

EAP: Architecture Specification and Prototype

Project Vision

STEAL! is an online marketplace focused on selling Content Distribution Keys (CDKs). It offers a secure, user-friendly platform for gamers seeking affordable game keys, aiming to become the top destination for accessible, high-quality digital gaming.

"SO AFFORDABLE IT'S LIKE STEALING!"

A7: Web Resources Specification

This artifact documents the architecture of the web application to be developed, detailing the catalog of resources, their properties, and the operations available for each resource. The specification follows the OpenAPI Specification (OAS). This page includes the following operations over data: create, read, update, and delete.

1. Overview

Module	Description
M01: Authentication and Profile	Web resources for user authentication and profile management, including handling user information and roles.
M02: Products Browse and Details	Web resources for searching, filtering, sorting, and listing available products, as well as viewing product details.
M03: Shopping Cart	Web resources for adding products to the shopping cart and managing the cart.
M04: Checkout	Web resources for proceeding to checkout, including navigation of purchase history.
M05: User Administration	Web resources for administrative functions, including managing users, products, and maintaining the shop's functionality.
M06: Static Pages	Web resources for static content and pages such as About, Contact, and FAQs.

2. Permissions

Role	Description	Permissions
PUB	Public	Group of users without privileges, namely "Anonymous users". Can browse public content.
USR	User	General authenticated users. Can update their profiles.
BYR	Buyer	Authenticated users who can purchase products.

Role	Description	Permissions
SLR	Seller	Authenticated users who can list products for sale but cannot purchase products.
ADM	Administrator	Group of administrators with full access to manage users, products, and site settings.

3. OpenAPI Specification

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

```

openapi: 3.0.0

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

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

externalDocs:
  description: Find more info here.
  url: https://gitlab.up.pt/lbaw/lbaw2425/lbaw2435

tags:
- name: 'M01: Authentication and Individual Profile'
- name: 'M02: Products Browse and Details'
- name: 'M03: Shopping Cart'
- name: 'M04: Checkout'
- name: 'M05: User Administration'
- name: 'M06: 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: 'Ok. Show log-in UI'

    post:
      operationId: R102
      summary: 'R102: Login Action'
      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:
            type: string
          password:
            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:
  post:
    operationId: R103
    summary: 'R103: Logout Action'
    description: 'Logout the current authenticated user. Access: USER'
    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'

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:
            role:
              type: string
              enum: [buyer, seller]
            username:
              type: string
            name:
              type: string
            email:
              type: string
            password:
              type: string
            confirm_password:
              type: string
            birthday:
              type: string
              format: date
          required:
            - role
            - username
            - name
            - email
            - password
            - confirm_password
            - birthday
  responses:
    '302':
      description: 'Redirect after processing the new user
information.'
      headers:
        Location:
```

```
    schema:
      type: string
    examples:
      302Success:
        description: 'Successful registration and authentication. Redirect to home page.'
        value: '/home'
      302Failure:
        description: 'Failed registration. Redirect to registration form.'
        value: '/register'

/profile:
  get:
    operationId: R106
    summary: 'R106: View user profile'
    description: 'Show the individual user profile. Access: USR'
    tags:
      - 'M01: Authentication and Individual Profile'
    parameters:
      - in: path
        name: id
        schema:
          type: integer
        required: true
    responses:
      '200':
        description: 'Ok. Show view profile UI'
        content:
          application/json:
            schema:
              type: object
              properties:
                id:
                  type: integer
                username:
                  type: string
                name:
                  type: string
                email:
                  type: string
                role:
                  type: string
                nif:
                  type: string
                  description: 'Only for buyers'
                birth_date:
                  type: string
                  format: date
                  description: 'Only for buyers'
                coins:
                  type: integer
                  description: 'Only for buyers'
```

```
put:
  operationId: R107
  summary: 'R107: Edit user profile'
  description: 'Edit the individual user profile. Access: USR'
  tags:
    - 'M01: Authentication and Individual Profile'
  parameters:
    - in: path
      name: id
      schema:
        type: integer
      required: true
  requestBody:
    required: true
    content:
      application/json:
        schema:
          type: object
          properties:
            username:
              type: string
            name:
              type: string
            nif:
              type: string
          required:
            - username
            - name
            - nif
  responses:
    '200':
      description: 'Profile updated successfully'
    '400':
      description: 'Invalid input'
    '404':
      description: 'Profile not found'

post:
  operationId: R108
  summary: 'R108: Change user password'
  description: 'Change the user password. Access: USR'
  tags:
    - 'M01: Authentication and Individual Profile'
  parameters:
    - in: path
      name: id
      schema:
        type: integer
      required: true
  requestBody:
    required: true
    content:
      application/json:
        schema:
```

```
      type: object
      properties:
        current_password:
          type: string
        new_password:
          type: string
        confirm_new_password:
          type: string
      required:
        - current_password
        - new_password
        - confirm_new_password
    responses:
      '200':
        description: 'Password changed successfully'
      '400':
        description: 'Invalid input'
      '404':
        description: 'Profile not found'
      '401':
        description: 'Unauthorized. Current password is incorrect'

/explorer:
  get:
    operationId: R201
    summary: 'R201: Explore and Search'
    description: 'Searches for items and returns the results as JSON.'
  Access: PUB.'
  tags:
    - 'M02: Products Browse and Details'
  parameters:
    - in: query
      name: query
      description: String to use for full-text search
      schema:
        type: string
      required: false
    - in: query
      name: sort
      description: Criteria to sort the search results
      schema:
        type: string
        enum: [all, new-releases, top-sellers, top-rated]
      required: false
    - in: query
      name: page
      description: Page number to show
      schema:
        type: integer
      required: false
  responses:
    '200':
      description: 'Ok. Show search results'
      content:
```

```
application/json:
  schema:
    type: array
    items:
      type: object
      properties:
        id:
          type: integer
        name:
          type: string
        categories:
          type: array
          items:
            type: string
        players:
          type: array
          items:
            type: string
        platforms:
          type: array
          items:
            type: string
        release_date:
          type: string
          format: date
        price:
          type: number
          format: float
        rating:
          type: integer
  example:
    - id: 1
      name: 'The Legend of Zelda: Breath of the Wild'
      description: 'An open-world adventure game.'
      categories: ['Adventure', 'Action']
      players: ['SinglePlayer', 'MultiPlayer']
      platforms: ['Steam', 'Epic Games']
      release_date: '2017-03-03'
      price: 59.99
      rating: 97
    - id: 2
      name: 'Among Us'
      categories: ['Party', 'Strategy']
      players: ['Multiplayer']
      platforms: ['Steam', 'Epic Games']
      release_date: '2018-06-15'
      price: 4.99
      rating: 85
```

