WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: Teleinformatyka

Kamil Dudek
Nr albumu studenta w68560

Program "Sklep"

Promotor: dr inż. Janusz Korniak

PRACA DYPLOMOWA INŻYNIERSKA

Rzeszów 2024

Spis treści

Opis założeń projektu

Program będzie aplikacją konsolową napisaną w języku C#. Głównym celem projektu będzie prowadzenie działaności sprzedaży ubrań. Obsługiwał on będzie funkcjonalność zarówno od strony klienta jak i sprzedawcy. Klient będzie mógł przeglądać dostępne towary, a następnie wybierać i dodawać do koszyka te, które zechce kupić w wybranym rozmiarze i ilości. Ponadto będzie też możliwość sortowania, filtrowania oraz grupowania produktów po cenie. Wszystko to, aby ułatwić klientowi wybór, a sprzedawcy pracę.

Sprzedawca będzie mógł dodawać nowe towary do bazy danych zarówno ręcznie jak i importując produkty z pliku csv. Możliwy będzie również eksport produktów do pliku csv. Dzięki bazie danych Sql, dane będą przechowywane w bezpieczny sposób, a pobieranie ich oraz modyfikacja w trakcie działania programu będzie szybka i wygodna.

Rozdział 1

Wymagania projektu

1.1 Wymagania funkcjonalne

Wymagania funkcjonalne definiują zbiór działań i funkcji, które system powinien realizować, aby spełnić określone cele biznesowe. W przypadku projektu konsolowego sklepu w języku C#, wymagania funkcjonalne obejmują szereg operacji związanych zarówno z zarządzaniem produktami, jak i obsługą zamówień.

1.1.1 Zarządzanie produktami

System umożliwia dodawanie nowych produktów do bazy danych poprzez interfejs konsolowy. Podczas dodawania produktu należy podać jego nazwę, cenę, ilość na stanie oraz inne istotne informacje. Sprzedawca ma możliwość edycji istniejących produktów, w tym zmiany nazwy, ceny, ilości na stanie itp. Zmiany dokonane na produkcie powinny być odzwierciedlone w bazie danych sklepu. System umożliwia usunięcie produktów z bazy danych. Usunięcie produktu powinno być potwierdzone przez użytkownika, aby zapobiec przypadkowemu usunięciu istotnych danych.

1.1.2 Zarządzanie zamówieniami

Klienci mogą przeglądać dostępne produkty i dodać je do koszyka zakupowego. System umożliwia składanie zamówień poprzez interaktywny interfejs konsolowy, gdzie klient może zobaczyć sumaryczną kwotę zamówienia. Użytkownicy mogą edytować zawartość swojego koszyka, zmieniając ilość produktów lub usuwając je. W przypadku zmian w zamówieniu, system powinien na bieżąco aktualizować wartość zamówienia. Klient może zatwierdzić i złożyć zamówienie, co powinno spowodować zapisanie zamówienia w bazie danych.

1.1.3 Autoryzacja i uwierzytelnianie

System umożliwia klientom logowanie do swojego konta lub rejestrację nowego konta w celu dokonywania zakupów. Użytkownicy mogą mieć różne poziomy dostępu w systemie, np. klient, sprzedawca, co wpływa na dostępne funkcje. Wyboru rodzaju konta należy dokonać przy włączeniu aplikacji.

1.2 Wymagania niefunkcjonalne

Wymagania niefunkcjonalne definiują cechy systemu, które nie są związane bezpośrednio z jego funkcjonalnością, ale mają istotne znaczenie dla jego jakości, wydajności, bezpieczeństwa czy użyteczności. Poniżej przedstawione są główne wymagania niefunkcjonalne dla projektu konsolowego sklepu.

1.2.1 Wydajność

System powinien reagować na interakcje użytkownika w sposób szybki i płynny, zapewniając czas odpowiedzi poniżej 1 sekundy dla podstawowych operacji. Aplikacja powinna efektywnie zarządzać zasobami, takimi jak pamięć i procesor, minimalizując zużycie zasobów systemowych.

1.2.2 Bezpieczeństwo

System powinien zapewnić mechanizmy autoryzacji i uwierzytelniania użytkowników, aby zapobiec nieautoryzowanemu dostępowi do danych. W zależności od roli w systemie użytkownik będzie miał dostęp tylko do swoich funkcjonalności. Dane użytkowników (takie jak dane osobowe i hasło) powinny być przechowywane w sposób bezpieczny, zapewniający poufność i integralność.

