

Politechnika Wrocławска
Wydział Informatyki i Telekomunikacji

Kierunek: **Inżynieria systemów (INS)**

**PRACA DYPLOMOWA
INŻYNIERSKA**

**System webowy wspierający katalogowanie
produktów sklepu oraz składanie zapytań
przez klientów**

Mikołaj Kaźmierczak

Opiekun pracy
dr inż. Michał Szczepanik

Słowa kluczowe: aplikacja webowa, BaaS, panel administracyjny, katalog sklepowy

WROCŁAW 2022

TYTUŁ PRACY

System webowy wspierający katalogowanie produktów sklepu oraz składanie zapytań przez klientów

STRESZCZENIE

Cel niniejszej pracy dotyczy opracowania systemu internetowego dla firmy zajmującej się sprzedażą gadżetów reklamowych. System ma umożliwiać administratorom katalogowanie produktów sklepu, a potencjalnym nabywcom wysyłanie zapytań do administratorów. Obecny system wykorzystywany przez firmę ma kilka wad, w tym powolny czas ładowania, trudny w obsłudze interfejs oraz brak spersonalizowanych kalkulatorów cenowych. Nowy system rozwiązuje te problemy, jest spersonalizowany i dostosowany do administratorów. System zawiera bazę danych, interfejs programowania aplikacji oraz panel administracyjny i stronę dla potencjalnych nabywców.

Wdrożona aplikacja internetowa spełnia oczekiwania klienta oraz wszystkie określone wymagania funkcjonalne i niefunkcjonalne. System pozwala na logowanie się administratorów i edytowanie elementów systemu, a także daje możliwość ukrywania ich przed potencjalnymi nabywcami. Użytkownicy mogą przeglądać publiczną stronę i wysyłać zapytania do administratorów. System ma również potencjał do dalszego rozwoju, jak na przykład automatyczna, cykliczna aktualizacja cen i stanów magazynowych korzystając z interfejsów programowania aplikacji zewnętrznych dostawców towaru. Projekt został zrealizowany w metodyce zwinnej i przeszedł testy użyteczności dla administratorów systemu. System jest funkcjonalnie kompletnym narzędziem gotowym do zastosowania w świecie rzeczywistym.

THESIS TITLE

Web system supporting product catalog management and client queries

ABSTRACT

The goal of this work is to develop a web-based system for a company that sells promotional gadgets. The system is to allow administrators to catalog the store's products and potential buyers to send inquiries to the administrators. The current system used by the company has several drawbacks, including slow loading times, a difficult-to-use interface and a lack of personalized price calculators. The new system solves these problems and is personalized and tailored to the administrators. The system includes a database, an application programming interface, and an administration panel and page for potential buyers.

The implemented web application meets the client's expectations and all specified functional and non-functional requirements. The system allows administrators to log in and edit elements of the system, and provides the ability to hide them from potential buyers. It also supports creating personalized calculators. Users can browse the public site and send inquiries to administrators. The system also has the potential for further development, such as automatic cyclic updating of prices and inventory using the application programming interfaces of third-party commodity suppliers. The project was implemented in an agile methodology and passed usability tests for system administrators. The system can be considered a functionally complete tool ready for use in the real world.

SPIS TREŚCI

Wstęp	3
Geneza problemu	3
Cel pracy	3
Zakres pracy	4
Motywacja	4
1. Podstawy teoretyczne	5
1.1. Technologie webowe	5
1.2. Czysty kod	5
1.3. Słownik pojęć	6
1.3.1. API	6
1.3.2. Framework	6
1.3.3. Backend, Frontend, Fullstack	6
1.3.4. BaaS	7
1.3.5. Ciasteczka	7
1.3.6. UX	7
1.3.7. SEO	8
2. Projekt	9
2.1. Konsultacje z klientem	9
2.1.1. Przypadki użycia	10
2.2. Wymagania funkcjonalne	10
2.3. Wymagania niefunkcjonalne	11
2.4. Przegląd dostępnych rozwiązań	12
2.4.1. Shopify	12
2.4.2. WooCommerce	12
2.4.3. MedusaJS	13
2.5. Wybór narzędzi	14
2.6. Architektura systemu	15
2.6.1. Baza danych	16
2.6.2. Directus	16
2.6.3. Heimdall	16
2.6.4. SvelteKit	17
2.7. Projekt graficzny	18
2.7.1. Ikony	18
2.7.2. Panel administracyjny	18

2.7.3. Publiczna strona internetowa	19
3. Implementacja	20
3.1. Widoki w panelu administracyjnym	20
3.1.1. Panel logowania	20
3.1.2. Panel aktywności	21
3.1.3. Kalkulacje	21
3.1.4. Produkty	27
3.1.5. Kolory	31
3.1.6. Kategorie	32
3.1.7. Menu	32
3.1.8. Strony	34
3.1.9. Fragmenty	34
3.1.10. Zapytania	35
3.1.11. Biblioteka plików	36
3.2. Strona internetowa dla użytkownika publicznego	37
3.2.1. Menu	38
3.2.2. Lista produktów	39
3.2.3. Produkt	40
3.3. Testy użyteczności	42
3.3.1. Efektywność systemu	45
4. Dalsze prace	52
Podsumowanie	54
Bibliografia	55
Spis rysunków	57
Spis tabel	60
Dodatki	61
A. Kod źródłowy	62

WSTĘP

W niniejszej pracy opracowany zostanie system webowy, który umożliwi katalogowanie produktów sklepu oraz pozwoli potencjalnym nabywcom tych produktów wysłać zapytanie do administratorów systemu.

GENEZA PROBLEMU

Firma REED Dariusz Kaźmierczak i Artur Fornalczyk sp.j. [12] (zwana dalej klientem) to mikroprzedsiębiorstwo zajmujące się sprzedażą gadżetów reklamowych. Istotną częścią jego działalności jest też znakowanie gadżetów poprzez użycie między innymi technik obróbki laserowej, ale także wielu innych. Klient potrzebuje systemu webowego, który spełnia określone przez niego wymagania. Do głównych z nich należą możliwość wprowadzania oraz edycji produktów przez właścicieli sklepu (zwanych dalej administratorami), a także umożliwienie wysyłania zapytań potencjalnemu nabywcy produktów.

Należy zaznaczyć, że klient jest już w posiadaniu systemu o tak określonej funkcjonalności, jednakże według informacji otrzymanych od klienta obecny ma kilka znaczących wad:

- Firma odpowiedzialna za jego wdrożenie wycofuje się z tej części swojej działalności.
- Długie czasy ładowania podstron.
- Według klienta interfejs panelu administracyjnego wymaga zbyt wielu interakcji do wykonania podstawowych czynności administracyjnych. Zaburza to między innymi zasadę trzech kliknięć [28].
- Brakuje w nim między innymi spersonalizowanych kalkulatorów cenowych oraz połączenia z interfejsami programowania aplikacji zewnętrznych dostawców.
- Był wytwarzany jako system uniwersalny dla przedsiębiorstw o podobnym rodzaju działalności. W związku z tym gdy zachodziła potrzeba personalizacji, dodatkowe funkcjonalności były tworzone w krótkim czasie, bez należynej uwagi. Przełożyło się to na niską ich jakość.

CEL PRACY

Celem jest utworzenie spersonalizowanego systemu webowego, który rozwiąże problemy klienta z poprzednim systemem. Nowy system powinien być dostosowany bezpośrednio do administratorów. Autor skupi się w pracy przede wszystkim na opracowaniu panelu administracyjnego serwisu. Mniejsza uwaga zostanie przyłożona do publicznej strony

internetowej skierowanej do potencjalnych nabywców produktów. Powodem jest fakt, że w momencie pisania pracy jej projekt graficzny nie został jeszcze zatwierdzony przez klienta.

Najważniejszym wyznacznikiem osiągnięcia założonego celu będzie potwierdzenie przez klienta, że system spełnia wszystkie jego najważniejsze wymagania.

ZAKRES PRACY

Autor zaimplementuje system webowy, w którego skład wejdą baza danych, interfejs programowania aplikacji (który ułatwi zmiany w bazie danych), a także panel administracyjny i strona skierowana do potencjalnych nabywców. Baza danych i interfejs programowania aplikacji będą zarządzane przez wybrane rozwiązanie BaaS (skrót od ang. *Backend as a Service* lub w wolnym tłumaczeniu "backend jako serwis"). Panel administracyjny oraz strona dla potencjalnych nabywców zostaną zbudowane od podstaw. Panel administracyjny powinien umożliwiać:

- Dodawanie, odczyt, edycję, usuwanie oraz przeszukiwanie produktów, kategorii, elementów menu, podstron, wybranych fragmentów strony, zapytań od potencjalnych nabywców oraz plików.
- Nadawanie produktom określonych właściwości określonych wcześniej przez klienta.
- Odczyt zapytań od potencjalnych nabywców.
- Automatyczną kalkulację cenników na podstawie konfigurowalnych kalkulatorów.

Publiczna strona internetowa powinna umożliwiać potencjalnym nabywcom wyświetlanie kategorii i produktów oraz wypełnienie formularza przy produkcie, którego wyniki administratorzy będą mogli odczytać w panelu administracyjnym.

MOTYWACJA

Główną motywacją jest rozwiązanie problemu klienta. Co więcej Autor od lat interesuje się medium w jakim to rozwiązanie ma funkcjonować, czyli internet.

Ze stron internetowych korzysta bowiem prawie każdy. Dają one możliwość łatwego dotarcia do ogromnej liczby użytkowników niskim kosztem. Myśl, że projekt będzie faktycznie funkcjonować w realnym świecie biznesu bardzo motywuje do działania. Co więcej, metoda implementacji da Autorowi możliwość nabrania doświadczenia w pracy z nowymi dla niego narzędziami. Wybór samych narzędzi został podyktowany między innymi ich wydajnością oraz łatwością użycia. Biorąc pod uwagę skalę projektu są to cechy szczególnie istotne. Warto też zaznaczyć ich wysoką popularność na rynku pracy [18].

1. PODSTAWY TEORETYCZNE

1.1. TECHNOLOGIE WEBOWE

Skrót WWW rozwija się jako *world wide web*, stąd właśnie nazwa technologie webowe. Jest to zbiór technologii funkcjonujących w internecie.

Należą do nich strony internetowe oraz aplikacje webowe. Podział ten jest dość zatarty. W ogólności uznaje się że strona internetowa ma coś użytkownikowi zaprezentować, a aplikacja webowa ma mu pomóc wykonać określone zadanie [2]. Co istotne, aplikacja webowa niekoniecznie musi posiadać interfejs użytkownika. Może ona funkcjonować stricte jako *API* (podsekcja 1.3.1).

Dawniej programistów zajmujących się tym obszarem IT nazywano webmasterami. Dziś korzysta się raczej z określenia programista webowy (ang. *web developer*). Z kolei programistów webowych dzieli się dalej na mniejsze specjalizacje, np. programista frontend, programista backend (podsekcja 1.3.3) [13].

1.2. CZYSTY KOD

Czystość kodu to zagadnienie, które obejmuje jego czytelność i zrozumiałość. Najlepszym wyznacznikiem czy kod jest "czysty" to jego zrozumiałość przez programistę innego niż ten, który go napisał. Inne ważne aspekty czystego kodu to między innymi:

- Łatwość w utrzymaniu.
- Spójny styl i struktura.
- Brak niepotrzebnej złożoności kodu (ang. *overengineering*).
- Modułowość kodu, inaczej wielokrotność użytku.
- Przetestowanie kodu w adekwatny sposób.

Ważnym elementem tworzenia oprogramowania powiązany z czystym kodem jest też refaktoryzacja kodu. Polega na zmianie formy kodu bez zmiany rezultatów jego działania. Najczęściej wiąże się z wykorzystaniem automatycznych testów jednostkowych oraz metodologią *TDD* (ang. *Test Driven Development*), według której testy powinny zostać napisane zanim napisany zostanie sam kod.

Podczas pisania pracy wybrano kilka ważnych książek, którymi inspirowano się projektując architekturę systemu oraz pisząc kod [22, 29, 31, 23, 30, 27, 26, 25].

1.3. SŁOWNIK POJĘĆ

1.3.1. API

Interfejs programowania aplikacji (ang. *application programming interface*) lub *API* pozwala na wymianę danych między aplikacjami i usługami [5].

Dawniej do komunikacji między usługowej wykorzystywano protokół *SOAP*. Pozwalał on na wymianę danych w formacie *XML*. Usprawnieniem tego protokołu było *XML-RPC*. Później na podobnej podstawie został stworzony protokół *JSON-RPC*, który do komunikacji wykorzystywał już inny, w mniemaniu Autora wygodniejszy, format danych - *JSON*. Opracowany został też *REST*, czyli zestaw reguł opartych na architekturze *API*. Większość obecnie wykorzystywanych *API* opiera się właśnie na regułach *REST API*.

1.3.2. Framework

Zgodnie z wiedzą Autora, framework to słowo pochodzące z języka angielskiego, którego nie tłumaczy się bezpośrednio na język polski, natomiast odmiana słowa odbywa się według normalnych jego reguł. Według słownika *Cambridge Dictionary* jest to "struktura, szkielet" (ang. "the basic supporting structure of anything") [10].

Jest to rodzaj narzędzia, które wspomaga rozwiązywanie określonego rodzaju podstawowych problemów związanych z danym zakresem IT w uporządkowany (i często bardziej efektywny) sposób [1]. Przykładem dla przypadku pracy Autora są strony internetowe, które buduje się z trzech podstawowych formatów plików: *HTML*, *CSS*, *JavaScript*. Wszystkie te pliki trzeba w określony sposób zorganizować i sprawić by ze sobą współgrały. Istnieje wiele sposobów żeby tego dokonać, jednak wybrany przez danego programistę framework ma za zadanie z góry rozwiązać tą czasochlonną część pracy nad projektem, aby można było skupić się na rozwiązywaniu bardziej istotnych problemów. Co więcej frameworki z reguły udostępniają też określone funkcje czy metody, które znacznie upraszczają pracę z określonym zestawem technologii.

1.3.3. Backend, Frontend, Fullstack

Zgodnie z wiedzą Autora słowa *backend*, *frontend* i *fullstack* nie mają bezpośredniego tłumaczenia na język polski. Odmienia się je zgodnie z normalnymi zasadami języka polskiego. Spotyka się także potoczne zapisy "front-end", "back-end" oraz "full stack".

Aspekt *backend* (w wolnym tłumaczeniu z ang. "tylna strona") aplikacji webowej opiera się na serwerze, wdrożonym *API*, a także komunikacji i często także zarządzaniu bazą danych. Zajmuje się tym programiści *backend* (ang. *backend developer*). Problemy *backend* rozwiązuje się z wykorzystaniem całej gamy technologii oraz języków programowania. W niniejszej pracy wykorzystane zostanie rozwiązanie *BaaS* (skrót od ang. *Backend as a Service*) czyli *backend* jako serwis. Tego typu serwisy oferują kompleksowe

rozwiązań backendowych, czyli między innymi zarządzanie bazą danych, przechowywanie mediów, a także uwierzytelnianie i autoryzacja użytkowników [19].

Aspekt frontend (w wolnym tłumaczeniu z ang. "przednia strona") aplikacji webowej ma do czynienia głównie z interfejsem użytkownika końcowego. Zajmują się nim programiści frontend (ang. *frontend developer*). Do problemów frontendowych należą między innymi implementacja projektu graficznego strony, komunikacja z różnymi *API* oraz troska o *UX* interfejsu użytkownika. W dzisiejszych czasach do rozwiązywania problemów frontendowych najczęściej wykorzystuje się frameworki frontendowe (np. *React*, *Vue*, *Angular*, *Svelte*) oparte na języku *JavaScript* [18].

Termin fullstack (w wolnym tłumaczeniu z ang. "pełny stos") to połączenie aspektów frontendu i backendu. Każda aplikacja webowa potrzebuje ich obu do działania. Programista, który potrafi zajmować się jednym i drugim aspektem nazywa się programistą fullstack (ang. *fullstack developer*).

1.3.4. BaaS

Backend jako serwis (ang. *Backend as a Service*) to platformy, które oferują całościowe rozwiązania backendowe. Ułatwiają tworzenie aplikacji webowych oraz mobilnych. Najczęściej oferują zarządzanie bazą danych oraz użytkownikami, a także udostępniają API, które ułatwiają interakcję z narzędziem.

1.3.5. Ciasteczka

Ciasteczka (ang. *cookies*) to jedno z narzędzi przeglądarki udostępnianych programistom, które służy do przechowywania tymczasowych danych o użytkowniku bezpośrednio na jego urządzeniu [21]. W uproszczeniu można uznać je za pliki które przechowują określone wartości, np. informacje o preferencjach reklamowych użytkownika, czy też klucz autoryzacyjny użytkownika do serwisu.

1.3.6. UX

Doświadczenie użytkownika (ang. *user experience*) lub *UX* to pojęcie, które określa wrażenia użytkownika po kontakcie z produktem. Im lepsze, tym większa satysfakcja użytkownika z serwisu [25].

Na jakość doświadczenia użytkownika z aplikacją webową czy stroną internetową ma wpływ wiele elementów. Najważniejsze z nich to między innymi korzystanie ze schematów znanych już użytkownikom z innych stron internetowych, a także projektowanie systemu zgodnie z 10 Heurystykami Nielsena. Jest to jednak bardzo szeroka dziedzina, dynamicznie się rozwija, a jej zasady ulegają nieustannym zmianom.

1.3.7. SEO

Optymalizacja strony internetowej pod kątem wyszukiwarek (ang. *Search Engine Optimization*) polega na budowie strony internetowej w taki sposób, aby silniki wyszukiwarek internetowych umieszczały witrynę jak najwyżej w wynikach wyszukiwania. Głównymi sposobami na poprawę *SEO* jest optymalizacja treści oraz struktury strony. Innym czynnikiem pomocnym w usprawnianiu wyników jest ilość powiązań witryny z innymi stronami internetowymi poprzez odnośniki oraz słowa kluczowe i metadane strony, choć w dzisiejszych czasach jest to mniej istotne ze względu na nadużycia i próby manipulacji silników przez autorów stron internetowych poprzez umieszczanie w metadanych fałszywych informacji.

2. PROJEKT

2.1. KONSULTACJE Z Klientem

Rozpoczęcie projektu polegało na przeprowadzeniu z klientem początkowego wywiadu. Dowiedziano się z niego jakie podstawowe funkcje powinien wspierać nowy system oraz w jaki sposób powinien to robić. Poznano też wizję klienta na to jak powinno wyglądać narzędzie. Klient zaprezentował też jak działa stary system, z którego aktualnie korzysta. Do głównych problemów z jakimi zmagał się klient podczas korzystania ze starego narzędzia to między innymi:

- Niska responsywność systemu.

Czasy ładowania podstron są bardzo niezadowalające. Przejście z jednej podstrony do kolejnej wymaga około pół sekundy, co w dłuższym użytkowaniu daje się odczuć.

- Edycja drobnych elementów powoduje przeładowanie strony.

Kiedy użytkownik chce przenieść określony element (na przykład element menu) wyżej w hierarchii następuje przeładowanie strony. Podczas edycji dużego fragmentu menu jest to bardzo frustrujące i czasochłonne.

- System nie wspiera automatycznych kalkulacji cen.

Dużym problemem jest potrzeba kalkulacji cen na podstawie wielu atrybutów. Cennik towaru klienta jest zależny od nakładu, czyli ilości produktu jaką chce zakupić nabywca. Dla każdego nakładu następnie trzeba naliczyć marżę oraz doliczyć odpowiednie wartości za transport oraz obróbkę.

Obecne rozwiązanie korzysta z arkuszy *Excel*. Ceny które wygeneruje arkusz dla każdego nakładu trzeba następnie przepisać do aktualnego systemu. Odbywa się to w następujący sposób: klient wypisuje wartości każdego nakładu pośredniku do odpowiedniego pola w panelu edycji produktu (np. 100;200;300), a następnie robi to samo dla cen wpisując je do innego dedykowanego pola (np. 5.00;4.00;3.00).

Czasami nieuwaznie wpisana wartość powoduje błąd w kalkulacji, co z kolei powoduje, że w interfejsie dla użytkownika publicznego wyświetlana jest wartość *Nan* (próba konwersji wartości nieliczbowej na liczbę - ang. *Not a Number*). Spowodowane jest to błędą implementacją i brakiem informacji dla użytkownika o błędnie wprowadzonych danych. Co więcej klient twierdzi, że nie wiedział skąd brały się niepoprawne wartości, w związku z czym próbował przepisywać je od nowa.

