

Projekt Zespołowy

„Sklep z podzespołami elektronicznymi”

wyk.:

Paulina Klimczuk

Bartosz Stafiej

Piotr Heinzelman

Politechnika Warszawska

OKNO 20224

Spis treści

| | | |
|-------|--|----|
| 1. | Skład zespołu, role i zadania..... | 4 |
| 2. | Projekt - inspiracje | 5 |
| 2.1 | Projekt – rozszerzenia | 7 |
| 2.2 | Sklep z podzespołami elektronicznymi – screeny z działającego sklepu | 8 |
| 3. | Dokumentacja projektowa | 12 |
| 3.1 | Rejestr produktowy | 13 |
| 3.2 | Rejestr Sprintów | 16 |
| 3.2.1 | Rejestr Sprintu 1 | 16 |
| 3.2.2 | Rejestr Sprintu 2 | 17 |
| 3.2.3 | Rejestr Sprintu 3 | 19 |
| 3.2.4 | Rejestr Sprintu 4 | 20 |
| 3.2.5 | Rejestr Sprintu 5 | 21 |
| 3.3 | Założenia – wykorzystane technologie i narzędzia..... | 21 |
| 3.4 | Opis wymagań funkcjonalnych systemu..... | 24 |
| 3.4.1 | Model przypadków użycia..... | 24 |
| 4. | Ograniczenia | 25 |
| 4.1 | Ograniczenia techniczne | 25 |
| 4.1.1 | Brak implementacji opcji płatności | 25 |
| 4.1.2 | Brak możliwości wysyłania maili..... | 25 |
| 4.2 | Zarządzanie użytkownikami | 25 |
| 4.2.1 | Zewnętrzne usługi autoryzacji | 25 |
| 4.2.2 | Trudności z zarządzaniem użytkownikami i uprawnieniami..... | 25 |
| 4.3 | Ograniczenia czasowe i zasobowe | 26 |
| 4.3.1 | Ograniczony czas realizacji | 26 |
| 4.3.2 | Ograniczone zasoby ludzkie | 26 |
| 4.4 | Technologiczne wyzwania | 26 |
| 4.4.1 | Integracja z zewnętrznymi API..... | 26 |
| 4.4.2 | Złożoność architektury aplikacji..... | 26 |
| 5. | Testowanie aplikacji..... | 27 |
| 5.1 | Metody testowania | 27 |
| 5.1.1 | Testy jednostkowe..... | 27 |
| 5.1.2 | Testy integracyjne | 27 |
| 5.1.3 | Testy funkcjonalne | 27 |
| 5.2 | Przebieg testowania..... | 28 |

| | | |
|-------|----------------------------------|----|
| 5.2.1 | Testowanie podczas sprintów..... | 28 |
| 5.2.2 | Testowanie regresyjne..... | 28 |
| 5.3 | Wyniki testów i wnioski..... | 28 |
| 5.3.1 | Usterki krytyczne | 28 |
| 5.3.2 | Usterki mniej istotne | 28 |
| 6. | Kod źródłowy aplikacji..... | 29 |

1. Skład zespołu, role i zadania

Paulina Klimczuk

Role: prowadząca, Scrum master

Zadania:

- prowadzenie spotkań,
- koordynowanie procesu projektowania aplikacji przez zespół,
- przygotowanie dokumentacji,
- testowanie aplikacji.

Bartosz Stafiej

Role: programista frontend

Zadania:

- zaprojektowanie szaty graficznej,
- przygotowanie i implementacja szablonów (HTML, CSS, JS),
- testowanie aplikacji.

Piotr Heinzelman

Role: programista backend

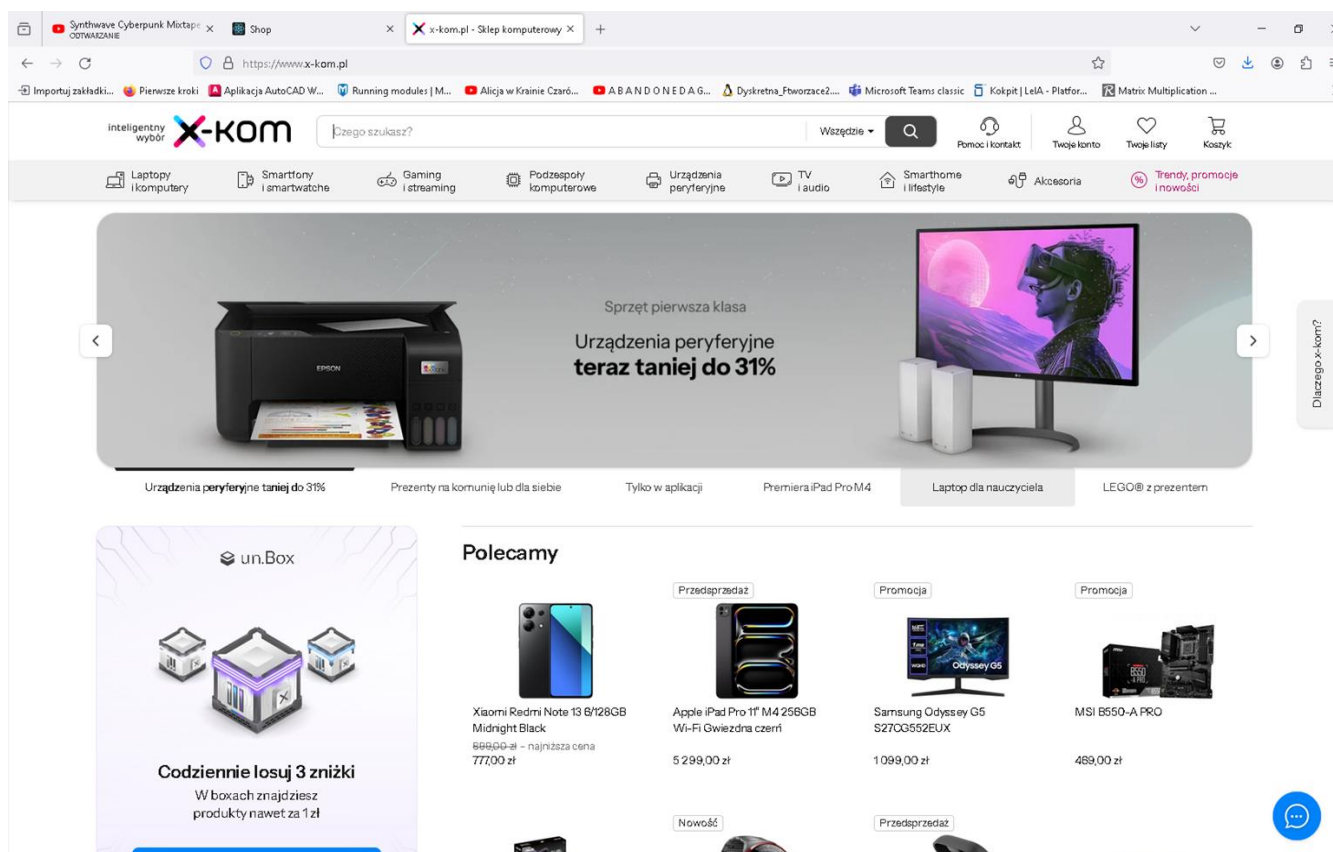
Zadania:

- zaprojektowanie i implementacja struktury aplikacji
- integracja z bazą danych,
- organizacja przestrzeni na GitHub,
- współtworzenie dokumentacji projektowej,
- testowanie aplikacji.

2. Projekt - inspiracje

Przed przystąpieniem do realizacji naszego projektu, przeprowadziliśmy szczegółowy research konkurencji na rynku polskim. Naszym celem było zrozumienie, jakie funkcje i udogodnienia oferują inne sklepy internetowe z podzespołami elektronicznymi oraz jakie elementy mogą wyróżniać naszą aplikację na tle konkurencji.

