

Dokument Projektowy

Cel Projektu

Celem projektu jest stworzenie aplikacji ułatwiającej robienie zakupom osobom, które szukają okazji i zniżek. Chcemy sprawdzić oraz porównać ceny w konkurencyjnych sklepach, które uda nam się zagregować. Na podstawie stworzonej listy stwierdzimy gdzie konsument zapłaci najmniej za dane produkty. Dzięki temu nabywca dóbr będzie mógł wygodnie dowiedzieć się gdzie najkorzystniej będzie mu zaopatrzyć się w potrzebne rzeczy.

Projekt będzie aplikacją webową, która zawierać będzie moduły rejestracji, logowania użytkownika. Po zalogowaniu, użytkownik dostaje dostęp do bazy danych produktów, których ceny chce sprawdzić oraz panel do filtrowania wyników. Ponadto autoryzowany użytkownik będzie miał możliwość edytowania swoich danych logowania (takich jak nazwa użytkownika lub hasło) i w ostateczności usunięcia swojego konta.

Motywacja

Stworzenie aplikacji ułatwiającej łowienie okazji na produkty codziennego użytku, z punktu widzenia budżetu studenckiego bardzo pożądana. Na rynku istnieje wiele aplikacji i stron pozwalających na porównywanie cen w sklepach aczkolwiek żadna z nich nie skupia się stricte w większym stopniu na części spożywczej. Jedyna konkurencja to Pepper, jednak opiera się na zgłoszeniach użytkowników i nie tworzy najbardziej korzystnej listy zakupów oraz w jakim sklepie można zrobić takie zakupy.

Założenia Projektu

Metodologią naszej pracy jest SCRUM, który należy do zwinnych podejść do tworzenia projektów (Agile project management). Wszystkie te ramy postępowania zwiększają szanse powodzenia projektu poprzez zdolność adaptacji do wszelkich zmian. Przyjęcie zwinnego podejścia pozwoli zachować dużą elastyczność i wygodę członkom zespołu w doborze zadań jak i czasie ich wykonania. Każdy z członków zespołu posiada dużą autonomię – jednak w przypadku sporów, problemów lider ma za zadanie kierować pracą zespołu. Podstawową metodyką pracy będą sprinty rozwojowe. Są to krótkie cykle iteracyjne, w których dany projekt jest rozwijany lub ulepszany. Zwykle trwają od 14 do 20 dni i zaczynają się od spotkań planistycznych, gdzie ustala się następne cele. Zespół decyduje o obciążeniu jakie może na siebie nałożyć. W pierwszej kolejności skupiamy się na rzeczach obligatoryjnych i stopniowo ulepszamy działające już funkcje/cechy/wizualia. Na każdym etapie produkcyjnym staramy zachować porządek w kodzie/plikach aby ułatwić pracę innym członkom zespołu z już stworzonymi elementami projektu.

Istnieją również narzędzia ułatwiające pracę w systemie scrumowym. Jednym z wyróżnionych pomysłów były jednak tablice Kanban. Stanowiły to nowatorski sposób na pozyskiwanie i aktualizowanie cennej wiedzy na jakim etapie są postępy w projekcie. Formy elektroniczne powoli wypierają klasyczne podejście, gdzie na tablicy znajdowały się zadania podzielone na małych kartkach na grupy: do zrobienia, w trakcie pracy, zrobione. Odpowiednie dla naszego projektu funkcjonalności zapewni Jira.

Jira – potężne narzędzie do zarządzania pracą we wszelkich zastosowaniach — od zarządzania przypadkami testowymi po rozwój oprogramowania z wykorzystaniem metodyki Agile. Dzięki Jirze możemy wspólnie planować i wykonywać zadania, przedstawiać nowe pomysły, oraz dzielić się swoją wiedzą z innymi. Pomocna jest również w udostępnianiu zasobów, informowaniu o statusie zadań i etapach realizacji projektu. Jira pomaga także określić ramy współpracy w zespole.

