

## Stage: F4

Group symbol: **K07-20a**

Team: **1**

Project title: **InfluBridge**

**Team members** (*filled by PM, Team Leader*):

No.	Name	Surname	Student ID	Role
1	Dawid	Galik	205780	<i>PM, Team Leader</i>
2	Dominika	Rzepka	271301	<i>Team member</i>
3	Jakub	Gałązka	250060	<i>Team member</i>
4	Mateusz	Guściora	228884	<i>Team member</i>

## 4 Construction and tests (F4)

### 4.1 Implementation

The following description provides an overview of the implementation of the Influbridge web platform application, which facilitates influencer brand collaborations. The application is developed using Angular, Node.js and Spring Boot frameworks. This documentation focuses on the programming language source codes and associated artefacts. It provides a comprehensive list of created artefacts, configuration files, program resources, pre-requirements, and resulting programs. Additionally documentation consists technical requirements for application use. Technical requirements consists among others: Java 17, Spring Boot 6.0.1, IntelliJ IDE, Angular, Node.js, postman. Below there are presented screenshots of generated Documentation of the back-end application in the form of JavaDoc, collection of endpoints and remote project repositories for frontend and backend application.

The screenshot displays the JavaDoc interface for the `CampaignService` class. The top navigation bar includes links for OVERVIEW, PACKAGE, CLASS (highlighted), TREE, INDEX, and HELP. Below this, a search bar is present. The main content area shows the package `pl.edu.pwr.ist.ssd.influbridge.campaign` and the class `CampaignService`, which implements `java.lang.Object` and `@Service`. A brief description states: "The CampaignService class provides operations for managing campaigns. It handles the creation, joining, leaving, retrieval, and listing of campaigns." Below this, the "Constructor Summary" section lists a single constructor: `CampaignService()`. The "Method Summary" section is divided into "All Methods", "Instance Methods", and "Concrete Methods". It lists several methods with their modifiers, types, and descriptions:

Modifier and Type	Method	Description
Campaign	<code>create(CampaignRegisterDTO dto)</code>	Creates a new campaign.
<code>List&lt;Campaign&gt;</code>	<code>findAll()</code>	Retrieves all campaigns.
<code>List&lt;Campaign&gt;</code>	<code>getBrandCampaigns(Long id)</code>	Retrieves all campaigns associated with a brand.
Campaign	<code>getId(Long id)</code>	Retrieves a campaign by its ID.
<code>List&lt;Campaign&gt;</code>	<code>getInfluencerCampaigns(Long id)</code>	Retrieves all campaigns associated with an influencer.
boolean	<code>join(Long campaignId, Long influencerId)</code>	Joins an influencer to a campaign.
boolean	<code>leave(Long campaignId, Long influencerId)</code>	Leaves a campaign for an influencer.

At the bottom, it lists "Methods inherited from class java.lang.Object": `equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`.

Figure 1: Documentation of the backend application in the form of JavaDoc.