/game/{id}:

get:

operationId: R202

summary: 'R202: Get Game Details'

description: 'Retrieves detailed information about a specific game.'


```
Access: PUB.'
tags:
  - 'M02: Products Browse and Details'
parameters:
  - in: path
    name: id
    description: ID of the game to retrieve
    schema:
      type: integer
    required: true
responses:
  '200':
    description: 'Ok. Show game details'
    content:
      application/json:
        schema:
          type: object
          properties:
            id:
              type: integer
            name:
              type: string
            description:
              type: string
            seller:
              type: string
            minimum_age:
              type: integer
            price:
              type: number
              format: float
            rating:
              type: integer
            platforms:
              type: array
              items:
                type: string
            categories:
              type: array
              items:
                type: string
            languages:
              type: array
              items:
                type: string
            players:
              type: array
              items:
                type: string
          example:
            id: 1
            name: 'The Legend of Zelda: Breath of the Wild'
            description: 'An open-world adventure game.'
            seller: 'Nintendo'
```

```

        minimum_age: 12
        price: 59.99
        rating: 97
        platforms: ['Steam', 'Epic Games']
        categories: ['Adventure', 'Action']
        languages: ['English', 'Japanese']
        players: ['SinglePlayer', 'MultiPlayer']
    '404':
        description: 'Game not found'

/cart:
  get:
    operationId: R301
    summary: 'R301: View Shopping Cart'
    description: 'Retrieves the items in the shopping cart. Access:
PUB.'
    tags:
      - 'M03: Shopping Cart'
    responses:
      '200':
        description: 'Ok. Show cart items'
        content:
          application/json:
            schema:
              type: array
              items:
                type: object
                properties:
                  product_id:
                    type: integer
                  name:
                    type: string
                  price:
                    type: number
                    format: float
                  quantity:
                    type: integer
            example:
              - product_id: 1
                name: 'Product 1'
                price: 19.99
                quantity: 2
              - product_id: 2
                name: 'Product 2'
                price: 9.99
                quantity: 1

/cart/add_product:
  post:
    operationId: R302
    summary: 'R302: Add Product to Shopping Cart'
    description: 'Allows users to add a product to the shopping cart.
Access: PUB.'
    tags:

```

```

    - 'M03: Shopping Cart'
  requestBody:
    required: true
    content:
      application/json:
        schema:
          type: object
          properties:
            product_id:
              type: integer
              description: 'ID of the product to add'
            required:
              - product_id
  responses:
    '200':
      description: 'Product added to shopping cart successfully'
      content:
        application/json:
          schema:
            type: object
            properties:
              message:
                type: string
              cart:
                type: object
                properties:
                  product_id:
                    type: integer
            example:
              message: 'Product added to shopping cart successfully'
              cart:
                product_id: 1
    '400':
      description: 'Invalid input'
    '404':
      description: 'Product not found'

/cart/increase_quantity:
  post:
    operationId: R303
    summary: 'R303: Increase Product Quantity in Shopping Cart'
    description: 'Allows users to increase the quantity of a product in
the shopping cart. Access: PUB.'
    tags:
      - 'M03: Shopping Cart'
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              product_id:
                type: integer

```

```

        description: 'ID of the product to update'
        quantity:
          type: integer
        description: 'Quantity to increase'
      required:
        - product_id
        - quantity
    responses:
      '200':
        description: 'Product quantity increased successfully'
        content:
          application/json:
            schema:
              type: object
              properties:
                message:
                  type: string
                cart:
                  type: object
                  properties:
                    product_id:
                      type: integer
                    quantity:
                      type: integer
            example:
              message: 'Product quantity increased successfully'
              cart:
                product_id: 1
                quantity: 3
      '400':
        description: 'Invalid input'
      '404':
        description: 'Product not found'

```

```

/cart/decrease_quantity:
  post:
    operationId: R304
    summary: 'R304: Decrease Product Quantity in Shopping Cart'
    description: 'Allows users to decrease the quantity of a product in
the shopping cart. Access: PUB.'
    tags:
      - 'M03: Shopping Cart'
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              product_id:
                type: integer
                description: 'ID of the product to update'
              quantity:
                type: integer

```

```
        description: 'Quantity to decrease'
        required:
          - product_id
          - quantity
    responses:
      '200':
        description: 'Product quantity decreased successfully'
        content:
          application/json:
            schema:
              type: object
              properties:
                message:
                  type: string
                cart:
                  type: object
                  properties:
                    product_id:
                      type: integer
                    quantity:
                      type: integer
            example:
              message: 'Product quantity decreased successfully'
              cart:
                product_id: 1
                quantity: 1
      '400':
        description: 'Invalid input'
      '404':
        description: 'Product not found'

/cart/remove:
  delete:
    operationId: R305
    summary: 'R305: Remove Product from Shopping Cart'
    description: 'Allows users to remove a product from the shopping
cart. Access: PUB.'
    tags:
      - 'M03: Shopping Cart'
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              product_id:
                type: integer
                description: 'ID of the product to remove'
            required:
              - product_id
    responses:
      '200':
        description: 'Product removed from shopping cart successfully'
```

```
    content:
      application/json:
        schema:
          type: object
          properties:
            message:
              type: string
          example:
            message: 'Product removed from shopping cart successfully'
    '400':
      description: 'Invalid input'
    '404':
      description: 'Product not found'
```

/checkout:

```
  post:
    operationId: R401
    summary: 'R401: Checkout'
    description: 'Allows buyers to proceed to checkout. Access: BYR.'
    tags:
      - 'M04: Checkout'
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              cart_id:
                type: integer
                description: 'ID of the shopping cart'
            required:
              - cart_id
    responses:
      '200':
        description: 'Proceed to payment'
        content:
          application/json:
            schema:
              type: object
              properties:
                message:
                  type: string
                payment_url:
                  type: string
              example:
                message: 'Proceed to payment'
                payment_url: '/checkout/payment'
      '400':
        description: 'Invalid input'
      '404':
        description: 'Cart not found'
```