Git – rozproszony system kontroli wersji. Oprogramowanie dedykowane programistom. Z jego pomocą niezależnie od miejsca, bez połączenia z siecią mogą zapisywać zmiany, a także wymieniać je pomiędzy lokalnymi repozytoriami i gałęziami (branch). W praktyce oznacza to możliwość równoczesnej pracy programistów nad projektem. Wspólny dostęp do kodu i możliwość pracy kilku osób jednocześnie nad tym samym projektem to wsparcie dla jego efektywnej realizacji. Wykorzystywanym przez nas serwisem umieszczania repozytorium gita będzie Github. GitHub to usługa hostingu umożliwiająca zarządzanie repozytoriami Git. Przy jego pomocy będziemy w stanie udostępnić swój kod w jednym miejscu dla wszystkich. Dzięki temu – w tym samym czasie – zapewniona zostanie możliwość aktywnej współpracy z pozostałymi członkami projektu.

Środowiska

Pycharm - IDE stworzone przez firmę JetBrains stosowane głównie do programowania w języku Python. Dostarcza narzędzia do analizy kodu, graficznego debugowania i zintegrowanych testów jednostkowych.

Dbeaver - IDE stworzone do tworzenia, administracji i zarządzania bazami danych.

Pozostałe środowiska które mogą się okazać konieczne wyjdą później.

Technologie

Web scrapping - to technika wyodrębniania danych ze stron internetowych, która zastępuje ręczne, powtarzalne wpisywanie lub kopiowanie i wklejanie. Dodatkowo pozyskane dane często są przechowywane w ustrukturyzowanym formacie umożliwiając szybkie ich przetwarzanie. W naszym przypadku web scrapping zostanie użyty w celu zbudowania bazy danych oraz pomocny będzie w przypadku konieczności jej aktualizacji.

Python – język programowania wysokiego poziomu ogólnego przeznaczenia, o rozbudowanym pakiecie bibliotek standardowych, którego ideą przewodnią jest czytelność i klarowność kodu źródłowego. Jego składnia cechuje się przejrzystością i zwięzłością.

Python wspiera różne paradygmaty programowania: obiektowy, imperatywny oraz w mniejszym stopniu funkcyjny. Posiada w pełni dynamiczny system typów i automatyczne zarządzanie pamięcią. Wybraliśmy go ponieważ Python oferuje dużą ilość bibliotek które będą pomocne w realizacji zadań.

Flask – mikro framework aplikacji webowych napisany w języku Python. Jest sklasyfikowany jako micro-framework, ponieważ nie wymaga określonych narzędzi ani bibliotek. Nie ma warstwy abstrakcji bazy danych, sprawdzania poprawności formularzy ani żadnych innych komponentów, w których istniejące biblioteki stron trzecich zapewniają wspólne funkcje.

Świetnie nadaje się do małych projektów tworzonych przez jednego lub małą ilość programistów jak ma to miejsce w naszym przypadku. Ponadto jest bardzo intuicyjny.

Front-end:

- **HTML5** - język znaczników wykorzystywany do tworzenia i prezentowania stron internetowych www. Wykorzystanie strony internetowej pozwoli na łatwiejszy dostęp ze względu na brak wymogu instalowania aplikacji.
- **CSS3** - język służący do opisu formy stron www. CSS został stworzony do odseparowania struktury dokumentu od formy jego prezentacji. Stosowanie zewnętrznych arkuszy CSS daje możliwość zmiany wyglądu wielu stron naraz bez ingerowania w sam kod HTML, ponieważ arkusze mogą być wspólne dla wielu dokumentów.

Wykorzystujemy HTML5 i CSS3 ponieważ nasz projekt jest stroną internetową

Bazy danych:

- **Relacyjne bazy danych** - pozwala na przechowywanie powiązanych ze sobą danych i łatwy do nich dostęp. Baza danych jest oparta na modelu relacyjnym który jest prosty i intuicyjny w prezentowaniu ich w tablicach. Pozwala to na szybkie przetwarzanie i wyświetlanie odpowiednich danych dla użytkownika
- **SQLite** - to darmowy open-source system zarządzania danymi pozwalający na wydajną pracę na stronach internetowych ze względu na jego lekkość wynikającą z wymagania małej ilości zasobów
- **SQLAlchemy** - open-source biblioteka SQL dla pythona służąca do pracy z danymi i mapowania obiektowo-relacyjnego. Obsługiwana przez MySQL i SQLite3, które Wykorzystujemy

