includes features such as voting on news articles and comments, adding tags to a news article, removing tags from a news article and following tags.
M04: Topics	Web resources related to topics. This includes features such as associating a news article with a topic and suggesting new topics.
M05: Members	Web resources associated with the interaction between members. This includes features such as viewing other members' profiles and following other members.
M06: Notifications	Web resources related to the notification system. This includes viewing all different kinds of notifications.
M07: Search	Web resources associated with the searching mechanism. This includes features such as searching for news articles, comments and users, and all the associated filters.

Module	Description
M08: Member Administration and Static Pages	Web resources associated with the management of users and content items and the static pages of the web application. This includes features such as reviewing content and user reports, banning users, editing and deleting news articles, deleting comments and approving or dismissing suggested topics. It also contains the "About Us", "FAQ" and "Contact Us" static pages pages.

Table 1: The Popcorn Post's modules.

2. Permissions

Identifier	Name	Description
PUB	Public	Users without privileges
MEM	Member	Authenticated users
OWN	Owner	User that is owner of the information (e.g. it's own profile, news articles and comments)
ADM	Administrator	System administrators

Table 2: The Popcorn Post's permissions.

3. OpenAPI Specification

The OpenAPI Specification file is described [here](#)

```

openapi: 3.0.0

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

servers:
- url: http://lbaw2356.lbaw.fe.up.pt
  description: Production server

tags:
- name: 'M01: Authentication and Individual Profile'
- name: 'M02: Content Items'
- name: 'M03: Votes and Tags'
- name: 'M04: Topics'
- name: 'M05: Members'
- name: 'M06: Notifications'
- name: 'M07: Search'
- name: 'M08: Member Administration and Static Pages'

paths:
  /login:

```

```

get:
  operationId: R101
  summary: 'R101: Login Form'
  description: 'Provide login form. Access: PUB'
  tags:
    - 'M01: Authentication and Individual Profile'
  responses:
    '200':
      description: 'Success. Display Log-in Form (UI06)'
post:
  operationId: R102
  summary: 'R102: Login Authentication'
  description: 'Processes the login form submission. Access: PUB'
  tags:
    - 'M01: Authentication and Individual Profile'

requestBody:
  required: true
  content:
    application/x-www-form-urlencoded:
      schema:
        type: object
        properties:
          email: # <!-- form field name
            type: string
          password: # <!-- form field name
            type: string
        required:
          - email
          - password

responses:
  '302':
    description: 'Redirect after processing the login credentials.'
    headers:
      Location:
        schema:
          type: string
    examples:
      302Success:
        description: 'Successful authentication. Redirect to home page.'
        value: '/home'
      302Error:
        description: 'Failed authentication. Redirect to login form.'
        value: '/login'

/logout:
get:
  operationId: R103
  summary: 'R103: Logout Action'
  description: 'Logout the current authenticated member. Access: MEM, ADM'
  tags:
    - 'M01: Authentication and Individual Profile'
  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:
    operationId: R104
    summary: 'R104: Register Form'
    description: 'Provide new user registration form. Access: PUB'
    tags:
      - 'M01: Authentication and Individual Profile'
    responses:
      '200':
        description: 'Ok. Show Sign-Up UI (UI07)'

  post:
    operationId: R105
    summary: 'R105: Register Action'
    description: 'Processes the new user registration form submission. Access:
PUB'
    tags:
      - 'M01: Authentication and Individual Profile'

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

```

```

responses:
  '302':
    description: 'Redirect after processing the new user information.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Successful authentication. Redirect to user
profile.'
            value: '/members/{username}'
          302Failure:
            description: 'Failed authentication. Redirect to register form.'
            value: '/register'

/members/{username}:
  get:
    operationId: R106
    summary: 'R106: View user profile'
    description: 'Show user profile. Access: MEM, ADM'
    tags:
      - 'M01: Authentication and Individual Profile'
      - 'M05: Members'
      - 'M08: Member Administration and Static Pages'

    parameters:
      - in: path
        name: username
        schema:
          type: string
        required: true

    responses:
      '200':
        description: 'Ok. Show User Profile UI (UI09)'
      '403':
        description: 'Forbidden. User is not allowed to view this profile.'
      '404':
        description: 'Not Found. User does not exist.'

  patch:
    operationId: R107
    summary: 'R107: Edit user profile action'
    description: 'Processes the user profile edit form submission. Access: OWN,
ADM'

    tags:
      - 'M01: Authentication and Individual Profile'
      - 'M08: Member Administration and Static Pages'

    parameters:
      - in: path
        name: username
        schema:

