Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych

Architektura i Projektowanie Systemów Informacyjnych

System zarządzania wydatkami domowymi

Projekt – Iteracja 2

Zespół:
Jakub Borowski
Wojciech Matuszewski
Maciej Suchecki
Daniel Waślicki

Prowadzący:
dr inż. Andrzej Ciemski

23 czerwca 2015

Spis treści

1 Wstęp

Celem utworzenia niniejszego dokumentu jest opisanie oraz zaprezentowanie realizacji projektu systemu informatycznego dla firmy *Money Solutions sp. z o.o.* Wspomniana firma jest liderem w branży doradztwa finansowego, skierowanego głównie do klientów prywatnych. Z racji na rosnące zapotrzebowanie rynkowe na usługi doradcze w zakresie prowadzenia budżetu domowego, firma *Money Solutions* zdecydowała się na zamówienie systemu pozwalającego na zarządzanie finansami domowymi, który będzie odpłatnie udostępniany jej klientom.

Zabieg taki pozwoli na transfer niektórych obowiązków związanych ze zbieraniem informacji nt. budżetu domowego klientów do nich samych. Dzięki temu każdy z pracowników firmy *Money Solutions* będzie mógł świadczyć swoje usługi większej liczbie klientów, jednocześnie zwiększając zyski firmy. Ponadto, wspomniany system będzie mógł być również oferowany jako osobna usługa, bez wspomnianego wcześniej doradztwa pracowników firmy.

Realizacja systemu została zlecona przez firmę *Money Solutions sp. z o.o.*, zwaną dalej Zleceniodawcą, firmie *Four Developers sp. z o.o.*, zwanej dalej Zleceniobiorcą, autorem tego dokumentu. System powinien spełnić wszystkie wymagania biznesowe postawione przed Zleceniobiorcą, które zostaną opisane w dalszej części dokumentu. Zleceniobiorca zobowiązuje się do wykonania wszystkich opisanych w dokumencie części systemu, uruchomienie jej w siedzibie firmy *Money Solutions* oraz utrzymania systemu po wdrożeniu (z czym wiąże się miesięczna opłata).

2 Zakres realizacji

Niniejszy rozdział jest poświęcony opisowi zakładanego sposobu realizacji projektu. Zostają tutaj zdefiniowane zespoły, ich zadania oraz role, a także uściślany jest wstępny zarys technologiczny projektu.

2.1 Realizacja projektu

Projekt będzie realizowany w siedzibie firmy *Four Developers sp. z o.o.* z wykorzystaniem metodyki *Scrum*. Zleceniodawca zobowiązał się do oddelegowania jednego z doświadczonych pracowników, który zna specyfikę projektu oraz wymagania, które stawia przed nim Zleceniodawca. Osoba ta będzie uczestniczyła w pracach zespołu jako *Product Owner*. Pozostałych członków zespołu zapewni firma *Four Developers*.

2.2 Zarys technologiczny

Aplikacja tworzona w ramach projektu zostanie zrealizowana jako aplikacja z dostępem z poziomu przeglądarki internetowej. Dzięki temu, klienci firmy *Money Solutions* będą mieli ułatwiony dostęp do aplikacji po wykupieniu do niej dostępu. Równocześnie, zarządzanie użytkownikami oraz danymi będzie prostsze z racji na scentralizowanie danych na serwerze bazodanowym.