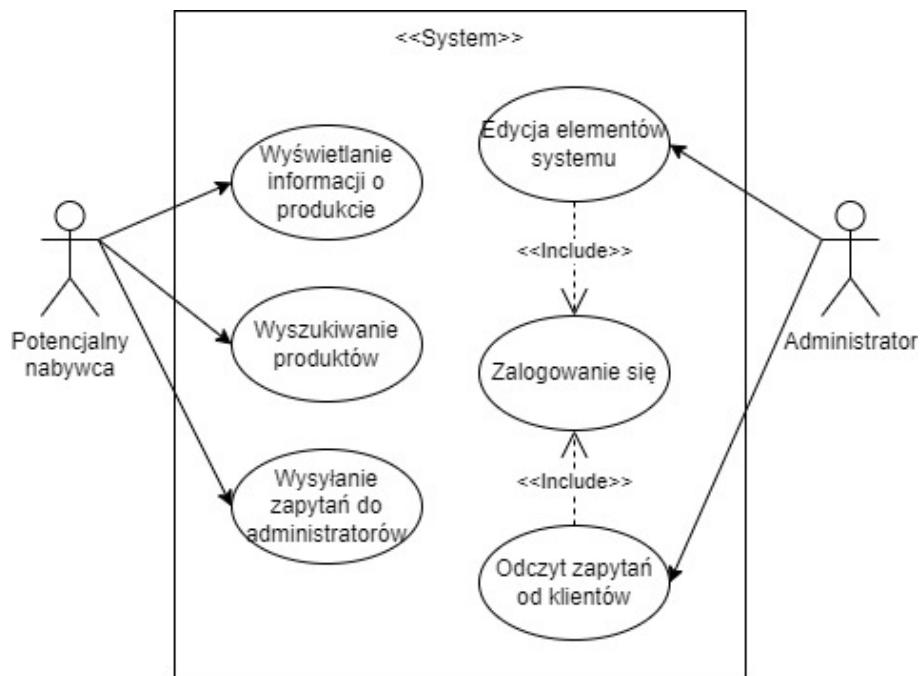
- System był projektowany jako uniwersalny dla wielu podobnych przedsiębiorstw.

Wydawałoby się, że nie jest to wada jako taka, natomiast jako, że system był tworzony dla

wielu przedsiębiorstw jednocześnie okazał się kłopotliwy dla każdego z osobna. Duża część funkcjonalności jest niepotrzebna co sprawia, że system jest niewygodny w użyciu. Zbyt dużo przycisków i pól sprawia, że trzeba dużo czasu spędzić na samym szukaniu odpowiedniego elementu do edycji. Co więcej, funkcjonalności których brakowało w systemie dla określonego przedsiębiorstwa były w krótkim czasie dodawane do systemu. Spowodowało to ich niską jakość i niewygodne użycie.

2.1.1. Przypadki użycia

Klient określił jakie podstawowe przypadki użycia powinien wspierać system (rys. 2.1). Administrator powinien móc edytować elementy systemu oraz odczytywać zapytania klientów, ale dopiero po zalogowaniu. Potencjalny nabywca natomiast powinien móc wyszukiwać produkty, wyświetlać informacje o nich, a także wysyłać zapytania na ich temat do administratorów systemu.



Rys. 2.1: Diagram przypadków użycia (opracowanie własne)

2.2. WYMAGANIA FUNKCJONALNE

- Administrator loguje się do panelu administracyjnego przy użyciu swojej nazwy użytkownika i hasła.
- Administrator dodaje, wyświetla, edytuje oraz usuwa:
 - produkty
 - kategorie
 - elementy menu

- podstrony publicznej strony internetowej
- wydzielone fragmenty publicznej strony internetowej
- zapytania od użytkowników publicznych
- Administrator korzysta z automatycznych kalkulatorów aby zmieniać cenniki w produktach.
- Administrator ukrywa wybrane produkty, kategorie, elementy menu i strony przed potencjalnym nabywcą produktów.
- System wspiera wgrywanie i edycję plików.
- System umożliwia szybkie transformacje zdjęć, a w szczególności zmianę ich formatu i rozmiaru.
- Potencjalny nabywca wyświetla publiczną stronę internetową, a w szczególności:
 - produkty
 - kategorie
 - elementy menu
- Potencjalny nabywca składa zapytania do produktów.
- System uniemożliwia potencjalnemu nabywcy wyświetlanie określonych wcześniej własności wybranych elementów - na przykład ukrytych cenników.
- System informuje administratorów o bieżących zmianach w elementach które aktualnie mają wpływ na to co edytuje administrator.

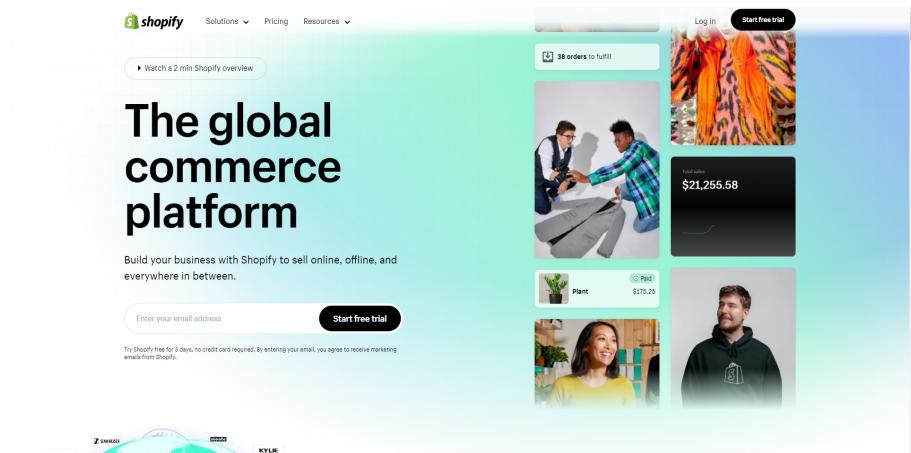
2.3. WYMAGANIA NIEFUNKCJONALNE

- Interfejs panelu administracyjnego wyświetla się poprawnie w przeglądarkach opartych na *Chromium* [3].
- Interfejs panelu administracyjnego ma stosunkowo neutralne kolory, aby nie rozpraszać administratora przy pracy.
- Interfejs panelu administracyjnego nie wymaga instrukcji, z wyjątkiem panelu do edycji kalkulatorów, który wyjaśnia w jaki sposób obliczane są wartości w cennikach.
- Interfejs publicznej strony internetowej zawiera akcent kolorystyczny klienta, to jest kolor czerwony o kodzie *hex*: "#bf0417".
- Dostępność systemu wynosi 99% czasu w roku.
- System jest zabezpieczony, a panel administracyjny wymaga hasła.
- Panel administracyjny obsługuje kilku użytkowników jednocześnie.
- Interfejs obecnego systemu ma długie czasy wczytywania kolejnych podstron. Ten problem zostanie zaadresowany w nowym systemie.

2.4. PRZEGŁĄD DOSTĘPNYCH ROZWIĄZAŃ

Na rynku jest dostępnych wiele rozwiązań e-commerce. Potencjalne wybory stanowią między innymi *Shopify*, *WooCommerce* oraz *MedusaJS*.

2.4.1. *Shopify*



Rys. 2.2: Strona internetowa platformy *Shopify* [14]

Zalety:

- Łatwa w obsłudze.
- Dostępność obsługi klienta jest zadowalająca.
- Gotowy hosting.

Wady:

- Cena jest dość wysoka.
- Mała rozszerzalność platformy.
- Brak opcji eksportu danych w razie zmiany serwisu.

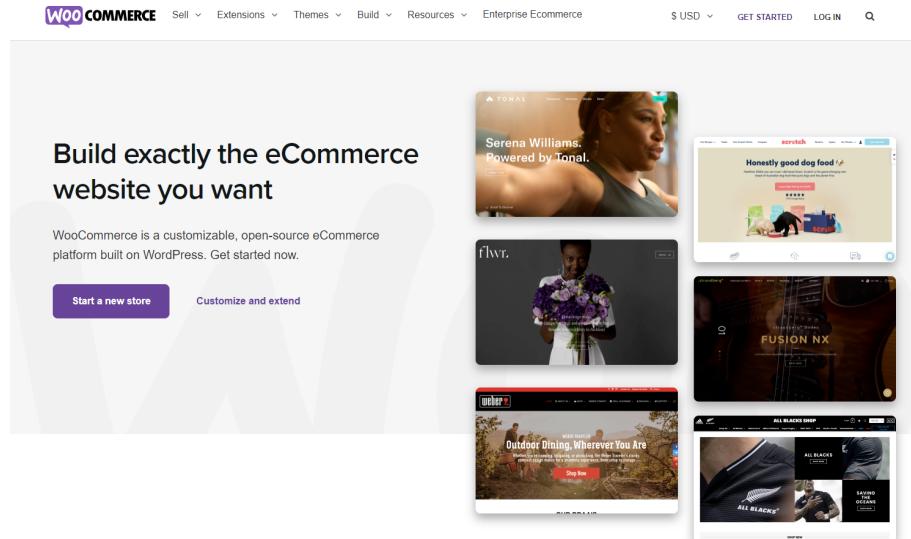
2.4.2. *WooCommerce*

Zalety:

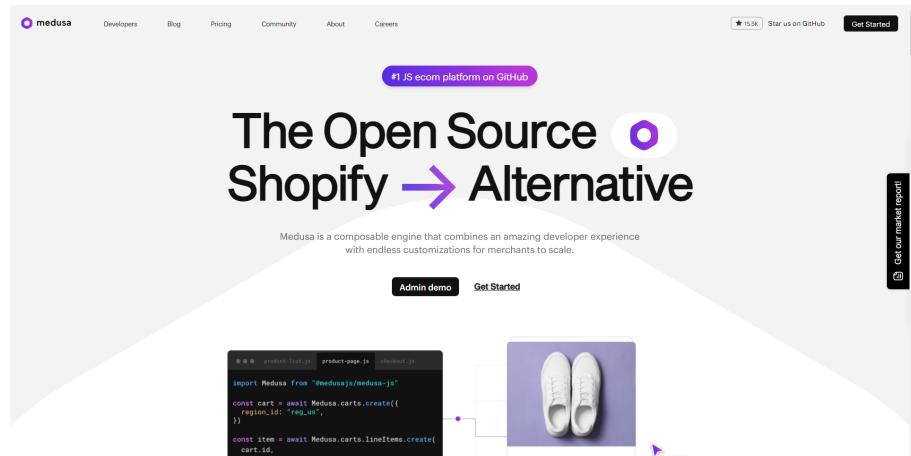
- Duża rozszerzalność platformy.
- Duża, pomocna społeczność.
- Profesjonalny wygląd.

Wady:

- Jest zależna od platformy *Wordpress*.
- Darmowa, ale rozszerzenia wymagają płatności.



Rys. 2.3: Strona internetowa platformy *WooCommerce* [20]



Rys. 2.4: Strona internetowa platformy *MedusaJS* [11]

2.4.3. MedusaJS

Zalety:

- Platforma o otwartym kodzie źródłowym.
- Duża rozszerzalność.

Wady:

- Należy samemu zarządzać serwerem, z którego korzysta serwis.
- Stworzona jako darmowa alternatywa dla Shopify, które jest bardziej od niej rozwinięte.

Potrzeba automatycznych aktualizacji cenników na podstawie określonych wcześniej kalkulatorów sprawia, że wymagane było by wiele zmian w powyższych systemach, które

należały wprowadzić do ich kodu źródłowego. Jako, że tego typu duże platformy otrzymują bardzo częste aktualizacje, szybko mogłyby się okazać, że z każdą taką aktualizacją wymagana była by zmiana wprowadzonych modyfikacji. Tego typu rozwiązanie było by nieopłacalne w dłuższej perspektywie. Dodatkowo klient nie potrzebuje rozwiązań typu e-commerce, ze względu na złożoność tego typu platform.

W związku z tymi rozważeniami została podjęta decyzja o opracowaniu platformy od podstaw, dedykowanej specjalnie do potrzeb klienta.

2.5. WYBÓR NARZĘDZI

Naturalnym kolejnym krokiem po podjęciu decyzji o opracowaniu serwisu od podstaw, było znalezienie rozwiązania, które ułatwi pracę z bazą danych. Do tego celu najlepiej nada się narzędzie typu *BaaS*. Budżet był ograniczony, w związku z czym zwróciło się do rozwiązań o otwartym kodzie źródłowym oraz umożliwiających użycie komercyjne bez dodatkowych opłat. Wobec tego porównano dwie platformy, które wydawały się najbardziej obiecujące: *Directus* [6] oraz *Strapi* [15]. Ta selekcja została podjęta na bazie popularności tych rozwiązań wśród społeczności. Jest to ważna metryka, z tego względu, że jeśli rozwiązanie jest popularne, będzie stale rozwijane, co oznacza aktualizacje bezpieczeństwa.

Po bliższym przyjrzeniu się obu rozwiązaniom Strapi wydawało się zwycięzcą pod względem popularności oraz łatwości użycia. Okazało się jednak, że nie umożliwia granularnej edycji permisji własności elementów, co wyklucza to rozwiązanie ze względu na sprzeczność z wymaganiami funkcjonalnymi.

W związku z tym, ostatecznie wybraną platformą *BaaS* był *Directus*.

Baza danych z której będzie korzystać Directus to *SQLite3*. Została ona wybrana z kilku powodów:

- *Directus* wspiera jedynie bazy danych SQL.
- *SQLite3* jest bardzo szybką bazą danych jeśli chodzi o operacje odczytu i zapisu.
- Ta baza danych funkcjonuje jako zwykły plik o formacie ".db" zamiast całego dodatkowego serwera, dzięki czemu nie zwiększy złożoności systemu.
- Skalowalność horyzontalna systemu nie jest istotna, ze względu na niski ruch na serwisie. W związku z tym *SQLite3* poradzi sobie ze spodziewanym obciążeniem bez żadnych problemów.

Frameworkiem wybranym do stworzenia frontendu był *Svelte* [16]. Ten wybór został podjęty ze względu na preferencje Autora. Każdy framework frontendowy spełnia to samo zadanie. Każdy z nich jednak podchodzi do niego w nieco inny sposób. *Svelte* jest stosunkowo nowy na rynku, ale szybko zyskuje popularność. Zdaniem Autora oferuje znacznie lepsze doświadczenie dla programisty niż na przykład bardzo popularny *React*.

Bardzo dużo frameworków frontendowych ma też dedykowane frameworki do aplikacji. Jest pewna subtelna różnica między frameworkiem do interfejsów, a frameworkiem do aplikacji. Framework do interfejsów umożliwia na przykład między innymi samo tworzenie komponentów, które będą funkcjonować w interfejsie użytkownika. Framework do aplikacji natomiast umożliwia na przykład *routing* (zgodnie z wiedzą Autora nie istnieje odpowiednik tego słowa w języku polskim), czyli określenie jakie komponenty mają być wyświetlane w zależności od ścieżki wybranej w przeglądarce. Na przykład "/strona-główna" wyświetli stronę główną, a "/kontakt" wyświetli formularz kontaktowy.

Dedykowanym frameworkiem do aplikacji dla frameworka do interfejsów *Svelte* jest *SvelteKit* [17]. To na nim zostanie zbudowana część frontendowa systemu.

Orócz tego wyróżkowane będą wiele bibliotek, które umożliwiają, bądź ułatwiają implementację systemu. Najważniejsze z nich to:

- *directus/sdk* - *SDK*, które ułatwia korzystanie z platformy *Directus*.
- *polka* - mikroserwer, który będzie odpowiedzialny za utrzymanie serwera do informowania użytkowników o zmianach wprowadzonych przez innych użytkowników, oraz w przyszłości do cyklicznej aktualizacji cenników oraz stanów magazynowych.
- *socket.io* - biblioteka *socket.io* ułatwia korzystanie z *WebSockets*.
- *diff* - ułatwia porównywanie dwóch obiektów *JSON*.
- *klona* - umożliwia głębokie kopiowanie obiektów.
- *marked* - konwertuje *markdown* do *HTML*.
- *nanoid* - umożliwia generację losowych ciągów znaków bezpiecznych kryptograficznie.
- *prettier* - automatycznie formatuje kod, dzięki czemu nie trzeba zastanawiać się nad konwencją.
- *slugify* - przetwarza串 na znaków na taki, który jest bezpieczny do umieszczenia w adresie przeglądarki.

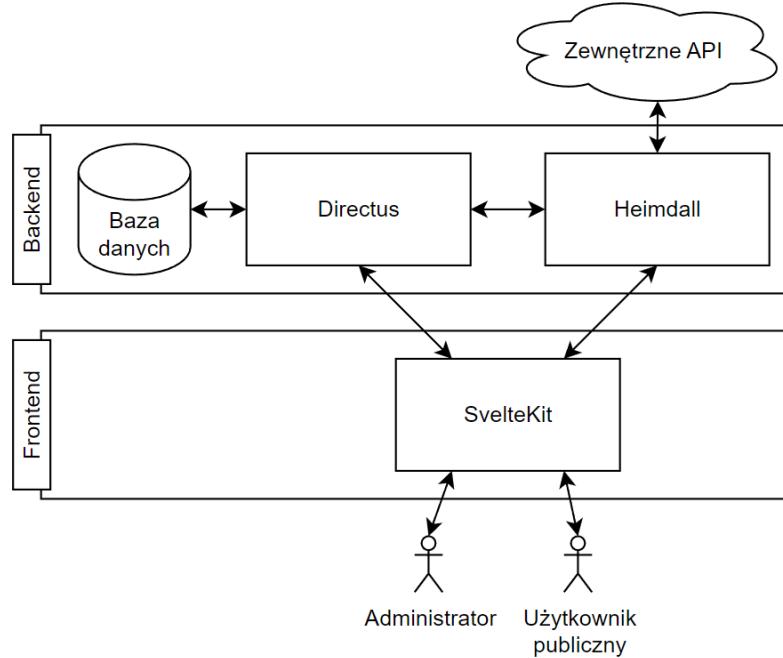
2.6. ARCHITEKTURA SYSTEMU

W systemie naturalnie można wydzielić dwa główne podsystemy, czyli frontend i backend. Oprócz tego można jeszcze zaznaczyć obecność zewnętrznych *API*, oraz samych użytkowników systemu.

Podsystem backendu będzie dość złożony, można w nim wydzielić 3 elementy: bazę danych, serwer *Directus* oraz serwer *Heimdall*.

Podsystem frontendu będzie znacznie prostszy i zasadniczo będzie posiadał jeden element, czyli serwer *SvelteKit*, który będzie dostarczać pliki budujące interfejs użytkownika.

Graficzna reprezentacja zależności między elementami systemu została przedstawiona na rysunku 2.5.



Rys. 2.5: Graficzna reprezentacja zależności między elementami systemu (opracowanie własne)

2.6.1. Baza danych

Bazą danych będzie *SQLite3*. Jako, że klient nie przewiduje bardzo dużych obciążen stron, skalowalność nie będzie tu problemem. Stąd *SQLite3* poradzi sobie z ruchem na stronie internetowej bez żadnego problemu.

2.6.2. Directus

Z bazy danych będzie korzystać *Directus*, który nałoży na nią *API*, z którego następnie będzie korzystać frontend oraz *Heimdall*. *Directus* będzie jedynym elementem systemu przez, który będzie odbywać się edycja bazy danych.

2.6.3. Heimdall

Heimdall to nazwa, którą Autor nadał serwerowi, który będzie funkcjonować na backen- dzie. Serwer będzie korzystać z technologii *WebSockets* do informowania użytkowników o zmianach w bieżących elementach wprowadzanych przez innych użytkowników. Oprócz tego jest przyszłościowo przewidziany jako narzędzie cyklicznej aktualizacji cenników oraz stanów magazynowych.

Autorska nazwa została zapożyczona z mitologii nordyckiej [24] od imienia jednego z bogów o tym samym imieniu. Bóg Hajmdal (ang. "Heimdall"), według mitów był czujnym obserwatorem i obrońcą Asgardu, domu bogów. Nazwanie jego imieniem serwera, który pełni dość podobne funkcje dla systemu wydawało się właściwe. Potrzeba aby serwer ten miał swoją nazwę własną w systemie wzięła się głównie z ograniczeń platformy *Directus*.

Aby za pomocą *Directusa* można było dokonać jakichkolwiek zmian w bazie danych trzeba być zalogowanym użytkownikiem - zmiany nie mogą być bezosobowe. W związku z tym kiedy serwer będzie aktualizować elementy automatycznie potrzebuje autora, stąd potrzeba nazwy własnej.

2.6.4. SvelteKit

Wszystkie potrzeby klienta wynikają z tego, że większość biznesów w dzisiejszych czasach potrzebuje własnej strony internetowej. Aby taka mogła działać potrzebne są do tego dwa elementy, czyli panel administracyjny (w formie aplikacji webowej bądź innej) oraz strona internetowa skierowana do użytkownika publicznego.

Oba te elementy (panel administracyjny i strona dla użytkownika publicznego) mogą być ze sobą spójne graficznie, choć nie jest to najistotniejsze.

2.6.4.1. Panel administracyjny

Panel administracyjny jest elementem gdzie administratorzy powinni móc wprowadzać do systemu produkty oraz inne elementy, a także opcjonalnie posiadać możliwość choćby fragmentarycznej konfiguracji serwisu. Powinien być spójny i łatwy w użyciu. Kolory użyte w jego projekcie graficznym powinny być neutralne, aby nie rozpraszać administratora podczas pracy.

Należy tutaj zaznaczyć jakie elementy powinien móc wprowadzać do systemu administrator. Główne z nich to produkty, kolory, kategorie, kalkulatory, elementy menu, strony, fragmenty, zapytania oraz pliki. Jako, że system jest tworzony specjalnie pod klienta, nazwy te zostały uzgodnione wraz z administratorami.

Większość elementów powinna mieć możliwość ukrycia ich przed użytkownikiem publicznym. W tym celu stworzono pole odpowiadające za widoczność elementu. Będzie się ono powtarzać przy wielu elementach, nie tylko tych głównych.

