

# Końcowy przegląd projektu

Nazwa projektu: TURBOBANK.

Zespół wykonawcy w składzie:

1. Marcin Kurzawski

## Krótką charakterystyka organizacji zamawiającego

- Zamawiający jest członkiem zarządu banku Superbank i działa w imieniu banku.
- Bank został założony przez międzynarodowy fundusz inwestycyjny dwa lata temu. Jego siedziba znajduje się w prestiżowej lokalizacji Żłotej 44 w Warszawie. Firma podzielona jest na kilka działów, z których najważniejszymi jest dział obsługi udzielonych kredytów, dział analityczny oraz dział obsługi klienta.
- Superbank jest instytucją finansową, która zajmuje się działalnością kredytową.
- Superbank zatrudnia ponad 3000 pracowników, z czego 2000 to pracownicy działu obsługi klienta.



# Cel projektu

Celem projektu jest zbudowanie systemu informatycznego TURBOBANK Bigdata, który umożliwi automatyzację procesów związanych z działalnością kredytową banku Superbank. Pozwoli to zmniejszyć koszty działalności banku, przyspieszyć proces wydawania decyzji oraz ułatwić kontakt pomiędzy bankiem, a jego klientami.

# Żądania Zamawiającego

## **STRQ1: Przechowywanie informacji o klientach banku**

Bank Superbank wymaga, aby system przechowywał wszystkie informacje o kliencie

## **STRQ 2: Wspomaganie procesu obsługi wniosku kredytowego w zakresie obliczania scoringu kredytowego, wykrywania podejrzanych transakcji i jego opiniowania**

Bank Superbank wymaga, aby system wykonywał automatyczne kalkulacje zdolności kredytowej po wypełnieniu wniosku kredytowego

## **STRQ3: Wspomaganie opiniowania wniosku**

Bank Superbank wymaga, aby system automatycznie obliczał zdolność kredytową oraz ułatwiał Analitykowi podjęcie decyzji analitycznej

## **STRQ4: Automatyzację generacji raportów dziennych dotyczących podejrzanych transakcji oraz raportów działalności banku wysyłanych do Komisji Nadzoru Finansowego**

Bank Superbank wymaga, aby system automatycznie generował raport dzienny podejrzanych transakcji oraz wykonywany na jego podstawie raport dzienny z działalności banku

## **STRQ5: Przechowywanie informacji o wnioskach kredytowych i dokumentach związanych z wnioskiem;**

Bank Superbank wymaga, aby system przechowywał wszystkie informacje o danym wniosku kredytowym oraz dokumenty w formie elektronicznej, które są z nim związane

## **STRQ6: Przechowywanie raportów dziennych z działalności banku**

Bank Superbank wymaga, aby system przechowywał wszystkie wygenerowane raporty dzienne z działalności banku



# Wymagania na system (FEAT)

<b>FEAT1:</b>	<b>FUN-024-Kurzawski Przyspieszenie obliczania decyzji analitycznej</b>
<b>FEAT1.1:</b>	<b>FUN-029-Kurzawski Ułatwienie pracy pracownikowi obsługi klienta</b>
<b>FEAT2:</b>	<b>FUN-028-Kurzawski Ułatwienie wydawania opinii analitycznej w decyzji analitycznej</b>
<b>FEAT3:</b>	<b>FUN-027-Kurzawski Przyspieszenie obiegu informacji w firmie</b>
<b>FEAT4:</b>	<b>FUN-026-Kurzawski Przechowywanie dokumentów w formie cyfrowej</b>
<b>FEAT5:</b>	<b>FUN-025-Kurzawski Ułatwienie komunikacji pomiędzy bankiem a klientem</b>

# Wymagania funkcjonalne na system (UC)

<b>UC1:</b>	Zarządzanie procesowaniem wniosku kredytowego
<b>UC1.1:</b>	FUN-007-Kurzawski Automatyczne obliczanie zdolności kredytowej
<b>UC1.2:</b>	FUN-006-Kurzawski Zarządzanie przepływem wniosku
<b>UC1.3:</b>	FUN-005-Kurzawski Zarządzanie statusem wniosku
<b>UC1.4:</b>	FUN-002-Kurzawski Zarządzanie szczegółami kredytu
<b>UC1.5:</b>	FUN-004-Kurzawski Zarządzanie informacjami o kliencie
<b>UC1.6:</b>	FUN-003-Kurzawski Zarządzanie dokumentami elektronicznymi
<b>UC1.7:</b>	FUN-008-Kurzawski Wydawanie opinii analitycznej

# Wymagania funkcjonalne na system (UC)

<b>UC1:</b>	FUN-009-Kurzawski Zarządzanie procesem codziennego raportowania
<b>UC1.1:</b>	FUN-017-Kurzawski Zarządzanie opcjonalnymi parametrami do raportu dziennego
<b>UC1.2:</b>	FUN-018-Kurzawski Automatyczne przekazywanie raportu dziennego do KNF
<b>UC1.3:</b>	FUN-012-Kurzawski Automatyczne generowanie codziennych raportów
<b>UC1.4:</b>	FUN-015-Kurzawski Zarządzanie raportami dziennymi z działalności banku
<b>UC1.5:</b>	FUN-016-Kurzawski Zarządzanie raportami dziennymi z podejrzanych transakcji



# Mapowanie żądań na wymagania - macierz

[illegible]



# Specyfikacja wymagań dodatkowych

## 1 Wymagania jakościowe:

Głównym wymaganiem jakości systemu informatycznego TURBOBANK jest bezpieczne przechowywanie i przesyłanie danych. Dostęp do danych dla pracowników możliwy jest tylko i wyłącznie z sieci wewnętrznej firmy. System musi być niezawodny pod względem przechowywania danych. Wszystkie dane muszą mieć podwójną kopię bezpieczeństwa, aby w przypadku awarii, zawsze możliwe było odtworzenie historycznych decyzji i raportów.

## 2 Wymagania systemowe i sprzętowe:

Użytkownik systemu musi mieć dostęp do komputera wewnątrz firmy z systemem operacyjnym Windows 10 lub Windows 11, podłączonego do wewnętrznej sieci internetowej. Potrzebny będzie też zainstalowana przeglądarka internetowa MS Edge, przez którą użytkownicy będą użytkowali system poprzez dedykowany UI.

# Specyfikacja wymagań dodatkowych

## 3 Wymagania wydajnościowe:

System musi być w stanie obsługiwać naraz do 2000 użytkowników systemu/pracowników banku. Przy czym obliczenie zdolności kredytowej oraz sprawdzenie podejrzanych transakcji nie powinno trwać dłużej niż 3 godziny. System musi obsługiwać duże zbiory danych (kilkaset terabajtów). Maksymalna czas odpowiedzi serwera i potwierdzenie wykonania czynności to 5 sekund.

## 4 Wymagania związane ze środowiskiem pracy:

Maksymalna temperatura otoczenia, w którym będą stały maszyny, na którym będzie działał system informatyczny TURBOBANK i na których będą przechowywane dane to 30 stopni Celsjusza. W pomieszczeniach nie mogą znajdować się żadne łatwopalne materiały. Dodatkowo musi być zainstalowany system wykrywania pożarów oraz system gaśniczy wykorzystujący do gaszenia pożarów proszek gaśniczy lub dwutlenek węgla.

## 4 Wymagania skalowalności:

Obciążenie systemu jest zmienne w czasie. Wobec tego systemu musi być odpowiednio skalowany.



# Szczegółowa specyfikacja wymagań na system:xxx

Nazwa projektu:TURBOBANK

Zespół wykonawcy w składzie:

