

Sprawozdanie końcowe z realizacji projektu pt.
„Noticeboard”

Wykonał: Franciszek Wysocki
Sprawdzający: mgr inż. Paweł Zawadzki
Data: 04-06-2021

Spis treści

1 Cel projektu	3
2 Wykorzystane technologie	3
3 Zmiany względem specyfikacji	3
4 Uruchomienie aplikacji	3
5 Przykładowe scenariusze użytkowania	4
6 Reakcje na błędy ze strony użytkownika	10
7 Podsumowanie i wnioski	11

1 Cel projektu

Celem projektu było stworzenie serwisu ogłoszeniowego - aplikacji webowej w architekturze klient-serwer, która umożliwia m.in. rejestrację i logowanie, wyświetlanie, dodawanie, edytowanie i kasowanie pojedyńczych ogłoszeń oraz pobieranie (paginacja, lazy loading), sortowanie i filtrowanie stron z ogłoszeniami.

2 Wykorzystane technologie

Baza danych:

- PostGreSQL;
- Docker.

Back-end:

- Java;
- Spring Boot.

Front-end:

- JavaScript;
- React;
- Redux;
- Bootstrap 5;
- Css;
- Html.

Wersje zostały wymienione w specyfikacji implementacyjnej.

3 Zmiany względem specyfikacji

Zaplanowana struktura projektu, wykorzystane narzędzia oraz sposoby testowania nie zmieniły się znacząco względem założeń.

Zmianie uległo natomiast kilka funkcjonalności: dodanie profilu użytkownika, dodanie możliwości zmiany hasła, możliwości skasowania konta, aktywację konta poprzez e-mail.

Z podstawowych założeń, które zakładałem nie implementowałem jedynie podziału na kategorie (uznałem to za zbędne i wolałem skupić się na innych dodatkowych funkcjach).

4 Uruchomienie aplikacji

Aplikacja w wersji produkcyjnej jest uruchomiona na stałe i można z niej korzystać pod adresem <https://noticeboard.pl>.

Aplikację starałem się zrobić możliwie bezpieczną (hashowane hasła użytkownika, szyfrowane hasła do bazy danych, maila itd... w pliku application.yml, szyfrowane połączenie HTTPS), jednak wolałbym nie odpowiadać za pozostawione tam dane, dlatego polecam skorzystać z tymczasowego adresu e-mail, generatora imion i nazwisk oraz generatorka twarzy, a do samych treści ogłoszeń np. z innego serwisu ogłoszeniowego.

Aby uruchomić aplikację w trybie deweloperskim należy oddziennie uruchomić (1) serwer z bazą danych, (2) back-end i (3) front-end.

1. Uruchomienie serwera bazy danych PostgreSQL w dockerze (port: 5555):
`docker-compose up database`

2. Uruchomienie aplikacji back-endowej (port: 8080):

Aby utworzyć plik wykonywalny można wykorzystać Mavenu:

```
mvn clean install
```

a następnie:

```
java -Djasypt.encryptor.password=secretKey -jar noticeboard-0.0.1-SNAPSHOT.jar
```

W miejsce secretKey należy podać klucz umożliwiający odszyfrowanie haseł do bazy danych, skrzynki mailowej oraz magazynu na klucze (SSL), znajdujących się w pliku application.yml.

3. Uruchomienie aplikacji front-endowej (port: 3000):

Tutaj również na początku należy skorzystać z menadżera pakietów (zewnętrzne pakiety nie są przechowywane w repozytorium):

```
npm install
```

a następnie można uruchomić aplikację:

```
npm start
```

Testy można uruchomić niezależnie (nie wymagają one podania klucza):

