

# EAP: Architecture Specification and Prototype

Bidding the future and Selling the past.

## A7: Web Resources Specification

This artifact documents the architecture of the web application to be developed, indicating the catalog of resources, the properties of each resource, and the format of JSON responses. This specification adheres to the OpenAPI standard using YAML.

This artifact presents the documentation for InfinityAuctions, including the CRUD (create, read, update, delete) operations for each resource.

### 1. Overview

Identify and overview the modules that will be part of the application.

| Model                                   | Description  |
|---|--|
| M01:<br>Registration and authentication | This module provides user authentication resources. The following resources are provided: login/logout, registration and password recovery.  |
| M02: Home Page and Static pages         | This module provides static content and Home page layout. The following resources are provided: Home page, about page, services page, FAQ page, contacts page.   |
| M03: Platform administration            | This module provides administration resources. The following resources are provided: view auction reports, view user reports , manage reports ,view categories, manage categories.   |
| M04: User                               | This module provides account management and user profile resources. The following resources are provided: view/manage personal account information, view user profile, add/withdraw funds to/from personal account, change password.   |
| M05: Auctions                           | This module provides resources associated with auction search and auction management. The following resources are provided: search engine, create/edit/cancel auctions, view auction information, manage auctions and follow auctions. |
| M06: Reports                            | This module handles auction and user reporting resources. The following resources are provided: report user, report auctions.  |
| M07: Notifications                      | This module handles system notifications.The following resources are provided: auction notifications, bid notifications and wishlist notifications.  |

### 2. Permissions

Define the permissions used by each module, necessary to access its data and features.

| Identifier | Name           | Description  |
|------------|----------------|--|
| PUB        | Public         | Users without privileges   |
| USR        | User           | Authenticated Users  |
| BDR        | Bidder         | Authenticated users that bid on auctions                                 |
| OWN        | Auction Owners | Authenticated users that own auctions (including media and descriptions) |
| ADM        | Administrator  | System administrators (act on reports, sanctions auctions and users)     |

### 3. OpenAPI Specification

OpenAPI specification in YAML format to describe the vertical prototype's web resources.

Link to the [a7\\_openapi.yaml](#) file in the group's repository.

[https://git.fe.up.pt/lbaw/lbaw2223/lbaw2271/-/blob/main/a7\\_openapi.yaml](https://git.fe.up.pt/lbaw/lbaw2223/lbaw2271/-/blob/main/a7_openapi.yaml)

```
openapi: 3.0.0

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

servers:
  - url: https://lbaw2271.lbaw.fe.up.pt/
    description: Production server

externalDocs:
  description: Find more info here.
  url: https://git.fe.up.pt/lbaw/lbaw2223/lbaw2271/-/blob/main/a7_openapi.yaml

tags:
  - name: 'M01: Registration and authentication'
  - name: 'M02: Home page and Static pages'
  - name: 'M03: Platform administration'
  - name: 'M04: Users'
  - name: 'M05: Auctions'
  - name: 'M06: Reports'
  - name: 'M07: Notifications'

paths:
  # M01: Registration and authentication
  /login:
    get:
      operationId: R101
```

```

summary: 'R101: Login Form.'
description: 'Provide login form UI. Access: PUB.'
tags:
  - 'M01: Registration and authentication'
responses:
  '200':
    description: 'Ok. Show Log-in UI [UI02].'
```

post:

```

  operationId: R102
  summary: 'R102: Login Action.'
  description: 'Processes the login form submission. Access: PUB.'
  tags:
    - 'M01: Registration and authentication'
```

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: '/'
      302Error:
        description: 'Failed authentication. Redirect to login
form.'
```

```

        value: '/login'
```

/logout:

```

  post:
    operationId: R103
    summary: 'R103: Logout Action.'
    description: 'Logout the current authenticated user. Access: USR,
ADM.'
```

tags:

```

  - 'M01: Registration and authentication'
```

```
responses:
  '302':
    description: 'Redirect after processing logout.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Successful logout. Redirect to Home page.'
            value: '/'
          302Error:
            description: 'Failed to logout. Redirect to Home page.'
            value: '/'

/register:
  get:
    operationId: R104
    summary: 'R104: Register Form.'
    description: 'Provide new user registration form UI. Access: PUB.'
    tags:
      - 'M01: Registration and authentication'
    responses:
      '200':
        description: 'Ok. Show Sign-Up UI [UI04].'

