

Kacper Jess  
Eryk Węclawiak

***DOKUMENTACJA TECHNICZNA PROJEKTU***

***WESS-POS APLIKACJA OKIENKOWA WSPOMAGAJĄCA SPRZEDAŻ W SKLEPACH  
STACJONARNYCH***

***PRZEDMIOT: PROGRAMOWANIE ZAAWANSOWANIE***

## Spis treści

Tematyka projektu.....	2
Diagram przypadków użycia.....	2
Diagram klas.....	3
Opis techniczny projektu.....	3
Potencjalne możliwe problemy i zagrożenia.....	4

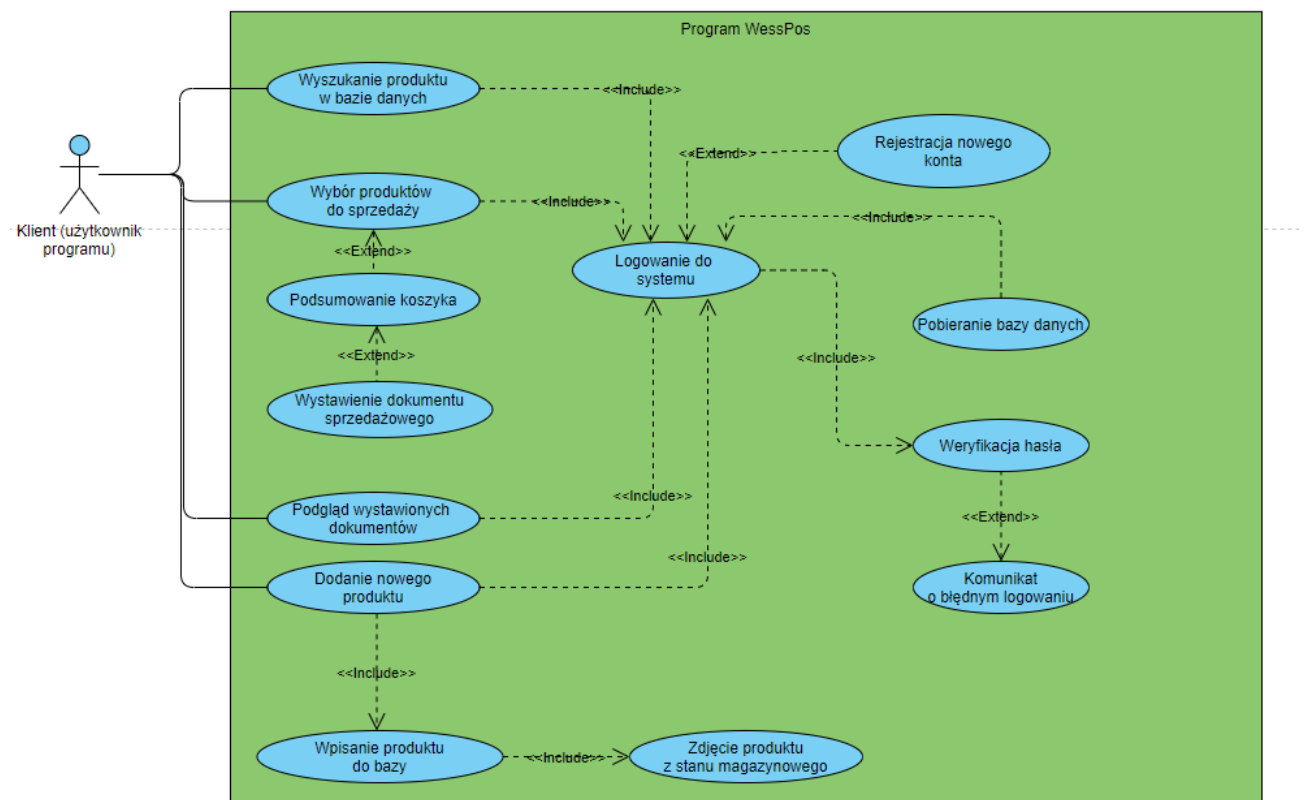
## Tematyka projektu

Tematem projektu jest stworzenie oprogramowania, które ma na celu wspomóc pracę obsługi sklepu stacjonarnego w zakresie organizacji i sprzedaży produktów. Program wymaga stałego dostępu do Internetu, w celu zapewnienia dostępu do bazy danych z każdego miejsca.