Deployment:

- **Heroku** - platforma chmurowa działająca w modelu PaaS (platform as a service). Domyślnym systemem operacyjnym wykorzystywanym przez Heroku jest Debian Linux platforma wspiera cały szereg najchętniej wykorzystywanych przez twórców aplikacji webowych języków. Można wśród nich wymienić między innymi takie języki, jak Java, JavaScript (Node.js), Scala, Clojure, Python oraz PHP. Możliwości są więc naprawdę duże, stąd też spore zainteresowanie ze strony programistów. Aplikacje webowe stworzone za pomocą tej platformy można wdrażać między innymi za pomocą Git Huba, ale nie tylko.

Głównym zasobem który wykorzystujemy będzie serwer na Heroku, cały deployment będzie zrobiony w tym miejscu. Będziemy mieć postawioną tam zarówno bazę jak i aplikację.

Najczęściej wykonywane zapytanie:

```
SELECT NAZWA, MARKA, CENA, SKLEP FROM PRODUKT  
WHERE PRODUKT.NAZWA = '';
```

```
SELECT EMAIL FROM UZYTKOWNIK  
WHERE UZYTKOWNIK.EMAIL = '';
```

```
SELECT HASLO FROM UZYTKOWNIK;
```

Sposób optymalizacji:

-nie używamy join

-optymalne rozmieszczenie kolumn

-optymalizacja zapytań:

-ograniczenie liczby kolumn do wyświetlenia w instrukcji select

-filtrowanie danych poprzez zastosowanie klauzuli WHERE

-nie powinno się stosować porządkowania danych,
jeżeli nie jest to konieczne (order by)