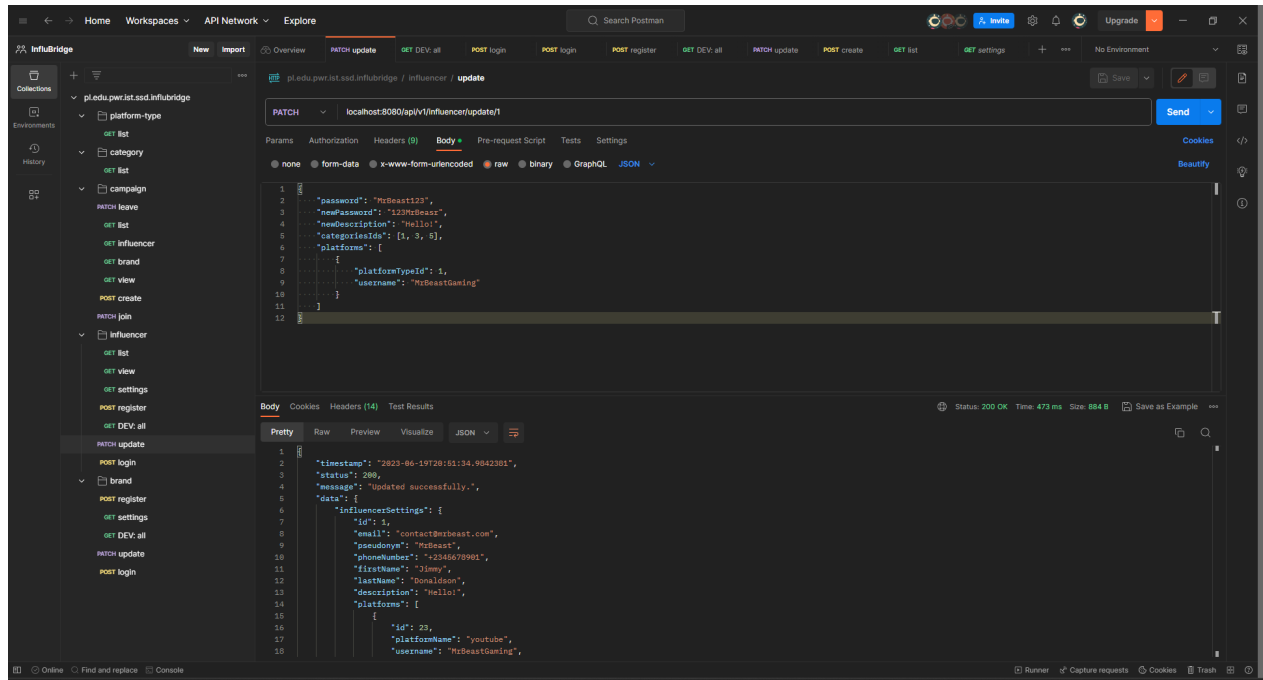


Figure 2: A collection of all endpoints with examples.

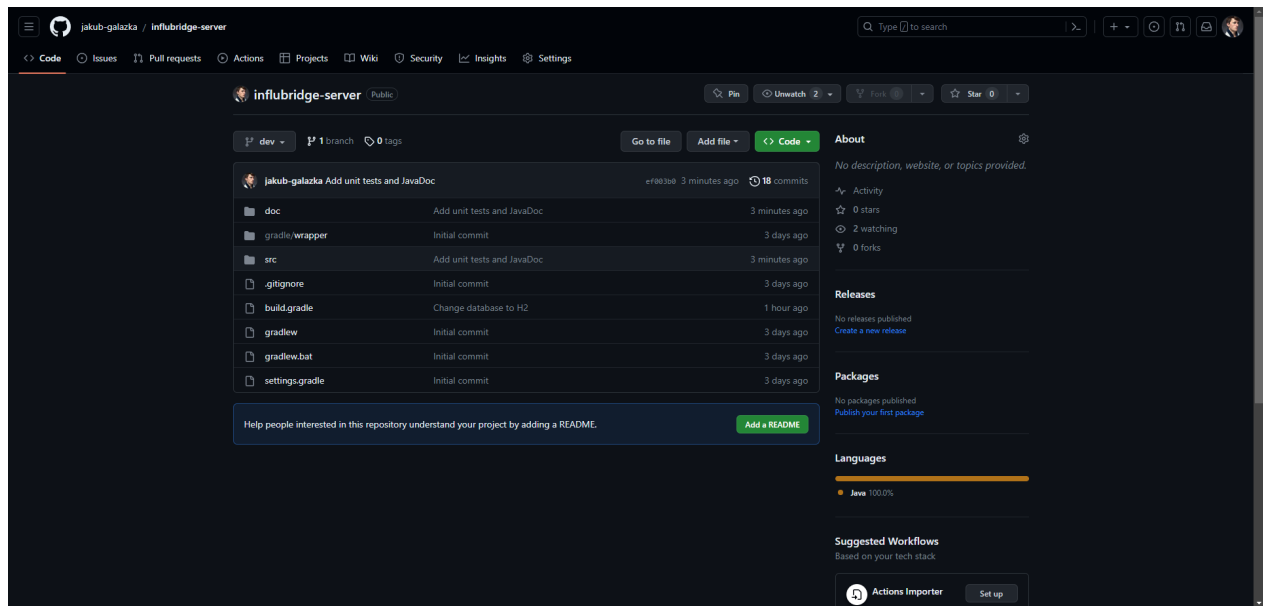


Figure 3: Remote project repository (backend application).

