

# Księgarnia internetowa

Dokumentacja projektu końcowego MAS

## Spis treści

1.	Opis wymagań.....	2
2.	Diagram przypadków użycia .....	3
3.	Diagram klas analityczny.....	4
4.	Scenariusz .....	5
5.	Diagram aktywności .....	6
6.	Diagram stanów.....	7
7.	Skutki analizy dynamicznej.....	7
8.	Decyzje projektowe.....	7
8.1	Implementacja hierarchii użytkowników.....	7
8.2	Realizacja zamówień.....	8
8.3	Wysyłka .....	8
8.4	Nadzór .....	8
9.	Diagram klas projektowy .....	9
10.	Koncept GUI dla przypadku użycia „Złożenia zamówienia przez klienta” .....	10
10.1	Widok koszyka .....	10
10.2	Wybór metody płatności.....	11
10.3	Podsumowanie zamówienia .....	11
10.4	Komunikat o pomyślnym złożeniu zamówienia .....	12

# 1. Opis wymagań

## Opis konceptu aplikacji:

System zarządzania magazynem to aplikacja webowa, która ma na celu ułatwienie i usprawnienie kupowania książek przez Internet. Aplikacja pozwala na realizowanie zamówień, śledzenie stanu magazynowego, zarządzanie dostawami i wysyłkami, śledzeniem ich statusu oraz generowanie raportów dotyczących stanu zapasów oraz najpopularniejszych książek.

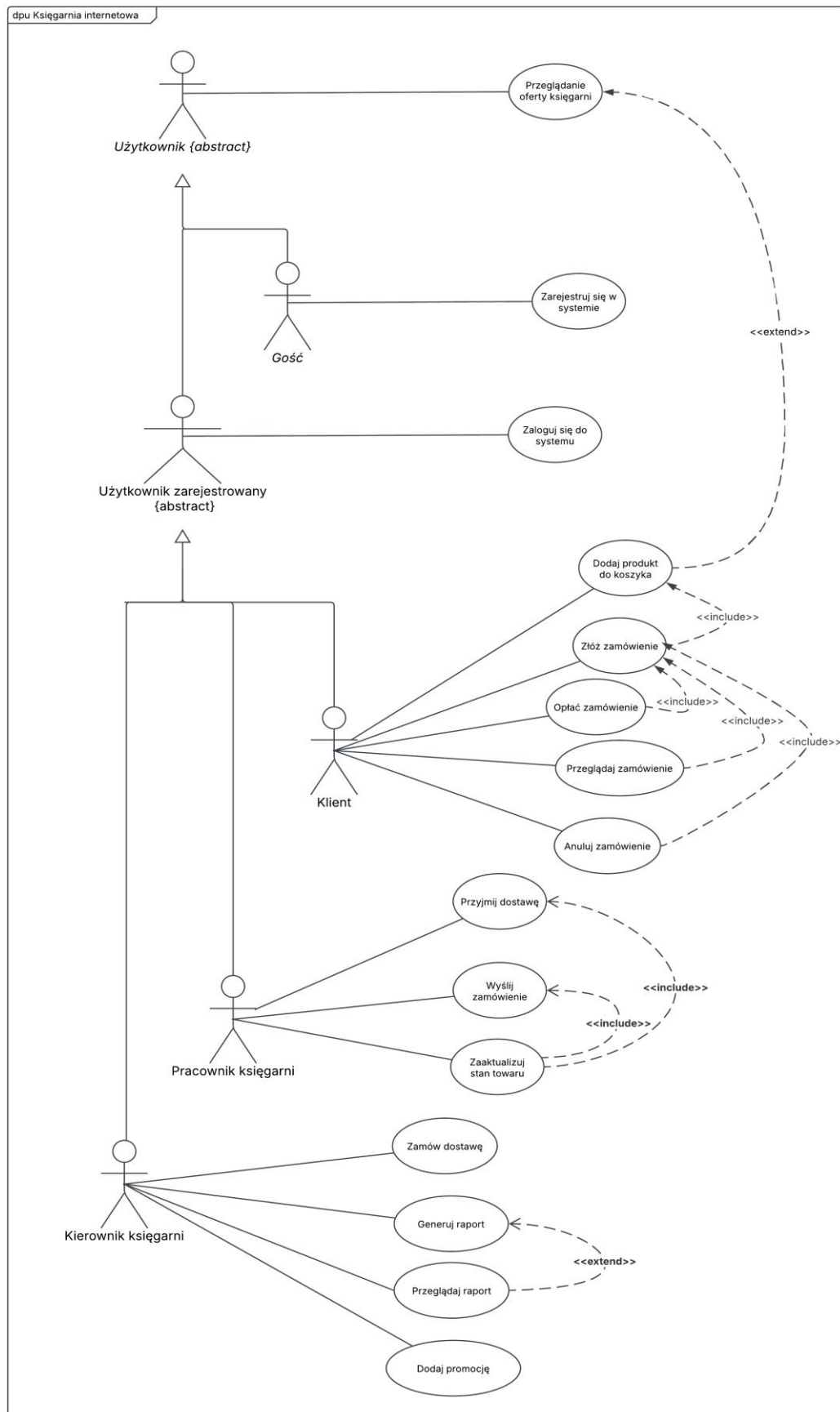
## Problemy, które rozwiązuje:

- Niewydajne zarządzanie katalogiem książek.
- Konieczność ręcznego przetwarzania zamówień.
- Trudności w śledzeniu stanu zapasów.
- Ręczne wprowadzanie danych, co prowadzi do błędów.
- Trudności w masowym zarządzaniu towarami

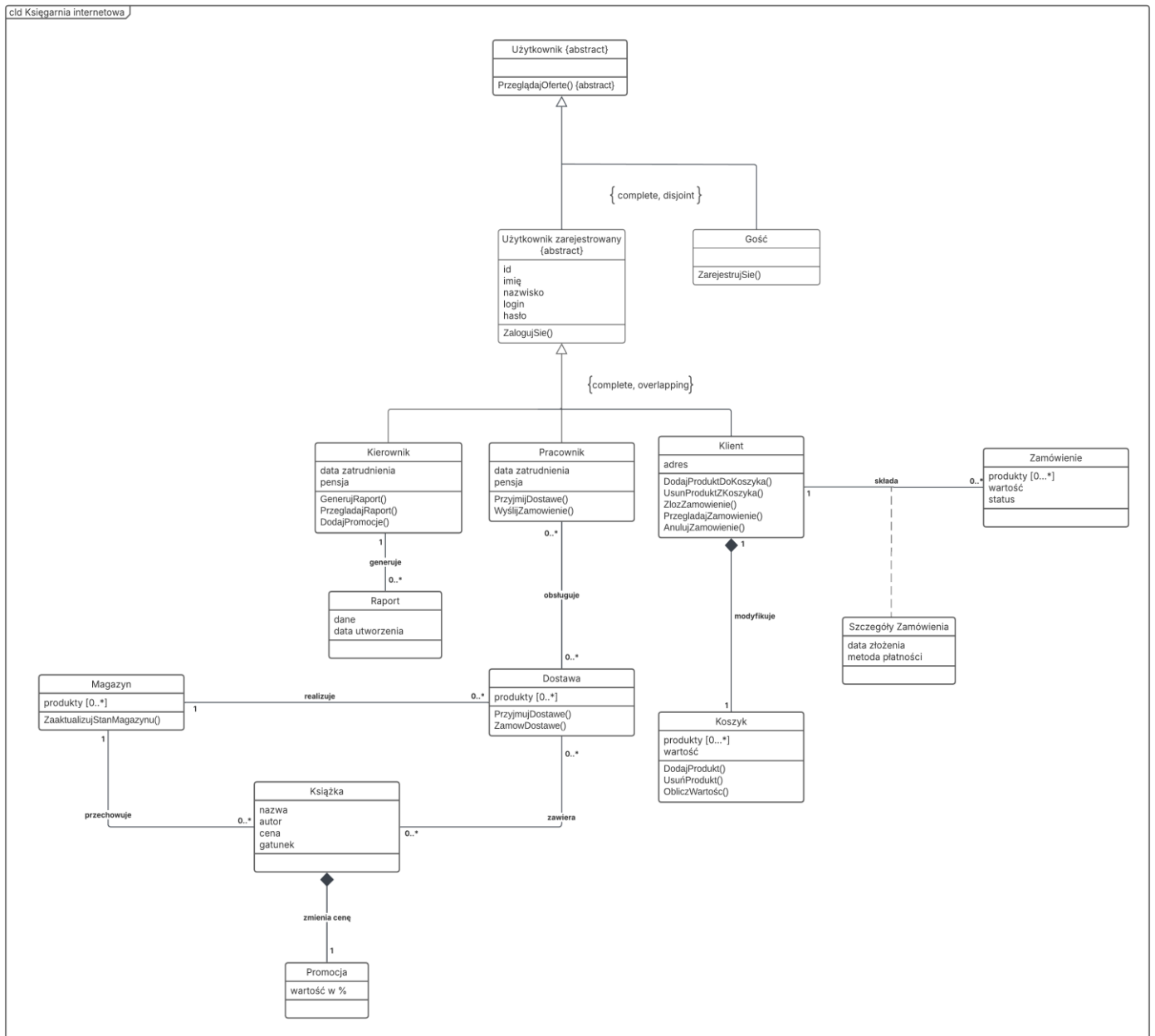
## Użytkownicy systemu i ich możliwości:

- **Gość:** Przegląda ofertę księgarni, może się zarejestrować.
- **Klient:** Składa zamówienia z możliwością anulowania, obserwuje ich status.
- **Pracownik księgarni:** Odpowiedzialny za przyjmowanie i wydawanie towarów, aktualizację stanu magazynowego.
- **Kierownik magazynu:** Odpowiedzialny za generowanie i przeglądanie raportów oraz dodawanie promocji i zamawianie dostaw.

## 2. Diagram przypadków użycia



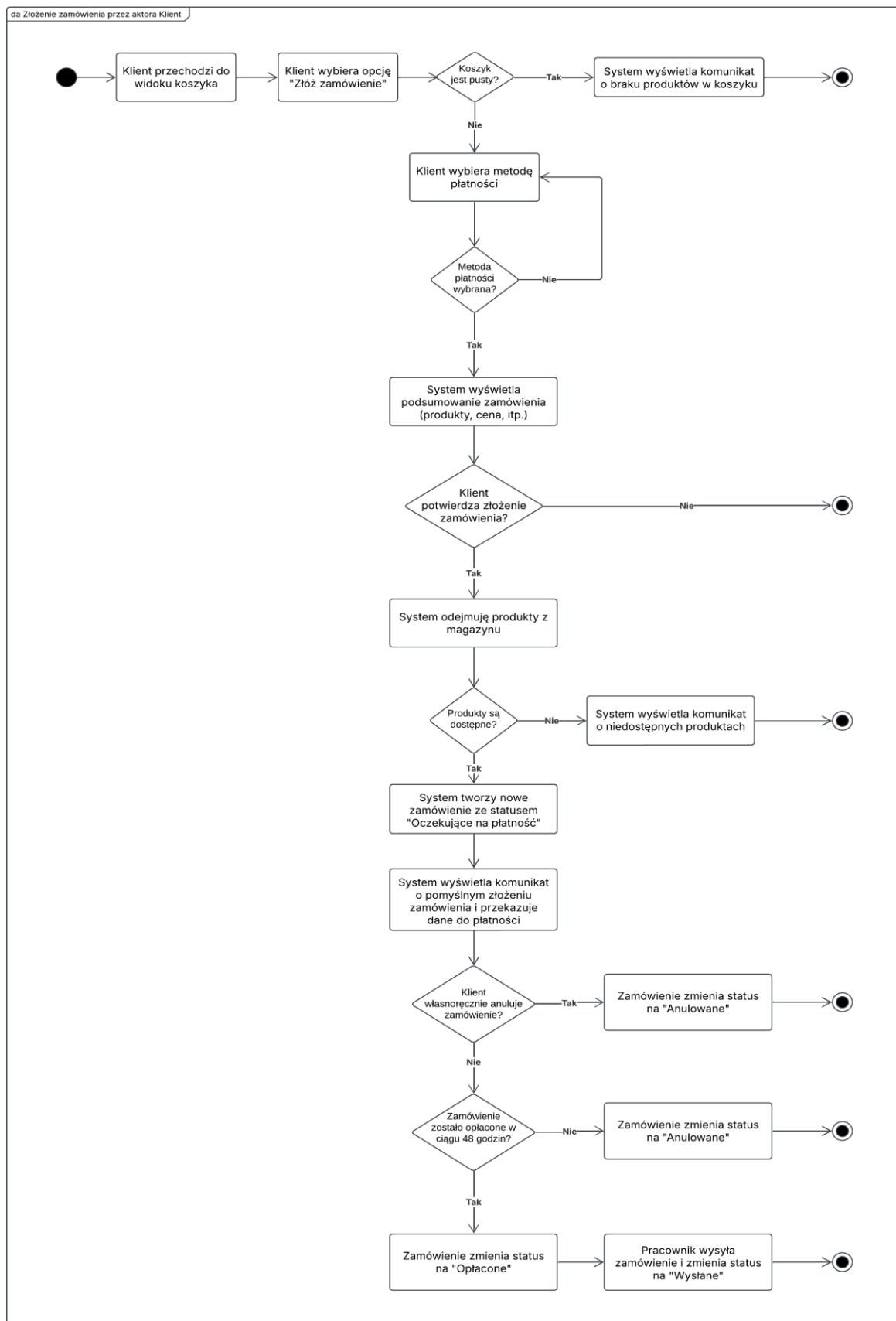
### 3. Diagram klas analityczny



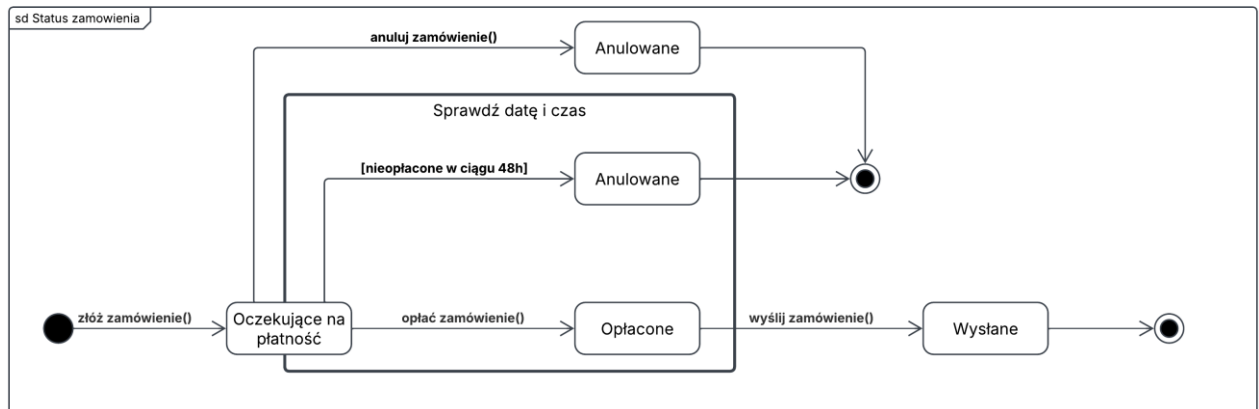
## 4. Scenariusz

<b>Nazwa przypadku użycia</b>	Złożenie zamówienia przez klienta
<b>Warunek początkowy</b>	Klient jest zalogowany do systemu i ma co najmniej jeden produkt w koszyku
<b>Główny przepływ zdarzeń</b>	<ol style="list-style-type: none"> <li>1. Klient przechodzi do widoku koszyka</li> <li>2. Klient wybiera opcję „Złóż zamówienie”</li> <li>3. System sprawdza, czy koszyk nie jest pusty</li> <li>4. Klient wybiera metodę płatności</li> <li>5. System wyświetla podsumowanie zamówienia (produkty, cena, itp.)</li> <li>6. Klient potwierdza złożenie zamówienia</li> <li>7. System odejmuje odpowiednią liczbę danych produktów z magazynu</li> <li>8. System tworzy nowe zamówienie z wybranymi produktami i nadaje mu status „Oczekujące na płatność”</li> <li>9. System wyświetla komunikat o pomyślnym złożeniu zamówienia i wyświetla dane do opłacenia zamówienia (numer konta, link do płatności online)</li> <li>10. Klient opłaca zamówienie wybraną wcześniej metodą</li> <li>11. System zmienia status zamówienia na „Opłacone”</li> <li>12. Pracownik wysyła opłacone zamówienie i zmienia status na „Wysłane”</li> </ol>