Ta konkretna aplikacja będzie realizowana w architekturze trójwarstwowej – ponieważ jest to najczęściej wykorzystywana architektura, kiery rozważamy aplikacje z dostępem poprzez przeglądarkę internetową. Wspomniana architektura jest przykładem architektury klient-serwer. Aplikacja jest podzielona na warstwy, co sprawia, że rozwój i ewentualne modyfikacje są łatwiejsze. Architektura ta składa się z trzech warstw: warstwy danych, warstwy aplikacyjnej (logiki biznesowej), oraz warstwy prezentacji. Opis każdej z warstw znajduje się poniżej:

- Warstwa prezentacji – jedyna widoczna dla użytkownika – zapewnia interfejs dla aplikacji. Ze względu na założenie projektowe – dostęp za pośrednictwem przeglądarki internetowej – stos technologiczny sprowadza się do HTML, CSS oraz JavaScript.
- Warstwa logiki biznesowej – poziom odpowiedzialny za komunikację między dwoma innymi warstwami – danych oraz prezentacji.
- Warstwa danych – używana do tworzenia zapytań, przechowywania i zarządzania danymi aplikacji. Składa się ona z systemu zarządzania bazą danych oraz biblioteki, która zapewnia dedykowane API dostępu do danych zawartych w bazie danych z poziomu kodu aplikacji.

2.3 Zespół

W skład zespołu będą wchodzili – oprócz wspomnianego wcześniej *Product Ownera* – analitycy oraz projektanci.

2.3.1 Analitycy – wymagania

Analitycy będą zajmować się prowadzeniem analizy biznesowej, badaniem potrzeb klientów, projektując rozwiązania dla systemu. Każdy z analityków jest odpowiedzialny za sprecyzowanie wymagań (zarówno funkcjonalnych jak i niefunkcjonalnych). Jego zadaniem jest również dbałość o ich prawidłową realizację. To głównie z analitykami będzie współpracował *Product Owner*.

Główne zadania:

- analiza środowiska systemowego,
- określenie wymagań stawianych przed systemem,
- tworzenie specyfikacji wymagań,
- tworzenie dokumentacji projektowej,
- tworzenie planu testów.

2.3.2 Projektanci – wymagania

Każdy z projektantów będzie odpowiedzialny za stworzenie architektury projektowanego systemu oraz udokumentowanie jej. Wyniki prac tego zespołu będą kluczowe dla prac programistów implementujących właściwy system.

Główne zadania:

- utworzenie projektu systemu,
- utworzenie dokumentacji technicznej projektu systemu,
- wybór technologii oraz metod realizacji systemu,
- dostosowywanie wymagań do postępów prac.

3 Wymagania

W tej sekcji znajduje się lista wymagań, jakie spełniać powinien budowany system. Podane są one z podziałem na dwie kategorie. Pierwsza to wymagania funkcjonalne – określające funkcjonalności systemu oraz sposoby ich użycia. Natomiast druga to wymagania niefunkcjonalne, które opisują ilościowe i jakościowe warunki działania systemu.

3.1 Aktorzy

W systemie rozróżniany będzie tylko jeden aktor – użytkownik systemu. Rozróżnianie typów użytkowników nie jest w tym przypadku zasadne, ponieważ mają oni dostęp do tych samych funkcjonalności i nie wyróżniają się żadnymi szczególnymi cechami. Z tego powodu nie istnieją również zależności pomiędzy aktorami.

3.2 Wymagania funkcjonalne

3.2.1 Użytkownicy

Wymagania funkcjonalne dotyczące zarządzaniem użytkownikami w systemie.

- logowanie do aplikacji,
- wyświetlanie listy użytkowników,
- dodawanie nowego użytkownika,
- edycja danych użytkownika,
- usunięcie użytkownika.

3.2.2 Konta

Wymagania funkcjonalne dotyczące zarządzaniem kontami pieniężnymi w systemie.

- wyświetlanie listy kont,
- dodawanie nowego konta,
- edycja danych konta,
- usunięcie konta.

3.2.3 Transakcje

Wymagania funkcjonalne dotyczące zarządzaniem transakcjami w systemie.

- wyświetlanie listy transakcji,
- dodawanie nowej transakcji,
- edycja danych transakcji,
- usunięcie transakcji.

3.2.4 Budżet

Wymagania funkcjonalne dotyczące zarządzaniem budżetami w systemie.