Oczywiście istnieje wiele podobnych aplikacji, takich jak Komputronik, Morele.net oraz sklepy oferujące szerszy asortyment, jak Avans, RTV Euro AGD, NeoNet, MediaExpert, Media Markt. Najbardziej spodobały nam się rozwiązania zaproponowane przez x-kom.pl:



Główne cechy i funkcjonalności przeanalizowanych sklepów:

x-kom.pl

- szeroki asortyment produktów elektronicznych,
- zaawansowane filtry wyszukiwania,
- program lojalnościowy dla stałych klientów,
- możliwość zakupu na raty oraz liczne promocje,

MediaExpert.pl

- różnorodność produktów elektronicznych i agd,
- dostawa w 24 godziny na wybrane produkty,
- usługi dodatkowe, takie jak montaż i instalacja zakupionych urządzeń,
- liczne akcje promocyjne i wyprzedaże sezonowe.

Komputronik.pl:

- specjalizacja w sprzęcie komputerowym i akcesoriach,
- możliwość zakupu na raty,
- profesjonalne porady i recenzje produktów,
- dział z ofertami specjalnymi oraz outlet.

Nasza aplikacja sklepowa to połączenie modułu prezentacji produktów z modułem sprzedaży, a właściwie zbierania zamówień. Budowę wewnętrzną, założenia oraz procesy projektowaliśmy sami, inspirowaliśmy się jedynie warstwą graficzną oraz niektórymi parametrami, np. obiektem adres.

Większość decyzji projektowych podejmowaliśmy wspólnie, opierając się na doświadczeniach zdobytych podczas korzystania z innych sklepów.

Nasze unikalne funkcje

Chociaż nasz sklep czerpie inspiracje z wymienionych wyżej konkurentów, wprowadziliśmy kilka unikalnych rozwiązań, które mają na celu wyróżnienie naszej aplikacji na rynku:

Konsultacje z rzeczoznawcą

Klienci mogą skorzystać z profesjonalnych porad naszych ekspertów, którzy pomogą w doborze odpowiednich podzespołów oraz rozwiązywaniu problemów technicznych.

Pomoc w sprowadzeniu unikalnych części

Oferujemy możliwość zamówienia rzadkich lub unikalnych części z zagranicy. Dzięki współpracy z międzynarodowymi dostawcami, możemy sprowadzać trudno dostępne komponenty na życzenie klientów.

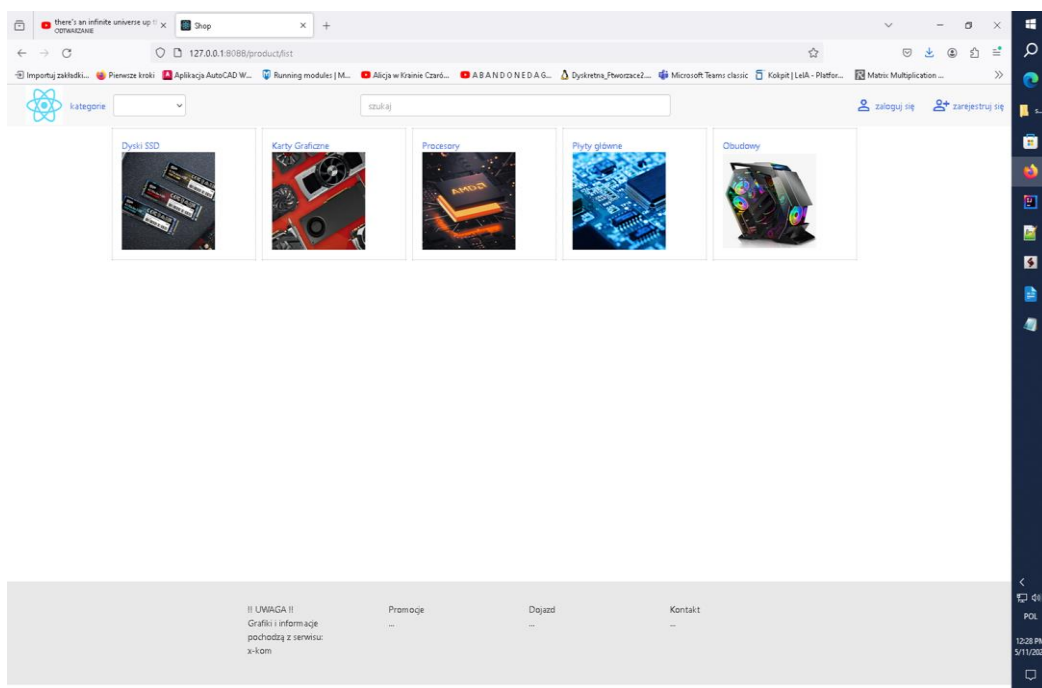
Dokładny research konkurencji pozwolił nam zrozumieć, jakie elementy są kluczowe dla użytkowników sklepów z podzespołami elektronicznymi. Dzięki temu mogliśmy zaprojektować naszą aplikację w sposób, który nie tylko spełnia standardy rynkowe, ale również wprowadza innowacyjne rozwiązania, podnoszące wartość naszej oferty i zadowolenie klientów. Nasze unikalne funkcje, takie jak konsultacje z rzeczoznawcą, pomoc w sprowadzaniu unikalnych części, personalizowane rekomendacje oraz interaktywne narzędzia do konfiguracji, pozwalają nam wyróżnić się na tle konkurencji i zapewnić naszym klientom wyjątkowe doświadczenia zakupowe.

2.1 Projekt – rozszerzenia

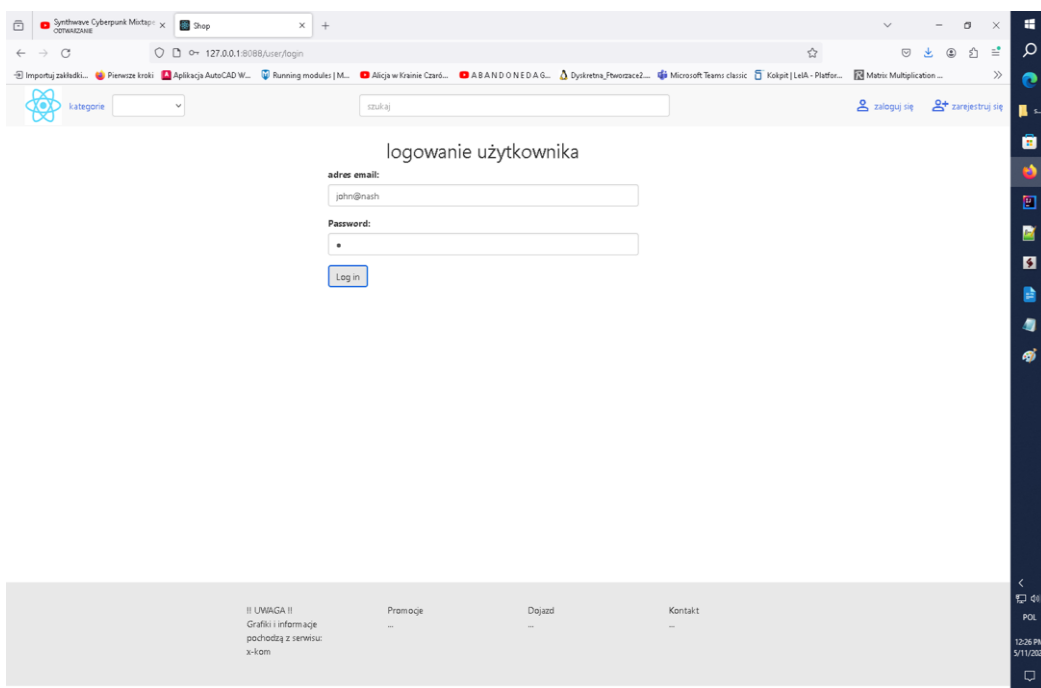
W naszym sklepie wprowadziliśmy dodatkowy moduł analizy sprzedaży konkretnych produktów, który oferuje graficzną prezentację wyników. Umożliwia on również pobieranie danych do pliku EXCEL, co pozwala działowi sprzedaży na dalszą, szczegółową analizę.