1. Back-end:

```
mvn test
```

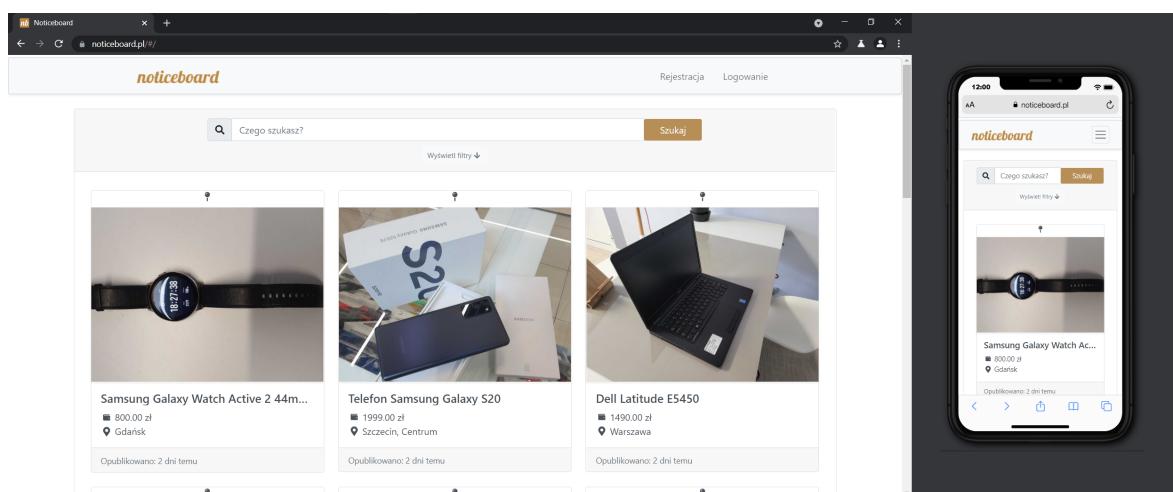
2. Front-end:

```
npm test
```

5 Przykładowe scenariusze użytkowania

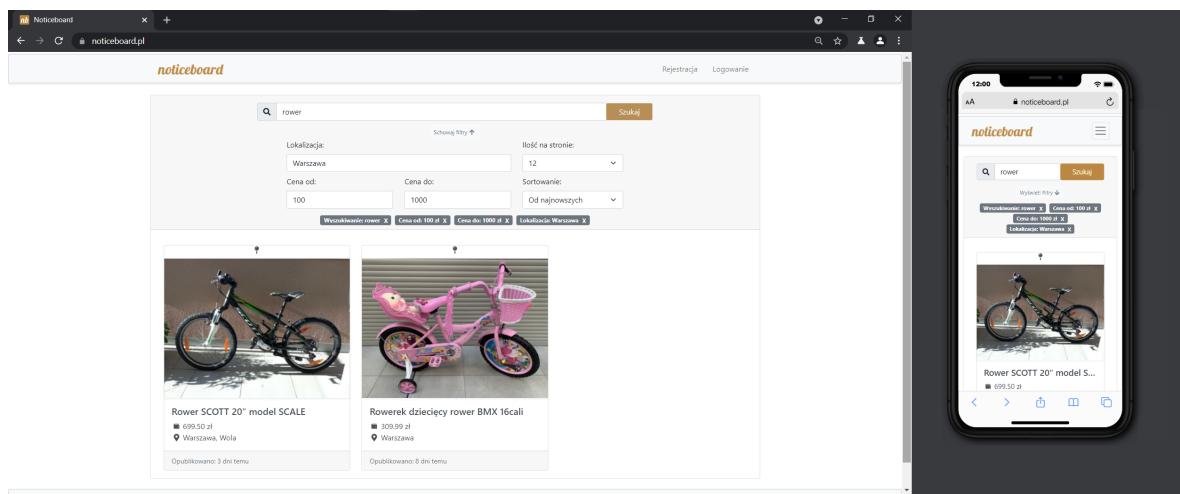
- Użytkownik jest ogłoszeniobiorcą.

1. Użytkownik przechodzi pod wskazany adres URL <https://noticeboard.pl> i bez zalogowania może przeglądać strony z ogłoszeniami.



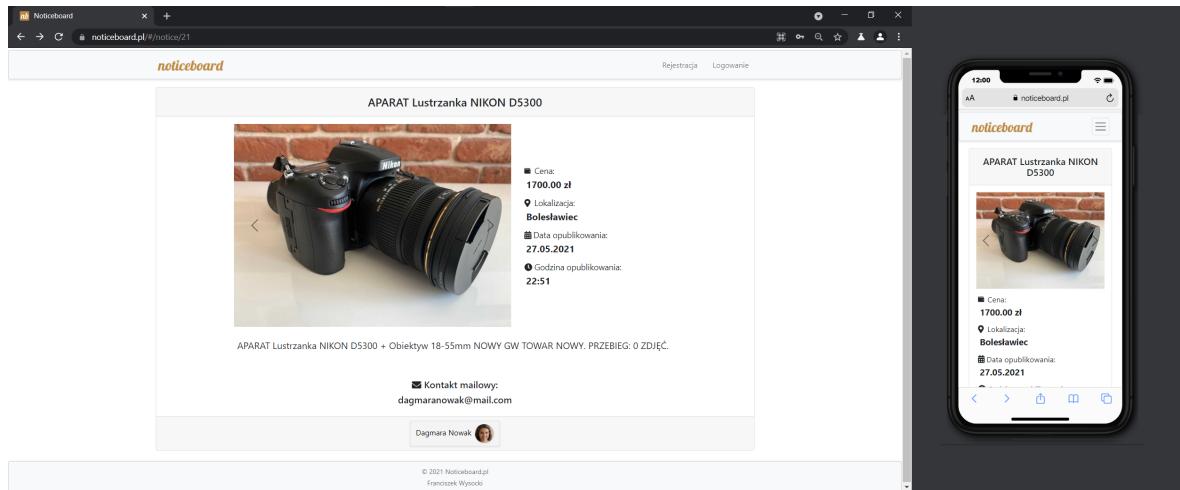
Zrzut ekranu 1

2. Użytkownik filtruje i sortuje ogłoszenia.



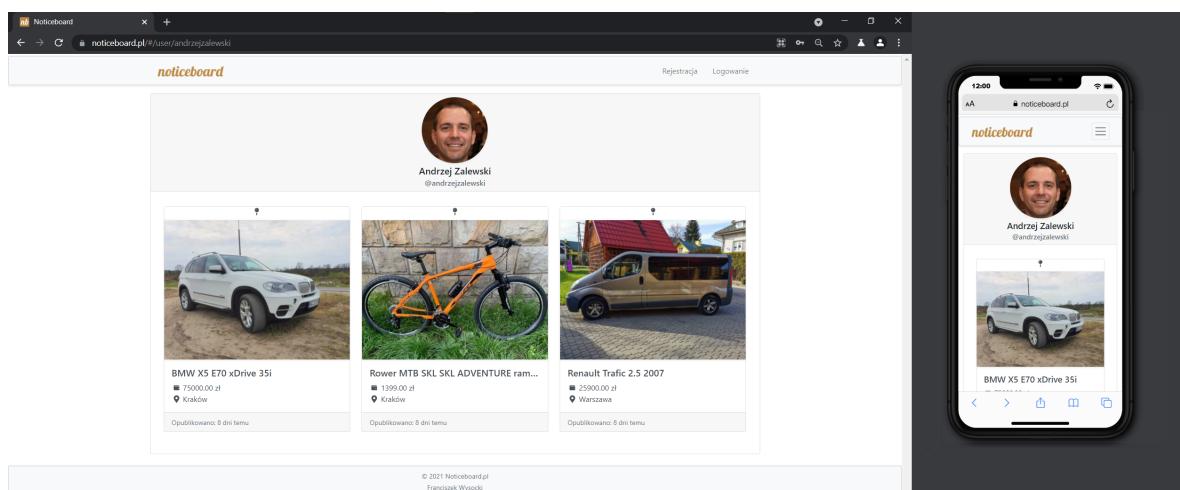
Zrzut ekranu 2

- Użytkownik otwiera ogłoszenie i wyświetla szczegółowy opis, dane sprzedawcy i jego adres email do kontaktu.