/checkout/payment:

```
get:
  operationId: R402
  summary: 'R402: Get Payment Methods'
  description: 'Retrieves available payment methods. Access: BYR.'
  tags:
    - 'M04: Checkout'
  responses:
    '200':
      description: 'Ok. Show available payment methods'
      content:
        application/json:
          schema:
            type: array
            items:
              type: object
              properties:
                method_id:
                  type: integer
                method_name:
                  type: string

post:
  operationId: R403
  summary: 'R403: Choose Payment Method'
  description: 'Allows buyers to choose a payment method. Access:
BYR.'
  tags:
    - 'M04: Checkout'
  requestBody:
    required: true
    content:
      application/json:
        schema:
          type: object
          properties:
            cart_id:
              type: integer
              description: 'ID of the shopping cart'
            payment_method:
              type: string
              description: 'Chosen payment method'
          required:
            - cart_id
            - payment_method
  responses:
    '200':
      description: 'Payment method chosen successfully'
      content:
        application/json:
          schema:
            type: object
            properties:
              message:
                type: string
```

```
        receipt_url:
          type: string
      example:
        message: 'Payment method chosen successfully'
        receipt_url: '/checkout'
'400':
  description: 'Invalid input'
'404':
  description: 'Cart not found'

/checkout/receipt:
  get:
    operationId: R404
    summary: 'R404: Get Receipt'
    description: 'Retrieves the receipt after a successful order.'
  Access: BYR.
  tags:
    - 'M04: Checkout'
  parameters:
    - in: query
      name: order_id
      description: 'ID of the order'
      schema:
        type: integer
      required: true
  responses:
    '200':
      description: 'Ok. Show receipt'
      content:
        application/json:
          schema:
            type: object
            properties:
              order_id:
                type: integer
              purchases:
                type: array
                items:
                  type: object
                  properties:
                    gameId:
                      type: string
                    cdKCode:
                      type: string
                    value:
                      type: number
                      format: float
            example:
              order_id: 123
              purchases:
                - gameId: 'The Legend of Zelda: Breath of the Wild'
                  cdKCode: 'ABC123XYZ'
                  value: 59.99
                - gameId: 'The Legend of Zelda: Breath of the Wild 2'
```



```
        cdKCode: 'XYZ321ABC'
        Value: 79.99
    '400':
        description: 'Invalid input'
    '404':
        description: 'Order not found'

/user/{id}/order-history:
  get:
    operationId: R405
    summary: 'R405: Get Purchase History'
    description: 'Retrieves the purchase history for a specific user.'
  Access: BYR.
  tags:
    - 'M04: Checkout'
  parameters:
    - in: path
      name: id
      description: 'ID of the user'
      schema:
        type: integer
      required: true
    - in: query
      name: sortBy
      description: 'Field to sort by (e.g., time, totalPrice)'
      schema:
        type: string
        enum: [time, totalPrice]
      required: false
    - in: query
      name: direction
      description: 'Sort direction (asc or desc)'
      schema:
        type: string
        enum: [asc, desc]
      required: false
  responses:
    '200':
      description: 'Ok. Show purchase history'
      content:
        application/json:
          schema:
            type: object
            properties:
              orders:
                type: array
                items:
                  type: object
                  properties:
                    order_id:
                      type: integer
                    payment_method:
                      type: string
                    purchases:
```

```

      type: array
      items:
        type: object
        properties:
          game:
            type: string
          cdk:
            type: string
          value:
            type: number
            format: float
      total_price:
        type: number
        format: float
      formatted_time:
        type: string
example:
  orders:
    - order_id: 1
      payment_method: 'Credit Card'
      purchases:
        - game: 'The Legend of Zelda: Breath of the Wild'
          cdk: 'ABC123XYZ'
          value: 59.99
        - game: 'Among Us'
          cdk: 'DEF456UVW'
          value: 4.99
      total_price: 64.98
      formatted_time: '2023-01-01 12:00:00'
    - order_id: 2
      payment_method: 'PayPal'
      purchases:
        - game: 'Minecraft'
          cdk: 'GHI789JKL'
          value: 26.95
      total_price: 26.95
      formatted_time: '2023-02-01 15:30:00'
'400':
  description: 'Invalid input'
'404':
  description: 'User not found'

/admin/users/search:
  get:
    operationId: R501
    summary: 'R501: Search Users'
    description: 'Allows admin to search for users by username or email.
Access: ADM.'
    tags:
      - 'M05: User Administration'
    parameters:
      - in: query
        name: user_query
        description: 'Search query for username or email'