1.2.3 Użyteczność

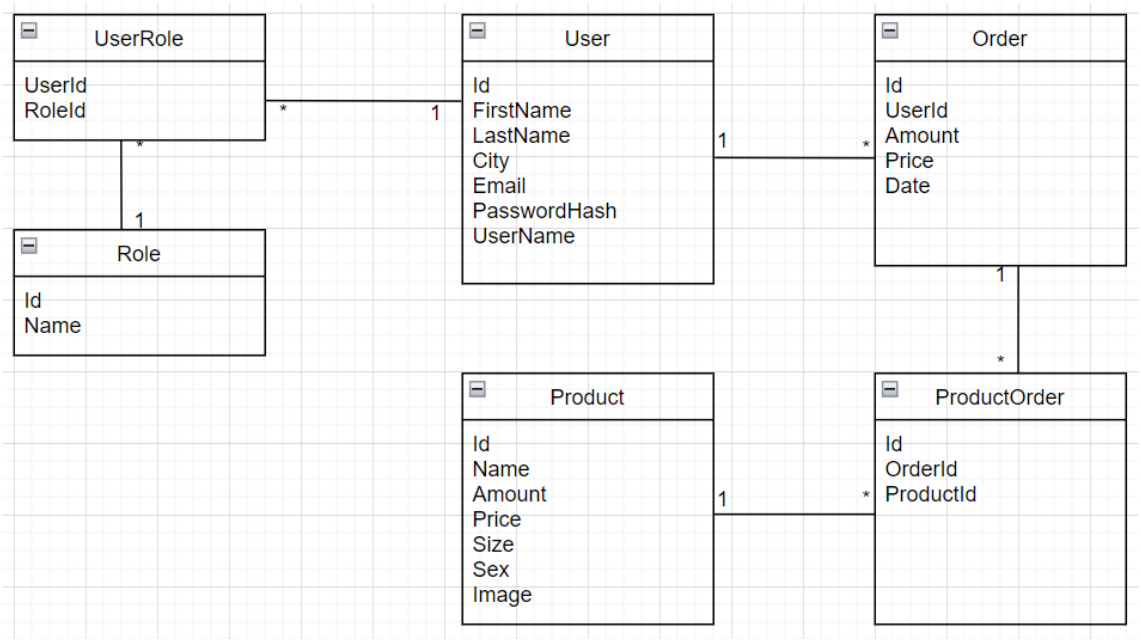
Interfejs konsolowy powinien być przejrzysty i intuicyjny, umożliwiając użytkownikom łatwe korzystanie z funkcji sklepu. Komunikaty o błędach powinny być czytelne i informatywne, aby użytkownicy mogli łatwo zrozumieć przyczyny problemów i podjąć odpowiednie działania naprawcze.

Rozdział 2

Opis struktury projektu

Diagram klas reprezentuje schemat powiązań pomiędzy klasami i encjami w bazie danych. Pola, które znajdują się w tabelach, są właściwościami w klasach i zawierać będą wartości, które będą zapisane w bazie. Na łączeniach pomiędzy tabelami widać rodzaj relacji.

2.1 Diagram klas



Rysunek 2.1: Diagram klas programu Sklep

Powyższy diagram przedstawia klasy, które reprezentują dane użytkownika, produktów oraz zamówień. Klasy odpowiadające na dane klienta oraz role w systemie zostaną wygenerowane przez bibliotekę Identity, która służy do autoryzacji i autentykacji użytkowników. Udostępnia ona również gotowe mechanizmy logowania, rejestracji i wiele innych. Użytkownik połączony jest z rolą w relacji wiele do wielu, dlatego też istnieje klasa łącząca UserRole.

W takiej samej relacji znajdują się klasa Product oraz Order. Jeden produkt może znajdować się w wielu zamówieniach i tak samo jedno zamówienie może zawierać wiele produktów, przez co w tym przypadku również zastosowano powiązanie poprzez klasę łączącą. Użytkownik może mieć wiele zamówień w sklepie, a jedno zamówienie przypisane jest tylko do jednego klienta, przez co połączony jest w relacji jeden do wielu.

Klasa użytkownika zawiera podstawowe informacje o kliencie, poprzez które będzie można go zweryfikować. Produkt zawiera informacje takie jak nazwa, ilość sztuk dostępna w sklepie, cena za sztukę, rozmiar, płeć, oraz obrazek, który w formie ascii artu będzie przedstawiał dany artykuł na konsoli. Natomiast tabela Order przechywać będzie dane o zamówieniu takie jak kto i kiedy złozył zamówienie ilość produktów oraz całkowita kwota zamówienia.

Rozdział 3

Opis techniczny projektu

3.1 Wykorzystana technologia

Projekt sklepu to aplikacja konsolowa napisana w języku C#. Jej głównym założeniem jest podział funkcjonalności, na te które są w uprawnieniach sprzedawcy, a także, te do których dostęp ma klient. Obsługa menu programu polega na przechodzeniu pomiędzy opcjami przy pomocy strzałek co ułatwia korzystanie z interface'u.

W celu uatrakcyjnienia wyglądu, użyto kolorowania składni oraz obrazków reprezentujących produkty sklepu odzieżowego przy pomocy ASCII art. Mechanizm koszyka, do którego klient może dodawac wybrane produkty, aby później móc je kupić zaimplementowano korzystając z klasy statycznej. Przy pomocy polecenia *Console.SetCursorPosition(y, x)* zmieniane są miejsca kursora na konsoli, dzięki czemu pisanie tekstu można było zaczynać z dowolnego miejsca.

Głównym elementem programu, który został wykorzystany wielokrotnie przy każdorazowym przejściu do kolejnego menu była pętla while. We wspomnianej pętli wyświetlane zostały menu, tak długo, aż spełniony został warunek przerwania, który zachodził, gdy zaznaczona, a następnie zatwierdzona została opcja *Wyjście*.

