

PA - Product and Presentation

Last edited by [racoelhosilva](#) just now

PA - Product and Presentation

ProGram aims to bring together programmers, developers and other people passionate about informatics, while providing a supportive environment for communication and discussion. Our vision is to create a safe space where like-minded individuals can share ideas, learn new tricks and even start projects together.

Table of Contents

- [PA - Product and Presentation](#)
 - [Table of Contents](#)
 - [A9: Product](#)
 - [1. Installation](#)
 - [2. Usage](#)
 - [2.1. Administration Credentials](#)
 - [2.2. User Credentials](#)
 - [2.3 Password Recovery \(Mailtrap\).](#)
 - [2.4 Real-Time Notifications \(Pusher\).](#)
 - [2.5 External API \(Postman\).](#)
 - [3. Application Help](#)
 - [Error Pages](#)
 - [Toast Messages](#)
 - [Client and Server Input Validation](#)
 - [Tooltips](#)
 - [Notifications](#)
 - [Frequently Asked Questions \(FAQ\) Page](#)
 - [4. Input Validation](#)
 - [5. Check Accessibility and Usability](#)
 - [6. HTML & CSS Validation](#)
 - [7. Revisions to the Project](#)
 - [8. Implementation Details](#)
 - [8.1. Libraries Used](#)
 - [8.2 User Stories](#)
 - [8.3 Business Rules](#)
 - [A10: Presentation](#)
 - [1. Product presentation](#)

A9: Product

ProGram is an online social media application designed for programmers, developers and informatics enthusiasts. Through its intuitive interface, it allows users to create posts sharing their ideas related to informatics. These posts are written with rich text and can contain images and associated tags with them.

These posts can then be seen by other users of the application either on their homepage feed, the search page or on the author's profile page.

Authenticated users can even interact with posts by leaving comments or reacting to the post or comments of other users.

Each user can also have a customized profile page featuring avatar and banner, description and even listings about their top languages, technologies and projects. In order to have a more personalized experience, users can follow other users, making it so the recommendations are always different and relevant content can be shown first.

Another great way to meet new people is to join different groups, allowing users with shared interests to connect and interact with each other. All of these different interactions also have an associated notification system, making it easier for everyone to receive updates.

ProGram also has a great focus on privacy so users can at any time, make their profiles, posts or groups private, meaning that only certain people have access to them.

To easily supervise and control the entire website, admins have a special dashboard interface to control everything that happens and take action when needed.

One final feature is that, since this application is targeted for programmers, users can also generate a specific token and interact with the product through our external API, not needing to open the website to view and share content!

1. Installation

In order to run the project, it is necessary to login into gitlab.up.pt using docker:

```
docker login gitlab.up.pt:5050
```

After that, the project is available by running the following command to download and run the container:

```
docker run -d --name lbaw2411 -p 8001:80 gitlab.up.pt:5050/lbaw/lbaw2425/lbaw2411
```

This should make the website available at localhost:8001 .

The Release Tag for the Final Product can be checked here: [Final Product](#).

2. Usage

2.1. Administration Credentials

Administration URL: <http://localhost:8001/admin>

E-mail	Password
kherrera@program.com	AdminsRule

2.2. User Credentials

Type	Username	Password
User	cindyburton@gmail.com	ILoveProgram

2.3 Password Recovery (Mailtrap)

Mailtrap URL: <https://mailtrap.io/inboxes/3327402/messages>

Login with the Google account:

E-mail	Password
programlbaw@gmail.com	programlbaw2411

2.4 Real-Time Notifications (Pusher)

Pusher URL: <https://dashboard.pusher.com/apps/1909573>

Login with the Google account:

E-mail	Password
programlbaw@gmail.com	programlbaw2411

2.5 External API (Postman)

We developed an external API which users can access with a generated token.
The API reference is located here: <http://127.0.0.1:8001/apireference>. Tokens can be generated from the user settings or by filling this link: <http://127.0.0.1:8001/user/{user-id}/token>.