2.2 Sklep z podzespołami elektronicznymi – screeny z działającego sklepu

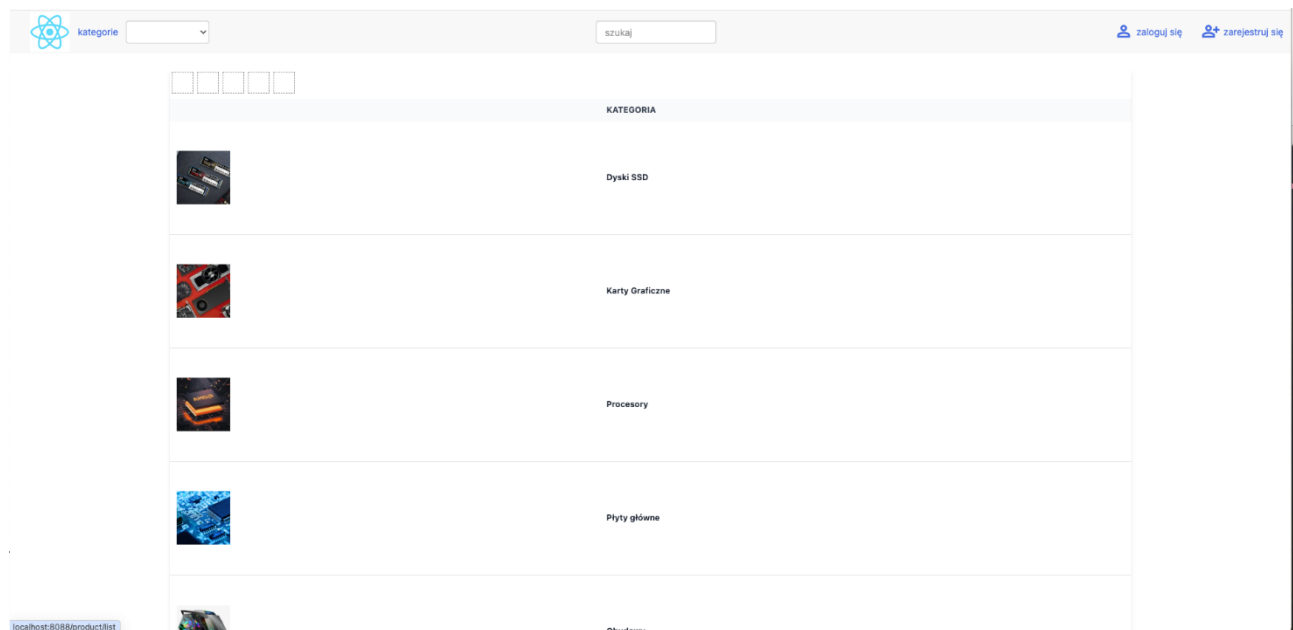
Prezentacja kategorii:



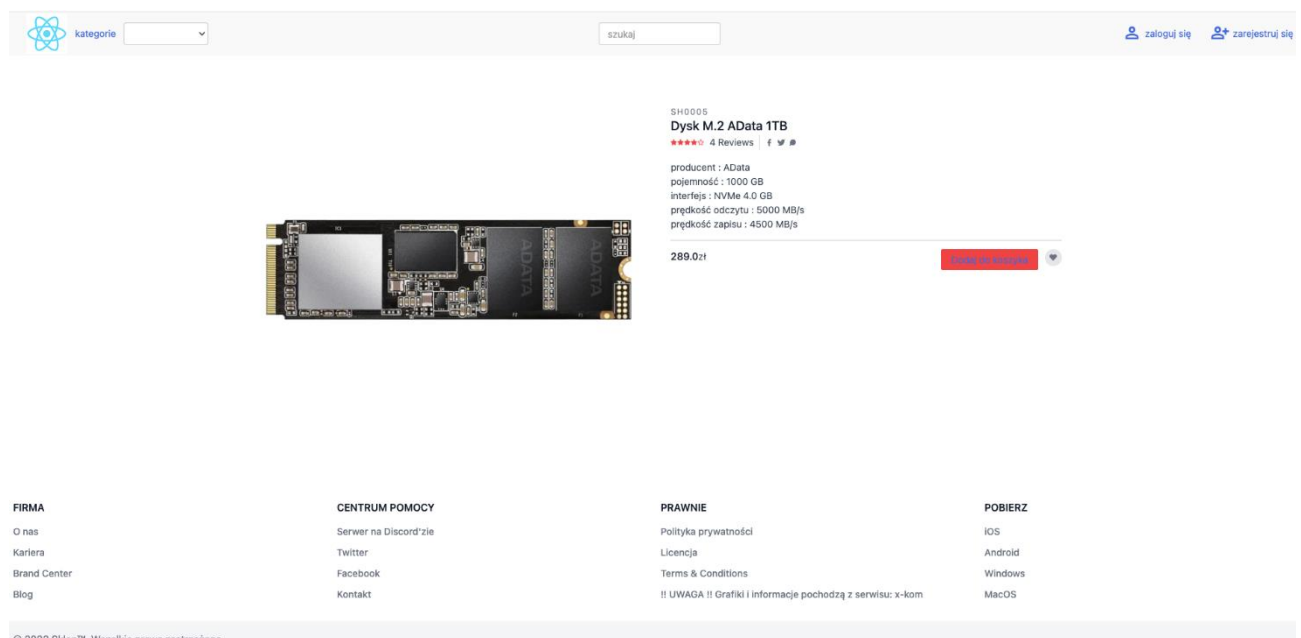
Logowanie użytkownika:



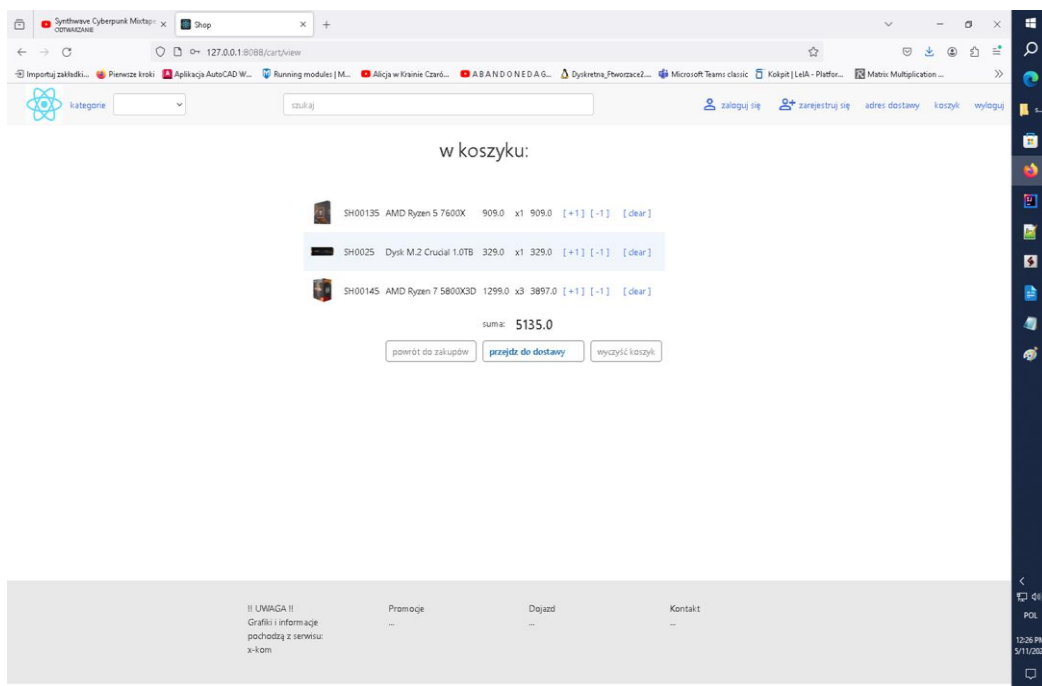
Prezentacja grup produktów:



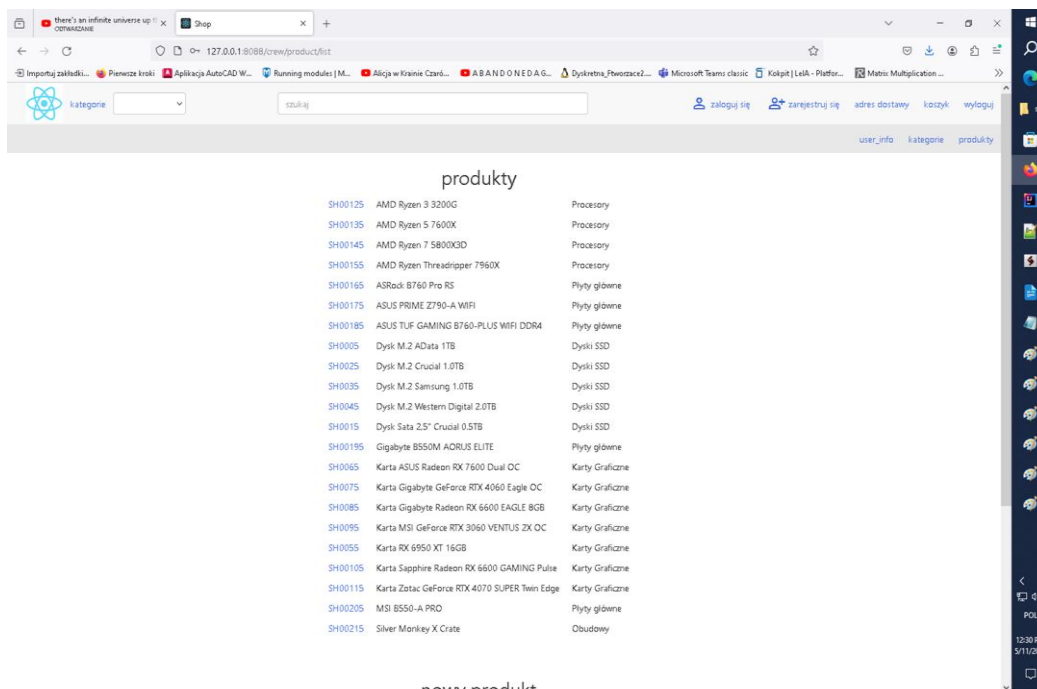
Prezentacja szczegółów produktu:



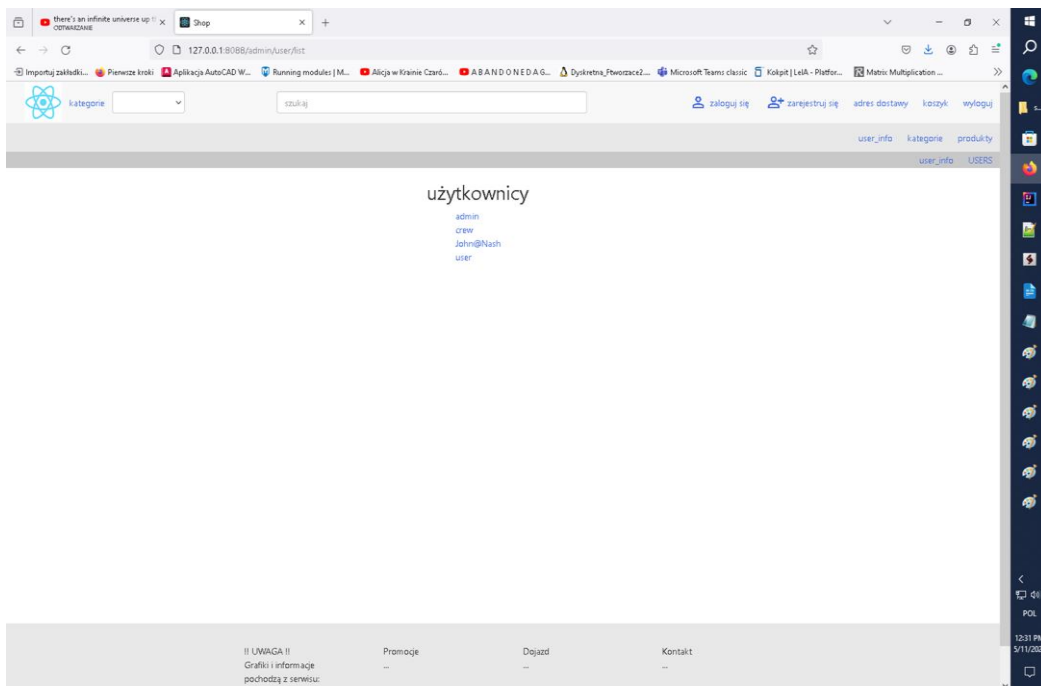
Podgląd koszyka:



Administracja – podgląd listy produktów w ofercie:



Administracja – lista użytkowników:



3. Dokumentacja projektowa

Podczas prac nad projektem prowadziliśmy szczegółową dokumentację projektową, która obejmowała:

- Rejestr produktowy z historią zmian,
- Rejestr sprintów,
- Opis wymagań funkcjonalnych programu.

Do modelowania przypadków użycia wykorzystywaliśmy narzędzie Modelio, co pozwoliło nam na dokładne przedstawienie i analizę funkcjonalności aplikacji.

Sklep z podzespołami elektronicznymi

Autorzy:

Status: roboczy

Wersja: 0.1

Data modyfikacji: 20.05.2024

Abstrakt: Dokument zawiera dokumentację metodyki, wymagań oraz projekt oprogramowania dla systemu Sklepu z podzespołami elektronicznymi.

Historia zmian:

| Wersja | Data | Kto | Opis |
|--------|------------|-----|--|
| 0.1 | 03.03.2024 | | Założenie dokumentu |
| 0.2 | 14.04.2024 | | Przygotowanie rejestru produktowego i sprintów |
| 0.3 | 21.04.2024 | | Sprint 1 |

3.1 Rejestr produktowy

User stories użytkowników: Administratora i Klienta.

User Stories dla Administratora

- Jako administrator, chcę móc logować się do panelu administracyjnego, aby zarządzać sklepem.
 - Kryteria akceptacji:
 - możliwość zalogowania się za pomocą nazwy użytkownika i hasła;
 - dostęp do panelu tylko po autoryzacji;
 - komunikat błędu przy nieudanej próbie logowania.
- Jako administrator, chcę zarządzać danymi produktów, aby aktualizować, dodawać lub usuwać produkty z oferty.
 - Kryteria akceptacji:
 - możliwość dodawania nowych produktów;
 - edycja istniejących produktów;
 - usuwanie produktów;
 - przeglądanie listy produktów.
- Jako administrator, chcę zarządzać czasowymi promocjami, aby przyciągać więcej klientów poprzez oferty specjalne.
 - Kryteria akceptacji:
 - tworzenie nowych promocji;
 - modyfikacja istniejących promocji;
 - usuwanie promocji;
 - przeglądanie aktywnych promocji.
- Jako administrator, chcę zarządzać danymi klientów, aby mieć aktualne i kompletne informacje o użytkownikach sklepu.
 - Kryteria akceptacji:
 - przeglądanie listy klientów i ich zakupów, generowanie raportów sprzedaży;
 - edycja danych klientów;
 - usuwanie klientów z systemu.