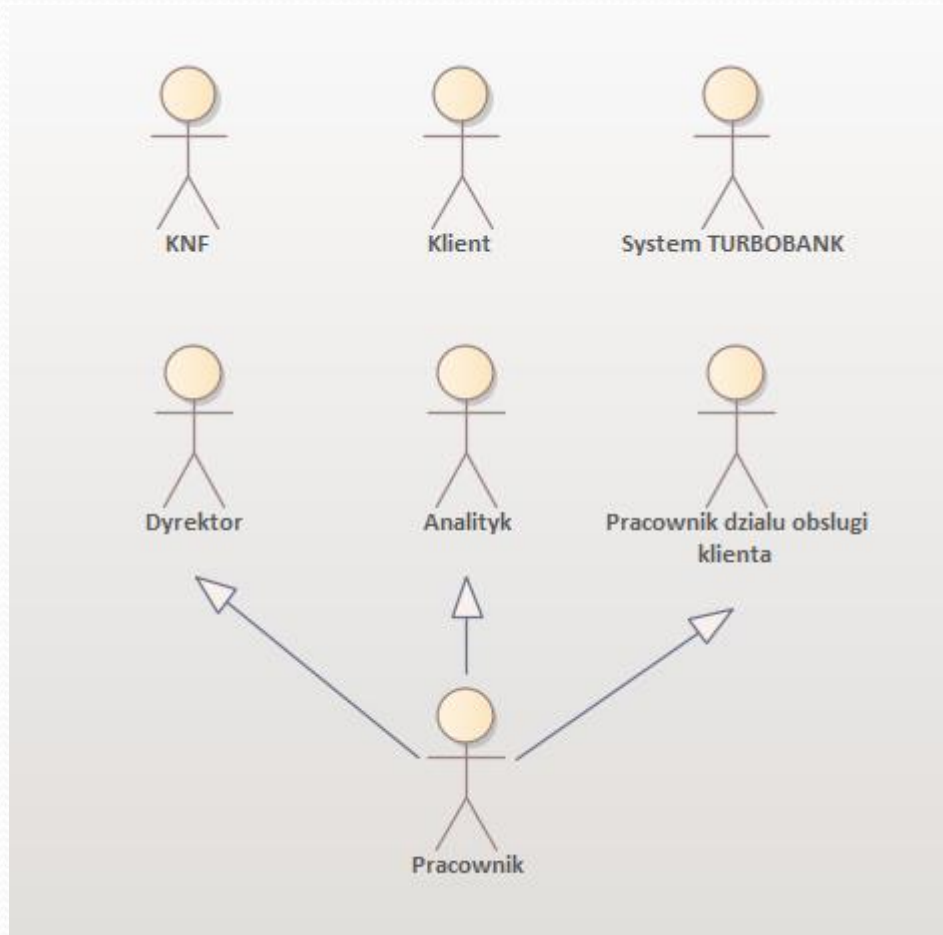
1. Marcin Kurzawski





# Model przypadków użycia

# Specyfikacja aktorów



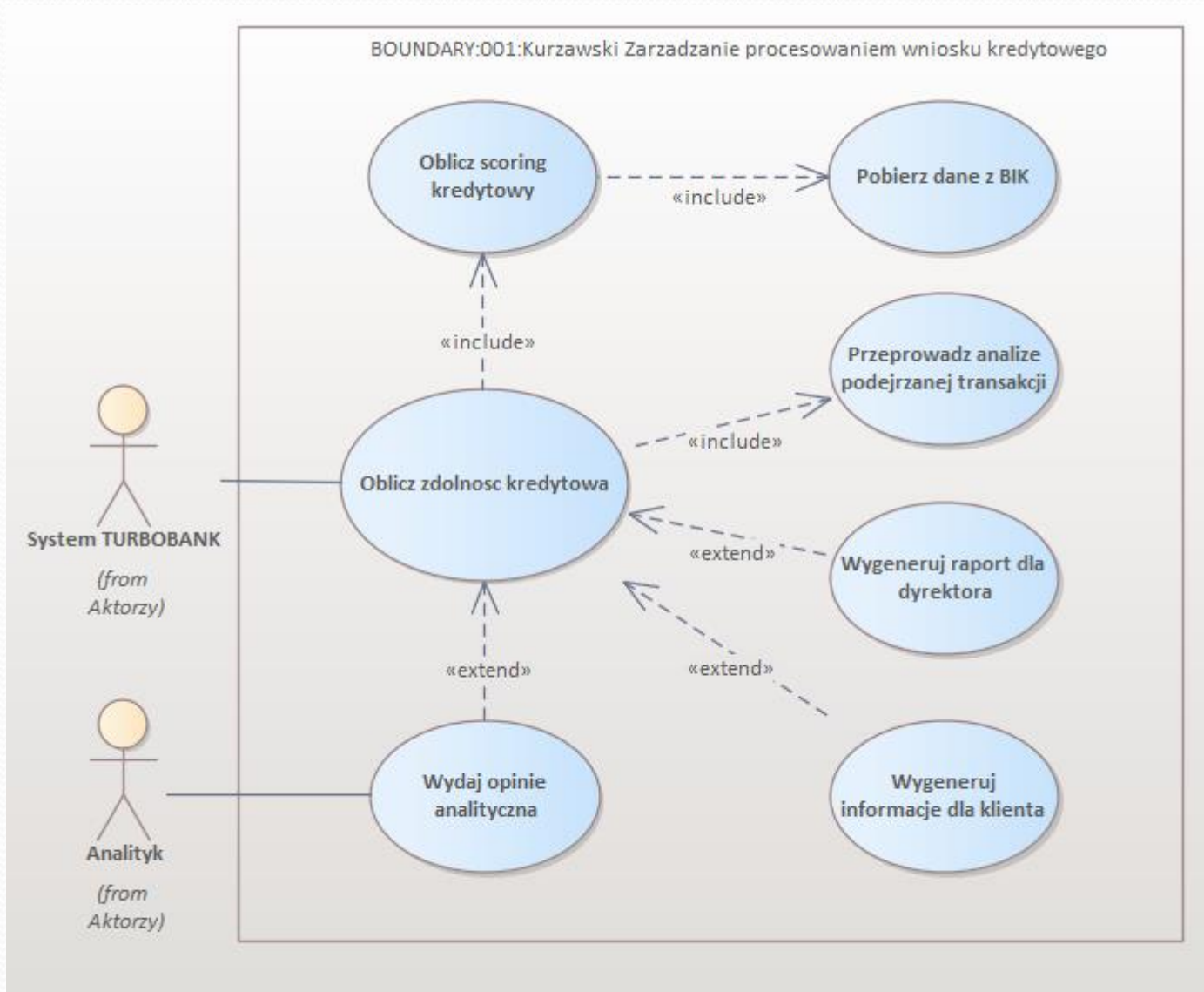
# Specyfikacja aktorów

Aktor	Opis
<b>KNF</b>	Instytucja nadzorująca bank i odbierająca i przeglądająca raporty.
<b>Klient</b>	Klient to osoba, która zwraca się do banku z prośbą o kredyt.
<b>Pracownik działu obsługi klienta</b>	Odpowiada za kontakty z klientami i wprowadzanie danych do systemu Turbobank.
<b>Analitik</b>	Analitik wydaje opinię analityczną – ocenia wnioszek.
<b>Dyrektor</b>	Nadzoruje bank Superbank, przegląda raporty z podejrzanych transakcji oraz akceptuje raporty dzienne.



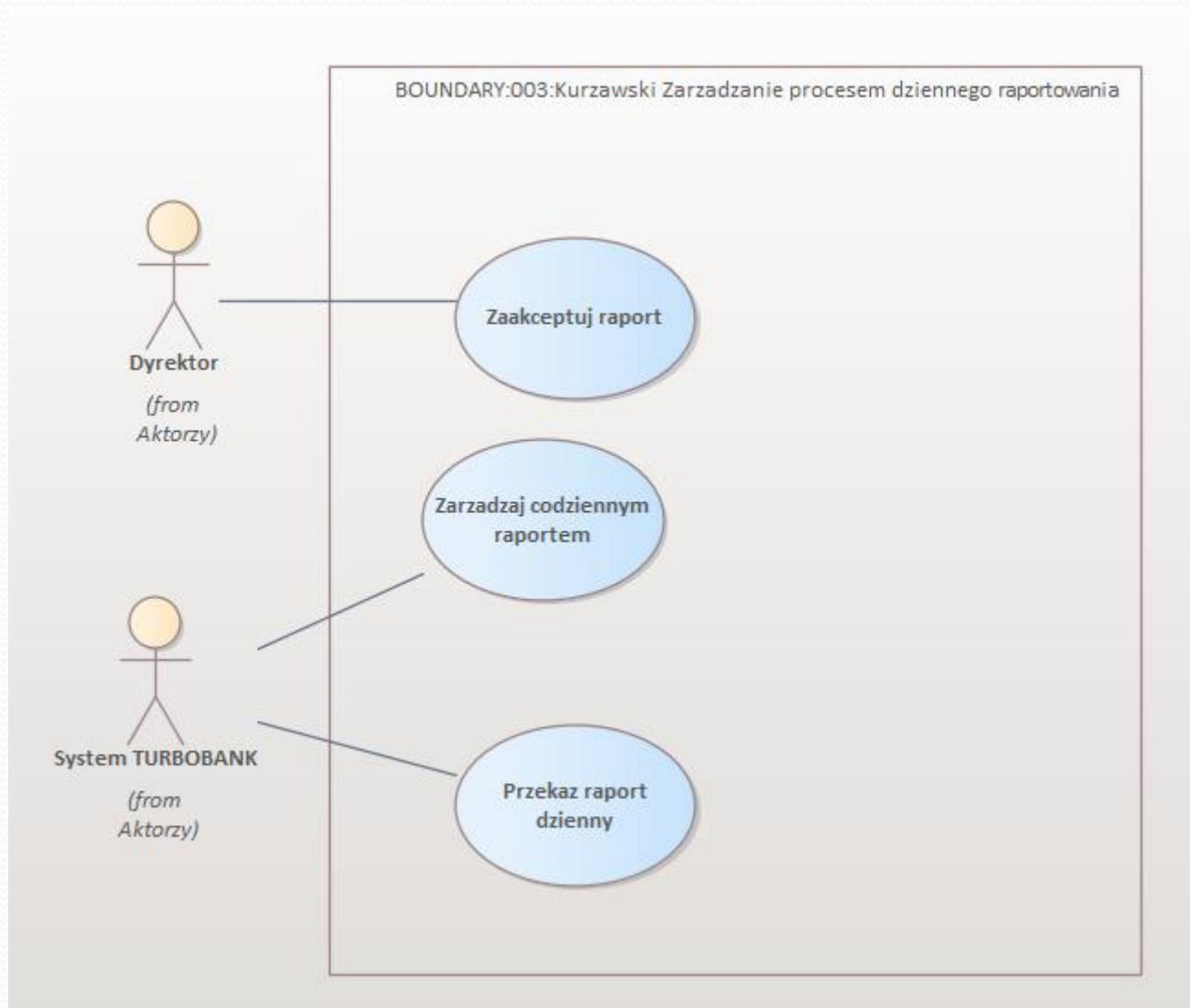
# Diagram (głównych) przypadków użycia

## Zarządzanie obliczaniem zdolności kredytowej



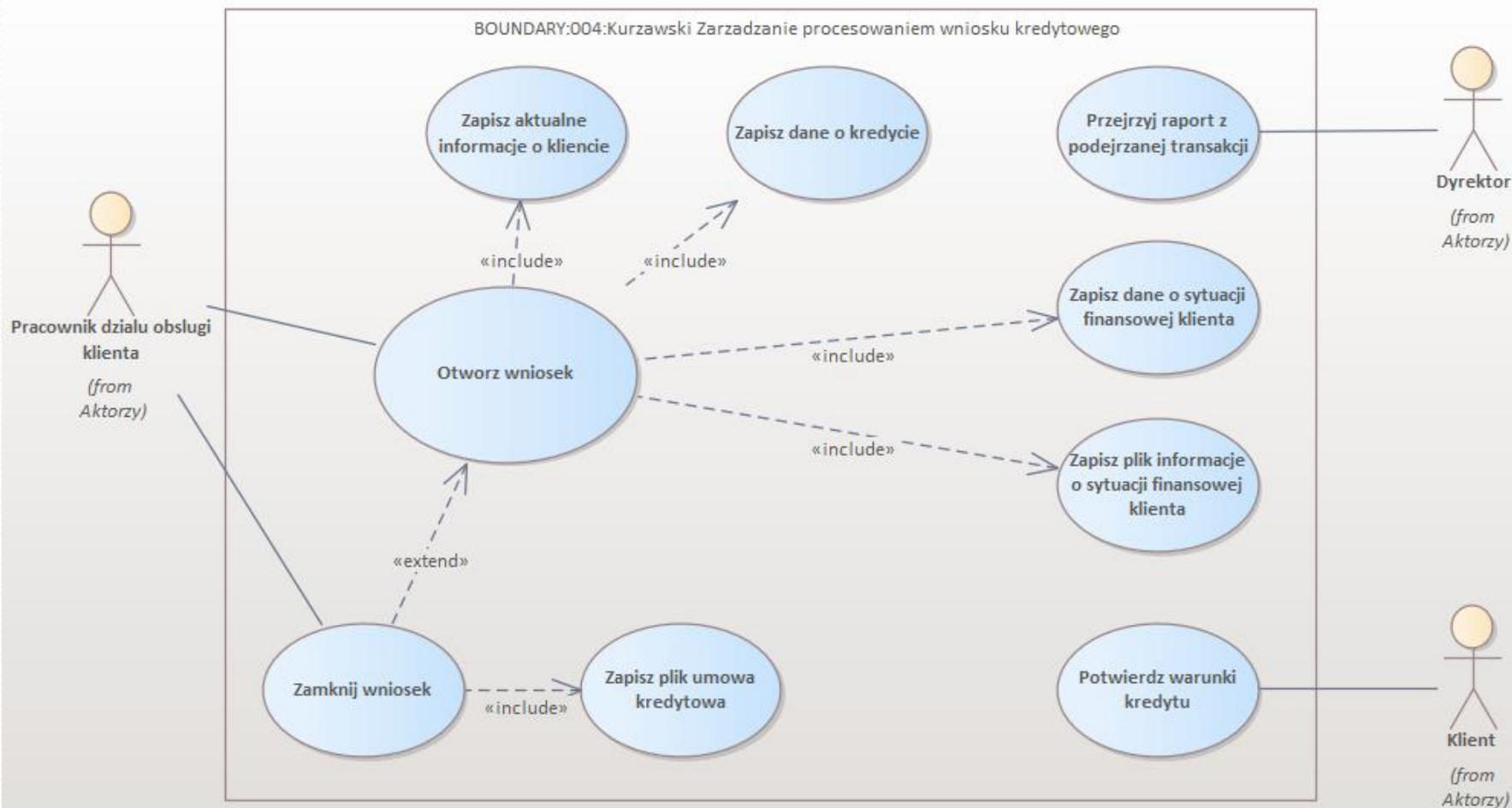
# Diagram (głównych) przypadków użycia

Zarządzanie procesem dziennego raportowania



# Diagram (głównych) przypadków użycia

## Zarządzanie procesowaniem wniosku kredytowego





# Scenariusz główny dla przypadku użycia zaakceptuj raport

Główny scenariusz dla przypadku użycia:

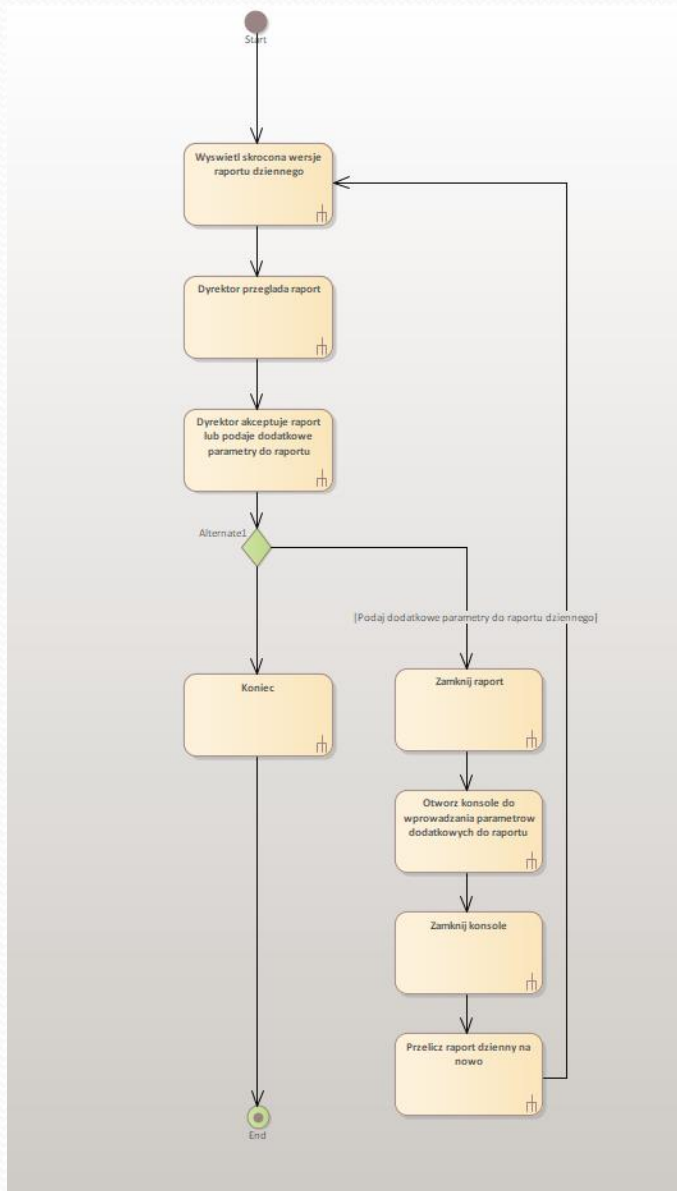
1. Dyrektor przechodzi na stronę akceptacji raportu
2. System wyświetla informacje skrócone informacje o raporcie
3. Dyrektor akceptuje raport
4. Raport zapisywany jest w systemie jako ostateczny i wysyłany do KNF

# Scenariusz alternatywny dla przypadku użycia zaakceptuj raport

Główny scenariusz dla przypadku użycia:

1. Dyrektor odrzuca raport i podaje dodatkowe parametry do raportu
2. System zamyka formularz akceptacji raportu i ponownie generuje raport dzienny z działalności banku.

# Diagram aktywności dla przypadku użycia zaakceptuj raport





# Analiza systemu TURBOBANK

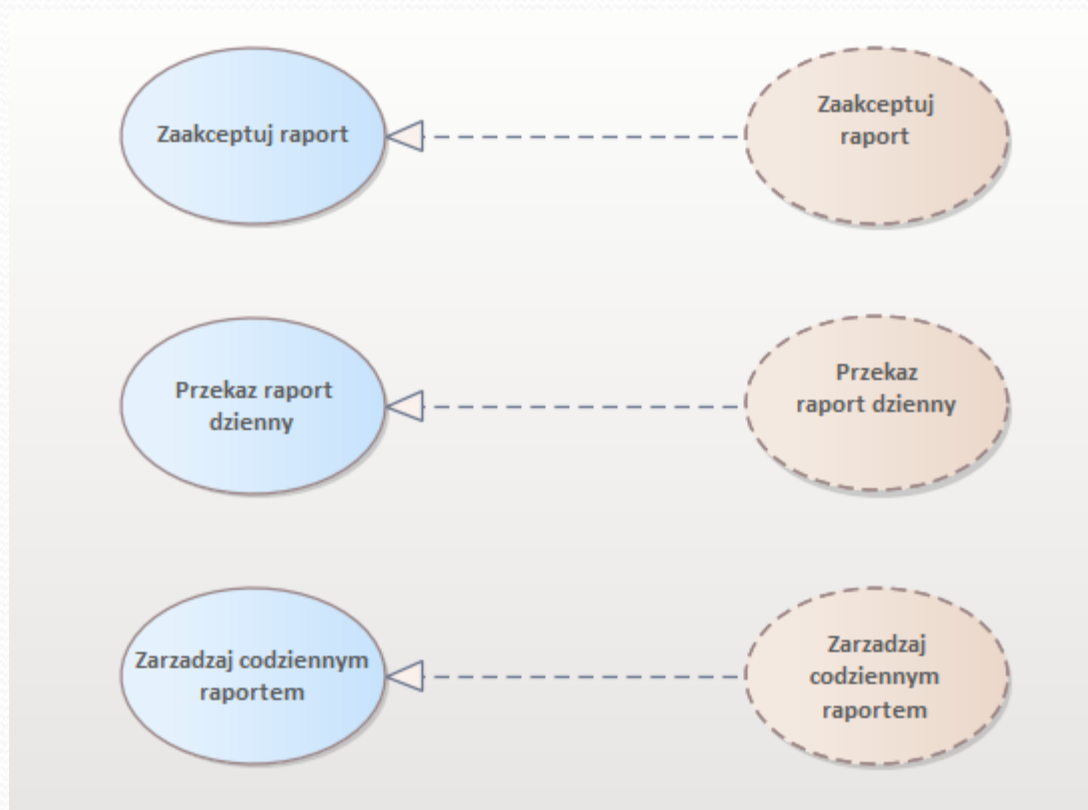
Nazwa projektu: TURBOBANK

Zespół wykonawcy w składzie:

1. Marcin Kurzawski

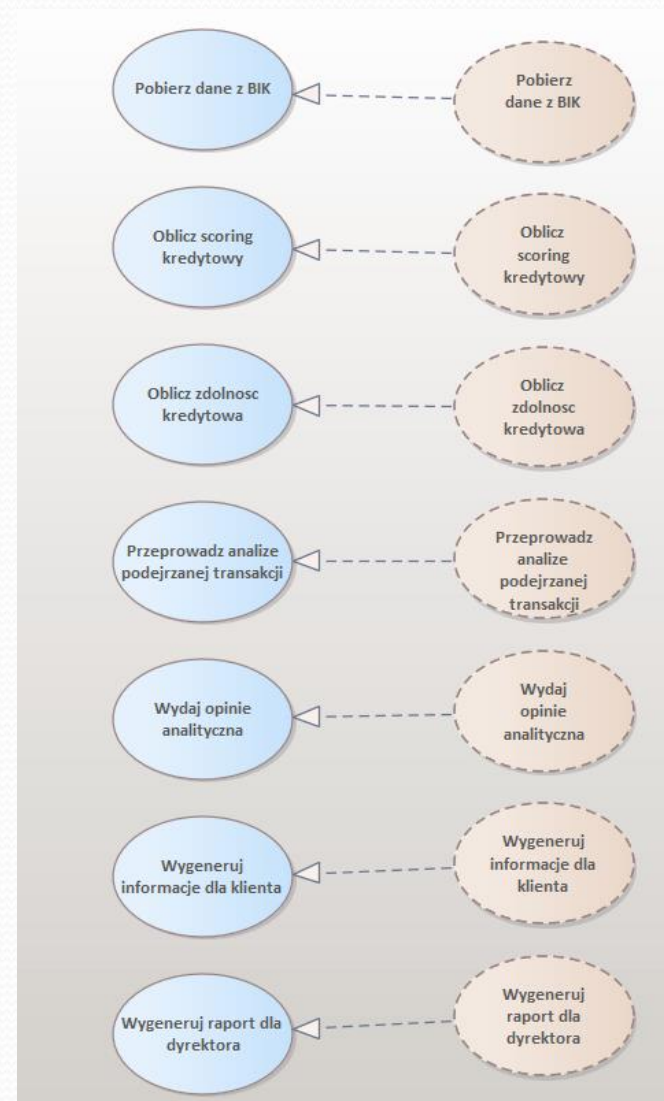
## Diagram realizacji przypadków użycia

### **Realizacje usług systemu: Zarządzanie procesem dziennego raportowania**



## Diagram realizacji przypadków użycia

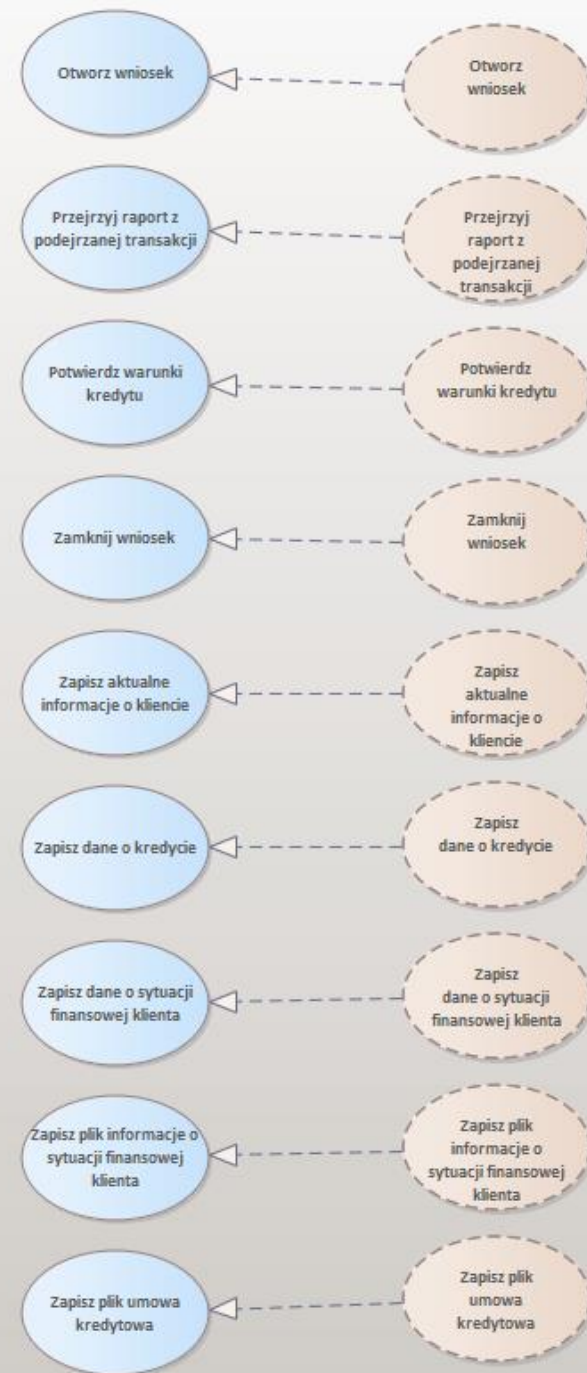
### Realizacje usług systemu: Zarządzanie obliczaniem zdolności kredytowej