To use the API, we provide the following Postman collection: [ProGram API](#)

Alternatively, you can use the following Curl command structure to access the API:

```
curl -X GET "http://localhost:8001/api/<ROUTE>" \
-H "Authorization: Bearer <ACCESS_TOKEN>"
```

3. Application Help

Error Pages

Some resources in our application have permissions associated with them or they might have been deleted or the user trying to access the resource might even be banned. In these cases, custom error pages are presented, showing custom relevant messages.

Fig. 1 Error page examples

Toast Messages

When performing actions in our application, users may receive toast messages of error, information or success with indicative messages of the result of the action.

Fig. 2 Toast Message examples

Client and Server Input Validation

Throughout the application, the user can perform multiple actions submitting data to the server. In these cases, we use forms, buttons and checks to validate the data before sending it to the server. This validation is done not only on the client-side, but also on the server-side with conditions and regular expressions, ensuring that the data is correct and secure.

Fig. 3 Client and Server Input Validation examples

Note: The examples above are for toast messages for when a user tries to like a post without being logged in (client-side input validation) and for when a user tries to create an account with a password that is too small (server-side input validation). We also have client-side information for when field are left empty or don't match the expected format.

Tooltips

In some fields of the forms mentioned above, we have added tooltips with descriptions to help the user understand what is expected in that field. These tooltips are shown has a little information icon that the user can hover over to see the tooltip.

Fig. 4 Tooltips examples

Notifications

In our application, users can interact with each other through follows, likes on posts, comments on posts and even likes on comments. To keep users updated on these interactions, we have implemented a notification system that shows the user the most recent interactions they have received and also let's users receive them in real-time through Pusher.

Fig. 5 Notification examples

Frequently Asked Questions (FAQ) Page

In order to help users with common questions, we have created a Frequently Asked Questions page where users can find answers to common questions they might have about the application.

Fig. 6 FAQ Page

4. Input Validation

In our social media application, data input is one of the main sources of interaction between users and the platform. To ensure that the data is correct and secure, we have implemented input validation on both the client-side and server-side:

- **Client-side:**
 - Using HTML5 form validation attributes (e.g. `required` for mandatory fields, `type="email"` for email fields)
 - Using JavaScript to check conditions before sending requests to the server (e.g. checking if a user is logged in before liking a post)
- **Server-side:**
 - Using regular expressions on routes to ensure they are correctly defined and pages that don't exist return a 404 error
 - Using middlewares to check if users can perform certain actions (e.g. only admins can access the admin dashboard, banned users can't access the website, users have to be logged in to access certain resources)
 - Using policies and gates to check if users can perform certain actions (e.g. random users cannot edit or delete posts from other users)
 - Using type, length and format checks on the server-side to ensure that the data is correct and secure (e.g. checking if the password is at least 8 characters long, checking if when creating a post associated tags exist in the database)

5. Check Accessibility and Usability

Accessibility Report: 16/18
[HTML Results](#)

Usability Report: 26/28
[HTML Results](#)

Exporting results in PDF lead to very bad interfaces, therefore we provide the HTML results for both reports.

6. HTML & CSS Validation

HTML Validation Report

- Home Page: [PDF Results](#)
- Post Page: [PDF Results](#)

- Profile Page: [PDF Results](#)
- Create Post Page: [PDF Results](#)
- Search Page: [PDF Results](#)

Note: the errors shown are due to using Firefox, which adds an extra charset specification to the HTML and the Blade @csrf directive which is a hidden input with autocomplete. These do not affect the usability of the website.

CSS Validation Report: [PDF Results](#)

Note: since we used Tailwind CSS, the CSS validation tool will show some warnings, but these are expected and do not affect the usability of the website. We also only provide one report since the CSS is the same for all pages.