2.6.4.2. Publiczna strona internetowa

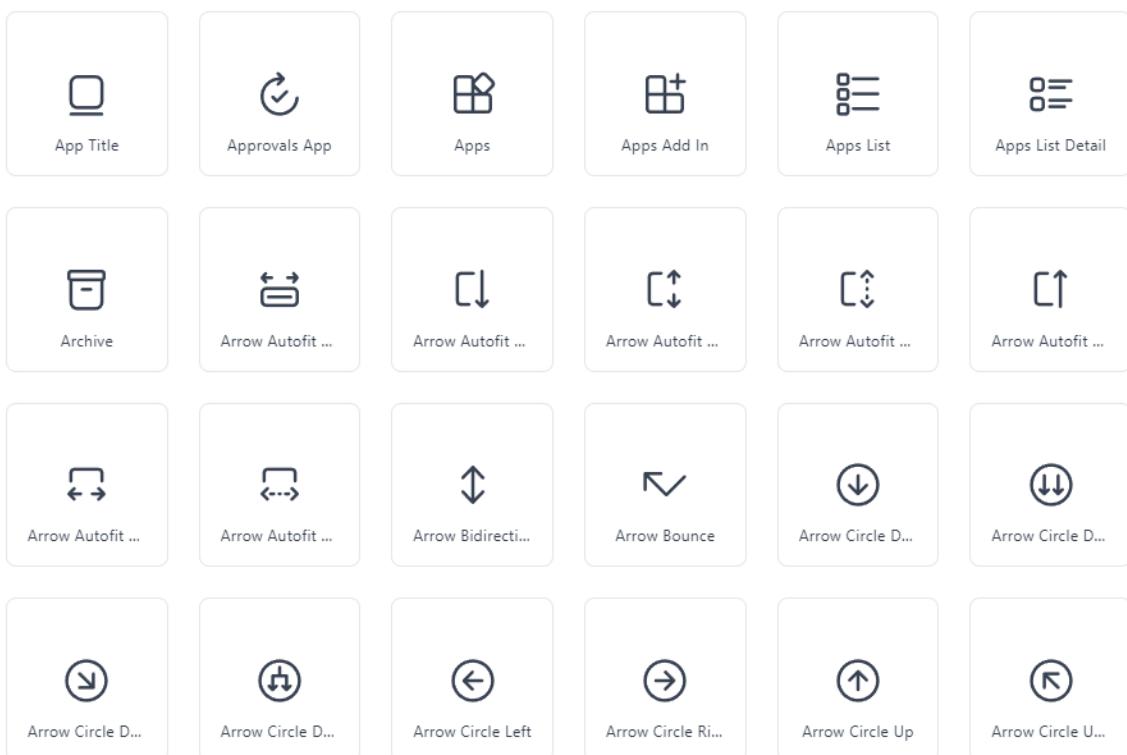
Strona internetowa natomiast powinna służyć głównie jako element gdzie użytkownik publiczny może wyświetlać określone informacje, które do systemu wprowadził administrator, a także elementy wcześniej ustalone z klientem, które są niezmienne.

Oprócz tego musi zawierać formularz przy każdym produkcie, za pomocą, którego można skontaktować się z administratorami i w efekcie nabyć towar.

2.7. PROJEKT GRAFICZNY

2.7.1. Ikony

Przy projekcie użyto ikon *Fluenticons* [9] o otwartym kodzie źródłowym, stworzonych przez firmę Microsoft. Ikony są zaokrąglone i bez wypełnienia. Wypróbowano wiele zestawów ikon, natomiast ikony *Fluenticons* wydawały się wyglądać najbardziej profesjonalnie, przy zachowaniu zaokrąglonych konturów. Co więcej, jako że administratorzy korzystają z systemu *Windows* ikony będą dla nich znajome dzięki czemu system wyda się bardziej komfortowy.



Rys. 2.6: Przykładowe ikony z zestawu *Fluenticons* [9].
Ikony mają zaokrąglone krawędzie oraz brak wypełnienia

2.7.2. Panel administracyjny

2.7.2.1. Kolory

Wedle wymagań klienta zdecydowano się na neutralną paletę barw. Wybrano kolor niebieski, który kojarzony jest z profesjonalizmem oraz spokojem. Dodatkowo jako akcent wybrano kolor szary. Następnie stworzone zostało kilka wariantów tych kolorów przy użyciu *Material Color Tool* [4].

2.7.2.2. Układ

Po lewej umieszczono pionowy pasek nawigacyjny odróżniający się kolorystycznie od reszty panelu. U samej góry paska znajduje się przycisk prowadzący do strony internetowej użytkownika publicznego. Jako kolejne rozmieszczone zostały odnośniki do kolejnych widoków panelu administracyjnego. Od góry są to: panel aktywności, produkty, kolory, kategorie, kalkulacje, menu, strony, fragmenty, zapytania, biblioteka plików, API. Na dole paska nawigacyjnego umieszczono przycisk powiadomień, który w przyszłości będzie informować użytkowników o automatycznych zmianach pobieranych z *API* zewnętrznych dostawców. Następnie umieszczono przycisk wylogowania użytkownika oraz jego awatar. Po najechaniu na awatar wyświetla się też imię i nazwisko aktualnie zalogowanego użytkownika.

Na górze witryny umieszczono belkę, na której umieszczono nazwę widoku, w którym aktualnie znajduje się użytkownik oraz ikonę mu odpowiadającą.

2.7.3. Publiczna strona internetowa

2.7.3.1. Kolory

Na publicznej stronie internetowej korzystano z wielu kolorów, natomiast głównie wykorzystywany jest motyw przewodniego koloru czerwonego o kodzie *hex* "#bf0417", który został dostarczony przez klienta i jest powiązany z jego logo. Wykorzystywane są również jaśniejsze oraz ciemniejsze warianty tego koloru.

2.7.3.2. Układ

Aktualny układ publicznej strony internetowej to tylko tymczasowa propozycja, póki klient nie zaakceptuje określonego projektu strony.

Do góry strony umieszczone jest logo, pole wyszukiwania produktów oraz przyciski menu (pierwszego zagnieźdzenia). Po najechaniu na każdy z przycisków wyświetla się większe menu z kolejnymi zagnieźdzeniami. Niżej umieszczono przyciski które mają służyć jako szybkie filtrowanie produktów według kategorii. Po wybraniu kategorii można wybrać kolejną, według hierarchii, i tak schodzić od ogółu do szczegółu.

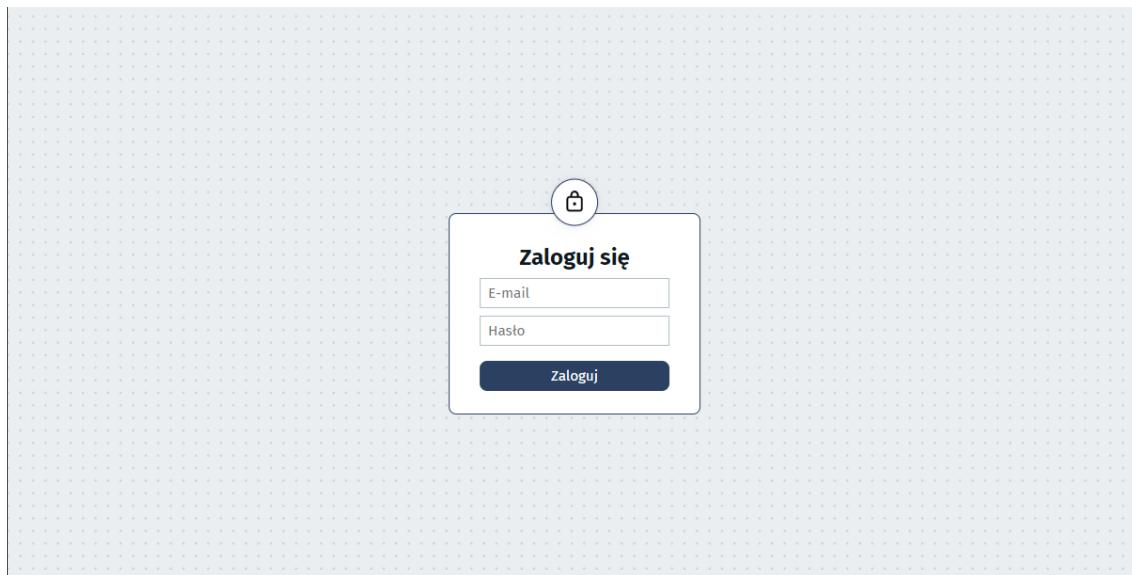
Następnie wyświetlana jest lista produktów jako kafelki. Kafelek wyświetla główne zdjęcie z galerii oraz nazwę, kod, cenę (opcjonalnie może też być oznaczone jako zawierająca w sobie także cenę znakowania) oraz kolory.

3. IMPLEMENTACJA

3.1. WIDOKI W PANELU ADMINISTRACYJNYM

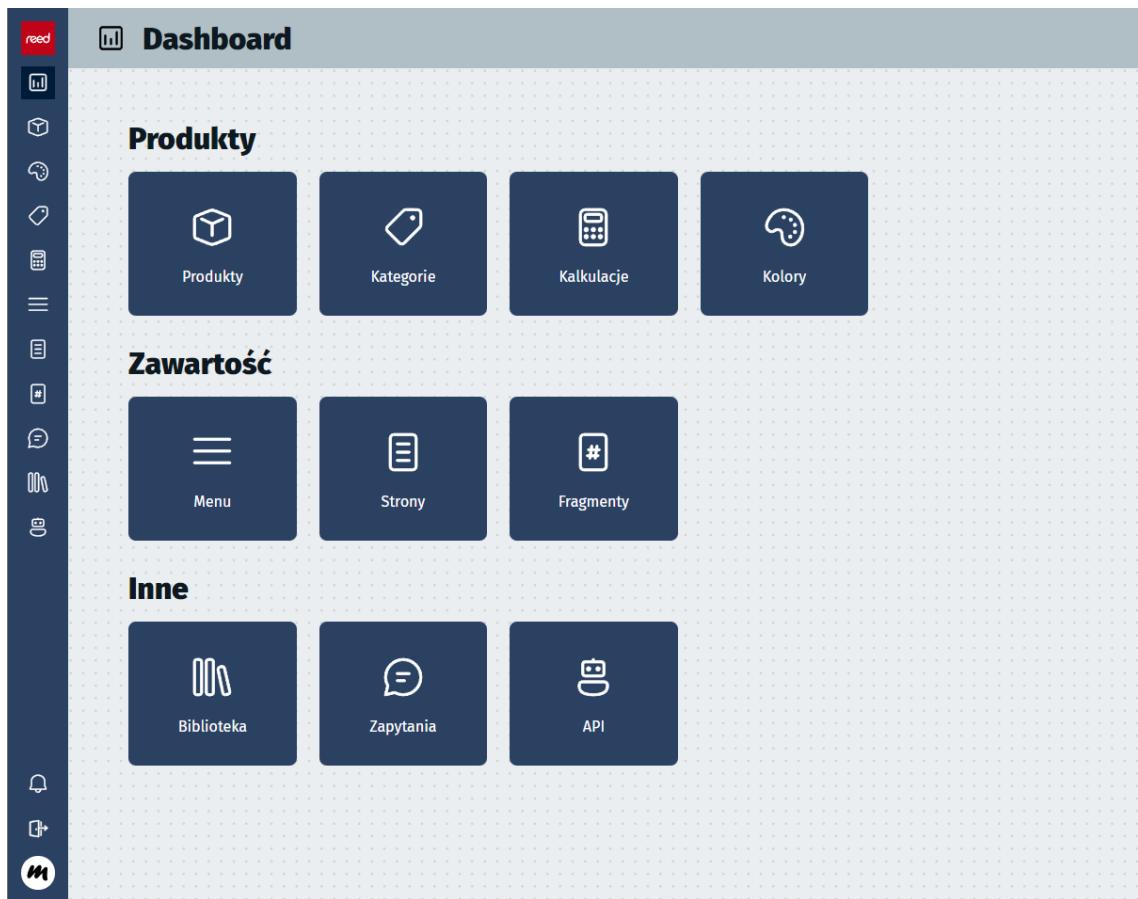
Panel administracyjny posiada wiele widoków, poprzez które użytkownik może wprowadzać zmiany w systemie. Aby go uruchomić, należy dopisać na końcu adresu *URL* "/admin". Zalogowany użytkownik może też użyć przycisków dostępnych w lewym dolnym rogu publicznej strony internetowej.

3.1.1. Panel logowania



Rys. 3.1: Widok panelu logowania (opracowanie własne)

To pierwszy panel, który widzi niezalogowany użytkownik systemu. Do pól email i hasło należy wpisać odpowiednie parametry. Po kliknięciu przycisku "Zaloguj" system spróbuje dokonać uwierzytelnienia użytkownika. System poinformuje użytkownika o błędnych danych w przypadku pomyłki. Po zalogowaniu sesja użytkownika będzie aktywna przez tydzień. Sesja jest przetrzymywana w *cookies*. Każda kolejna akcja podjęta w systemie korzysta z pliku *cookie* do autoryzacji oraz odświeża sesję. Po wylogowaniu plik *cookie* jest usuwany, a użytkownik przekierowany z powrotem do panelu logowania.



Rys. 3.2: Widok panelu aktywności (opracowanie własne)

3.1.2. Panel aktywności

Aktualnie w panelu aktywności (ang. *Dashboard*) zostały umieszczone odnośniki prowadzące do kolejnych widoków. W przyszłości najpewniej pojawią się tu wizualizacje danych odnoszące się na przykład do liczby produktów czy częstotliwości zapytań od użytkowników publicznych.

3.1.3. Kalkulacje

Zanim opisany zostanie widok samych produktów warto najpierw przedstawić widok kalkulacji, gdzie odbywa się edycja kalkulatorów. Produkty są bowiem od nich bezpośrednio zależne. W menu widok kalkulacji został jednak umieszczony niżej od produktów, jako że korzystać z niego będzie znacznie rzadziej.

Widok kalkulacji (rys. 3.3) zawiera kilka elementów. Po lewej umieszczony został panel boczny, gdzie edytować można odgórne marże produktów oraz "widoki cenowe". Odgórne marże to marże które można ustalić jako odgórnie nakładane na produkt lub produkt oraz znakowanie razem (całość). Można je też nadpisywać w samym widoku produktu. "Widoki cenowe" natomiast, to listy nakładów, które mają być obliczone i wyświetcone w cennikach określonego produktu. W panelu produktu można wybrać z którego "widoku cenowego"

The screenshot shows a software interface for calculating margins and product tables. On the left, there is a vertical toolbar with various icons. The main area is divided into sections:

- Ogólne marże**: A table for setting general margins. It has two columns: "Produkt" (Product) and "Całość" (Whole). Each column has two rows: "Marża" (Margin) and "Minimum" (Minimum). The first row uses percentages (%), and the second row uses amounts in złoty (zł).
- Widoki**: A section for viewing and managing product configurations. It contains four boxes:
 - Domyślny**: Shows values 100;200;300;500;1000 with dimensions 100 x 200 x 300 x 500 x 1000 x.
 - Skrócony**: Shows values 500;1000 with dimensions 500 x 1000 x.
 - Standard 50**: Shows values 50;100;200;300;500;1000 with dimensions 50 x 100 x 200 x 300 x 500 x 1000 x.
 - 25; 50; 100; 200; 300**: Shows values 25;50;100;200;300 with dimensions 25 x 50 x 100 x 200 x 300 x.
- Asgard**, **Macma**, and **MidOcean**: Tables for calculating margins for different products. Each table has columns for Nazwa (Name), Kod (Code), Typ (Type), and various margin values (M, MIN, P, T, T). Buttons for "Dodaj" (Add) are located at the bottom of each table.

Rys. 3.3: Widok panelu kalkulacji (opracowanie własne).

Poufne wartości zostało ocenzurowane aby chronić interesy przedsiębiorstwa.

Po lewej panel boczny gdzie można edytować odgórne marże oraz "widoki".

Po prawej tabele kalkulatorów

ma korzystać. Oprócz tego w panelu bocznym umieszczono przycisk, który wyświetla administratorowi przystępną instrukcję tego w jaki sposób ustalana jest cena. Instrukcja przeszła kilka interakcji, obecna jest w pełni zrozumiała dla administratorów (rysunki 3.4 oraz 3.5). Po prawej natomiast umieszczone są edytowalne tabele kalkulatorów. Tabele są podzielone na producentów danego towaru, aby w przyszłości ułatwić automatyczną aktualizację cen na podstawie ich API.

Cena produktu jest zawsze zależna od wybranego nakładu. Sprzedaż odbywa się bowiem głównie hurtowo. W związku z tym każdy nakład (czasami nazywany ilością) ma swój odpowiednik cenowy. Cena ta wyliczana jest na podstawie wcześniej określonych przedziałów cenowych.

Aktualnie administratorzy wykorzystują do kalkulacji cen arkusze *Excel*. Jest ich bardzo wiele i są dość złożone. Co więcej, każdy nowy zewnętrzny producent to potrzeba stworzenia kolejnych. Arkusze korzystają też z języka *Basic* do kalkulacji, ale także jako formy przetrzymywania danych poprzez wykorzystanie łańcuchów wyrażeń warunkowych.

$$\text{PRODUKT} = \text{MAX } A * \text{MARZA}; A + \text{MIN}$$

$$\text{ZNAKOWANIE} = \text{MAX } B * \text{MARZA}; B + \text{MIN}$$

$$\text{CAŁOŚĆ} = \text{MAX } C * \text{MARZA}; C + \text{MIN}$$

TRANSPORT = JEŚLI A > PRÓG TO 0 INACZEJ CENA

CENA DLA NAKŁADU = (CAŁOŚĆ + TRANSPORT) / NAKŁAD

Kalkulacje	
<input checked="" type="checkbox"/> Włączone	<input type="button" value="Wyczyść"/>
REED RO 2 Laser RO	
Ilość	Cena
25	68,59
300	63,28
<input type="checkbox"/> Odgórna marża na znakowanie (0%, min 0 zł)	
Marża %	Minimum zł
30	40
Pole znakowania mm	
30	x 50

Odgórne marże	
Produkt	Całość
Marża	Marża
Minimum	Minimum
40 %	60 zł
50 %	70 zł
Cena	
25,87	
<input checked="" type="checkbox"/> Odgórna marża na produkt (40%, min 60 zł)	
<input type="checkbox"/> Odgórna marża na całość (50%, min 70 zł)	
Marża %	Minimum zł
30	40

Odgórna marża na znakowanie		M	MIN	P	T	TP	Ryczałt	20	50	100
40	50	50	20	500	90	0,75				
		50	20	500	125	1,03				
		50	20	500	65	1,04	0,88	0,75		

Jeśli próg jest pusty
Jeśli cena jest ustawiona
transport będzie
zawsze uwzględniany.
na 0 lub pusta wykorzystywana
jest wcześniejsza wartość.

Rys. 3.4: Jedna ze starszych wersji instrukcji dla administratorów w jaki sposób ustalana jest cena (opracowanie własne). Forma instrukcji była nieprzystępna, w związku z czym została zmieniona. Instrukcja może być nieczytelna w pracy, natomiast w panelu administratora została dostosowana do wielkości ekranu

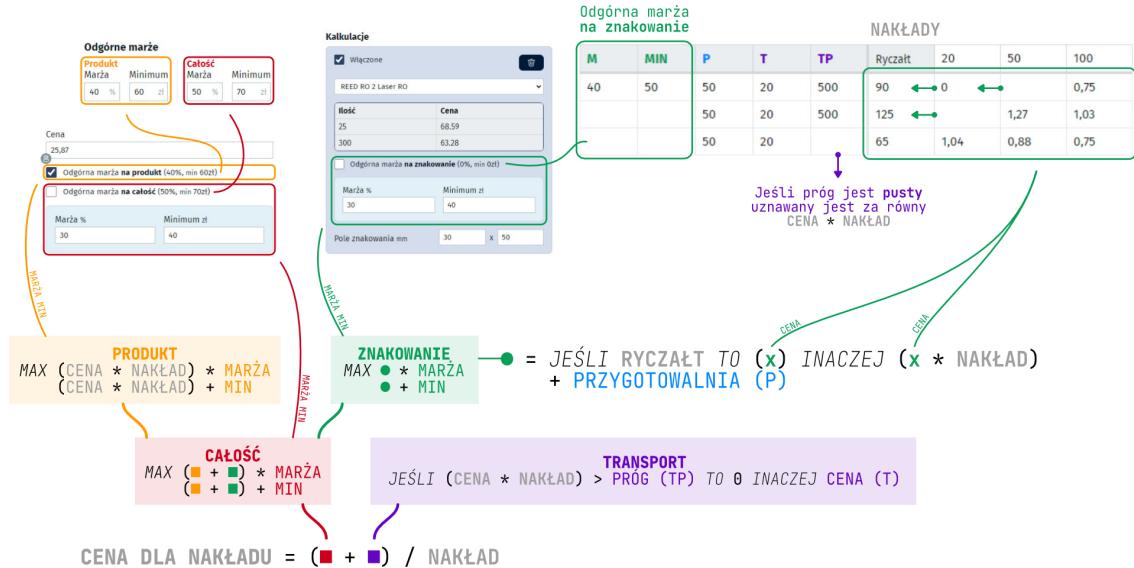
Rozwiązanie jest dość nieeleganckie i utrudnia aktualizację cenników w razie ich zmiany u producenta.