Aby zapewnić w aplikacji korzystanie z tych samych danych, które znajdowały się podczas ostatniego użytkownia przed jej zamknięciem, zastosowano bazę danych SQL. Jest to relacyjna baza danych, która przechowuje informacje w relacji między tabelami. W tym celu zainstalowano bibliotekę EntityFrameworkCore, w wersji 6, która kompatybilna jest z również 6 wersją .NET, który wykorzystany został do stworzenia danej aplikacji.

Sprzedawca ma możliwość generowania raportu PDF z asortymentu dostępnego w sklepie. Funkcjonalność tą zaimplementowano przy pomocy biblioteki iText7, która specjalizuje się w tworzeniu plików pdf. Do projektu dołączono ją poprzez NuGet.

3.2 Baza danych

W programie klasą odpowiedzialną za konfigurację połączenia bazy danych z aplikacją jest klasa *ApplicationDbContext*. Dziedziczy ona po klasie DbContext, dzięki czemu można nadpisać metodę OnConfiguring. W jej ciele, znajduje się konfiguracja połączenia z lokalną bazą danych podaną w adresie *ConnectionString*. Użyto w niej metodę *UseLazyLoadingProxies*. Dzięki temu, kiedy nastąpi próba dostępu do powiązanych encji, będą one leniwie ładowane, nawet w kontekście asynchronicznym. Co za tym idzie, w modelach, które opisują budowę poszczególnych tabel zastosowano słówko *virtual*, aby zaciągać dane obiektów będanych w relacji.

Logika działania programu zawarta jest w klauzuli *using*, w której tworzony jest kontekst bazy danych, co umożliwia do nich dostęp. W jej obrębie stworzono obiekty, które reprezentują repozytoria. Każde repozytorium pełni funkcję dostępu do poszczególnych typów danych zawartych w tabelach bazy danych. Oddzielają one kod logiki realizujący operację od dostępu do informacji.

Rozdział 4

Harmonogram prac

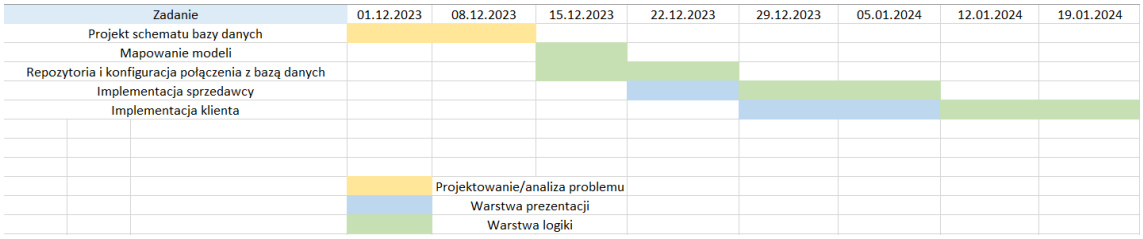
4.1 Diagram Gantta

Diagram Gantta to narzędzie wykorzystywane do przedstawiania harmonogramu zadań w czasie. Składa się z osi poziomej reprezentującej czas oraz osi pionowej przedstawiającej zadania. Każde zadanie jest przedstawiane za pomocą prostokąta, a jego długość odpowiada czasowi trwania zadania.

Na diagramie Gantta reprezentującym harmonogram pracy podczas tworzenia aplikacji sklepu znajduje się również legenda, która opisuje kolory użyte do prezentacji zadania. Kolor pomarańczowy reprezentuje prace projektowe oraz analityczne. Zazwyczaj takie prace wykonywane są jako pierwsze jeszcze przed implementacją kodu. Na podstawie ustaleń w tym procesie powstaje architektura projektu.

Kolorem niebieskim oznaczono prace frontowe odpowiadajace za wygląd aplikacji. Najczęściej prace te wykonywane są jednocześnie z pracami backendowymi. Jednak w przypadku aplikacji konsolowej najpierw wykonano prace odpowiadające za wizualną formę programu.

Kolorem zielonym zaznaczono na diagramie prace nad implementacją operacji i mechanizmów logicznych.



