

System obsługi sklepu internetowego

Spis treści

1.	Dziedzina problemowa	1
2.	Cel	1
3.	Zakres odpowiedzialności systemu	1
4.	Użytkownicy systemu	2
5.	Wymagania użytkownika	2
6.	Wymagania funkcjonalne	4
7.	Opis struktury systemu	5
8.	Analiza dynamiczna przypadku użycia „Zakup produkt”	6
9.	Analiza dynamiczna klasy „Payment”	9
10.	Decyzje projektowe	9
11.	Schemat logiczny systemu	10
12.	Projekt interfejsu użytkownika	11
13.	Wymagania niefunkcjonalne	16
14.	Słownik	17

1. Dziedzina problemowa

System ten jest lekką i prostą implementacją sklepu online. Jest dobrym narzędziem do rozpoczęcia działalności komercyjnej w Internecie

2. Cel

System jest stworzony dla początkującego e-sprzedawcy chcącego zbudować własną stronę sprzedaży online ponosząc niskie koszty, a co za tym idzie, niskie ryzyko inwestycji w sklep internetowy.

3. Zakres odpowiedzialności systemu

System ma umożliwiać przeglądanie oferty sklepu internetowego oraz dokonywanie zakupu i jego ewentualny zwrot. System ma również mieć możliwość łatwego zarządzania stroną poprzez panel Admina dostępny po zalogowaniu na konto admina. W ten sposób właściciel sklepu nie musi znać się na bazach danych żeby uzupełniać informacje o produktach.

4. Użytkownicy systemu

Z systemu korzystać będą:

1. Administrator
2. Gość
3. Klient

5. Wymagania użytkownika

1. W systemie należy przechowywać Informacje o użytkownikach systemu. Są nimi Gość, Klient, i Admin. Dla Gościa nie należy przechowywać żadnych danych.
2. Klient jest zarejestrowanym użytkownikiem i dla niego należy przechowywać jego dane do wysyłki, nazwę konta (jednocześnie login do logowania się na konto), email i hasło. Klient ma dostęp do listy życzeń. Na liście życzeń przechowywane są produkty.
3. Admin jest administratorem strony. Dla Admina należy przechować login i hasło.
4. System ma przechowywać dane na temat produktu: id, nazwa, cena, obecna zniżka, ilość w magazynie, kategorię i oceny produktu (w skali 1-5 wraz z komentarzem) oraz średnią ocenę klientów.
5. Dane do wysyłki to: Imię, nazwisko, numer telefonu, email, adres zamieszkania. Adres zamieszkania składa się z Nazwy Ulicy, numeru mieszkania (jeśli jest), miasta, kodu pocztowego i państwa.
6. W koszyku należy przechowywać dodane produkty oraz ich ilość.
7. Zamówienie budowane jest w oparciu o przedmioty w koszyku. Przechowuje więc listę przedmiotów i ich ilości, unikalny numer zamówienia, metodę dostawy i status zamówienia. Statusy zamówienia są następujące: W trakcie, oczekiwanie na płatność, oczekiwanie na dostarczenie, Dostarczono, Anulowano, Zwrócono.
8. Metoda dostawy powinna zawierać id, nazwę metody dostawy, cenę, dane do wysyłki i przewidywany termin dostawy.
9. Zamówienie zawiera również opłatę za nie. Opłata zawiera informację o całkowitym koszcie zamówienia wraz z dostawą, metodę płatności oraz status płatności. Status płatności może być jednym z: W trakcie, ukończony, nieudany, anulowany
10. Zamówienie jest możliwe do zwrócenia. Zwrot musi zawierać id, listę przedmiotów zwracanych, powód zwrotu i jego status; w trakcie, zwrócone, odrzucone. Można dokonać więcej niż jeden zwrot dla danego zamówienia.

