

# **System zarządzania finansami (FMS)**

Version 1.0.0

Jerzy Jancewicz  
16 czerwca 2024

## Spis treści

<b>1</b>	<b>Dziedzina problemowa</b>	<b>2</b>
<b>2</b>	<b>Cel</b>	<b>2</b>
<b>3</b>	<b>Wymagania użytkownika</b>	<b>3</b>
<b>4</b>	<b>Diagram przypadków użycia</b>	<b>6</b>
<b>5</b>	<b>Diagram klas - Analityczny</b>	<b>7</b>
<b>6</b>	<b>Diagram klas - Projektowy</b>	<b>8</b>
<b>7</b>	<b>Scenariusz przypadku użycia</b>	<b>9</b>
<b>8</b>	<b>Diagram stanów dla klasy Transakcja</b>	<b>10</b>
<b>9</b>	<b>Diagram aktywności</b>	<b>11</b>
<b>10</b>	<b>Projekt GUI</b>	<b>12</b>
10.1	Ekran Startowy . . . . .	12
10.2	Szczegóły budżetu . . . . .	13
10.3	Szczegóły transakcji . . . . .	14
10.4	Nowa Transakcja . . . . .	15
10.5	Akceptacja Wersji roboczej . . . . .	16
10.6	Akceptacja transakcji . . . . .	17
10.7	Odrzucenie transakcji . . . . .	18
<b>11</b>	<b>Omówienie decyzji projektowych i skutków analizy dynamicznej</b>	<b>19</b>

# 1 Dziedzina problemowa

W dzisiejszych czasach wiele firm, organizacji zмага się z problemami natury finansowej, między innymi takich jak kontrolowaniem przepływów środków, optymalizacją wydatków, skomplikowanym prawem podatkowym. Te niedogodności mogą prowadzić do wielu długo, jak i krótko terminowych trudności w danej spółce, takich jak uwikłanie się w bardzo kosztowne problemy natury prawnej, nieoptymalnej alokacji środków. Wszystkie te komplikacje, mogą skutkować dużymi stratami budżetowymi, co z kolei może prowadzić do dużo wolniejszego rozwoju, a w niekorzystnym przypadku może nawet spowodować upadek danej organizacji.

Metodą na zapobieganie powyższych zagrożeń jest implementacja w danej firmie Systemu zarządzania systemami (FMS - Financial management system), który pozwala kontrolować wszystkie działania finansowe, jak i monitorować wszystkie podmioty wewnątrz przedsiębiorstwa.

## 2 Cel

Celem budowy Systemu Zarządzania Finansami (FMS) jest usprawnienie procesów zarządzania finansami w organizacjach. System ten ma na celu:

- Automatyzację procesów finansowych:  
Zmniejszenie liczby manualnych operacji, co prowadzi do redukcji błędów ludzkich przy dokumentowaniu, jaki i dokonywaniu transakcji.
- Uporządkowanie biurokracji:  
Ułatwienie zarządzania fakturami, rachunkami, budżetami oraz innymi dokumentami finansowymi przez pracowników firmy.
- Ułatwienie raportowania:  
Generowanie dokładnych raportów finansowych, które dostarczają menadżerom niezbędne informacje rachunkowe firmy.
- Poprawę efektywności:  
Szybsza i łatwiejsza obsługa transakcji, prowadzi do zwiększenia zysków danego przedsiębiorstwa.

### 3 Wymagania użytkownika

1. Dane firm:  
System powinien przechowywać dane firm, takie jak nazwa, adres, NIP (unikalny) oraz numer telefonu (maksymalnie 12 cyfr). Firma może być klasyfikowana jako klient lub dostawca. Firma może wystawiać wiele rachunków, lecz rachunek może być przypisany tylko do jednej firmy. Dana firma może zmienić rolę z klienta na dostawcę i odwrotnie, lecz nie może być dwoma rolami jednocześnie.
2. Dane klientów:  
System powinien przechowywać dane klientów, w tym datę rejestracji oraz przecenę, którą wylicza się na podstawie długości współpracy z firmą. Klient może otrzymywać rachunki wystawione przez pracownika.
3. Dane dostawców:  
System powinien przechowywać dane dostawców, takie jak data dostawy. Dostawca może wystawiać rachunki za dostarczony towar.
4. Operacje bankowa:  
System powinien zarządzać różnymi operacjami finansowymi, takimi jak, faktury i rachunki. Operacje finansowe muszą zawierać informacje o dacie, kwocie netto i brutto oraz statusie. Co ważne kwota netto jest mniejsza, bądź równa kwocie brutto. Możliwe jest także dodanie Opisu operacji, lecz nie jest to konieczne. Księgowy zarządza wieloma operacjami bankowymi, które są przypisane do jednego księgowego.
5. Rachunek:  
System powinien przechowywać informacje o typie transakcji wystawionym na rachunku, takich jak wystawiony i otrzymany. Rachunek może mieć przypisane różne podatki w zależności od typu transakcji.
6. Podatki:  
System powinien przechowywać informacje o podatkach, takie jak typ podatku i stawka. Każdy podatek może być związany z wieloma rachunkami.

7. Faktura:

System powinien przechowywać informacje o fakturach, takie jak nazwa nabywcy i sprzedającego. Faktura posiada co najmniej jeden rachunek, lecz dany rachunek może być powiązany tylko z jedną fakturą. Wszystkie rachunki na fakturze są przechowywane w formie uporządkowanej po kwocie netto rachunku.

8. Operacja finansowa:

Każda operacja finansowa (raport finansowy, prognoza finansowa, plan finansowy) posiada dane dotyczące nazwy operacji, okresu czasu danej operacji i kwoty przychodu. Każda z operacji wylicza tę kwotę na własny sposób, lecz częścią wspólną jest zawsze podany okres (data początkowa i data końcowa) danej operacji.

9. Raporty i prognozy finansowe:

System powinien umożliwiać tworzenie raportów finansowych zawierających kwotę obrotu. Specjalista finansowy zarządza wieloma raportami finansowymi, które są przypisane do jednego specjalisty. Prognozy finansowe powinny zawierać wyliczalny procent zysku. Specjalista finansowy zarządza również wieloma prognozami finansowymi, które są zarządzane przez jednego specjalistę.

10. Plany finansowe:

System powinien umożliwiać tworzenie planów finansowych, które zawierają nazwę planu, poziom ryzyka w skali od 1 do 10, wyliczalną potencjalną kwotę przychodu oraz wartość mówiącą czy dany plan został zrealizowany. Dyrektor finansowy zarządza wieloma planami finansowymi, które są zarządzane przez jednego dyrektora.