```

```
    schema:
      type: string
      required: false
  responses:
    '200':
      description: 'Ok. Show search results'
      content:
        application/json:
          schema:
            type: array
            items:
              type: object
              properties:
                id:
                  type: integer
                username:
                  type: string
                email:
                  type: string
          example:
            - id: 1
              username: 'john_doe'
              email: 'john@example.com'
            - id: 2
              username: 'jane_doe'
              email: 'jane@example.com'

/admin/all-users:
  get:
    operationId: R502
    summary: 'R502: List All Users'
    description: 'Allows admin to list all users, including buyers and sellers. Access: ADM.'
    tags:
      - 'M05: User Administration'
    responses:
      '200':
        description: 'Ok. Show all users'
        content:
          application/json:
            schema:
              type: object
              properties:
                buyers:
                  type: array
                  items:
                    type: object
                    properties:
                      id:
                        type: integer
                      username:
                        type: string
            example:
              - id: 1
```

```

        username: 'john_doe'
    sellers:
      type: array
      items:
        type: object
        properties:
          id:
            type: integer
          username:
            type: string
      example:
        - id: 2
          username: 'jane_doe'
  '400':
    description: 'Invalid input'
  '404':
    description: 'Users not found'

/admin/users/{id}:
  get:
    operationId: R503
    summary: 'R503: View User Profile'
    description: 'Allows admin to view detailed information about a
user. Access: ADM.'
    tags:
      - 'M05: User Administration'
    parameters:
      - in: path
        name: id
        description: 'ID of the user'
        schema:
          type: integer
        required: true
    responses:
      '200':
        description: 'Ok. Show user information'
        content:
          application/json:
            schema:
              type: object
              properties:
                id:
                  type: integer
                username:
                  type: string
                name:
                  type: string
                email:
                  type: string
                status:
                  type: string
                role:
                  type: string
                nif:

```

```

        type: string
    birth_date:
        type: string
        format: date
    coins:
        type: integer
example:
  id: 1
  username: 'john_doe'
  name: 'John Doe'
  email: 'john@example.com'
  status: 'active'
  role: 'buyer'
  nif: '123456789'
  birth_date: '1990-01-01'
  coins: 100

```

/admin/users/{id}/change-username:

post:

operationId: R504

summary: 'R504: Change Users Username'

description: 'Allows admin to change the username of a user. Access: ADM.'

tags:

- 'M05: User Administration'

parameters:

- in: path
 - name: id
 - description: 'ID of the user'
 - schema:
 - type: integer
 - required: true

requestBody:

required: true

content:

application/json:

schema:

- type: object
- properties:
 - username:
 - type: string
- required:
 - username

responses:

'200':

description: 'Username updated successfully'

content:

application/json:

- schema:
 - type: object
 - properties:
 - message:
 - type: string

example:

```
        message: 'Username updated successfully'
      '400':
        description: 'Invalid input'
      '404':
        description: 'User not found'

/admin/users/{id}/change-name:
  post:
    operationId: R505
    summary: 'R505: Change Users Name'
    description: 'Allows admin to change the name of a user. Access:
ADM.'
    tags:
      - 'M05: User Administration'
    parameters:
      - in: path
        name: id
        description: 'ID of the user'
        schema:
          type: integer
          required: true
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              name:
                type: string
                required:
                  - name
    responses:
      '200':
        description: 'Name updated successfully'
        content:
          application/json:
            schema:
              type: object
              properties:
                message:
                  type: string
            example:
              message: 'Name updated successfully'
      '400':
        description: 'Invalid input'
      '404':
        description: 'User not found'

/admin/users/{id}/change-coins:
  post:
    operationId: R506
    summary: 'R506: Change Coins'
    description: 'Allows admin to change the coins of a user. Access:
```

```
ADM.'  
  tags:  
    - 'M05: User Administration'  
  parameters:  
    - in: path  
      name: id  
      description: 'ID of the user'  
      schema:  
        type: integer  
        required: true  
  requestBody:  
    required: true  
    content:  
      application/json:  
        schema:  
          type: object  
          properties:  
            coins:  
              type: integer  
              required:  
                - coins  
  responses:  
    '200':  
      description: 'Coins updated successfully'  
      content:  
        application/json:  
          schema:  
            type: object  
            properties:  
              message:  
                type: string  
            example:  
              message: 'Coins updated successfully'  
    '400':  
      description: 'Invalid input'  
    '404':  
      description: 'User not found'  
  
/contact:  
  get:  
    operationId: R601  
    summary: 'R601: Get Contact Information'  
    description: 'Retrieves contact information. Access: PUB.'  
    tags:  
      - 'M06: Static Pages'  
    responses:  
      '200':  
        description: 'Ok. Show contact information'  
        content:  
          application/json:  
            schema:  
              type: object  
              properties:  
                contacts:
```

```
        type: array
        items:
          type: object
          properties:
            contact:
              type: string
      example:
        contacts:
          - contact: 'contact@example.com'
          - contact: 'support@example.com'
/faqs:
  get:
    operationId: R602
    summary: 'R602: Get FAQs'
    description: 'Retrieves frequently asked questions. Access: PUB.'
    tags:
      - 'M06: Static Pages'
    responses:
      '200':
        description: 'Ok. Show FAQs'
        content:
          application/json:
            schema:
              type: array
              items:
                type: object
                properties:
                  question:
                    type: string
                  answer:
                    type: string
            example:
              - question: 'What is this website about?'
                answer: 'This website is a project for the Database and
Web Applications Laboratory (LBAW) course at the University of Porto
(FEUP).'
              - question: 'How can I contact support?'
                answer: 'You can contact support at
support@example.com.'

/about:
  get:
    operationId: R603
    summary: 'R603: Get About Information'
    description: 'Retrieves information about the website. Access: PUB.'
    tags:
      - 'M06: Static Pages'
    responses:
      '200':
        description: 'Ok. Show about information'
        content:
          application/json:
            schema:
              type: array
```



```
items:
  type: object
  properties:
    content:
      type: string
example:
  - content: 'content 1'
  - content: 'content 2'
```

A8: Vertical prototype

This artifact contains the implementation of most of the high priority user stories of our web application and all mandatory features for its vertical prototype (features marked with an * in the common and theme requirements documents). Our implementation is based on the LBAW Framework and includes work both on the client and server side of the application, focusing both on user interface and server data access. By implementing all the features described in this section, our prototype has a functional implementation of all CRUD operations (Create, Read, Update, Delete).

1. Implemented Features

1.1. Implemented User Stories

Identification of the user stories that were implemented in the prototype.

User Story reference	Name	Priority	Responsible	Description
US01	Browse Marketplace	High	Ricardo Yang	As an Anonymous User, I want to browse the marketplace, view and filter the product list and categories, so that I can explore available CDKs.
US02	View Product Details	High	Ricardo Yang	As an Anonymous User, I want to view detailed information about a game, including reviews, so that I can decide if I want to purchase it.
US03	Register Account	High	Bruno Huang	As an Anonymous User, I want to register an account, so that I can access additional features.
US04	Sign In Account	High	Bruno Huang	As an Anonymous User, I want to sign in to my account, so that I can have access additional features.
US05	Add to Shopping Cart	High	Daniel Basílio	As an Anonymous User, I want to add games to a shopping cart, so that I can purchase multiple items at once.

User Story reference	Name	Priority	Responsible	Description
US06	Manage Shopping Cart	High	Daniel Basílio	As an Anonymous User, I want to manage my shopping cart, so that I can update quantities or remove items before purchasing.
US07	Full Text Search	High	Ricardo Yang	As an Anonymous User, I want to perform full-text searches across the game catalog, so that I can quickly find games based on keywords.
US08	Search Games by Genre/Platform	High	Ricardo Yang	As an Anonymous User, I want to search for games by genre, platform, price, language and rating, so that I can quickly find the games I am interested in.
US09	Access Static Pages	Medium	Ricardo Yang	As an Anonymous User, I want to view static pages (About Us, Contact Us, FAQs), so that I can understand the platform's purpose and policies.
US11	Edit Profile	High	Bruno Huang	As an Authenticated User, I want to be able to edit my profile, so that I can personalize my account.
US12	Log Out	High	Bruno Huang	As an Authenticated User, I want to log out of my account, so that my session is securely terminated and my account is protected.
US13	View Profile	High	Bruno Huang	As an Authenticated User, I want to view my profile information, so that I can see my account details and ensure they are correct.
US14	Update Profile Information	High	Bruno Huang	As an Authenticated User, I want to update my profile information, so that my account details are up to date.
US15	Change Password	High	Bruno Huang	As an Authenticated User, I want to change my password, so that I can maintain the security of my account.
US18	Checkout Items	High	Francisco Magalhães	As a Buyer, I want to complete the checkout process for items in my cart or individual items, so that I can finalize my purchases efficiently.