System ma umożliwić wykonywanie następujących czynności:

1. Założenie konta (Gość)
2. Zalogowanie się na konto (Gość)
3. Dodanie przedmiotu do koszyka (Gość, Klient)
4. Usunięcie przedmiotu z koszyka (Gość, Klient)
5. Zobaczenie koszyka (Gość, Klient)
6. Złożenie zamówienia (Gość, Klient)
7. Dokonanie płatności (Gość, Klient)
8. Wykonanie zwrotu (Gość, Klient)
9. Wylogowanie się (Klient)
10. Przypisanie danych do wysyłki do konta (Klient)

11. Dodanie produktu do listy życzeń (Klient)
12. Usunięcie produktu z listy życzeń (Klient)
13. Dodanie oceny produktu (Klient)
14. Dodanie nowego produktu (Admin)
15. Aktualizacja dane produktu (Admin)
16. Usunięcie produktu (Admin)
17. Dodanie nowej kategorii (Admin)
18. Usunięcie kategorii (Admin)
19. Usunięcie oceny produktu (Admin)
20. Wysyłanie informacji o aktualnej promocji (Podsystem czasu)
21. Procesowanie płatności przez zewnętrzny system (Zewnętrzny system płatności)

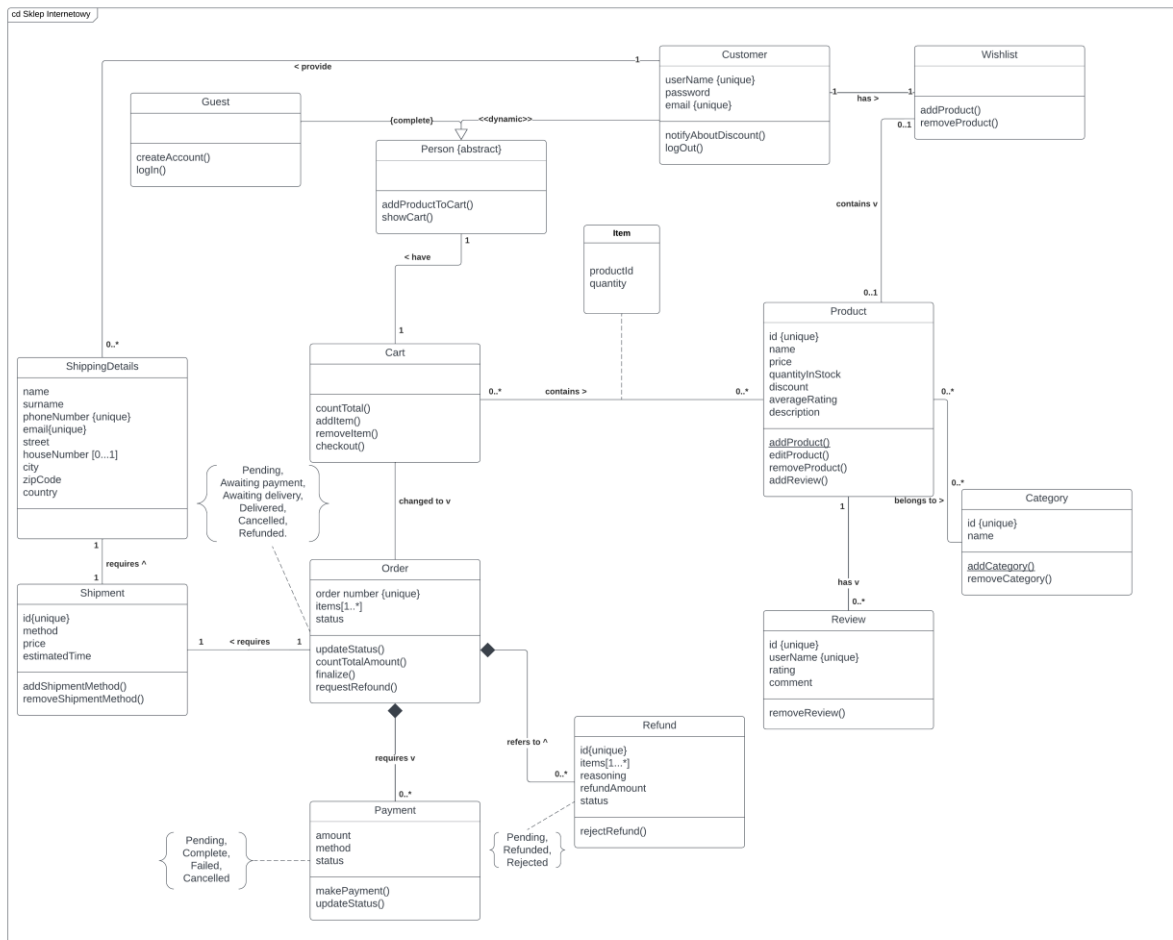
Wobec systemu stawiane są określone wymagania:

1. Dostępność systemu
2. System ma przechowywać dane w konkretnym systemie bazy danych
3. Wprowadzane dane muszą być spójne i kompletne