Nowe, przyjęte po konsultacjach z klientem rozwiązanie problemu, przyjmie formę tabelaryczną (rys. 3.1). Komórki tabel będą przyjmować określone wartości, uogólnione w wystarczającym stopniu, tak, aby umożliwiły implementację cenników wewnętrznych, ale także cenników każdego zewnętrznego producenta. Dzięki temu wykorzystana zostanie pewna znajomość administratorów z programem Excel, a jednocześnie ułatwi to edycję wiele cenników jednocześnie. Umożliwi też całosciowe spojrzenie porównawcze na wszystkie cenniki. Jedna z opracowanych tabeli kalkulatorów została przedstawiona na rysunku 3.6.

Nazwa	Kod	Typ	Marża	Min	...	1	100	...
Bez znakowania	-	-	0%	0 zł	...	0 zł	0 zł	...
Laser	R1	1 kolor	30%	40 zł	...	70 zł	0.7 zł	...
Tampodruk	RT	2 kolory	0%	0 zł	...	50 zł	0.5 zł	...

Tabela 3.1: Propozycja tabelarycznej formy wprowadzania wartości dla kalkulatora cennika (opracowanie własne)

Wartości, na podstawie których wyliczana jest cena jednostkowa dla określonego nakładu, to nakład, cena produktu, marża dla produktu, minimum dla produktu, cena znakowania pobrana z tabeli, marża dla znakowania, minimum dla znakowania, przygotowalnia, transport, próg do uwzględnienia transportu, marża dla całości (produkt oraz znakowanie) oraz minimum dla całości.



Rys. 3.5: Ostateczna wersja instrukcji dla administratorów w jaki sposób ustalana jest cena (opracowanie własne). Zrozumienie instrukcji zostało sprawdzone bezpośrednio na administratorach. Forma instrukcji jest bardziej obrazowa, linie prowadzą po kolejie według komponentów aż do ostatecznej ceny dla określonego nakładu. Instrukcja może być nieczytelna w pracy, natomiast w panelu administratora została dostosowana do wielkości ekranu

Cena dla każdego nakładu ustalana jest za pomocą funkcji *formula*, która korzysta z funkcji *fraction*, *round*, *sanitize* oraz *matchLabelingPriceRange*:

```

1 function fraction(percent) {
2     // Zamień procent na ulamek.
3     // np. 25% -> 1.25; 0% -> 1
4     return percent / 100 + 1;
5 }
6
7 function round(num) {
8     // Zaokrąglaj do dwóch miejsc po przecinku.
9     return Math.round((num + Number.EPSILON) * 100) / 100;
10}
11
12 function sanitize(data) {
13     // Rekurencyjnie zamień wszystkie wartości
14     // "falsy" na 0 w zagnieżdzonym obiekcie.
15     // np. { a: null, b: { c: null }, d: [{ e: null }], f: [] }
16     //      { a: 0,   b: { c: 0   }, d: [{ e: 0   }], f: [] }
17     if (typeof data == 'object' && data != null) {
18         if (Array.isArray(data)) {
19             for (let [i, val] of data.entries()) {
20                 data[i] = sanitize(val);
21             }
22         } else {
23             for (let [key, val] of Object.entries(data)) {

```

```

24         data[key] = sanitize(val);
25     }
26   }
27 } else data ||= 0;
28 return data;
29 }
30
31 function matchLabelingPriceRange(amount, pricesPerAmount) {
32   // Dobierz odpowiednia cene z tabeli dla nakladu.
33   pricesPerAmount.sort((a, b) => a.amount - b.amount);
34   let matchingRange = null;
35   for (let range of pricesPerAmount) {
36     if (range.amount <= amount) {
37       // puste ceny sa ignorowane
38       // ceny ustawione na 0 sa uznawane za pomylki
39       if (range.price) matchingRange = range;
40     } else {
41       break;
42     }
43   }
44   if (matchingRange === null) {
45     return {
46       price: 0,
47       isLumpsum: false
48     };
49   }
50   return {
51     price: matchingRange.price,
52     isLumpsum: matchingRange.amount == 1
53   };
54 }
55
56 function formula(amount, product, labeling, full,
57                  prepress, transport, transportThreshold) {
58   // Naklad
59   // amount: number
60   amount = sanitize(amount);
61
62   // Produkt - cena, marza, minimum
63   // product: { price: number, margin: number, minimum: number }
64   product = sanitize(product);
65
66   // Znakowanie - cena, marza, minimum
67   // labeling: { prices: [number], margin: number, minimum: number }
68   labeling = sanitize(labeling);
69
70   // Calosc - cena, marza, minimum

```

```

71 // full: { margin: number, minimum: number }
72 full = sanitize(full);
73
74 // Przygotowalnia
75 // prepress: number
76 prepress = sanitize(prepress);
77
78 // Transport
79 // transport: number
80 transport = sanitize(transport);
81
82 // Prog transportu
83 // transportThreshold: number
84 // -- nie sanityzuj, bo 0 oznacza darmowy transport
85
86 // Oblicz cene produktu uwzgledniajac marze.
87 const productPrice = amount * product.price;
88 const productPriceWithMargin = Math.max(
89   productPrice * fraction(product.margin),
90   productPrice + product.minimum
91 );
92
93 // Oblicz cene znakowania uwzgledniajac marze.
94 const range = matchLabelingPriceRange(amount, labeling.prices);
95 const labelingPrice =
96   (range.isLumpsum ? range.price : amount * range.price)
97   + prepress;
98 const labelingPriceWithMargin = Math.max(
99   labelingPrice * fraction(labeling.margin),
100  labelingPrice + labeling.minimum
101 );
102
103 // Oblicz cene produktu oraz znakowania uwzgledniajac marze.
104 const price = productPriceWithMargin + labelingPriceWithMargin;
105 const singlePrice =
106   Math.max(price * fraction(full.margin), price + full.minimum);
107
108 // Dolicz transport jesli prog zostal przekroczony.
109 const transportPrice =
110   transportThreshold != null && productPrice > transportThreshold
111   ? 0 : transport;
112 const fullPrice = singlePrice + transportPrice;
113
114 return round(fullPrice / amount); // cena jednostkowa
115 }
```

		Nazwa	Kod	Typ	M	MIN	P	T	TP	Ryczałt	20	50	100	250
☆	↑	↓	▼	Tampodruk	T1									
☆	↑	↓	▼	Tampodruk	T2									
☆	↑	↓	▼	Tamopodruk	T3									
☆	↑	↓	▼	Tampodruk	T4									
☆	↑	↓	▼	Termotransfer	TT1	1 kolor								
☆	↑	↓	▼	Termotransfer	TT2	1 kolor								
☆	↑	↓	▼	Termotransfer	TT3	1 kolor								
☆	↑	↓	▼	Druk cyfrowy	DC0									
☆	↑	↓	▼	Druk cyfrowy	DC1	do 10 ci								
☆	↑	↓	▼	Druk cyfrowy	DC2	do 30 c								
☆	↑		▼	Tłoczenie	HS									

[+ Dodaj](#)

Rys. 3.6: Widok tabeli kalkulatorów (opracowanie własne). Poufne wartości zostały ocen-zurowane aby chronić interesy przedsiębiorstwa

3.1.4. Produkty

Widok produktów (rys. 3.7) pozwala na dodawanie (również od razu w określonej kategorii), przeszukiwanie, paginację oraz edycję produktów. Produkty to najważniejszy element systemu. Na nim opiera się większość pozostałych. Korzysta z Kolorów, Kategorii, Kalkulacji oraz Plików. Co istotne, w obecnym systemie klient przy przeglądaniu Produktów mógł szybko filtrować je według hierarchicznie zorganizowanych kategorii. Ważne jest więc zachowanie obecnego *status quo* i przeniesienie tej funkcjonalności również do nowego systemu. Zostało to osiągnięte za pomocą panelu bocznego zlokalizowanego po lewej. Po wybraniu kategorii z listy wyświetlane są tylko produkty należące do tej kategorii.

Edycja produktu jest podzielona na sekcje.

Pierwsza sekcja (rys. 3.8) nie ma określonego tytułu, wypełnianie są w niej podstawowe informacje o produkcie. Najważniejsze jest oznaczenie widoczności produktu. Produkt musi mieć możliwość ukrycia go przed użytkownikiem publicznym. Oprócz tego ma możliwość oznaczenia go jako nowość. Następymi polami do edycji są nazwa i kod produktu. Kod powinien być unikalny dla każdego produktu, aby dało się go łatwo wyszukiwać. Następną sekcją jest sekcja *SEO* w której wypełnić można tytuł oraz skrócony opis produktu. Jeśli potencjalny nabywca wyszuka określonego długopisu w wyszukiwarce internetowej, te pola będą wyświetlać się w jego wynikach. Kolejnym atrybutem do uzupełnienia są kategorie. Każdy produkt może mieć nieograniczoną ilość kategorii, natomiast pierwsza w liście jest uznawana za główną. Można zmieniać ich kolejność oraz usuwać. Następnie określa się pola dotyczące *API* zewnętrznych dostawców, czyli przede wszystkim producenta towaru (które to pole informuje z którego *API* należy skorzystać) oraz kodu produktu u zewnętrznego producenta oraz *id* (ang. *identifier*, czyli "identyfikator") w jego bazie danych. Kod z reguły pokrywa się z kodem produktu w bazie danych klienta, natomiast *id* jest istotne, ponieważ dzięki niemu będzie można szybko pobrać dane z zewnętrznego *API*. Co więcej, pole to

The screenshot shows a software interface for managing products. On the left, there's a sidebar with various icons and a vertical list of categories under 'Kategorie'. The '1.3.2 Długopisy metalowe' category is currently selected. The main area displays a table of products with columns for ID, Kod, Nazwa, Utworzenie, and Aktualizacja. There are also buttons for adding new products and searching. At the bottom, there's a pagination element and a dropdown for selecting how many items to show per page.

			Dodaj	Dodaj w Długopisy metalowe	Szukaj... Ctrl+Q		
			ID	Kod	Nazwa	Utworzenie	Aktualizacja
✓	✗	✗	1	R-	Długopis metalow	Mikołaj Kaźmierczak 0	Dariusz Kaź
✓	✗	✗	42	R0011	Długopis metalow	Dariusz Kaźmierczak 1	Mikołaj Kaź
✓	✗	✗	49	RE0012	Długopis metalow	Dariusz Kaźmierczak 2	Mikołaj Kaź
✓	✗	✓	50	RE2137	Długopis znanej n	Mikołaj Kaźmierczak 25	Mikołaj Kaź
✓	✗	✗	51	RE0002	Długopis metalow	Dariusz Kaźmierczak 0	Dariusz Kaź
✓	✗	✗	55	C-	Długopis metalow	Dariusz Kaźmierczak 0	

← 1 →

Na stronie 50

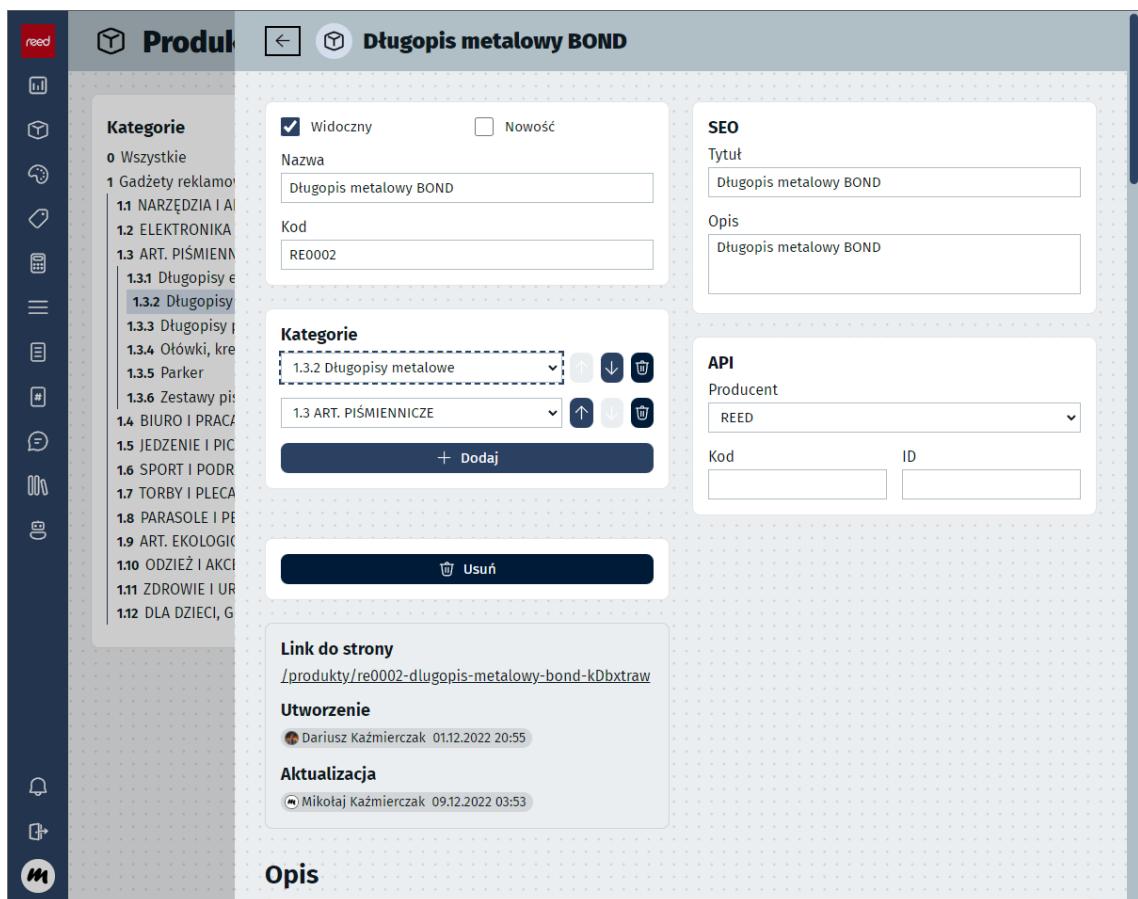
Rys. 3.7: Widok produktów (opracowanie własne). U góry widać małą belkę, na której znajdują się przyciski dodania nowego produktu (także przycisk dodania go od razu do wybranej aktualnie kategorii) oraz przeszukiwanie produktów (które można uruchomić skrótem klawiszowym "Ctrl+Q". Na dole znajduje się element paginacji produktów, można w nim wybrać ile produktów ma być pokazywanych jednocześnie

zostanie wypełnione automatycznie przy pierwszym pobraniu danych produktu z API. Pola do dotyczące API zostały umieszczone w interfejsie przyszłościowo, ponieważ ten moduł nie został jeszcze ukończony.

Następnie wyświetlane jest kilka informacji o produkcie, czyli między innymi bezpośredni odnośnik do odpowiedniej podstrony na publicznej stronie internetowej oraz informacje o autorze produktu (użytkowniku, który dodał go do bazy danych) oraz autorze ostatniej aktualizacji. Oba pola dotyczące użytkowników zmian mają też określony czas tych zmian.

Drugą sekcję (rys. 3.9) jest "Opis", który aktualnie jest prostym polem edycji tekstu wspierającym język *markdown*. Administratorzy znają jego podstawową składnię, choć ten komponent w przyszłości zostanie zastąpiony bardziej przystepnym edytorem. Po prawej od pola samej edycji znajduje się okno podglądu tekstu w formie w, której będzie wyświetlany na publicznej stronie internetowej. Jest aktualizowane na bieżąco.

Trzecią sekcją jest cennik produktu. Ceny różnią się w zależności od tego jak dużo



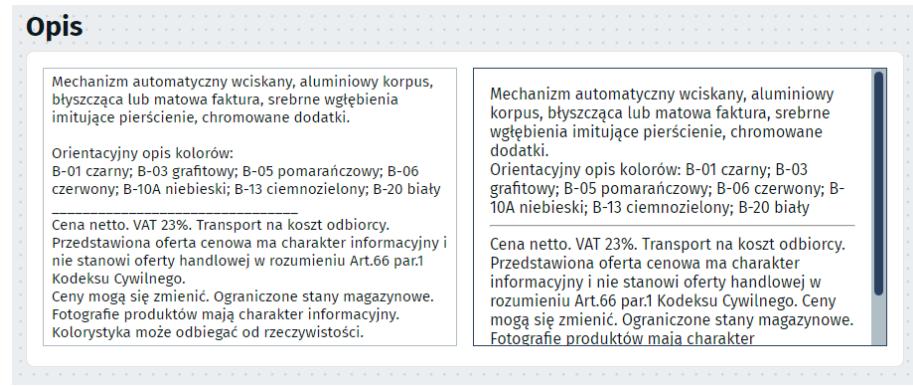
Rys. 3.8: Panel edycji pojedynczego produktu (opracowanie własne). Na górnej belce edytora znajduje się tytuł produktu oraz przycisk umożliwiający powrót do listy produktów. Jeśli użytkownik wprowadzi zmiany w produkcie, przycisk ten jest zastępowany przez przyciski zapisu oraz anulowania bieżących zmian

towaru będzie chciał kupić klient - sprzedaż jest głównie nastawiona na hurt. Zasadniczo istnieją dwie wersje sekcji do określania cennika: manualne wpisywanie cen dla określonych nakładów (rys. 3.10) lub wybranie z których kalkulatorów ma korzystać cennik (rys. 3.11).

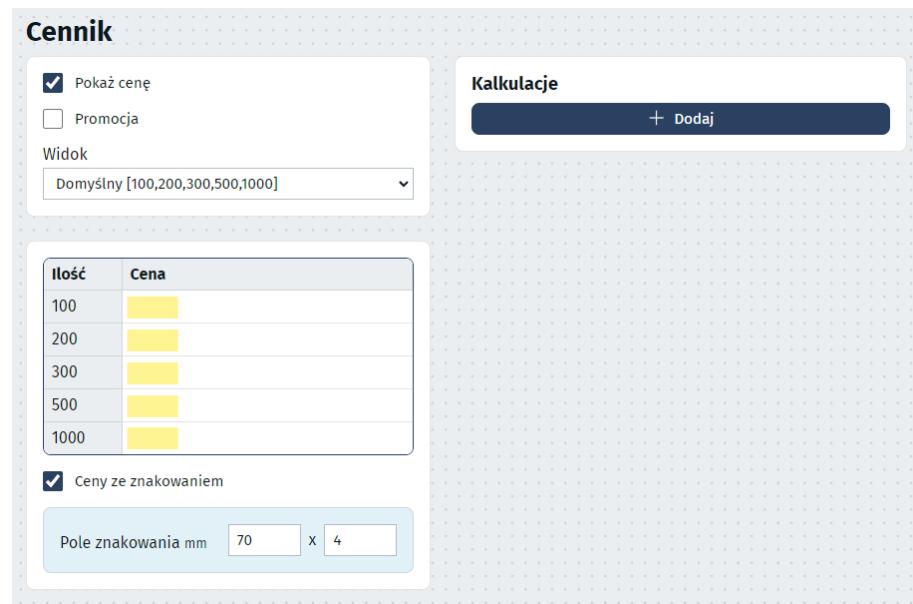
W obu wersjach określania cennika należy określić czy cena ma być w ogóle widoczna, czy cena jest na promocji (co spowoduje wyświetlenie dodatkowej kolumny do określenia cen promocyjnych) oraz "widok cennika" który określa jakie nakłady mają być wyświetlane w cenniku.

W wersji z manualnym wpisywaniem cen można też oznaczyć czy w ceny wliczone jest znakowanie (oraz jak duże jest jego pole, czyli jak duża może być grafika która zostanie oznaczona na produkcie). Ważne jest także aby użytkownik publiczny wiedział czy cena produktu ma w sobie wliczone znakowanie. Według klienta bardzo częstym problemem jest pewna dezorientacja jego klientów w tym zakresie. Istotne więc jest zwrócenie uwagi użytkownika publicznego na ten aspekt cennika.

W wersji gdzie administrator wybiera kalkulatory na podstawie których wyliczany



Rys. 3.9: Pole edycji opisu produktu (opracowanie własne)

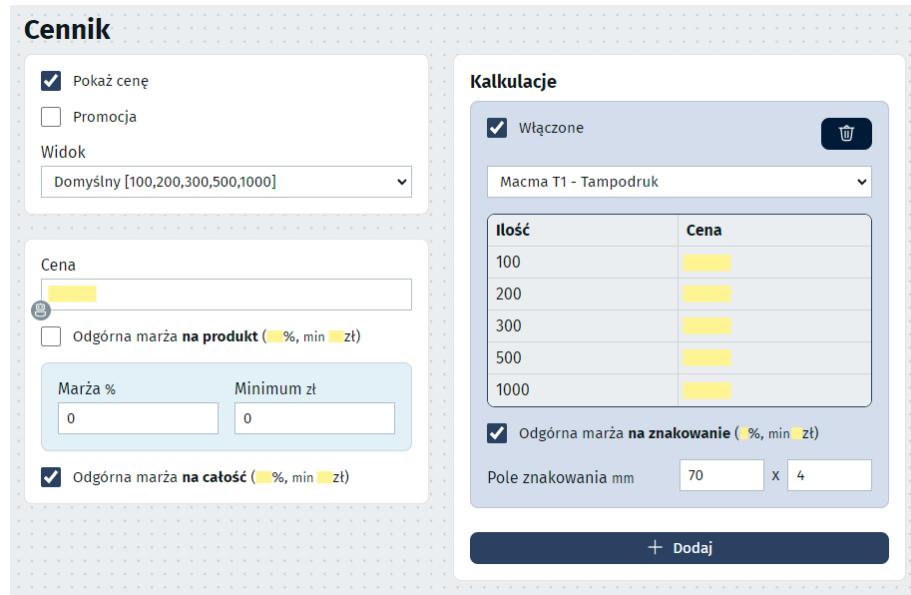


Rys. 3.10: Pola dostosowywania cennika produktu (opracowanie własne). Wersja z manualnym wypełnianiem cennika według wybranego widoku. Poufne wartości zostały ukryte aby chronić interesy klienta

jest cennik podawana jest tylko cena jednostkowa produktu. Oprócz tego można nadpisać odgórne marże dla produktu, całości oraz znakowania.