User Stories dla Klienta

- Jako klient, chcę się zalogować, aby mieć dostęp do mojego profilu i historii zakupów.
 - Kryteria akceptacji:
 - możliwość zalogowania się za pomocą e-maila i hasła;
 - przypomnienie zapomnianego hasła;
 - dostęp do profilu tylko dla zalogowanych użytkowników.
- Jako klient, chcę przeglądać listę produktów, aby wybrać i kupić te, które mnie interesują.
 - Kryteria akceptacji:
 - przeglądanie listy produktów z opcjami sortowania, wyszukiwania i filtrowania;
 - wyświetlanie szczegółowych informacji o produkcie;
 - możliwość dodania produktu do koszyka.
- Jako klient, chcę korzystać z koszyka, aby gromadzić produkty przed zakupem.
 - Kryteria akceptacji:
 - dodawanie produktów do koszyka;
 - modyfikacja ilości produktów w koszyku;
 - usuwanie produktów z koszyka;
 - wyświetlanie sumy ceny za wszystkie produkty w koszyku.
- Jako klient, chcę dokonać zakupu poprzez proces złożenia zamówienia.
 - Kryteria akceptacji:
 - złożenie zamówienia;
 - podanie danych do wysyłki;
 - wybór metody płatności;
 - potwierdzenie zamówienia i otrzymanie potwierdzenia mailem.

Sprint 1**Cel: konfiguracja środowiska pracy i stworzenie szkieletu aplikacji.**

| Przypadek użycia | Nazwa | Priorytet | Ilość godzin |
|------------------|--|---------------|--------------|
| PU01 | Przygotowanie środowiska deweloperskiego i narzędziowego dla zespołu | Bardzo Wysoki | 24 |
| PU02 | Projektowanie architektury systemu i bazy danych | Wysoki | 8 |
| PU03 | Logowanie administratora | Bardzo Wysoki | 8 |
| PU04 | Rejestracja i logowanie klienta | Wysoki | 8 |

Po realizacji **Sprintu 1** uzyskamy system logowania działający dla administratorów i klientów z podstawowym interfejsem użytkownika.

Sprint 2**Cel: implementacja podstawowych funkcji związanych z użytkownikami oraz wprowadzenie podstawowych funkcji zarządzania produktami.**

| Przypadek użycia | Nazwa | Priorytet | Ilość godzin |
|------------------|---|---------------|--------------|
| PU05 | Zarządzanie danymi produktów (CRUD): dodawanie,, wyświetlanie, usuwanie, edytowanie | Bardzo Wysoki | 16 |
| PU06 | Zarządzanie kontami użytkowników: usuwanie, edytowanie | Bardzo Wysoki | 12 |

Po realizacji **Sprintu 2** uzyskamy funkcje zarządzania produktami i użytkownikami z możliwością edycji i usuwania.

Sprint 3**Cel: rozwój funkcjonalności związanych z produktami.**

| Przypadek użycia | Nazwa | Priorytet | Ilość godzin |
|------------------|------------------------------|-----------|--------------|
| PU07 | Wyszukiwanie produktów | Wysoki | 16 |
| PU08 | Sortowanie produktów | Wysoki | 8 |
| PU09 | Filtrowanie produktów | Wysoki | 16 |
| PU10 | Szczegółowe widoki produktów | Wysoki | 8 |
| PU11 | Recenzje i oceny produktów | Średni | 12 |

Po realizacji **Sprintu 3** uzyskamy rozszerzone funkcje zarządzania produktami z wyszukiwaniem i filtrowaniem.

Sprint 4**Cel: implementacja koszyka zakupowego i procesu składania zamówień.**

| Przypadek użycia | Nazwa | Priorytet | Ilość godzin |
|------------------|--|-----------|--------------|
| PU12 | Dodawanie i usuwanie produktów do koszyka | Wysoki | 12 |
| PU13 | Modyfikacja ilości produktów w koszyku | Wysoki | 8 |
| PU14 | Składanie i realizacja zamówienia | Wysoki | 12 |
| PU15 | Dodanie metod płatności i integracja z zewnętrznymi systemami płatniczymi. | Wysoki | 20 |

Po realizacji **Sprintu 4** uzyskamy funkcjonalność koszyka zakupowego i procesu realizacji zamówienia.

Sprint 5

Cel: przygotowanie aplikacji do wdrożenia i uruchomienia.

| Przypadek użycia | Nazwa | Priorytet | Ilość godzin |
|------------------|--|-----------|--------------|
| PU16 | Generowanie raportów sprzedaży w panelu administratora | Średni | 12 |
| PU17 | Zarządzanie promocjami | Średni | 8 |
| PU18 | Zarządzanie danymi klientów | Niski | 8 |

Po realizacji **Sprintu 5** uzyskamy w pełni funkcjonalny sklep internetowy z możliwościami zarządzania, zakupów oraz raportowania.

3.2 Rejestr Sprintów

3.2.1 Rejestr Sprintu 1

| PU | Opis | h | P | W | Ś | C | Pt |
|------|--|---|---|---|---|---|----|
| PU01 | Spotkanie zespołowe, wybór szkieletu aplikacji, wybór rodzaju bazy danych, wybór IDE | 2 | | 2 | | | |
| | Spotkanie zespołu graficznego, omówienie wstępnej koncepcji graficznej | 8 | | 2 | 2 | 4 | |
| | Konfiguracja IntelliJ 2023.2.5 Community Edition dla każdego członka zespołu, instalacja środowisk (Spring Boot 3.2.3), przygotowanie obrazów, dystrybucja kluczy, ustanowienie repozytorium kodu | 8 | | 2 | 3 | 3 | |
| PU02 | Projektowanie modeli danych dla użytkowników i produktów, definiowanie relacji i schematu bazy danych; projektowanie ogólnej architektury aplikacji, w tym podziału na warstwy i główne komponenty systemu | 2 | | 2 | | | |
| | Implementacja klas encji dla użytkowników i produktów w Javie, konfiguracja Spring Data JPA dla operacji na bazie danych | 1 | | 1 | | | |

| | | | | | | |
|------|--|---|---|---|---|---|
| PU03 | Stworzenie podstawowego interfejsu użytkownika oraz struktury nawigacji – projektowanie wzoru <i>User Interface</i> przy użyciu HTML5, CSS3 i JavaScript | 3 | | 3 | | |
| | Implementacja podstawowej nawigacji po stronie aplikacji (menu, przeglądanie produktów, strona logowania / rejestracji) | 2 | 2 | | | |
| | Projektowanie i implementacja <i>User Interface</i> formularza logowania dla administratorów, przygotowanie warstwy frontend | 1 | 1 | | | |
| | Stworzenie mechanizmu autentykacji i sesji administratora, przygotowanie warstwy backend | 3 | | | 3 | |
| | Integracja systemu logowania z bazą danych H2 dla przechowywania danych o użytkownikach | 2 | 2 | | | |
| | Implementacja bezpiecznego przechowywania haseł | 1 | 1 | | | |
| | Testowanie funkcjonalności logowania | 1 | | | | 1 |
| PU04 | Projektowanie i implementacja <i>User Interface</i> formularza rejestracji i logowania dla klientów, przygotowanie warstwy frontend | 2 | 2 | | | |
| | Stworzenie mechanizmu rejestracji, autentykacji i sesji klienta, przygotowanie warstwy backend | 2 | 2 | | | |
| | Integracja systemu logowania z bazą danych H2 dla przechowywania danych o użytkownikach | 1 | 1 | | | |
| | Implementacja mechanizmów bezpieczeństwa | 2 | | 1 | 1 | |
| | Testowanie funkcjonalności logowania | 1 | | | | 1 |