Zrzut ekranu 3

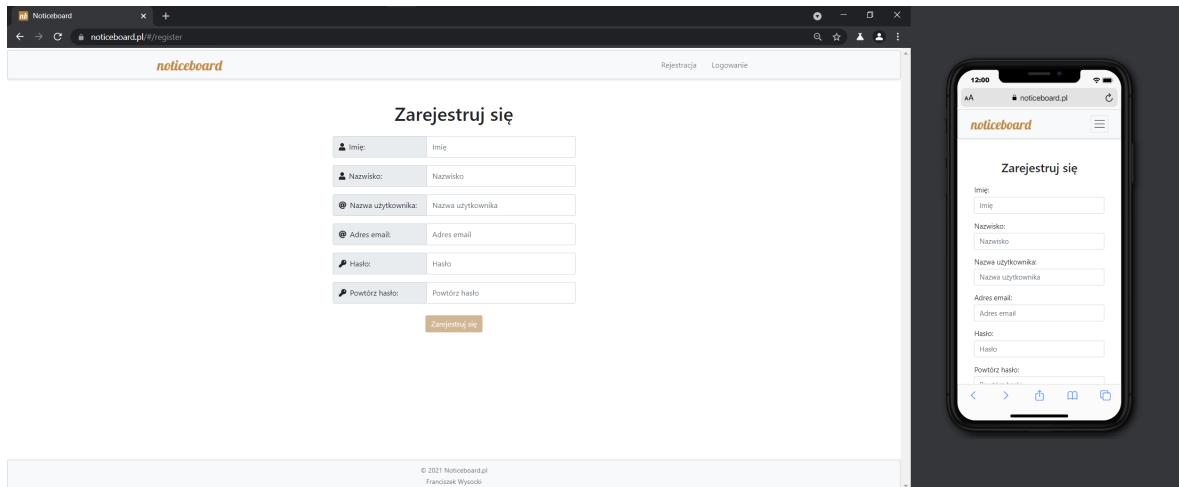
- Użytkownik przechodzi na stronę (w tym przypadku innego) sprzedawcy i przegląda jego inne ogłoszenia.



Zrzut ekranu 4

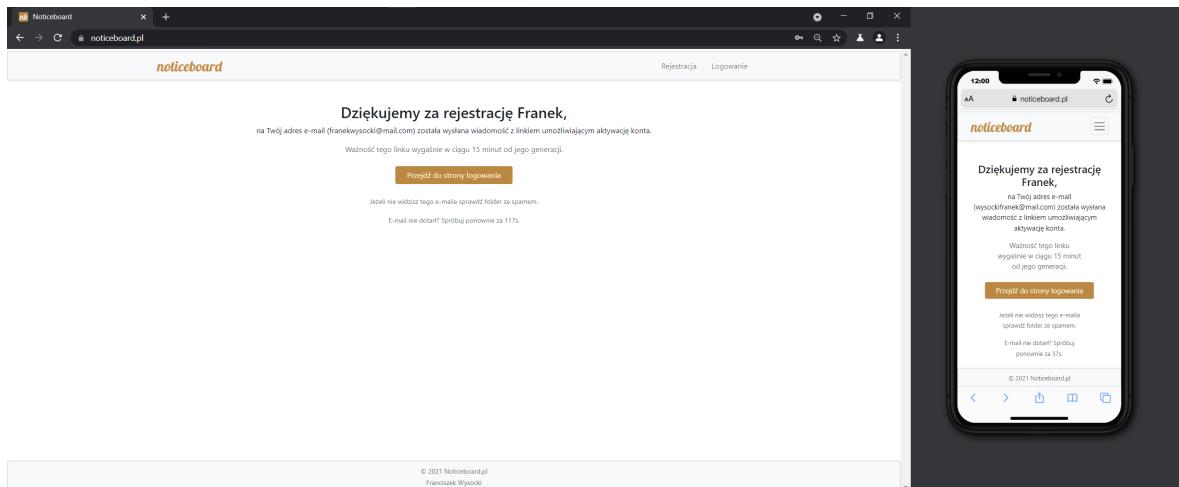
- Użytkownik zakłada konto oraz loguje się:

- Użytkownik przechodzi na stronę rejestracji i wprowadza wymagane dane.



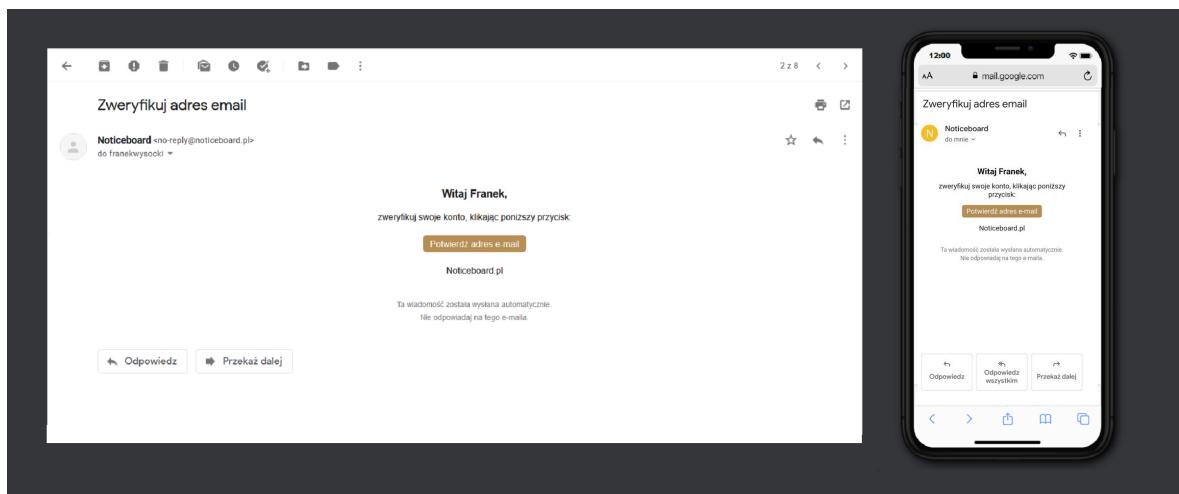
Zrzut ekranu 5

- Użytkownik zostaje poinformowany o wysłaniu maila z linkiem aktywacyjnym.