Galeria jest prostym komponentem (rys. 3.12), gdzie wybrać można zdjęcia, które zostaną wyświetlane obok produktu. Istnieje jedno zdjęcie główne, które wyświetlane jest jako pierwsze.

W systemie można określić warianty kolorystyczne produktu (rys. 3.13). Tutaj podawane są też ilości magazynowe każdego z wariantów oraz ich kod w API zewnętrznych dostawców. Kod ten doklejany jest do głównego kodu produktu, co tworzy całościowy kod - produktu razem z jego wariantem kolorystycznym. Na przykład: kod produktu to "RE0023", a kod koloru "-01". Razem określa to kod "RE0023-01" co daje dokładne określenie o jaki przedmiot chodzi. Takie rozgraniczenie jest spowodowane tym, że bardzo często zewnętrzni



Rys. 3.11: Pola dostosowywania cennika produktu (opracowanie własne). Wersja z okresem ceny jednostkowej i doborem kalkulatorów. Poufne wartości zostały ukryte aby chronić interesy klienta

dostawcy tworzą oddzielne produkty w swoich bazach dla każdego wariantu kolorystycznego.

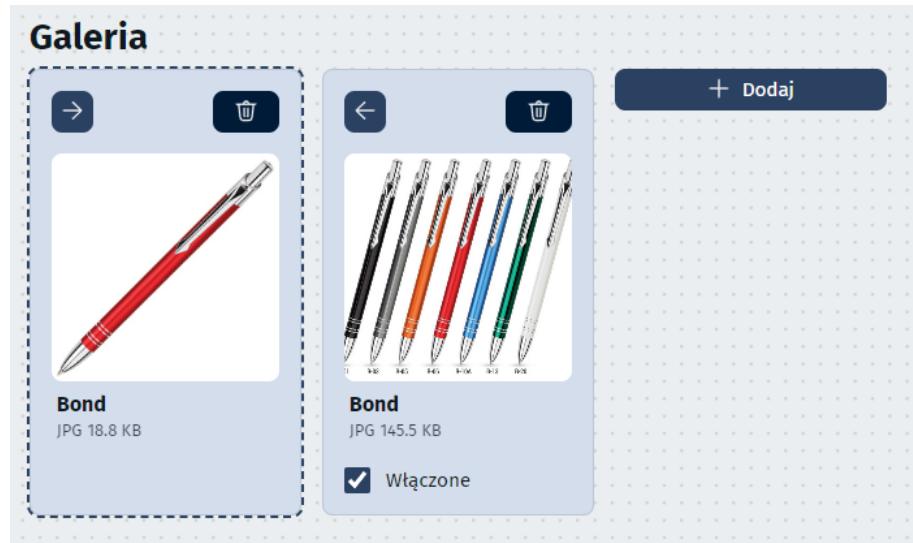
Oprócz stanu magazynowego i kodu należy też określić same kolory. Według klienta nie spotyka się wariantów kolorystycznych, które mają więcej niż dwa kolory. W związku z tym ustalono, że można wybrać maksymalnie dwa kolory. Jeśli zajdzie potrzeba wariantu, który ma więcej niż dwa kolory wykorzystana zostanie flaga "Wielokolorowe". Na publicznej stronie internetowej wariant wielokolorowy zostanie oznaczony odpowiednią ikoną.

Ostatnim elementem wariantu kolorystycznego są jego zdjęcia. Można to uznać za galerię dla określonego wariantu kolorystycznego. Przy każdym zdjęciu można oznaczyć także, że ma być wyświetlane również w głównej galerii, aby uniknąć potrzeby dodawania zdjęcia w obu miejscach.

3.1.5. Kolory

Widok kolorów (rys. 3.14) umożliwia dodawanie, przeszukiwanie oraz edycję kolorów wykorzystywanych w produktach. W tabeli przy kodzie produktu wyświetlany jest dodatkowo kolor w formie małego kwadracika. Ułatwia to przeglądanie listy.

Kolor (rys. 3.15) to prosty element, w którym zawiera się jego nazwa oraz kod koloru w formacie *hex*. Na przykład kolorowi "czerwony" odpowiadałby kod "#ff0000". Kolor można wybrać korzystając z wbudowanego w przeglądarkę narzędzia, które ułatwia ten proces.



Rys. 3.12: Edytor galerii produktu (opracowanie własne)

3.1.6. Kategorie

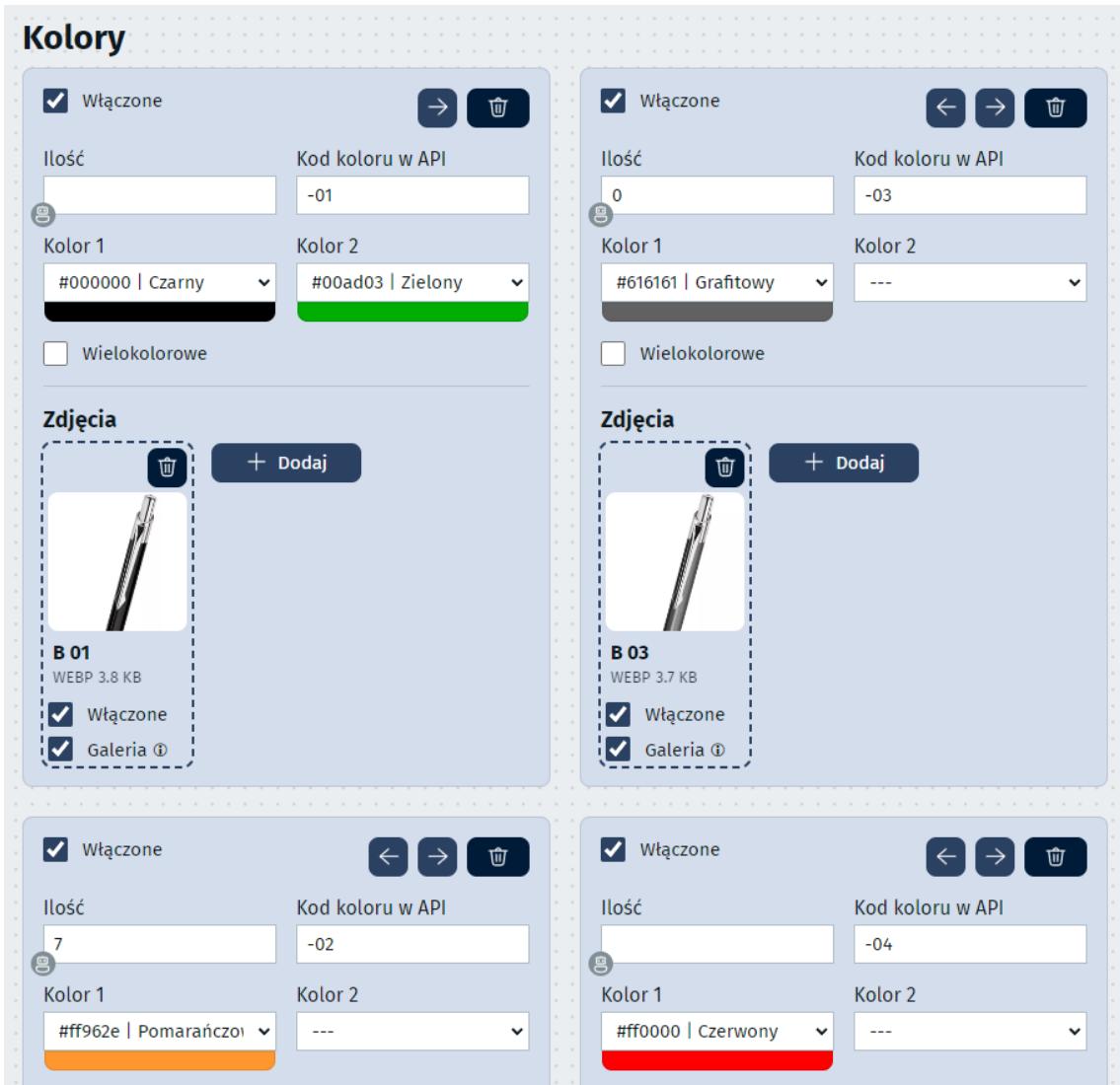
Widok kategorii (rys. 3.16) pozwala na dodawanie, przeszukiwanie oraz zmianę kategorii. Lista kategorii jest hierarchiczna i zagnieźdzona, to znaczy każda kategoria może mieć także podkategorie. Dla przykładu kategoria "Artykuły piśmiennicze" może zawierać pod sobą kategorie "Długopisy" oraz "Pióra". "Długopisy" może z kolei zawierać pod sobą "Długopisy metalowe" oraz "Długopisy plastikowe" i tak dalej. Głębokość możliwych zagnieździeń jest nieograniczona. W praktyce jednak klient nie zamierza tworzyć więcej niż cztery zagnieździenia.

Opracowano prosty sposób zmiany hierarchii oraz zagnieździenia poprzez przeciąganie elementów tabeli. Po kliknięciu i przytrzymaniu przycisku myszy nad specjalnym elementem, a następnie zmianą pozycji kurSORA przy elementach tabeli zaczyna pojawiać się pola gdzie można przenieść element (rys. 3.18). Pola podświetlają się kiedy mysz jest bezpośrednio nad nimi.

Pola wypełniane w samej kategorii (rys. 3.17) są podobne do wielu pól, które wypełniane są też w produkcie. Są to widoczność kategorii (która jeśli jest wyłączona jednocześnie uniemożliwia potencjalnemu nabywcy wyświetlanie na publicznej stronie internetowej kategorii zagnieźdzonych pod nią), nazwa, tytuł oraz opis *SEO*, a także wybór zdjęcia i opis (działający tak samo jak odpowiednik w produkcie).

3.1.7. Menu

Widok menu (rys. 3.19) pozwala na dodawanie, przeszukiwanie oraz zmianę elementów menu. Na publicznej stronie docelowo będą istniały dwa menu: główne menu na górnej belce oraz menu stopki. W widoku można wybrać które menu jest aktualnie edytowane za pomocą przycisków pod górną belką. Aktualnie wybrany ma ciemniejsze tło.



Rys. 3.13: Edycja wariantów kolorystycznych produktu (opracowanie własne)

Zasada działania oraz sposób zmiany hierarchii i zagnieździenia elementów jest taki sam jak w kategoriach.

Do edytowalnych (rys. 3.20) właściwości elementu należą widoczność, tytuł, oznaczenie elementu jako folder oraz link. Oprócz tego do kategorii można wybrać odpowiadające jej zdjęcie. Pole wpisywania linku jest specyficzny elementem. Można do niego wkleić link do produktu, kategorii oraz strony, a system wykryje o który element chodzi. Na przykład: wklejamy link "www.adresstrony.pl/produkty/slug-produktu" a system wykryje, że chodzi o określony produkt i przyporządkuje go do elementu menu. Dzięki temu w razie przyszłych aktualizacji tego produktu i zmianie jego linku element menu dalej będzie wskazywać na odpowiedni produkt. Warto też zaznaczyć, że jeśli wykrytym elementem jest kategoria, to na stronie internetowej zostaną również wyświetlane wszystkie zagnieżdżone pod nią kategorie. Jest to spowodowane faktem, że klient w dużej mierze opiera część menu na kategoriach i w efekcie musi przepisywać każdą kategorię do menu, co sprawiało, że musiał

	ID	Nazwa	Kolor	Utworzenie	Aktualizacja
✓	1	Czerwony	#ff0000	Mikołaj Kaźmierczak 04.03.2022 2	Mikołaj Kaźmierczak 29.11.2022 11
✓	2	Zielony	#00ad03	Mikołaj Kaźmierczak 04.03.2022 2	Dariusz Kaźmierczak 28.11.2022 2
✓	4	Niebieski	#2273BF	Mikołaj Kaźmierczak 04.03.2022 2	
✓	5	Pomarańczowy	#ff962e	Mikołaj Kaźmierczak 04.03.2022 2	Dariusz Kaźmierczak 29.11.2022 2
✓	6	Srebrny	#dedede	Mikołaj Kaźmierczak 04.03.2022 2	Dariusz Kaźmierczak 28.11.2022 2
✓	7	Czarny	#000000	Mikołaj Kaźmierczak 04.03.2022 2	Dariusz Kaźmierczak 28.11.2022 2
✓	10	Biały	#ffffff	Mikołaj Kaźmierczak 23.11.2022 17	
✓	11	Granatowy	#0300d1	Dariusz Kaźmierczak 28.11.2022 2	
✓	12	Błękitny	#47b9ff	Dariusz Kaźmierczak 28.11.2022 2	
✓	13	Brązowy	#853838	Dariusz Kaźmierczak 28.11.2022 2	
✓	14	Fioletowy	#7000fa	Dariusz Kaźmierczak 28.11.2022 2	
✓	15	Bordowy	#cc0000	Dariusz Kaźmierczak 28.11.2022 2	
✓	16	Żółty	#ffff00	Dariusz Kaźmierczak 28.11.2022 2	Dariusz Kaźmierczak 28.11.2022 2
✓	17	Zielony jasny	#36ec64	Dariusz Kaźmierczak 28.11.2022 2	Dariusz Kaźmierczak 28.11.2022 2
✓	18	Szary	#949494	Dariusz Kaźmierczak 28.11.2022 2	
✓	19	Grafitowy	#616161	Dariusz Kaźmierczak 28.11.2022 2	
✓	20	Oliwkowy	#719721	Dariusz Kaźmierczak 28.11.2022 2	
✓	21	Beżowy	#feeca9	Dariusz Kaźmierczak 28.11.2022 2	Dariusz Kaźmierczak 28.11.2022 2
✓	22	Zielony ciemny	#027000	Dariusz Kaźmierczak 28.11.2022 2	
✓	23	Przezroczysty	#fafafa	Dariusz Kaźmierczak 28.11.2022 2	
✓	24	Różowy	#ff75e1	Dariusz Kaźmierczak 28.11.2022 2	
✓	25	Turkusowy	#00d6c8	Dariusz Kaźmierczak 28.11.2022 2	
✓	26	Szampański	#e3d08c	Dariusz Kaźmierczak 28.11.2022 2	
✓	27	Złoty	#d1a400	Dariusz Kaźmierczak 28.11.2022 2	

Rys. 3.14: Widok listy kolorów (opracowanie własne)

powtarzać podobny proces dwukrotnie. Przy obecnym rozwiązaniu ten problem został rozwiązany. Warto zaznaczyć, że jeśli w linku nie zostanie wykryty produkt, kategoria, bądź strona link zostanie zapisany jako link do zewnętrznej strony.

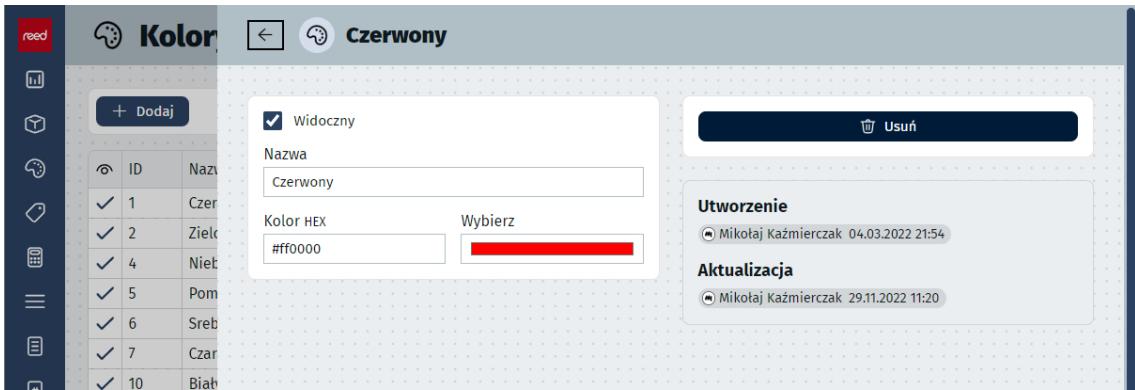
3.1.8. Strony

Widok stron (rys. 3.22) jest bardzo prosty. Pozwala na dodawanie, przeszukiwanie oraz paginację stron. Strony to wszelkie podstrony na publicznej stronie internetowej. Na przykład strona z obowiązkiem informacyjnym czy polityką prywatności serwisu.

Edycja strony (rys. 3.23) polega na wypełnieniu jej tytułu, tytułu i krótkiego opisu *SEO* oraz opisu funkcjonującego w ten sam sposób jak ten w produkcie. Opis w tym wypadku można uznać za treść strony.

3.1.9. Fragmenty

Widok fragmentów (rys. 3.24) pozwala na przeszukiwanie oraz paginację fragmentów. Fragmenty to wszelkie drobne fragmenty strony, które są w nią wbudowane z góry. Na przykład informacje na stronie "Kontakt", czy też krótki opis działalności firmy w stopce.



Rys. 3.15: Widok edycji pojedynczego koloru (opracowanie własne)

Edycja fragmentu (rys. 3.25) polega na wypełnieniu jego tytułu, opisu (treści fragmentu) oraz danych. Dane to specyficzny element – można w nim umieszczać dane w formacie *JSON*. Służą do umieszczania bardziej specyficznych informacji o fragmencie, na przykład listy zdjęć czy też linku do *Google Maps*. W ogólności można powiedzieć, że to lista wszystkich niestandardowych danych, które mogą się okazać przydatne w określonym fragmencie. Daje to dużą elastyczność i nie ogranicza projektu w przyszłości. Format danych jest też zrozumiały przez administratorów, głównie dlatego, że mieli w przeszłości doświadczenie z formatem *XML*.

3.1.10. Zapytania

Widok zapytań (rys. 3.26) pozwala na dodawanie, przeszukiwanie oraz paginację zapytań. Zapytania mają głównie za zadanie kontakt z potencjalnym nabywcą produktów poprzez formularz na stronie internetowej umieszczony przy każdym produkcie. Użytkownik wypełnia swoje dane osobowe oraz treść zapytania a następnie administrator może sprawdzić zapytanie w panelu administracyjnym. Uznano jednak, że zapytania mogą też służyć jako narzędzie wewnętrzne, to znaczy administrator sam może dodać zapytanie do systemu. Na przykład potencjalny nabywca mógł wykonać połączenie telefoniczne i w ten sposób złożyć zamówienie. Administrator może następnie wpisać to zamówienie do systemu w formie zapytania.

W przyszłości każde zapytanie powinno także mieć ustaloną procentową szansę na to że jest to spam. Formularz mógł być wypełniony przez robota, który wpisuje automatycznie wiadomości do pól formularza i naciska przycisk wysłania. Każde zapytanie ma też oznaczenie czy jest w nim dodany załącznik, a także czy formularz został wysłany ze strony produktu czy strony "Kontakt". Aktualnie na publicznej stronie został zaimplementowany jedynie formularz przy produkcie, natomiast sama funkcjonalność zapytań jest stworzona z myślą o przyszłości.