3.2.2 Rejestr Sprintu 2

| PU | Opis | Ilość godzin | pn | wt | śr | czw | pt |
|------|---|--------------|----|----|----|-----|----|
| PU05 | Projektowanie i implementacja <i>User Interface</i> formularza dodawania, wyświetlania, edycji i usuwania produktów, przygotowanie warstwy frontend | 2 | | 2 | | | |
| | Tworzenie formularza w panelu administracyjnym do wprowadzania danych produktu (opcja „Dodaj produkt”), edycji produktu, dodanie opcji „Usuń” w interfejsie użytkownika panelu administracyjnego przy każdym produkcie, przygotowanie warstwy backend | 1 | | 1 | | | |
| | Integracja formularza z bazą danych | 6 | | | 3 | 3 | |
| | Zapewnienie walidacji danych na formularzu | 2 | | 2 | | | |
| | Implementacja potwierdzenia dodania, modyfikacji, usunięcia produktu | 1 | | 1 | | | |
| | Testy jednostkowe dla funkcji dodawania, wyświetlania, edycji i usuwania produktów, | 1 | | | | | 1 |

| | | | | | | | |
|------|---|---|--|---|---|---|---|
| PU06 | Projektowanie i implementacja <i>User Interface</i> do zarządzania kontami użytkowników | 2 | | 2 | | | |
| | Przygotowanie warstwy backend | 1 | | 1 | | | |
| | Integracja z bazą danych | 6 | | | 3 | 3 | |
| | Implementacja potwierdzenia modyfikacji, usunięcia konta | 2 | | 2 | | | |
| | Testy jednostkowe | 1 | | | | | 1 |

3.2.3 Rejestr Sprintu 3

| PU | Opis | Ilość godzin | pn | wt | śr | czw | pt |
|------|--|--------------|----|----|----|-----|----|
| PU07 | Projektowanie i implementacja <i>User Interface</i> do wyszukiwania produktów na podstawie nazwy, kategorii, ceny, dodanie opcji wyszukania w interfejsie użytkownika na stronie z listą produktów, przygotowanie warstwy frontend | 2 | | 2 | | | |
| | Przygotowanie warstwy backend | 1 | | 1 | | | |
| | Integracja funkcji wyszukiwania z bazą danych | 6 | | | 3 | 3 | |
| | Testy jednostkowe dla funkcji wyszukiwania | 1 | | | | | 1 |
| PU08 | Projektowanie i implementacja <i>User Interface</i> do sortowania produktów na podstawie nazwy, kategorii, ceny, dodanie opcji wyszukania w interfejsie użytkownika na stronie z listą produktów, przygotowanie warstwy frontend | 1 | | 1 | | | |
| | Przygotowanie warstwy backend | 6 | | | 3 | 3 | |
| | Integracja logiki sortowania z bazą danych | 2 | | 2 | | | |
| | Testy dla funkcji sortowania | 1 | | | | | 1 |
| PU09 | Projektowanie i implementacja <i>User Interface</i> do filtrowania produktów na podstawie nazwy, kategorii, ceny, dodanie opcji filtrowania w interfejsie użytkownika na stronie z listą produktów, przygotowanie warstwy frontend | 6 | | | 3 | 3 | |
| | Przygotowanie warstwy backend | 2 | | 2 | | | |
| | Integracja logiki filtrowania w bazie danych | 1 | | 1 | | | |
| | Testy dla funkcji filtrowania | 1 | | | | | 1 |
| PU10 | Projektowanie i implementacja szczegółowej strony produktu, przygotowanie warstwy frontend | 2 | | 2 | | | |
| | Przygotowanie warstwy backend | 1 | | 1 | | | |
| | Integracja z bazą danych dla wyświetlenia kompletnych informacji | 6 | | | 3 | 3 | |
| | Testy dla szczegółowego widoku produktu | 1 | | | | | 1 |
| PU11 | Projektowanie i implementacja <i>User Interface</i> do dodawania i wyświetlania recenzji i ocen, przygotowanie warstwy frontend | 1 | | 1 | | | |
| | Przygotowanie warstwy backend | 6 | | | 3 | 3 | |
| | Integracja systemu ocen i recenzji z bazą danych | 2 | | 2 | | | |

| | | | | | | | |
|--|-------|---|--|--|--|--|---|
| | Testy | 1 | | | | | 1 |
|--|-------|---|--|--|--|--|---|

3.2.4 Rejestr Sprintu 4

| PU | Opis | Ilość godzin | pn | wt | śr | czw | pt |
|------|--|--------------|----|----|----|-----|----|
| PU12 | Projektowanie i implementacja interfejsu dla dodawania i usuwania produktów z koszyka, przygotowanie warstwy frontend | 2 | | 2 | | | |
| | Przygotowanie warstwy backend | 1 | | 1 | | | |
| | Implementacja logiki zarządzania stanem koszyka na serwerze i w przeglądarce użytkownika | 6 | | | 3 | 3 | |
| | Testy jednostkowe i integracyjne | 1 | | | | | 1 |
| PU13 | Projektowanie i implementacja interfejsu dla modyfikacji ilości produktów w koszyku, przygotowanie warstwy frontend | 1 | | 1 | | | |
| | Przygotowanie warstwy backend | 6 | | | 3 | 3 | |
| | Implementacja mechanizmów aktualizacji ilości produktów w koszyku bez przeładowywania strony | 2 | | 2 | | | |
| | Testy jednostkowe i integracyjne | 1 | | | | | 1 |
| PU14 | Projektowanie i implementacja <i>User Interface</i> dla procesu składania zamówienia, przygotowanie strony podsumowania zamówienia, przygotowanie warstwy frontend | 6 | | | 3 | 3 | |
| | Przygotowanie warstwy backend | 2 | | 2 | | | |
| | Integracja z bazą danych (z systemem zarządzania stanem produktów dla aktualizacji dostępności po złożeniu zamówienia) | 1 | | 1 | | | |
| | Implementacja wysyłania potwierdzeń zamówienia na adres email klienta | 6 | | | 3 | 3 | |
| | Testy całego procesu zamówienia | 1 | | | | | 1 |
| PU15 | Projektowanie i implementacja interfejsu API dla obsługi różnych metod płatności, przygotowanie warstwy frontend | 1 | | 1 | | | |
| | Przygotowanie warstwy backend | 6 | | | 3 | 3 | |
| | Testy | 1 | | | | | 1 |

3.2.5 Rejestr Sprintu 5

| PU | Opis | Ilość godzin | pn | wt | śr | czw | pt |
|------|--|--------------|----|----|----|-----|----|
| PU16 | Projektowanie i implementacja <i>User Interface</i> do zbierania danych sprzedażowych i generowania raportów, przygotowanie warstwy frontend | 2 | | 2 | | | |
| | Przygotowanie warstwy backend | 1 | | 1 | | | |
| | Integracja funkcji raportowania z bazą danych | 6 | | | 2 | 2 | 2 |
| | Testy jednostkowe i integracyjne | 1 | | | | | 1 |
| PU17 | Projektowanie i implementacja <i>User Interface</i> do zarządzania promocjami, przygotowanie warstwy frontend | 1 | | 1 | | | |
| | Integracja logiki promocji z systemem sklepu i bazą danych (m.in. aktualizacja cen produktów w czasie promocji), przygotowanie warstwy backend | 6 | | | 3 | 3 | |
| | Implementacja walidacji czasu trwania promocji i automatyczne wyłączanie zakończonych promocji | 2 | | 2 | | | |
| | Testy jednostkowe i integracyjne | 1 | | | | | 1 |
| PU18 | Projektowanie i implementacja <i>User Interface</i> do zarządzania danymi klientów, przygotowanie warstwy frontend | 6 | | | 3 | 3 | |
| | Przygotowanie warstwy backend (przeglądanie, edycja danych klientów, usuwanie kont klientów) | 4 | | 2 | 1 | 1 | |
| | Integracja z bazą danych | 1 | | 1 | | | |
| | Testy jednostkowe i integracyjne | 1 | | | | | 1 |

3.3 Założenia – wykorzystane technologie i narzędzia

Środowisko: **Spring Boot 3.2.3**

Java: version 17

IDE: **IntelliJ 2023.2.5 Community Edition**

Baza danych: **środowiska developerskie: H2**

Do realizacji części backend wykorzystano: JAVA, Spring.

