**Obraz zawierający tekst, Czcionka, logo, wizytówka

Opis wygenerowany automatycznie**

**Łukasz Śliwa**

**125167**

***„Jalape******ño – Sklep Online z Ostrymi Paprykami i Nasionami”***

**Praca projektowa z Inżynierii oprogramowania**

Prowadzący: mgr inż. Ewa Żesławska

Rzeszów 2024

Spis treści

[1. Opis biznesowy projektowanego systemu 3](#_Toc189173831)

[2. Modelowanie systemu i jego otoczenia 5](#_Toc189173832)

[3. Opis techniczny projektowanego systemu 20](#_Toc189173833)

[4. Opis struktury projektu 21](#_Toc189173834)

[5. Projekt interfejsu użytkownika 24](#_Toc189173835)

[6. Podsumowanie 29](#_Toc189173836)

[Spis rysunków 30](#_Toc189173837)

[Spis tabel 31](#_Toc189173838)

[Oświadczenie studenta o samodzielności pracy 32](#_Toc189173839)

1. Opis biznesowy projektowanego systemu

**1.1. Opis świata rzeczywistego**

Aplikacja internetowa „Jalapeño” to platforma e-commerce specjalizująca się w sprzedaży papryk o wysokim poziomie ostrości oraz ich nasion, skierowana zarówno do amatorów, jak i profesjonalnych hodowców papryk. Użytkownicy mogą przeglądać katalog różnych gatunków papryk i nasion, zapoznawać się z ich specyfikacjami, poziomem ostrości, krajem pochodzenia oraz metodami uprawy. Aplikacja umożliwia dokonywanie zakupów online, zarządzanie koszykiem oraz realizację płatności, jednocześnie zapewniając narzędzia administracyjne do obsługi zamówień. Zaimplementowane funkcje uwzględniają zarządzanie katalogiem produktów, co pozwala administratorom na dodawanie i edytowanie ofert.

**1.2. Wymagania funkcjonalne**

1. Logowanie użytkowników:

- System powinien zapewniać funkcjonalność logowania oraz bezpieczne przechowywanie haseł.

2. Zarządzanie kontem użytkownika:

- Użytkownicy mogą przeglądać oraz aktualizować dane swojego konta (imię, nazwisko, adres dostawy).  
- Administrator może zarządzać kontami użytkowników.

- System powinien pozwalać użytkownikom na resetowanie hasła w przypadku jego utraty.

3. Przegląd i zarządzanie katalogiem papryk:

- System wyświetla szczegółowe informacje o dostępnych paprykach oraz nasionach, w tym ich nazwę, cenę, poziom ostrości, pochodzenie oraz instrukcje uprawy.

- Administratorzy mogą dodawać, edytować i usuwać produkty z katalogu, w tym aktualizować ich ceny i zdjęcia.

4. Koszyk i finalizacja zamówienia:

- Użytkownicy mogą dodawać produkty do koszyka, zarządzać ilością oraz usuwać wybrane pozycje.

- System umożliwia finalizację zamówienia i dokonanie płatności online.

**1.3. Wymagania niefunkcjonalne**

1. Wydajność:

- System powinien obsługiwać jednoczesne logowanie co najmniej 1000 użytkowników i płynne przetwarzanie zamówień bez opóźnień.

2. Bezpieczeństwo:

- Wszystkie dane użytkowników, w tym hasła i dane płatności, muszą być przechowywane bezpiecznie oraz szyfrowane podczas transmisji (SSL/TLS).

3. Skalowalność:

- Aplikacja musi być skalowalna, tak aby w przyszłości mogła obsługiwać zwiększoną liczbę produktów i użytkowników o 100% bez modyfikacji architektury.

4. Dostępność:

- System musi być dostępny 24/7, z maksymalnym czasem przestoju nieprzekraczającym 1 godziny miesięcznie.

5. Łatwość użycia:

- Interfejs aplikacji musi być intuicyjny i prosty w obsłudze, szczególnie dla użytkowników nieposiadających specjalistycznej wiedzy technicznej.

6. Kompatybilność i responsywność:

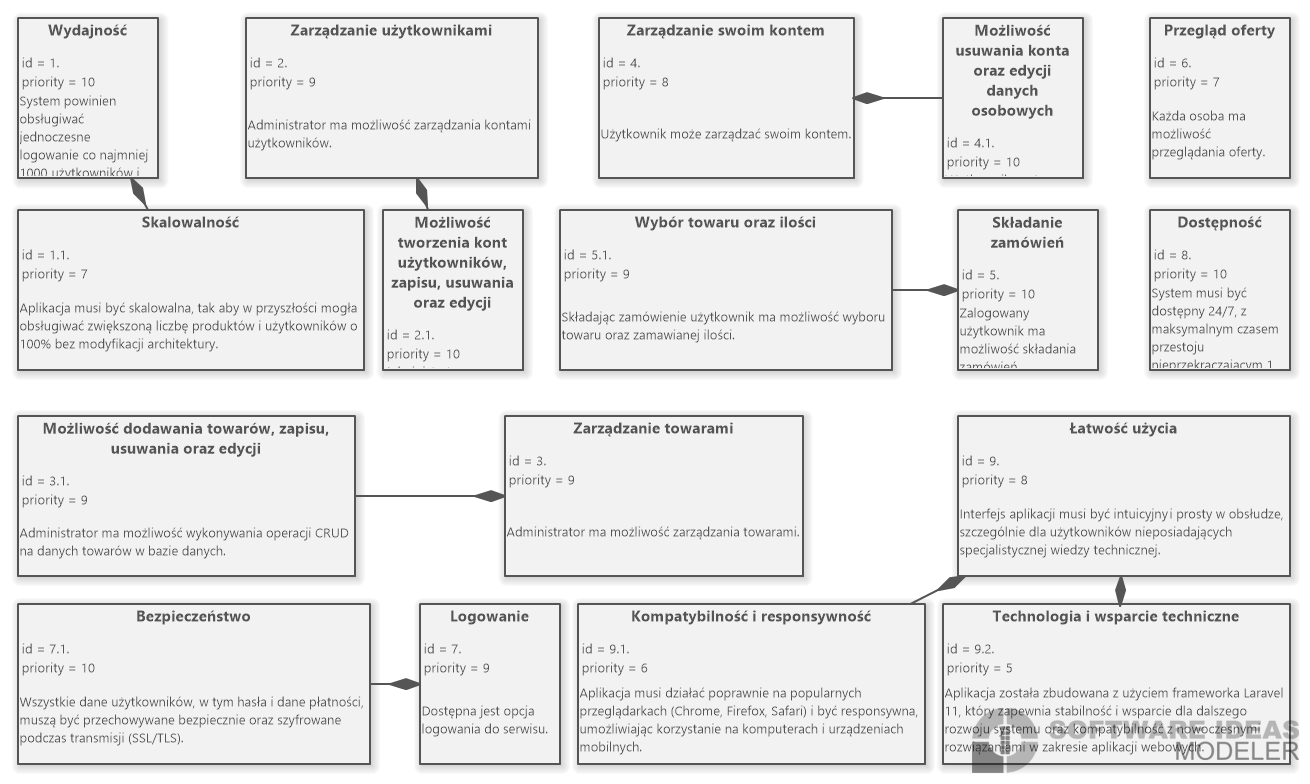
- Aplikacja musi działać poprawnie na popularnych przeglądarkach (Chrome, Firefox, Safari) i być responsywna, umożliwiając korzystanie na komputerach i urządzeniach mobilnych.