Rysunek 4.1: Diagram Gantta

Rozdział 5

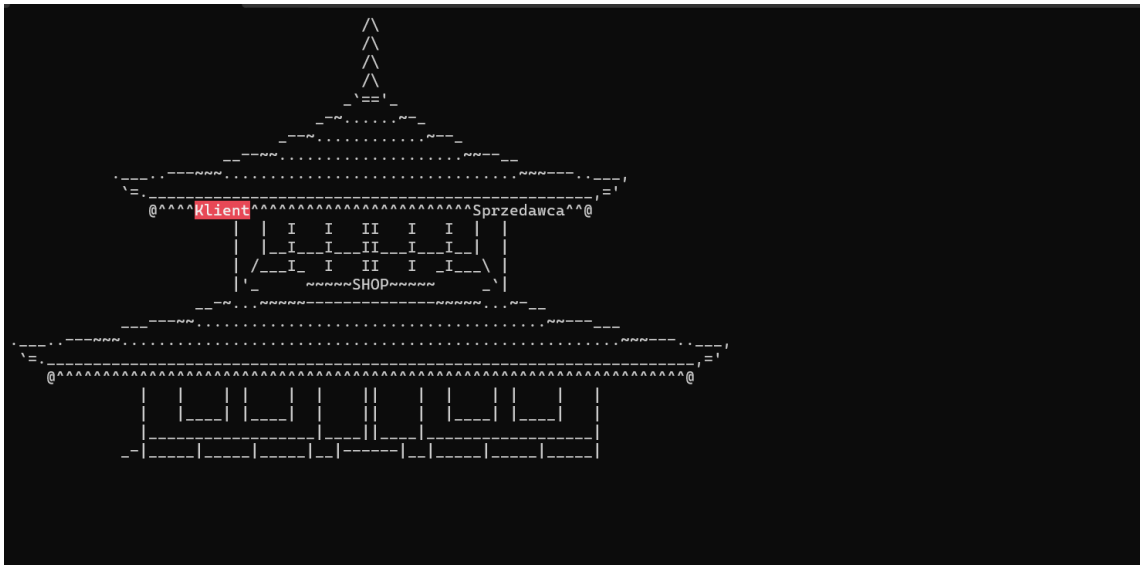
Prezentacja warstwy użytkowej

Projekt sklepu został w całości zrealizowany jako aplikacja konsolowa. Oznacza to, że zarówno logika działania programu oraz jej wygląd zostały napisane w języku C#. Jest to także jednoznaczne z tym, że poruszanie się po programie przez użytkownika odbywa się w konsoli bez dodatkowego interfejsu graficznego, przy pomocy jedynie klawiatury.

Poniżej zostaną przedstawione zrzuty ekranu z najważniejszych funkcjonalności w programie, aby zaprezentować w ten sposób wygląd oraz pogląd działania aplikacji.

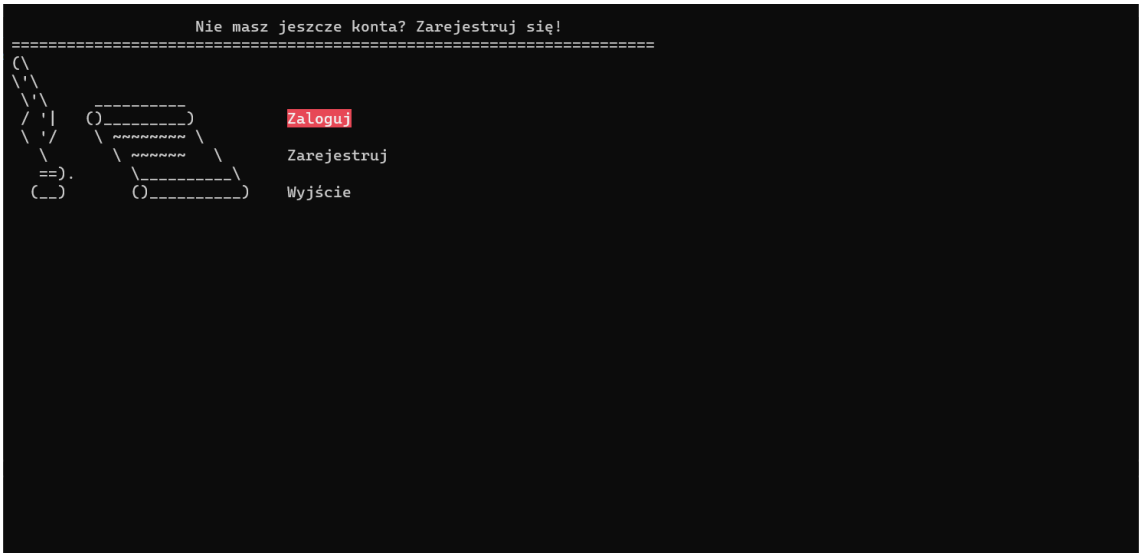
Po uruchomieniu aplikacji pierwszym ekranem jest ekran powitalny, który jednocześnie jest miejscem wyboru roli użytkownika, który będzie korzystał z programu. Z programu można korzystać jako sprzedawca lub klient.