```

```

        type: string
        required: true

requestBody:
  required: true
  content:
    application/x-www-form-urlencoded:
      schema:
        type: object
        properties:
          biography:
            type: string
          picture:
            type: string
            format: binary
        required:
          - biography
          - picture

responses:
  '302':
    description: 'Redirect after processing the new user information.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Member profile updated. Redirect to user
profile.'
            value: '/members/{username}'
          302Failure:
            description: 'Member profile not updated. Redirect to user
profile edit form.'
            value: '/members/{username}/edit'
  '403':
    description: 'Forbidden. User is not allowed to edit this profile.'
  '404':
    description: 'Not Found. User does not exist.'

delete:
  operationId: R108
  summary: 'R108: Delete member account action'
  description: 'Processes the delete account form submission. Access: OWN, ADM'
  tags:
    - 'M01: Authentication and Individual Profile'
    - 'M08: Member Administration and Static Pages'
  parameters:
    - in: path
      name: username
      schema:
        type: string
      required: true
  responses:
    '302':

```

```
    description: 'Member account deleted successfully.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Member account deleted successfully. Redirect to
home page.'
            value: '/home'
      '403':
        description: 'Forbidden. User is not allowed to view this page.'
      '404':
        description: 'Not Found. User does not exist.'
```

/members/{username}/edit:

```
get:
  operationId: R109
  summary: 'R109: Edit user profile'
  description: 'Show user profile edit form. Access: OWN, ADM'
  tags:
    - 'M01: Authentication and Individual Profile'
    - 'M08: Member Administration and Static Pages'
  responses:
    '200':
      description: 'Success. Display Log-in Form (UI10)'
    '403':
      description: 'Forbidden. User is not allowed to edit this profile.'
    '404':
      description: 'Not Found. User does not exist.'
```

/members/{username}/settings:

```
get:
  operationId: R110
  summary: 'R110: Member account settings'
  description: 'Show member account settings page. Access: OWN, ADM'
  tags:
    - 'M01: Authentication and Individual Profile'
    - 'M08: Member Administration and Static Pages'

parameters:
  - in: path
    name: username
    schema:
      type: string
    required: true
  responses:
    '200':
      description: 'Success. Display Member Account Settings UI (UI08)'
    '403':
      description: 'Forbidden. User is not allowed to view this page.'
    '404':
      description: 'Not Found. User does not exist.'
patch:
```

```
operationId: R111
summary: 'R111: Edit member account settings action'
description: 'Processes the member account settings form submission. Access:
OWN, ADM'
tags:
  - 'M01: Authentication and Individual Profile'
  - 'M08: Member Administration and Static Pages'
parameters:
  - in: path
    name: username
    schema:
      type: string
    required: true
requestBody:
  required: true
  content:
    application/x-www-form-urlencoded:
      schema:
        type: object
        properties:
          firstName:
            type: string
          lastName:
            type: string
          username:
            type: string
          email:
            type: string
          oldPassword:
            type: string
          newPassword:
            type: string
          confirmNewPassword:
            type: string
responses:
  '302':
    description: 'Redirect after processing the member account settings
information.'
    headers:
      Location:
        schema:
          type: string
    examples:
      302Success:
        description: 'Member account settings updated.'
        value: '/members/{username}'
      302Failure:
        description: 'Member account settings not updated. Redirect to
member account settings page.'
        value: '/members/{username}/settings'
  '403':
    description: 'Forbidden. User is not allowed to view this page.'
  '404':
    description: 'Not Found. User does not exist.'
```


/api/members/{username}/follow:

post:

operationId: R112

summary: 'R112: Follow member action'

description: 'Processes the follow member form submission. Access: OWN'

tags:

- 'M05: Members'

parameters:

- in: path

name: username

schema:

type: string

description: 'Member to follow'

required: true

responses:

'302':

description: 'Redirect after processing the follow member request.'

headers:

Location:

schema:

type: string

examples:

302Success:

description: 'Member followed successfully. Redirect to member profile.'

value: '/members/{username}'

302Failure:

description: 'Member not followed. Redirect to member profile.'

value: '/members/{username}'

'403':

description: 'Forbidden. User is not allowed to view this page.'

'404':

description: 'Not Found. User does not exist.'

delete:

operationId: R113

summary: 'R113: Unfollow member action'

description: 'Processes the unfollow member form submission. Access: OWN'

tags:

- 'M05: Members'

parameters:

- in: path

name: username

schema:

type: string

description: 'Member to unfollow'

required: true

responses:

'200':

description: 'Member unfollowed successfully.'

'403':

description: 'Forbidden. User is not allowed to view this page.'

'404':

description: 'Not Found. User does not exist.'