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

  requestBody:
    required: true
    content:
      application/x-www-form-urlencoded:
        schema:
          type: object
          properties:
            name:
              type: string
            gender:
              type: string
              enum: ['M', 'F', 'NB', 'O']
            phone:
              type: string
            email:
              type: string
              format: email
            birthdate:
              type: string
              format: date
```

```

        address:
          type: string
        password:
          type: string
          format: password
        confirm_password:
          type: string
          format: password
      required:
        - name
        - gender
        - phone
        - email
        - birthdate
        - address
        - password
        - confirm_password

    responses:
      '302':
        description: 'Redirect after processing new user information.'
        headers:
          Location:
            schema:
              type: string
        examples:
          302Success:
            description: 'Successful Registration. Redirect to user
profile.'
            value: '/'
          302Failure:
            description: 'Failed Registration. Redirect to Register
Form.'
            value: '/register'

# M02: Home page and Static pages
/:
  get:
    operationId: R201
    summary: 'R201: Home page.'
    description: 'The Home page is the center of navigation of the web
app. Access: PUB.'
    tags:
      - 'M02: Home page and Static pages'
    responses:
      '200':
        description: 'Ok. Show Home Page UI [UI01].'

/about:
  get:
    operationId: R202
    summary: 'R202: About page.'
    description: 'About Page containing info about our webpage. Access:
PUB.'
```

```
tags:
  - 'M02: Home page and Static pages'
responses:
  '200':
    description: 'Ok. Show about us UI [UI05].'
```

**/services:**

```
get:
  operationId: R203
  summary: 'R203: Services page.'
  description: 'Services Page containing info about the services we
provide. Access: PUB.'
  tags:
    - 'M02: Home page and Static pages'
  responses:
    '200':
      description: 'Ok. Show Services Page UI [UI07].'
```

**/faq:**

```
get:
  operationId: R204
  summary: 'R204: FAQ page.'
  description: 'FAQ page containing the most frequently asked
questions. Access: PUB.'
  tags:
    - 'M02: Home page and Static pages'
  responses:
    '200':
      description: 'Ok. Show FAQ UI [UI06].'
```

**/contacts:**

```
get:
  operationId: R205
  summary: 'R205: Contacts page.'
  description: 'This webPage provides the contacts and social media of
the webApp creators. Access: PUB.'
  tags:
    - 'M02: Home page and Static pages'
  responses:
    '200':
      description: 'Ok. Show contacts page UI [UI08].'
```

# M03: Platform administration

**/manage:**

```
get:
  operationId: R301
  summary: 'R301: Admin Panel.'
  description: "Show admin control panel. Access: ADM. NOTE: All the
methods the admin uses to cancel or edit auctions are already indicated in
M05: Auctions. Due to the reporting module(M06: Reports) not having top
priority, we did not include it in the prototype, thus limiting the
administrator's ability to manage the platform which depends on reports.
Therefore, this module is not as developed as the others."
  tags:
    - 'M03: Platform administration'
```

```
responses:
  '200':
    description: 'Ok. Show Admin Panel UI [UI22].'
```

# M04: User

```
/user/{id}:
get:
  operationId: R401
  summary: 'R401: My profile UI.'
  description: 'Show signed user profile, including personal
information, form to edit personal information, bidding activity, owned
auctions and followed auctions. Access: USR.'
  tags:
    - 'M04: Users'

  parameters:
    - in: path
      name: id
      required: true
      schema:
        type: integer
        minimum: 1

  responses:
    '200':
      description: 'Ok. Show My Profile UI [UI14]. Edit profile section
UI [UI18]. Bidding History section UI [UI20]. Followed auctions section UI
[UI17]. Owned auctions section UI [UI09]'
```

```
    '403':
      description: 'Error user does not have permissions.'
```

```
post:
  operationId: R402
  summary: 'R402: Edit profile.'
  description: 'Process new profile information. Access: USR, ADM.'
  tags:
    - 'M04: Users'

  parameters:
    - in: path
      name: id
      required: true
      schema:
        type: integer
        minimum: 1