11. Budżet:

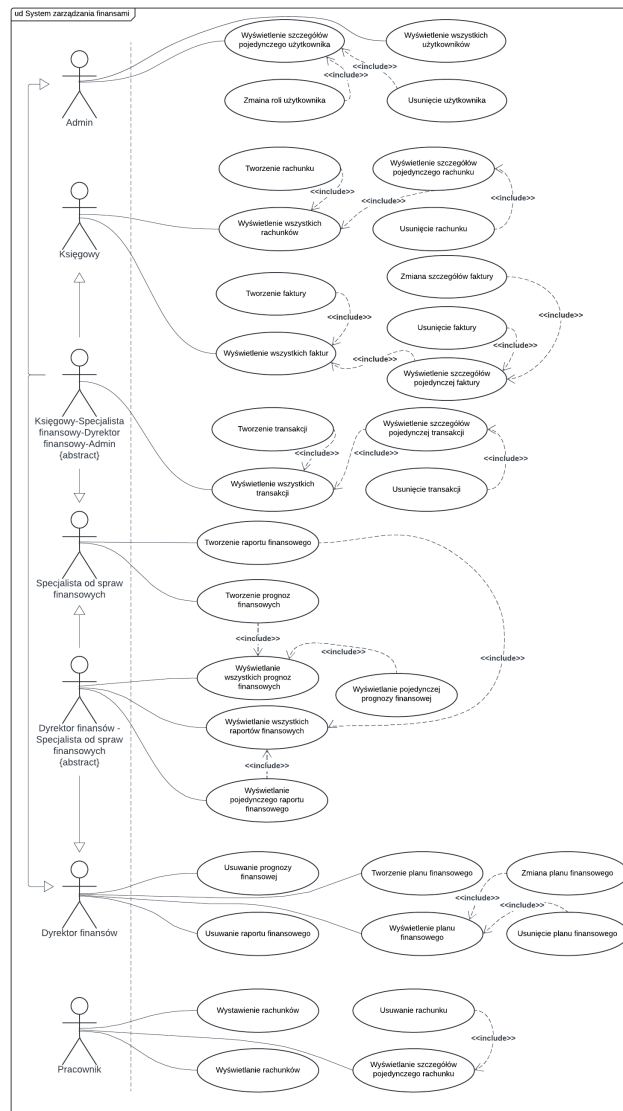
System powinien umożliwiać zarządzanie budżetem, w którym możemy znaleźć informacje na temat nazwy, nazwy projektu, kwoty oraz możliwego opisu. Konto bankowe nie istnieje bez budżetu, więc każdy budżet może zarządzać wieloma kontami bankowymi. Konto bankowe posiada informacje o Nazwie banku w którym te konto się znajduje, saldo na danym koncie i numerze konta.

12. Użytkownicy i role:

System powinien obsługiwać różne role użytkowników: administratora, księgowego, specjalisty finansowego, dyrektora finansowego oraz pracownika. Każda rola ma swoje unikalne uprawnienia. Budżet może być powiązany z wieloma użytkownikami, a każdy użytkownik może mieć wiele budżetów. Operacje są zapisane w ramach każdej transakcji, którą wykonuje użytkownik. Transakcja powinien zawierać dane, takie jak data operacji, kwota, rodzaj transakcji i status. Użytkownik może mieć dostęp do jednej, lub 2 ról na raz, o ile te role to księgowy i specjalista finansowy. Każdy użytkownik posiada dane, takie jak Imię, Nazwisko, Telefon komórkowy nie większy niż 12 liczb, email i Nazwę.

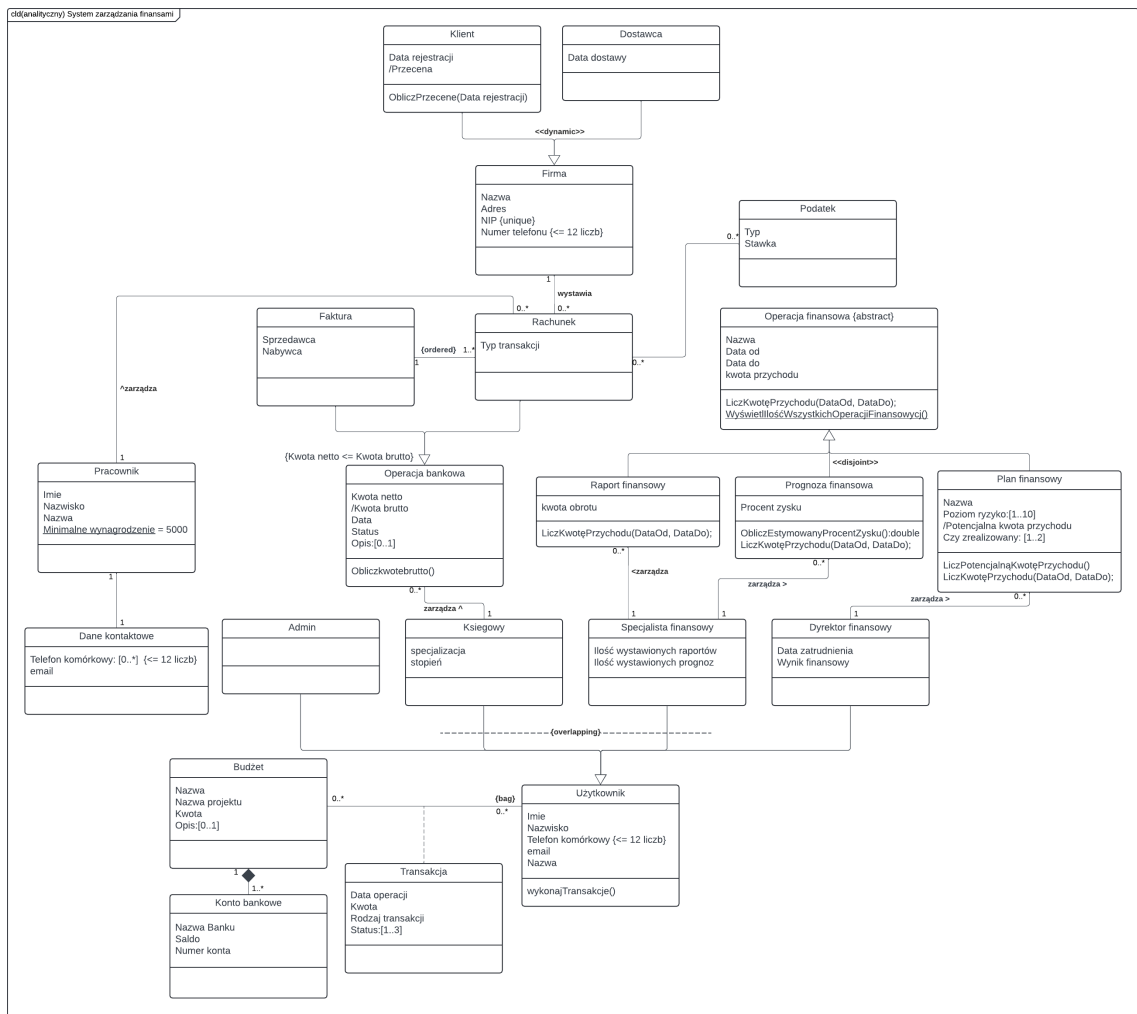
- Admin: Może wyświetlać szczegóły użytkowników, przeglądać listę wszystkich użytkowników, zmieniać role użytkowników, usuwać użytkowników oraz tworzyć transakcje.
- Księgowy: Może wyświetlać wszystkie transakcje, przeglądać szczegóły transakcji, usuwać transakcje, tworzyć faktury, przeglądać faktury, wyświetlać szczegóły faktur, zmieniać szczegóły faktur, usuwać faktury oraz tworzyć raporty finansowe. Księgowy zarządza wieloma operacjami bankowymi.
- Specjalista Finansowy: Może tworzyć prognozy finansowe, przeglądać prognozy finansowe, wyświetlać szczegóły prognoz finansowych, przeglądać raporty finansowe, wyświetlać historię raportów finansowych oraz przeglądać historię prognoz finansowych. Specjalista finansowy zarządza wieloma raportami finansowymi oraz prognozami finansowymi.
- Dyrektor Finansowy: Może usuwać prognozy finansowe, tworzyć plany finansowe, zmieniać plany finansowe, usuwać raporty finansowe, wyświetlać plany finansowe oraz usuwać plany finansowe. Dyrektor finansowy zarządza wieloma planami finansowymi.
- Pracownik: Może wystawiać rachunki, usuwać rachunki, wyświetlać rachunki oraz przeglądać szczegóły rachunków. Pracownik zarządza wieloma rachunkami, a rachunek może być zarządzany przez jednego pracownika.