7. Revisions to the Project

- Added 3 attributes on the user table to support GitHub, GitLab and Google OAuth
- Revisions on some triggers of the database
- Updated database schema to use `ON DELETE CASCADE` on foreign keys

8. Implementation Details

8.1. Libraries Used

- **Tailwind:** Utility-first CSS framework
 - [Tailwind](#)
 - Used to style the entire website

Fig. 7 Tailwind Example

- **Pusher:** Real-time notifications
 - [Pusher](#)
 - Used to send real-time notifications to users

Fig. 8 Pusher Example

- **Mailtrap:** Email testing
 - [Mailtrap](#)
 - Used to test email sending and password recovery

Fig. 9 Mailtrap Example

- **Quill:** Rich text editor
 - [Quill](#)
 - Used to create rich text posts

Fig. 10 Quill Example

8.2 User Stories

Module	Specification
M01: Home and Static Pages	Web resources associated with the home page and pages with static content. Includes the following system features: visualization of public and personalized timelines, about, FAQ and contact us pages.
M02: Authentication and Individual Profile	Web resources associated with user authentication, user profile and user settings management. Includes the following system features: user login, user registration (including OAuth), user logout, password reset, profile visualization, edition and account deletion.
M03: Posts and Announcements	Web resources associated with post and announcements, including comments and likes. Includes the following system features: post, comment and announcement creation, edition and deletion, liking and unliking of content (posts and comments).
M04: Search	Web resources associated with the search functionalities. Includes the following system features: full-text searches for users and content, exact match search and search filters.
M05: Administration	Web resources associated with administrator pages and actions. Includes the following system features: separate administrator login, user and content (posts, tags, technologies and language) management.
M06: Groups	Web resources associated with groups. Includes the following system features: group creation and edition, post group creation and deletion, group members, invites and requests.

Module	Specification
M07: Notifications and Interactions	Web resources associated with notification events and other user interactions (follows and requests). Includes the following system features: notification page, pusher notifications on events, follow actions and follow management.
M08: External API	Web resources associated with the provided external API, which uses credential tokens and allows authenticated and unauthenticated users to interact with the application.
M09: Accessibility	User stories associated with improving accessibility and maintaining design consistency throughout the web application.

Fig. 11 Application Modules

US Identifier	Name	Module	Priority	Team Members	State
US104	Full-text Search	M04	High	Bruno	100%
US102	Search for Public Users	M04	High	Bruno	100%
US105	Exact Match Search	M04	High	Bruno	100%
US202	Login	M02	High	Henrique	100%
US201	Registration	M02	High	Henrique	100%
US301	Logout	M02	High	Henrique	100%
US302	Create Post	M03	High	Rodrigo	100%
US303	Edit Post	M03	High	Rodrigo	100%
US304	Delete Post	M03	High	Rodrigo	100%
US103	View Public Timeline	M01	High	Rodrigo	100%
US305	View Personalized Timeline	M01	High	Rodrigo	100%
US321	Like Posts	M03	Medium	Henrique	100%
US322	Like Comment	M03	Medium	Henrique	100%
US101	View Public Profiles	M02	High	José	100%
US306	View My Profile	M02	High	José	100%
US307	View User Profiles	M02	High	José	100%
US603	Administrator Login	M05	High	Henrique	100%
US601	Manage Users	M05	High	Henrique	100%
US602	Manage Content	M05	High	Henrique	100%
US604	Ban Users	M05	Medium	Henrique	100%
US605	Unban Users	M05	Medium	Henrique	100%
US308	Edit Profile	M02	High	José	100%
US309	Add Profile Picture	M02	Medium	José	100%
US333	Manage Post Visibility	M03	Low	Bruno	100%
US204	OAuth API Register	M02	Medium	Henrique	100%
US205	OAuth API Login	M02	Medium	Henrique	100%
US111	FAQ	M01	Medium	José	100%
US112	About Us	M01	Medium	José, Rodrigo	100%
US113	Contact Us	M01	Medium	José	100%