/api/members/{username}/content:

get:

operationId: R114

summary: 'R114: Retrieve member content items (articles and comments)'

description: 'Retrieve member content items (articles and comments). Access:

MEM'

tags:

- 'M02: Content Items'
- 'M05: Members'

parameters:

- in: path
name: username
schema:
type: string
description: 'Member username'
required: true
- in: query
name: type
schema:
type: string
description: 'Content type (articles or comments or Both)'
- in: query
name: page
schema:
type: integer
description: 'Page number'
required: true

responses:

'200':

description: 'Successfully retrieved member content items.'

content:

application/json:

schema:

type: array

items:

oneOf:

- \$ref: '#/components/schemas/SearchArticle'
- \$ref: '#/components/schemas/SearchComment'

'400':

description: 'Bad Request. Invalid form data.'

'403':

description: 'Forbidden. User is not allowed to view this page.'

'404':

description: 'Not Found. User does not exist.'

/api/tag/{name}/follow:

post:

operationId: R115

summary: 'R115: Follow tag action'

description: 'Processes the follow tag form submission. Access: OWN'

tags:

- 'M03: Votes and Tags'

```

parameters:
  - in: path
    name: name
    schema:
      type: string
    description: 'Tag to follow'
    required: true
responses:
  '302':
    description: 'Redirect after processing the follow tag request.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Tag followed successfully. Redirect to tag page.'
            value: '/tag/{name}'
          302Failure:
            description: 'Tag not followed. Redirect to tag page.'
            value: '/tag/{name}'
  '403':
    description: 'Forbidden. User is not allowed to view this page.'
  '404':
    description: 'Not Found. Tag does not exist.'
delete:
  operationId: R116
  summary: 'R116: Unfollow tag action'
  description: 'Processes the unfollow tag form submission. Access: OWN'
  tags:
    - 'M03: Votes and Tags'
  parameters:
    - in: path
      name: name
      schema:
        type: string
      description: 'Tag to unfollow'
      required: true
  responses:
    '200':
      description: 'Tag unfollowed successfully.'
    '403':
      description: 'Forbidden. User is not allowed to view this page.'
    '404':
      description: 'Not Found. Tag does not exist.'

/:
get:
  operationId: R117
  summary: 'R117: Redirects to home page'
  description: 'Shows the home page. Access: PUB'
  tags:
    - 'M02: Content Items'
    - 'M03: Votes and Tags'

```

```
responses:
  '302':
    description: 'Redirect to home page.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302:
            description: 'Redirect to home page.'
            value: '/home'
```

```
/home:
  get:
    operationId: R118
    summary: 'R118: Shows the home page'
    description: 'Shows the home page. Access: PUB'
    tags:
      - 'M02: Content Items'
      - 'M03: Votes and Tags'
    responses:
      '200':
        description: 'Success. Display Home Page UI (UI01)'
```

```
/api/feed:
  get:
    operationId: R119
    summary: 'R119: Retrieve feed items (Top or Latest)'
    description: 'Retrieve feed items (Top or Latest). Access: PUB'
    tags:
      - 'M02: Content Items'
      - 'M03: Votes and Tags'
    parameters:
      - in: query
        name: type
        schema:
          type: string
        description: 'Feed type (Top or Latest)'
        required: true
      - in: query
        name: queryType
        schema:
          type: string
        description: 'Query type (Articles and Comments, Articles or
Comments)'
        required: true
      - in: query
        name: page
        schema:
          type: integer
        description: 'Page number'
        required: true
    responses:
      '200':
```

```

    description: 'Successfully retrieved feed items.'
    content:
      application/json:
        schema:
          type: array
          items:
            oneOf:
              - $ref: '#/components/schemas/SearchArticle'
              - $ref: '#/components/schemas/SearchComment'
  '400':
    description: 'Bad Request. Invalid form data.'
  '403':
    description: 'Forbidden. User is not allowed to do this operation.'
```

/api/feed/{username}:

```

get:
  operationId: R120
  summary: 'R120: Retrieve member feed items'
  description: 'Retrieve member feed items. Access: OWN'
  tags:
    - 'M02: Content Items'
    - 'M03: Votes and Tags'
    - 'M05: Members'
  parameters:
    - in: path
      name: username
      schema:
        type: string
        description: 'Member username'
        required: true
    - in: query
      name: contentType
      schema:
        type: string
        description: 'Content type (Articles and Comments, Articles or
Comments)'
    - in: query
      name: page
      schema:
        type: integer
        description: 'Page number'
        required: true
  responses:
    '200':
      description: 'Successfully retrieved member feed items.'
      content:
        application/json:
          schema:
            type: array
            items:
              oneOf:
                - $ref: '#/components/schemas/SearchArticle'
                - $ref: '#/components/schemas/SearchComment'
    '400':
```

```
    description: 'Bad Request. Invalid form data.'
  '403':
    description: 'Forbidden. User is not allowed to view this page.'
  '404':
    description: 'Not Found. User does not exist.'
```

