**Temat: Sklep z odzieżą**

**1 etap - Specyfikacja wymagań**

**Autorzy:**   
**Mateusz Błach 264010, Michał Lewandowski 264458, Jakub Jakubowicz 263925**

**Wstęp:**

Nasz projekt zakłada stworzenie aplikacji bazodanowej, która usprawni zarządzanie sklepem z odzieżą online. Aplikacja ta ma na celu ułatwić klientom przeglądanie i zakup produktów, a jednocześnie umożliwić administratorom i pracownikom sprawną obsługę sklepu. Aplikacja zostanie podzielona na trzy główne grupy użytkowników: administratorów, pracowników i klientów, z każdą grupą posiadającą dostęp do określonych funkcjonalności. Dla administratorów, głównym celem będzie efektywne zarządzanie produktami, zamówieniami oraz kontami klientów. Pracownicy będą odpowiedzialni za akceptację i anulację zamówień, a klienci będą mogli wygodnie przeglądać i zamawiać produkty.

# Wymagania funkcjonalne:

## Dla administratora:

1. **Logowanie:** Administrator ma możliwość zalogowania się do systemu za pomocą unikalnego identyfikatora (login) i hasła
2. **Zarządzanie produktami:** Administrator ma dostęp do funkcji dodawania, edytowania i usuwania produktów w bazie danych.
3. **Zarządzanie zamówieniami:** Administrator powinien być w stanie przeglądać, edytować i usuwać.
4. **Zarządzanie kontami:** Administrator ma prawo tworzyć, edytować i usuwać konta klientów sklepu.
5. **Backup i odzyskiwanie danych:** Administrator ma możliwość stworzenia backupu stanu produktów, użytkowników oraz zamówień.

## Dla pracownika:

1. **Przeglądanie produktów:** Pracownik może przeglądać produkty w bazie, sprawdzać dostępność, filtrowanie produktów po różnych parametrach.
2. **Logowanie:** Pracownik ma możliwość zalogowania się do systemu za pomocą unikalnego identyfikatora (login) i hasła.
3. **Akceptacja zamówienia:** Każde zamówienie złożone przez klienta musi zostać zaakceptowane przez pracownika.
4. **Anulowanie zamówienia:** Każde zamówienie złożone przez klienta może zostać anulowane przez pracownika.