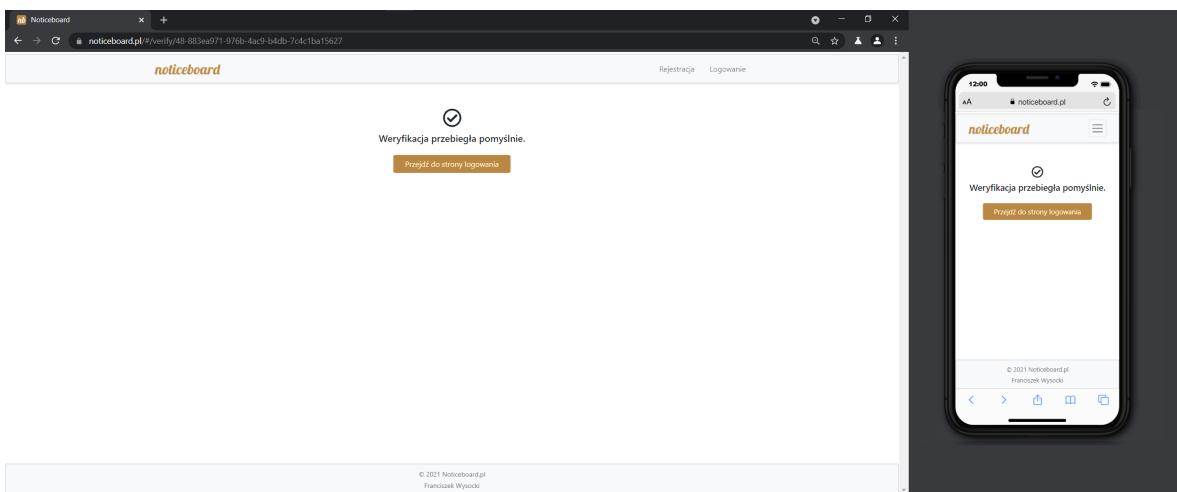
Zrzut ekranu 6

- Użytkownik otwiera skrzynkę mailową i kliką w link aktywacyjny.



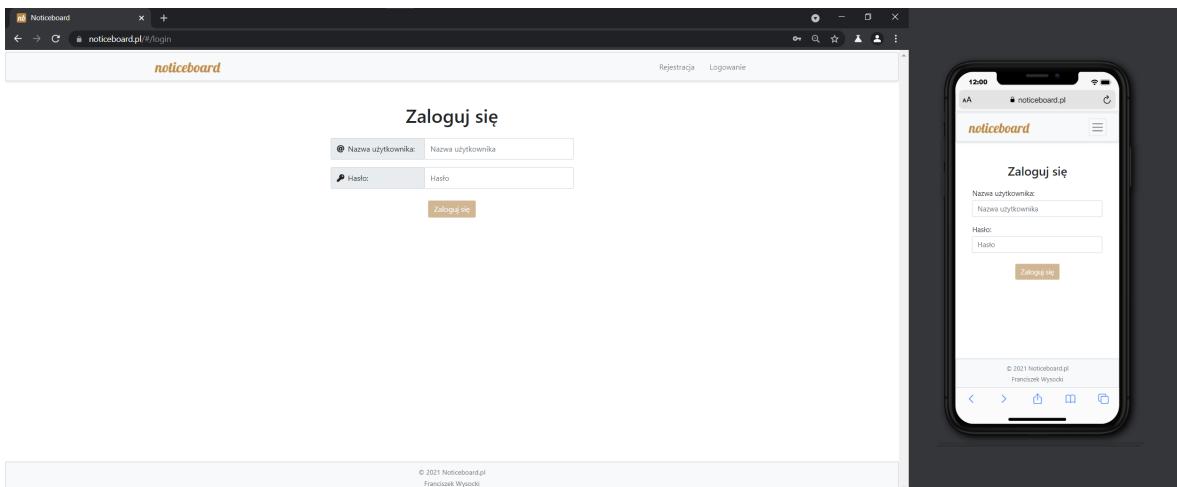
Zrzut ekranu 7

- Użytkownik zostaje poinformowany o sukcesie operacji i ma możliwość przejścia do strony logowania.