## 4 Diagram przypadków użycia



Rysunek 1: Diagram przypadków użycia

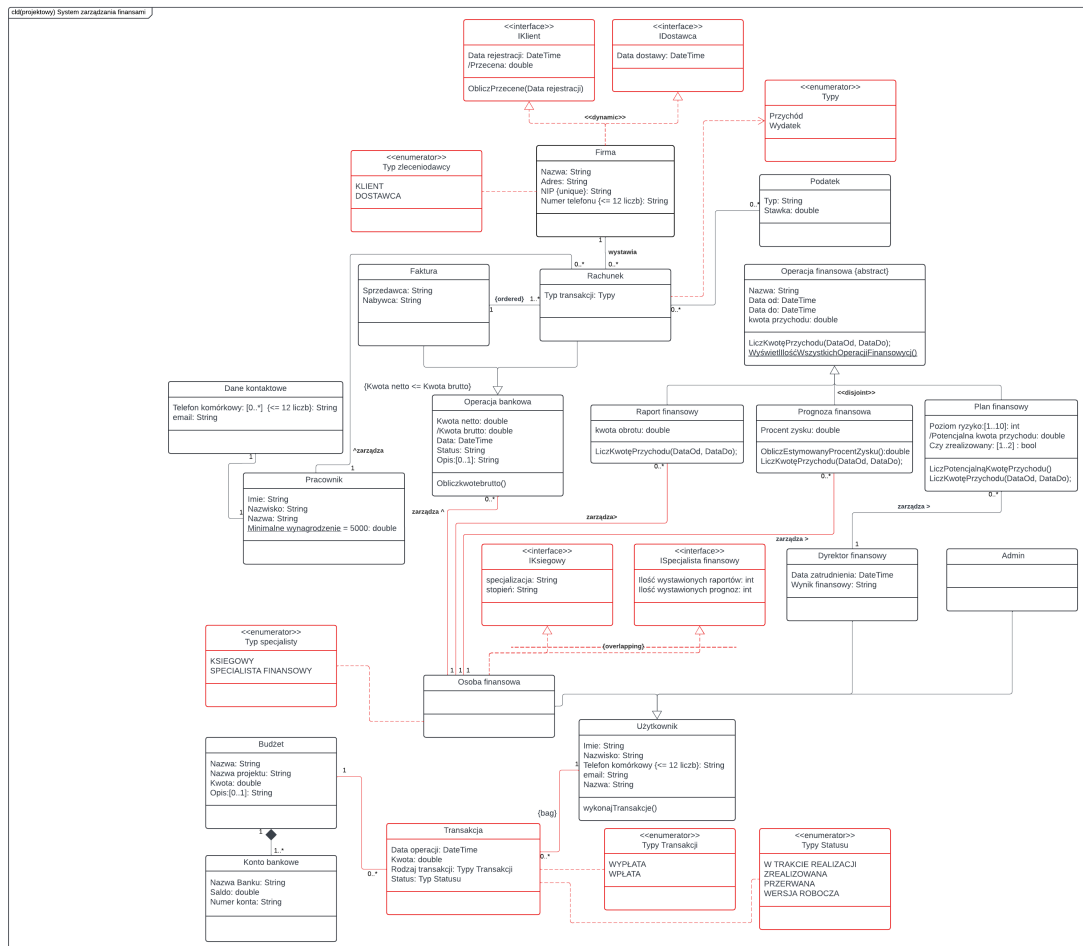
## 5 Diagram klas - Analityczny



Rysunek 2: Diagram klas - Analityczny



## 6 Diagram klas - Projektowy



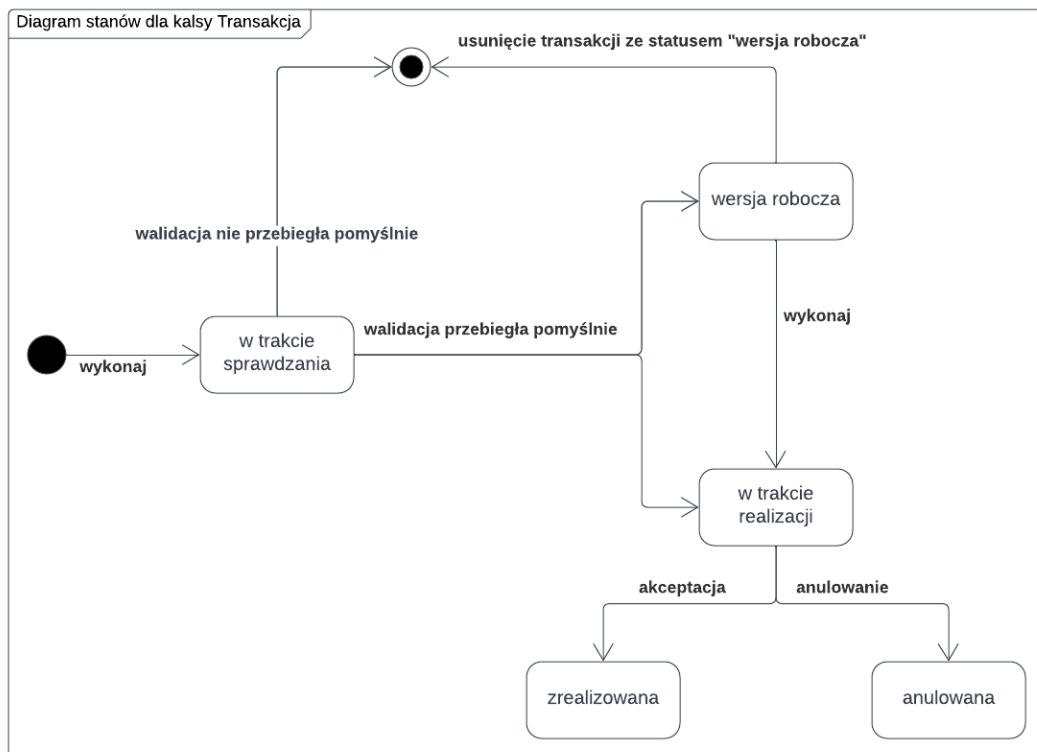
Rysunek 3: Diagram klas - Projektowy

## 7 Scenariusz przypadku użycia

Tworzenie transakcji
<b>Warunek początkowy:</b> 1. Aktor księgowy jest zalogowany do systemu
<b>Główny przepływ zdarzeń:</b> 1. Przypadek użycia wywołuje Księgowy. 2. System wyświetla wszystkie przypisane budżety dla tego użytkownika. 3. Aktor Księgowy wybiera dany budżet z listy budżetów. 4. System wyświetla szczegóły danego budżetu włącznie z przypisanymi transakcjami i użytkownikami. 5. Aktor Księgowy klika przycisk "Dokonaj transakcji". 6. System wyświetla formularz, w którym znajduje się pole do wyboru typu transakcji i pole kwoty transakcji. 7. Aktor Księgowy uzupełnia pole typ transakcji na wpłata, pole kwoty na 5000zł i zatwierdza transakcję przyciskiem "Wykonaj". 8. System zatwierdza dane i sprawdza ilość środków w budżecie. 9. System przeprowadza odpowiednie operacje na budżecie. 10. System ustawia datę wykonania transakcji. 11. System ustawia status transakcji jako "zrealizowana". 12. System wyświetla komunikat o pomyślnym dokonaniu transakcji. 13. Koniec przypadku użycia.
<b>Alternatywne przepływy zdarzeń:</b> 2a. System nie wyświetla żadnego budżtu przypisanego do tego użytkownika, z powodu ich braku. System przechodzi do punktu 13. 7a. Aktor Księgowy uzupełnia pole typ transakcji na wypłata, pole kwoty na 5000zł i zatwierdza transakcję przyciskiem "Wykonaj". 