Spring

Nowoczesny framework do tworzenia aplikacji w języku Java. Posiada wbudowany konfigurowany automatycznie serwer webowy - Tomcat, wsparcie dla popularnych narzędzi, jak Hibernate zarządzający bazami danych zgodnie z ORM. Silnik szablonów Thymeleaf. Bardzo dobrze współpracuje z IntelliJ IDEA. Spring dostarcza wzorzec "Wstrzykiwanie zależności" (ang. Dependency Injection, DI), co jest sposobem osiągnięcia luźnych powiązań (ang. loose coupling). Dzięki DI otrzymujemy przejrzystą, elastyczną i łatwą do testowania architekturę aplikacji.

Do realizacji części frontend wykorzystano: HTML, Tailwind CSS oraz JavaScript. Poniżej znajduje się krótki opis tych technologii oraz narzędzia IntelliJ IDEA.

Technologie

HTML

HTML (HyperText Markup Language) to podstawowy język wykorzystywany do tworzenia stron internetowych. Użyłem go do zbudowania struktury naszej strony, definiując różne elementy takie jak nagłówki, akapity, obrazy i formularze.

Tailwind CSS

Tailwind CSS to framework CSS, który pozwala szybko i łatwo tworzyć nowoczesne interfejsy użytkownika. W przeciwieństwie do tradycyjnych frameworków, Tailwind nie dostarcza gotowych komponentów, lecz zestaw klas pomocniczych, które można bezpośrednio stosować w kodzie HTML. Dzięki temu tworzenie i modyfikowanie stylów jest bardzo elastyczne i szybkie.

Zalety Tailwind CSS:

- Elastyczność: umożliwia łatwe dostosowanie stylów do naszych potrzeb;
- Szybkość: szybkie prototypowanie dzięki gotowym klasom;
- Modularność: tworzenie modułowych stylów, które można wielokrotnie wykorzystywać.

JavaScript

JavaScript to język programowania używany do tworzenia dynamicznych i interaktywnych elementów na stronach internetowych. W naszym projekcie użyłem JavaScript do obsługi logiki interakcji użytkownika oraz manipulacji elementami DOM (Document Object Model).

Narzędzie - IntelliJ IDEA

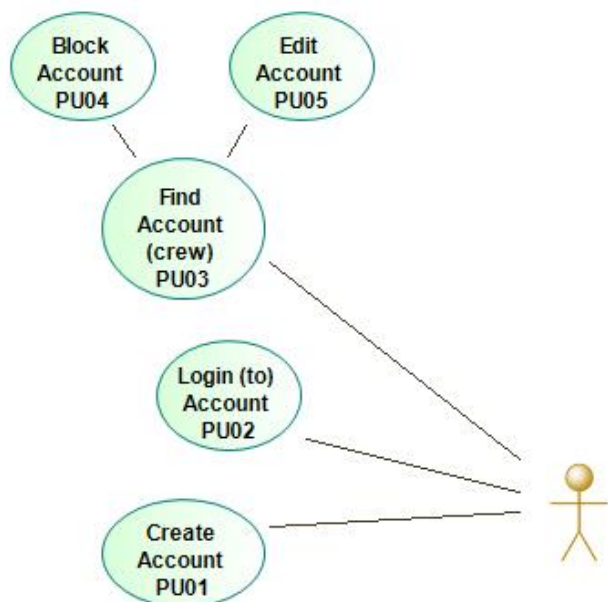
IntelliJ IDEA to zaawansowane narzędzie programistyczne (IDE - Integrated Development Environment) stworzone przez firmę JetBrains. Jest bardzo pomocne przy rozwijaniu aplikacji webowych, mobilnych i desktopowych. IntelliJ IDEA wspiera wiele języków programowania, w tym JavaScript, HTML i CSS, co sprawia, że jest idealnym narzędziem do pracy nad frontendem naszego projektu.

Główne funkcje IntelliJ IDEA:

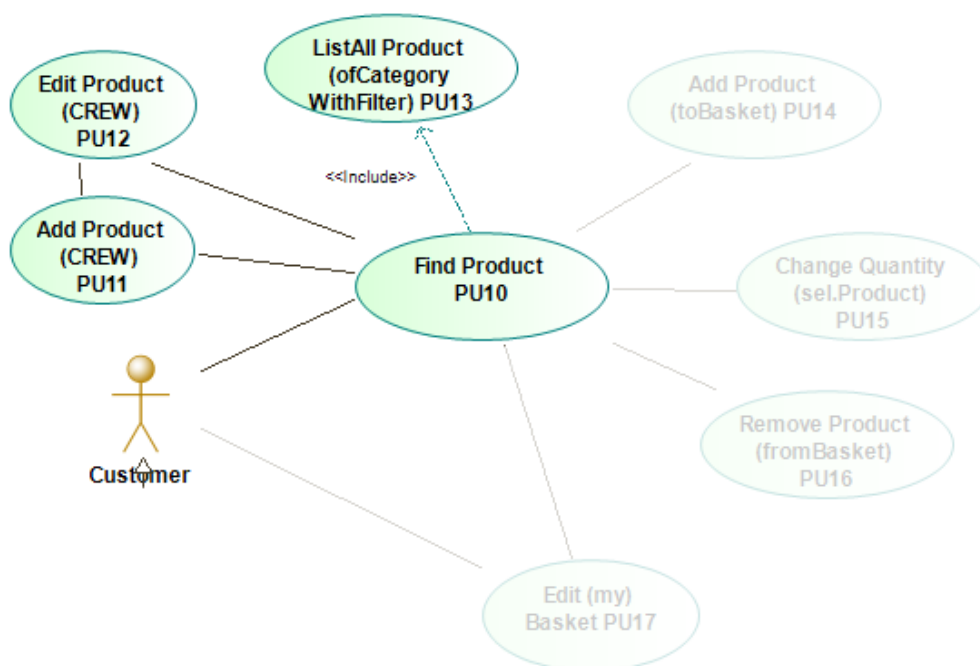
- Inteligentne podpowiedzi kodu: pomaga pisać kod szybciej i z mniejszą liczbą błędów;
- Wbudowany debugger: umożliwia śledzenie i naprawianie błędów w kodzie;
- Integracja z systemami kontroli wersji: wspiera Git, SVN i inne systemy kontroli wersji;
- Refaktoryzacja kodu: ułatwia modyfikowanie struktury kodu w bezpieczny sposób.

3.4 Opis wymagań funkcjonalnych systemu

3.4.1 Model przypadków użycia



Rysunek 1 : Diagram przypadków użycia dla Sprintu 1



Rysunek 1 : Diagram przypadków użycia dla Sprintu 1

4. Ograniczenia

4.1 Ograniczenia techniczne

4.1.1 Brak implementacji opcji płatności

Ze względu na to, że projekt zespołowy nie posiada jednoznacznego fizycznego położenia, zdecydowaliśmy, że nie będziemy implementować opcji płatności. Integracja systemów płatności wymaga zaawansowanej wiedzy z zakresu bezpieczeństwa oraz współpracy z zewnętrznymi dostawcami usług płatniczych, co wykraczało poza ramy naszego projektu.

4.1.2 Brak możliwości wysyłania maili

Zrezygnowaliśmy również z możliwości wysyłania maili przez aplikację. Integracja z serwerami SMTP oraz zarządzanie szablonami mailowymi stanowiłoby dodatkowe wyzwanie, na które nie mieliśmy wystarczających zasobów czasowych. Zamiast tego, postawiliśmy na prostsze formy komunikacji z użytkownikami, takie jak powiadomienia w aplikacji.