-trzeba unikać zagnieżdżonych zapytań i klauzuli GROUP BY

SCHEMAT BAZY DANYCH

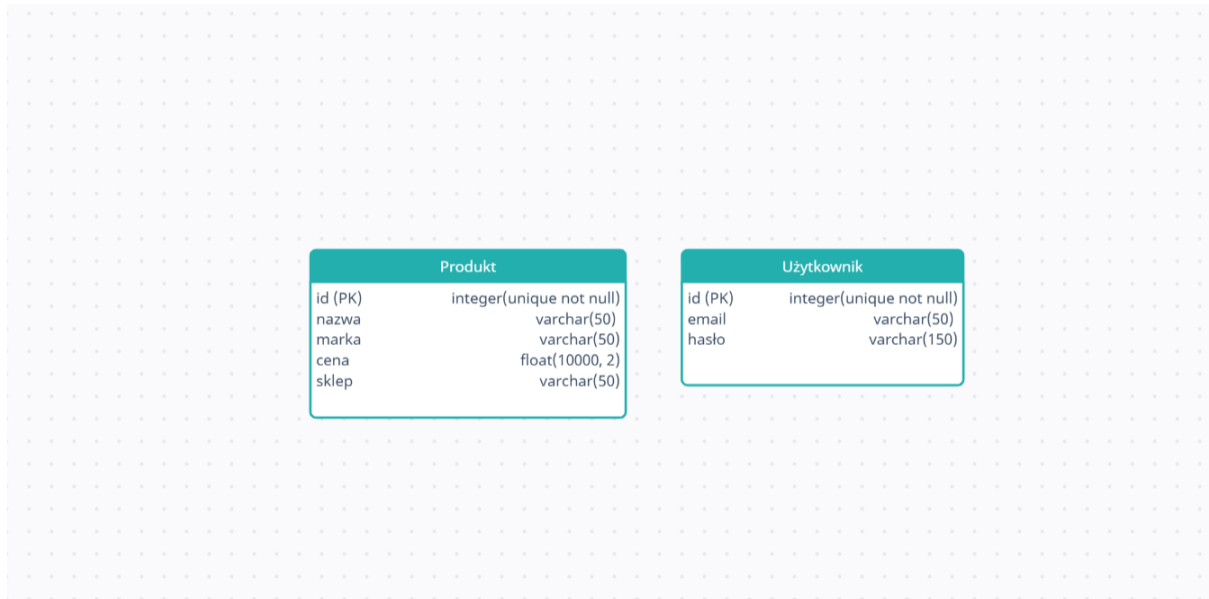


Diagram przypadków użycia