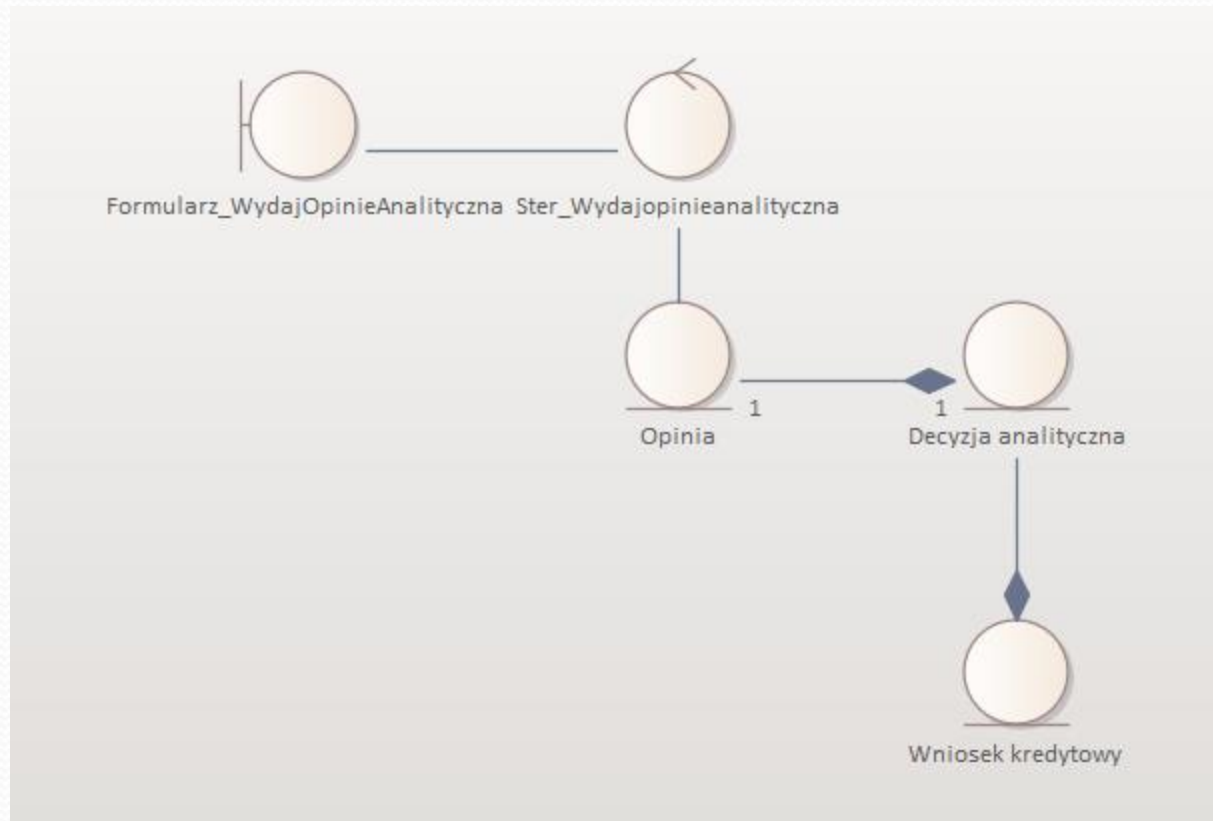


## Diagram realizacji przypadków użycia

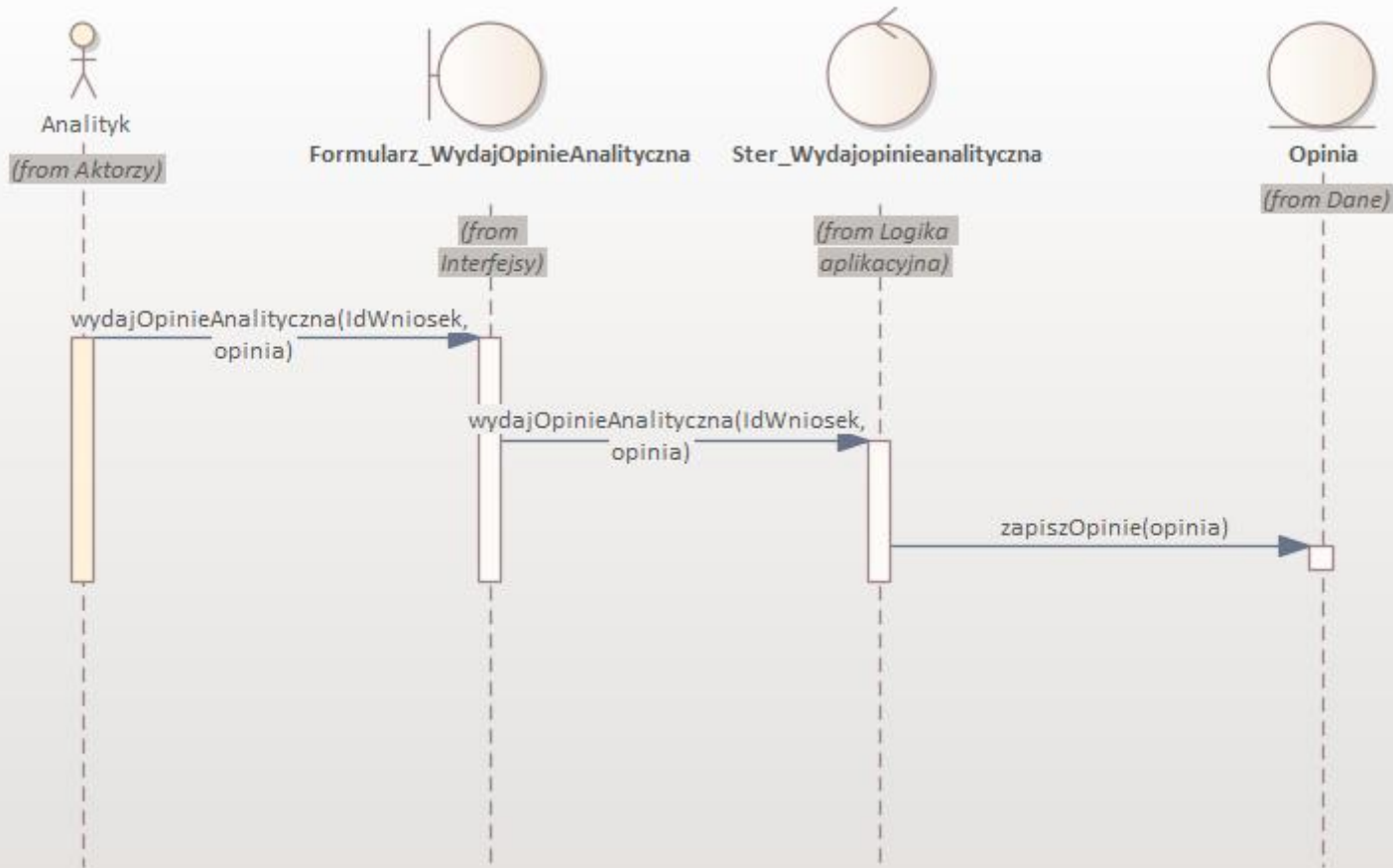
### Realizacje usług systemu: Zarządzanie procesowaniem wniosku kredytowego



# Diagram VOPC dla realizacji przypadku użycia Wydadaj opinie analityczną



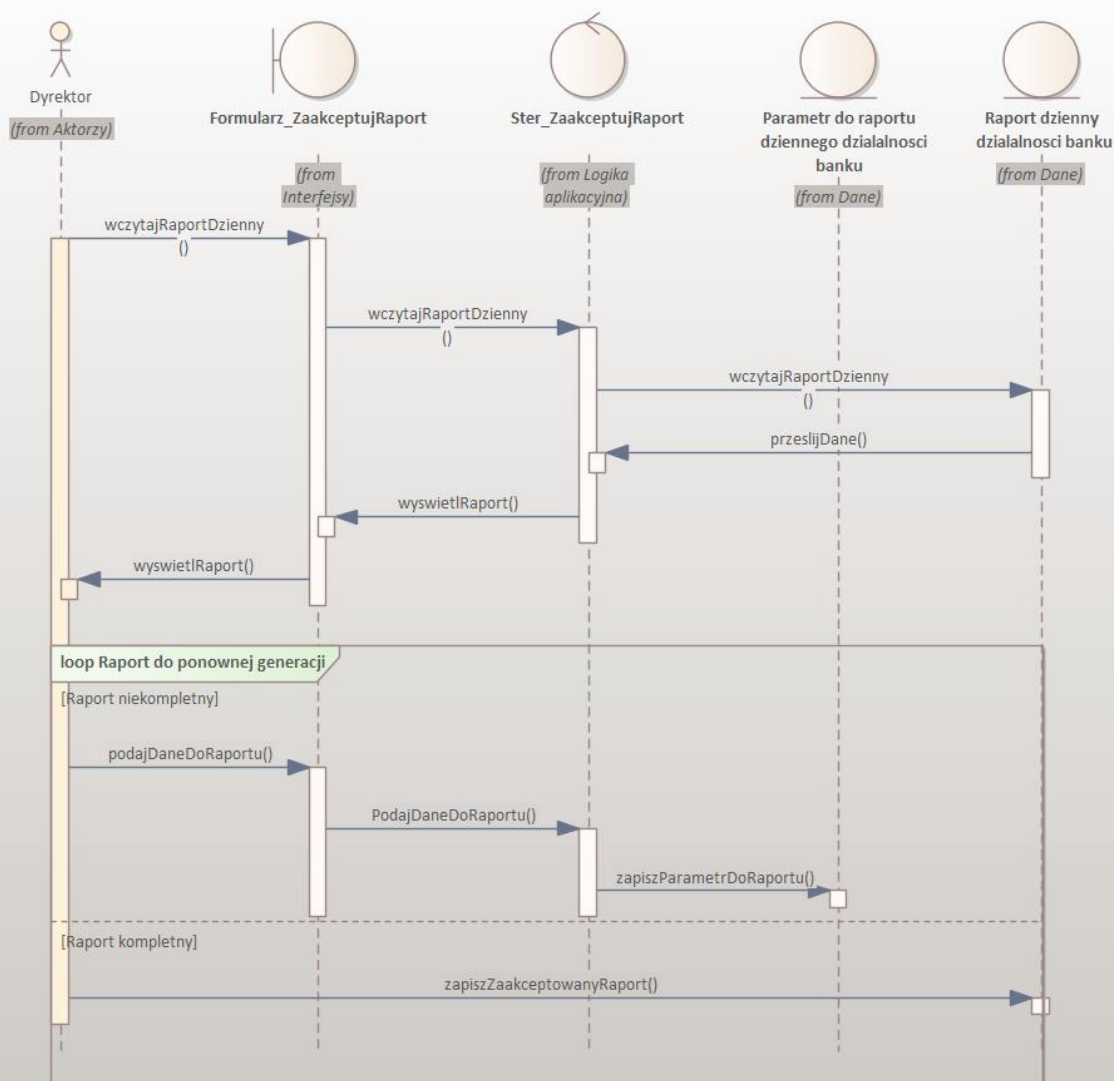
# Diagram sekwencji dla realizacji przypadku użycia Wyдай opinię analityczną





# Diagram sekwencji dla realizacji przypadku użycia

## Zaakceptuj raport



# Definicje klas analitycznych (opis)

## Klasy analityczne w ramach obszarów funkcjonalnych

W projekcie pogrupowano klasy analityczne na diagramach ze względu na obszary funkcjonalne systemu. Wyróżniono trzy takie obszary:

- Zarządzanie obliczaniem zdolności kredytowej
- Zarządzanie procesem dziennego raportowania
- Zarządzanie procesowaniem wniosku kredytowego

## Klasy analityczne w ramach roli jaką charakteryzują

Klasy analityczne podzielono na trzy odrębne zestawy.