5.1 Sprzedawca



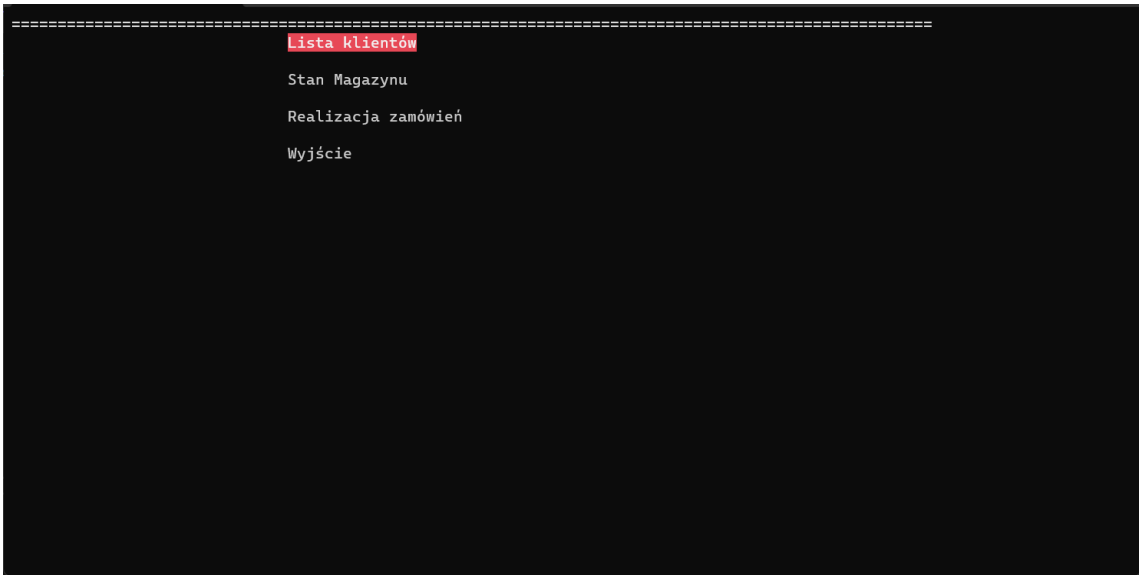
Rysunek 5.1: Wybór roli użytkownika

Po wyborze roli pojawia się ekran, w którym należy zdecydować pomiędzy zalogowaniem się lub rejestracją. Wygląd tych ekranów jest identyczny zarówno dla sprzedawcy jak i klienta. W zależności od wyboru wyświetlają się formularze gdzie kolejno należy podawać kolejne informacje. W przypadku rejestracji w systemie tworzy się konto użytkownika z rolą odpowiadającą wcześniejszemu wyborowi.



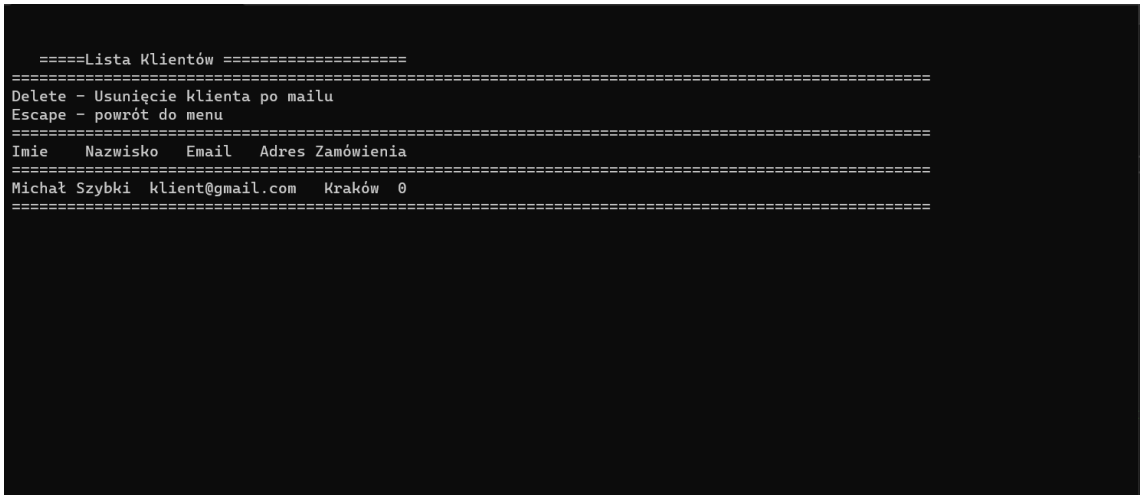
Rysunek 5.2: Ekran logowania i rejestracji

Logując się do aplikacji jako sprzedawca ma on do wyboru opcje widoczne na *Rysunku 5.3*.