/articles/create:

get:

```
  operationId: R121
  summary: 'R121: Show article creation form'
  description: 'Show article creation form. Access: MEM'
  tags:
    - 'M02: Content Items'
  responses:
    '200':
      description: 'Success. Display Article Creation Form UI (UI13)'
    '403':
      description: 'Forbidden. User is not allowed to view this page.'
```

post:

```
  operationId: R122
  summary: 'R122: Create article action'
  description: 'Processes the article creation form submission. Access: MEM'
  tags:
    - 'M02: Content Items'
  requestBody:
    required: true
    content:
      application/x-www-form-urlencoded:
        schema:
          type: object
          properties:
            title:
              type: string
            body:
              type: string
            topic:
              type: string
            tags:
              type: array
            items:
              type: string
            images:
              type: array
            items:
              type: string
            format: binary
          required:
            - title
            - body
            - topic
            - tags
            - images
  responses:
    '302':
```

```

        description: 'Redirect after processing the new article information.'
        headers:
            Location:
                schema:
                    type: string
            examples:
                302Success:
                    description: 'Article created successfully. Redirect to
article page.'
                    value: '/articles/{id}'
                302Failure:
                    description: 'Article not created. Redirect to article
creation form.'
                    value: '/articles/create'
        '400':
            description: 'Bad Request. Invalid form data.'
        '403':
            description: 'Forbidden. User is not allowed to view this page.'

/articles/{id}:
    get:
        operationId: R123
        summary: 'R123: Shows article'
        description: 'Shows article. Access: PUB'
        tags:
            - 'M02: Content Items'
            - 'M03: Votes and Tags'
            - 'M04: Topics'
        parameters:
            - in: path
              name: id
              schema:
                  type: integer
              description: 'Article id'
              required: true
        responses:
            '200':
                description: 'Success. Display Article / News Page UI (UI12)'
            '404':
                description: 'Not Found. Article page does not exist.'
    delete:
        operationId: R124
        summary: 'R124: Delete article action'
        description: 'Processes the delete article form submission. Access: OWN, ADM'
        tags:
            - 'M02: Content Items'
            - 'M08: Member Administration and Static Pages'
        parameters:
            - in: path
              name: id
              schema:
                  type: integer
              description: 'Article id'
              required: true

```

```
responses:
  '302':
    description: 'Redirect after processing the delete article
information.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Article deleted successfully. Redirect to home
page.'
            value: '/home'
          302Failure:
            description: 'Article not deleted. Redirect to article page.'
            value: '/articles/{id}'
  '403':
    description: 'Forbidden. User is not allowed to view this page.'
  '404':
    description: 'Not Found. Article does not exist.'
patch:
  operationId: R125
  summary: 'R125: Edit article action'
  description: 'Processes the article edit form submission. Access: OWN, ADM'
  tags:
    - 'M02: Content Items'
    - 'M08: Member Administration and Static Pages'
  parameters:
    - in: path
      name: id
      schema:
        type: integer
      description: 'Article id'
      required: true
  requestBody:
    required: true
    content:
      application/x-www-form-urlencoded:
        schema:
          type: object
          properties:
            title:
              type: string
            body:
              type: string
            topic:
              type: string
            tags:
              type: array
              items:
                type: string
            images:
              type: array
              items:
```



```

        type: string
        format: binary
    required:
        - title
        - body
        - topic
        - tags
        - images
    responses:
        '302':
            description: 'Redirect after processing the edit article information.'
            headers:
                Location:
                    schema:
                        type: string
            examples:
                302Success:
                    description: 'Article edited successfully. Redirect to article
page.'
                    value: '/articles/{id}'
                302Failure:
                    description: 'Article not edited. Redirect to article edit
form.'
                    value: '/articles/{id}/edit'
        '400':
            description: 'Bad Request. Invalid form data.'
        '403':
            description: 'Forbidden. User is not allowed to view this page.'
        '404':
            description: 'Not Found. Article does not exist.'

/articles/{id}/edit:
    get:
        operationId: R126
        summary: 'R126: Show article edit form'
        description: 'Show article edit form. Access: OWN, ADM'
        tags:
            - 'M02: Content Items'
            - 'M08: Member Administration and Static Pages'
        parameters:
            - in: path
              name: id
              schema:
                  type: integer
              description: 'Article id'
              required: true
        responses:
            '200':
                description: 'Success. Display Article / News Edit Form UI (UI15)'
            '403':
                description: 'Forbidden. User is not allowed to view this page.'
            '404':
                description: 'Not Found. Article does not exist.'