7. Technologia i wsparcie techniczne:

Aplikacja została zbudowana z użyciem frameworka Laravel 11, który zapewnia stabilność i wsparcie dla dalszego rozwoju systemu oraz kompatybilność z nowoczesnymi rozwiązaniami w zakresie aplikacji webowych.

**1.4. Diagram wymagań**

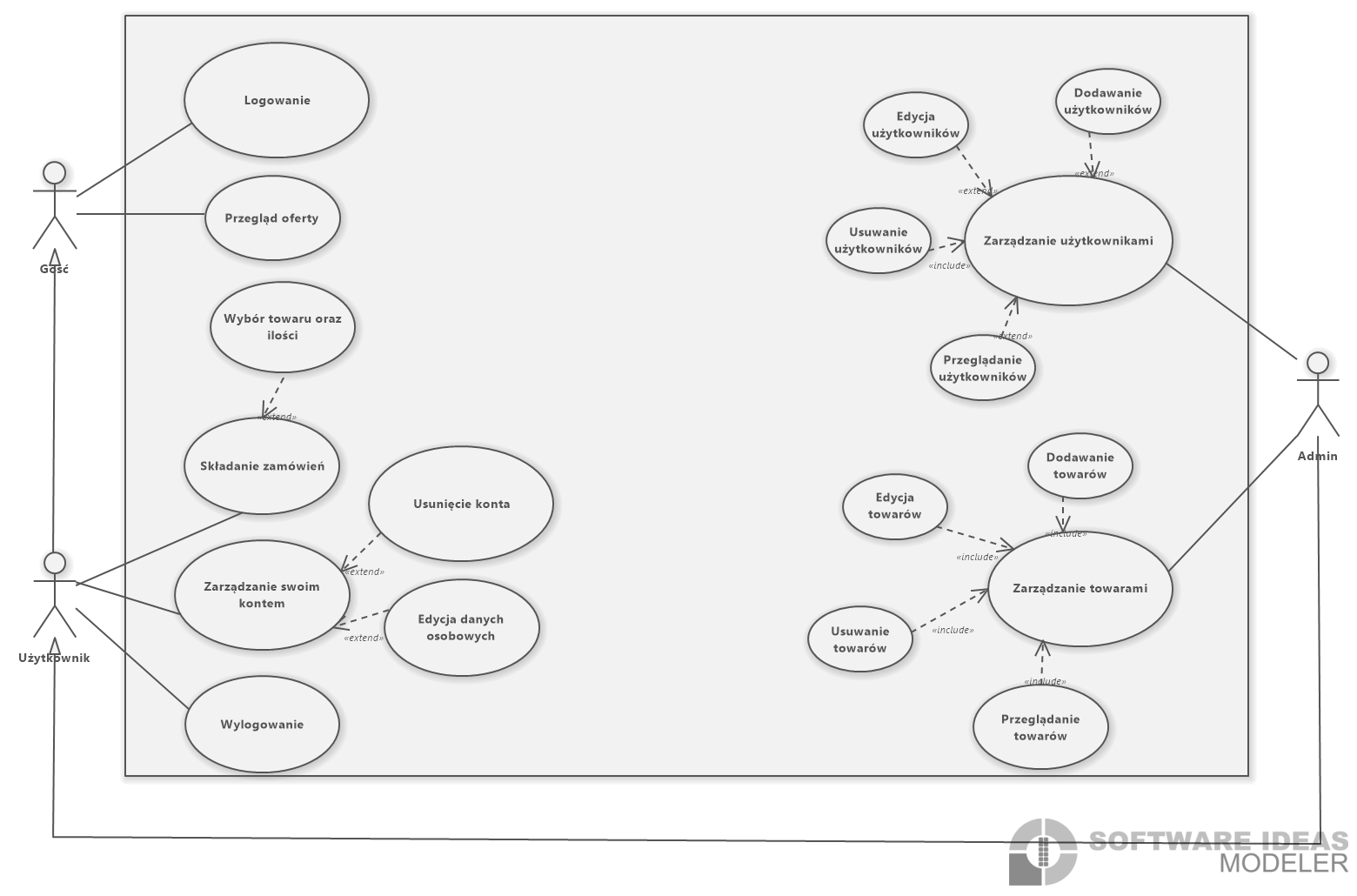
Na Rysunku 1 przedstawiono diagram wymagań dla projektu aplikacji e-commerce „Jalapeño”. Diagram ten ilustruje kluczowe funkcjonalności i wymagania systemu, obejmujące obsługę użytkowników, zarządzanie katalogiem papryk, obsługę koszyka i zamówień oraz funkcje administracyjne. Uwzględnia zarówno wymagania funkcjonalne, takie jak logowanie użytkowników, możliwość edycji konta, dodawanie produktów do koszyka czy zarządzanie zamówieniami, jak i wymagania niefunkcjonalne, takie jak wydajność, bezpieczeństwo oraz responsywność interfejsu. Każde wymaganie w diagramie posiada unikalny identyfikator (ID), priorytet, określający jego znaczenie w systemie, oraz opis, który precyzuje jego rolę i wpływ na działanie aplikacji.

****

Rysunek : Diagram wymagań

1. Modelowanie systemu i jego otoczenia
   1. **Diagram przypadków użycia**

Na Rysunku 2 przedstawiono diagram przypadków użycia dla aplikacji e-commerce „Jalapeño”. Diagram ten pokazuje wszystkie możliwe interakcje użytkowników i administratora z systemem. Uwzględniono kluczowe działania, takie jak logowanie i rejestracja, przeglądanie katalogu produktów, dodawanie papryk do koszyka, składanie zamówień oraz zarządzanie kontem użytkownika. Administrator ma dodatkowe możliwości, takie jak dodawanie i edytowanie produktów, przeglądanie zamówień oraz zarządzanie użytkownikami.



Rysunek : Diagram przypadków użycia

* 1. **Przykłady scenariuszy przypadków użycia**

Tabela : Scenariusz 1 (Logowanie UC001)

|  |  |
| --- | --- |
| Element | Opis |
| Przypadek użycia | Zaloguj się do systemu (UC001) |
| Aktor | Gość |
| Cel | Gość chce uzyskać dostęp do swojego konta. |
| Warunki początkowe | Gość jest zarejestrowany i zna swoje dane logowania. |
| Przebieg główny | 1. Gość otwiera stronę logowania. 2. Gość wprowadza swoją nazwę użytkownika i hasło. 3. System weryfikuje dane logowania. 4. System przyznaje dostęp i przekierowuje gościa do strony głównej. |
| Scenariusz alternatywny | Krok 3: Jeśli dane logowania są niepoprawne, system wyświetla komunikat o błędzie i pozwala na ponowne wprowadzenie danych. |
| Warunki końcowe | Gość jest zalogowany i ma dostęp do swojego konta albo widzi komunikat o błędnych danych logowania. |