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

```
        name:
          type: string
        phone:
          type: string
        email:
          type: string
          format: email
        birthdate:
          type: string
          format: date
        address:
          type: string
      required:
        - name
        - phone
        - email
        - birthdate
        - address

    responses:
      '302':
        description: 'Redirect after processing new user information.'
        headers:
          Location:
            schema:
              type: string
        examples:
          302Success:
            description: 'Successful edit. Redirect to user profile.'
            value: '/user/{id}'
          302Failure:
            description: 'Failed to edit. Redirect to user Form.'
            value: '/user/{id}'
      '403':
        description: 'Error user does not have permissions.'
      '500':
        description: 'Invalid database parameters.'

/users:
  get:
    operationId: R403
    summary: 'R403: List Users.'
    description: 'Show list of users that are registered in the platform.'
  Access: PUB.'
  tags:
    - 'M04: Users'

  responses:
    '200':
      description: 'Ok. List users UI [UI14].'
```

```
/users/{id}:
  get:
    operationId: R404
```



```
summary: 'R404: View user profile.'
description: 'Show user profile. Access: PUB.'
tags:
  - 'M04: Users'

parameters:
  - in: path
    name: id
    schema:
      type: integer
      minimum: 1
      required: true

responses:
  '200':
    description: 'Ok. Show User Profile UI.'
```

# M05: Auctions

```
/auctions/{auction_id}:
get:
  operationId: R501
  summary: 'R501: View Auction.'
  description: 'Show auction page. Access: PUB.'
  tags:
    - 'M05: Auctions'

  parameters:
    - in: path
      name: auction_id
      schema:
        type: integer
        minimum: 1
        required: true

  responses:
    '200':
      description: 'Ok. Show auction page UI [UI10]'
```

```
/sell:
get:
  operationId: R502
  summary: 'R502: Create Auction page.'
  description: 'Fetch Create Auction Form. Access: USR.'
  tags:
    - 'M05: Auctions'

  responses:
    '200':
      description: 'Ok. Show Create Auction Form UI [UI11].'
```

```
post:
  operationId: R503
  summary: 'R503: Create Auction action.'
  description: 'Processes the creation form submission. Access: USR.'
```

```
tags:
  - 'M05: Auctions'

requestBody:
  required: true
  content:
    application/x-www-form-urlencoded:
      schema:
        type: object
        properties:
          title:
            type: string
          description:
            type: string
          images:
            type: array
            items:
              type: string
              format: binary
          base_price:
            type: integer
          start_date:
            type: string
            format: date-time
          end_date:
            type: string
            format: date-time
          buy_now:
            type: integer
            nullable: true
          categories:
            type: array
            items:
              type: boolean
        required:
          - title
          - description
          - images
          - base_price
          - start_date
          - end_date
          - buy_now
          - categories

responses:
  '302':
    description: 'Redirect after creating auction.'
    headers:
      Location:
        schema:
          type: string
    examples:
      302Success:
        description: 'Successfully created auction. Redirect to
```

```
auction page.'
    value: '/auction/{id}'
    302Failure:
      description: 'Failed to create auction. Redirect create
auction page.'
    value: '/sell'
    '403':
      description: 'Error admins do not create auctions.'

/auctions/edit/{id}:
  get:
    operationId: R504
    summary: 'R504: Edit Auction page.'
    description: 'Fetch form to edit the auction. Access: OWN, ADM'
    tags:
      - 'M05: Auctions'

    parameters:
      - in: path
        name: id
        schema:
          type: integer
          minimum: 1
          required: true

    responses:
      '200':
        description: 'Show edit form UI [UI12].'

  post:
    operationId: R505
    summary: 'R505 Edit Auction Action.'
    description: 'Processes form in order to edit the auction. Access:
OWN, ADM.'

    tags:
      - 'M05: Auctions'

    parameters:
      - in: path
        name: id
        schema:
          type: integer
          minimum: 1
          required: true

    requestBody:
      required: true
      content:
        application/x-www-form-urlencoded:
          schema:
            type: object
            properties:
              title:
```

```

        type: string
      description:
        type: string
      base_price:
        type: integer
      start_date:
        type: string
        format: date-time
      end_date:
        type: string
        format: date-time
      buy_now:
        type: integer
        nullable: true
      required:
        - title
        - description
        - base_price
        - start_date
        - end_date
        - buy_now

    responses:
      '302':
        description: 'Redirect after editing auction.'
        headers:
          Location:
            schema:
              type: string
          examples:
            302Success:
              description: 'Successfully edited auction. Redirect to
auction page.'
              value: '/auction/{id}'
            302Failure:
              description: 'Failed to edit auction. Redirect edit
Auction page.'
              value: '/auction/edit/{id}'
      '403':
        description: 'Error user does not have permissions.'

/auctions/cancel/{id}:
  post:
    operationId: R506
    summary: 'R506: Cancel Auction.'
    description: 'Enables Owner to set auction state to cancelled.
Access: OWN, ADM.'
    tags:
      - 'M05: Auctions'

    parameters:
      - in: path
        name: id
        schema:

```

```
    type: integer
    minimum: 1
    required: true

responses:
  '302':
    description: 'Redirect after cancelling the auction.'
    headers:
      Location:
        schema:
          type: string
        examples:
          302Success:
            description: 'Successfully canceled. Redirect to auction
page.'
            value: '/'
          302Failure:
            description: 'Failed to cancel. Redirect to auction
page.'
            value: '/auction/{id}'
  '403':
    description: 'Error user does not have permissions.'

/api/auctions/getAllBids/{auction_id}:
get:
  operationId: R507
  summary: 'R507: Auctions bids'
  description: 'List the bids of a certain auction. Access: PUB'
  tags:
    - 'M05: Auctions'

  parameters:
    - in: path
      name: auction_id
      schema:
        type: integer
        minimum: 1
        required: true

  responses:
    '200':
      description: 'Successfully fetched data.'
      content:
        application/json:
          schema:
            type: array
            items:
              type: object
              properties:
                bid_id:
                  type: string
                  minimum: 1
                date:
                  type: string
```

```
        format: date-time
    amount:
      type: integer
      minimum: 1
    user_id:
      type: integer
      minimum: 1
    auction_id:
      type: integer
      minimum: 1
    username:
      type: string
  required:
    - bid_id
    - date
    - amount
    - user_id
    - auction_id
    - username
  example:
    - bid_id: 100
      date: '2021-07-11 00:00:00+00'
      amount: 300
      user_id: 4
      auction_id: 9
      username: Manel
    - bid_id: 32
      date: '2021-07-09 00:00:00+00'
      amount: 200
      user_id: 2
      auction_id: 1
      username: Maria
```

#### /api/auctions:

```
  post:
    operationId: R508
    summary: 'R508: Bid on auction.'
    description: 'User bids on the auction. Access: BDR.'
    tags:
      - 'M05: Auctions'

  requestBody:
    required: true
    content:
      application/json:
        schema:
          type: object
          properties:
            user_id:
              type: integer
              minimum: 1
            amount:
```

```

        type: integer
        minimum: 1
    auction_id:
        type: integer
        minimum: 1
    required:
        - user_id
        - value
        - auction_id

responses:
  '302':
    description: 'Redirect after bidding on the auction.'
    headers:
      Location:
        schema:
          type: string
    examples:
      302Success:
        description: 'Successfull bid. Redirect to auction page.'
        value: '/auction/{id}'
      302Failure:
        description: 'Failed to make a bid. Redirect to auction
page.'
        value: '/auction/{id}'
  '403':
    description: 'Error user does not have permissions.'

/api/search:
  get:
    operationId: R509
    summary: 'R509: Search auctions api.'
    description: 'This resource implements a search engine used to find
specific auctions. User can search by text and filter the results based on
the auction category. If no parameters get passed, returns all auctions.
Access: PUB.'

    tags:
      - 'M05: Auctions'

    parameters:
      - in: query
        name: search
        schema:
          type: string
        required: false
        examples:
          searchQuery:
            description: 'Full text search of an item that matches
computer dell.'
            value: 'computer dell' # ?search=computer+dell
      - in: query
        name: category
        schema:

```

```

    type: array
    items:
      type: integer
    required: false
    examples:
      oneFilter:
        description: 'Getting the results of one category.'
        value: [1] # ?category%5B0%5D=1
      multipleFilters:
        description: 'Getting results matching many categories id.'
        value: [1, 5, 7] # ?
category%5B0%5D=1&category%5B1%5D=5&category%5B2%5D=7

responses:
  '200':
    description: 'Successfully fetched data.'
    content:
      application/json:
        schema:
          type: array
          items:
            type: object
            properties:
              id:
                type: string
                minimum: 1
              name:
                type: string
              description:
                type: string
              base_price:
                type: integer
              start_date:
                type: string
                format: date-time
              end_date:
                type: string
                format: date-time
              buy_now:
                type: integer
                nullable: true
              state:
                type: string
                enum: ['Cancelled', 'Running', 'To be started',
'Ended']

              auction_owner_id:
                type: integer
            required:
              - id
              - name
              - description
              - base_price
              - start_date
              - end_date