```

```
/articles/{id}/comments:
  get:
    operationId: R127
    summary: 'R127: Retrieve article article comments'
    description: 'Retrieve article article comments. Access: PUB'
    tags:
      - 'M02: Content Items'
    parameters:
      - in: path
        name: id
        schema:
          type: integer
        description: 'Article id'
        required: true
      - in: query
        name: page
        schema:
          type: integer
        description: 'Page number'
        required: true
    responses:
      '200':
        description: 'Successfully retrieved article comments.'
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/Comment'
      '404':
        description: 'Not Found. Article does not exist.'

/tag/{name}:
  get:
    operationId: R128
    summary: 'R128: Shows tag page'
    description: 'Shows tag page. Access: PUB'
    tags:
      - 'M03: Votes and Tags'
    parameters:
      - in: path
        name: name
        schema:
          type: string
        description: 'Tag name'
        required: true
    responses:
      '200':
        description: 'Success. Display Tag Page UI (UI11)'
      '404':
        description: 'Not Found. Tag does not exist.'

  delete:
    operationId: R129
    summary: 'R129: Delete tag action'
```

```

description: 'Processes the delete tag form submission. Access: ADM'
tags:
  - 'M03: Votes and Tags'
  - 'M08: Member Administration and Static Pages'
parameters:
  - in: path
    name: name
    schema:
      type: string
    description: 'Tag name'
    required: true
responses:
  '302':
    description: 'Redirect after processing the delete tag information.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Tag deleted successfully. Redirect to home
page.'
            value: '/home'
          302Failure:
            description: 'Tag not deleted. Redirect to tag page.'
            value: '/tag/{name}'
  '403':
    description: 'Forbidden. User is not allowed to view this page.'
  '404':
    description: 'Not Found. Tag does not exist.'

```

/api/articles/{id}/comment:

```

post:
  operationId: R130
  summary: 'R130: Create comment action'
  description: 'Processes the create comment form submission. Access: MEM'
  tags:
    - 'M02: Content Items'
    - 'M05: Members'
  parameters:
    - in: path
      name: id
      schema:
        type: integer
      description: 'Article id'
      required: true
  requestBody:
    required: true
    content:
      application/x-www-form-urlencoded:
        schema:
          type: object
          properties:
            body:

```

```

        type: string
        required:
          - body
    responses:
      '302':
        description: 'Redirect after processing the create comment
information.'
        headers:
          Location:
            schema:
              type: string
            examples:
              302Success:
                description: 'Comment created successfully. Redirect to
article page.'
                value: '/articles/{id}'
              302Failure:
                description: 'Comment not created. Redirect to article page.'
                value: '/articles/{id}'
      '400':
        description: 'Bad Request. Invalid form data.'
      '403':
        description: 'Forbidden. User is not allowed to view this page.'
      '404':
        description: 'Not Found. Article does not exist.'

/api/articles/{articleID}/comment/{commentID}:
  patch:
    operationId: R131
    summary: 'R131: Edit comment action'
    description: 'Processes the edit comment form submission. Access: OWN, ADM'
    tags:
      - 'M02: Content Items'
      - 'M08: Member Administration and Static Pages'
    parameters:
      - in: path
        name: articleID
        schema:
          type: integer
        description: 'Article id'
        required: true
      - in: path
        name: commentID
        schema:
          type: integer
        description: 'Comment id'
        required: true
    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:

```

```

        body:
          type: string
        required:
          - body
      responses:
        '200':
          description: 'Comment edited successfully.'
        '400':
          description: 'Bad Request. Invalid form data.'
        '403':
          description: 'Forbidden. User is not allowed to view this page.'
        '404':
          description: 'Not Found. Article or comment does not exist.'
  delete:
    operationId: R130
    summary: 'R130: Delete comment action'
    description: 'Processes the delete comment form submission. Access: OWN, ADM'
    tags:
      - 'M02: Content Items'
      - 'M08: Member Administration and Static Pages'
    parameters:
      - in: path
        name: articleID
        schema:
          type: integer
        description: 'Article id'
        required: true
      - in: path
        name: commentID
        schema:
          type: integer
        description: 'Comment id'
        required: true
    responses:
      '302':
        description: 'Redirect after processing the delete comment
information.'
        headers:
          Location:
            schema:
              type: string
            examples:
              302Success:
                description: 'Comment deleted successfully. Redirect to
article page.'
                value: '/articles/{articleID}'
              302Failure:
                description: 'Comment not deleted. Redirect to article
page.'
                value: '/articles/{articleID}'
        '403':
          description: 'Forbidden. User is not allowed to view this page.'
        '404':
          description: 'Not Found. Article or comment does not exist.'