User Story reference	Name	Priority	Responsible	Description
US22	Track Purchase History	Medium	Francisco Magalhães	As a Buyer, I want to track my purchase history, so that I can review my past orders.
US26	Cancel Order	Medium	Francisco Magalhães	As a Buyer, I want to cancel my order, so that I can manage my purchases effectively if I change my mind.
US29	Multiple Payment Options	Low	Francisco Magalhães	As a Buyer, I want to complete my purchase using multiple payment methods as by PayPal, MBWay or credit card, so that I can choose the most convenient method for me.
US39	Administer User Accounts (Search, View, Edit, Create)	High	Bruno Huang	As an Administrator, I want to manage user accounts by searching for, viewing, editing, and creating user profiles, so that I can ensure that user information is accurate and up to date on the platform.

1.2. Implemented Web Resources

Module M01: Authentication and User Profile

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
R106: View Profile	GET /profile
R107: Edit Profile	PUT /profile/edit
R108: Change Password	POST /profile

Module M02: Products Browse and Details

Web Resource Reference	URL
R201: Game Explore and Search	GET /explore
R202: View Game Details	GET /game/{id}

Module M03: Shopping Cart

Web Resource Reference	URL
R301: View Shopping Cart	GET /cart
R302: Add Product To Shopping Cart	POST /cart/add_product
R303: Increase Quantity in Shopping Cart	POST /cart/increase_quantity
R304: Decrease Quantity in Shopping Cart	POST /cart/decrease_quantity
R305: Remove Product From Shopping Cart	POST /cart/remove_product

Module M04: Checkout

Web Resource Reference	URL
R401: Proceed to Checkout	POST /checkout
R402: Get Payment Methods	GET /checkout/payment
R403: Choose Payment Method	POST /checkout/payment
R404: Get Receipt	GET /checkout/receipt
R405: View Purchase History	GET /user/{id}/order-history

Module M05: User Administration

Web Resource Reference	URL
R501: Search Users	GET /admin/users/search
R503: List All Users	GET /admin/all-users
R502: View User Profile	GET /admin/users/{id}
R504: Change User's Username	POST /admin/users/{id}/change-username
R505: Change User's Name	POST /admin/users/{id}/change-name
R506: Change User's Coins	POST /admin/users/{id}/change-coins

Module M06: Static Pages

Web Resource Reference	URL
R601: Contact	GET /contact
R602: FAQs	GET /faqs
R603: About	GET /about

2. Prototype

The prototype Docker image is available at GitLab's Registry Container and can be run with:

```
docker run -d --name lbaw2435 -p 8001:80
gitlab.up.pt:5050/lbaw/lbaw2425/lbaw2435
```

For **ARM** users (e.g., Apple Silicon Macs), Docker Desktop supports architecture emulation using **qemu**. You can run the image with:

```
docker run --platform linux/amd64 -d --name lbaw2435 -p 8001:80
gitlab.up.pt:5050/lbaw/lbaw2425/lbaw2435
```

The application will be available at <http://localhost:8001>

Credentials:

- admin user: admin@example.com
- buyer user: buyer@example.com
- seller user: seller@example.com
- password (for all of them): 1234

The submitted version of the code is available at <https://gitlab.up.pt/lbaw/lbaw2425/lbaw2435>

Revision history

Changes made to the first submission:

no changes were made

GROUP2435, 24/11/2024

- Bruno Huang, up202207517@up.pt
- Daniel Basilio, up201806838@up.pt
- Francisco Magalhães, up202007945@up.pt
- Ricardo Yang, up202208465@up.pt (Editor)