<b>Alternatywne przepływy zdarzeń</b>	<p>A1. Koszyk jest pusty</p> <ol style="list-style-type: none"> <li>3a. System wyświetla komunikat o braku produktów w koszyku</li> <li>3b. Proces zamówienia zostaje przerwany</li> </ol> <p>A2. Brak metody płatności</p> <ol style="list-style-type: none"> <li>4a. Klient nie wybrał metody płatności</li> <li>4b. Powrót do kroku 4.</li> </ol> <p>A3. Brak potwierdzenia zamówienia</p> <ol style="list-style-type: none"> <li>6a. Klient nie potwierdza złożenia zamówienia</li> <li>6b. Proces zamówienia zostaje przerwany</li> </ol> <p>A4. Brak dostępnych produktów w magazynie</p> <ol style="list-style-type: none"> <li>7a. System wykrywa brak dostępnych produktów w magazynie</li> <li>7b. System anuluje zamówienie i informuje klienta, które produkty są niedostępne</li> <li>7c. Proces zamówienia zostaje przerwany</li> </ol> <p>A5. Klient nie opłacił zamówienia</p> <ol style="list-style-type: none"> <li>10a. Klient nie opłaca zamówienia w ciągu 48 godzin lub własnoręcznie anuluje zamówienie.</li> <li>10b. System przywraca produkty do magazynu</li> <li>10c. System zmienia status zamówienia na „Anulowane”</li> </ol>
<b>Zakończenie</b>	Zamówienie zostaje złożone i opłacone przez klienta.