Zrzut ekranu 8

5. Użytkownik loguje się.

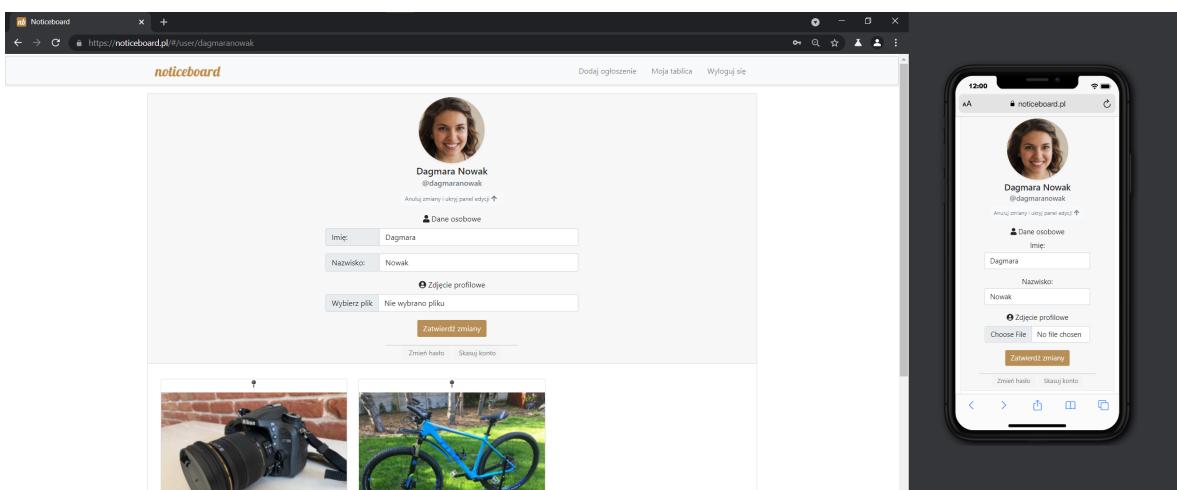


Zrzut ekranu 9

6. Użytkownik zostaje przekierowany na stronę główną.

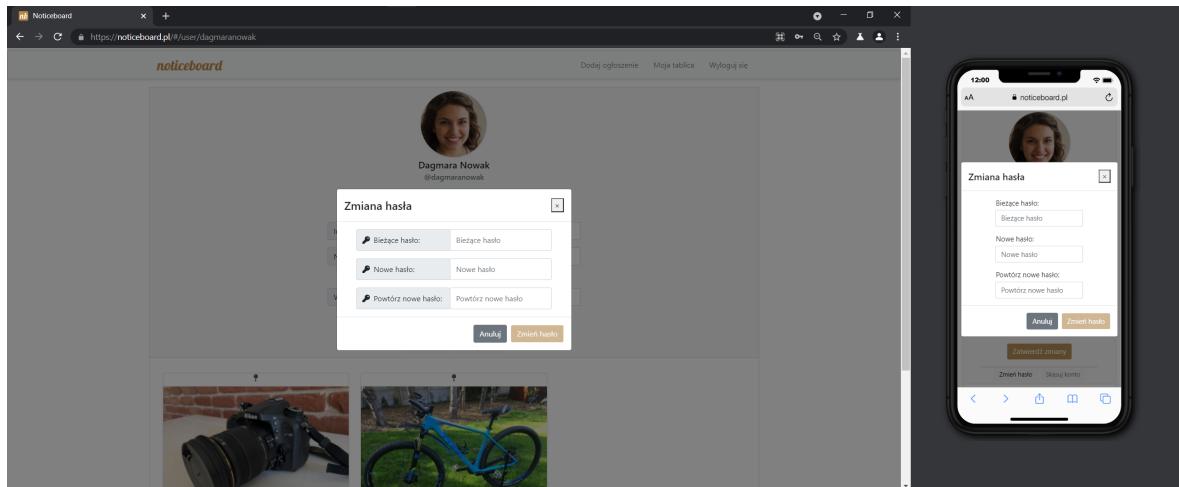
- Użytkownik zarządza swoim kontem

- Zalogowany użytkownik przechodzi na swój profil. Jedynie on ma możliwość wyświetlenia panelu edycji. Użytkownik w samym panelu może zmienić zdjęcie profilowe, imię oraz nazwisko.