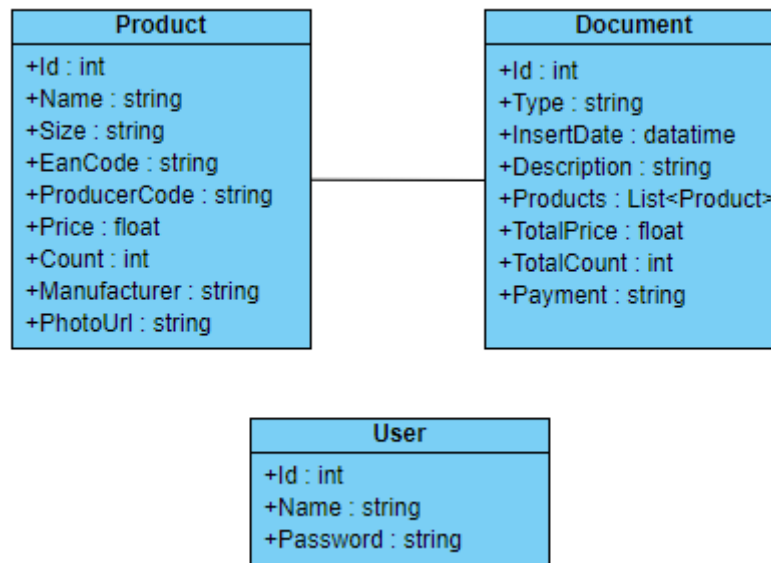
Program ma za zadanie:

- Kontrola dostępu – użytkownik musi zarejestrować nowe konto lub zalogować się na istniejące, aby móc korzystać z oprogramowania.
- Rejestracja sprzedaży i kontrola stanów – wystawienie dokumentu sprzedażowego wiąże się z zmniejszeniem stanu magazynowego punktu sprzedażowego. Użytkownik podczas obsługi klienta może wybrać sposób płatności oraz typ wystawianego dokumentu.
- Wgląd do dokumentów historycznych – program umożliwia wyszukanie wystawionych dokumentów opierając się na wybranych kryteriach.
- Wprowadzanie nowych towarów – wprowadzanie nowych towarów wymaga wprowadzenia podstawowych informacji o produkcie.

## Diagram przypadków użycia



## Diagram klas



## Opis techniczny projektu

Program jest aplikacją desktopową, która zrealizowana została wykorzystując framework Windows Forms oraz Entity Framework. Wykorzystanie Windows Forms umożliwiło szybkie i łatwe stworzenie layoutu aplikacji, natomiast Entity Framework odpowiada za stworzenie bazy danych zgodnie z wskazanymi klasami oraz pozwala na operowanie bazą (insert, update, select).

Aby zachować elastyczność aplikacji i możliwość dostępu z każdego miejsca, sama baza przechowywana jest w formie pliku za pomocą SQLite oraz wysyłana jest na serwer FTP. Takie rozwiązanie zostało zaimplementowane z uwagi na wykorzystanie SQLite. Baza podczas uruchamiania aplikacji zostaje pobierana z serwera FTP oraz wysyłana podczas zamykania aplikacji. Z uwagi na te operacje aplikacja wymaga stałego połączenia z Internetem.

Aplikacja dba o bezpieczeństwo wrażliwych danych poprzez wymaganie logowania przy każdorazowym włączeniu aplikacji. Do hashowania wykorzystany został algorytm kryptograficzny MD5, który zapewnia odpowiedni poziom bezpieczeństwa aplikacji.

Problemem podczas tworzenia aplikacji było operowanie plikiem z bazą danych. Wysyłka pliku db nie zostało zrealizowane w tym samym rozwiązaniu z uwagi na dostęp do używanego pliku. Zostało to zaimplementowane w osobnym programie uruchamianym każdorazowo podczas zamykania okna głównego (głównego okna sprzedażowego, okna logowania, okna rejestracji).

Użytkownik, podczas użytkowania z aplikacji, wprowadza dane do bazy bezpośrednio tylko w momencie dodawania nowego produktu. W tym miejscu od użytkownika wymaga się podania wszystkich niezbędnych danych dotyczących dodawanego produktu.

W pozostałych opcjach użytkownik wprowadza tylko odpowiednie frazy (podczas wyszukiwania dokumentów lub produktów) albo wybiera odpowiednią opcję z puli podanych przez aplikację.

W aplikacji wprowadzona została kontrola stanów magazynowych. Po wystawieniu dokumentu sprzedażowego (paragonu lub faktury), program odejmuje od stanu magazynową odpowiednią ilość produktu znajdującego się na dokumencie.

Użytkownik ma wgląd do bazy za pośrednictwem aplikacji. Użytkownik może wybrać jaki dokument lub produkt chce wyszukać. Następnie serwer przetworzy zapytanie, a następnie odpowie odpowiednim pakietem danych.

## Potencjalne możliwe problemy i zagrożenia

Program w przypadku braku dostępu do Internetu nie będzie w stanie pobrać, ani wysłać, bazy danych z serwera FTP, a co za tym idzie, aplikacja nie będzie funkcjonowała prawidłowo.

Zagrożeniem może być przerwanie dostępu do Internetu podczas działania programu. Baza danych z wprowadzonymi nowymi danymi nie zostanie wysłana na serwer FTP. Potencjalnym rozwiązaniem tego problemu jest zastosowanie innego typu bazy danych (np. SQL Server) i okresowa wysyłka danych do bazy (weryfikacja bazy danych po stronie klienta pod kątem zawartości bazy po stronie serwera). Rozwiązanie to zniwelowałoby konieczność usuwania bazy danych z dysku lokalnego klienta.