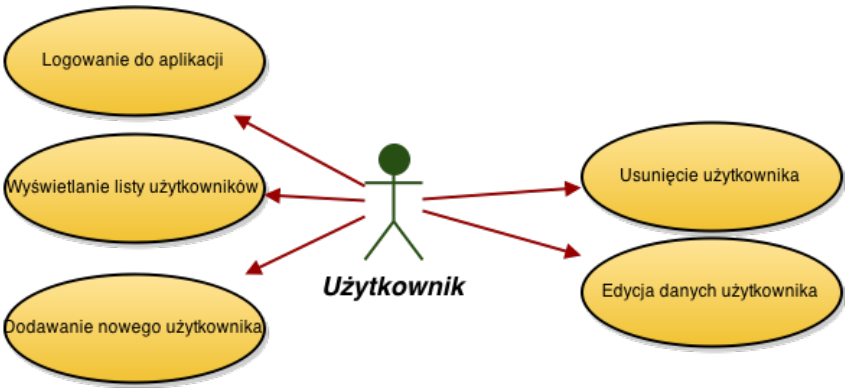
- wyświetlenie budżetów ustawionych w systemie,
- ustawienie budżetu dla określonego użytkownika,
- modyfikacja budżetu sumarycznego dla wszystkich użytkowników,
- modyfikacja budżetu określonego użytkownika,
- usunięcie budżetu określonego użytkownika.

3.2.5 Raporty

Wymagania funkcjonalne dotyczące generowania raportów w systemie.

- konfiguracja raportu,
- wyświetlanie raportu,
- zapisywanie raportu do pliku.

3.2.6 Opis przypadków użycia – użytkownicy



Rysunek 1: Diagram przypadków użycia związanych z kontami użytkowników.

3.2.6.1 Dodawanie nowego użytkownika

Opis słowny – niniejszy przypadek użycia umożliwia użytkownikowi stworzenie konta, poprzez które będzie miał on dostęp do systemu.

Aktor	Użytkownik
Warunki początkowe	Użytkownik nie jest zarejestrowany
Opis przebiegu interakcji	Wybór strony z rejestracją, wprowadzenie danych, kliknięcie przycisku, zatwierdzenie
Sytuacje wyjątkowe	Niepoprawnie wprowadzone dane
Warunki końcowe	Nowy użytkownik w systemie