<b>Warunek końcowy</b>	Zamówienie znajduje się w systemie ze statusem „Opłacone”, a magazyn został zaktualizowany

## 5. Diagram aktywności



## 6. Diagram stanów



## 7. Skutki analizy dynamicznej

Podczas tworzenia diagramu aktywności i diagramu stanów uznałem, że w klasie *Order* należy dodać metodę *PayForOrder()* w celu umożliwienia opłacenia zamówienia. Dodałem również metodę wyświetlającą szczegóły zamówienia *PrintOrderDetails()* oraz przeniostem metodę *CancelOrder()* z klasy *Client* do klasy *Order* w celu usprawnienia procesu zarządzania zamówieniem.

Dodatkowo utworzyłem klasę *OrderStatus* jako Enum zawierający możliwe statusy zamówień. Odpowiada on za atrybut *Status* w klasie *Order*.

## 8. Decyzje projektowe

### 8.1 Implementacja hierarchii użytkowników

W projekcie zastosowałem hierarchię dziedziczenia w klasach użytkowników, aby ujednolicić wspólne cechy przy jednoczesnym zachowaniu różnych ról w systemie.

- **User** jest abstrakcyjną klasą bazową, po której dziedziczą wszystkie pozostałe klasy użytkowników systemu.
  - **Guest** jest klasą dla użytkownika niezarejestrowanego umożliwiającą przeglądanie oferty księgarni oraz rejestrację.
  - **Registered user** jest abstrakcyjną klasą bazową dla aktorów posiadających konto w systemie.
    - **Client** jest klasą dla użytkowników dokonujących zakupów w systemie.
    - **Employee** jest klasą dla użytkowników realizujących zamówienia *klientów*.
    - **Manager** jest klasą dla użytkowników nadzorujących system.