## Dla klienta:

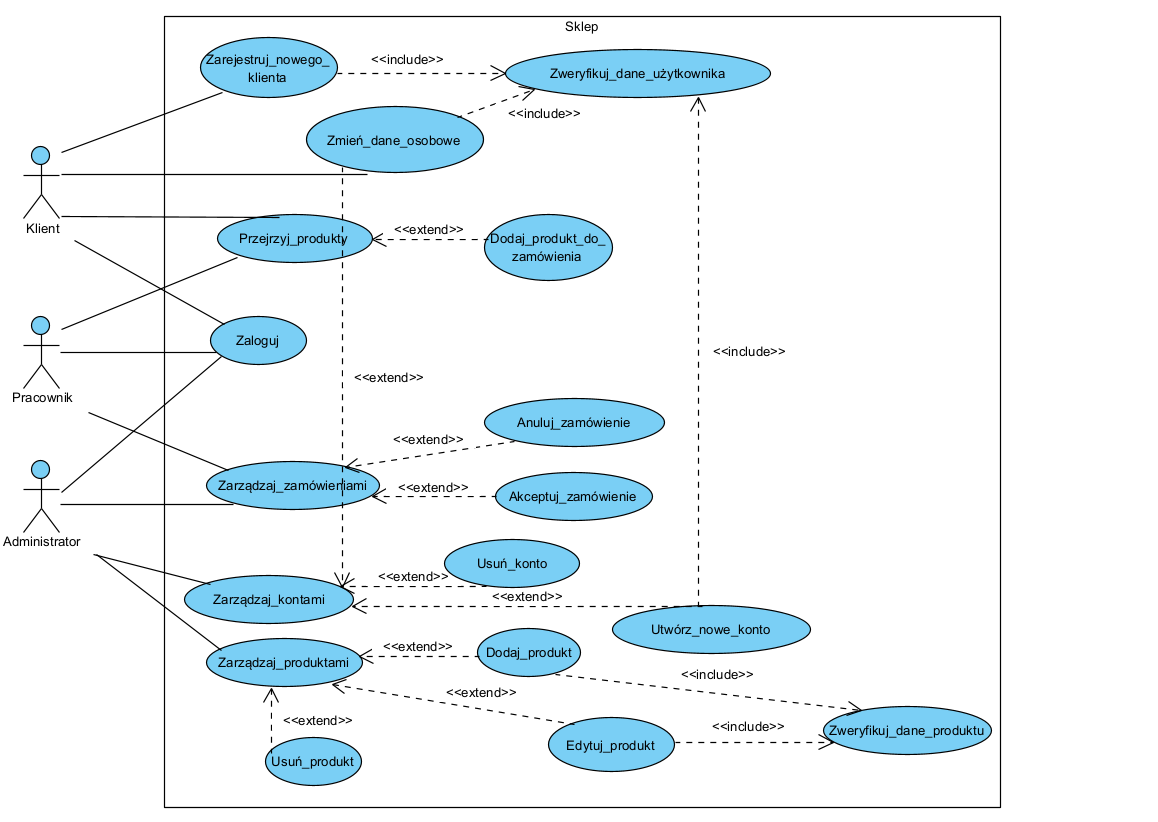
1. **Zmiana danych osobowych:** Klienci mogą usunąć swoje konto z bazy oraz edytować swoje dane.
2. **Logowanie:** Klienci mogą logować się za pomocą loginu i hasła.
3. **Rejestracja:** Klienci mogą stworzyć konto.
4. **Przeglądanie produktów:** Klienci mogą przeglądać produkty według kategorii, cen i marki.
5. **Dodawanie do zamówienia:** Klienci mogą dodawać produkty do zamówienia.
6. **Składanie zamówień:** Klienci mogą składać zamówienia podając dane dostawy i płatności.
7. **Śledzenie zamówień:** Klienci mogą sprawdzać status swoich zamówień i historię zakupów.

## Dla wszystkich użytkowników:

1. **Weryfikacja danych logowania:** Weryfikacja czy konto istnieje w bazie danych dla podanych danych logowania.
2. **Weryfikacja wprowadzonych danych:** Weryfikacja wprowadzonych danych.

# Wymagania niefunkcjonalne:

1. **Autoryzacja operacji:** Każda operacja na bazie danych jest dostępna w zależności od typu konta użytkownika.
2. **Walidacja wprowadzonych danych:** Weryfikacja czy podane dane przez użytkownika są prawidłowe.
3. **Wykorzystanie procedur**: Wszystkie zapytania kierowane z aplikacji do bazy danych będą wykonywane poprzez procedury.
4. **Hashowanie haseł:** Zastosowanie funkcji haszujących do bezpiecznego przechowywania haseł w bazie danych.
5. **Jedno konto administratora:** Istnienie jednego wbudowanego konta administratora w bazie danych
6. **Wymaganie logowania:** Konieczność logowania się dla klientów korzystających z aplikacji bazodanowej
7. **Akceptacja zamówień**: Każde zamówienie musi zostać zaakceptowane przez odpowiedniego pracownika przed dalszym przetwarzaniem.

Diagram przypadków użycia:

**Scenariusze przypadków użycia:**  
**Nazwa PU: Zarejestruj\_nowego\_klienta**

**Cel: Zarejestrowanie nowego klienta**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Wprowadzenie unikalnej nazwy użytkownika**

**Scenariusz:**

1. **Należy podać imię, nazwisko, adres, nazwę użytkownika i hasło.**
2. **Należy sprawdzić czy dany użytkownik już nie istnieje i czy hasło jest odpowiednio skonstruowane poprzez PU Zweryfikuj\_wprowadzone\_dane**
3. **Jeżeli wprowadzone dane nie są prawidłowe należy zakończyć PU, w przeciwnym wypadku należy zapisać dane w bazie danych.**