Tabela : Scenariusz 2 (Składanie zamówienia UC002)

|  |  |
| --- | --- |
| Element | Opis |
| Przypadek użycia | Złóż zamówienie (UC002) |
| Aktor | Zalogowany użytkownik |
| Cel | Użytkownik chce złożyć zamówienie na wybrane towary. |
| Warunki początkowe | Użytkownik jest zalogowany i przegląda ofertę. |
| Przebieg główny | 1. Użytkownik wybiera towary i określa ich ilość. 2. System podsumowuje wybrane produkty i oblicza koszt. 3. Użytkownik zatwierdza zamówienie. |
| Scenariusz alternatywny | Krok 3: Jeśli użytkownik anuluje zamówienie, system wyświetla informację o anulowaniu i usuwa wybrane produkty z koszyka. |
| Warunki końcowe | Zamówienie jest zarejestrowane w systemie albo koszyk pozostaje pusty. |

Tabela : Scenariusz 3 (Zarządzanie użytkownikami UC003)

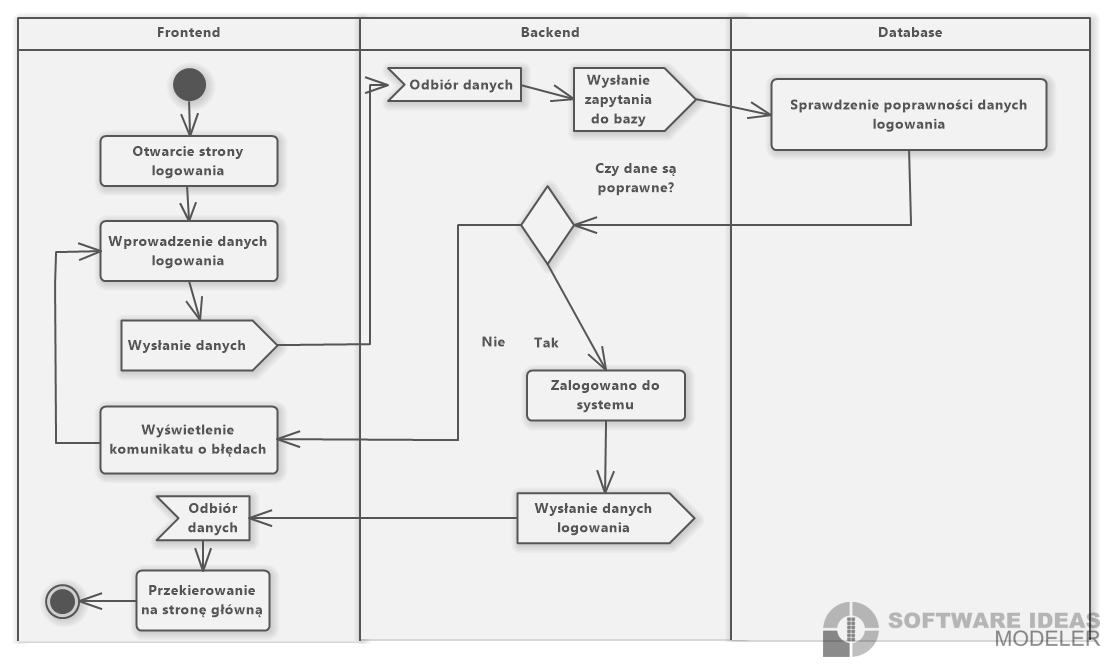
|  |  |
| --- | --- |
| Element | Opis |
| Przypadek użycia | Zarządzaj użytkownikami (UC003) |
| Aktor | Administrator |
| Cel | Administrator chce zarządzać kontami użytkowników. |
| Warunki początkowe | Administrator jest zalogowany i posiada odpowiednie uprawnienia. |
| Przebieg główny | 1. Administrator otwiera panel zarządzania użytkownikami. 2. Wybiera akcję: dodanie, edycja lub usunięcie konta. 3. Wprowadza lub aktualizuje dane użytkownika. 4. System zapisuje zmiany w bazie danych. |
| Scenariusz alternatywny | Krok 2: Jeśli dane użytkownika są nieprawidłowe, system wyświetla komunikat o błędzie i prosi o poprawienie danych. |
| Warunki końcowe | Konto użytkownika zostaje dodane, zaktualizowane lub usunięte, a zmiany zapisane w systemie. |

Tabela : Scenariusz 4 (Logowanie UC004)

|  |  |
| --- | --- |
| Element | Opis |
| Przypadek użycia | Usuń konto (UC004) |
| Aktor | Zalogowany użytkownik |
| Cel | Użytkownik chce usunąć swoje konto. |
| Warunki początkowe | Użytkownik jest zalogowany. |
| Przebieg główny | 1. Użytkownik otwiera panel zarządzania kontem. 2. Wybiera opcję „Usuń konto”. 3. System wyświetla komunikat z podwójnym potwierdzeniem. 4. Użytkownik potwierdza decyzję. 5. System usuwa konto i informuje użytkownika o zakończeniu operacji. |
| Scenariusz alternatywny | Krok 3: Jeśli użytkownik anuluje proces usuwania, system przerywa operację i zachowuje dane użytkownika. |
| Warunki końcowe | Konto użytkownika zostaje usunięte z systemu albo pozostaje nienaruszone. |

**2.3 Diagramy aktywności**

Na Rysunku 3 przedstawiono diagram aktywności dla procesu logowania użytkownika w aplikacji „Jalapeño”. Diagram podzielony jest na trzy partycje: Frontend, Backend oraz Database, które reprezentują odpowiedzialność poszczególnych warstw systemu. Proces rozpoczyna się od otwarcia strony logowania i wprowadzenia danych przez użytkownika. Następnie system przesyła dane do serwera, który sprawdza ich poprawność w bazie danych. W przypadku błędu użytkownik otrzymuje komunikat i może ponowić próbę. Jeśli dane są poprawne, użytkownik zostaje zalogowany i przekierowany na stronę główną.



Rysunek : Diagram aktywności – logowanie

Na Rysunku 4 przedstawiono diagram aktywności dla procesu składania zamówienia. Diagram podzielony jest na trzy warstwy: Frontend, Backend i Database, odzwierciedlając interakcję między użytkownikiem, serwerem i bazą danych. Użytkownik przegląda ofertę, wybiera produkty oraz określa ich ilość. Po zatwierdzeniu zamówienia system oblicza koszty i zapisuje dane w bazie. Jeśli użytkownik nie potwierdzi zamówienia, proces zostaje anulowany.

Obraz zawierający diagram, tekst, zrzut ekranu, Plan

Opis wygenerowany automatycznie

Rysunek : Diagram aktywności - składanie zamówienia

Na Rysunku 5 przedstawiono diagram aktywności dla zarządzania użytkownikami. Proces obejmuje dodawanie, edytowanie i usuwanie kont przez administratora. Po wyborze akcji administrator wprowadza dane użytkownika, które są przesyłane do systemu. System weryfikuje poprawność danych i aktualizuje je w bazie. W przypadku błędów administrator otrzymuje komunikat o konieczności poprawienia danych.