Edycja zapytania (rys. 3.27) polega na wypełnieniu imienia i nazwiska (opcjonalnie),

The screenshot shows a software application window titled "Kategorie". On the left is a vertical toolbar with icons for file operations like add, delete, edit, and search. The main area contains a table with the following data:

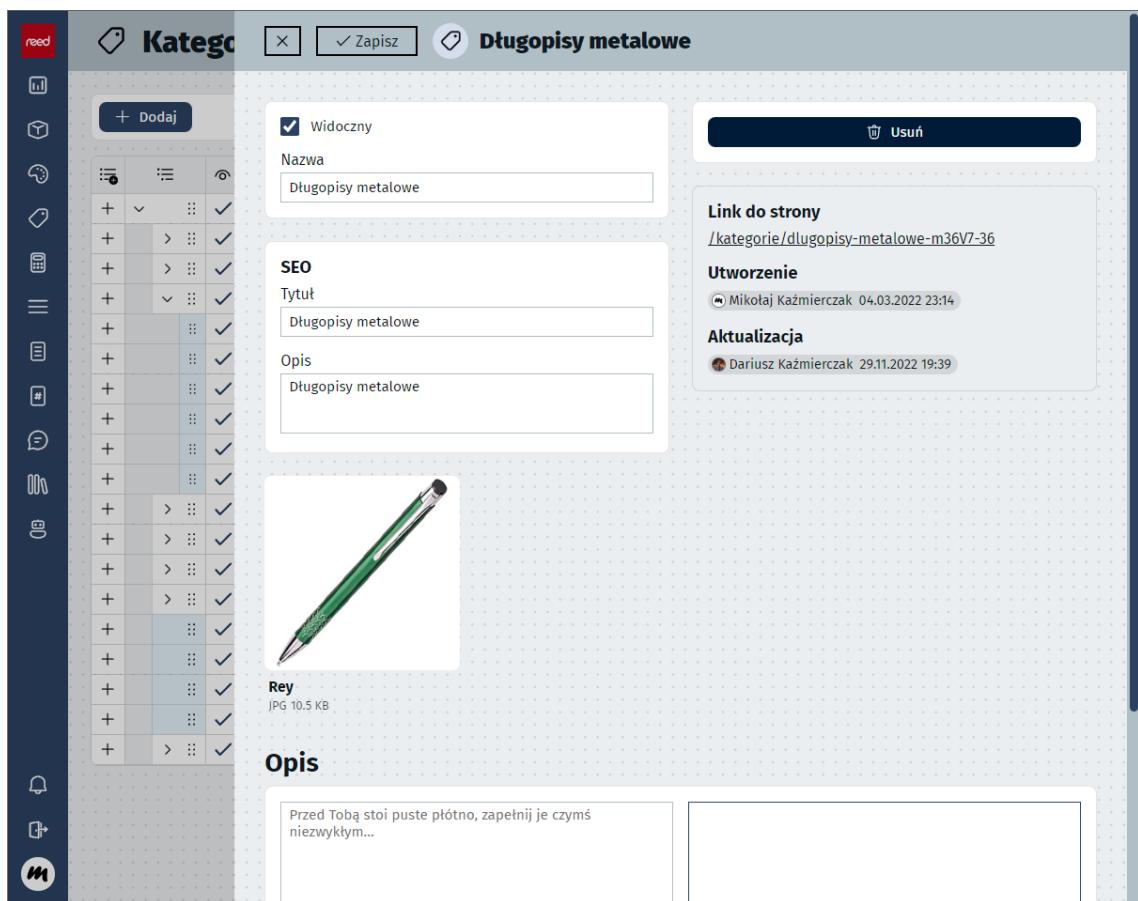
	ID	Nazwa	Utworzenie	Aktualizacja
+	✓ 88	1 Gadżety reklamowe	(Mikołaj Kaźmierczak 28.11.2022 21:20)	(Mikołaj Kaźmierczak 28.11.2022 23:56)
+	✓ 64	1.1 NARZĘDZIA I AKCESORIA	(Dariusz Kaźmierczak 28.11.2022 20:48)	(Dariusz Kaźmierczak 29.11.2022 19:46)
+	✓ 89	1.2 ELEKTRONIKA I AKCESORIA	(Dariusz Kaźmierczak 29.11.2022 19:29)	(Dariusz Kaźmierczak 29.11.2022 19:44)
+	✓ 1	1.3 ART. PIŚMIENNICE	(Mikołaj Kaźmierczak 04.03.2022 23:14)	(Dariusz Kaźmierczak 29.11.2022 19:37)
+	✓ 81	1.3.1 Długopisy ekologiczne	(Dariusz Kaźmierczak 28.11.2022 20:58)	(Dariusz Kaźmierczak 29.11.2022 19:39)
+	✓ 2	1.3.2 Długopisy metalowe	(Mikołaj Kaźmierczak 04.03.2022 23:14)	(Dariusz Kaźmierczak 29.11.2022 19:39)
+	✓ 3	1.3.3 Długopisy plastikowe	(Mikołaj Kaźmierczak 04.03.2022 23:18)	(Dariusz Kaźmierczak 29.11.2022 19:39)
+	✓ 82	1.3.4 Ołówki, kredki	(Dariusz Kaźmierczak 28.11.2022 20:59)	(Dariusz Kaźmierczak 29.11.2022 19:39)
+	✓ 83	1.3.5 Parker	(Dariusz Kaźmierczak 28.11.2022 21:00)	(Dariusz Kaźmierczak 29.11.2022 19:39)
+	✓ 32	1.3.6 Zestawy piśmiennicze	(Dariusz Kaźmierczak 12.11.2022 17:43)	(Dariusz Kaźmierczak 29.11.2022 19:39)
+	✓ 68	1.4 BIURA I PRACA	(Dariusz Kaźmierczak 28.11.2022 20:50)	(Dariusz Kaźmierczak 29.11.2022 19:37)
+	✓ 4	1.5 JEDZENIE I PICIE	(Mikołaj Kaźmierczak 05.03.2022 00:27)	(Dariusz Kaźmierczak 29.11.2022 19:48)
+	✓ 78	1.6 SPORT I PODRÓŻE	(Dariusz Kaźmierczak 28.11.2022 20:56)	(Dariusz Kaźmierczak 29.11.2022 20:35)
+	✓ 30	1.7 TORBY I PLECAKI	(Dariusz Kaźmierczak 23.10.2022 14:47)	(Dariusz Kaźmierczak 29.11.2022 20:34)
+	✓ 77	1.8 PARASOLE I PELERYNY	(Dariusz Kaźmierczak 28.11.2022 20:56)	(Dariusz Kaźmierczak 29.11.2022 19:48)
+	✓ 71	1.9 ART. EKOLOGICZNE	(Dariusz Kaźmierczak 28.11.2022 20:52)	(Dariusz Kaźmierczak 29.11.2022 19:48)
+	✓ 67	1.10 ODZIEŻ I AKCESORIA	(Dariusz Kaźmierczak 28.11.2022 20:50)	(Dariusz Kaźmierczak 29.11.2022 19:48)
+	✓ 80	1.11 ZDROWIE I URODA	(Dariusz Kaźmierczak 28.11.2022 20:57)	(Dariusz Kaźmierczak 29.11.2022 19:48)
+	✓ 92	1.12 DLA DZIECI, GRY	(Dariusz Kaźmierczak 29.11.2022 19:51)	=

Rys. 3.16: Widok listy kategorii (opracowanie własne). Warto zaznaczyć przyciski (druga kolumna tabeli), które umożliwiają przegląd zagnieżdżonych elementów a także zmianę hierarchii i poziomu zagnieżdżenia

email, telefonu (opcjonalnie) oraz treści zapytania. Do zapytania można także dodać załącznik.

3.1.11. Biblioteka plików

Widok biblioteki plików (rys. 3.28) pozwala na dodawanie, przeszukiwanie oraz pagination (rys. 3.29) plików. Pliki można dodawać poprzez przeciągnięcie pliku (lub wielu plików jednocześnie) w wyznaczone u góry strony pole, lub też poprzez kliknięcie na to pole, co otworzy eksplorator plików w OS komputera użytkownika. Biblioteka plików wspiera wszystkie formaty plików. Nie ma też ograniczeń jeśli chodzi o ich wielkość, uznano, że to administrator powinien podjąć decyzję o sensowności wgrywania dużego pliku. Biblioteka wspiera natomiast automatyczne szybkie transformacje zdjęć. Odbywają się one z użyciem *API Directus*. Na publicznej stronie internetowej wczytywane są już przetworzone zdjęcia. Na przykład użytkownik wgrał duże zdjęcie o rozdzielczości 10000x10000 pikseli, które waży 20mb. Na stronie internetowej wczytane zostanie zdjęcie o rozdzielczości 200x200 pikseli (dodatkowo o zmienionym rozszerzeniu na *webp*, które jest znacznie bardziej wydajnym pojemnościowo formatem graficznym niż na przykład *jpeg*) i rozmiarze 15kb.



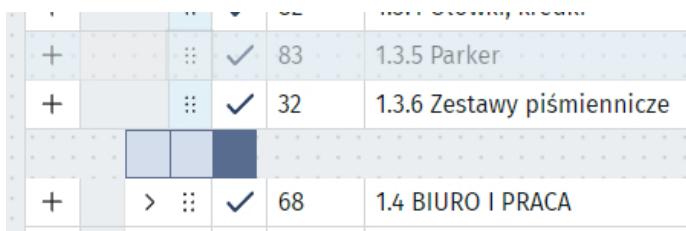
Rys. 3.17: Widok edycji pojedynczej kategorii (opracowanie własne). Warto zaznaczyć, że na górnjej belce wyświetlane są przyciski anulowania lub zapisania zmian

Dzięki temu potencjalny nabywca przeglądający produkty nie będzie zmuszony wczytywać grafik o nieadekwatnych rozmiarach, ale wciąż będzie miał możliwość pobrania zdjęcia w pełnej rozdzielczości.

Edycja pliku (rys. 3.27) polega jedynie na możliwości podmiany pliku na inny bądź usunięciu go. W panelu edycji można też sprawdzić kilka dodatkowych informacji o pliku, czyli jego formacie, nazwie przy pobraniu (nazwie jaką miał plik podczas wgrywania) oraz jego szerokości i wysokości. Tutaj wyświetlane jest też zdjęcie o pełnej (oryginalnej) rozdzielczości.

3.2. STRONA INTERNETOWA DLA UŻYTKOWNIKA PUBLICZNEGO

Publiczna strona internetowa dla potencjalnych nabywców produktów (rys. 3.31) została zaprojektowana jako rozwiązanie tymczasowe. Nie posiada jeszcze wszystkich funkcjonalności. Powodem jest fakt, że projekt graficzny całej witryny nie został jeszcze ustalony z klientem. Mimo tego, rozwiązanie aktualne posiada najważniejsze funkcjonalności, aby zaprezentować ich działanie.



Rys. 3.18: Interfejs podczas przeciągania elementu hierarchii w nowe miejsce. Sam element zmienia przezroczystość aby pokazać że zostanie zmieniony jego stan. Pola które oznaczają gdzie element heirarchi zostanie umieszczony po puszczeniu przycisku myszy są umieszczone poniżej i intuicyjnie oznaczają gdzie znajdzie się element (opracowanie własne)

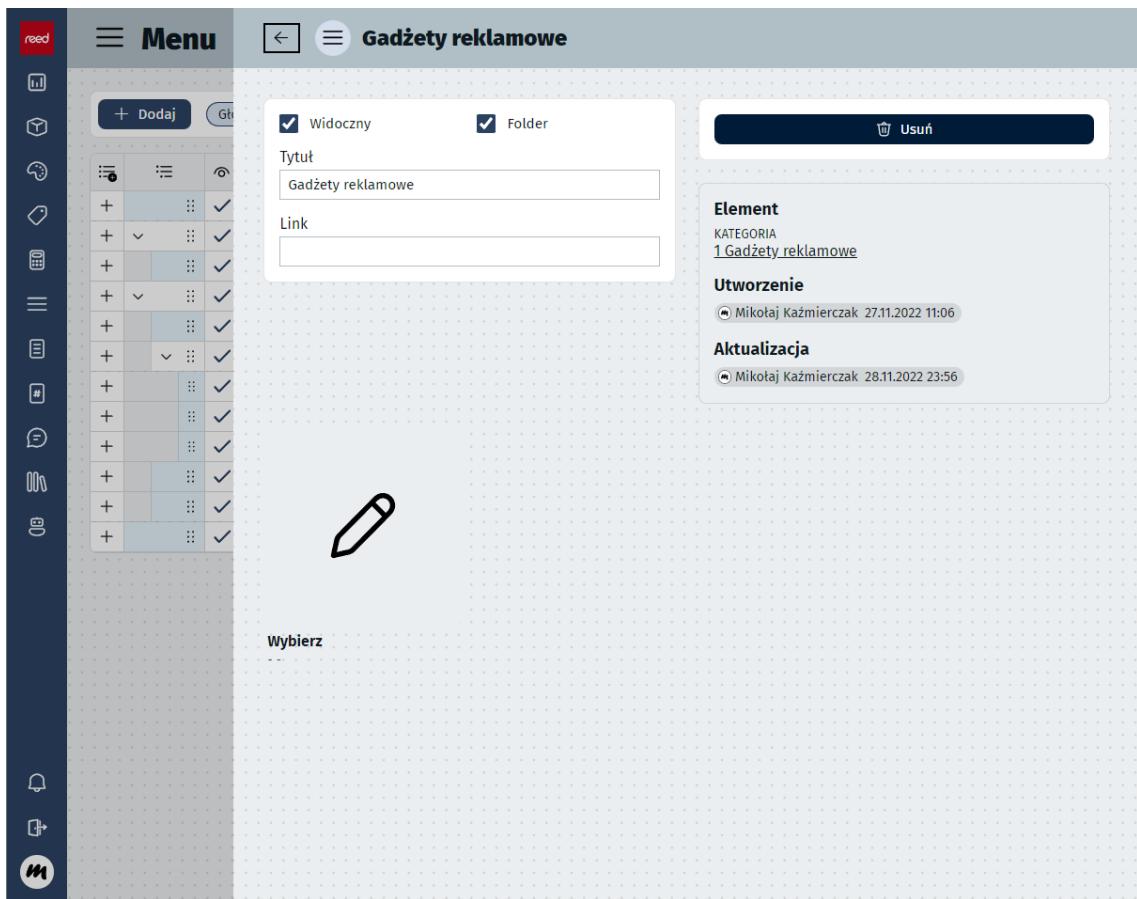
ID	Tytuł	Element	Utworzenie	Aktualizacja
6	1 Gadżety reklamowe	1 Gadżety reklamowe	Mikołaj Kaźmierczak 27	Mikołaj Kaźmierczak 28
10	2 Kalendarze		Mikołaj Kaźmierczak 27	Mikołaj Kaźmierczak 28
17	3 Test		Mikołaj Kaźmierczak 27	Mikołaj Kaźmierczak 09
3	3.1 Usługi laserowe	Uslugi laserowe	Mikołaj Kaźmierczak 25	Mikołaj Kaźmierczak 27
18	3.2 Tytuł		Mikołaj Kaźmierczak 27	Mikołaj Kaźmierczak 27
19	3.2.1 Kolejny		Mikołaj Kaźmierczak 27	Mikołaj Kaźmierczak 28
22	3.2.2 Podtytuł		Mikołaj Kaźmierczak 27	Mikołaj Kaźmierczak 28
23	3.2.3 Kolejne produkty		Mikołaj Kaźmierczak 27	Mikołaj Kaźmierczak 28
20	3.4 Sekcja		Mikołaj Kaźmierczak 27	Mikołaj Kaźmierczak 28
21	3.5 Kolumna		Mikołaj Kaźmierczak 27	Mikołaj Kaźmierczak 28
1	4 Drukarnia	Drukarnia	Mikołaj Kaźmierczak 06	Mikołaj Kaźmierczak 28

Rys. 3.19: Widok listy elementów menu (opracowanie własne). Aktualnie edytowane menu można wybrać za pomocą przycisków pod górną belką ("Główne", "Stopka")

Warto zaznaczyć, że w lewym dolnym rogu witryny został umieszczony komponent, który wyświetlany jest tylko dla użytkowników zalogowanych. Są to przyciski które prowadzą bezpośrednio do panelu administracyjnego oraz pozwalają na wylogowanie się bez wchodzenia do niego.

3.2.1. Menu

Menu (rys. 3.32) zostało opracowane w sposób kolumnowy. Kolumny opierają się na hierarchii oraz zagnieździeniach elementów menu. Co więcej jeśli elementem menu jest kategoria konwertuje ona kolejne zagnieżdżone kategorie na zagnieżdżone elementy menu. Pierwszy poziom zagnieżdżenia menu generuje główne elementy menu widoczne na górnej belce. Następnie wyświetlane są kolejne zagnieżdżenia w formie pogrubionych tytułów, następnie mniejsze odnośniki, a na końcu bardzo małe odnośniki. Istnieją więc 4 poziomy wizualne zagnieżdżeń. Każde kolejne zagnieżdżenie przyjmuje ostatnią formę wizualną.

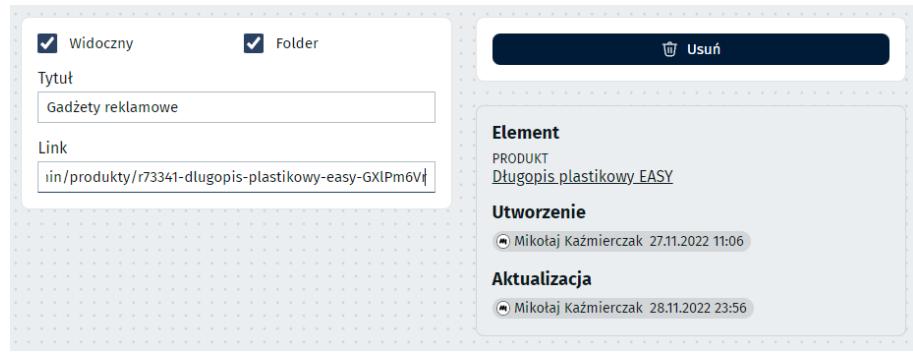


Rys. 3.20: Widok edycji elementu menu (opracowanie własne)

3.2.2. Lista produktów

Nad listą produktów znajdują się szybkie filtry produktów według kategorii. Są one hierarchiczne i oparte na zagnieżdżeniach, a kolejne poziomy wyświetla się po wybraniu poprzednich. Wybranie określonej kategorii wyświetla też wszystkie produkty należące do wszystkich kategorii jej zagnieżdżonych. Jest to rozwiązanie inne niż w panelu administracyjnym. Uznano, że administrator powinien widzieć tylko produkty ściśle przynależne do danej kategorii (od szczegółu do ogólnego), a potencjalny nabywca także produkty należące kategoriom niżej w hierarchii (od ogólnego do szczegółu).

Sama lista produktów ma układ kafelków (rys. 3.33). Każdy z nich wyświetla określone informacje o produkcie: nazwę, kod, cenę (jeśli jest) oraz kolory. Co istotne, jeśli cena zawiera znakowanie jest to bardzo bezpośrednio oznaczone obok ceny. Jak twierdził klient, potencjalni nabywcy często mieli problem z tym aspektem systemu. Stare kafelki (rys. 3.33a) miały jeden istotny problem – oznaczenie, że cena znakowania jest wliczona w cenę wyświetlającą na kafelku nie była wystarczająco przejrzysta dla części potencjalnych nabywców. Mimo, że informacja była podana na spodzie komponentu, użytkownicy najpewniej nie zwracali na nią uwagi ze względu na szare tło bądź niski kontrast tekstu na tle. Nowy kafelek (rys. 3.33a) rozwiązuje ten problem poprzez umieszczenie informacji tuż obok ceny.



Rys. 3.21: Edycja linku menu (opracowanie własne). System automatycznie rozpoznaje czy link prowadzi do określonych elementów systemu, czy też jest zewnętrzny

	ID	Nazwa	Utworzenie	Aktualizacja
✓	1	Drukarnia	Mikołaj Kaźmierczak 23.11.2022 22:30	Mikołaj Kaźmierczak 25.11.2022 06:18
✓	2	Uslugi laserowe	Mikołaj Kaźmierczak 24.11.2022 02:28	Mikołaj Kaźmierczak 27.11.2022 17:21

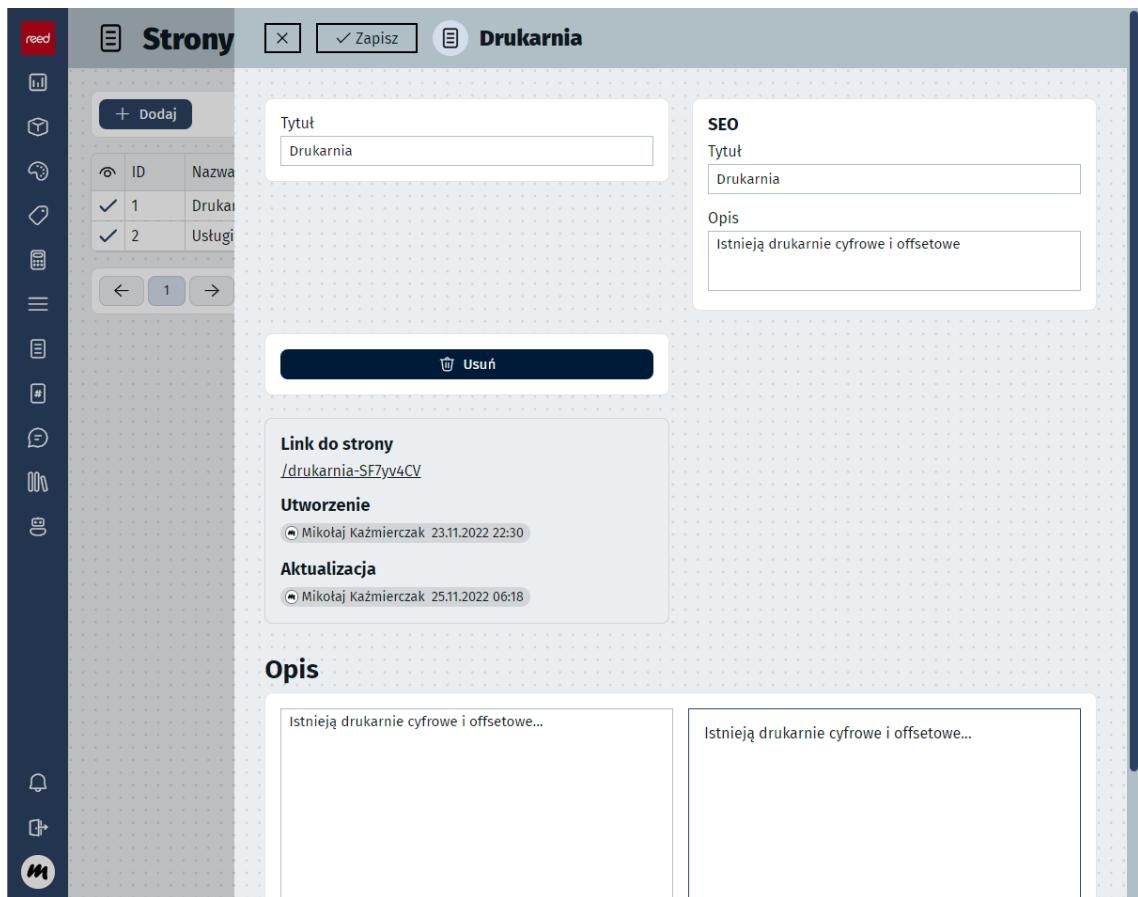
Rys. 3.22: Widok listy stron (opracowanie własne)

3.2.3. Produkt

Strona na której wyświetlany jest produkt (rys. 3.34) zawiera kilka elementów. Po lewej wyświetlona jest galeria w której zdjęcia zmieniają się po najechaniu na ich miniatury, a także warianty kolorystyczne (rys. 3.35, które mają formę listy wraz z ich kodem, nazwą koloru oraz własną, mniejszą galerią. Po prawej wyświetlany jest tytuł (wraz z oflagowaniem promocji lub nowości jeśli taka występuje), kod produktu, opis, cennik oraz formularz kontaktu z administratorem strony.