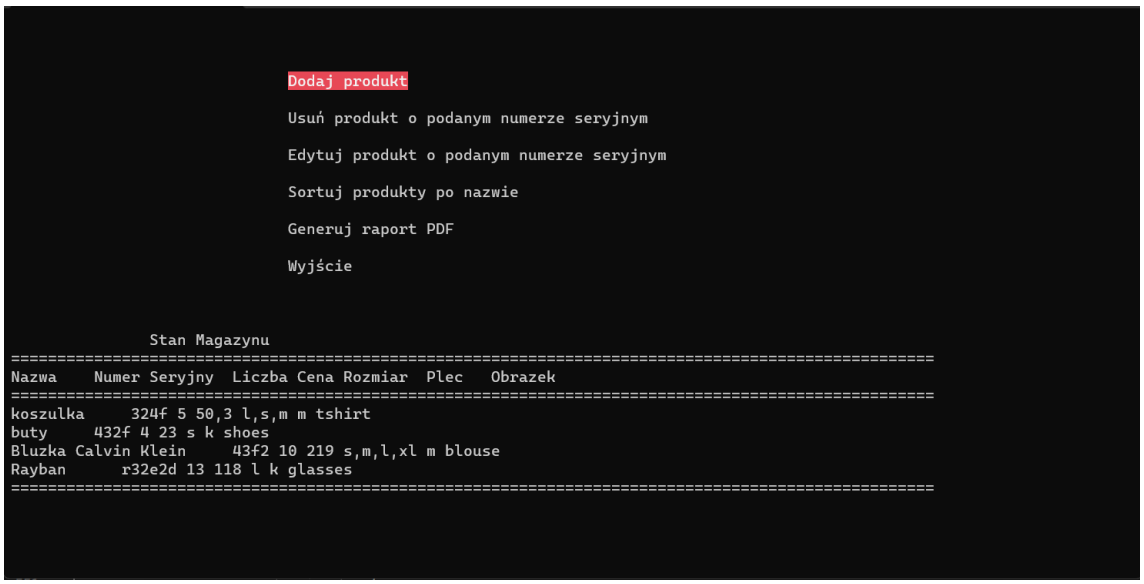
Rysunek 5.3: Menu sprzedawcy

Po przejściu do opcji *Lista klientów*, sprzedawca ma możliwość usunięcia klienta po adresie e-mail. Na liście wyświetlają się tylko użytkownicy z rolą klienta. Widać ich podstawowe dane personalne oraz liczbę zamówień złożonych w systemie. Ekran ten zaprezentowany został na *Rysunku 5.4*.



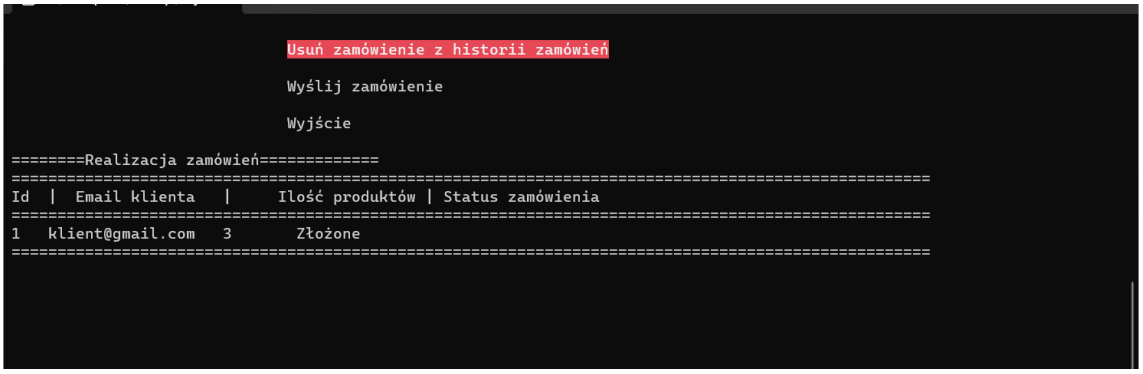
Rysunek 5.4: Lista klientów

Kolejną możliwą opcją jest wyświetlenie stanu magazynu, czyli wszystkich dostępnych produktów w sklepie. Ponadto, w tym menu można również dodać nowy produkt, usunąć go lub edytować. Możliwe jest również posortowanie na liście produktów po nazwie. Jeśli sprzedawca uzna, że chce na przykład wydrukować daną listę lub ją wyeksportować, ma możliwość wygenerowania pliku pdf z dostępnymi w sklepie towarami.



Rysunek 5.5: Menu stan magazynu

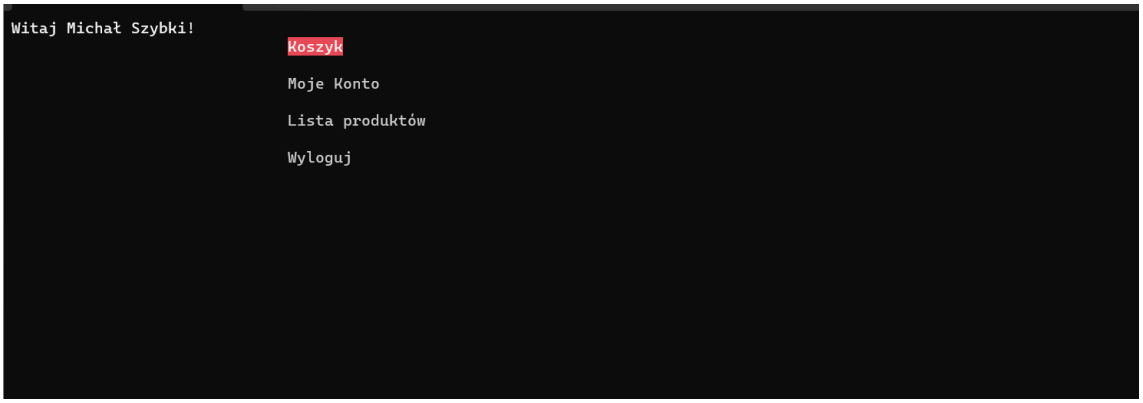
W otatniej zakładce od strony sprzedawcy, mianowicie *Realizacja zamówień*, wyświetlane są wszystkie zamówienia w systemie zarówno te zrealizowane jak i te, które dopiero zostały złożone. Sprzedawca może usunąć wybrane zamówienie z listy zamówień lub zrealizować zamówienie. Metoda ta zmienia status zamówienia z *złożone* na *zrealizowane*.



Rysunek 5.6: Menu realizacja zamówień

5.2 Klient

Klient po poprawnym zalogowaniu do systemu, oprócz wiadomości powitalnej w lewym górnym rogu aplikacji ma do wyboru jedną z kilku opcji widocznych na *Rysunku 5.7*.