7aa. System nie zatwierdza transakcji z powodu braku środków w budżecie. 7ab. System wyświetla komunikat o błędzie przy dokonywaniu transakcji z szczegółami błędu i przechodzi do punktu 4. 7b. Aktor Księgowy uzupełnia pole typ transakcji na wpłata, pole kwoty na <=0zł i zatwierdza transakcję przyciskiem "Wykonaj". 7ba. System nie zatwierdza transakcji z powodu błędnie wypełnionego formularza. 7bb. System wyświetla komunikat o błędzie przy dokonywaniu transakcji z szczegółami błędu i przechodzi do punktu 4. 7c. Aktor Księgowy uzupełnia pole typ transakcji na wypłata, pole kwoty na <=0zł i zatwierdza transakcję przyciskiem "Wykonaj". 7ca. System nie zatwierdza transakcji z powodu błędnie wypełnionego formularza. 7cb. System wyświetla komunikat o błędzie przy dokonywaniu transakcji z szczegółami błędu i przechodzi do punktu 4. 7d. Aktor Księgowy uzupełnia prawidłowo pola typ transakcji, kwota i zatwierdza transakcję przyciskiem "Dodaj jako wersja robocza". 7da. System przeprowadza odpowiednie operacje na budżecie i zatwierdza transakcję, ustawia status jako "wersja robocza". 7e. Aktor Księgowy uzupełnia nieprawidłowo pola typ transakcji, kwota i zatwierdza transakcję przyciskiem "Dodaj jako wersja robocza". 7ea. System nie zatwierdza transakcji z powodu błędnie wypełnionego formularza. 7eb. System wyświetla komunikat o błędzie przy dokonywaniu transakcji z szczegółami błędu i przechodzi do punktu 4.
<b>Warunek końcowy:</b> 1. Zostanie stworzona transakcja powiązana z danym budżetem i pracownikiem. 2. Zostanie ustawiony status transakcji jako "zrealizowana" lub "wersja robocza".