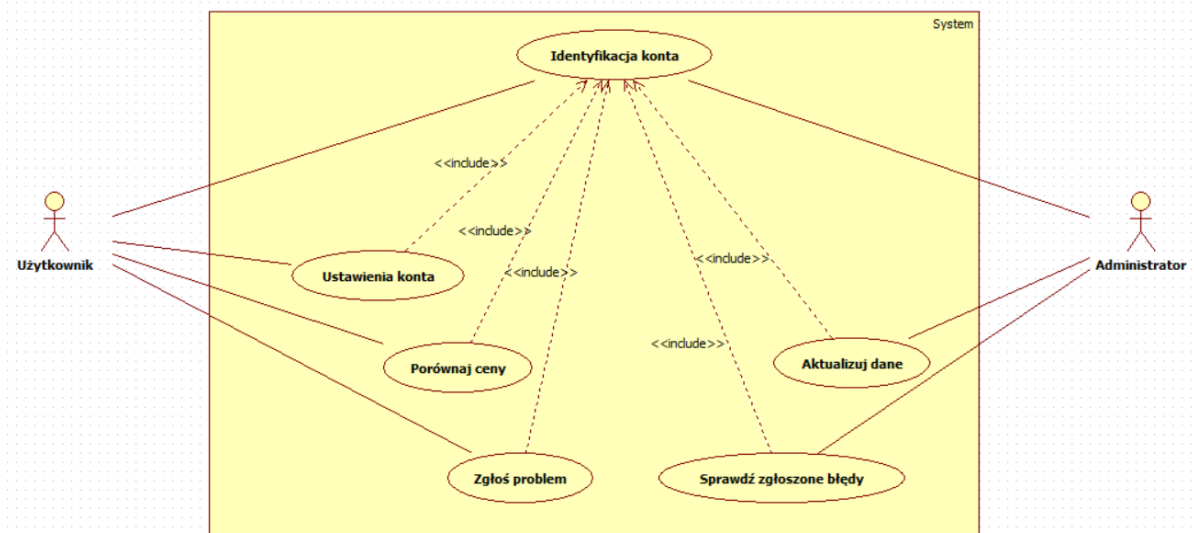


Diagram sekwencji

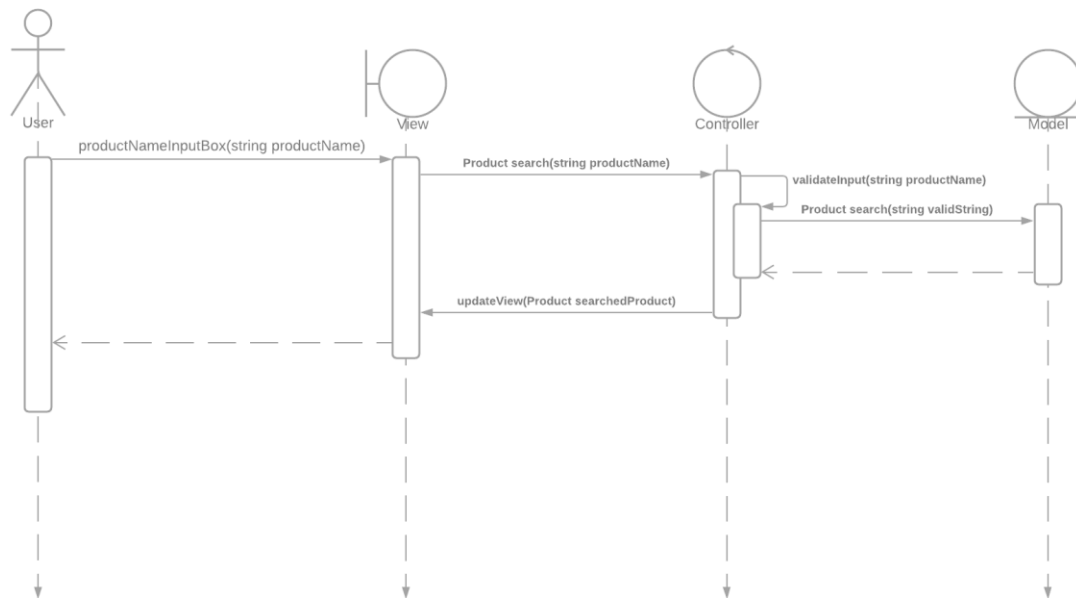


Diagram czynności

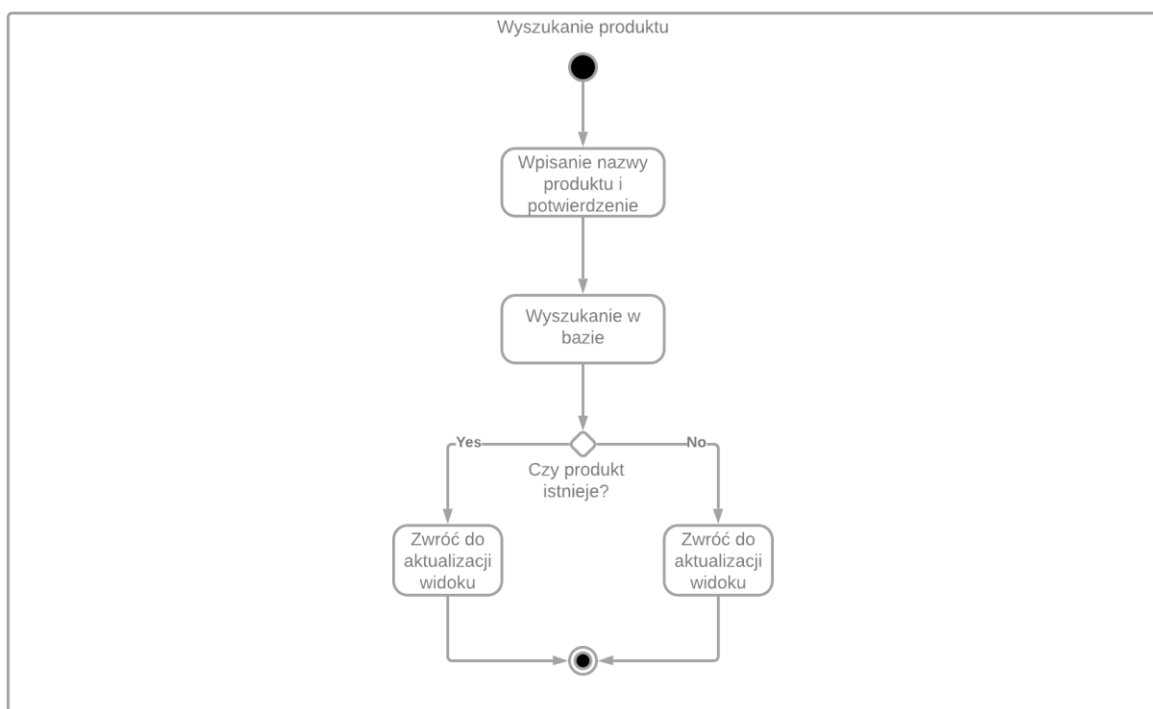