6. Wymagania funkcjonalne



7. Opis struktury systemu



8. Analiza dynamiczna przypadku użycia „Zakup produkt”

Scenariusz przypadku użycia

Warunek początkowy: W koszyku jest przynajmniej jeden przedmiot

1. Przypadek użycia uruchamia się, gdy użytkownik Person wybiera opcję zamów w koszyku.
2. System uruchamia przypadek użycia „złóż zamówienie”
 - 3a. System prosi o uzupełnienie danych do wysyłki.
 - 3b. System automatycznie wypełnia dane do wysyłki na podstawie danych Klienta
4. System prosi o wybranie sposobu dostawy.
5. System prosi o wybranie sposobu płatności
6. System wyświetla finalną cenę usługi i oczekuje przejścia do płatności przez użytkownika
7. System uruchamia przypadek użycia dokonaj płatności
8. System przekierowuje procesowanie płatności do zewnętrznego systemu płatności
 - 9a. System informuje użytkownika o zakończonym procesie i kończy przypadek użycia.
 - 9b. System informuje o niepowodzeniu i proponuje ponowienie płatności

Diagram aktywności

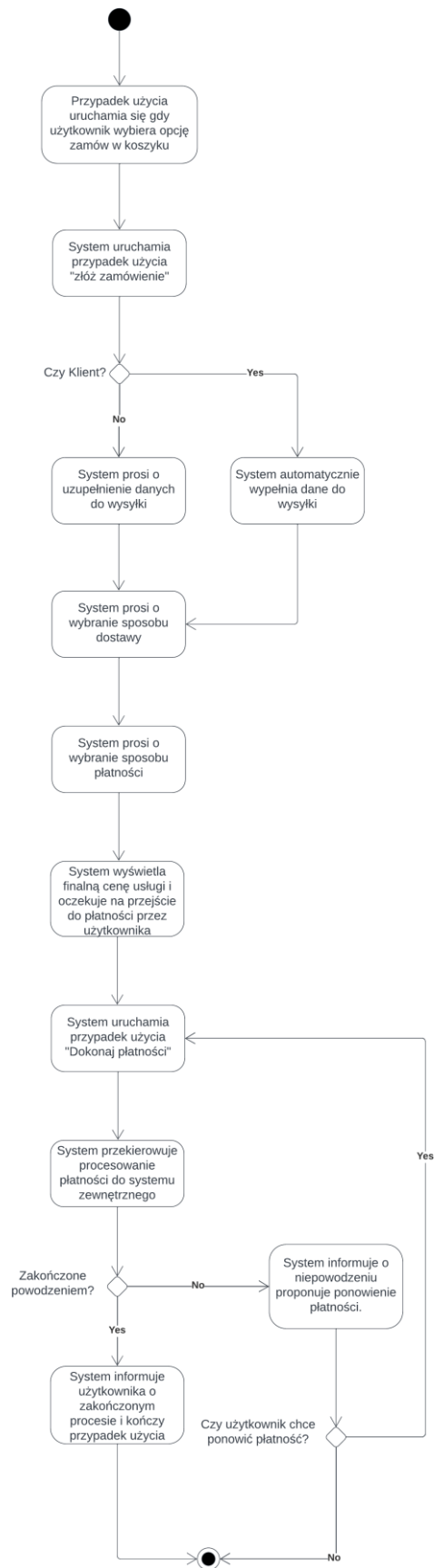
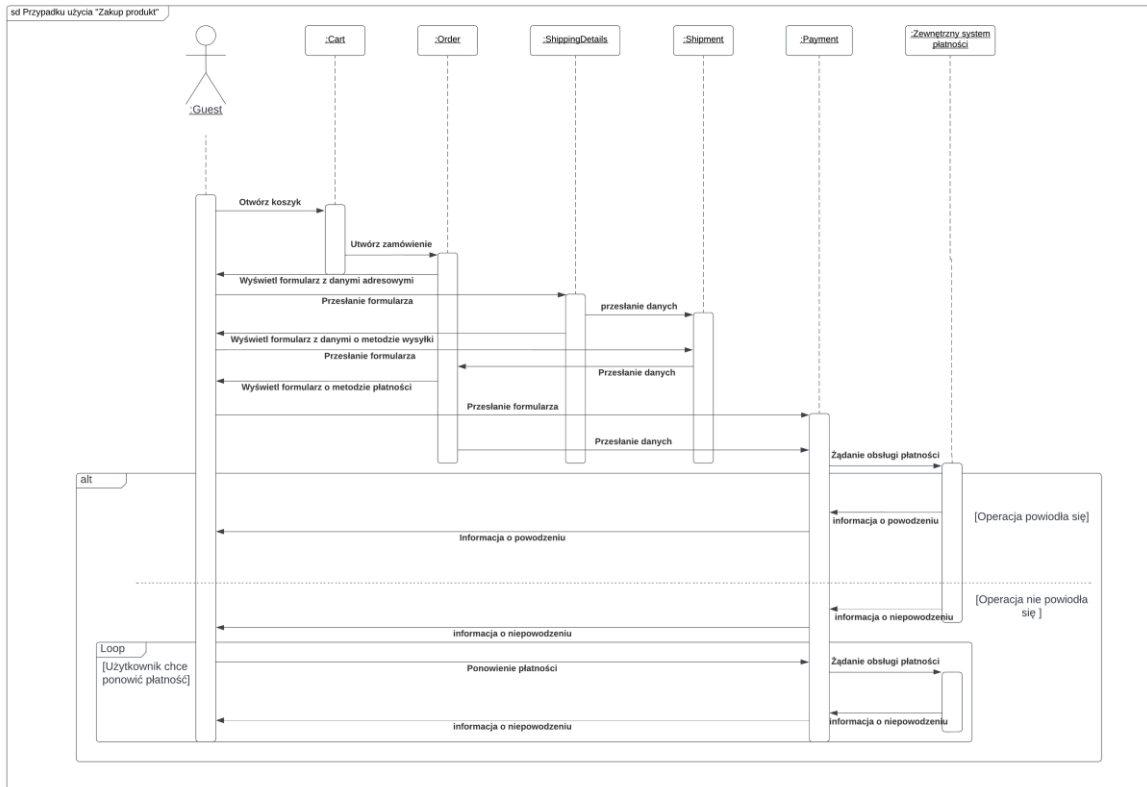
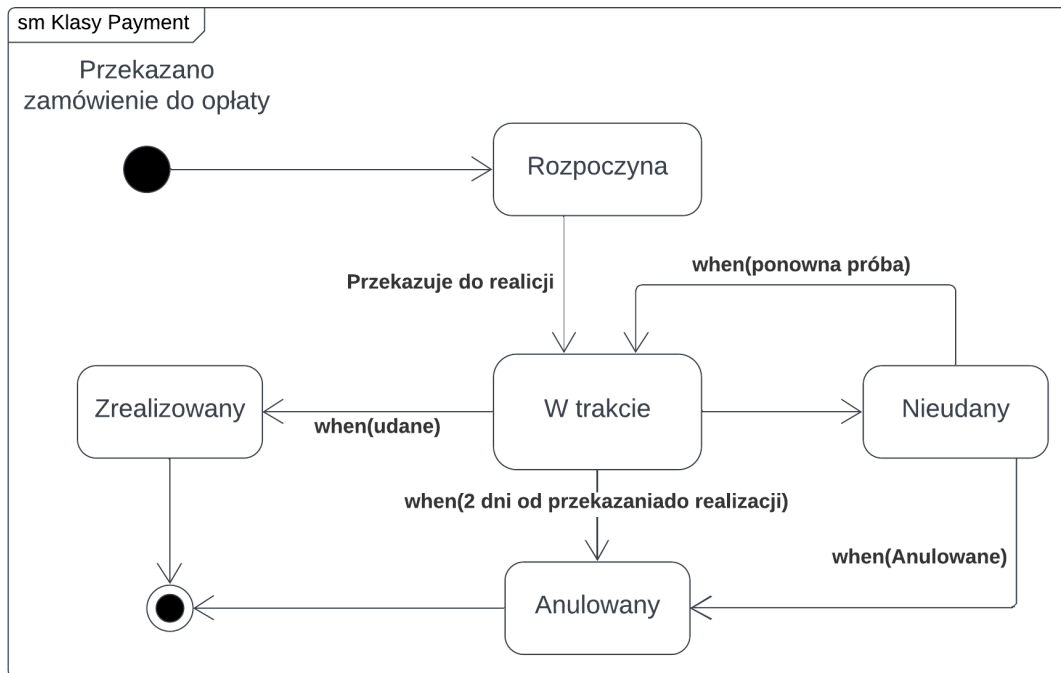