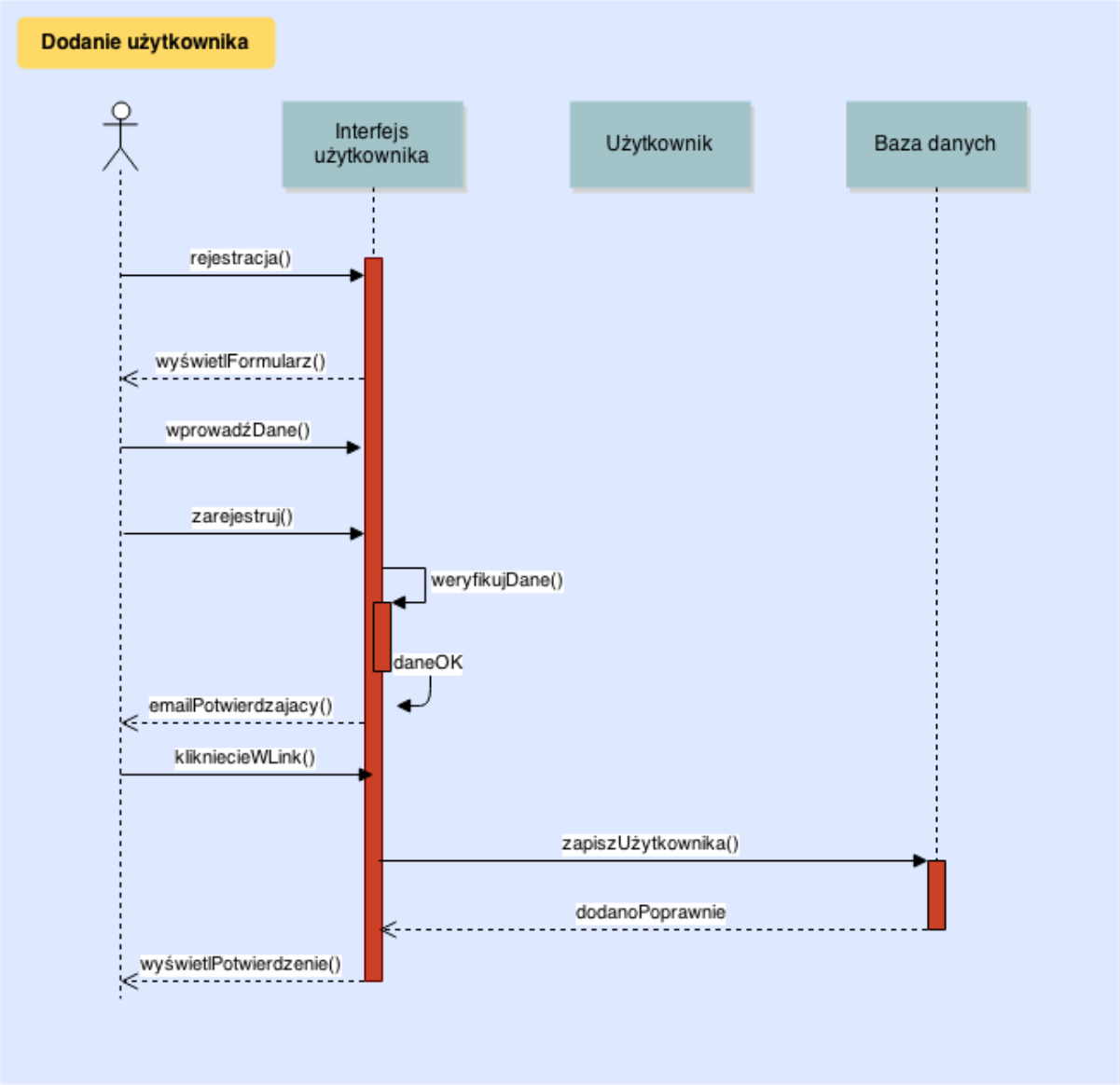
Scenariusz główny:

1. Użytkownik otwiera stronę „Logowanie/Rejestracja”.
2. Użytkownik wybiera opcję „Rejestracja” i wprowadza niezbędne dane.
3. Użytkownik naciska przycisk „Zarejestruj”.
4. Aplikacja sprawdza wprowadzone dane.
5. Jeżeli nie wystąpiły żadne błędy, system wysyła aktywacyjną wiadomość e-mail na adres wprowadzony przez użytkownika i informuje go, aby go odebrał.
6. Użytkownik otwiera wiadomość i klika w link aktywacyjny.
7. Aplikacja rejestruje nowego użytkownika i przekierowuje go do strony logowania.

Scenariusz alternatywny – niepoprawne dane:

1-4. Jak w scenariuszu głównym.

5. System wyświetla informacje o wykrytym błędzie.
6. Użytkownik poprawia wprowadzone dane i naciska przycisk „Zarejestruj”.
7. Powrót do kroku 4. ze scenariusza głównego.



Rysunek 2: Diagram sekwencji dla przypadku użycia 3.2.6.1 – Dodawanie nowego użytkownika (scenariusz główny)

3.2.6.2 Logowanie do aplikacji

Opis słowny – niniejszy przypadek użycia umożliwia użytkownikowi zalogowanie się do systemu, dzięki czemu będzie on miał dostęp do wszystkich funkcjonalności.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zarejestrowany
Opis przebiegu interakcji	Wybór strony z logowaniem, wprowadzenie danych, kliknięcie przycisku
Sytuacje wyjątkowe	Niepoprawnie wprowadzone dane
Warunki końcowe	Użytkownik jest zalogowany