Diagram stanu User

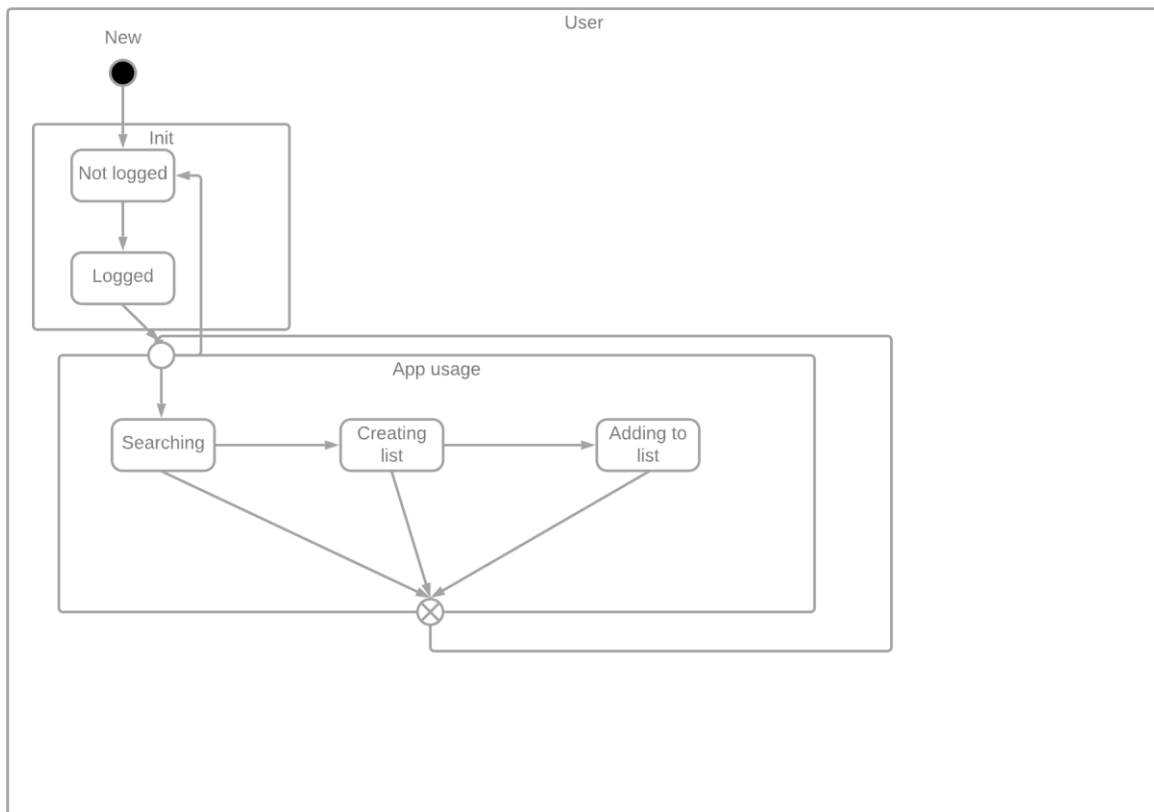
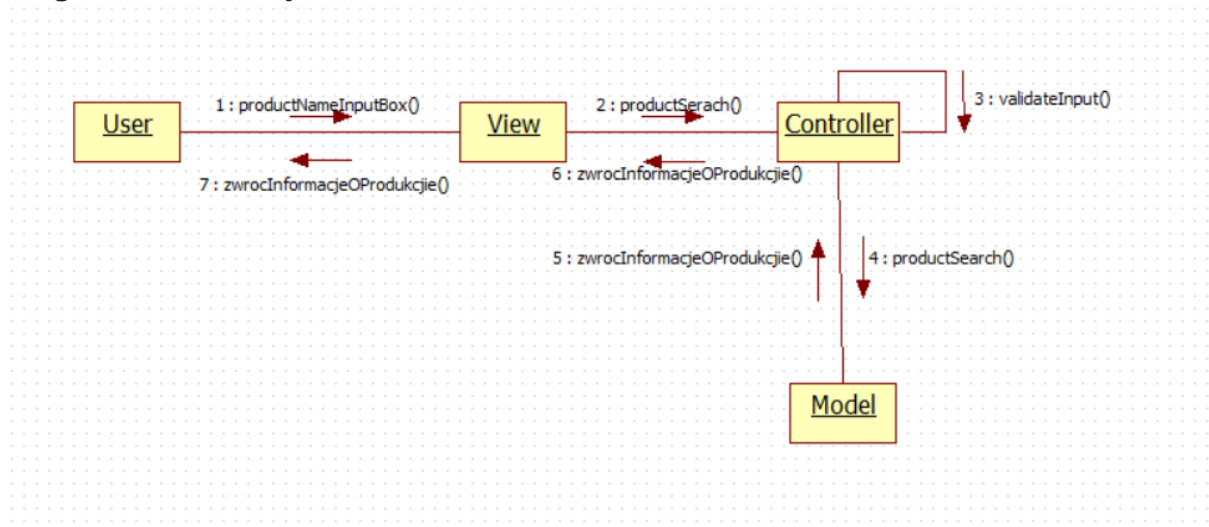
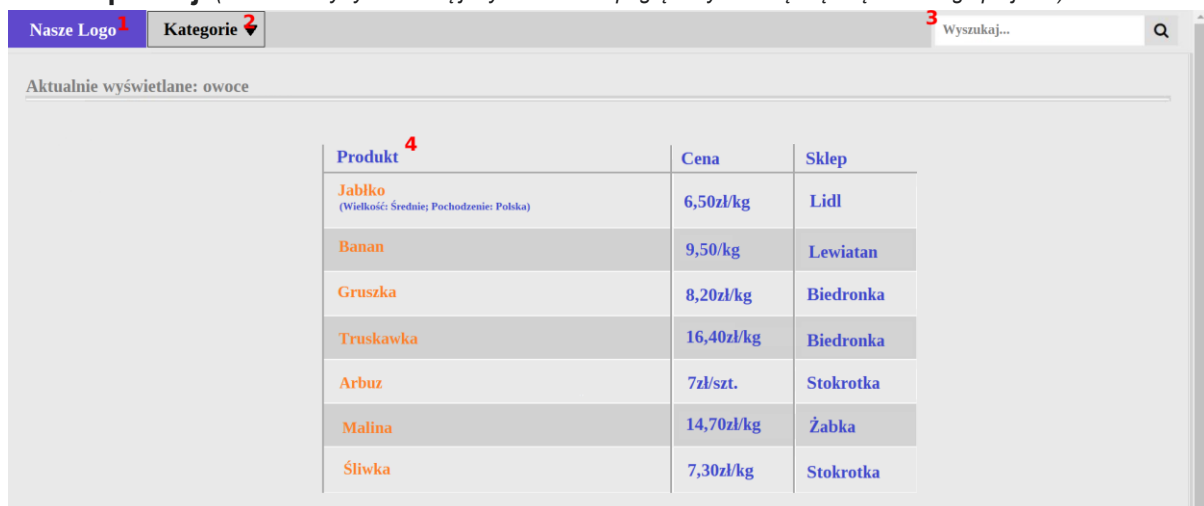