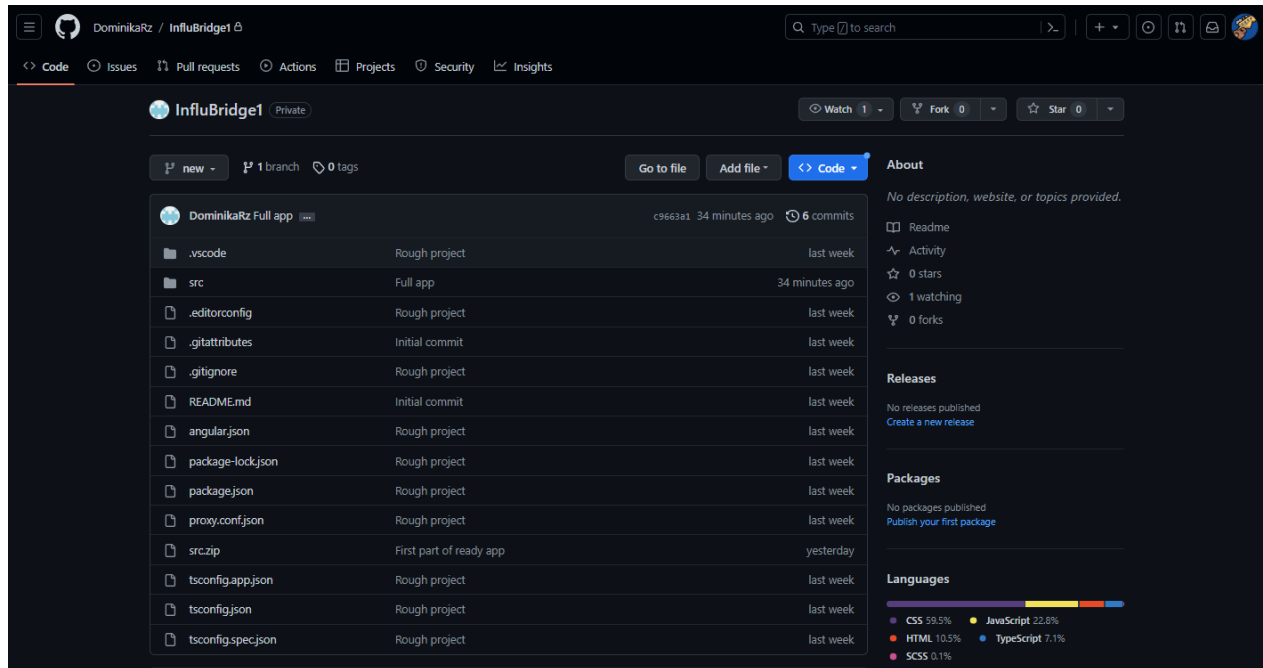


Figure 4: Remote project repository (frontend application).