Obraz zawierający tekst, diagram, zrzut ekranu, Plan

Opis wygenerowany automatycznie

Rysunek : Diagram aktywności – zarządzanie użytkownikami

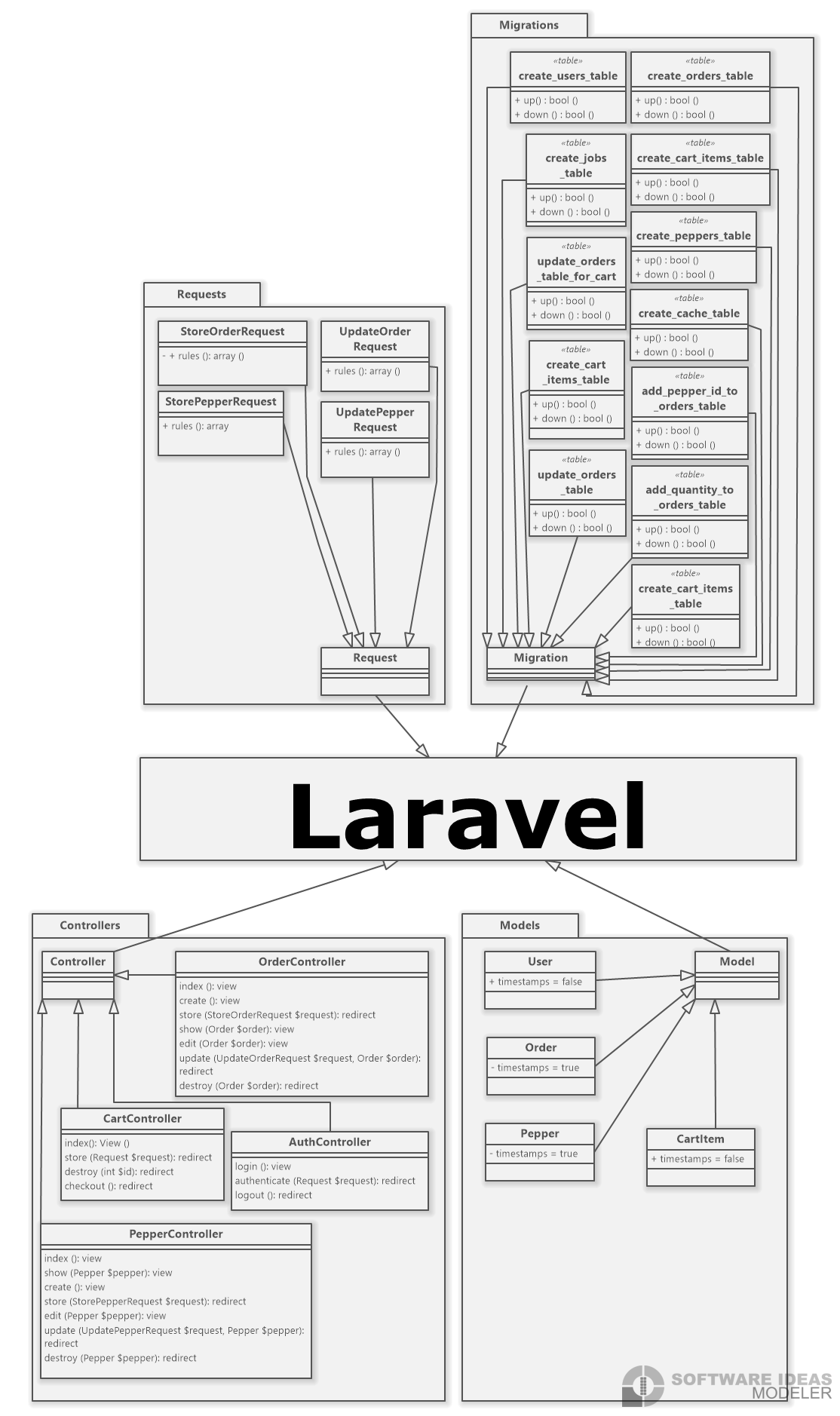
Na Rysunku 6 przedstawiono diagram aktywności dla procesu usuwania konta użytkownika. Użytkownik inicjuje usunięcie konta poprzez panel zarządzania kontem. Po potwierdzeniu decyzji system sprawdza dane użytkownika, usuwa konto z bazy i informuje o sukcesie operacji. Jeśli użytkownik anuluje proces, konto pozostaje aktywne.