### 1 Logika aplikacyjna

Logika aplikacyjna przedstawia w jaki sposób sterowane są poszczególne części systemu Turbobank

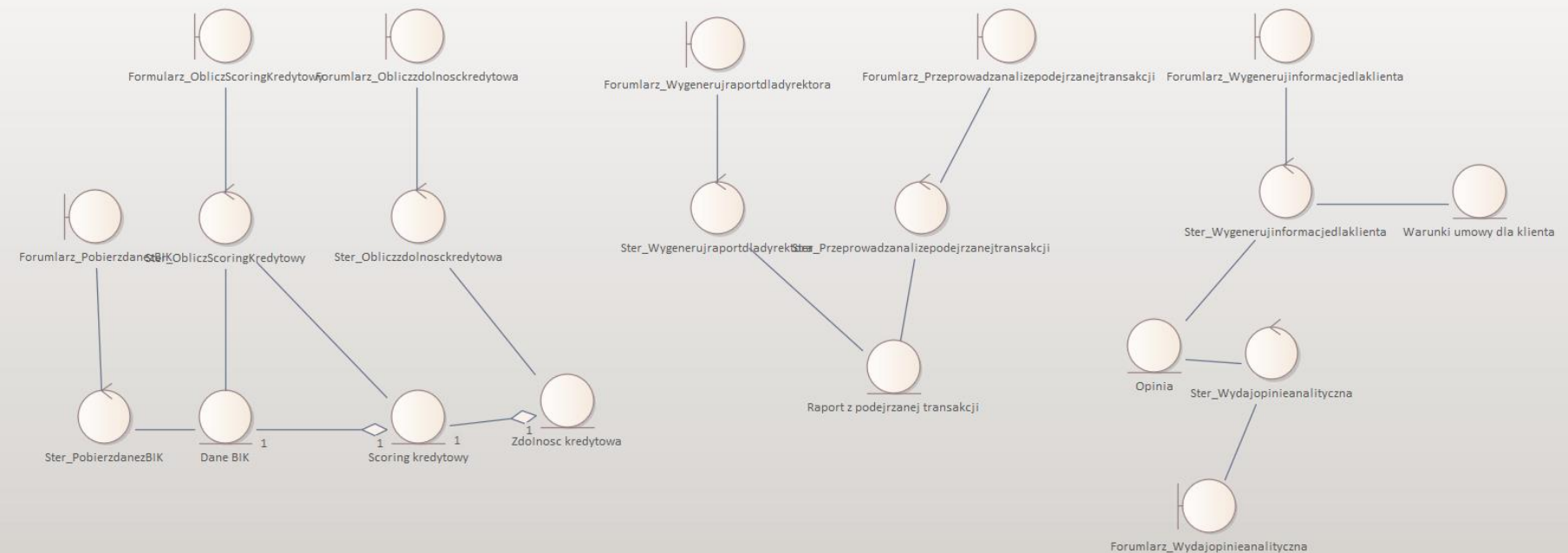
### 2 Interfejsy

Interfejsy przedstawiają ogólną definicję parametrów i wartości, które są wymieniane pomiędzy użytkownikiem i systemem lub pomiędzy poszczególnymi elementami samego systemu.

### 3 Dane

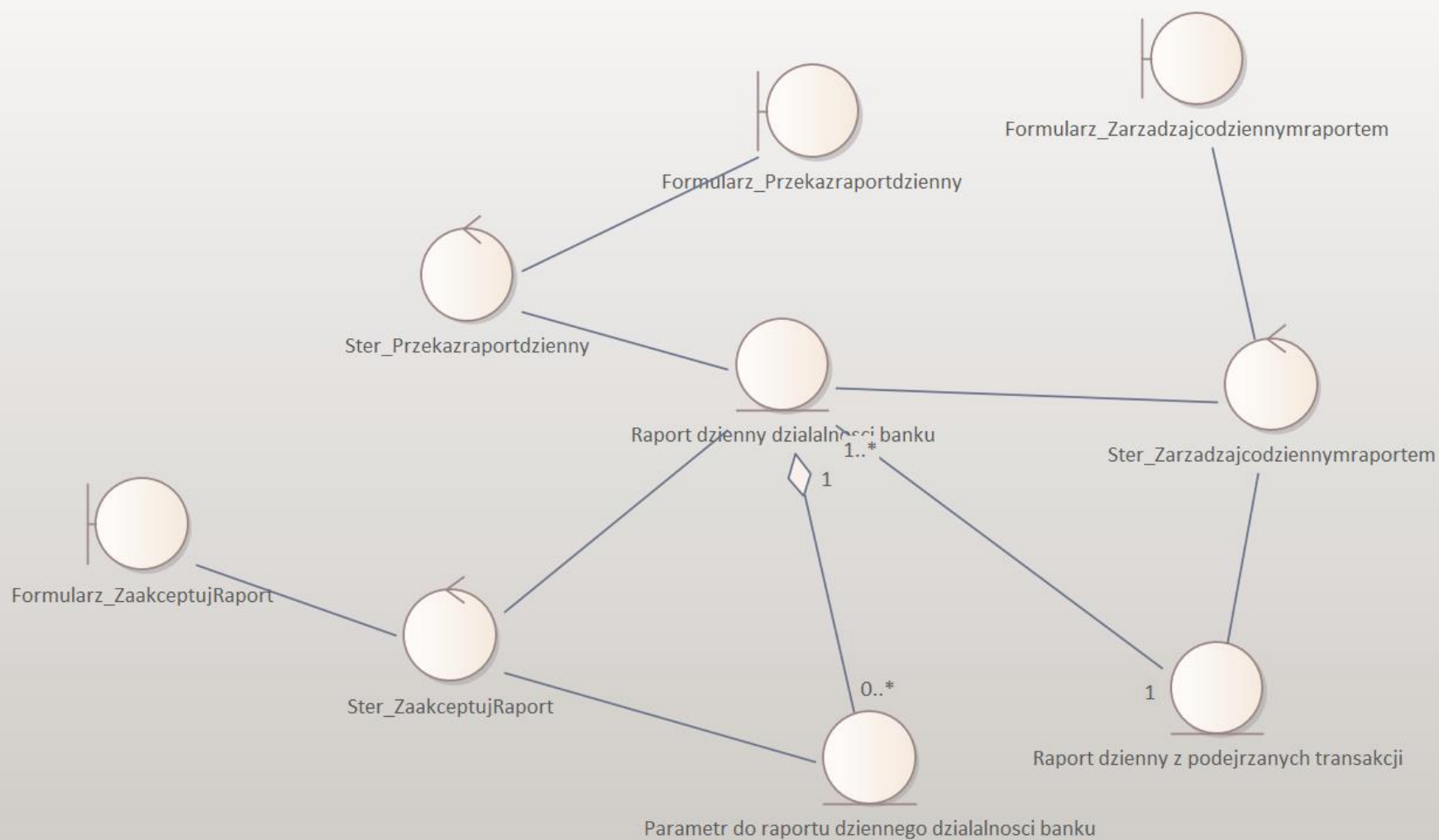
Dane prezentują informacje, które przechowuje system