Rysunek 5.7: Menu klienta

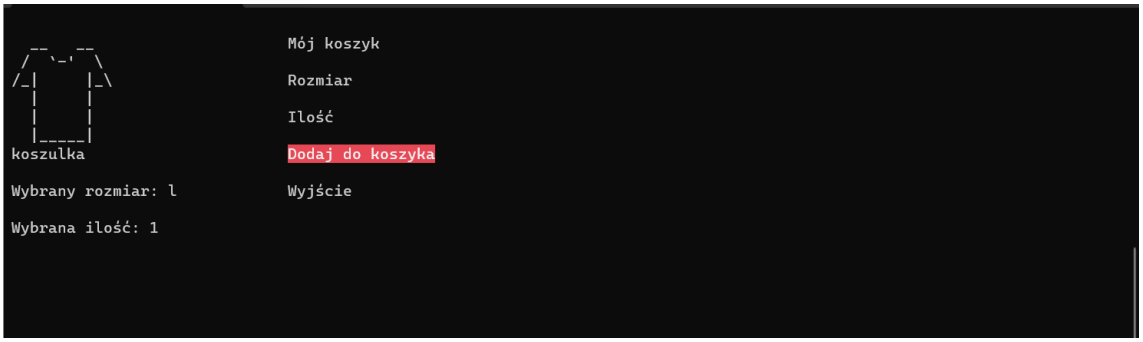
Aby jednak w pierwszych dwóch pozycjach z menu wyświetliły się jakieś dane w przypadku nowo utoworzonego konta, należy najpierw przejść do opcji *Lista produktów*. Aplikacja przekieruje użytkownika do widoku z dostępnymi produktami w sklepie, które następnie można sortować lub wyszukać po nazwie.

Rysunek 5.8: Przeglądanie listy produktów w menu klienta

Rezultat wyszukiwania produktu po nazwie z frazą *calvin* widoczny na *Rysunku 5.9*.

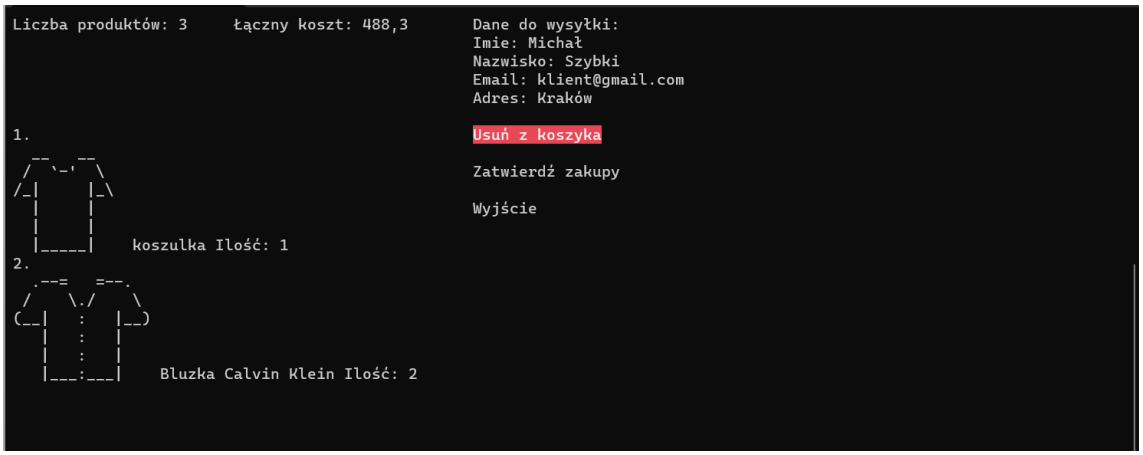
Rysunek 5.9: Wyszukiwanie produktu po nazwie

Po wybraniu upatrzonego produktu, użytkownik może wybrać rozmiar oraz ilość sztuk. Jeśli wszystkie parametry zostały wybrane poprawnie, klient może dodać produkt do koszyka.



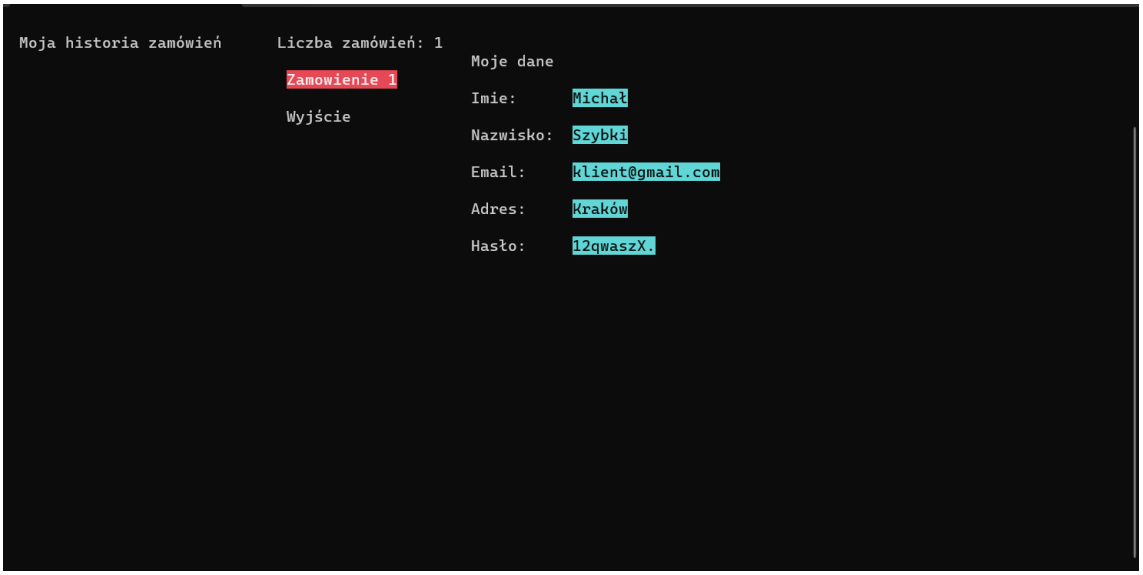
Rysunek 5.10: Dodawanie produktu do koszyka