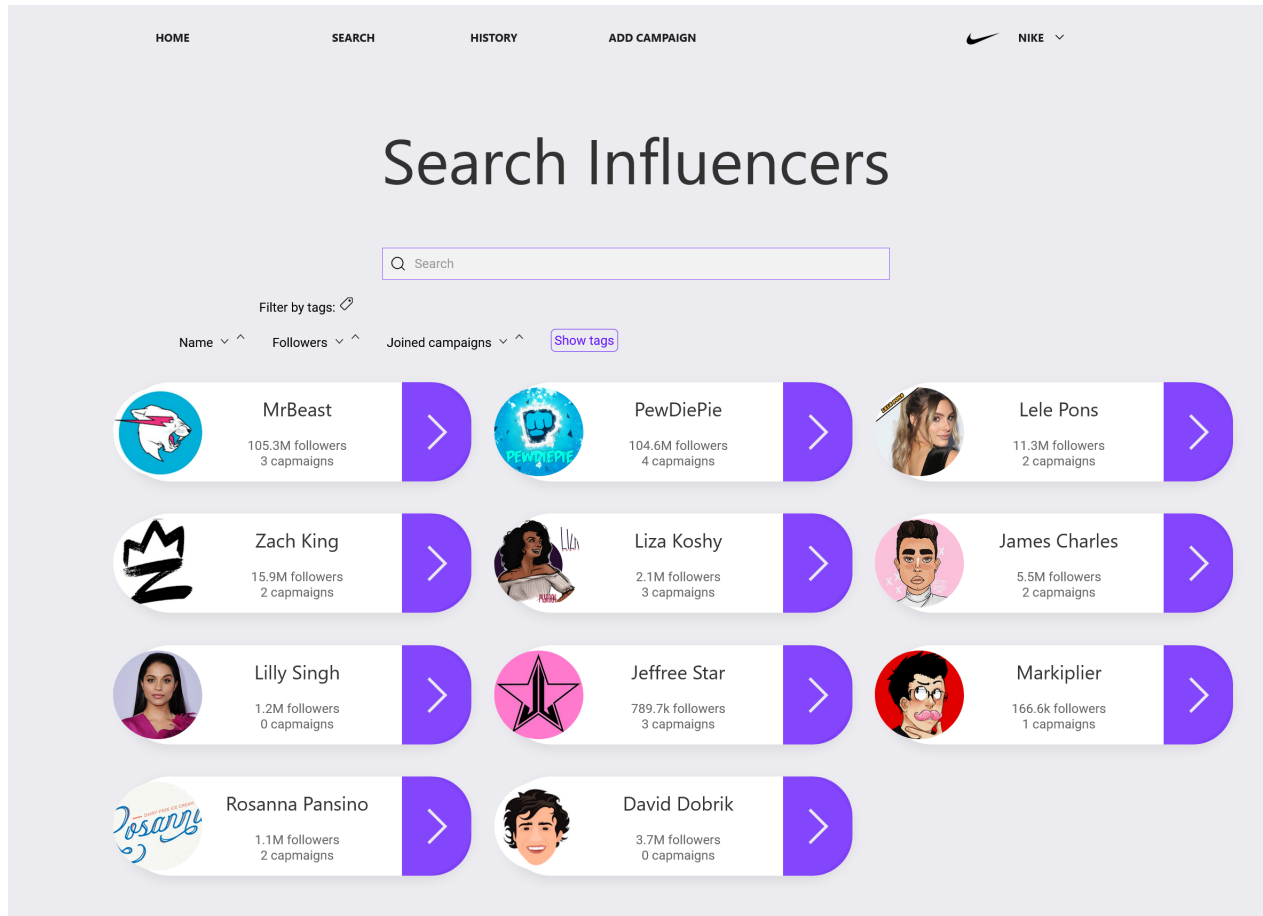


Figure 5: Search influencers as brand.