Diagram komunikacji



Szkic aplikacji (czerwone cyfry stanowią jedynie element poglądowy i nie są częścią finalowego projektu)



Funkcje interfejsu (od lewej):

1. Kliknięcie w logo przenosi do strony głównej
2. Zakładka "Kategorie" pozwala na zaaplikowanie filtrów do wyświetlanych produktów
3. Pasek wyszukiwań pozwala wyszukać konkretny produkt
4. Kliknięcie w "Produkt", "Cena", "Sklep" posortuje daną kolumnę alfabetycznie/cenowo, w zależności od jej zawartości

Scenariusze testowe

| Id | Nazwa | Opis | Typ | Czynności | Rezultat |
|----|-----------------------------------|---|--------------------|--|---|
| 1 | Przeniesienie do strony startowej | Sprawdzenie czy zostaniemy przeniesieni do strony startowej | Testy funkcjonalne | 1.Sprawdzenie poprawności ścieżki URL | Przeniesienie do strony startowej |
| 2 | Aplikowanie filtrów | Wyświetlanie wyników z uwzględnieniem filtrów | Testy funkcjonalne | 1.Wybranie filtrów 2.Sprawdzenie poprawnego filtrowania | Wyświetlenie listy z zaaplikowanymi filtrami |
| 3 | Wyszukiwanie | Sprawdzenie funkcjonalności paska wyszukiwania | Testy funkcjonalne | 1.Wpisanie nazwy produktu | Wyświetlenie odpowiednich produktów |
| 4 | Sortowanie | Sprawdzenie funkcjonalności sortowania kolumn | Testy funkcjonalne | 1. Sprawdzenie poprawności sortowania | Wyświetlenie zawartości kolumny z odpowiednim sortowaniem |

Spis członków:

- Marta Malinowska - Lider zespołu/Inżynier
 - Organizacja pracy
 - Baza danych
- Jakub Markowski - Inżynier
 - Opracowanie schematów
 - Programowanie
- Dominik Lis - Inżynier
 - Programowanie
- Jakub Grusiewicz - Inżynier
 - Organizacja pracy
 - Opracowanie schematów
 - Wsparcie techniczne
- Szymon Chłopek - Tester
 - Testowanie oprogramowania
 - Wsparcie techniczne

Link do JIRA:

<https://dynamicshoppinglist.atlassian.net/jira/software/projects/DREAM/boards/1>