4.2 Zarządzanie użytkownikami

4.2.1 Zewnętrzne usługi autoryzacji

Zarządzanie procesem autoryzacji użytkowników to skomplikowane zadanie, które wymaga odpowiedniego poziomu bezpieczeństwa. Postanowiliśmy, że proces ten powinien być prowadzony przez zewnętrzną usługę, taką jak OAuth, która jest obecnie najpopularniejszym i najbezpieczniejszym rozwiązaniem. Implementacja własnego systemu autoryzacji mogłaby narazić aplikację na luki bezpieczeństwa.

4.2.2 Trudności z zarządzaniem użytkownikami i uprawnieniami

Zarządzanie użytkownikami i uprawnieniami było dla naszego zespołu wyzwaniem. Proces ten wymaga specjalistycznej wiedzy, ponieważ obejmuje obiekty, które podlegają ścisłemu nadzorowi przez wewnętrzne elementy systemu. Dostęp do tych obiektów wymagał określonych ścieżek postępowania, co znacznie skomplikowało implementację.

4.3 Ograniczenia czasowe i zasobowe

4.3.1 Ograniczony czas realizacji

Projekt miał ściśle określony czas realizacji, co zmusiło nas do priorytetyzowania zadań i skupienia się na kluczowych funkcjonalnościach. Niektóre zaawansowane funkcje, które mogłyby znacznie wzbogacić aplikację, musiały zostać odłożone na później lub całkowicie z nich zrezygnowano.

4.3.2 Ograniczone zasoby ludzkie

Nasz zespół składał się z trzech osób, co ograniczało nasze możliwości w zakresie równoczesnego prowadzenia wielu złożonych zadań. Każdy członek zespołu miał określone role i zadania, co wymagało efektywnej współpracy i zarządzania czasem, aby projekt mógł zostać ukończony zgodnie z założeniami.

4.4 Technologiczne wyzwania

4.4.1 Integracja z zewnętrznymi API

Integracja z zewnętrznymi API, takimi jak systemy płatności czy usługi wysyłania maili, wymaga zaawansowanej wiedzy i doświadczenia. W ramach naszego projektu skupiliśmy się na podstawowych funkcjonalnościach, które mogliśmy zrealizować w pełni samodzielnie, bez konieczności korzystania z zewnętrznych dostawców.

4.4.2 Złożoność architektury aplikacji

Budowa złożonej aplikacji sklepowej wiąże się z wieloma wyzwaniami architektonicznymi, takimi jak zapewnienie skalowalności, wydajności i bezpieczeństwa. Mimo że udało nam się stworzyć stabilną wersję aplikacji, niektóre zaawansowane aspekty, takie jak pełna optymalizacja wydajności, wymagają dalszej pracy i testów.

Podsumowując, mimo napotkanych ograniczeń, udało nam się stworzyć funkcjonalną i stabilną aplikację sklepową, która spełnia podstawowe wymagania użytkowników oraz dostarcza unikalną wartość w postaci kontaktu z rzeczoznawcą. Wspólna praca nad projektem pozwoliła nam zdobyć cenne doświadczenie, które z pewnością wykorzystamy w przyszłych przedsięwzięciach.

5. Testowanie aplikacji

Testowanie aplikacji przeprowadzaliśmy w ramach sprintów, oraz w trakcie procesu tworzenia, a ewentualne usterki naprawialiśmy na bieżąco. Zakładamy więc że wielu usterek w aplikacji nie ma. W razie gdyby użytkownicy zgłaszali problemy będziemy reagować najszybciej jak możliwe.

5.1 Metody testowania

5.1.1 Testy jednostkowe

Testy jednostkowe koncentrowały się na poszczególnych modułach aplikacji, sprawdzając poprawność działania pojedynczych funkcji i metod. Dzięki temu mogliśmy szybko zidentyfikować i naprawić błędy na poziomie kodu źródłowego.

5.1.2 Testy integracyjne

Testy integracyjne sprawdzały współdziałanie różnych modułów aplikacji. Ich celem było upewnienie się, że poszczególne części systemu poprawnie współpracują ze sobą, co jest kluczowe dla płynnego działania całej aplikacji.

5.1.3 Testy funkcjonalne

Testy funkcjonalne były przeprowadzane z perspektywy użytkownika końcowego. Skupiały się na sprawdzeniu, czy wszystkie funkcje aplikacji działają zgodnie z założeniami i czy interfejs użytkownika jest intuicyjny oraz łatwy w obsłudze.

5.2 Przebieg testowania

5.2.1 Testowanie podczas sprintów

Podczas każdego sprintu przeprowadzaliśmy testy w ramach zadań przypisanych do danego cyklu. Umożliwiało to bieżące wykrywanie i naprawianie błędów, co przyczyniało się do utrzymania wysokiej jakości aplikacji na każdym etapie jej rozwoju.

5.2.2 Testowanie regresyjne

Po wprowadzeniu nowych funkcji lub dokonaniu zmian w kodzie, przeprowadzaliśmy testy regresyjne. Miały one na celu upewnienie się, że nowe zmiany nie wprowadziły błędów w już istniejących i przetestowanych wcześniej funkcjonalnościach.

5.3 Wyniki testów i wnioski

Wyniki testów pokazały, że aplikacja spełnia założenia projektowe i działa zgodnie z oczekiwaniami. Większość wykrytych usterek została naprawiona na bieżąco, dzięki czemu finalna wersja aplikacji jest wolna od krytycznych błędów.

5.3.1 Usterki krytyczne

Żadne usterki krytyczne nie zostały zgłoszone po zakończeniu testów regresyjnych. Wszystkie wcześniej wykryte błędy zostały skutecznie naprawione.

5.3.2 Usterki mniej istotne

Nieliczne drobne usterki, które nie wpływały na kluczowe funkcjonalności aplikacji, zostały zidentyfikowane i naprawione w ramach późniejszych sprintów.

Testowanie aplikacji było procesem ciągłym, którego celem było zapewnienie wysokiej jakości produktu końcowego. Dzięki systematycznemu podejściu i wykorzystaniu nowoczesnych narzędzi testowych, udało nam się stworzyć aplikację, która jest stabilna, funkcjonalna i przyjazna dla użytkownika.

6. Kod źródłowy aplikacji

Przykładowa metoda w klasie kontrolera – strona backend:

```
@RequestMapping
(value = {"/crew/product/list/{categoryId}","/crew/product/list"}, method = Re-
questMethod.GET)
public String crewProductListGET( @PathVariable(required = false) Long catego-
ryId, Model model ){
    Category category=null;
    if ( categoryId!=null ) { Optional<Category> OC=categoryService.findById(
categoryId ); if (OC.isPresent()) { category=OC.get();} }
    prepareModelForProductList( model , category );
    return "product/crew_list";
}
```

Przykładowa metoda w klasie szablonu – strona frontend:

```
<body>
<h2>Kategorie</h2>
<th:block th:insert=~{fragments/form :: formS ( '/crew/product/edit' ) }"/>

<th:block th:insert=~{fragments/form :: hidden( id , ${product.id} ) }"/>
<th:block th:insert=~{fragments/form :: textRO( 'SKU' , ${product.SKU} , 'kod produktu', '' ) }"/>
<th:block th:insert=~{fragments/form :: text( 'name' , ${product.name} , 'nazwa produktu', '' ) }"/>
<th:block th:insert=~{fragments/form :: text( 'description' , ${product.description} , 'opis produk-
tu', '' ) }"/>

<!-- gallery -->
<th:block th:insert=~{fragments/form :: gallery( ${product.pictures} , 70 ) }"/>
<th:block th:insert=~{fragments/form :: formE }"/>
</body>
```