**Nazwa PU: Zweryfikuj\_dane\_użytkownika**

**Cel: Zweryfikowanie poprawności wprowadzonych danych**

**Warunki początkowe: Jest uruchamiany z następujących PU: Zarejestruj\_nowego\_klienta, Utwórz\_nowe\_konto, Zmień\_dane\_osobowe**

**Warunki końcowe: Zwraca wynik określający czy wprowadzone dane są prawidłowe**

**Scenariusz:**

1. **Porównuje nazwę użytkownika z nazwami użytkownika istniejącymi już w bazie danych.**
2. **W przypadku znalezienia użytkownika o podanej nazwie zwraca te nazwę użytkownika i kończy przeszukiwanie bazy danych.**
3. **W przypadku nieznalezienia podanej nazwy użytkownika zwraca wynik negatywny**

**Nazwa PU: Zmień\_dane\_osobowe**

**Cel: Zmiana danych osobowych klienta**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Wprowadzona nazwa użytkownika musi być unikalna**

**Scenariusz:**

1. **Należy podać dane, które chcemy zmienić**
2. **Jeżeli wśród zmienionych danych występuje nazwa użytkownika to uruchamia PU Zwerfikuj\_dane\_użytkownika.**
3. **Jeżeli wprowadzone dane nie są prawidłowe należy zakończyć PU, w przeciwnym wypadku należy zapisać dane w bazie danych.**

**Nazwa PU: Przejrzyj\_produkty**

**Cel: Przegląd dostępnych produktów**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Kryteria przeglądania muszą być poprawne**

**Scenariusz:**

1. **Użytkownik aplikacji podaje określone kryteria wedle których aplikacja filtruje produkty.**
2. **Zgodnie z określonymi kryteriami zwracana jest lista produktów.**

**Nazwa PU: Dodaj\_produkt\_do\_zamówienia**

**Cel: Dodanie wybranego produktu do zamówienia**

**Warunki początkowe: Jest uruchamiany z PU Przejrzyj\_produkty**

**Warunki końcowe: Wybrany produkt musi być dostępny**

**Scenariusz:**

1. **Użytkownik wybiera produkt**
2. **Jeżeli produkt jest dostępny to dodaje go do zamówienia, w przeciwnym wypadku kończy PU**

**Nazwa PU: Zaloguj**

**Cel: Zalogowanie użytkownika**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Użytkownik musi podać prawidłowe dane logowania**

**Scenariusz:**

1. **Należy podać nazwę użytkownika oraz hasło**
2. **Sprawdzana jest poprawność wprowadzonych danych**
3. **Jeżeli podano prawidłowe dane logowania to użytkownik ma dostęp do aplikacji, w przeciwnym wypadku kończymy PU**

**Nazwa PU: Zarządzaj\_zamówieniami**

**Cel: Zarządzanie zamówieniami**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Użytkownik musi być administratorem lub pracownikiem**

**Scenariusz:**

1. **Użytkownik wybiera zamówienie, które może zaakceptować korzystając z PU Akcpetuj\_zamówienie lub anulować korzystając z PU Anuluj\_zamówienie**
2. **W przypadku edycji zamówienia należy zapisać zmiany w bazie danych, w innym przypadku należy zakończyć PU**

**Nazwa PU: Anuluj\_zamówienie**

**Cel: Anulowanie zamówienia**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_zamówieniami**

**Warunki końcowe: Zamówienie musi znajdować się w bazie**

**Scenariusz:**

1. **W przypadku anulowania zamówienia jest ono usuwane z bazy danych**
2. **W innym wypadku należy zakończyć PU**

**Nazwa PU: Akceptuj\_zamówienie**

**Cel: Akceptacja zamówienia**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_zamówieniami**

**Warunki końcowe: Zamówienie musi istnieć**

**Scenariusz:**

1. **W przypadku akceptacji zamówienia jest ono dodawane do bazy danych**
2. **W innym wypadku należy zakończyć PU**

**Nazwa PU: Zarządzaj\_kontami**

**Cel: Zarządzanie kontami użytkowników**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Użytkownik musi być administratorem**