W hierarchii pojawiają się również adnotacje:

- **{complete, disjoint}** dla *Registered user* oraz *Guest*, gdyż podział użytkowników na tych z kontem oraz bez jest kompletny, oraz aby rozłączyć uprawnienia użytkownika z kontem w systemie oraz bez niego.
- **{complete, overlapping}** dla **Client**, **Employee**, **Manager**, w celu umożliwienia robienia zakupów w systemie przez *pracowników* oraz *kierowników*, oraz zapewnieniu możliwości zmiany ról użytkowników.

## 8.2 Realizacja zamówień

Proces realizacji zamówień został zaprojektowany jako zestaw powiązanych elementów:

- Klasa **Order** reprezentuje jedno zamówienie złożone przez użytkownika **Client**.
- Szczegóły zamówienia są zawarte w klasie **OrderDetails**, która jest asocjacją z atrybutem pomiędzy klasami **Client** oraz **Order**.
- Atrybut Status w klasie **Order** jest reprezentowany jako typ wyliczeniowy *Enum* w **OrderStatus**, co ogranicza dozwolone stany zamówienia.
- Klasa **Cart** będąca w asocjacji z klasą **Client** przekazuje dane o wybranych produktach do klasy **Order**.
- Produkty są reprezentowane poprzez klasę **Book**.

## 8.3 Wysyłka

Proces wysyłania i przyjmowania zamówień jest realizowany przez *Pracownika* w następujący sposób:

- **Employee** obsługuje klasę **Delivery** w asocjacji wiele do wielu, aby umożliwić obsługiwanie różnych dostaw przez różnych pracowników.
- Zamówienia z klasą **Delivery** są dostarczane z lub do magazynu w klasie **Warehouse**.
- Informację o zamówieniach klasa **Delivery** czerpie poprzez asocjację wiele do wielu z klasą **Order**, która zawiera informacje o produktach.