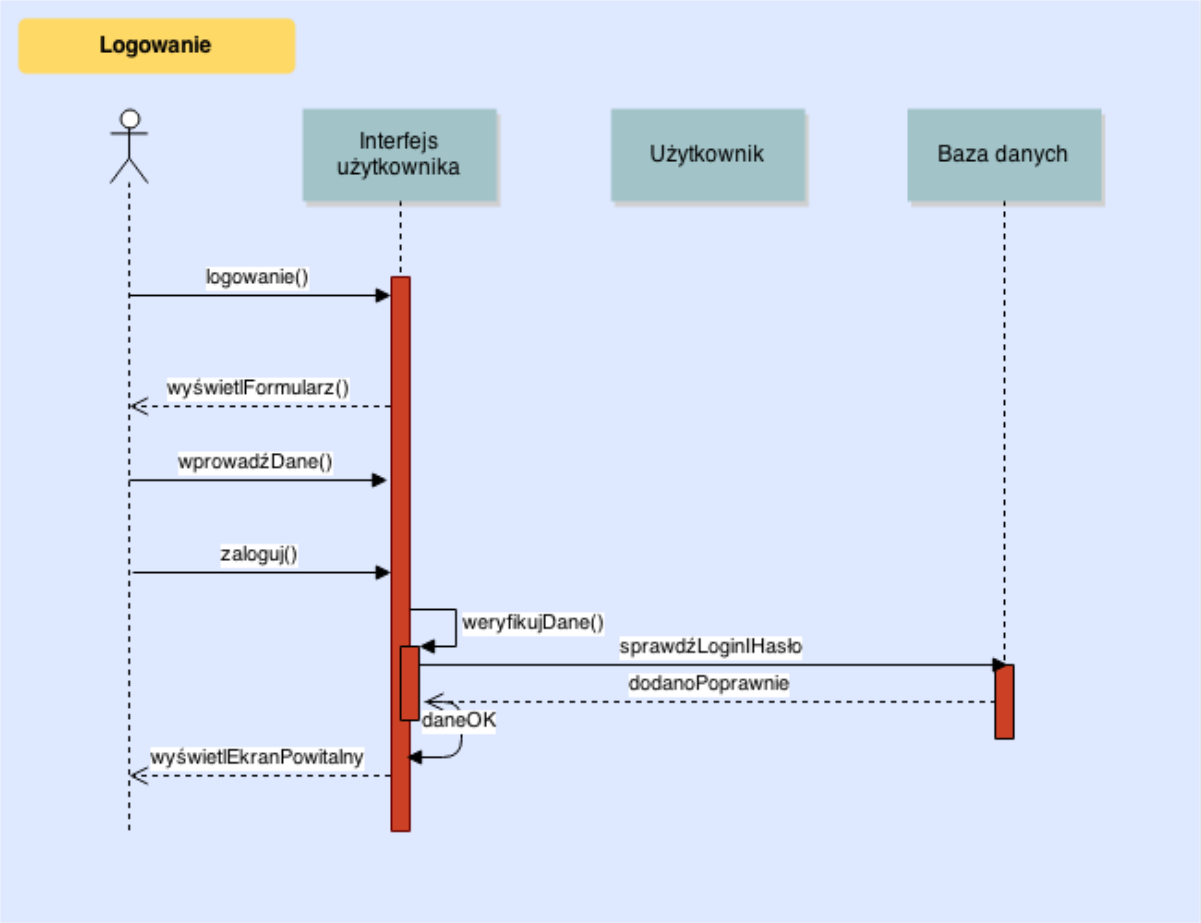
Scenariusz główny:

1. Użytkownik otwiera stronę „Logowanie/Rejestracja".
2. Użytkownik wybiera opcję „Logowanie" i wprowadza dane: adres e-mail oraz hasło.
3. Użytkownik naciska przycisk „Zaloguj".
4. Aplikacja sprawdza wprowadzone dane.
5. Jeżeli para (e-mail, hasło) znajduje się w systemie, aplikacja przekierowuje użytkownika na docelową stronę.