Obraz zawierający tekst, diagram, zrzut ekranu, Równolegle

Opis wygenerowany automatycznie

Rysunek : Diagram aktywności – usuwanie konta

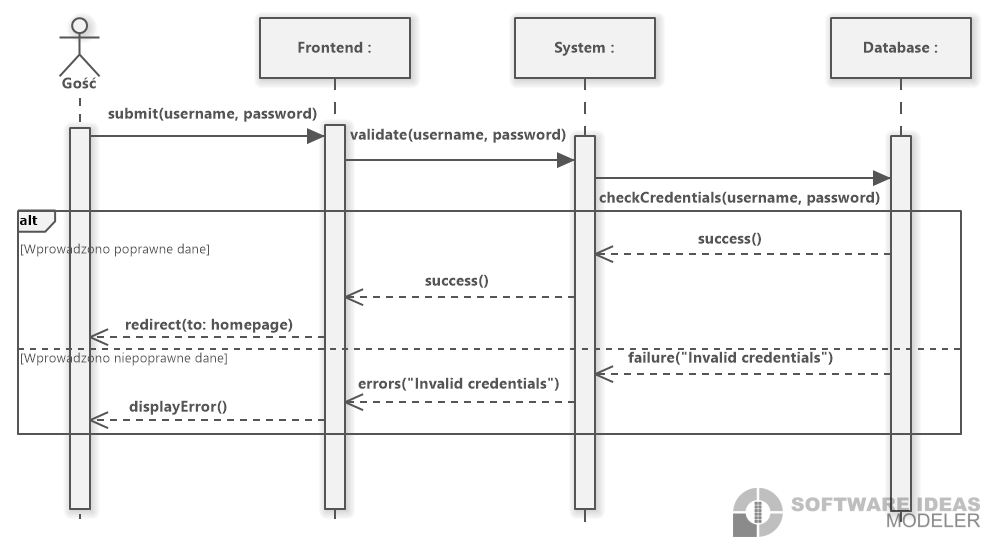
**2.4 Diagram klas**



Rysunek : Diagram klas

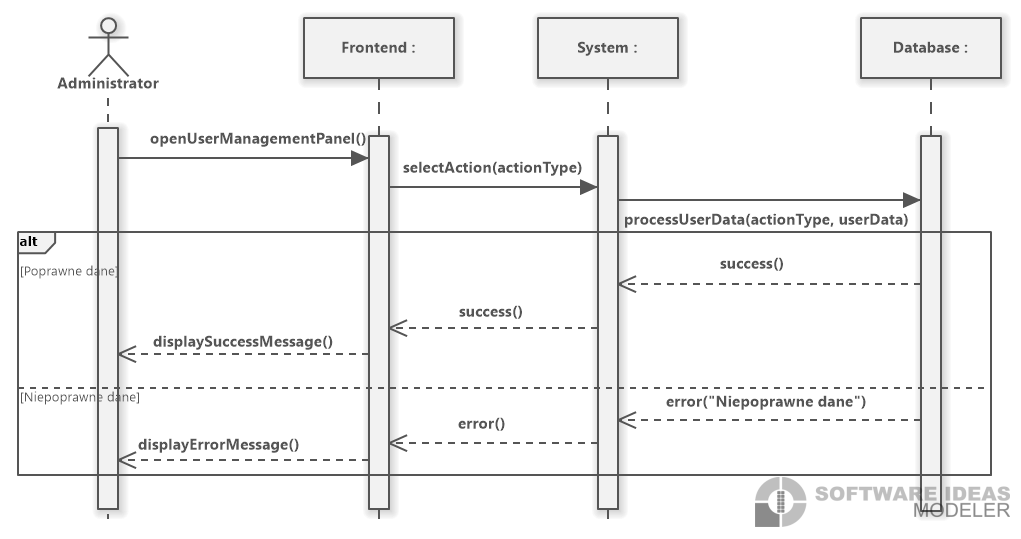
**2.5. Diagramy sekwencji**

Na Rysunku 8 przedstawiono diagram sekwencji dla procesu logowania użytkownika. Ukazuje on interakcję pomiędzy użytkownikiem, frontendem, systemem i bazą danych. Proces rozpoczyna się od wprowadzenia danych logowania, które są walidowane przez system. Jeśli dane są poprawne, użytkownik zostaje zalogowany i przekierowany na stronę główną. W przeciwnym razie wyświetlany jest komunikat o błędzie.



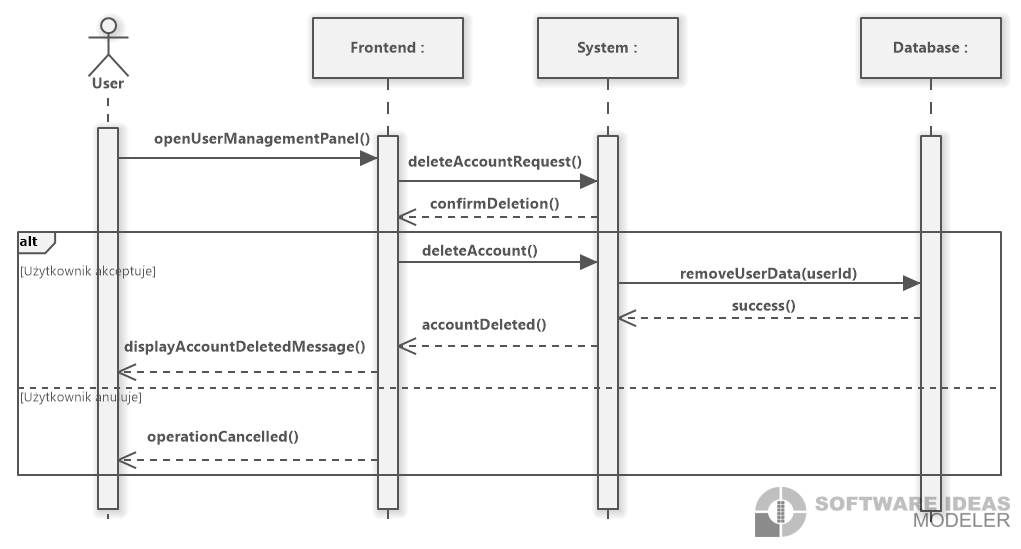
Rysunek : Diagram sekwencji – logowanie

Na Rysunku 9 zaprezentowano diagram sekwencji ilustrujący proces zarządzania użytkownikami przez administratora. Administrator otwiera panel zarządzania użytkownikami i wybiera odpowiednią akcję (dodanie, edycję lub usunięcie konta). System weryfikuje poprawność danych i aktualizuje je w bazie. Jeśli wystąpi błąd, użytkownik otrzymuje stosowny komunikat.



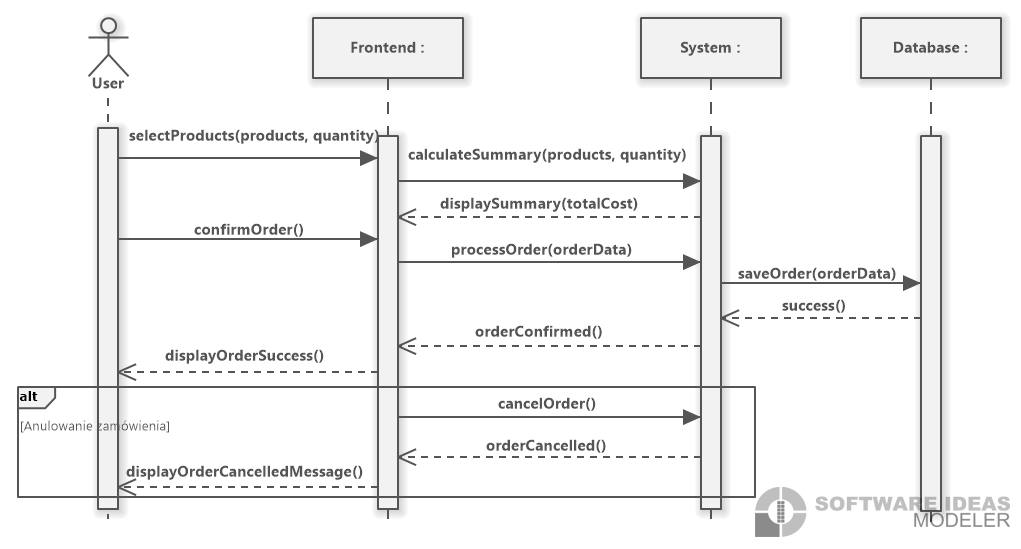
Rysunek : Diagram sekwencji – składanie zamówień

Na Rysunku 10 przedstawiono diagram sekwencji dla procesu usuwania konta użytkownika. Po otwarciu panelu zarządzania kontem użytkownik inicjuje proces usunięcia konta. System prosi o potwierdzenie operacji, po czym usuwa dane użytkownika z bazy i informuje o zakończeniu procesu. W przypadku anulowania operacji konto pozostaje aktywne.