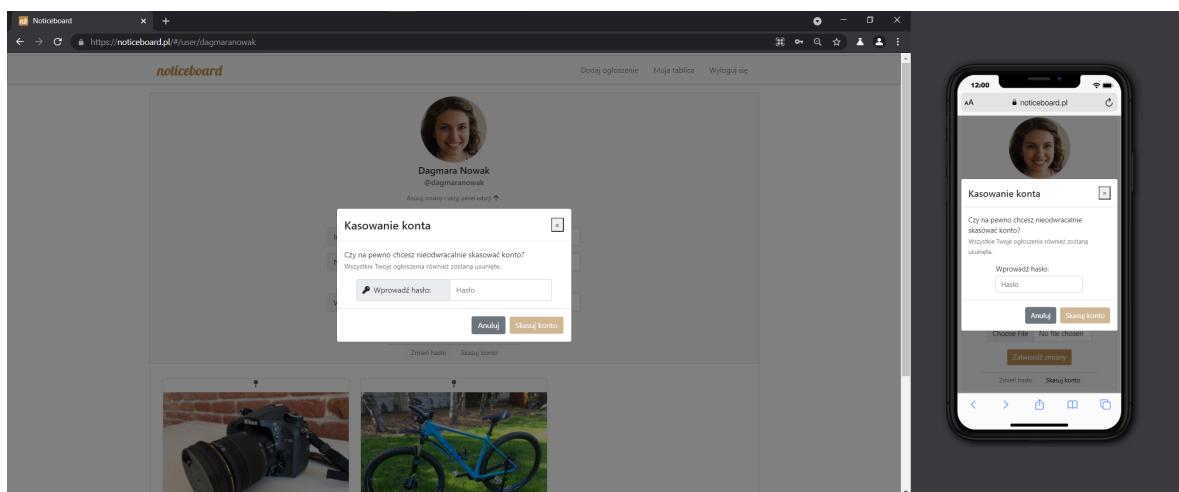
Zrzut ekranu 10

2. Użytkownik kilka „Zmień hasło”.



Zrzut ekranu 11

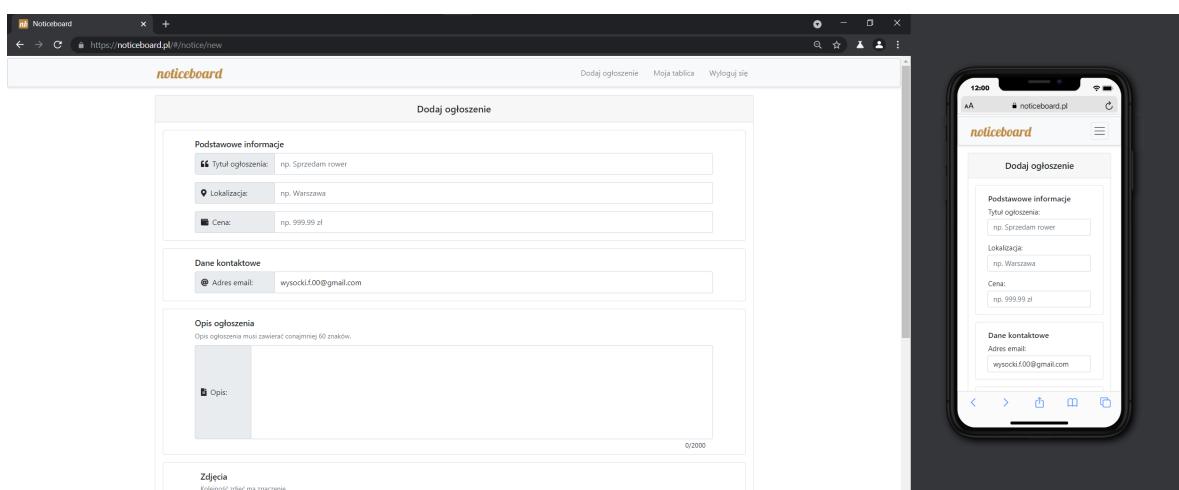
3. Użytkownik kilka „Skasuj konto”.



Zrzut ekranu 12

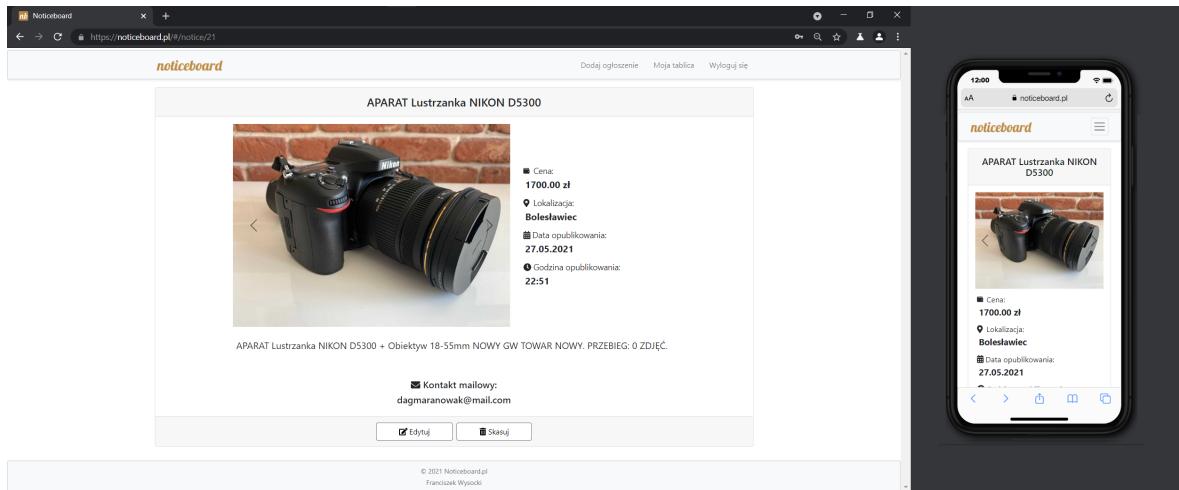
- Użytkownik jest ogłoszeniodawcą.

1. Zalogowany użytkownik kliknie „Dodaj ogłoszenie” na pasku nawigacyjnym.
2. Użytkownik podaje tytuł, opis, cenę, lokalizację, adres email oraz załadowuje zdjęcia, po czym zatwierdza operację.



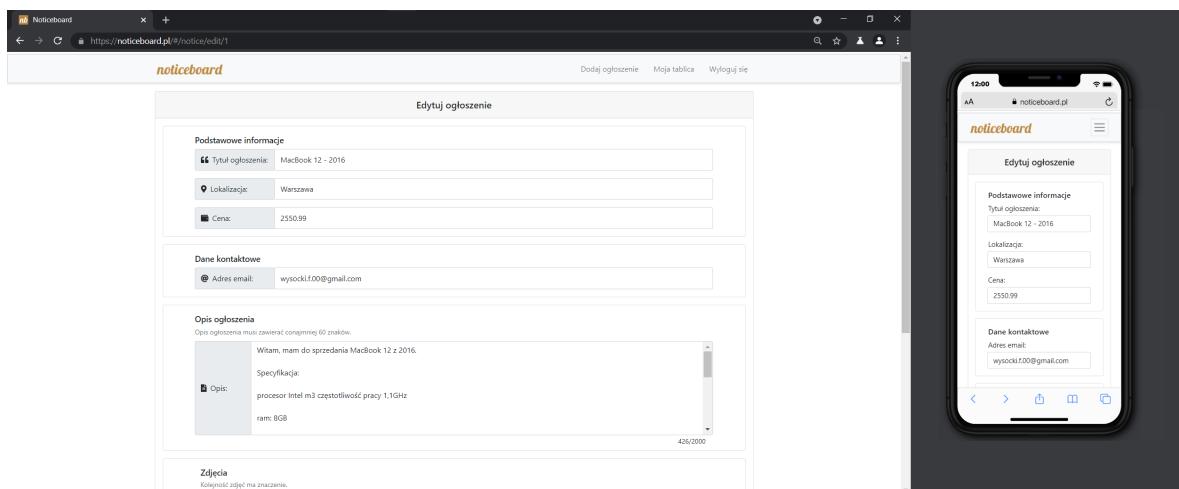
Zrzut ekranu 13

3. Użytkownik przechodzi na stronę swojego ogłoszenia (zamiast linku do jego profilu ma przyciski do skasowania i edycji ogłoszenia).