```

```

/api/articles/{articleID}/comment/{commentID}/reply:
post:
  operationId: R132
  summary: 'R132: Create reply action'
  description: 'Processes the create reply form submission. Access: MEM'
  tags:
    - 'M02: Content Items'
  parameters:
    - in: path
      name: articleID
      schema:
        type: integer
      description: 'Article id'
      required: true
    - in: path
      name: commentID
      schema:
        type: integer
      description: 'Comment id'
      required: true
  requestBody:
    required: true
    content:
      application/x-www-form-urlencoded:
        schema:
          type: object
          properties:
            body:
              type: string
            required:
              - body
  responses:
    '302':
      description: 'Redirect after processing the create reply
information.'
      headers:
        Location:
          schema:
            type: string
          examples:
            302Success:
              description: 'Reply created successfully. Redirect to
article page.'
              value: '/articles/{articleID}'
            302Failure:
              description: 'Reply not created. Redirect to article page.'
              value: '/articles/{articleID}'
    '400':
      description: 'Bad Request. Invalid form data.'
    '403':
      description: 'Forbidden. User is not allowed to view this page.'
    '404':
      description: 'Not Found. Article or comment does not exist.'

```

/search:

get:

operationId: R133

summary: 'R133: Shows search results page'

description: 'Shows search results page. Access: PUB'

tags:

- 'M07: Search'

parameters:

- in: query

name: query

schema:

type: string

description: 'Search query'

- in: query

name: type

schema:

type: string

description: 'Search type (Article or Comments or Tags or Members)'

- in: query

name: exactMatch

schema:

type: boolean

description: 'Exact match'

responses:

'200':

description: 'Success. Display Search Results Page UI (UI16)'

/api/search:

get:

operationId: R134

summary: 'R134: Search action. This is used to search for article, comments, tags and members'

description: 'Search action. This is used to search for article, comments, tags and members. Access: PUB'

tags:

- 'M07: Search'

parameters:

- in: query

name: query

schema:

type: string

description: 'Search query'

required: true

- in: query

name: type

schema:

type: string

default: articles_comments

description: 'Search type (Article or Comments or Tags or Members)'

- in: query

name: page

schema:

type: integer

```

        default: 1
        description: 'Page number'
      - in: query
        name: exactMatch
        schema:
          type: boolean
        description: 'Exact match'
    responses:
      '200':
        description: 'Successfully retrieved search results.'
        content:
          application/json:
            schema:
              type: array
              items:
                oneOf:
                  - $ref: '#/components/schemas/SearchArticle'
                  - $ref: '#/components/schemas/SearchComment'
                  - $ref: '#/components/schemas/SearchTag'
                  - $ref: '#/components/schemas/SearchMember'

/administration:
  get:
    operationId: R135
    summary: 'R135: Shows administration page'
    description: 'Shows administration page. Access: ADM'
    tags:
      - 'M08: Member Administration and Static Pages'
    responses:
      '200':
        description: 'Success. Display Administration Page UI'
      '403':
        description: 'Forbidden. User is not allowed to view this page.'

/administration/create_member:
  get:
    operationId: R136
    summary: 'R136: Shows create member form'
    description: 'Shows create member form. Access: ADM'
    tags:
      - 'M08: Member Administration and Static Pages'
    responses:
      '200':
        description: 'Success. Display Create Member Form UI'
      '403':
        description: 'Forbidden. User is not allowed to view this page.'
  post:
    operationId: R137
    summary: 'R137: Create member action'
    description: 'Processes the create member form submission. Access: ADM'
    tags:
      - 'M08: Member Administration and Static Pages'
    requestBody:
      required: true
      content:

```



```

    application/x-www-form-urlencoded:
      schema:
        type: object
        properties:
          firstName:
            type: string
          lastName:
            type: string
          username:
            type: string
          email:
            type: string
          password:
            type: string
          confirmPassword:
            type: string
        required:
          - firstName
          - lastName
          - username
          - email
          - password
          - confirmPassword
      responses:
        '302':
          description: 'Redirect after processing the new member information.'
          headers:
            Location:
              schema:
                type: string
          examples:
            302Success:
              description: 'Member created successfully. Redirect to
member profile.'
              value: '/members/{username}'
            302Failure:
              description: 'Member not created. Redirect to create member
form.'
              value: '/administration/create_member'

components:
  schemas:
    SearchMember:
      type: object
      properties:
        id:
          type: integer
        username:
          type: string
        firstName:
          type: string
        lastName:
          type: string
        picture:

```

```
    type: string
  email:
    type: string
  academyScore:
    type: integer
SearchArticle:
  type: object
  properties:
    id:
      type: integer
    title:
      type: string
    body:
      type: string
    date:
      type: string
      format: date-time
    topic:
      type: string
    tags:
      type: array
      items:
        type: string
    academyScore:
      type: integer
    author:
      $ref: '#/components/schemas/Author'
SearchComment:
  type: object
  properties:
    id:
      type: integer
    body:
      type: string
    date:
      type: string
      format: date-time
    isReply:
      type: boolean
    replyTo:
      type: integer
      nullable: true
    academyScore:
      type: integer
    author:
      $ref: '#/components/schemas/Author'
SearchTag:
  type: object
  properties:
    name:
      type: string
Author:
  type: object
  properties:
```

```
  id:
    type: integer
  username:
    type: string
  firstName:
    type: string
  lastName:
    type: string
Article:
  type: object
  properties:
    id:
      type: integer
    title:
      type: string
    body:
      type: string
    date:
      type: string
      format: date-time
    topic:
      type: string
    tags:
      type: array
      items:
        type: string
    academyScore:
      type: integer
Comment:
  type: object
  properties:
    id:
      type: integer
    body:
      type: string
    date:
      type: string
      format: date-time
    isReply:
      type: boolean
    replyTo:
      type: integer
      nullable: true
    academyScore:
      type: integer
    author:
      $ref: '#/components/schemas/Author'
    replies:
      type: array
      items:
        $ref: '#/components/schemas/Comment'
```

A8: Vertical prototype

The Vertical Prototype involves incorporating high-priority features identified as essential (marked with an asterisk) in both the common and theme requirements documents. This component is designed to validate the presented architecture and facilitate familiarity with the project's employed technologies.

The implementation is grounded in the LBAW Framework, encompassing work across all layers of the solution's architecture, including the user interface, business logic, and data access. The prototype encompasses the development of visualization, insertion, editing, and removal pages for information. It also addresses permission control for accessing these pages, along with the presentation of error and success messages.

1. Implemented Features

1.1. Implemented User Stories

User Story reference	Name	Priority	Description
US001	View Top News	High	As a user, I want a page of top news, so that I can view the most upvoted news in a selected time frame.
US002	View News Items	High	As a user, I want to access news items, so that I can see articles in further detail, such as their full extent, comments and related news.
US003	Exact Match Search	High	As a user, I want to use exact match search, so that I can quickly and precisely find the specific information or content I'm looking for.
US004	Full-Text Search	High	As a user, I want to use full-text search, so that I can find a variety of differently related content.
US005	View Recent News	Medium	As a user, I want a page of recent news, so that I can view the latest news.
US006	View News Item Comments	Medium	As a user, I want to view news item comments, so that I can stay informed about public opinion and discussions about current events.
US006	View News Item Comments	Medium	As a user, I want to view news item comments, so that I can stay informed about public opinion and discussions about current events.
US007	Search for News Items	Medium	As a user, I want to search for news items, so that I can quickly find articles on specific themes of interest.
US008	Search for Comments	Medium	As a user, I want to search for comments, so that I find members' opinions on specific themes of interest.

User Story reference	Name	Priority	Description
US011	Placeholders In Form Inputs	Medium	As a user, I want to have placeholders in form inputs, so that filling different forms(e.g.: creating news articles, registration, etc.) is more intuitive.
US012	Contextual Error Messages	Medium	As a user, I want to receive contextual error messages, so that I can understand why an action or input failed and take appropriate corrective actions efficiently.
US021	Search for Tags	Low	As a user, I want to search for tags, so that I can easily find and explore specific themes.
US022	Search for Users	Low	As a user, I want to search for users, so that I can easily find and explore content by specific users.
US101	Login	High	As a guest, I want to login into the platform, so that I can access my personalized content and features.
US102	Register	High	As a guest, I want to register on the platform, so that I can create an account where I can have a more personalized experience with the platform.
US201	Logout	High	As a member, I want to log out of the platform, so that my account privacy is secured.
US202	View News Item Details	High	As a member, I want to view news item details, so that I can see its author, publication date, votes, and other exclusive information.
US203	Member News Feed	High	As a member, I want to access a customized news feed, so that I can view content related to my interests (i.e., tags and members I follow).
US204	View Other Members' Profiles	High	As a member, I want to view other members' profiles, so that I can see news articles they authored and follow them.
US205	View Profile	High	As a member, I want to view my profile, so that I can review and manage my personal information, preferences, and activity on the platform.
US206	Edit Profile	High	As a member, I want to edit my profile, so that I can update my personal information, preferences, and other details to ensure my profile reflects my current status and interests accurately.
US207	Create News Items	High	As a member, I want to create news items, so that I can write about and share recent news.