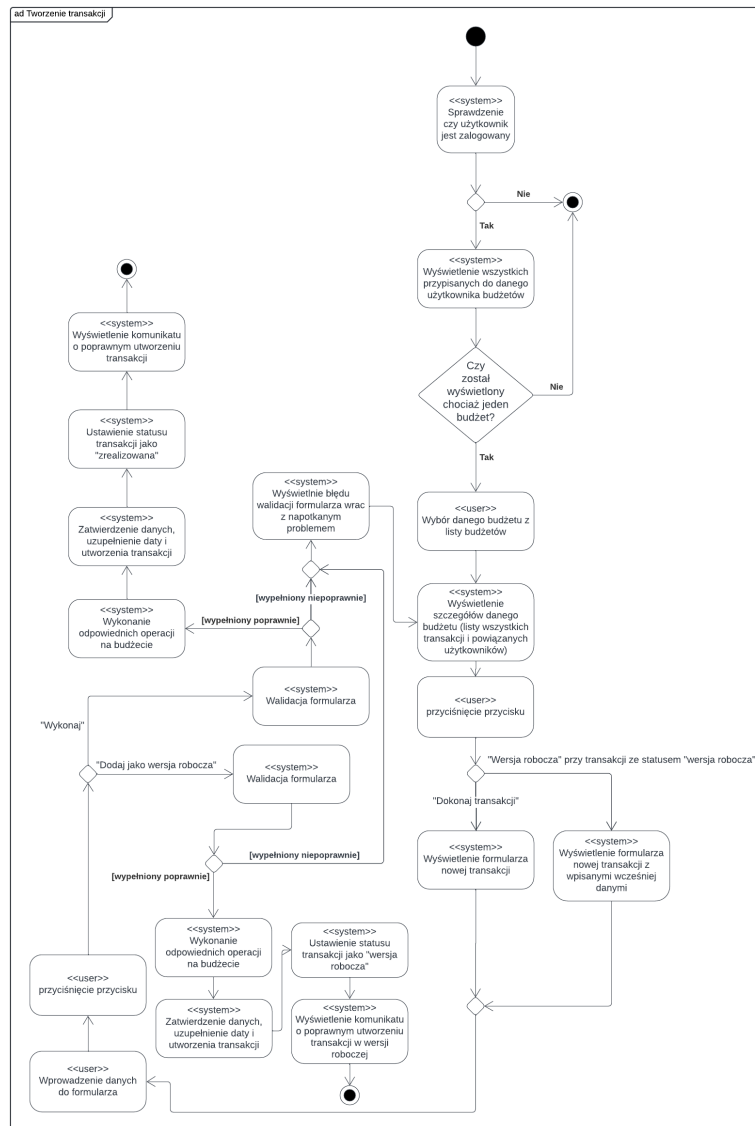
Rysunek 4: Scenariusz przypadku użycia

## 8 Diagram stanów dla klasy Transakcja



Rysunek 5: Diagram stanu dla klasy Transakcja

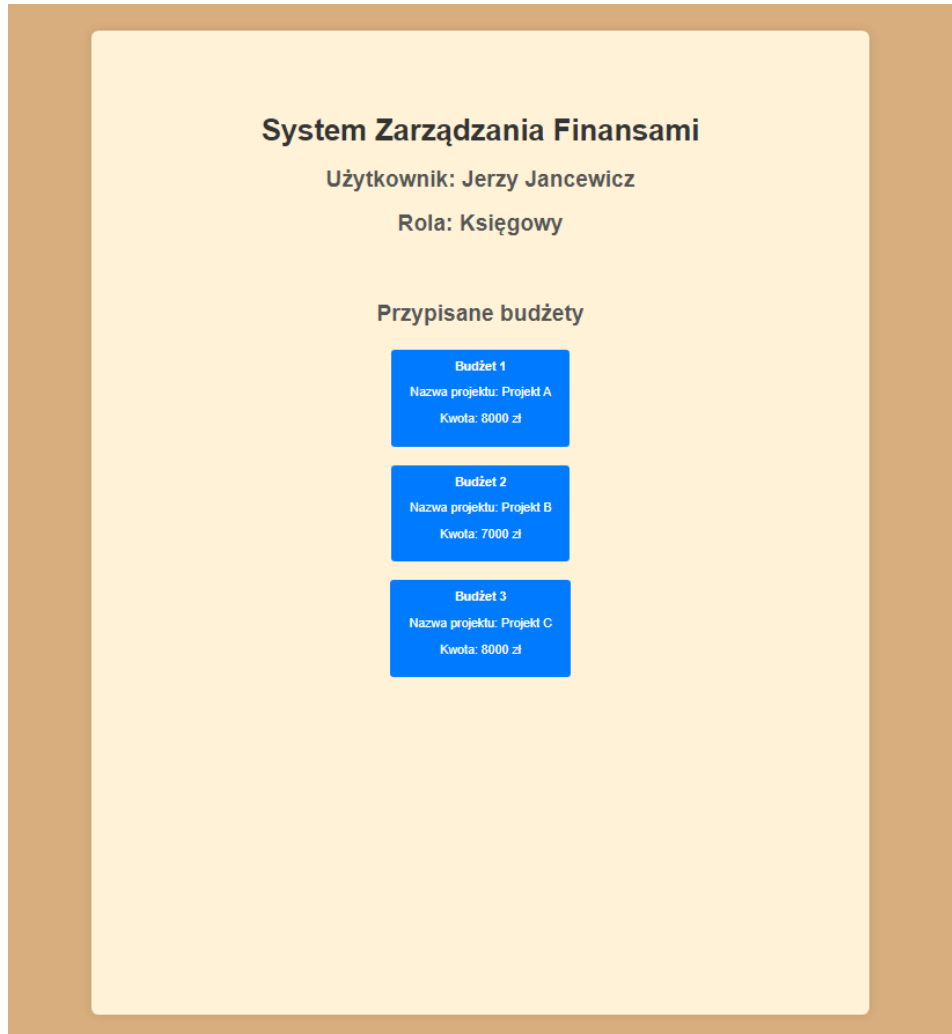
## 9 Diagram aktywności



Rysunek 6: Diagram aktywności

## 10 Projekt GUI

### 10.1 Ekran Startowy



Rysunek 7: Ekran Startowy

## 10.2 Szczegóły budżetu

Wstecz

System Zarządzania Finansami

Użytkownik: Jerzy Jancewicz

Rola: Księgowy

Szczegóły budżetu

Nazwa: Budżet 1

Nazwa projektu: Projekt A

Kwota: 8000 zł

Opis: Opis budżetu 1

Transakcje:

Data: 2023-01-01

Kwota: 2000 zł

Typ: WPLATA

Status: ZREALIZOWANA

Details

Data: 2023-02-01

Kwota: 500 zł

Typ: WYPŁATA

Status: ZREALIZOWANA

Details

Konta bankowe:

Nazwa Banku: Bank 1

Saldo: 5000 zł

Numer Konta: 1234567890

Nazwa Banku: Bank 2

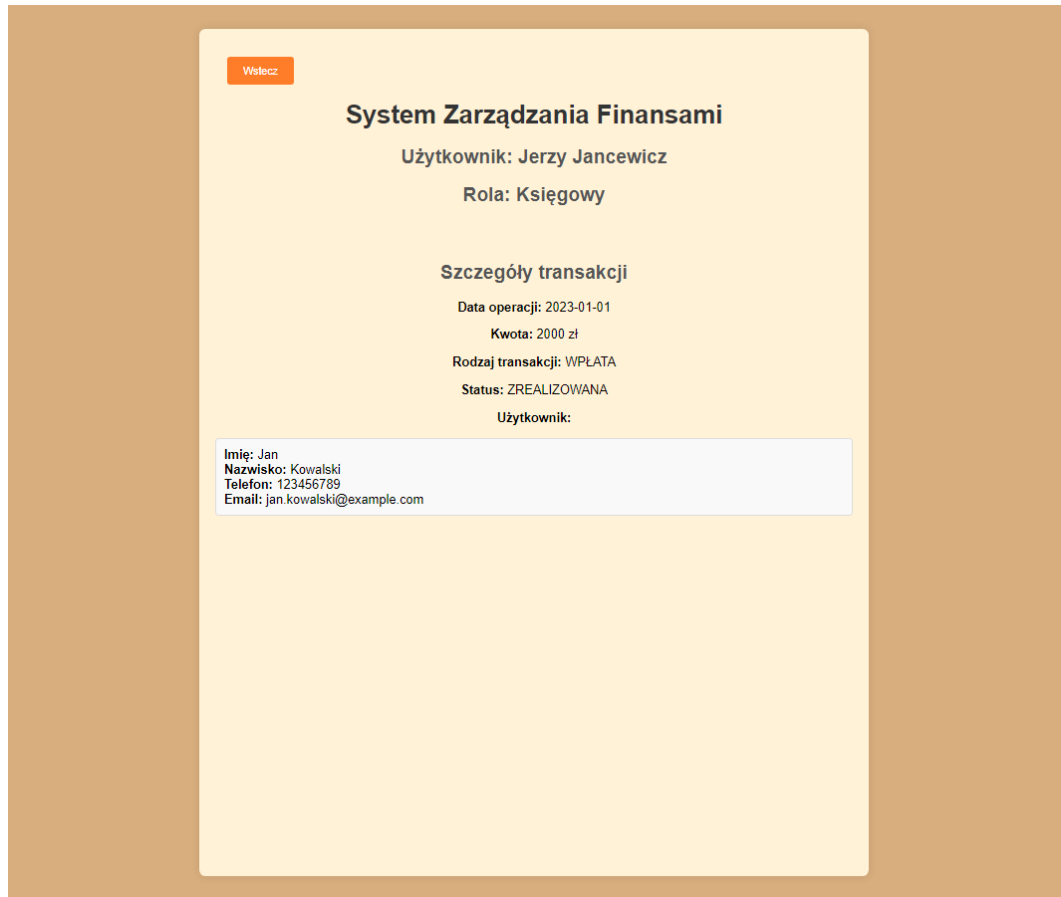
Saldo: 3000 zł

Numer Konta: 0987654321

Dokonaj transakcji

Rysunek 8: Szczegóły budżetu

## 10.3 Szczegóły transakcji



The screenshot displays a web application interface for a financial management system. At the top left, there is an orange button labeled 'Wstecz'. The main heading is 'System Zarządzania Finansami'. Below this, the user information is shown: 'Użytkownik: Jerzy Jancewicz' and 'Rola: Księgowy'. The section title 'Szczegóły transakcji' is followed by transaction details: 'Data operacji: 2023-01-01', 'Kwota: 2000 zł', 'Rodzaj transakcji: WPŁATA', 'Status: ZREALIZOWANA', and 'Użytkownik:'. At the bottom, a light blue box contains contact information: 'Imię: Jan', 'Nazwisko: Kowalski', 'Telefon: 123456789', and 'Email: jan.kowalski@example.com'.

Wstecz

**System Zarządzania Finansami**

Użytkownik: Jerzy Jancewicz

Rola: Księgowy

**Szczegóły transakcji**

Data operacji: 2023-01-01

Kwota: 2000 zł

Rodzaj transakcji: WPŁATA

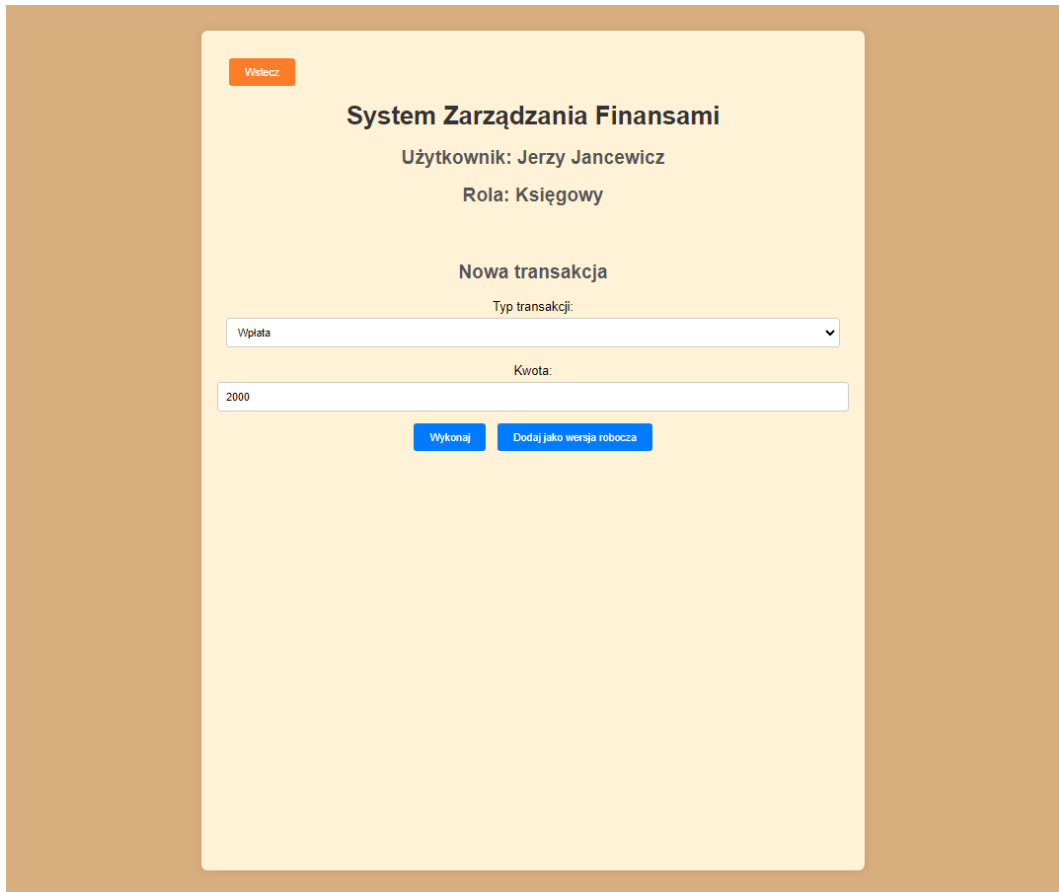
Status: ZREALIZOWANA

Użytkownik:

Imię: Jan  
Nazwisko: Kowalski  
Telefon: 123456789  
Email: jan.kowalski@example.com

Rysunek 9: Szczegóły transakcji

## 10.4 Nowa Transakcja



**System Zarządzania Finansami**

Użytkownik: Jerzy Jancewicz

Rola: Księgowy

**Nowa transakcja**

Typ transakcji:

Wpłata

Kwota:

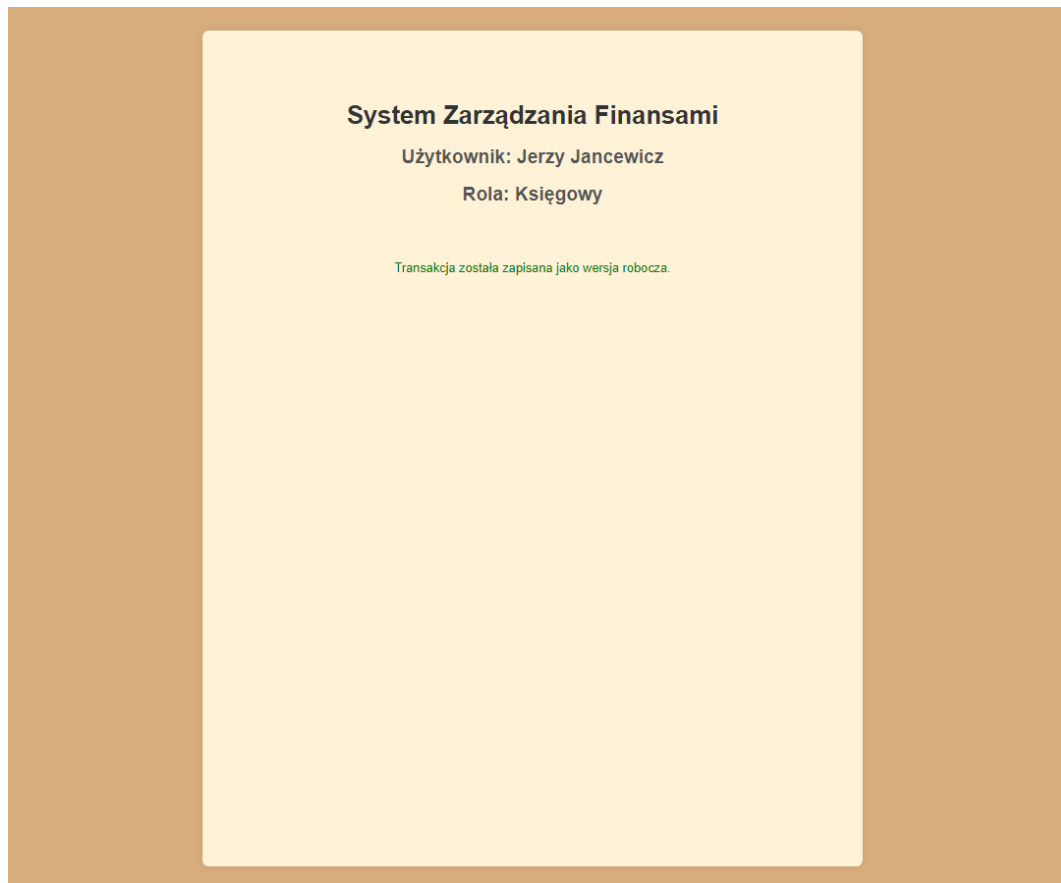
2000

Wykonaj Dodaj jako wersja robocza

Rysunek 10: Nowa Transakcja

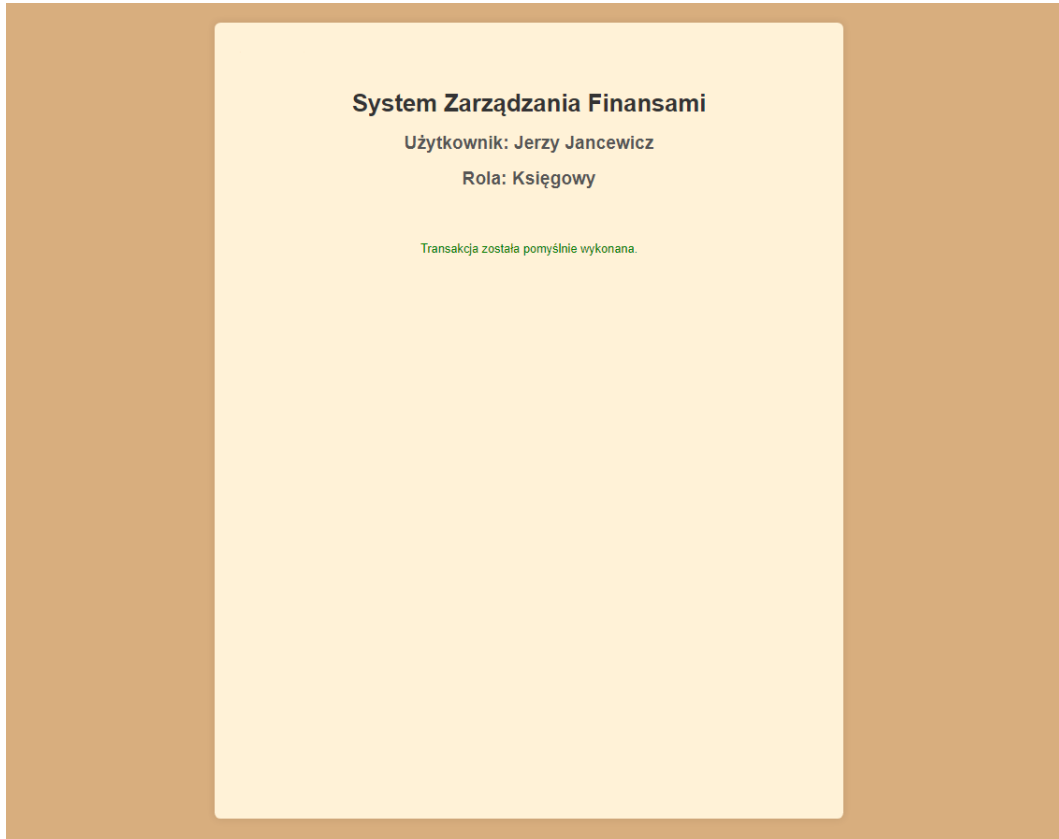


## 10.5 Akceptacja Wersji roboczej



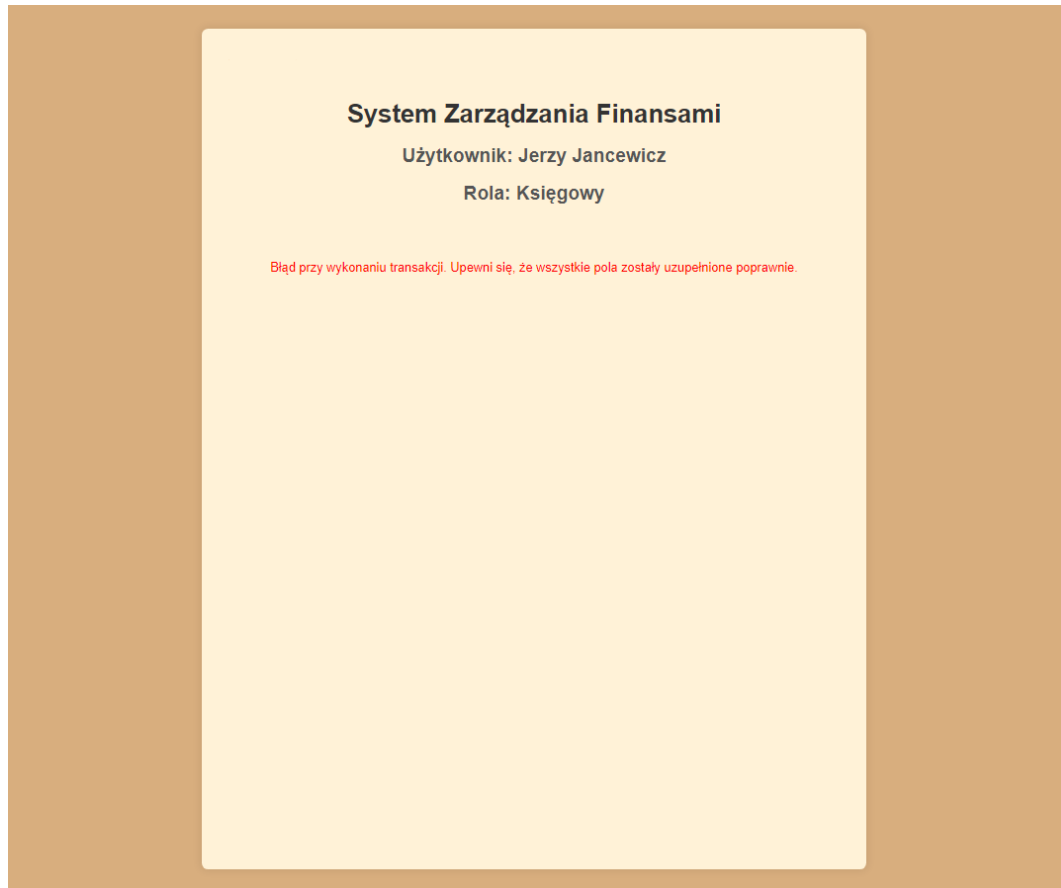
Rysunek 11: Akceptacja Wersji roboczej

## 10.6 Akceptacja transakcji



Rysunek 12: Akceptacja transakcji

## 10.7 Odrzucenie transakcji



Rysunek 13: Odrzucenie transakcji

## 11 Omówienie decyzji projektowych i skutków analizy dynamicznej

Aplikacja została stworzona w formie web aplikacji, ze względu na łatwy dostęp potencjalnych użytkowników do tego programu.

Ze względu na łatwość implementacji modeli i struktur obiektowych aplikacja została napisana, od strony backendu, w języku C# .Net.

Trwałość została zaimplementowana przy użyciu ORM - (Object Relational Mapping), a dokładnie Entity Framework Core. Użycie ORM znacząco ułatwia walidację modeli i implementację struktur obiektowych.

Po stronie Frontendu (GUI) został użyty Javascript, a dokładnie biblioteka React, która w znaczący sposób ułatwia tworzenie web aplikacji.

Część struktur obiektowych UML niestety nie da się przełożyć w całości z diagramu na języki obiektowe. W celu implementacji tego typu struktur trzeba było podjąć odpowiednie decyzje projektowe, które zostały opisane poniżej.

- Dziedziczenie overlapping na klasach Użytkownik - (Specjalista finansowy, Księgowy):  
Te dziedziczenie zostało zaimplementowane poprzez spłaszczenie hierarchii klasy Użytkownik. Ze względu na liczne powiązania (asocjacje) klasy "Użytkownik" i w celu zapobiegnięcia niechcianego dostępu do asocjacji klas "Księgowy" i "Specjalista finansowy" przez inne klasy dziedziczące po użytkowniku, została stworzona klasa pośrednia "Osoba finansowa", która dziedziczy po klasie użytkownik i posiada wszystkie asocjacje klasy "Księgowy i Specjalista finansowy". "Osoba finansowa" implementuje interfejsy "IKsięgowy" i "ISpecjalista finansowy", a dostęp do poszczególnych metod, zmiennych jest oparta na strukturze enum, która zawiera pola "KSIĘGOWY" i "SPECIALISTA FINANSOWY", które są odpowiedzialne za nazwy ról.