**Scenariusz:**

1. **Użytkownik wybiera konto i może edytować dane osobowe korzystając z PU Zmień\_dane\_osobowe lub usunać konto korzystając z PU Usuń\_konto**
2. **W przypadku, kiedy konto nie istnieje użytkownik może je utworzyć korzystając z PU Utwórz\_nowe\_konto**
3. **W wypadku edycji/utworzenia konta należy zapisać zmiany w bazie danych w innym przypadku należy zakończyć PU**

**Nazwa PU: Usuń\_konto**

**Cel: Usunięcie konta użytkownika**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_kontami**

**Warunki końcowe: Wybrane konto musi znajdować się w bazie**

**Scenariusz:**

1. **W przypadku usunięcia konta jest ono usuwane z bazy danych**
2. **W przeciwnym wypadku należy zakończyć PU**

**Nazwa PU: Utwórz\_nowe\_konto**

**Cel: Utworzenia nowego konta użytkownika**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_kontami**

**Warunki końcowe: Wprowadzone dane do konta muszą być unikalne**

**Scenariusz:**

1. **Należy zweryfikować czy podane dane do konta nie znajdują się już w bazie danych za pomocą PU Zweryfikuj\_dane\_użytkownika**
2. **Jeżeli wprowadzone dane są unikalne należy zapisać nowe konto w bazie danych, w przeciwnym wypadku należy zakończyć PU**

**Nazwa PU: Zarządzaj\_produktami**

**Cel: Zarządzanie produktami**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Użytkownik musi być administratorem**

**Scenariusz:**

1. **Użytkownik wybiera produkt, który może edytować korzystając z PU Edytuj\_produkt lub usunąć korzystając z PU Usuń\_produkt**
2. **Jeżeli produkt nie istnieje użytkownik może go utworzyć korzystająć z PU Dodaj\_produkt**
3. **W przypadku edycji/utworzenia produktu należy zapisać zmiany w bazie danych, w przeciwnym wypadku należy zakończyć PU**

**Nazwa PU: Dodaj\_produkt**

**Cel: Dodawanie produktu**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj produktami**

**Warunki końcowe: Wprowadzone dane produktu muszą być unikalne**

**Scenariusz:**

1. **Należy zweryfikować czy wprowadzone dane produktu są unikalne za pomocą PU Zweryfikuj\_dane\_produktu**
2. **Jeżeli wprowadzone dane produktu są unikalne i poprawne następuje dodanie produktu do bazy danych**
3. **W innym przypadku należy zakończyć PU**

**Nazwa PU: Zweryfikuj\_dane\_produktu**

**Cel: Weryfikacja poprawności danych produktu**

**Warunki początkowe: Jest uruchamiany z PU Dodaj\_produkt lub PU Edytuj\_produkt**

**Warunki końcowe: Zwraca wynik określający czy wprowadzone dane są poprawne lub nie**

**Scenariusz:**

1. **Porównuje dane produktu z danymi produktów znajdujących się już w bazie danych**
2. **W przypadku znalezienia produktu o podanych danych zwracany jest wynik pozytywny i następuje koniec przeszukiwania bazy danych**
3. **W innym przypadku zwracany jest wynik negatywny**

**Nazwa PU: Usuń\_produkt**

**Cel: Usunięcie produktu**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_produktami**

**Warunki końcowe: Wybrany produkt musi znajdować się w bazie**

**Scenariusz:**

1. **W przypadku usunięcia produkt jest usuwany z bazy danych**
2. **W innym wypadku należy zakończyć PU**

**Nazwa PU: Edytuj\_produkt**

**Cel: Edycja produktu**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_produktami**

**Warunki końcowe: Wprowadzone dane produktu muszą być unikalne**

**Scenariusz:**

1. **Należy zweryfikować czy wprowadzone dane produktu nie znajdują się już w bazie danych za pomocą PU Zweryfikuj\_dane\_produktu**
2. **Jeżeli dane produktu nie pokrywają się z danymi z bazy danych należy zaktualizować produkt w bazie danych, w innym wypadku należy zakończyć PU**