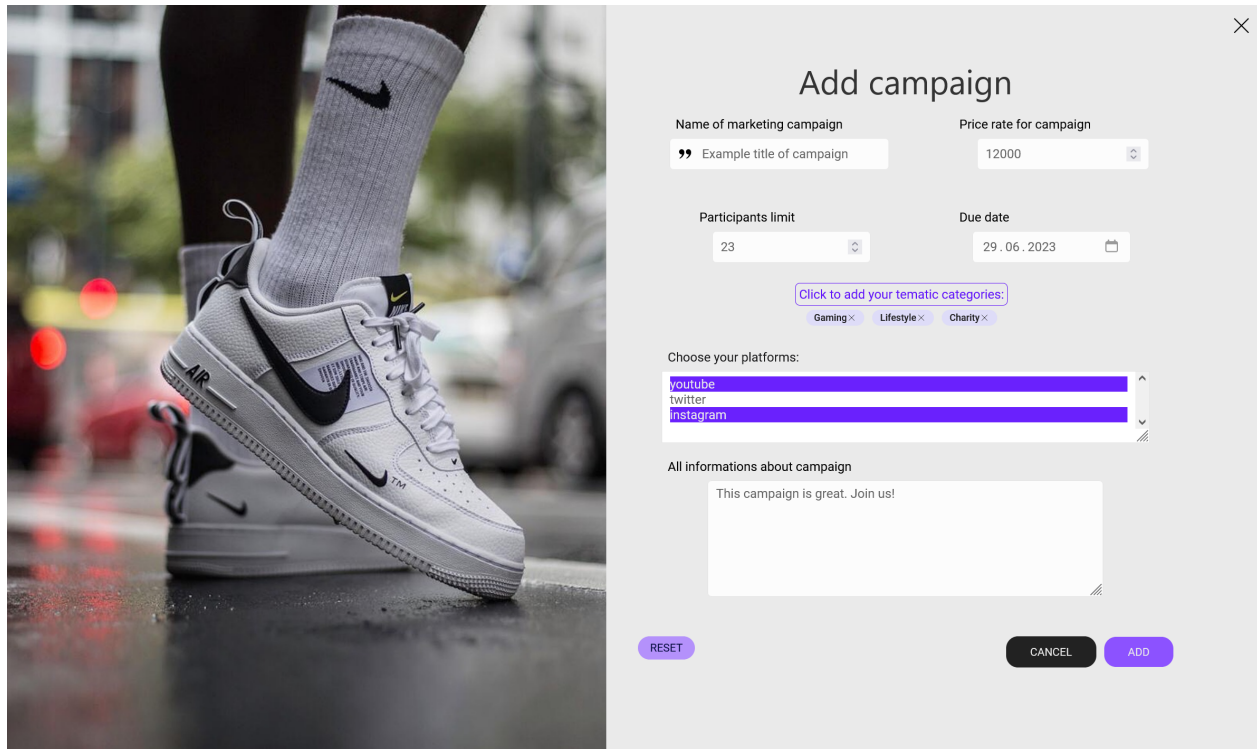


Figure 6: Add campaign.

## 4.2 Tests

This documentation provides an overview of the testing methods and test results for the specific elements of the InfluBridge web platform application. Acceptance tests cover both functional and non-functional requirements. In 4.2.1 chapter we presented description and results of Acceptance Tests. In 4.2.2 chapter remaining tests are described, where are three additional Acceptance Tests and an example of Unit Test. In case of Unit Test, in total four Unit Tests were conducted.

### 4.2.1 Requirements Tests

#### Functional Requirements

1. ACCEPTANCE TEST - FR-2 Business Logic - Allow influencers to seek partnership with brands - **TEST PASSED**

- **Test Case:** Test if influencers can seek partnership with brands.
- **Preconditions:**
  - (a) The application is installed and accessible.
  - (b) The user is registered as an "Influencer" and is logged into the application.
- **Steps:**
  - (a) Log into the system as an Influencer.

- (b) Enter search view to filter campaigns.
- (c) Browse through the list of brand campaigns that are available.
- (d) Select a brand and campaign that the influencer is interested in partnering with.
- (e) Fill out any necessary information or forms related to the partnership request, if required. This could include details such as proposed terms, reasons for partnership interest, etc.
- (f) On the selected campaign, locate and click on the "Join Campaign" button.
- (g) Submit the partnership request.
- (h) Check the application's response for confirmation of successful submission.
- (i) Go to the 'Home' of influencer that shows current Campaigns.

- **Acceptance criteria:**

- (a) The Influencer should be able to successfully browse and select brands and campaigns.
- (b) The "join campaign" function should be accessible and easy to use.
- (c) Upon submission, the Influencer should receive a confirmation message saying the partnership request has been sent successfully.
- (d) The partnership request should be visible in the 'My Campaigns' of influencer section.

- **Post-conditions:** The partnership request is successfully recorded and sent to the brand for consideration.

2. ACCEPTANCE TEST - FR-4 Business Logic - Allow influencers to present themselves

- **TEST PASSED**

- **Test Case:** Test if influencers can present themselves.