Rysunek : Diagram sekwencji – zarządzanie użytkownikami

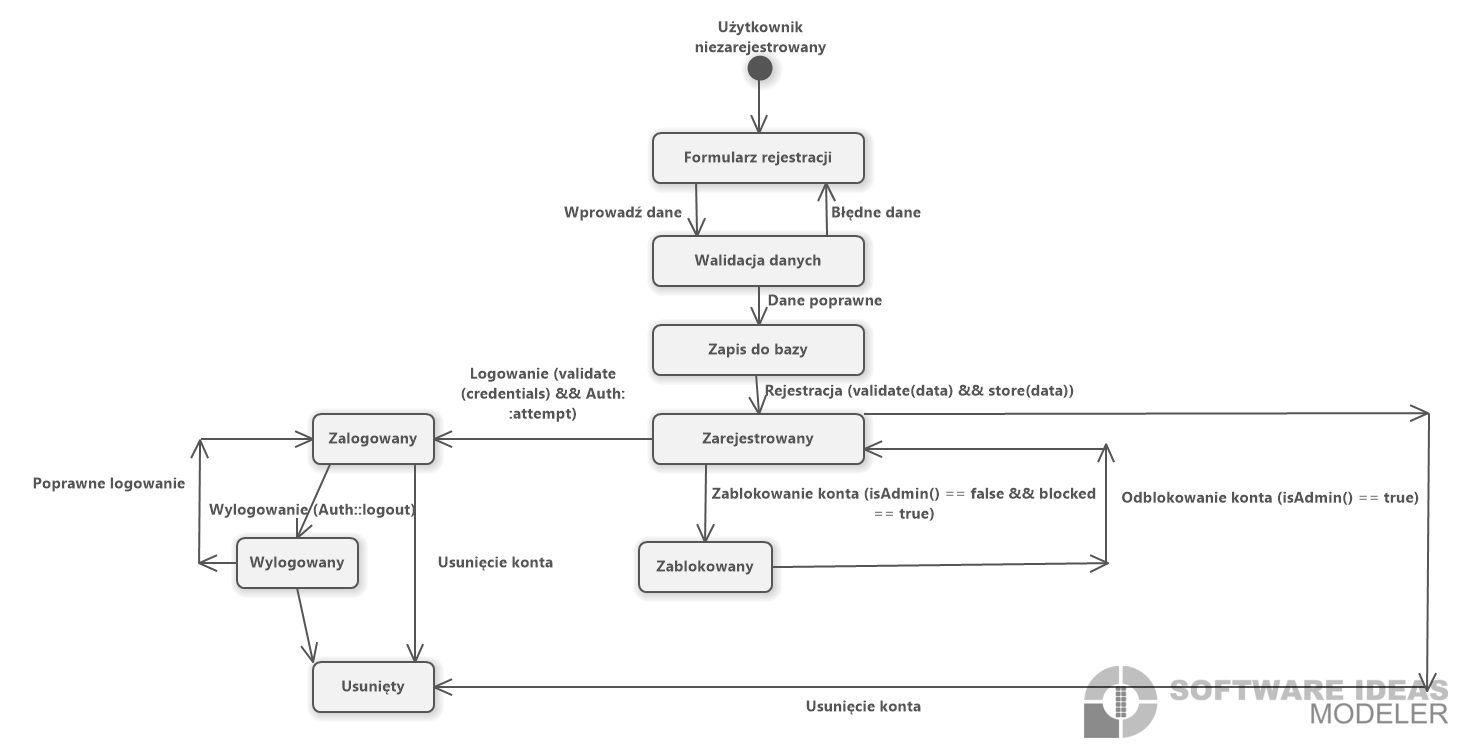
Na Rysunku 11 przedstawiono diagram sekwencji dla procesu składania zamówienia. Użytkownik wybiera produkty i ich ilość, a następnie system oblicza koszty i wyświetla podsumowanie. Po potwierdzeniu zamówienia dane są zapisywane w bazie, a użytkownik otrzymuje potwierdzenie. W przypadku anulowania operacji zamówienie nie zostaje zapisane.



Rysunek : Diagram sekwencji – usuwanie konta

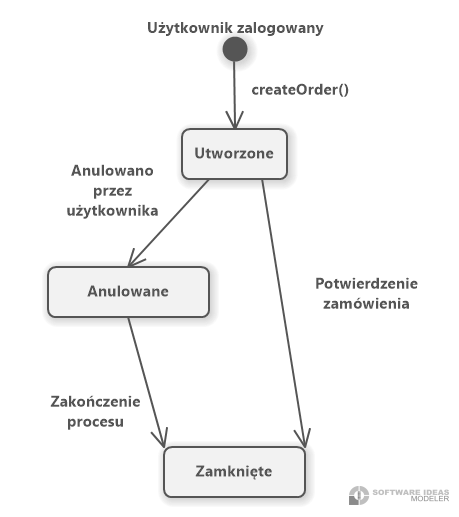
**2.6. Diagramy stanów**

Na Rysunku 12 przedstawiono diagram stanów użytkownika w procesie rejestracji i logowania do systemu. Proces rozpoczyna się od użytkownika niezarejestrowanego, który wypełnia formularz rejestracyjny. Jeśli dane są poprawne, system zapisuje je w bazie i przechodzi do stanu „Zarejestrowany”. Po zalogowaniu użytkownik przechodzi do stanu „Zalogowany”, a po wylogowaniu – do „Wylogowany”. System obsługuje także przypadki konta zablokowanego lub usuniętego przez administratora.



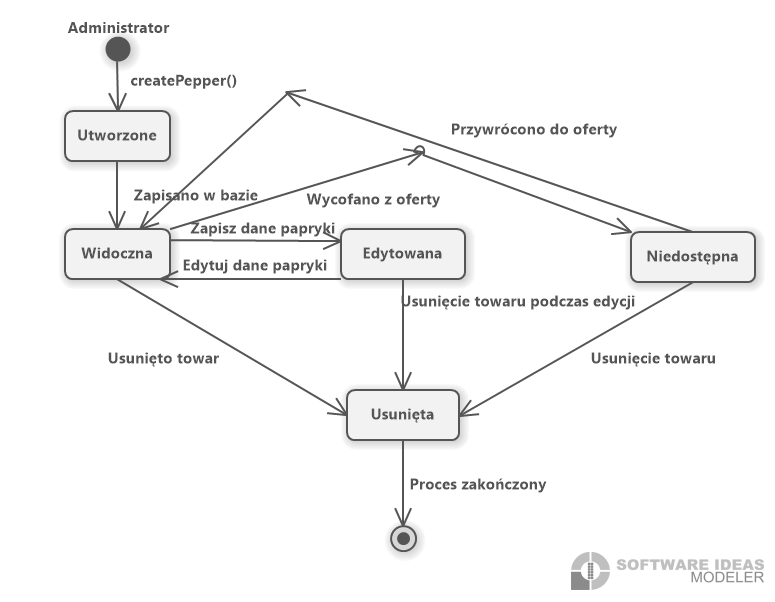
Rysunek : Diagram stanu – logowanie

Na Rysunku 13 przedstawiono diagram stanów zamówienia w systemie. Proces rozpoczyna się od utworzenia zamówienia przez zalogowanego użytkownika. Zamówienie może zostać anulowane przez użytkownika lub potwierdzone, co prowadzi do jego zamknięcia. Diagram odzwierciedla główne przejścia między stanami zamówienia, zapewniając przejrzystość procesu jego realizacji.



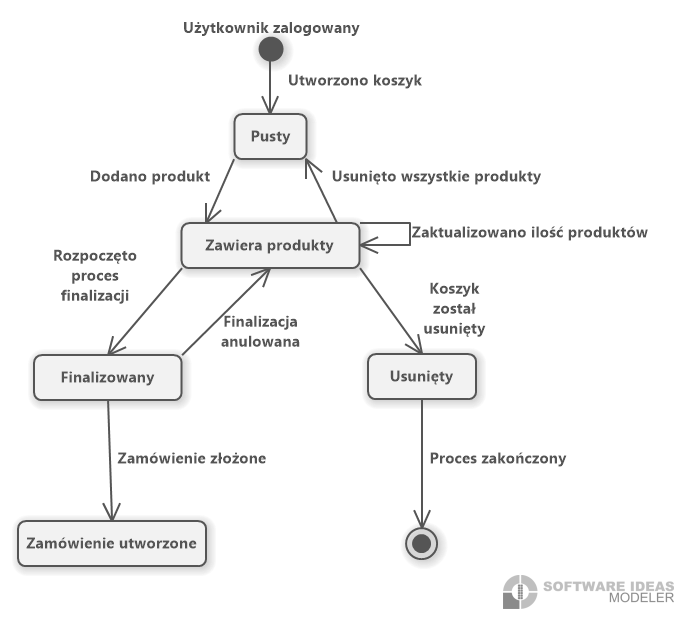
Rysunek : Diagram stanu – zamówienia

Na Rysunku 14 przedstawiono cykl życia produktu w systemie. Produkt może zostać utworzony przez administratora i przejść do stanu „Widoczna” po zapisaniu w bazie. Administrator może edytować produkt, usunąć go lub wycofać z oferty. Jeśli produkt zostanie usunięty, przechodzi do stanu „Usunięta”, co kończy jego cykl życia w systemie.