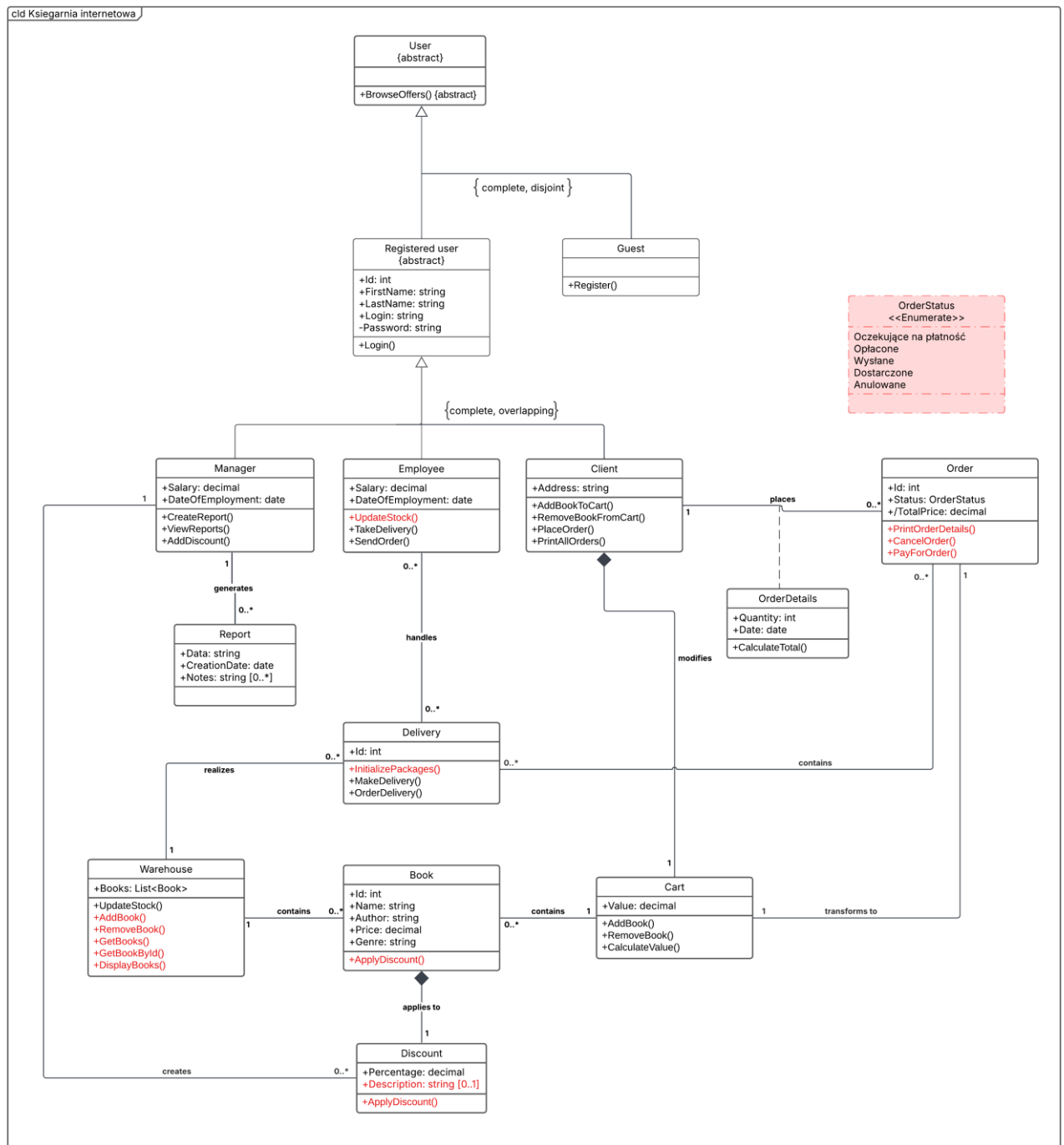
## 8.4 Nadzór

Proces nadzorowania działania systemu jest realizowany przez *Kierownika* w następujący sposób:

- Generowanie i przeglądanie raportów z poprzez klasę **Report**.
- Kontrolowanie i zarządzanie zniżkami odbywa się poprzez klasę **Discount**, która jest powiązana z **Book**.

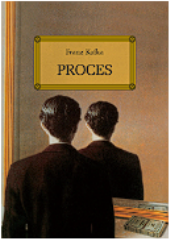



## 9. Diagram klas projektowy



## 10. Koncept GUI dla przypadku użycia „Złożenia zamówienia przez klienta”

### 10.1 Widok koszyka

Image	Product	Price	Quantity	Total	Remove
	Franz Kafka - Proces	\$10.00	<input type="text" value="1"/>	\$10.00	<button>X</button>
	George Orwell - Rok 1984	\$30.00	<input type="text" value="1"/>	\$30.00	<button>X</button>

UPDATE CART

CONTINUE SHOPPING

#### CART TOTALS

Cart value	\$40.00
Discount	-\$4.00
Total	\$36.00

PROCEED TO CHECKOUT

## 10.2 Wybór metody płatności

### Your Order

Product	Total
Franz Kafka - Proces x 1	\$10.00
George Orwell - Rok 1984 x 1	\$30.00
<b>Cart Subtotal</b>	\$36.00 (with -\$4 discount)
<b>Order Total</b>	<b>\$40.00 (+\$4 delivery)</b>

[Direct Bank Transfer](#)

[Paypal](#)

## 10.3 Podsumowanie zamówienia

### Billing Details

Country \*  
Poland

First Name \*  
Mikołaj

Last Name \*  
Wróblewski

Company Name

Address \*  
Koszykowa 86  
Apartment, suite, unit etc. (optional)

State / Country \*  
Warszawa

Posta / Zip \*  
12-345

Email Address \*  
pja@pjwstk.edu.pl

Phone \*  
123 456 789

### Your Order

Product	Total
Franz Kafka - Proces x 1	\$10.00
George Orwell - Rok 1984 x 1	\$30.00
<b>Cart Subtotal</b>	\$36.00 (with -\$4 discount)
<b>Order Total</b>	<b>\$40.00 (+\$4 delivery)</b>

**Payment method: Paypal**

PLACE ORDER

## 10.4 Komunikat o pomyślnym złożeniu zamówienia



# Thank you!

You order was successfully placed.

CANCEL ORDER

PROCEED TO PAYMENT