# Klasy analityczne: Zarządzanie obliczaniem zdolności kredytowej

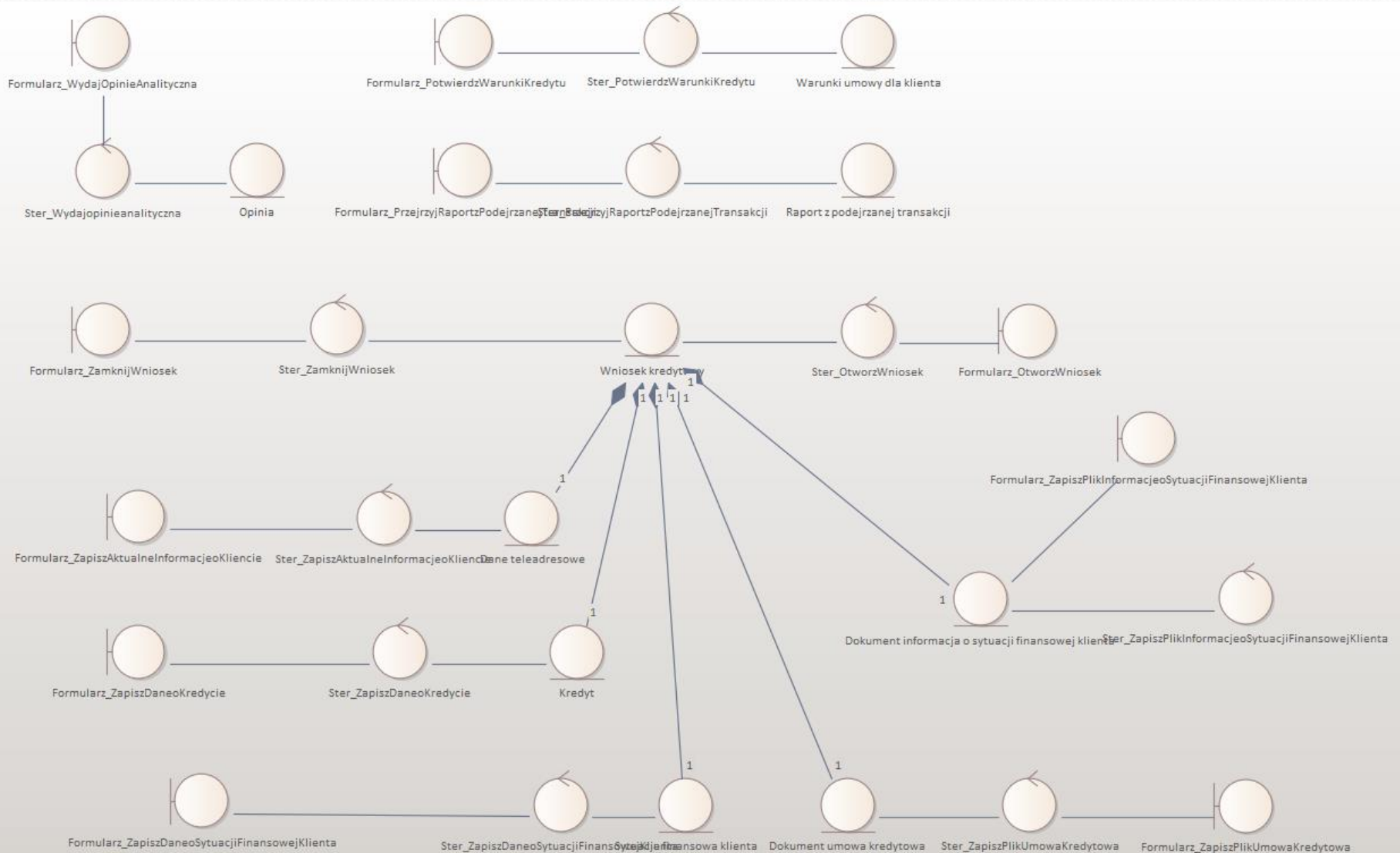




# Klasy analityczne: Zarządzanie procesem dziennego raportowania



# Klasy analityczne: Zarządzanie procesowaniem wniosku kredytowego



# Projektowanie systemu TURBOBANK

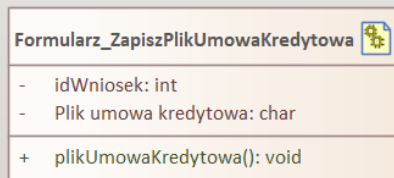
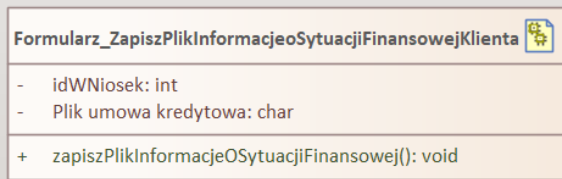
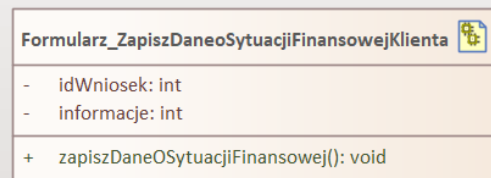
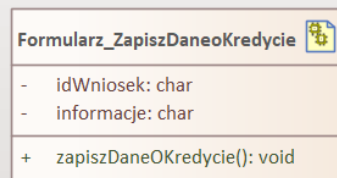
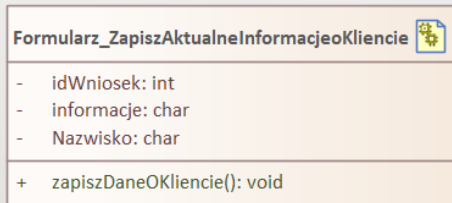
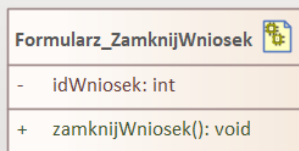
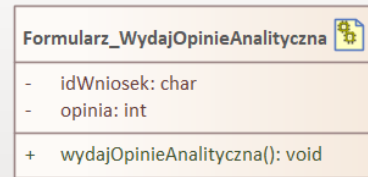
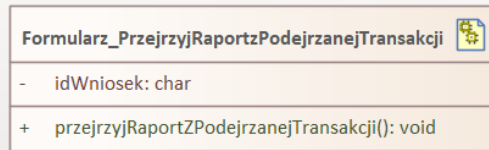
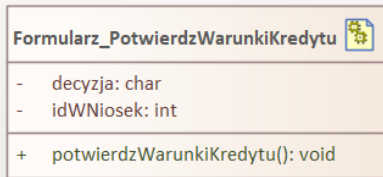
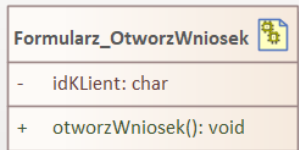
Nazwa projektu: TURBOBANK

Zespół wykonawcy w składzie:

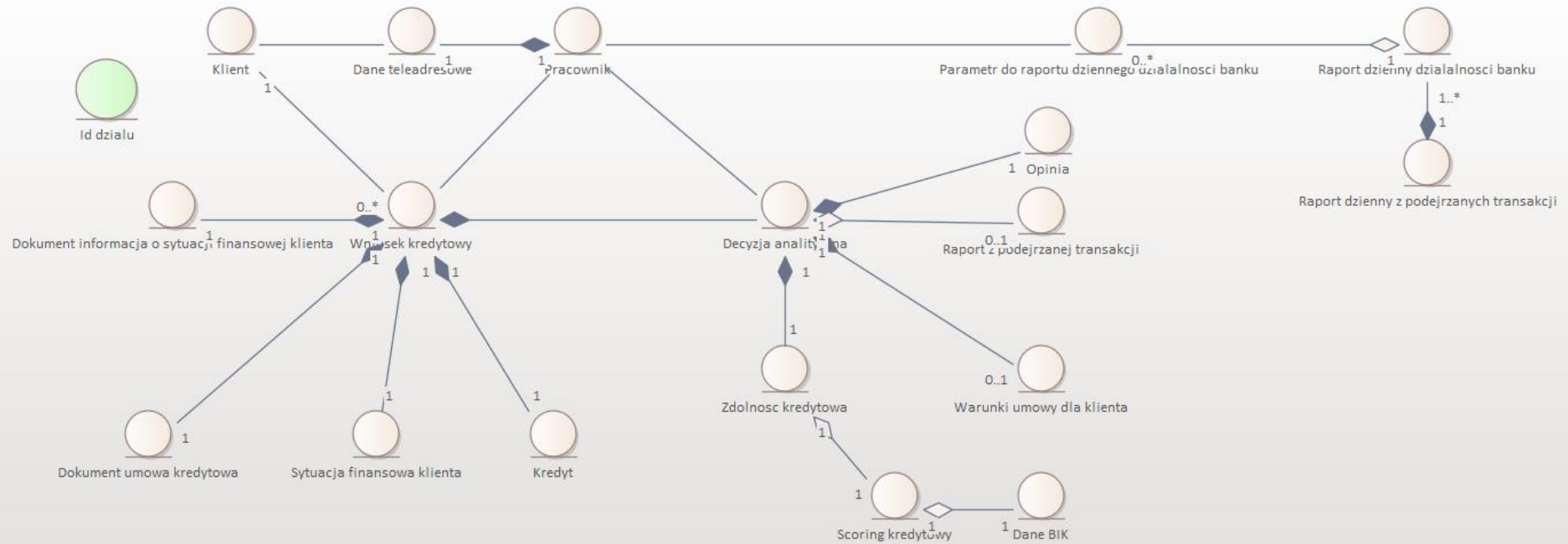
1. Marcin Kurzawski



# Fragment diagramu interfejsów



# Projektowa baza danych



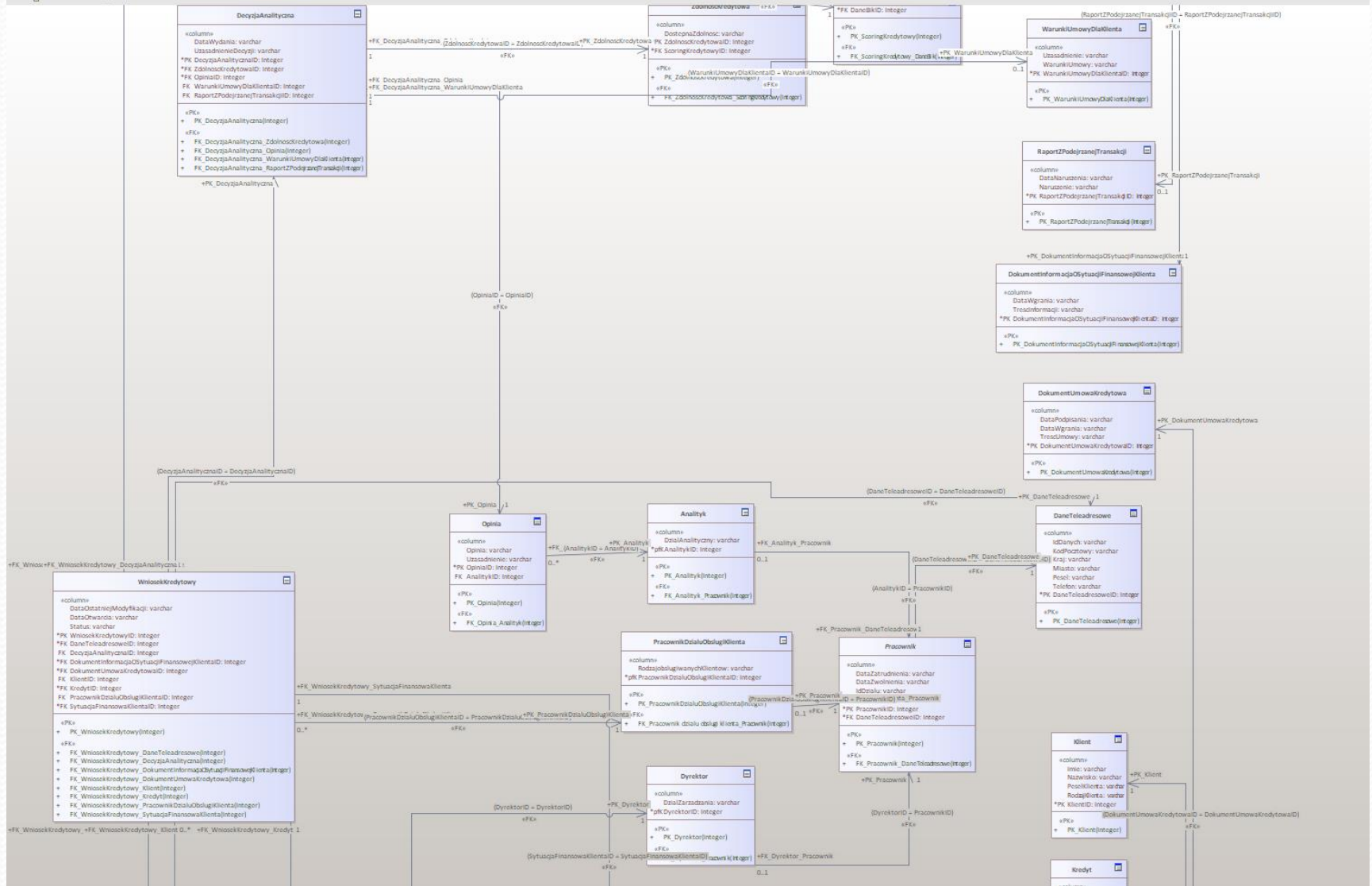
# Logika aplikacyjna



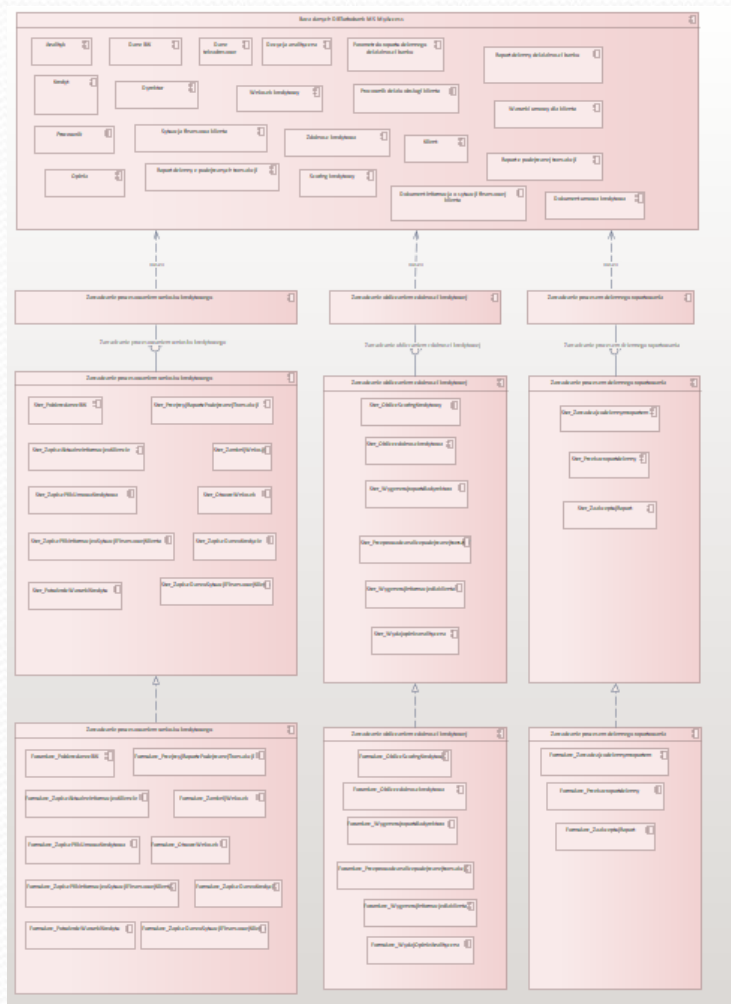


# Diagram DDL - fragment

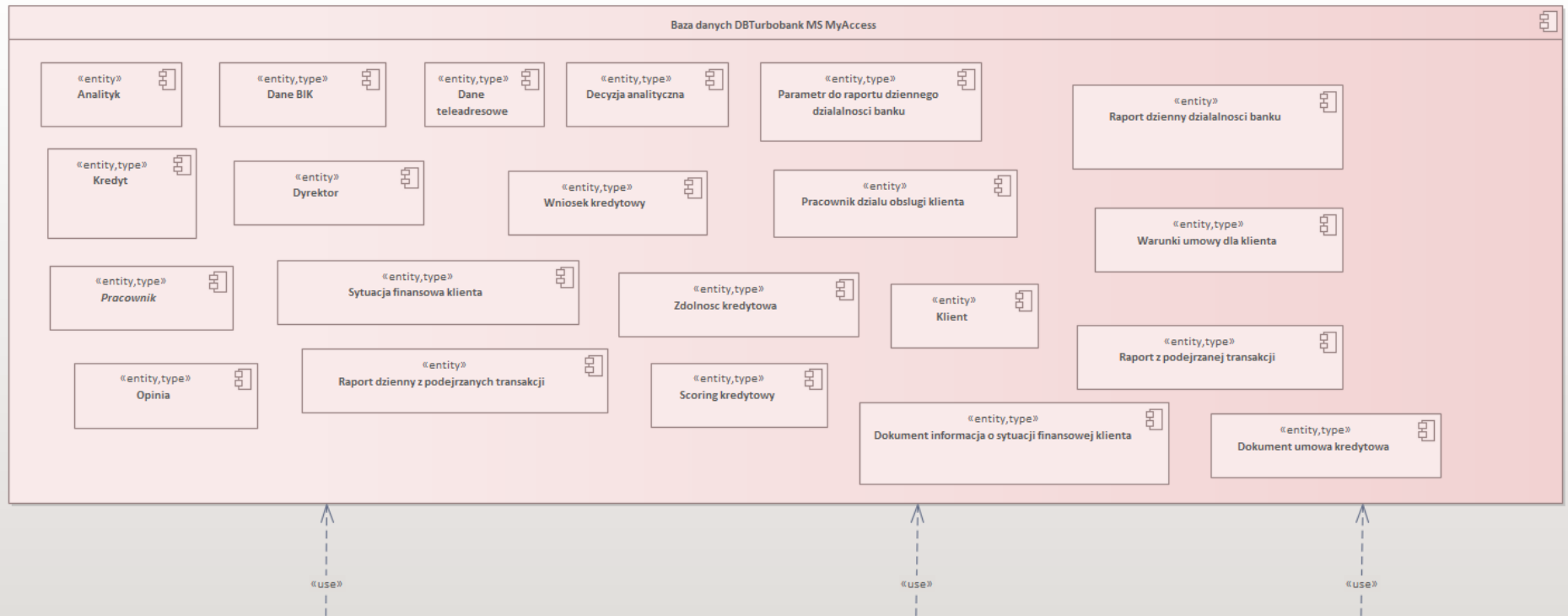
DDL Class Diagram



Ze względu na rozbudowanie systemu komponenty pogrupowano uwzględniając obszary funkcjonalne systemu informatycznego TURBOBANK

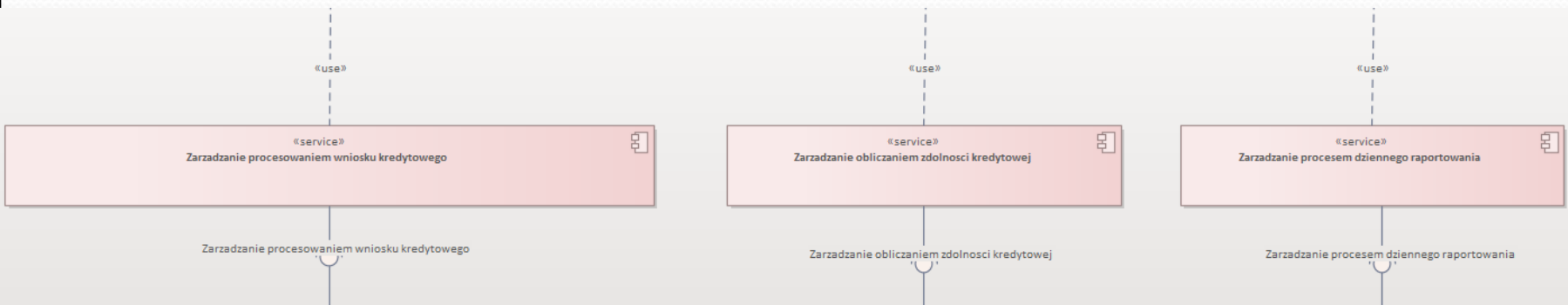


# Diagram komponentów systemu – szczegóły





# Diagram komponentów systemu – szczegóły



# Diagram komponentów systemu – szczegóły

Zarządzanie procesowaniem wniosku kredytowego

Zarządzanie obliczaniem zdolności kredytowej

Zarządzanie procesem dziennego raportowania

«process»  
Zarządzanie procesowaniem wniosku kredytowego

«process»  
Zarządzanie obliczaniem zdolności kredytowej

«process»  
Zarządzanie procesem dziennego raportowania

«process»  
Ster\_PobierzdanezBIK

«process»  
Ster\_PrzejrzyjRaportzPodejrzanejTransakcji

«process»  
Ster\_ZapiszAktualneInformacjeKlienta

«process»  
Ster\_ZamknijWniosek

«process»  
Ster\_ZapiszPlikUmowaKredytowa

«process»  
Ster\_OtworzWniosek

«process»  
Ster\_ZapiszPlikInformacjeSytuacjiFinansowejKlienta

«process»  
Ster\_ZapiszDaneoKredycie

«process»  
Ster\_PotwierdzWarunkiKredytu

«process»  
Ster\_ZapiszDaneoSytuacjiFinansowejKlienta

«process»  
Ster\_ObliczScoringKredytowy

«process»  
Ster\_ObliczZdolnoscKredytowa

«process»  
Ster\_Wygenerujraportdladyrektora

«process»  
Ster\_Przeprowadzanalizepodejrzanejtransakcji

«process»  
Ster\_Wygenerujinformacjediaklienta

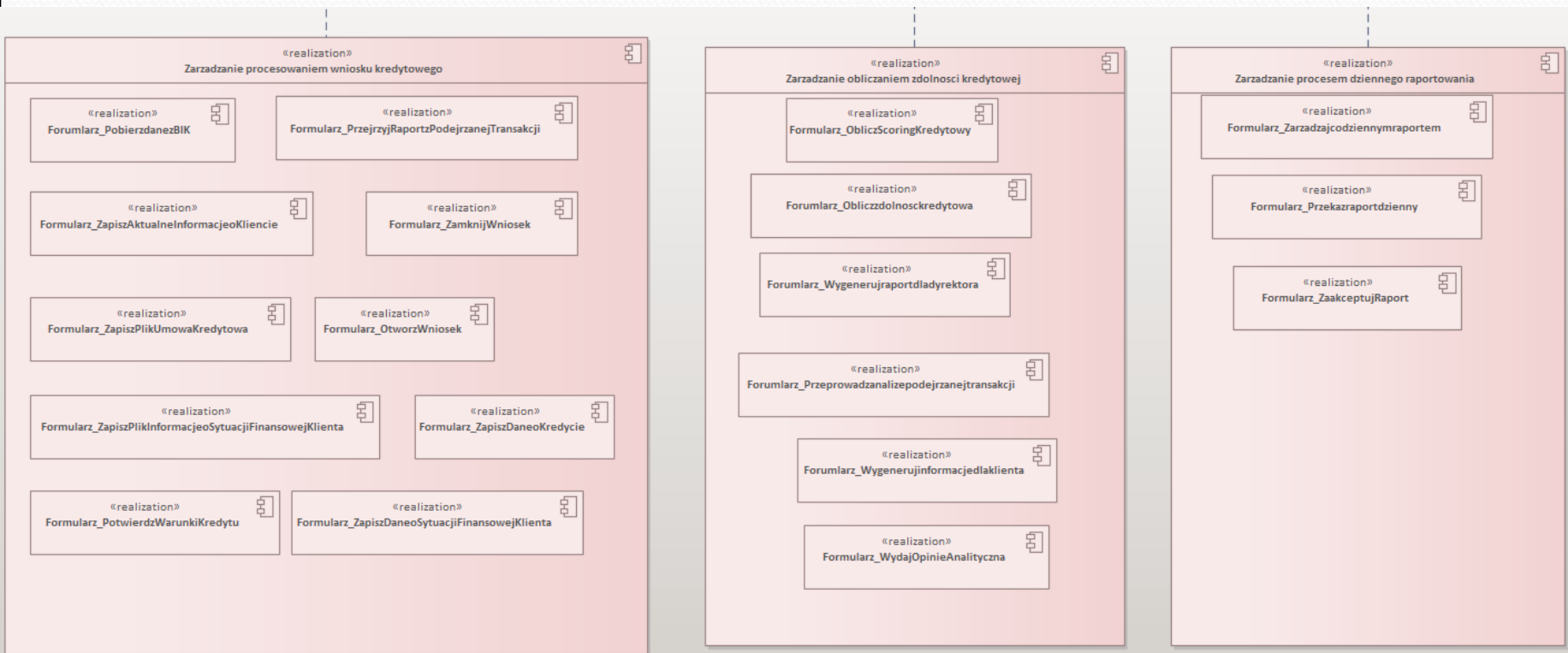
«process»  
Ster\_Wydajopinieanalityczna