Diagram sekwencji



9. Analiza dynamiczna klasy „Payment”

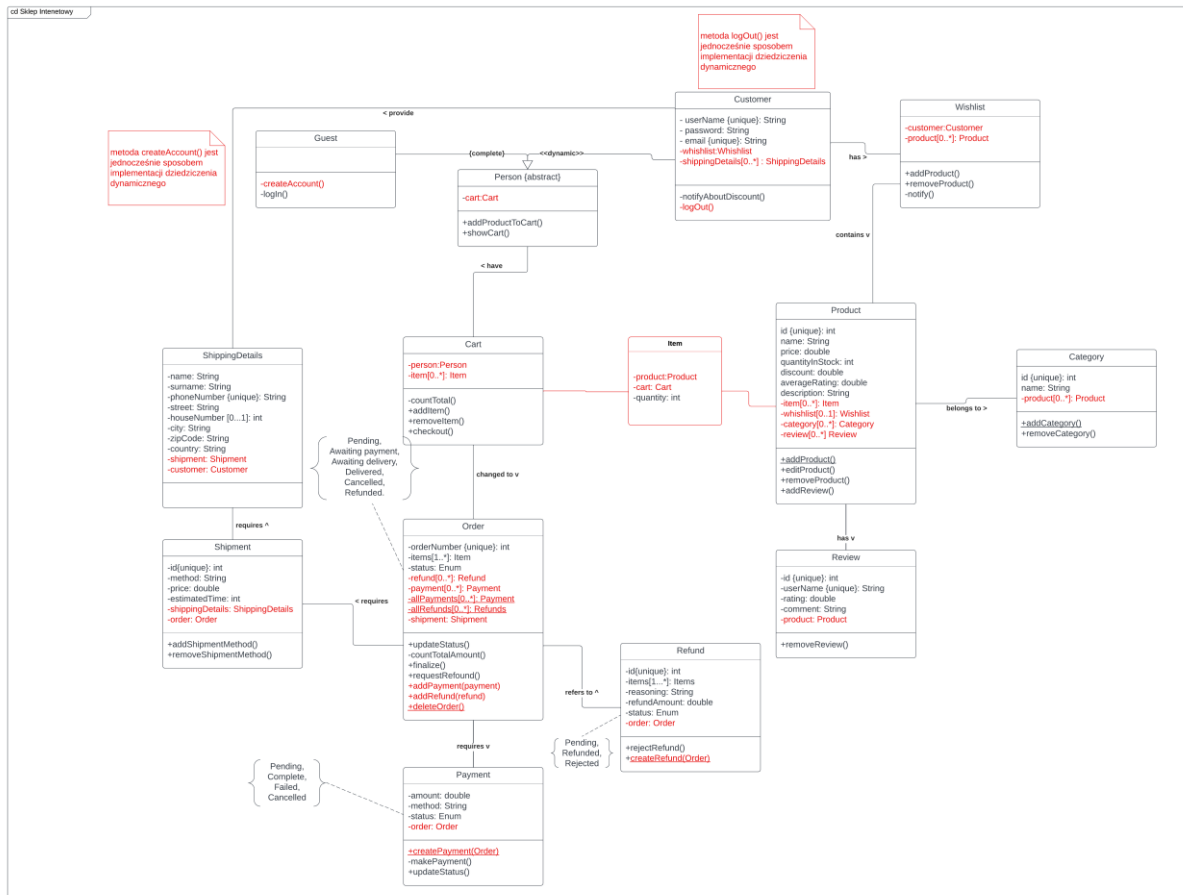


10. Decyzje projektowe

Konstrukcje niewystępujące w języku Java będą implementowane w następujący sposób:

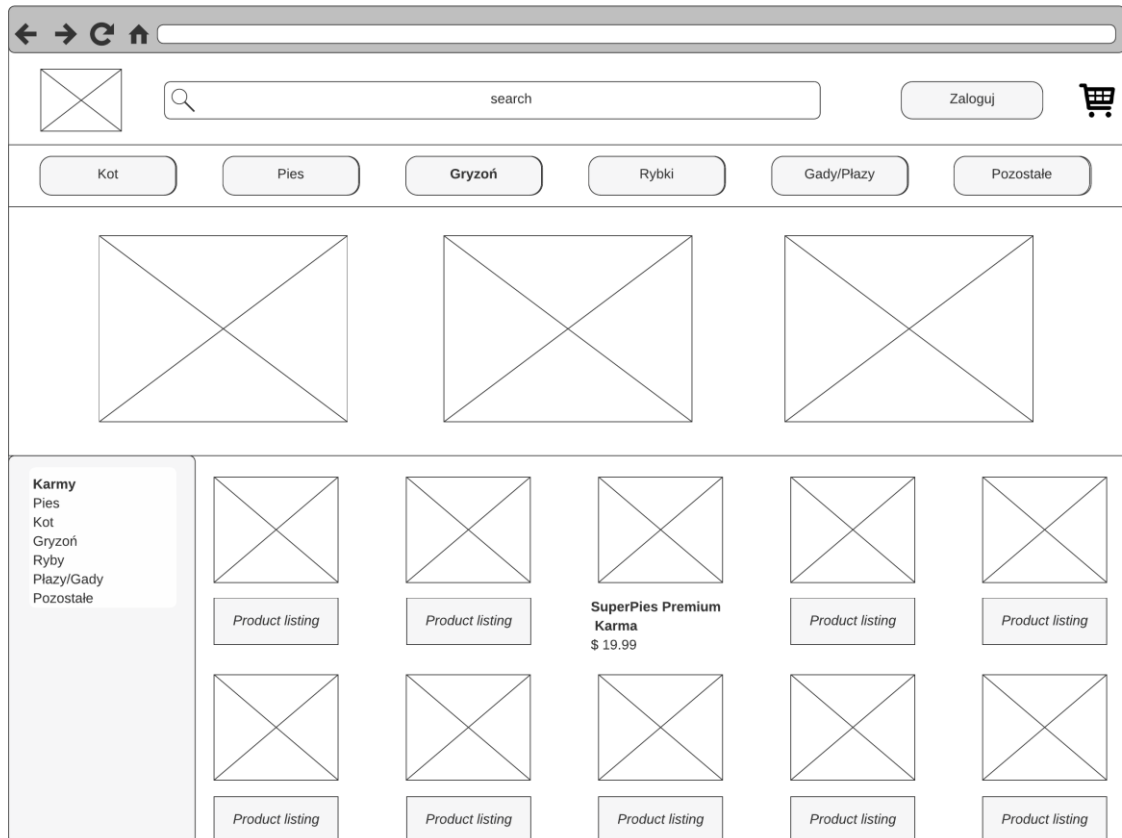
1. Asocjacja -> W zależności od liczności asocjacji będzie ona realizowana przez obiekt klasy, dla liczności 1 i listę obiektów klasy dla liczności *
2. Asocjacja z atrybutem -> Zostanie zrealizowana poprzez dodanie dodatkowej klasy a asocjacje między klasami jak w punkcie 1
3. Kompozycja -> Zostanie potraktowana jak asocjacja 1 do * z dodaniem metod do blokowania samodzielnego blokowania części, Zakazania współdzielenia części poprzez odpowiednio weryfikowaną globalną listę części oraz usuwanie części przy usuwaniu całości poprzez usunięcie metodą klasową całość z ekstensji oraz usunięcie części z globalnej listy części.
4. Dziedziczenie dynamiczne -> Do tego celu można wykorzystać istniejące już funkcje rejestrowania nowego użytkownika oraz funkcję wylogowania klienta poprzez kopiowanie obiektów.

11. Schemat logiczny systemu



12. Projekt interfejsu użytkownika

Strona główna



Zaloguj

KotPiesGryzońRybkıGady/PłazyPozostałe

Twój koszyk

SuperPies Premium Karma

-1+

\$19.99

Drzewko dla kota Pro Zabawka

-2+

\$79.98

Domek dla chomika Ultra Akcesoria

-1+

\$9.99

Wartość produktów:

\$109.96

Dostawa i płatność

Kontynuuj zakupy

Formularz Adresowy

Zaloguj

Dane adresowe

Imię

Nazwisko

numer telefonu

email

* Pole nieobowiązkowe

Anuluj

Ulica

Numer mieszkania *


Miasto


Kod pocztowy

Kraj

Akceptuj

Formularz Sposobu dostawy





Zaloguj

Sposób dostawy

Odbiór osobisty

☒

Poczta polska

☐

Kurier Firmex

☐

Kurier Dostawpol

☐

Anuluj

Akceptuj

Formularz sposobu opłaty

Zaloguj

Sposób płatności

Przelew tradycyjny

☐

Szybki przelew online

☐

Gotówką przy odbiorze

☐

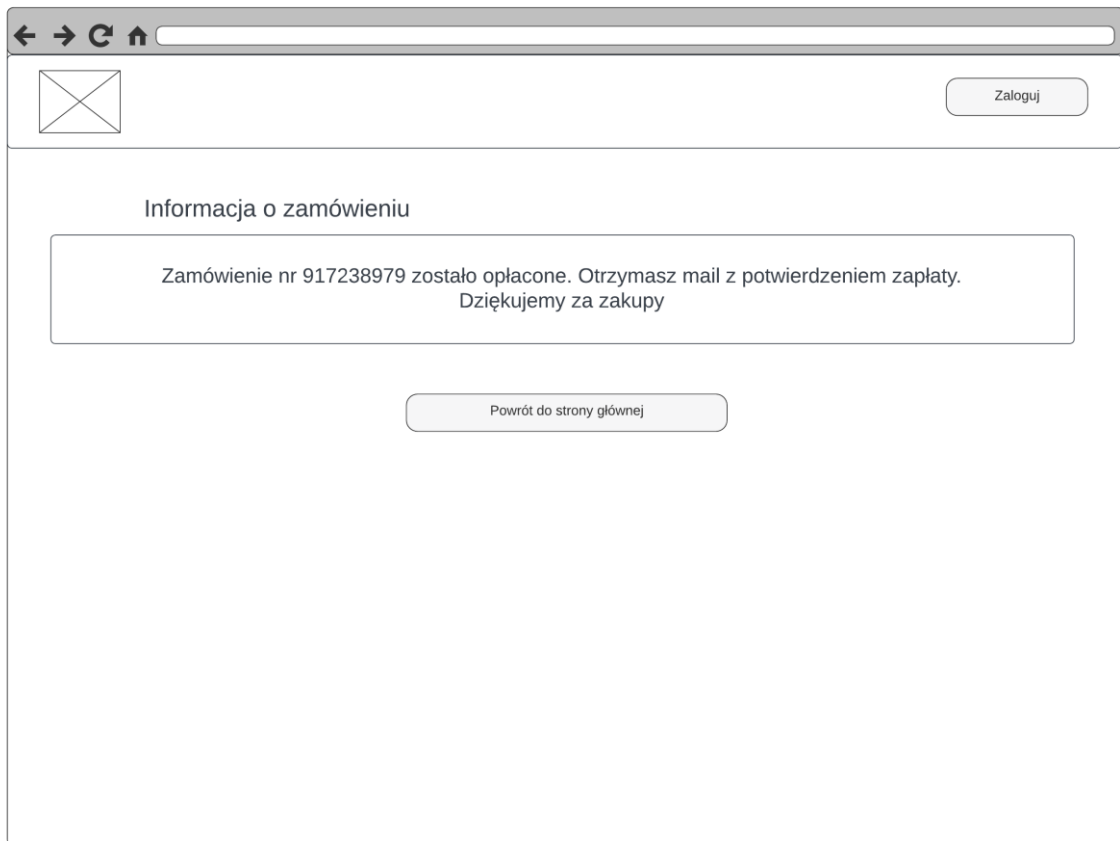
Blik

☒

Anuluj

Przejdź do płatności

Strona informująca o statusie płatności



13. Wymagania niefunkcjonalne

<i>Ograniczenie</i>	<i>Miara jakościowa</i>
Dostępność	System ma być dostępny dla użytkowników 24/7/364 (o każdej godzinie każdego dnia przez cały rok z przerwami na aktualizację systemu).
Przechowywanie danych w konkretnym systemie baz danych	Serwerem baz danych powinien być PostgreSQL.
Spójność i kompletność danych	System powinien sprawdzać formularze pod względem ich poprawności i kompletności.

14. Słownik

<i>Pojęcie</i>	<i>Synonim</i>	<i>Objaśnienie</i>
Administrator	Admin	Użytkownik o specjalnych uprawnieniach, mający
Gość		Użytkownik niezarejestrowany w systemie
Klient		Zarejestrowany użytkownik systemu
Lista życzeń		Lista tworzona przez użytkownika do obserwowania produktów na stronie.
Produkt		Obiekt na stronie internetowej możliwy do zakupienia
Przedmiot		Jest reprezentacją produktu i jego ilości w koszyku
Użytkownik		Dowolny użytkownik systemu. Może być to Gość lub Klient