**2 etap - Projekt, implementacja i testy bazy danych**

1. Identyfikacja encji.

Tabela Users:

* user\_id
* login
* password
* email
* phone
* name
* surname
* role

Tabela Orders:

* order\_id
* user\_id
* clothes\_id
* date
* amount
* delivery\_id
* payment\_id

Tabela Clothes:

* clothes\_id
* material
* size
* sex
* price
* collection\_id

Tabela Collections:

* collection\_id
* name
* start\_date
* end\_date

Tabela Delivery:

* delivery\_id
* city
* street
* number
* postal\_\_code
* country

Tabela Payments:

* payment\_id
* status
* payment\_form
* date

1. Sformułowanie wymagań dotyczących dostępu do bazy i jej zawartości.

Użytkownik z bazy danych może:

* Pobierać informację znajdując się w bazie danych o swoim koncie, poza jego user\_id

SELECT login, name, surname, role, email, phone FROM Users WHERE user\_id = :user\_id;

* Modyfikować niektóre informacje o swoim koncie (użytkownik nie może modyfikować user\_id, login, name, surname, role):

UPDATE Users SET email = :email, phone = :phone WHERE user\_id = :user\_id;

* Pobierać dostępne w tabeli Clothes ubrania i filtrować wyszukiwania po wszystkich jej atrybutach oraz filtrować po atrybutach z tabeli Collections

SELECT \* FROM Clothes WHERE collection\_id IN (

SELECT collection\_id FROM Collections WHERE start\_date <= CURRENT\_DATE AND end\_date >= CURRENT\_DATE

);

SELECT \* FROM Clothes

WHERE (:material IS NULL OR material = :material)

AND (:size IS NULL OR size = :size)

AND (:sex IS NULL OR sex = :sex)

AND (:price IS NULL OR price <= :max\_price);

* Tworzyć zamówienie:

INSERT INTO Orders (user\_id, clothes\_id, date, amount, delivery\_id, payment\_id)

VALUES (:user\_id, :clothes\_id, :date, :amount, :delivery\_id, :payment\_id);

* Tworzyć (dokonywać) płatności za stworzone przez siebie zamówienie:

INSERT INTO Payments (status, payment\_form, date, user\_id)

VALUES (:status, :payment\_form, CURRENT\_DATE, :user\_id);

* Pobierać informacje o płatności dokonanej przez swoje konto:

SELECT status, payment\_form, date FROM Payments WHERE payment\_id = :payment\_id;

* Tworzyć adres dostawy dla zamówienia:

INSERT INTO Delivery (city, street, number, postal\_code, country)

VALUES (:city, :street, :number, :postal\_code, :country);

* Modyfikować adres dostawy swoich zamówień:

UPDATE Delivery SET city = :city, street = :street, number = :number, postal\_code = :postal\_code, country = :country

WHERE delivery\_id = :delivery\_id;

1. Opracowanie diagramu konceptualnego modelu bazy.

A screenshot of a computer

Description automatically generated

1. Opracowanie diagramu fizycznego modelu bazy

A diagram of a computer

Description automatically generated

3 etap – Projekt, implementacja i testy aplikacji bazodanowej

1. Makieta interfejsu graficznego aplikacji
   1. Główne Menu

Obraz zawierający tekst, wizytówka, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* 1. Zarządzanie kontem

Obraz zawierający tekst, zrzut ekranu, linia, Prostokąt

Opis wygenerowany automatycznie

* 1. Zarządzanie zamówieniami

Obraz zawierający tekst, zrzut ekranu, wizytówka, Czcionka

Opis wygenerowany automatycznie

* 1. Zarządzanie koszykiem

Obraz zawierający tekst, zrzut ekranu, Równolegle, linia

Opis wygenerowany automatycznie

* 1. Przeglądanie artykułów

Obraz zawierający tekst, zrzut ekranu, Równolegle, linia

Opis wygenerowany automatycznie

1. Diagramy czynności
2. Diagram klas
3. Wykonanie i wdrożenie aplikacji