Rysunek : Diagram stanu – towary

Na Rysunku 15 przedstawiono cykl życia koszyka użytkownika. Koszyk rozpoczyna jako „Pusty”, do którego użytkownik może dodawać produkty. Po dodaniu produktów przechodzi do stanu „Zawiera produkty”. Jeśli użytkownik anuluje zamówienie, koszyk może zostać usunięty. Jeśli jednak przejdzie do finalizacji, koszyk zmienia stan na „Finalizowany”, a po złożeniu zamówienia przechodzi do „Zamówienie utworzone”, kończąc jego cykl życia.



Rysunek : Diagram stanu - koszyk

1. Opis techniczny projektowanego systemu

Projekt „Jalapeño – Sklep Online z Ostrymi Paprykami i Nasionami” to nowoczesna aplikacja webowa, umożliwiająca użytkownikom przeglądanie oferty papryk, dodawanie ich do koszyka oraz składanie zamówień online. System został zaprojektowany z wykorzystaniem innowacyjnych technologii, zapewniających elastyczność, bezpieczeństwo i intuicyjność obsługi.

**Technologie i rozwiązania**

Aplikacja wykorzystuje framework Laravel, który stanowi podstawę backendu. Laravel został wybrany ze względu na wydajność, dużą społeczność i wsparcie dla architektury MVC (Model-View-Controller). Dzięki temu możliwe jest zachowanie modularności i przejrzystości kodu.

Frontend został zrealizowany przy użyciu silnika szablonów Blade, co pozwala na dynamiczne generowanie widoków w oparciu o dane dostarczane przez backend. Stylizacja została wykonana z użyciem Bootstrap, co zapewnia responsywność i estetyczny wygląd interfejsu na różnych urządzeniach, zarówno mobilnych, jak i desktopowych.

JavaScript został zastosowany do obsługi dynamicznych elementów interfejsu, takich jak przyciski czy formularze. System bazuje na relacyjnej bazie danych MySQL, która umożliwia sprawną obsługę danych o produktach, zamówieniach i użytkownikach.

**Wymagania sprzętowe**

**Serwer aplikacji:**

- Procesor: Minimum 2 rdzenie obsługujące architekturę 64-bitową.

- Pamięć RAM: Co najmniej 4 GB.

- Przestrzeń dyskowa: 50 GB wolnego miejsca na potrzeby aplikacji (zdjęcia, logi) oraz bazy danych.

- System operacyjny: Linux (preferowany Ubuntu 20.04 lub nowszy) lub Windows Server.

- Oprogramowanie: PHP w wersji 8.2, MySQL 8.0 lub wyższy, serwer WWW (Apache lub Nginx).

**Klient końcowy:**

- System operacyjny: Windows, macOS, Linux.

- Przeglądarka internetowa: Najnowsze wersje Google Chrome, Mozilla Firefox lub Microsoft Edge (z obsługą JavaScript i CSS3).

- Rozdzielczość ekranu: Minimalna 1280x720.

- Połączenie internetowe: Stabilne, o przepustowości co najmniej 5 Mbps.

**Kluczowe założenia techniczne**

1. MVC: Architektura MVC zapewnia jasny podział odpowiedzialności między warstwami aplikacji, co ułatwia utrzymanie i rozwój kodu.

2. Responsywność: Dzięki Bootstrap aplikacja automatycznie dostosowuje układ do różnych wielkości ekranów, zapewniając optymalny wygląd na urządzeniach mobilnych i stacjonarnych.

3. Modularność: Projekt zaprojektowano z myślą o łatwej rozbudowie – nowe funkcjonalności mogą być dodawane bez ingerencji w podstawowe moduły systemu.

4. Bezpieczeństwo: System uwzględnia zaawansowane mechanizmy uwierzytelniania i zarządzania użytkownikami, co minimalizuje ryzyko nieautoryzowanego dostępu.

1. Opis struktury projektu

Projekt bazuje na frameworku Laravel i wykorzystuje bazę danych MySQL do zarządzania danymi. Struktura bazy danych została zaprojektowana w sposób umożliwiający efektywne przechowywanie i pobieranie informacji związanych z użytkownikami, zamówieniami oraz produktami (paprykami). Do komunikacji z bazą danych wykorzystywany jest Eloquent ORM, który umożliwia łatwe operowanie na danych przy użyciu modeli.

**4.1 Struktura bazy danych**

Baza danych składa się z następujących tabel:

**Użytkownicy:**

id – unikalny identyfikator użytkownika,

name, surname – imię i nazwisko użytkownika,

username, email – unikalne dane identyfikacyjne,

password – zaszyfrowane hasło,

role – rola użytkownika (np. admin, klient).

**Produkty:**

id – identyfikator papryki,

name, description – nazwa i opis,

price – cena produktu,

img – ścieżka do obrazka produktu.

**Koszyk** :

id – identyfikator rekordu,

user\_id – identyfikator użytkownika,

pepper\_id – identyfikator papryki,

quantity – ilość danego produktu w koszyku,

created\_at, updated\_at – daty utworzenia i aktualizacji.

**Zamówienia:**

id – identyfikator zamówienia,

user\_id – identyfikator użytkownika składającego zamówienie,

items – JSON zawierający szczegóły zamówienia,

total\_price – całkowita kwota zamówienia,

created\_at, updated\_at – daty utworzenia i aktualizacji.

**Sesje**:

id – identyfikator sesji,

user\_id – identyfikator użytkownika (może być NULL dla gości),

ip\_address, user\_agent – dane sesji,

last\_activity – ostatnia aktywność użytkownika.

**Tokeny resetowania hasła:**

email – adres e-mail użytkownika,

token – unikalny token,

created\_at – data utworzenia tokenu.

**Kolejki zadań:**

jobs – przechowuje aktualne zadania w kolejce.

job\_batches – grupuje zadania przetwarzane w partiach.

failed\_jobs – przechowuje informacje o błędnych zadaniach.

**Cache**:

Tabele przechowują dane cache’owane przez aplikację, np. sesje lub ustawienia systemowe.