- **Preconditions:**

- (a) The application is installed and accessible.
- (b) The user is registered as an "Influencer" and is logged into the application.

- **Steps:**

- (a) Log into the system as an Influencer.
- (b) Navigate to the Brands page.
- (c) Browse through the list of brands available.
- (d) Select a brand that the influencer is interested in partnering with.
- (e) On the selected brand's profile, locate and click on the "Seek Partnership" button or similar functionality.
- (f) Fill out any necessary information or forms related to the partnership request, if required. This could include details such as proposed terms, reasons for partnership interest, etc.
- (g) Submit the partnership request.

- (h) Check the application's response for confirmation of successful submission.
- (i) Go to the 'Partnership Requests' section or similar functionality to confirm that the partnership request has been sent to the brand.

- **Acceptance criteria:**

- (a) The Influencer should be able to successfully browse and select brands.
- (b) The "Seek Partnership" function should be accessible and easy to use.
- (c) Upon submission, the Influencer should receive a confirmation message saying the partnership request has been sent successfully.
- (d) The partnership request should be visible in for the Influencer and Brand user.

- **Post-conditions:** The partnership request is successfully recorded and sent to the brand for consideration.

3. ACCEPTANCE TEST - FR-5 Business Logic - Allow brands to search for influencers

- **TEST PASSED**

- **Test Case:** Test if the application allows brands to search for influencers.

- **Preconditions:**

- (a) The application is installed and accessible.
- (b) The user is registered as a "Brand" and is logged into the application.

- **Steps:**

- (a) Log into the system as a Brand.
- (b) Navigate to the Influencer Search page.
- (c) Check if there is a search bar, filtering, sorting and filter by tags functionality.
- (d) Input relevant search criteria (like influencer's name, category, or followers count) into the search bar.
- (e) Hit the "Search" button.
- (f) Check the results shown - they should match the search criteria.
- (g) Repeat the process with different search criteria.

- **Acceptance criteria:**

- (a) The application provides a clear and intuitive interface for brands to search and filter influencers.
- (b) All necessary fields or options for searching are available and functional.
- (c) Upon submitting the search criteria, the application returns results that match the criteria.
- (d) Different search criteria provide different and relevant results.

- **Post-conditions:** The search results displayed match the entered criteria.

4. ACCEPTANCE TEST - FR-11 User Interface - Provide interface for influencers to seek partnerships. - **TEST PASSED**



- **Test Case:** Test if the application provides an interface for influencers to seek partnerships.
- **Preconditions:**
  - (a) The application is installed and accessible.
  - (b) The user is registered as an "Influencer" and is logged into the application.
- **Steps:**
  - (a) Log into the system as an Influencer.
  - (b) Navigate to the Brands page or similar functionality.
  - (c) Browse through the list of brands available.
  - (d) Select a brand that the influencer is interested in partnering with.
  - (e) On the selected brand's profile, locate the "Seek Partnership" button or similar functionality.
  - (f) Click on the "Seek Partnership" button.
  - (g) Check if a partnership request form or similar feature is displayed.
  - (h) Fill out any necessary information or forms related to the partnership request, if required.
  - (i) Submit the partnership request.
  - (j) Check the application's response for confirmation of successful submission.
  - (k) Go to the 'Partnership Requests' section or similar functionality to confirm that the partnership request has been sent to the brand.
- **Acceptance criteria:**
  - (a) The application provides a clear and intuitive interface for influencers to browse and select brands.
  - (b) The "Seek Partnership" function should be accessible and easy to use from the brand's profile page.
  - (c) Upon clicking "Seek Partnership", a form or similar feature for making a partnership request should be displayed.
  - (d) Upon submission, the influencer should receive a confirmation message saying the partnership request has been sent successfully.
  - (e) The partnership request should be visible in the 'Partnership Requests' section.
- **Post-conditions:** The partnership request is successfully recorded and sent to the brand for consideration.

5. ACCEPTANCE TEST - FR-15 Integration - Allow brands to set criteria for campaign

= **TEST PASSED**

- **Test Case:** Test if the application allows brands to set criteria for campaigns.
- **Preconditions:**
  - (a) The application is installed and accessible.