W koszyku znajdują się produkty, które klient dodał tam podczas działania aplikacji. Są one w nim widoczne tylko na czas działania aplikacji, czyli na czas sesji. W koszyku wyświetlana jest zsumowana liczba wszystkich produktów oraz zsumowana kwota zakupów. Klient widzi również dane do wysyłki. Są to dane podane podczas procesu rejestracji. W tym miejscu klient może jeszcze usunąć wybrany produkt z koszyka podając jego numer z listy w koszyku. Użytkownik może również wybrać opcję potwierdzenia zakupów, która utworzy zamówienie w systemie, aktualizując liczbę produktów w bazie danych oraz wyczyści koszyk klienta.



Rysunek 5.11: Podgląd koszyka klienta

Ostatnią opcją, która nie została jeszcze opisana z menu klienta z *Rysunku 5.7*, jest zakładka *Moje konto*. Klient może tu zobaczyć swoje dane, w tym hasło. Co więcej, opcja ta wyświetla listę zamówień złożonych przez klienta w sklepie. Klient może wybrać konkretne zamówienie z listy i podejrzeć jakie produkty kupił w danym zamówieniu.



Rysunek 5.12: Historia zamówień klienta oraz dane jego konta

Rozdział 6

Podsumowanie

6.1 Plany rozbudowy aplikacji

Po zakończeniu pierwszej fazy projektu "Sklep", jedną ze ścieżek na rozwój aplikacji jest wprowadzenie nowych funkcjonalności oraz usprawnień mających na celu zwiększenie atrakcyjności, użyteczności i efektywności systemu.

Aktualny system oparty jest na interfejsie konsolowym, jednak w planach jest wprowadzenie interfejsu graficznego (GUI), co znacznie poprawi wrażenia użytkownika. Przejście na interfejs z grafiką umożliwi bardziej intuicyjną nawigację oraz lepszą prezentację produktów. Jedną z możliwości jest stworzenie aplikacji okienkowej w rozwiązaniu WPF. Przy zmianie na tego typu aplikację, logika biznesowa pozostanie bez zmian. Trzeba ją jednak będzie dopasować do nowego interfejsu graficznego.

Kolejną możliwością rozwoju jest przejście na wiele kategorii sprzedawanych produktów zwiększając asortyment lub podzial istniejących produktów na kategorie. Ułatwi to użytkownikom wyszukiwanie produktów oraz uporządkuje produkty w aplikacji.

Podążając za trendem zaobserwowanym w innych aplikacjach tego typu, można utworzyć mechanizm systemu oceny i recenzji. Dodanie funkcji, która umożliwi klientom wystawianie ocen i recenzji produktów, nie tylko zwiększy zaufanie do sklepu, ale także dostarczy istotnych informacji zwrotnych dla sprzedawcy. Na podstawie tych danych system będzie mógł wyświetlać spersonalizowane sugestie produktów i je reklamować.

6.2 Podsumowanie zrealizowanych prac

Podczas tworzenia projektu "Sklep"w języku C# w postaci programu konsolowego, głównym celem było stworzenie kompleksowego systemu obsługującego zarówno perspektywę sprzedawcy, jak i klienta. Dzięki zaimplementowanym funkcjonalnościom, aplikacja umożliwia sprawną obsługę procesów biznesowych, usprawnia zarządzanie asortymentem oraz zapewnia przyjazne i intuicyjne środowisko dla klientów dokonujących zakupów. Prace zostały zakończone sukcesem, a uzyskane rezultaty odpowiadają stawianym przed projektem celom.

Bibliografia

[1] https://stackoverflow.com/ z dnia 29.12.2023

[2] https://pasja-informatyki.pl/ z dnia 03.01.2024

[3] Joseph Albahari, Ben Albahari, *C# 9.0 in a Nutshell: The Definitive Reference*, O'Reilly Media, Sebastopol, Kalifornia, USA, 2021.

[4] Bartłomiej Filipek, *Programowanie w C#. Teoria i praktyka*, Helion, Gliwice 2019.

[5] Andrzej Ślęzak, Wojciech Ślęzak, *C# Od Podstaw. Wydanie V*, Helion, Gliwice 2018.