User Story reference	Name	Priority	Description
US209	View Comment Details	Medium	As a member, I want to view comment details, so that I can view its author, publication date, votes, and others.
US212	View Reputation of Other Members	Medium	As a member, I want to view the reputation of other members, so that I can evaluate a member's content liability.
US213	View News Item Tags	Medium	As a member, I want to view news item tags, so that I can follow tags of interest or view relevant news related to them.
US301	Edit News Items	High	As a news contributor, I want to edit news items authored by me, so that I can enhance the article information or correct possible inaccuracies.
US302	Delete News Items	High	As a news contributor, I want to delete news items authored by me, so that I can remove any articles I've submitted that are no longer relevant or accurate.
US501	Administer Member Accounts	High	As an administrator, I want to administer member accounts (i.e., search, view, edit, create), so that I can effectively manage and secure access to the platform resources and systems.
US503	Admin Account Privileges	Medium	As an administrator, I want to have an account with administrating privileges, so that I can oversee and manage the platform.
US505	Delete Member Accounts	Medium	As an administrator, I want to delete member accounts, so that members deemed undeserving of remaining in the platform are removed (e.g.: misconduct, spam accounts, etc.).

1.2. Implemented Web Resources

Module M01: Authentication and Individual Profile

Web Resource Reference	URL
R101: Login Form	GET/ login
R102: Login Authentication	POST/login
R103: Logout Action	GET/ logout
R104: Register Form	GET/ register
R105: Register Action	POST/register
R106: View user profile	GET/ members/{username}

Web Resource Reference	URL
R107: Edit user profile action	PATCH/members/{username}
R108: Delete member account action	DELETE/members/{username}
R109: Edit user profile	GET/ members/{username}/edit
R110: Member account settings	GET/ members/{username}/settings
R111: Edit member account settings action	PATCH/members/{username}/settings

Module M02: Content Items

Web Resource Reference	URL
R114: Retrieve member content items (articles and comments)	GET/ api/members/{username}/content
R117: Redirects to home page	GET/
R118: Shows the home page	GET/ home
R119: Retrieve feed items (Top or Latest)	GET/api/feed
R120: Retrieve member feed items	GET/api/feed/{username}
R121: Show article creation form	GET/ articles/create
R122: Create article action	POST/articles/create
R123: Shows article	GET/ articles/{id}
R124: Delete article action	DELETE/articles/{id}
R125: Edit article action	PATCH/articles/{id}
R126: Show article edit form	GET/ articles/{id}/edit
R127: Retrieve article comments	GET/articles/{id}/comments

Module M03: Votes and Tags

Web Resource Reference	URL
R117: Redirects to home page	GET/
R118: Shows the home page	GET/ home
R119: Retrieve feed items (Top or Latest)	GET/api/feed
R120: Retrieve member feed items	GET/api/feed/{username}
R123: Shows article	GET/ articles/{id}

Module M04: Topics

Web Resource Reference	URL
------------------------	-----

Web Resource Reference	URL
R123: Shows article	GET/ articles/{id}

Module M05: Members

Web Resource Reference	URL
R106: View user profile	GET/ members/{username}
R114: Retrieve member content items (articles and comments)	GET/api/members/{username}/content
R120: Retrieve member feed items	GET/api/feed/{username}

Module M06: Notifications

(No specific web resources implementer for the prototype)

Module M07: Search

Web Resource Reference	URL
R133: Shows search results page	GET/ search
R134: Search action. This is used to search for article, comments, tags and members	GET/api/search

Module M08: Member Administration and Static Pages

Web Resource Reference	URL
R106: View user profile	GET/ members/{username}
R107: Edit user profile action	PATCH/members/{username}
R108: Delete member account action	DELETE/members/{username}
R109: Edit user profile	GET/ members/{username}/edit
R110: Member account settings	GET/ members/{username}/settings
R111: Edit member account settings action	PATCH/members/{username}/settings
R124: Delete article action	DELETE/articles/{id}
R125: Edit article action	PATCH/articles/{id}
R126: Show article edit form	GET/ articles/{id}/edit
R135: Shows administration page	GET/ administration
R136: Shows create member form	GET/ administration/create_member
R137: Create member action	POST/administration/create_member

2. Prototype

The prototype is available at [The Popcorn Post](#).

Administrator credentials:

- *email*: jessica.pearson@popcornpost.com
- *password*: 1234

Member credentials

- *email*: harvey.specter@popcornpost.com
- *password*: 1234

The source code is available [here](#).

Revision history

Changes made to the first submission:

1. Item 1
 2. ..
-

GROUP2356, 26/10/2023

- João Filipe Oliveira Ramos, up202108743@up.pt (Editor)
- Tiago Filipe Castro Viana, up201807126@up.pt
- Marco André Pereira da Costa, up202108821@up.pt
- Diogo Alexandre Figueiredo Gomes, up201905991@up.pt