- Stałe W klasie Transakcja:  
Odpowiednio "Rodzaj transakcji" i "Status" korzystają ze struktur enum. Są zaimplementowane w postaci Listy z danym typem enum odpowiadającą za typ transakcji, czy statusu.

- Asocjacja z atrybutem (Budżet - Transakcja - Użytkownik):  
W tym celu zamiast bezpośrednio połączyć ze sobą klasy "Użytkownik" i "Budżet", zostały zastosowane dwie asocjacje jeden do wiele między "Budżet" a "Transakcja" i "Użytkownik" a "Transakcja" (klasa Transakcja stała się tabelą asocjacyjną). Przy tej implementacji trzeba upewnić się, że klucze obce odpowiadające klasom "Budżet" i "Użytkownik" są unikalne.
- Stałe W klasie Rachunek:  
Podobnie jak w powyższym przykładzie, typ transakcji jest zaimplementowany jako lista o typie enum "Typy".
- Dziedziczenie typu dynamic na klasach (Firma - (Klient, Dostawca)):  
Te dziedziczenie zostało zaimplementowane poprzez spłaszczenie hierarchii klasy Firma. Klasa firma implementuje dwa interfejsy "IKlient" i "IDostawca". W celu walidacji i identyfikacji dostępu do poszczególnych metod i zmiennych, została stworzona struktura enum "Typ zleceniodawcy".
- Dziedziczenie typu disjoint na klasach (Operacja Finansowa - Raport finansowy, Prognoza finansowa, Plan finansowy):  
Przez dużą różnorodność klas dziedziczących, zostało zaimplementowane dziedziczenie typu "TPT" - Table-per-type. Jest to wolniejsze podejście w porównaniu do typu "TPH" - Table-per-Hierarchy, inaczej Single-table, lecz przez złożoność i licznosc różnych asocjacji ten typ pozwoli na większą czytelność zapytań i mitygację ryzyka dotyczącego niechcianego dostępu innej klasy do danej asocjacji.