```



```

- state
- buy_now
- auction_owner_id
example:
- id: 22
  name: Adidas shirt
  description: Adidas shirt. Good state.
  base_price: 20
  start_date: '2021-07-11 00:00:00+00'
  end_date: '2021-10-01 00:00:00+00'
  state: 'Ended'
  buy_now: 300
  auction_owner_id: 4
- id: 45
  name: World cup Football
  description: Adidas Football. Good state.
  base_price: 100
  start_date: '2021-07-11 00:00:00+00'
  end_date: '2021-10-01 00:00:00+00'
  buy_now: null
  state: 'Running'
  auction_owner_id: 9

```

#### /search:

##### get:

operationId: R510

summary: 'R510: Search auctions UI.'

description: 'This resource passes the filtering parameters to the search API, receiving and listing the results and the respective filters on the web page. Access: PUB.'

##### tags:

- 'M05: Auctions'

##### parameters:

- in: query

name: search

schema:

type: string

required: false

examples:

searchQuery:

description: 'Full text search of an item that matches computer dell.'

value: 'computer dell' # ?search=computer+dell

- in: query

name: category

schema:

type: array

items:

type: integer

required: false

examples:

oneFilter:

```

        description: 'Getting the results of one category.'
        value: [1] # ?category%5B0%5D=1
    multipleFilters:
        description: 'Getting results matching many categories id.'
        value: [1, 5, 7] # ?
category%5B0%5D=1&category%5B1%5D=5&category%5B2%5D=7

    responses:
        '200':
            description: 'List search results UI [UI13].'
```

**/api/image/{id}:**

```

    delete:
        operationId: R511
        summary: 'R511: Delete auction image.'
        description: 'This api deletes an image that is associated with a
specific auction. Access: OWN, ADM'
        tags:
            - 'M05: Auctions'

    parameters:
        - in: path
          name: auction_id
          schema:
              type: integer
              minimum: 1
              required: true

    responses:
        '200':
            description: 'Ok. Image was successfully deleted.'
```

## A8: Vertical prototype

The Vertical Prototype includes the implementation of the features marked as necessary (with an asterisk) in the common and theme requirements documents. This artifact aims to validate the architecture presented, also serving to gain familiarity with the technologies used in the project.

The implementation is based on the LBAW Framework and includes work on all layers of the architecture of the solution to implement: user interface, business logic and data access. The prototype includes the implementation of pages of visualization, insertion, edition and removal of information, the control of permissions in the access to the implemented pages, and the presentation of error and success messages.

### 1. Implemented Features

#### 1.1. Implemented User Stories

Identify the user stories that were implemented in the prototype.

| User Story reference | Name                  | Priority | Description  |
|----------------------|-----------------------|----------|--|
| US01                 | Sign-in               | High     | As a Visitor, I want to authenticate into the system, so that I can access privileged information.   |
| US02                 | Sign-up               | High     | As a Visitor, I want to register myself into the system, so that I can authenticate myself into the system.  |
| US11                 | See Home page         | High     | As a User, I want to access the Home page, so that I can see a brief presentation of the website.  |
| US12                 | See About page        | High     | As a User, I want to access the About page, so that I can see a complete description of the website and its creators.  |
| US14                 | Consult FAQ           | High     | As a User, I want to access the FAQ, so that I can get quick answers to common questions.  |
| US15                 | Consult Contacts      | High     | As a User, I want to access contacts, so that I can come in touch with the platform creators.  |
| US16                 | Search                | High     | As a User, I want to search the platform keywords, so that I can quickly find goods and services that I am looking for.                                      |
| US17                 | Filters               | Medium   | As a User, I want to search an auction within a certain criteria, so that I can quickly find goods and services that I am looking for.                       |
| US21                 | Most active auctions  | Medium   | As a User, I want to have a shortcut to the most active auctions in the platform, so that I can see what is currently popular.                               |
| US22                 | New auctions          | Medium   | As a User, I want to have a shortcut to the most recent auctions in the platform, so that I can see the new business opportunities.                          |
| US23                 | Social networks       | Low      | As a User, I want to have a shortcut to the platform's social media accounts.  |
| US24                 | More from this seller | Low      | As a User, I want to access other auctions by the same seller whose auction I am on, so that I can check if the seller has any other item of interest to me. |
| US31                 | Create auctions       | High     | As an Authenticated user, I want to create auctions, so that I can auction a product to other users.   |
| US32                 | Bid an auction        | High     | As an Authenticated user, I want to bid on auctions, so that I can set a new price for the product I am willing to buy.                                      |
| US33                 | View activity         | High     | As an Authenticated user, I want to view my account history, so that I can consult previous auctions, as well as those that I am bidding on or following.    |