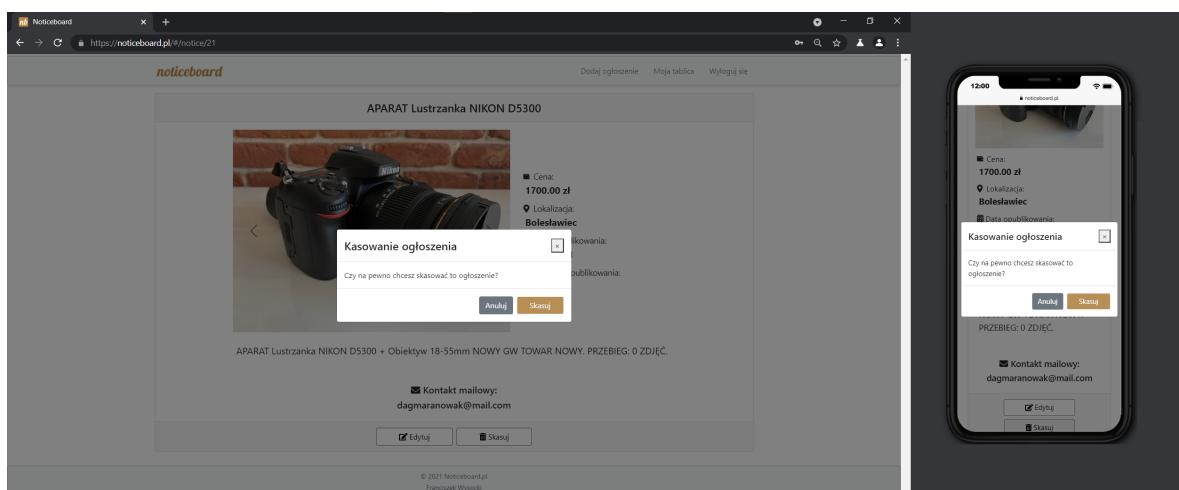
Zrzut ekranu 14

4. Użytkownik edytuje swoje (np. błędne) ogłoszenie.



Zrzut ekranu 15

5. Użytkownik kasuje swoje (np. nieaktualne) ogłoszenie.

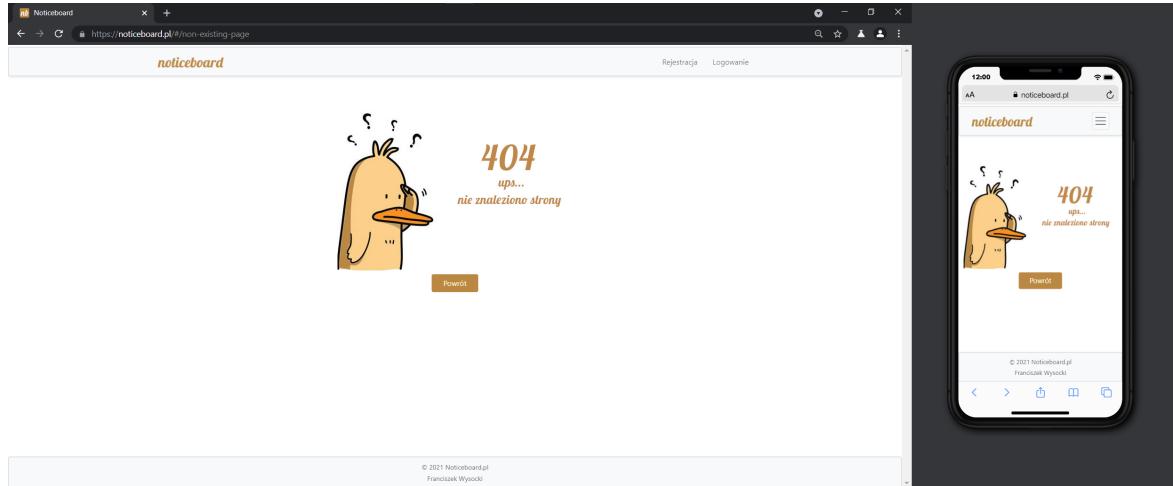


Zrzut ekranu 16

6. Użytkownik wylogowuje się i zostaje przekierowany na stronę główną.

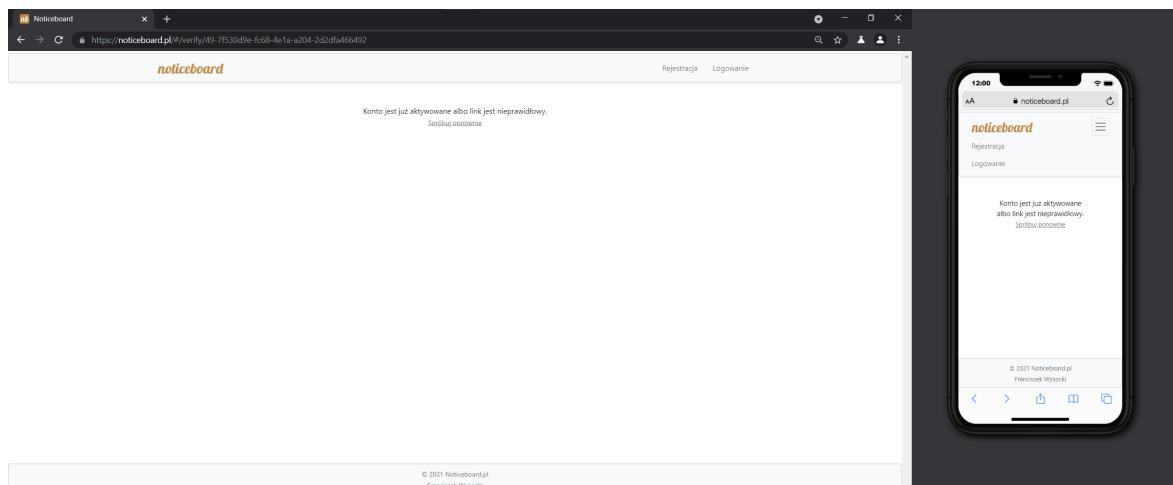
6 Reakcje na błędy ze strony użytkownika

- Użytkownik przechodzi pod niepoprawny adres URL:



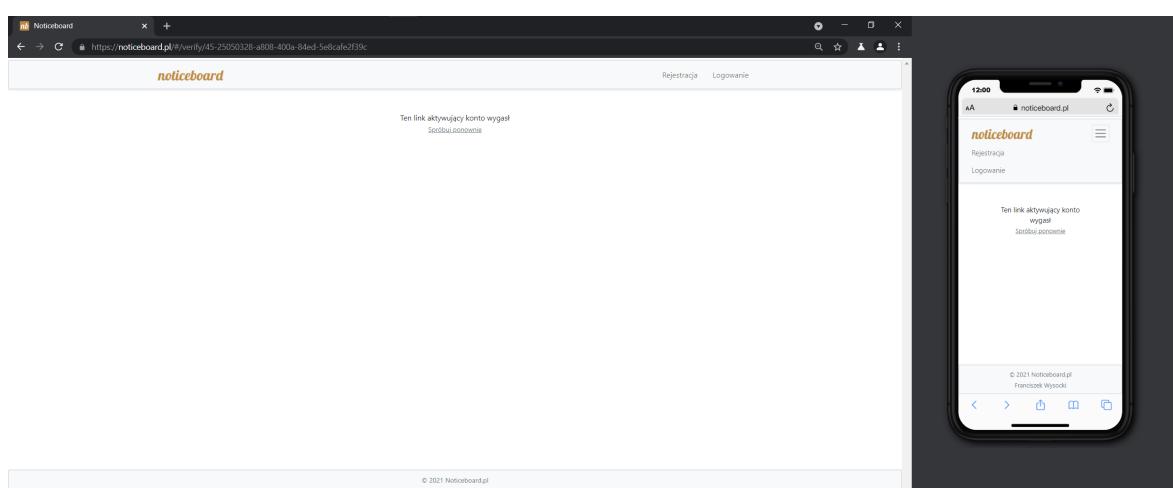
Zrzut ekranu 17

- Użytkownik kilka razy kliknie w uszkodzony link aktywujący konto:



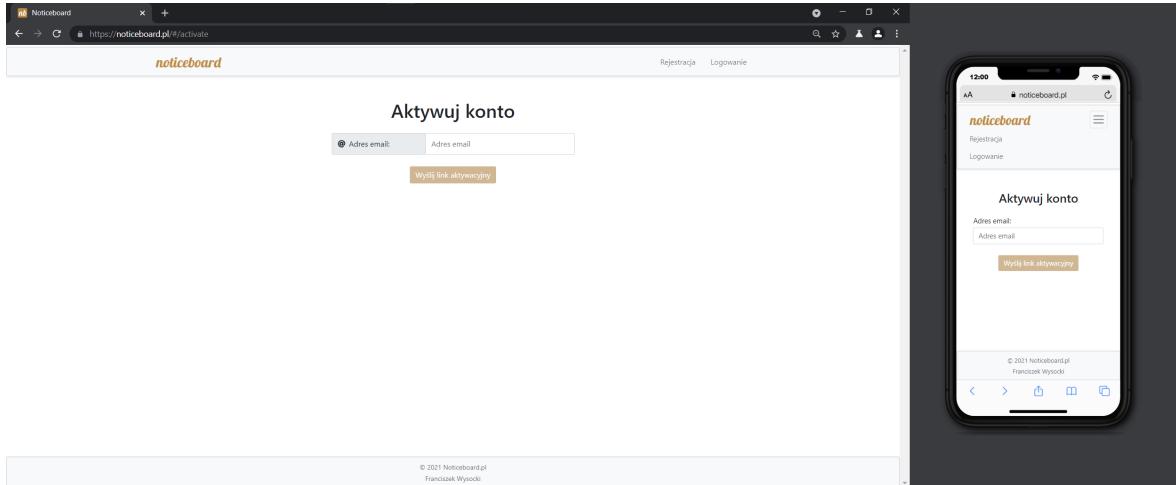
Zrzut ekranu 18

- Użytkownik kilka razy kliknie w wygasły link aktywujący konto:



Zrzut ekranu 19

Po kliknięciu „spróbuj ponownie” użytkownik ma możliwość wysłania ponownego linku aktywacyjnego:



Zrzut ekranu 20

- Pozostałe obsługiwane błędy:

- Pozostawienie pustego pola w formularzu lub za krótki/długi ciąg znaków (komunikat);
- Niedopasowanie hasła/adresu e-mail do wzorca np. podczas rejestracji (komunikat);
- Próba zalogowania, rejestracji, aktywacji konta, będąc zalogowanym (przekierowanie);
- Próba zarządzania czymś ogłoszeniem/kontem (przekierowanie + zablokowanie po stronie backendu).

7 Podsumowanie i wnioski

Największą przyjemność i jednocześnie największą trudność sprawiało mi implementowanie logiki aplikacji m.in. zarządzanie zdjęciami (od odczytu po serwowanie z cachowaniem), wysyłanie e-maili z tokenami (ważny-mi 15 minut) i sam CRUD. Dużo czasu poświęciłem na samo wdrożenie aplikacji z certyfikatem SSL, co nie okazało się bardzo skomplikowane, ale brakowało mi wiedzy, jak to zrobić. Bardzo spodobała mi się praca z Lombokiem, Mapstructem i Spring Data JPA, które pozwoliły na korzystanie z metod bez implementowania ich.

Front-end aplikacji chciałem stworzyć prosty i elegancki. Starałem się też, aby aplikacja była responsywna i przetestowałem ją na kilku urządzeniach i przeglądarkach. Praca nad front-endem była momentami uciążliwa np. przesuwanie się elementów po rozwinięciu panelu/filtrów, jednak udało mi się to wyeliminować.

Sam projekt uważam za udany i jestem z niego bardzo zadowolony. Zrozumiałem i nauczyłem się bardzo wielu rzeczy, które myślę, że z pewnością wykorzystam w przyszłości.