(b) The user is registered as a "Brand" and is logged into the application.

- **Steps:**

- (a) Log into the system as a Brand.
- (b) Navigate to the Campaigns page.
- (c) Click on the "Add campaign" button.
- (d) In the new campaign form, check for fields where the brand can set criteria such as price rate for campaign, participants limit, due date and thematic categories.
- (e) Fill in all necessary fields and set the desired criteria.
- (f) Submit the new campaign form.
- (g) Verify that the new campaign appears in the list of the brand's campaigns with the correct criteria.
- (h) Click on the newly created campaign to view its details.
- (i) Verify that all the criteria are displayed correctly.

- **Acceptance criteria:**

- (a) The application provides a clear and intuitive interface for brands to set criteria for campaigns.
- (b) All the necessary fields for setting criteria are available in the add campaign form.
- (c) Upon submission, the new campaign is successfully created with the correct criteria.
- (d) The newly created campaign is listed among the brand's campaigns.
- (e) The criteria of the newly created campaign are correctly displayed when viewing the campaign's details.

- **Post-conditions:** The new campaign is successfully recorded in the application with the correct criteria.

## Non-functional Requirements

1. ACCEPTANCE TEST - NFR-5 Portability - Design using platform-independent technologies - **TEST PASSED**

- **Test Case:** Test if the application can run on different platforms.

- **Preconditions:**

- (a) The application is designed and ready for testing.
- (b) Access to various platforms (browsers, operating systems, devices).

- **Steps:**

- (a) Open the application on a Windows machine using different browsers (Chrome, Firefox, Edge, etc.). Check if the application runs without errors.

- (b) Repeat step 1 on a MacOS machine.
- (c) Access the application on various mobile devices (Android, iOS) to see if it is responsive and works correctly.
- (d) For each platform, check the functionality by navigating through the application and using its features.
- (e) Note any errors or problems that occur.

- **Acceptance criteria:**

- (a) The application should load correctly on all platforms.
- (b) All the functionalities should work as expected on all platforms.
- (c) The layout and appearance of the application should be correct and consistent across all platforms.
- (d) No errors should occur during the testing.

- **Post-conditions:** The application is confirmed to be designed using platform-independent technologies.

## 2. ACCEPTANCE TEST - NFR-9 Usability - Have a consistent and intuitive user interface - **TEST PASSED**

- **Test Case:** Test if the application has a consistent and intuitive user interface.

- **Preconditions:**

- (a) The application is designed and ready for testing.
- (b) The user is logged in the application.

- **Steps:**

- (a) Navigate through the application, going through every screen.
- (b) Observe the UI elements (buttons, menus, input fields, typography, colors, images, layout, etc.) for consistency.
- (c) Check if the navigation menu is consistent across all pages.
- (d) Check if the action buttons are in consistent locations across the application.
- (e) Check if error messages are clear and provide guidance to the user on how to correct the problem.
- (f) Check if terminology and language used in the application are consistent and easy to understand.
- (g) Perform common tasks in the application to test if workflows are intuitive and straightforward.
- (h) Check for loading times and responsiveness of UI elements.
- (i) Check if there are helpful features like tool tips, auto-completion, etc.

- **Acceptance criteria:**

- (a) The application's UI is consistent throughout.
- (b) Navigation is intuitive and does not confuse the user.
- (c) UI elements behave as expected.

- (d) Error messages are clear and guide the user to rectify the problem.
- (e) The application does not have long loading times and the UI is responsive.
- **Post-conditions:** The application is confirmed to have a consistent and intuitive user interface.