3.2.3.1. Cennik

Na obecnej stronie klienta cenniki produktów to małe komponenty, w których użytkownik wybiera nakład a następnie wyświetlana jest mu cena (rys. 3.36a). Według klienta tego typu komponent zupełnie się nie sprawdził. Mimo, że wydaje się to być przyjazne w użyciu rozwiązanie klient twierdzi, że jego klienci bardzo często dzwonią do niego z pytaniem o ceny dla innych nakładów, mimo że przycisk "ILOŚĆ" pozwala na sprawdzenie innych cen. Co więcej komponent ten pozwala na wybranie innych opcji znakowania produktu za pomocą przyciski "OPCJE". Według klienta na tenże przycisk użytkownik publiczny już prawie wcale nie zwraca uwagi.



Rys. 3.23: Widok edycji pojedynczej strony (opracowanie własne)

Wraz z klientem wypracowano nowe rozwiązanie wyświetlania cennika. Przyjmie ono formę tabelaryczną (tab. 3.2). Dzięki niemu użytkownik publiczny, szczególnie ten mniej obity z komputerem, będzie mógł szybko zobaczyć jednocześnie wszystkie ceny. Jako, że z założenia wie jakiego nakładu potrzebuje, będzie mógł szybko znaleźć tą cenę, której potrzebuje lub choćby bliską niej.

Ilość	100	500	1000	...
Cena	2.54 zł	2.12 zł	1.36 zł	...

Tabela 3.2: Propozycja nowej, tabelarycznej formy cennika (opracowanie własne)

Nowy cennik wyświetla też ceny promocyjne w przejrzysty sposób. Oprócz tego pokazuje pole znakowania oraz powiadamia o fakcie, że znakowanie jest wliczone w cenę. Porównanie starego i nowego cennika pokazano na rysunku 3.36.

3.2.3.2. Formularz

Formularz zawiera pięć pól. Dwa z nich są obowiązkowe – email oraz treść wiadomości. Trzy z nich są opcjonalne – telefon, imię i nazwisko oraz załączenie pliku. U spodu

ID	Nazwa	Aktualizacja
1	Kontakt	Mikołaj Kaźmierczak 24.11.2022 02:32

Rys. 3.24: Widok listy fragmentów (opracowanie własne)

formularza znajduje się przycisk wysłania zapytania. Jeśli użytkownik nie wypełni pola email lub wiadomości wyświetlona zostanie informacja, że pola są wymagane.

3.3. TESTY UŻYTECZNOŚCI

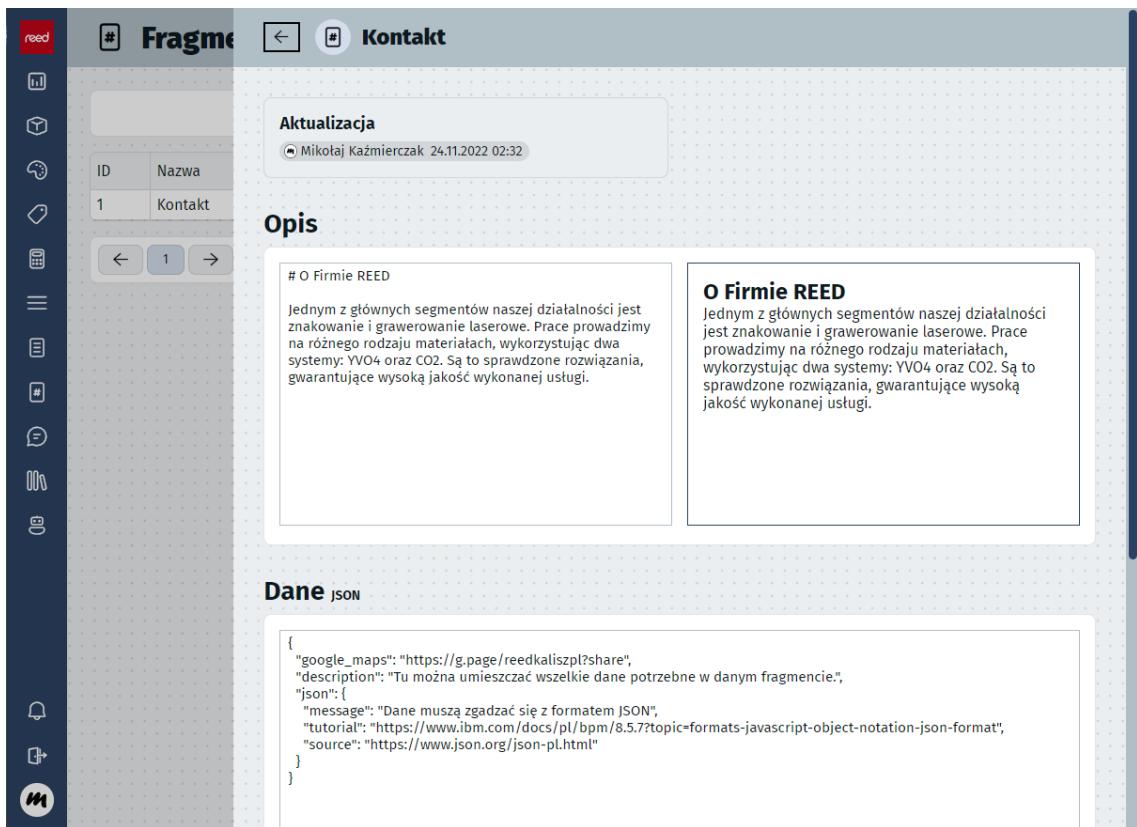
Testy użyteczności pozwalają ocenić *UX* produktu, takiego jak strona internetowa czy aplikacja. Testy są zazwyczaj przeprowadzane na małej grupie reprezentatywnych użytkowników, którzy są proszeni o wykonanie serii zadań na platformie. Celem testów użyteczności jest zidentyfikowanie wszelkich obszarów produktu, które są trudne lub mylące dla użytkowników oraz sformułowanie zaleceń dotyczących poprawy *UX*. Jest to ważny krok w procesie projektowania, ponieważ może pomóc w sprawdzeniu czy produkt jest łatwy i przyjemny w użyciu.

Scenariusze z których korzystano podczas badań z udziałem użytkowników panelu administracyjnego:

- Zaloguj się do systemu.
- Przejdź do widoku kalkulacji, zapoznaj się z nim i dodaj jeden element przy producentie "REED".
- Dodaj nowy produkt starając się wypróbować większość pól w edytorze.
- Wyszukaj produkt korzystając z jego kodu.
- Zmień hierarchię oraz zagnieźdzenie kategorii w liście kategorii.
- Dodaj kategorię.
- Dodaj zdjęcia i ustaw je jako grafikę kategorii.

Testy odbyły się z udziałem pięciu respondentów.

Pierwszym użytkownikiem był student Politechniki Wrocławskiej w wieku 23 lat. Użytkownik po podaniu mu danych do logowania bez problemu zalogował się do platformy. Następnie od razu przeszedł do widoku kalkulacji, który poznał po ikonie. Znalazł tekst "REED", choć stwierdził, że sam nie domyśliłby się, że ta nazwa to producent towaru. Po



Rys. 3.25: Widok edycji pojedynczego fragmentu (opracowanie własne)

spojrzeniu na wartości tabeli stwierdził, że nie wie co oznaczają. Zaczął rozglądać się po panelu i nacisnął przycisk "W jaki sposób wyliczane są ceny?". Tam zaczął analizować instrukcję i uznał że wie z czym powiązane są jej elementy (na tym etapie niektórych jeszcze nie widział, jako że są widoczne tylko w panelu edycji pojedynczego produktu). Porzucił zadanie ze względu na brak zrozumienia o co chodzi w wartościach tabeli. Następnie znalazł widok produktów i po krótkim obyciu się z listą produktów kliknął na przycisk "Dodaj". Następnie zaczął eksperymentować z kolejnymi polami edycji. Wyraził brak zrozumienia pól dotyczących producenta produktu. Innym elementem, który był niezrozumiałym było ustalanie cen. W tym momencie jednak rozpoznał elementy, które wyświetlane były wcześniej w instrukcji w widoku kalkulacji. Stwierdził, że prawdopodobnie instrukcja była bardziej zrozumiała gdyby wiedział w jaki sposób funkcjonuje biznes. Kolejne scenariusze nie sprawiły użytkownikom większych problemów. Użytkownik szybko wyszukał produkt o określonym produkcie. Zmiana hierarchii kategorii spotkała się z pewną dozą ekscytacji ze względu na dynamikę procesu. Użytkownikowi szczególnie zaimponowało jak łatwa jest zmiana poziomu zagnieżdżenia. Użytkownik nie miał też problemu z dodaniem nowej kategorii. Następnie dodał zdjęcie bezpośrednio w edytorze zdjęcia kategorii i ustawił je.

Kolejnych dwóch użytkowników także pochodziło z Politechniki Wrocławskiej i byli w tym samym wieku co pierwszy respondent. Użytkownicy dochodzili do podobnych rezultatów co pierwszy respondent. Warto zaznaczyć jednak, że trzeci użytkownik nie zauważył

Spam	ID	Imię i nazwisko	Email	Telefon	Utworzenie	Aktualizacja
×	2	Dariusz Kaźmierczak	dariusz@reed.kalisz.pl	321456987	Dariusz Kaźmierczak 2	Dariusz Kaźmierczak 2
✓	3	Mikołaj Kaźmierczak	mikolaj@kazmierczak	098321876	Mikołaj Kaźmierczak 2	Mikołaj Kaźmierczak 2
×	4	Mikołaj Kaźmierczak	mikolaj@kazmierczak.pl	3456789	Mikołaj Kaźmierczak 2	Mikołaj Kaźmierczak 2

Rys. 3.26: Widok listy zapytań (opracowanie własne)

przycisku "W jaki sposób wyliczane są ceny?" i w ogóle porzucił próby wypełnienia pól powiązanych z cenami. Większość problemów tych użytkowników można jednak powiązać z ich brakiem wiedzy na temat działalności biznesu.

Kolejnymi testowanymi użytkownikami byli docelowi administratorzy systemu. Tutaj przebieg kolejnych scenariuszy także przebiegał bardzo podobnie co u pierwszego respondenta, natomiast więcej problemów sprawiła zmiana hierarchii kategorii. Zrozumienie na którym poziomie zagnieżdżenia znajdzie się kategoria po zwolnieniu przycisku myszy było kłopotliwe. Obaj administratorzy zaczęli jednak zmieniać pozycję kolejnych kategorii i powoli nabierali zrozumienia tego jak działa rozwiązywanie. Można więc uznać, że rozwiązanie może wymagać nabrania przyzwyczajenia przez użytkowników. Administratorzy mieli jednak znacznie mniej problemów ze zrozumieniem cenników. Głównie wiąże się to z faktem, że w dużym stopniu spodziewali się w jaki sposób system powinien w tym zakresie działać. Bez problemu dodawali z zainteresowaniem kolejne rzędy tabeli w widoku kalkulacji i testowali w jaki sposób zmieniają się cenniki w widoku edytora produktu po wprowadzeniu zmian w widoku kalkulacji.

Kolejne testy z administratorami odbywały się w formie *agile*, z związku z tym był one często powtarzane, a zmiany wprowadzane na bieżąco. Panele kalkulacji zostały więc doprowadzone do pełnego zrozumienia, a instrukcja dopracowana w taki sposób, by administratorzy mogli do niej wracać, jeśli przez dłuższy czas nie mieli potrzeby aktualizacji kalkulatorów.

Testy użytkowniczości były bardzo pomocne w rozwoju oprogramowania. Dzięki nim wprowadzono poprawki w instrukcji działania kalkulacji. Oprócz tego podczas testów zauważono brak edytowalnego pola edycji wielkości znakowania przy każdym cenniku, a także dostosowano kilka ikon użytych w interfejsie na bardziej zrozumiałe. Ikony zostały też podpisane, a sam podpis wyświetlany jest po najechaniu na ikonę kursorem myszy. Te i wiele innych drobnych poprawek bardzo usprawniło ogólną jakość systemu.

Rys. 3.27: Widok edycji pojedynczego zapytania (opracowanie własne)

3.3.1. Efektywność systemu

Sprawdzono ile czasu zajmują określone czynności w starym oraz nowym systemie. Mierzenie czasu przy każdej czynności rozpoczynano od pierwszego domyślnego panelu określonego systemu.

Lista czynności:

- **A.** Logowanie do systemu.
- **B.** Dodanie nowego produktu wraz z trzema zdjęciami oraz wyliczeniem cennika.
- **C.** Dodanie nowej kategorii.
- **D.** Zmiana pozycji i zagnieżdżenia kategorii.

Wykonana czynność	A	B	C	D
Czas potrzebny w starym systemie [s]	12.96	442.13	18.41	-
Czas potrzebny w nowym systemie [s]	5.67	76.21	15.35	2.23

Tabela 3.3: Zestawienie czasu niezbędnego na wykonanie każdej określonej czynności w starym oraz nowym systemie (opracowanie własne)

W tabeli 3.3 widać, że wszystkie typowe czynności wykonuje się znacznie szybciej w nowym systemie. Co więcej, w starym systemie zmiana poziomu zagnieżdżenia w liście kategorii jest niemożliwa. W starym systemie należy stworzyć zupełnie nową kategorię i

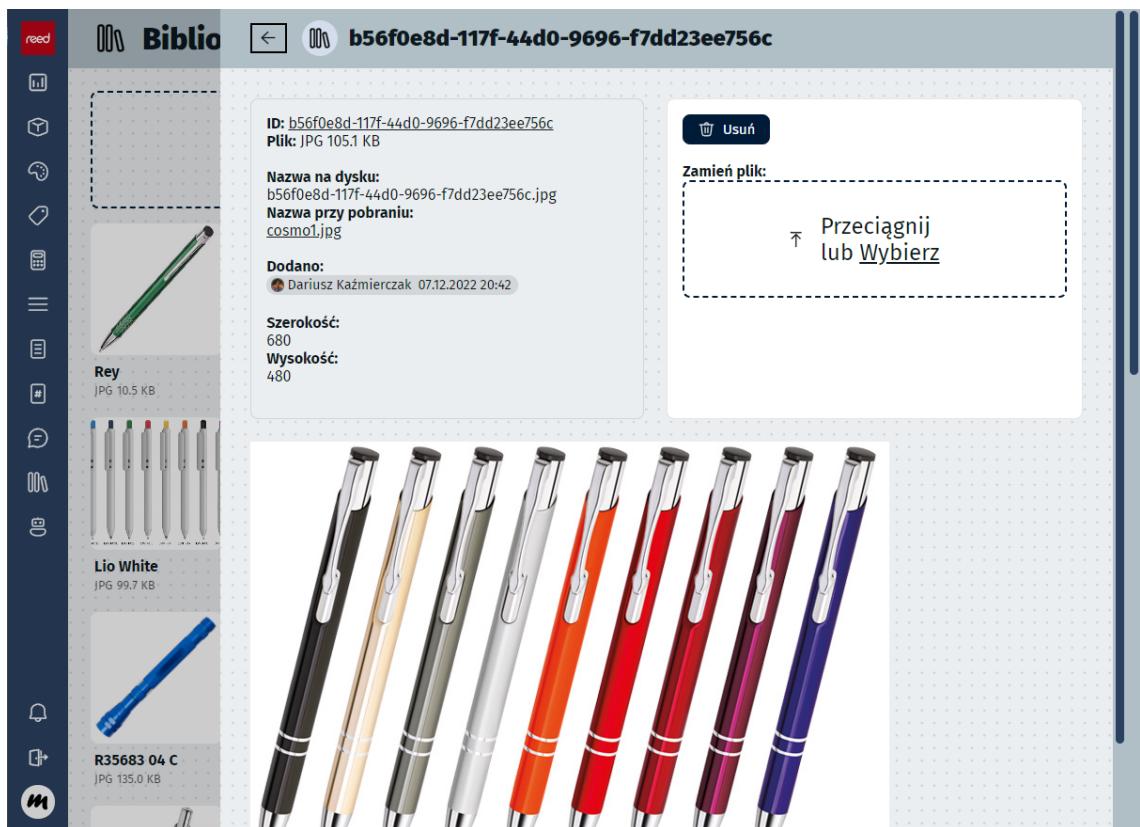


Rys. 3.28: Widok biblioteki plików (opracowanie własne)

przepisać wartości z poprzedniej, co zajmuje znacznie więcej czasu i właściwie należy to uznać za inny rodzaj czynności.



Rys. 3.29: Paginacja w bibliotece plików (opracowanie własne)



Rys. 3.30: Widok edycji pojedynczego pliku (opracowanie własne)

reed GADŻETY REKLAMOWE KALENDARZE TEST DRUKARNIA Szukaj...

Gadżety reklamowe

NARZĘDZIA I AKCESORIA ELEKTRONIKA I AKCESORIA ART. PIŚMIENNICZE BIURO I PRACA JEDZENIE I PICIE SPORT I PODRÓŻE
TORBY I PLECAKI PARASOLE I PELETRYNY ART. EKOLOGICZNE ODZIEŻ I AKCESORIA ZDROWIE I URODA DLA DZIECI, GRY
Długopisy ekologiczne Długopisy metalowe Długopisy plastikowe Ołówki, kredki Parker Zestawy piśmiennicze

Długopis metalowy REY R-
1.59 zł Ze znakowaniem
Długopis plastikowy EASY R73341
1.17 zł Ze znakowaniem
Zestaw piśmienniczy Jetmore R02312
14.67 zł Ze znakowaniem
Długopis metalowy Mooi R0011
1.55 zł Ze znakowaniem
Długopis metalowy Bello Touchpen RE0012
1.58 zł Ze znakowaniem

Rys. 3.31: Publiczna strona internetowa (opracowanie własne)

reed GADŻETY REKLAMOWE KALENDARZE TEST DRUKARNIA Szukaj...