| User Story reference | Name                  | Priority | Description  |
|----------------------|-----------------------|----------|--|
| US34                 | Edit account          | High     | As an Authenticated user, I want to be able to edit my account, so that I can keep my personal data updated.   |
| US35                 | Bidding history       | High     | As a Bidder, I want to have access to the bidding history of a certain auction, so that I can study my approach to it and maximize the chances of winning it with a smaller bid. |
| US36                 | View Users profile    | High     | As an Authenticated user, I want to be able to see other users' profiles, so that I can access their information.  |
| US51                 | Update bid            | High     | As a Bidder, I want to update the bid, so that I can set a new competitive price for the product.  |
| US61                 | Edit auctions         | High     | As an Auction Owner, I want to edit auctions, so that I can edit the product I'm currently bidding.  |
| US62                 | Manage auction status | High     | As an Auction Owner, I want to be able to manage my auction status so that I can change it when I see fit.   |
| US63                 | Cancel auction        | High     | As an Auction Owner, I want to be able to cancel my auctions so that my product no longer appears in the auction listing.  |
| US71                 | Manage auctions       | High     | As an Administrator, I want to manage auctions, so that I can alter multiple aspects of the auctions in the current auction list.  |
| US72                 | Cancel auctions       | High     | As an Administrator, I want to be able to cancel auctions, so that I can eliminate auctions that are infringing our policies.  |

## 1.2. Implemented Web Resources

Identify the web resources that were implemented in the prototype.

Module M01: Registration and 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 |

Module M02: Home page and Static pages

| Web Resource Reference | URL           |
|------------------------|---------------|
| R201: Home page        | GET /         |
| R202: About page       | GET /about    |
| R203: Services page    | GET /services |
| R204: FAQ page         | GET /faq      |
| R205: Contacts page    | GET /contacts |

#### Module M03: Platform administration

| Web Resource Reference | URL         |
|------------------------|-------------|
| R301: Admin Panel      | GET /manage |

#### Module M04: Users

| Web Resource Reference  | URL             |
|-------------------------|-----------------|
| R401: My profile UI     | GET /user/{id}  |
| R402: Edit profile      | POST /user/{id} |
| R403: List Users        | GET /users      |
| R404: View user profile | GET /users/{id} |

#### Module M05: Auctions

| Web Resource Reference      | URL                                       |
|-----------------------------|---|
| R501: View Auction          | GET /auctions/{auction_id}                |
| R502: Create Auction page   | GET /sell                                 |
| R503: Create Auction action | POST /sell                                |
| R504: Edit Auction page     | GET /auctions/edit/{id}                   |
| R505 Edit Auction Action    | POST /auctions/edit/{id}                  |
| R506: Cancel Auction        | POST /auctions/cancel/{id}                |
| R507: Auctions bids         | GET /api/auctions/getAllBids/{auction_id} |
| R508: Bid on auction        | POST /api/auctions                        |
| R509: Search auctions api   | GET /api/search                           |
| R510: Search auctions UI    | GET /search                               |
| R511: Delete auction image  | DELETE /users                             |

## 2. Prototype

URL of the prototype: <https://lbaw2271.lbaw.fe.up.pt/> URL of the source code:  
<https://git.fe.up.pt/lbaw/lbaw2223/lbaw2271>

### User credentials:

| Type of user  | Email               | Password |
|---------------|---------------------|----------|
| Administrator | testeadmin@fe.up.pt | 123456   |
| Regular user  | teste@fe.up.pt      | 123456   |

---

## Revision history

Changes made to the first submission:

1. Item 1
  2. ..
- 

GROUP2271, 24/11/2022

- André Sousa, up202005277@fe.up.pt (Editor)
- Pedro Moreira, up201905429@fe.up.pt
- Pedro Fonseca, up202008307@fe.up.pt
- Vítor Cavaleiro, up202004724@edu.fe.up.pt