«process»  
Ster\_Zarządzajcodziennymraportem

«process»  
Ster\_Przekazraportdzienny

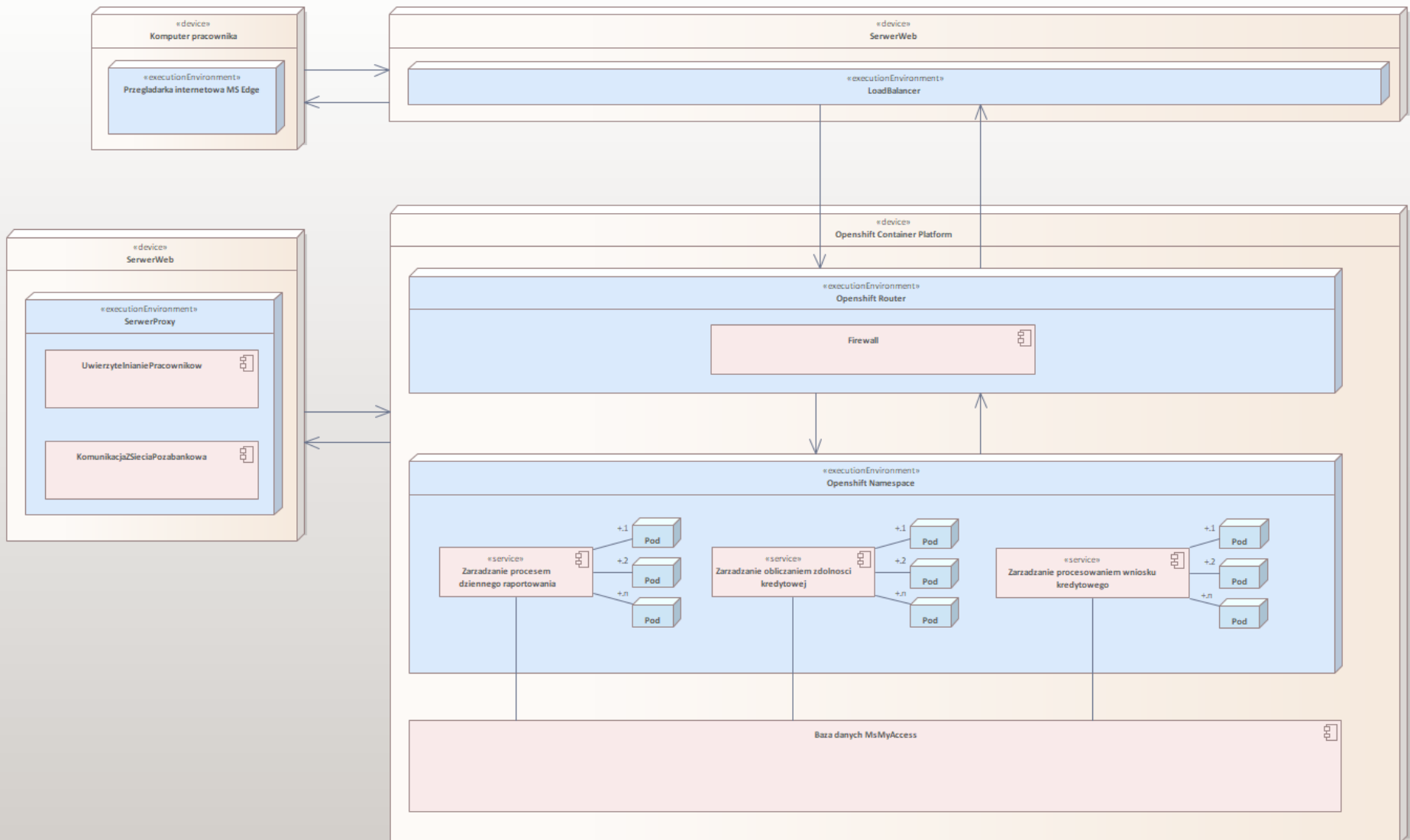
«process»  
Ster\_ZaakceptujRaport

# Diagram komponentów systemu – szczegóły – baza danych





# Diagram wdrożenia systemu



# Kod systemu: jako wynik transformacji

```
1  //////////////////////////////////////
2  // Formularz_Zaradzajcodziennymraportem.cs
3  // Implementation of the Class Formularz_Zaradzajcodziennymraportem
4  // Generated by Enterprise Architect
5  // Created on:      03-kwi-2023 02:08:10
6  // Original author: Marcin Kurzawski
7  //////////////////////////////////////
8
9  using System;
10 using System.Collections.Generic;
11 using System.Text;
12 using System.IO;
13
14
15
16 namespace Interfejsy {
17     public class Formularz_Zaradzajcodziennymraportem {
18
19         private char data;
20
21         public Formularz_Zaradzajcodziennymraportem(){
22
23         }
24
25         ~Formularz_Zaradzajcodziennymraportem(){
26
27         }
28
29         public Ster_Zaradzajcodziennymraportem Ster_Zaradzajcodziennymraportem{
30             get{
31                 return m_Ster_Zaradzajcodziennymraportem;
32             }
33             set{
34                 m_Ster_Zaradzajcodziennymraportem = value;
35             }
36         }
37
38         public void zaradzajRaportemDziennym(){
39
40         }
41
42     } //end Formularz_Zaradzajcodziennymraportem
43 } //end namespace Interfejsy
44
```

# Kod systemu: jako wynik transformacji

```
1  //////////////////////////////////////
2  // Formularz_ZapiszPlikInformacjeoSytuacjiFinansowejKlienta.cs
3  // Implementation of the Class Formularz_ZapiszPlikInformacjeoSytuacjiFinansowejKlienta
4  // Generated by Enterprise Architect
5  // Created on:      03-kwi-2023 02:07:59
6  // Original author: Marcin Kurzawski
7  //////////////////////////////////////
8
9  using System;
10 using System.Collections.Generic;
11 using System.Text;
12 using System.IO;
13
14
15
16 namespace Interfejsy {
17     public class Formularz_ZapiszPlikInformacjeoSytuacjiFinansowejKlienta {
18
19         private int idWNiosek;
20         private char Plik umowa kredytowa;
21
22         public Formularz_ZapiszPlikInformacjeoSytuacjiFinansowejKlienta() {
23
24         }
25
26         ~Formularz_ZapiszPlikInformacjeoSytuacjiFinansowejKlienta() {
27
28         }
29
30         public Dokument informacja o sytuacji finansowej klienta Dokument informacja o sytuacji finansowej klienta {
31             get {
32                 return m_Dokument informacja o sytuacji finansowej klienta;
33             }
34             set {
35                 m_Dokument informacja o sytuacji finansowej klienta = value;
36             }
37         }
38
39         public void zapiszPlikInformacjeoSytuacjiFinansowej() {
40
41         }
42
43     } //end Formularz_ZapiszPlikInformacjeoSytuacjiFinansowejKlienta
44 } //end namespace Interfejsy
```



# Kod systemu: jako wynik transformacji

```
DROP TABLE [ZdolnoscKredytowa];

CREATE TABLE [Analityk]
(
    [DzialAnalityczny] varchar NULL,
    [AnalitykID] Short NOT NULL
);

CREATE TABLE [DaneBik]
(
    [HistoriaBik] varchar NULL,
    [DaneBikID] Short NOT NULL
);

CREATE TABLE [DaneTeleadresowe]
(
    [IdDanych] varchar NULL,
    [KodPocztowy] varchar NULL,
    [Kraj] varchar NULL,
    [Miasto] varchar NULL,
    [Pesel] varchar NULL,
    [Telefon] varchar NULL,
    [DaneTeleadresoweID] Short NOT NULL
);

CREATE TABLE [DecyzjaAnalityczna]
(
    [DataWydania] varchar NULL,
    [UzasadnienieDecyzji] varchar NULL,
    [DecyzjaAnalitycznaID] Short NOT NULL,
    [ZdolnoscKredytowaID] Short NOT NULL,
    [OpiniaID] Short NOT NULL,
    [WarunkiUmowyDlaKlientaID] Short NULL,
    [RaportZPodejrzanejTransakcjiID] Short NULL
);

CREATE TABLE [DokumentInformacjaOSytuacjiFinansowejKlienta]
(
    [DataWgrania] varchar NULL,
    [TrescInformacji] varchar NULL,
    [DokumentInformacjaOSytuacjiFinansowejKlientaID] Short NOT NULL
);

CREATE TABLE [DokumentUmowaKredytowa]
(
    [DataPodpisania] varchar NULL,
    [DataWgrania] varchar NULL,
    [TrescUmowy] varchar NULL,
    [DokumentUmowaKredytowaID] Short NOT NULL
);
```

# Kod systemu: jako wynik transformacji

```
CREATE TABLE [ZdolnoscKredytowa]
(
    [DostepnaZdolnosc] varchar NULL,
    [ZdolnoscKredytowaID] Short NOT NULL,
    [ScoringKredytowyID] Short NOT NULL
);

ALTER TABLE [Analityk] ADD CONSTRAINT [PK_Analitek]
    PRIMARY KEY ([AnalitekID]);

ALTER TABLE [DaneBik] ADD CONSTRAINT [PK_DaneBik]
    PRIMARY KEY ([DaneBikID]);

ALTER TABLE [DaneTeleadresowe] ADD CONSTRAINT [PK_DaneTeleadresowe]
    PRIMARY KEY ([DaneTeleadresoweID]);

ALTER TABLE [DecyzjaAnalityczna] ADD CONSTRAINT [PK_DecyzjaAnalityczna]
    PRIMARY KEY ([DecyzjaAnalitycznaID]);

ALTER TABLE [DokumentInformacjaOSytuacjiFinansowejKlienta] ADD CONSTRAINT [PK_DokumentInformacjaOSytuacjiFinansowejKlienta]
    PRIMARY KEY ([DokumentInformacjaOSytuacjiFinansowejKlientaID]);

ALTER TABLE [DokumentUmowaKredytowa] ADD CONSTRAINT [PK_DokumentUmowaKredytowa]
    PRIMARY KEY ([DokumentUmowaKredytowaID]);

ALTER TABLE [Dyrektor] ADD CONSTRAINT [PK_Dyrektor]
    PRIMARY KEY ([DyrektorID]);

ALTER TABLE [Klient] ADD CONSTRAINT [PK_Klient]
    PRIMARY KEY ([KlientID]);

ALTER TABLE [Kredyt] ADD CONSTRAINT [PK_Kredyt]
    PRIMARY KEY ([KredytID]);

ALTER TABLE [Opinia] ADD CONSTRAINT [PK_Opinia]
    PRIMARY KEY ([OpiniaID]);

ALTER TABLE [ParametrDoRaportuDziennegoDzialalnosciBanku] ADD CONSTRAINT [PK_ParametrDoRaportuDziennegoDzialalnosciBanku]
    PRIMARY KEY ([ParametrDoRaportuDziennegoDzialalnosciBankuID]);

ALTER TABLE [Pracownik] ADD CONSTRAINT [PK_Pracownik]
    PRIMARY KEY ([PracownikID]);

ALTER TABLE [PracownikDzialuObslugiKlienta] ADD CONSTRAINT [PK_PracownikDzialuObslugiKlienta]
    PRIMARY KEY ([PracownikDzialuObslugiKlientaID]);

ALTER TABLE [RaportDziennyDzialalnosciBanku] ADD CONSTRAINT [PK_RaportDziennyDzialalnosciBanku]
    PRIMARY KEY ([RaportDziennyDzialalnosciBankuID]);
```