NARZĘDZIA I AKCESORIA	ELEKTRONIKA I AKCESORIA	ART. PIŚMIENNICZE	BIURO I PRACA	JEDZENIE I PICIE
Brelaki	Dla telefonów i smartfonów	Długopisy ekologiczne	Antystresy	Kubki ceramiczne
Latarki	Powerbanki	Długopisy metalowe	Notesy, karteczki	Kubki izotermiczne
Narzędzia	Pendrive	Długopisy plastikowe	Smycze reklamowe	Termosy
Miarki		Ołówki, kredki	Zapach reklamowy	Bidony, butelki
		Parker		
		Zestawy piśmiennicze		

SPORT I PODRÓŻE **TORBY I PLECAKI** **PARASOLE I PELETRYNY** **ART. EKOLOGICZNE** **ODZIEŻ I AKCESORIA** **ZDROWIE I URODA**

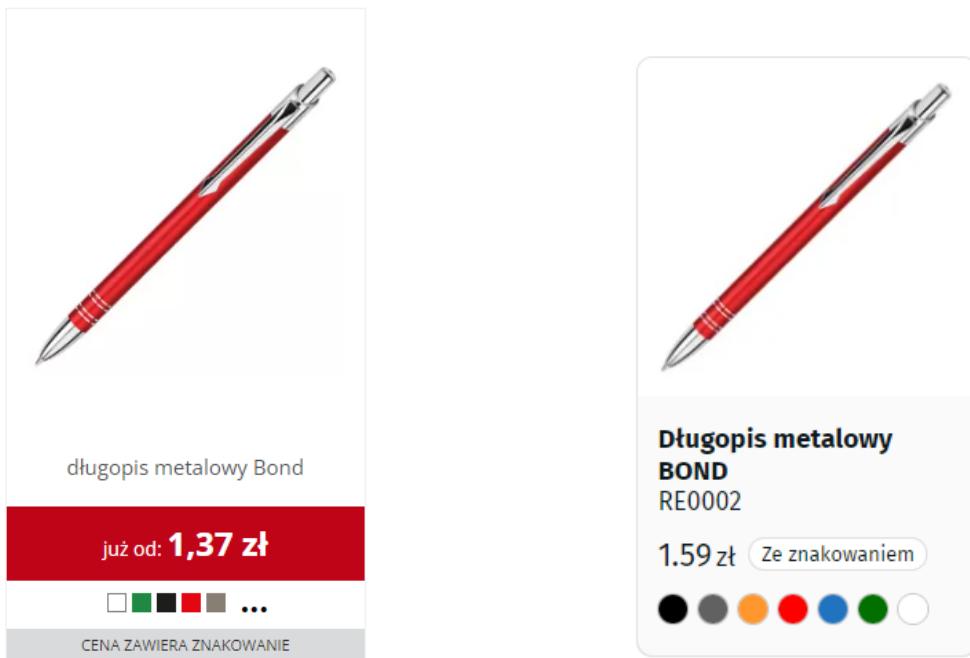
Gadżety odblaskowe Torby na zakupy
Artykuły rowerowe Torby sportowe
Plecaki Worki

DLA DZIECI, GRY

Maskotki
Gry

Długopis metalowy REY R-
1.59 zł Ze znakowaniem
Długopis plastikowy EASY R73341
1.17 zł Ze znakowaniem
Zestaw piśmienniczy Jetmore R02312
14.67 zł Ze znakowaniem
Długopis metalowy Mooi R0011
1.55 zł Ze znakowaniem
Długopis metalowy Bello Touchpen RE0012
1.58 zł Ze znakowaniem

Rys. 3.32: Menu na publicznej stronie internetowej (opracowanie własne)



(a) Komponent produktu w liście na aktualnej stronie internetowej [8]

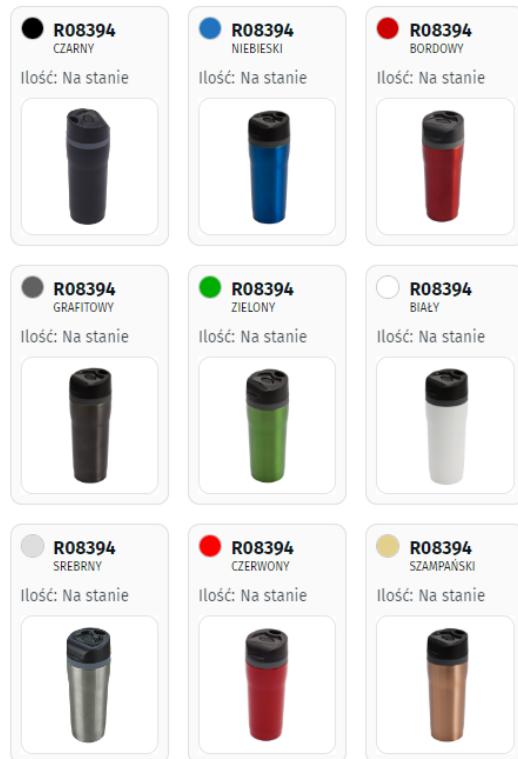
(b) Nowy komponent produktu (opracowanie własne)

Rys. 3.33: Porównanie komponentu produktu na liście produktów na obecnej oraz nowej publicznej stronie internetowej

Ilość	25	50	100	200	300
Cena	47,59 zł	45,18 zł	43,88 zł	43,23 zł	42,78 zł

Rys. 3.34: Widok pojedynczego produktu na publicznej stronie internetowej (opracowanie własne)

Kolory



Rys. 3.35: Lista wariantów kolorystycznych produktu wraz z małymi galeriami zdjęć (opracowanie własne)

1,11 zł

CENA ZAWIERA ZNAKOWANIE

ILOŚĆ

OPCJE

Cena jednostkowa netto

(a) Cennik produktu na obecnej stronie internetowej klienta [7]

Cennik

Pole znakowania 70x4 mm

Ilość	100	200	300	500	1000
Cena	1.89 zł 2.17	1.74 zł	1.29 zł 1.59	1.74 zł	2.17 zł

CENY ZE ZNAKOWANIEM

(b) Nowy cennik produktu (opracowanie własne)

Rys. 3.36: Porównanie cenników produktu na obecnej oraz nowej publicznej stronie internetowej

Zapytaj / Zamów

Zapytaj / Zamów

Email*

Telefon

Imię i nazwisko

Wiadomość*

Dzień dobry, czy "Kubek izotermiczny Winnipeg 350 ml" wciąż jest dostępny?
Pozdrawiam

Załącz plik
 Nie wybrano pliku

Zapytaj / Zamów

Email*

Email jest wymagany

Telefon

Imię i nazwisko

Wiadomość*

Treść wiadomości jest wymagana

Załącz plik
 Nie wybrano pliku

(a) Formularz z wypełnioną domyślnie za użytkownika treścią

(b) Formularz wypełniony niepoprawnie i wyświetlanie błędów

Rys. 3.37: Formularz wysyłania zapytań do klienta od potencjalnego nabywcy (opracowanie własne)

4. DALSZE PRACE

Warto zaznaczyć, że praca z klientem odbywała się w formie *agile*. Klient kolejne uwagi zgłasza więc na bieżąco. Są one następnie implementowane w jak najkrótszym czasie. Bardzo często są to drobne usprawnienia, które jednak bardzo usprawniają platformę całosciowo. Dobrym przykładem jest funkcjonalność określenia pola znakowania przy każdym cenniku. Na początku informacja ta była umieszczana bezpośrednio w opisie produktu przy informacjach technicznych, przez co była często pomijana przez potencjalnych nabywców. Innym przykładowym usprawnieniem było sprawienie by dodana do menu kategoria jednocześnie dodawała też wszystkie pozostałe zagnieżdżone kategorie. To usprawnienie sprawiło, że klient nie musiał już dwukrotnie wypełniać bardzo podobnej struktury.

Procesem, który jest stały podczas pracy nad oprogramowaniem jest też refaktoryzacja kodu. W tym projekcie również sporo czasu przeznaczane jest na ten proces. Kod jest modyfikowany w taki sposób, który nie zmienia jego funkcjonalności, ale poprawia strukturę i wydajność. Najczęściej przejawia się to zmianą nazw zmiennych, aby ich cel był bardziej jasny, reorganizacją kodu, aby był łatwiejszy do zrozumienia, poprawą algorytmów, aby program działał bardziej wydajnie. Dzięki takim zmianom kod jest łatwiejszy w utrzymaniu, a wprowadzanie kolejnych zmian jest znacznie szybsze. Co więcej, naprawa błędów staje się prostsza, ze względu na zrozumiałosć kodu.

Do przyszłych funkcjonalności, które w większości zostały już uzgodnione, bądź są w fazie rozmów należą:

- Automatyczne, cykliczne aktualizacje cen oraz stanów magazynowych z wykorzystaniem *API* zewnętrznych dostawców. Taka funkcjonalność znacznie zmniejszy czas, który trzeba spędzać na platformie. Większość elementów systemu, które są potrzebne do implementacji tej funkcjonalności jest już zaimplementowanych. Należy sprawdzić w jaki sposób działają *API* dostawców i zaimplementować funkcje, które przetworzą zewnętrzne dane na format dostosowany do *API* narzędzia *Directus*. Następnie należy cyklicznie uruchamiać te funkcje na serwerze *Heimdall*, który zajmie się aktualizacją danych bez potrzeby żadnej ingerencji użytkownika.
- Komponent, który pozwoli użytkownikom na wgranie do systemu, a następnie nałożenie grafiki na produkt w polu o określonej wielkości. Dzięki temu administratorzy dostaną już pogląd na to w jaki sposób powinno odbyć się znakowanie produktu. Obecnie administrator musi prowadzić rozmowę z potencjalnym nabywcą i samemu utworzyć najpierw

grafikę poglądową. Usprawniłyby to więc proces składania zamówienia i co więcej dałoby poczucie większej kontroli nad procesem znakowania potencjalnemu nabywcy.

- Mniej istotnym z punktu widzenia funkcjonalności, ale ważnym usprawnieniem jeśli chodzi o *UX*, jest poprawa projektu graficznego panelu administracyjnego. W obecnej formie projekt jest spójny graficznie, ale atrakcyjność wizualna zdecydowanie mogłaby zostać usprawniona.

Warto też wspomnieć o oczywistym kolejnym kroku, czyli ustaleniu z klientem projektu, a następnie implementacji publicznej strony internetowej. System jest już gotowy pod taki projekt, co więcej istnieje już implementacja części funkcjonalności, która została przedstawiona wcześniej. W obecnym stanie rozwoju została zaimplementowana bardziej po to, by zaprezentować możliwości systemu, oraz ułatwić administratorom umiejscowienie elementów wypełnianych w panelu administracyjnym na docelowej stronie.

PODSUMOWANIE

Zaimplementowana aplikacja webowa spełniła wszystkie oczekiwania klienta. Panel administracyjny został opracowany i dostosowany do administratorów systemu. Zrealizowane zostały wszystkie wszystkie zdefiniowane wymagania funkcjonalne i niefunkcjonalne. Administrator może zalogować się oraz edytować określone wcześniej elementy systemu. Jest też w stanie ukrywać je przed potencjalnym nabywcą towarów. Ten z kolei może przeglądać publiczną stronę internetową i wysyłać zapytania do administratorów. System stanowi pełny funkcjonalnie serwis do katalogowania produktów.

Istnieją też perspektywy dalszego rozwoju aplikacji aby jeszcze bardziej usprawniła zadania biznesowe. Są to między innymi automatyczne, cykliczne aktualizacje cen oraz stanów magazynowych korzystając z *API* zewnętrznych dostawców, komponent pozwalający na wgranie do systemu, a następnie nałożenie grafiki na zdjęcie produktu w polu o określonej wielkości. Najistotniejsze jest jednak uzgodnienie z klientem projektu graficznego, a następnie implementacja publicznej strony internetowej.

Praca z klientem odbywa się w formie agile. Klient jest zadowolony z tego w jaki sposób dotychczas przebiegały konsultacje oraz z przebiegu pracy nad systemem. Warto także zaznaczyć, że testy użytkowe wykazały zrozumiałość interfejsu aplikacji webowej dla administratorów systemu. Istnieje też możliwość dalszych usprawnień projektu graficznego. Na obecnym etapie, projekt można uznać za narzędzie kompletne funkcjonalnie w kontekście realnego scenariusza zastosowania systemu.

BIBLIOGRAFIA

- [1] *What Is a Framework?* | *Codecademy News.* 2021. Dostępne pod adresem: <https://www.codecademy.com/resources/blog/what-is-a-framework/>.
- [2] *Aplikacja webowa vs strona internetowa - cechy i różnice?* | *Strefa Inżyniera.* 2022. Dostępne pod adresem: <https://strefainzyniera.pl/artykul/11834/aplikacja-webowa-vs-strona-internetowa--cechy-i-roznice>.
- [3] *Chromium.* 2022. Dostępne pod adresem: <https://www.chromium.org/chromium-projects/>.
- [4] *Color Tool | Material Design.* 2022. Dostępne pod adresem: <http://material.io/tools/color/>.
- [5] *Czym jest API (Interfejs programowania)?* | *OVHCloud.* 2022. Dostępne pod adresem: <https://www.ovhcloud.com/pl/learn/what-is-api/>.
- [6] *Directus: The Modern Data Stack, Democratized.* 2022. Dostępne pod adresem: <https://directus.io/>.
- [7] *długopis metalowy cosmo / Długopisy metalowe / Artykuły piśmiennicze / Gadżety reklamowe | REED Kalisz.* 2022. Dostępne pod adresem: https://reed.kalisz.pl/produkt/dlugopisy-metalowe_38/dlugopis-metalowy-cosmo_18.html.
- [8] *Długopisy reklamowe z nadrukiem, gadżety dla firm, druk niskonakładowy, grawerowanie laserowe | REED Kalisz.* 2022. Dostępne pod adresem: <https://reed.kalisz.pl/>.
- [9] *Fluenticons | Fluent UI System Icons.* 2022. Publisher: Microsoft, Dostępne pod adresem: <https://github.com/microsoft/fluentui-system-icons>.
- [10] *framework | Cambridge Dictionary.* 2022. Dostępne pod adresem: <https://dictionary.cambridge.org/dictionary/english-polish/framework>.
- [11] *Medusa - the open source Shopify alternative | Medusa.* 2022. Dostępne pod adresem: <https://medusajs.com/>.
- [12] *Obowiązek informacyjny | REED Kalisz.* 2022. Dostępne pod adresem: <https://reed.kalisz.pl/obowiazek-informacyjny.html>.
- [13] *Programista czy developer? Mini-słownik nazw stanowisk w branży IT | Comarch.* 2022. Dostępne pod adresem: <https://kariera.comarch.pl/blog/programista-czy-developer-mini-slownik-nazw-stanowisk-w-branzy-it/>.
- [14] *Start and grow your e-commerce business | Shopify.* 2022. Dostępne pod adresem: <https://www.shopify.com/>.
- [15] *Strapi: Open source Node.js Headless CMS.* 2022. Dostępne pod adresem: <https://strapi.io/>.
- [16] *Svelte: Cybernetically enhanced web apps.* 2022. Dostępne pod adresem: <https://svelte.dev/>.

- [17] *SvelteKit: The fastest way to build Svelte apps*. 2022. Dostępne pod adresem: <https://kit.svelte.dev/>.
- [18] *The State of JS 2021: Libraries*. 2022. Dostępne pod adresem: <https://2021.stateofjs.com/en-US/libraries/>.
- [19] *What is BaaS? Backend-as-a-Service vs. serverless | CloudFlare*. 2022. Dostępne pod adresem: <https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/>.
- [20] *WooCommerce*. 2022. Dostępne pod adresem: <https://woocommerce.com/>.
- [21] Barth, A., *HTTP State Management Mechanism*, Request for Comments RFC 6265, Internet Engineering Task Force. 2011. Dostępne pod adresem: <https://datatracker.ietf.org/doc/rfc6265/>.
- [22] Flanagan, D., *JavaScript: the definitive guide: master the world's most-used programming language*, seventh edition wyd. (O'Reilly Media, Inc, Beijing [China] ; Sebastopol, CA, 2020).
- [23] Ford, N., Richards, M., Sadlage, P.J., Dehghani, Z., *Software architecture - The Hard Parts: Modern Trade-Off Analyses for Distributed Architectures*, first edition wyd. (O'Reilly, Bejing Boston Farnham Sebastopol Tokyo, 2021).
- [24] Gaiman, N., *Mitologia nordycka* (MAG, 2020). OCLC: 1332819948.
- [25] Krug, S., Cieslak, P., *Nie kaz mi myslę!: o zyciowym podejsciu do funkcjonalnosci stron internetowych*, wyd. 3 wyd. (Helion, Gliwice, 2014). OCLC: 892605323.
- [26] Martin, R.C., *Clean code: a handbook of agile software craftsmanship*, 17th printing wyd. (Prentice Hall, Upper Saddle River, NJ, 2016). OCLC: 1057907478.
- [27] Martin, R.C., *Clean architecture: a craftsman's guide to software structure and design* (Prentice Hall, Boston, 2018). OCLC: 1003645626.
- [28] Mościchowska, I., Rogoś-Turek, B., *Badania jako podstawa projektowania user experience*, wydanie i - 5 dodruk wyd. (Wydawnictwo Naukowe PWN, Warszawa, 2019). OCLC: 1142861105.
- [29] Resig, J., Bibeault, B., Maras, J., *Secrets of the JavaScript ninja*, second edition wyd. (Manning, Shelter Island, 2016). OCLC: ocn943594805.
- [30] Richards, M., *Fundamentals of software architecture: an engineering approach*, first edition wyd. (O'Reilly Media, Inc, Sebastopol, CA, 2020).
- [31] Thomas, D., Hunt, A., *The pragmatic programmer, 20th anniversary edition: journey to mastery*, second edition wyd. (Addison-Wesley, Boston, 2019).

SPIS RYSUNKÓW

2.1	Diagram przypadków użycia (opracowanie własne)	10
2.2	Strona internetowa platformy <i>Shopify</i> [14]	12
2.3	Strona internetowa platformy <i>WooCommerce</i> [20]	13
2.4	Strona internetowa platformy <i>MedusaJS</i> [11]	13
2.5	Graficzna reprezentacja zależności między elementami systemu (opracowanie własne)	16
2.6	Przykładowe ikony z zestawu <i>Fluenticons</i> [9]. Ikony mają zaokrąglone krawędzie oraz brak wypełnienia	18
3.1	Widok panelu logowania (opracowanie własne)	20
3.2	Widok panelu aktywności (opracowanie własne)	21
3.3	Widok panelu kalkulacji (opracowanie własne). Poufne wartości zostały ocenione aby chronić interesy przedsiębiorstwa. Po lewej panel boczny gdzie można edytować odgórne marże oraz "widoki". Po prawej tabele kalkulatorów	22
3.4	Jedna ze starszych wersji instrukcji dla administratorów w jaki sposób ustalana jest cena (opracowanie własne). Forma instrukcji była nieprzystępna, w związku z czym została zmieniona. Instrukcja może być nieczytelna w pracy, natomiast w panelu administratora została dostosowana do wielkości ekranu	23
3.5	Ostateczna wersja instrukcji dla administratorów w jaki sposób ustalana jest cena (opracowanie własne). Zrozumienie instrukcji zostało sprawdzone bezpośrednio na administratorach. Forma instrukcji jest bardziej obrazowa, linie prowadzą po kolejne według komponentów aż do ostatecznej ceny dla określonego nakładu. Instrukcja może być nieczytelna w pracy, natomiast w panelu administratora została dostosowana do wielkości ekranu	24
3.6	Widok tabeli kalkulatorów (opracowanie własne). Poufne wartości zostały ocenione aby chronić interesy przedsiębiorstwa	27
3.7	Widok produktów (opracowanie własne). U góry widać małą belkę, na której znajdują się przyciski dodania nowego produktu (także przycisk dodania go od razu do wybranej aktualnie kategorii) oraz przeszukiwanie produktów (które można uruchomić skrótem klawiszowym "Ctrl+Q". Na dole znajduje się element paginacji produktów, można w nim wybrać ile produktów ma być pokazywanych jednocześnie	28
3.8	Panel edycji pojedynczego produktu (opracowanie własne). Na górnjej belce edytora znajduje się tytuł produktu oraz przycisk umożliwiający powrót do listy produktów. Jeśli użytkownik wprowadzi zmiany w produkcie, przycisk ten jest zastępowany przez przyciski zapisu oraz anulowania bieżących zmian	29

3.9	Pole edycji opisu produktu (opracowanie własne)	30
3.10	Pola dostosowywania cennika produktu (opracowanie własne). Wersja z manualnym wypełnianiem cennika według wybranego widoku. Poufne wartości zostały ukryte aby chronić interesy klienta	30
3.11	Pola dostosowywania cennika produktu (opracowanie własne). Wersja z określeniem ceny jednostkowej i doboru kalkulatorów. Poufne wartości zostały ukryte aby chronić interesy klienta	31
3.12	Edytor galerii produktu (opracowanie własne)	32
3.13	Edycja wariantów kolorystycznych produktu (opracowanie własne)	33
3.14	Widok listy kolorów (opracowanie własne)	34
3.15	Widok edycji pojedynczego koloru (opracowanie własne)	35
3.16	Widok listy kategorii (opracowanie własne). Warto zaznaczyć przyciski (druga kolumna tabeli), które umożliwiają przegląd zagnieżdżonych elementów a także zmianę hierarchii i poziomu zagnieżdżenia	36
3.17	Widok edycji pojedynczej kategorii (opracowanie własne). Warto zaznaczyć, że na górnej belce wyświetlane są przyciski anulowania lub zapisania zmian	37
3.18	Interfejs podczas przeciągania elementu hierarchii w nowe miejsce. Sam element zmienia przezroczystość aby pokazać że zostanie zmieniony jego stan. Pola które oznaczają gdzie element heirarchi zostanie umieszczony po puszczaniu przycisku myszy są umieszczone poniżej i intuicyjnie oznaczają gdzie znajdzie się element (opracowanie własne)	38
3.19	Widok listy elementów menu (opracowanie własne). Aktualnie edytowane menu można wybrać za pomocą przycisków pod górną belką ("Główne", "Stopka")	38
3.20	Widok edycji elementu menu (opracowanie własne)	39
3.21	Edycja linku menu (opracowanie własne). System automatycznie rozpoznaje czy link prowadzi do określonych elementów systemu, czy też jest zewnętrzny	40
3.22	Widok listy stron (opracowanie własne)	40
3.23	Widok edycji pojedynczej strony (opracowanie własne)	41
3.24	Widok listy fragmentów (opracowanie własne)	42
3.25	Widok edycji pojedynczego fragmentu (opracowanie własne)	43
3.26	Widok listy zapytań (opracowanie własne)	44
3.27	Widok edycji pojedynczego zapytania (opracowanie własne)	45
3.28	Widok biblioteki plików (opracowanie własne)	46
3.29	Paginacja w bibliotece plików (opracowanie własne)	47
3.30	Widok edycji pojedynczego pliku (opracowanie własne)	47
3.31	Publiczna strona internetowa (opracowanie własne)	48
3.32	Menu na publicznej stronie internetowej (opracowanie własne)	48
3.33	Porównanie komponentu produktu na liście produktów na obecnej oraz nowej publicznej stronie internetowej	49
3.34	Widok pojedynczego produktu na publicznej stronie internetowej (opracowanie własne)	49

3.35	Lista wariantów kolorystycznych produktu wraz z małymi galeriami zdjęć (opracowanie własne)	50
3.36	Porównanie cenników produktu na obecnej oraz nowej publicznej stronie internetowej	50
3.37	Formularz wysyłania zapytań do klienta od potencjalnego nabywcy (opracowanie własne)	51

SPIS TABEL

3.1	Propozycja tabelarycznej formy wprowadzania wartości dla kalkulatora cennika (opracowanie własne)	23
3.2	Propozycja nowej, tabelarycznej formy cennika (opracowanie własne)	41
3.3	Zestawienie czasu potrzebnego na wykonanie każdej określonej czynności w starym oraz nowym systemie (opracowanie własne)	45

Dodatki

A. KOD ŹRÓDŁOWY

<https://github.com/mikolajkazmierczak/engineering-thesis>