**4.2 Diagram ERD**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Rysunek : Diagram ERD

**4.3 Dostęp do bazy danych w Laravelu**

Laravel wykorzystuje Eloquent ORM do operacji na bazie danych. Modele odpowiadają tabelom:

User.php → users

Pepper.php → peppers

CartItem.php → cart\_items

Order.php → orders

Przykład użycia Eloquent do pobrania zamówień użytkownika:

$user = User::find(1);

$orders = $user->orders;

Dodanie nowego zamówienia:

$order = new Order();

$order->user\_id = 1;

$order->items = json\_encode([...]);

$order->total\_price = 99.99;

$order->save();

1. Projekt interfejsu użytkownika

Należy przedstawić opis planowanych widoków GUI (warstwy użytkowej) projektowanej aplikacji.

**Strona główna:**

Ten widok stanowi stronę główną aplikacji, prezentującą ofertę sprzedaży nasion ostrych papryk. Zawiera karuzelę z promocyjnymi obrazami różnych odmian papryk, sekcję z losowymi produktami oraz tabelę cennikową. Dynamicznie pobiera dane o produktach z bazy danych i umożliwia użytkownikowi przejście do szczegółów każdej papryki.Obraz zawierający warzywo, pieprz, Naturalne jedzenie, produkcja

Opis wygenerowany automatycznie

Rysunek : Strona główna aplikacji

**Produkt:**

Ten widok wyświetla szczegółowe informacje o wybranej papryce, w tym nazwę, opis, cenę i zdjęcie. Umożliwia użytkownikowi dodanie produktu do koszyka, a administratorowi jego edycję lub usunięcie. Dodatkowo, jeśli użytkownik jest zalogowany, może przejść bezpośrednio do swojego koszyka.

*Obraz zawierający tekst, zrzut ekranu, Strona internetowa, oprogramowanie

Opis wygenerowany automatycznie*

Rysunek : Widok produktu

**Oferta**:

Przedstawia dostępne produkty w formie katalogu, z możliwością przejścia do szczegółów każdej papryki. Administrator ma dodatkową opcję dodania nowej oferty. Produkty są wyświetlane dynamicznie w responsywnej siatce z obrazkami i krótkimi opisami.

*Obraz zawierający tekst, owoce, zrzut ekranu, Naturalne jedzenie

Opis wygenerowany automatycznie*

Rysunek : Przegląd oferty

**Zamówienia**:

Wyświetla listę zamówień użytkowników, zawierającą informacje o kliencie, dacie, łącznej kwocie oraz zamówionych produktach. Administratorzy mogą edytować i usuwać zamówienia. Dane są prezentowane w formie tabeli z opcjami akcji dla każdego wpisu.

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Rysunek : Przegląd zamówień

**Koszyk:**

Widok "Twój koszyk" przedstawia zawartość koszyka użytkownika w aplikacji e-commerce. Wyświetla listę dodanych produktów wraz z ich nazwą, ilością, ceną jednostkową i łączną wartością. Umożliwia użytkownikowi usuwanie pozycji z koszyka oraz przejście do finalizacji zamówienia. Gdy koszyk jest pusty, wyświetlany jest stosowny komunikat oraz przycisk umożliwiający powrót do zakupów.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Rysunek : Koszyk

**Edycja towaru:**

Widok "Edycja oferty" umożliwia administratorowi aktualizację danych produktu w katalogu. Formularz zawiera pola do edycji nazwy, opisu, ceny oraz możliwość przesłania nowego zdjęcia. System zapewnia walidację danych, wyświetlając komunikaty o błędach w razie nieprawidłowych wartości. Po zapisaniu zmian produkt zostaje zaktualizowany w bazie danych.

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie**

Rysunek : Edycja towaru

**Walidacja:**

**Obraz zawierający tekst, oprogramowanie, Strona internetowa, Czcionka

Opis wygenerowany automatycznie**

Rysunek : Walidacja

1. Podsumowanie

Dokumentacja opisuje system „Jalapeño”, aplikację e-commerce do sprzedaży ostrych papryk i nasion, zbudowaną w Laravel 11 i MySQL. Przedstawiono wymagania, architekturę, strukturę bazy danych oraz diagramy UML. Opisano także interfejs użytkownika oraz sposób dostępu do danych poprzez Eloquent ORM. System zaprojektowano z myślą o wydajności, skalowalności i bezpieczeństwie, zapewniając intuicyjną obsługę i możliwość dalszego rozwoju.

Spis rysunków

Należy umieścić spis rysunków generowany automatycznie.

[Rysunek 1: Diagram wymagań 4](#_Toc189173808)

[Rysunek 2: Diagram przypadków użycia 5](#_Toc189173809)

[Rysunek 3: Diagram aktywności – logowanie 8](#_Toc189173810)

[Rysunek 4: Diagram aktywności - składanie zamówienia 9](#_Toc189173811)

[Rysunek 5: Diagram aktywności – zarządzanie użytkownikami 10](#_Toc189173812)

[Rysunek 6: Diagram aktywności – usuwanie konta 11](#_Toc189173813)

[Rysunek 7: Diagram klas 12](#_Toc189173814)

[Rysunek 8: Diagram sekwencji – logowanie 13](#_Toc189173815)

[Rysunek 9: Diagram sekwencji – składanie zamówień 14](#_Toc189173816)

[Rysunek 10: Diagram sekwencji – zarządzanie użytkownikami 14](#_Toc189173817)

[Rysunek 11: Diagram sekwencji – usuwanie konta 15](#_Toc189173818)

[Rysunek 12: Diagram stanu – logowanie 16](#_Toc189173819)

[Rysunek 13: Diagram stanu – zamówienia 17](#_Toc189173820)

[Rysunek 14: Diagram stanu – towary 18](#_Toc189173821)

[Rysunek 15: Diagram stanu - koszyk 19](#_Toc189173822)

[Rysunek 16: Diagram ERD 23](#_Toc189173823)

[Rysunek 17: Strona główna aplikacji 24](#_Toc189173824)

[Rysunek 18: Widok produktu 25](#_Toc189173825)

[Rysunek 19: Przegląd oferty 26](#_Toc189173826)

[Rysunek 20: Przegląd zamówień 26](#_Toc189173827)

[Rysunek 21: Koszyk 27](#_Toc189173828)

[Rysunek 22: Edycja towaru 27](#_Toc189173829)

[Rysunek 23: Walidacja 28](#_Toc189173830)

Spis tabel

Należy umieścić spis tabel generowany automatycznie.

[Tabela 1: Scenariusz 1 (Logowanie UC001) 6](#_Toc189169937)

[Tabela 2: Scenariusz 2 (Składanie zamówienia UC002) 6](#_Toc189169938)

[Tabela 3: Scenariusz 3 (Zarządzanie użytkownikami UC003) 7](#_Toc189169939)

[Tabela 4: Scenariusz 4 (Logowanie UC004) 7](#_Toc189169940)

Oświadczenie studenta o samodzielności pracy

Łukasz Śliwa

Imię i nazwisko studenta

Wydział Nauk Ścisłych i Technicznych

Informatyka

Nazwa kierunku

ls125167

Nr albumu

Oświadczam, że moja praca projektowa pt.: „Jalapeño – Sklep Online z Ostrymi Paprykami i Nasionami”

* została przygotowana przeze mnie samodzielnie,
* nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2021 r., poz. 1062) oraz dóbr osobistych chronionych prawem cywilnym,
* nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony

podpis studenta