#### 4.2.2 Remaining Tests

##### Unit Tests:

✓ Tests passed: 4 of 4 tests – 635 ms

Figure 7: Example of unit tests for BrandService.

```
@Test
public void testLogin() throws AuthenticationException {
    String email = "contact@nike.com";
    String password = "Nike123";
    String name = "Nike";
    String nip = "1234567890";
    String description = "The world's largest athletic apparel company, Nike is best known for its footwear, apparel, and equipment.";

    BrandLoginDTO loginDTO = new BrandLoginDTO(name, password);
    Brand brand = Brand.builder()
        .email(email)
        .password(password)
        .name(name)
        .nip(nip)
        .description(description)
        .build();

    Mockito.when(branchRepo.findByName(loginDTO.name())).thenReturn(Optional.of(branch));

    Brand result = brandService.login(loginDTO);

    Assertions.assertEquals(branch, result);
    Mockito.verify(branchRepo).findByName(loginDTO.name());
}
```

Figure 8: An example of code snippet, of a unit test testing brand login.

##### Acceptance Tests 1:

- **Test Case:** Verify if a brand can successfully login to the application. **TEST PASSED**
- **Preconditions:**
  - (a) The application is installed and accessible.
  - (b) The brand account is registered in the system.
- **Steps:**
  - (a) Access the application's login page.

- (b) Enter the brand's credentials (login and password).
- (c) Click on the "Login" button.
- (d) Validate if the brand is successfully logged into the application.

- **Acceptance criteria:**

- (a) The brand should be able to login without any errors.
  - (b) After successful login, the brand should be redirected to the brand's main page.

- **Post-conditions:** The brand user is logged into the application and can access the brand-specific features and functionalities.

### Acceptance Tests 2:

- **Test Case:** Test if an influencer can register a new account in the application. **TEST PASSED**

- **Preconditions:**

- (a) The application is installed and accessible.

- **Steps:**

- (a) Access the application's registration page.
  - (b) Fill in the required information (e.g., name, email, password) and additional informations in the influencer registration form.
  - (c) Submit the registration form.
  - (d) Verify if the registration is successful and the influencer is redirected to the influencer main page.

- **Acceptance criteria:**

- (a) The influencer should be able to register a new account without encountering any errors.
  - (b) After successful registration, the influencer should be automatically logged into the application.
  - (c) The influencer should have access to influencer-specific features and functionalities.

- **Post-conditions:** The influencer account is registered in the application, and the influencer can login and utilize the application's features.

### Acceptance Tests 3:

- **Test Case:** Validate application readiness for production use. **TO BE CONDUCTED**

- **Preconditions:**

- (a) The application is installed and configured in the production environment.

- (b) All necessary resources (servers, databases, APIs, etc.) are properly set up and accessible.
- **Steps:** Perform the following tests to evaluate the operational readiness of the application:
  - (a) System Availability: Verify that the application is accessible and available for use.
  - (b) Performance Testing: Test the application's performance under normal and peak load conditions, ensuring that response times and throughput meet the specified requirements.
  - (c) Scalability Testing: Evaluate the application's ability to handle increased user load by scaling up resources or adding additional instances.
  - (d) Security Testing: Conduct security assessments to ensure that the application's sensitive data is protected, access controls are in place, and potential vulnerabilities are addressed.
  - (e) Backup and Recovery Testing: Test backup and recovery procedures to ensure data can be successfully restored in case of system failures or data loss.
  - (f) Monitoring and Alerting: Set up and validate monitoring systems to track application health, performance, and other key metrics. Verify that appropriate alerts are triggered for critical events.
  - (g) Acceptance Tests Review: Review Acceptance Tests results to ensure that all critical acceptance criteria have been successfully validated by the users.
- **Acceptance criteria:**
  - (a) The application demonstrates high availability and can be accessed without significant downtime.
  - (b) Performance testing indicates that the application meets response time and throughput requirements under normal and peak load conditions.
  - (c) Scalability testing shows that the application can handle increased user load by effectively utilizing additional resources.
  - (d) Security testing identifies and addresses potential vulnerabilities, ensuring the application's data and access controls are robust.
  - (e) Backup and recovery testing confirms that data can be successfully backed up and restored, reducing the risk of data loss.
  - (f) Monitoring and alerting systems are in place and functioning correctly, providing real-time visibility into application health and performance.
  - (g) Acceptance tests review confirms that all critical acceptance criteria have been met and validated by the users.
- **Post-conditions:** The application is deemed operationally ready and suitable for production use.