US Identifier	Name	Module	Priority	Team Members	State
US203	Recover Password	M02	Medium	Henrique	100%
US339	Change Profile Visibility	M02	Low	Henrique	100%
US606	Delete User Accounts	M05	Medium	Henrique	100%
US310	Delete Account	M02	Medium	Henrique	100%
US318	Comment on Posts	M03	Medium	José, Bruno, Rodrigo	100%
US319	Edit Comment	M03	Medium	José	100%
US320	Delete Comment	M03	Medium	José	100%
US314	Send Follow Request	M07	Medium	Rodrigo	100%
US317	Manage Received Follow Requests	M07	Medium	Rodrigo	100%
US313	Follow other Users	M07	Medium	Rodrigo	100%
US315	Manage Followers	M07	Medium	Rodrigo	100%
US316	Manage Following	M07	Medium	Rodrigo	100%
US311	View Notifications	M07	Medium	Rodrigo	100%
US334	Notification on Post Like	M07	Low	Rodrigo	100%
US335	Notification on Post Comment	M07	Low	Rodrigo	100%
US336	Notification on Follow	M07	Low	Rodrigo	100%
US338	Notification on Comment Like	M07	Low	Rodrigo	100%
US325	Linking to external account	M02	Medium	Bruno	100%
US107	Search over Multiple Attributes	M04	Medium	Bruno	100%
US114	Order Results	M04	Low	Bruno	100%
US106	Search Filters	M04	Medium	Bruno	100%
US312	Search for Content	M04	Medium	Bruno	100%
US501	Add User to Group	M06	Medium	José	100%
US332	Request to Join Public Group	M06	Low	José	100%
US502	Edit Group Information	M06	Medium	José	100%
US401	View Group's Members	M06	Medium	Rodrigo, José	100%
US323	Create Group	M06	Medium	José	100%
US505	Manage Join Requests	M06	Low	José	100%
US402	Leave Group	M06	Medium	José	100%
US504	Send Group Invitation	M06	Medium	Rodrigo, José	100%
US503	Remove Group Member	M06	Medium	José	100%
US324	Manage Group Invitations	M06	Medium	Rodrigo, José	100%
US403	Post on Group	M06	Medium	Rodrigo, José	100%
US506	Change Group Visibility	M06	Low	Henrique	100%
US507	Remove Post From Group	M06	Low	Henrique	100%

US Identifier	Name	Module	Priority	Team Members	State
US328	Create Announcements	M03	Medium	José	100%
US327	Rich Text Post Formatting	M03	Medium	Bruno, Henrique	100%
US110	Contextual Help	M09	Medium	Bruno	100%
US326	Developer Stats	M02	Medium	Bruno	100%
US108	Placeholders in Form Inputs	M09	Medium	Bruno	100%
US109	Contextual Error Messages	M09	Medium	Henrique	100%
US701	Public Access to the API	M08	Medium	Bruno	100%
US702	API Token Access	M08	Medium	Bruno	100%
US703	Authenticated Access to the API	M08	Medium	Bruno	100%
US329	Appeal for Unban	M02	Low	Henrique	0%
US330	Tag Users	M03	Low	Bruno	0%
US331	View Followers' Feed	M01	Low	Rodrigo	0%
US337	Notification on Tag	M07	Low	Rodrigo	0%
US508	Delete Group	M06	Low	Henrique	0%

Fig. 12 All user stories with completion state

8.3 Business Rules

The following business rules are also fully implemented in the application:

Identifier	Name	Description
BR01	Spam Control	Users are limited to creating posts at a rate of 1 post per 30 seconds
BR02	Self-Interaction Restrictions	Users can comment but cannot like their own posts or comments
BR03	User Deletion Content Handling	When a user is deleted, their content is anonymized, including posts, comments, their names in group and profile
BR06	Public/Private Profiles	There should be public and privates profiles, private profiles are only available to followers

Fig. 13 Business Rules

A10: Presentation

This artifact contains a brief text and video presentation associated with the final product.

1. Product presentation

ProGram is a social media platform designed for programmers and tech enthusiasts to share ideas, collaborate, and connect. Users can create rich-text posts with images and tags, engage through comments and reactions, and personalize their profiles with avatars, skills, and projects. The platform offers tailored content recommendations, interest-based groups for community building, and robust privacy settings for posts, profiles, and groups. Developers can also use an external API for seamless interaction with the platform, ensuring a dynamic and secure experience for tech-savvy users.

GROUP2411, December 21, 2024

- Henrique Sardo Fernandes, up202204988@up.pt (Editor)
- Bruno Ricardo Soares Pereira de Sousa Oliveira, up202208700@up.pt
- José Carlos Malheiro de Sousa, up202208817@up.pt
- Rodrigo Albergaria Coelho e Silva, up202205188@up.pt