Scenariusz alternatywny – niepoprawne dane:

1-4. Jak w scenariuszu głównym.

5. System wyświetla informacje o wykrytym błędzie.
6. Użytkownik poprawia wprowadzone dane i naciska przycisk „Login".
7. Powrót do kroku 4. ze scenariusza głównego.



Rysunek 3: Diagram sekwencji dla przypadku użycia 3.2.6.2 – Logowanie do aplikacji (scenariusz główny)

3.2.6.3 Wyświetlanie listy użytkowników

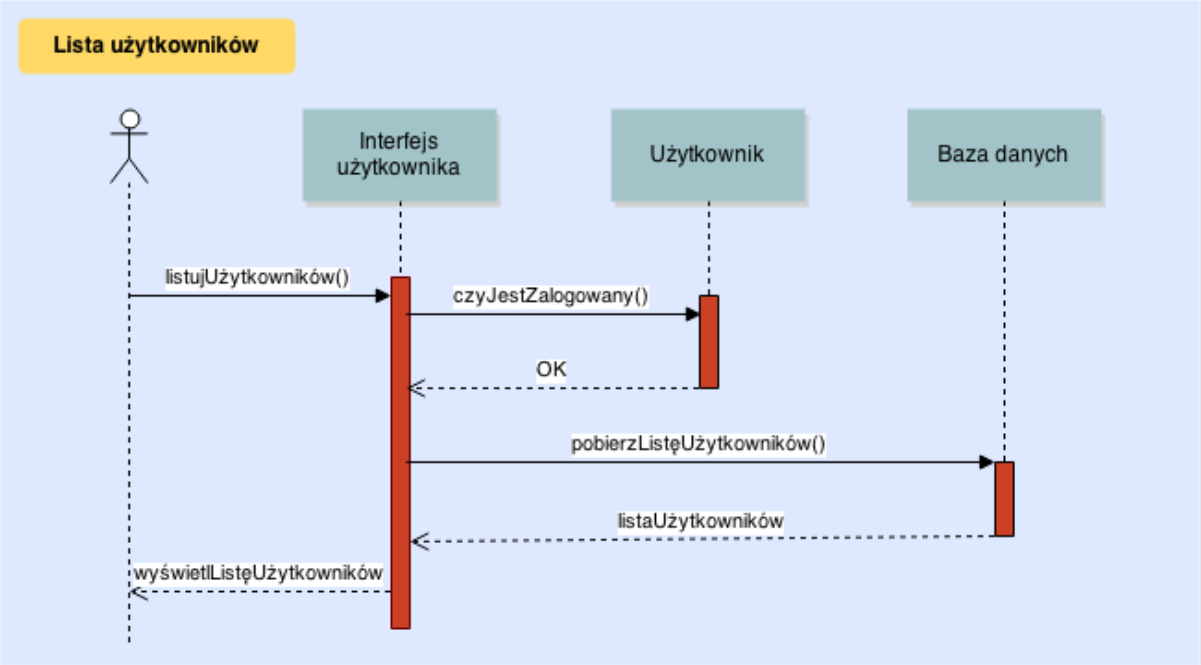
Korzysta z 3.2.6.2 – Logowanie do aplikacji.

Opis słowny – niniejszy przypadek użycia umożliwia użytkownikowi wyświetlenie listy użytkowników korzystających z systemu.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z użytkownikami
Sytuacje wyjątkowe	Użytkownik nie jest zalogowany
Warunki końcowe	Wyświetlona lista użytkowników

Scenariusz główny:

1. Użytkownik otwiera stronę „Użytkownicy”.
2. Jeśli użytkownik nie jest zalogowany, aplikacja wymaga zalogowania, inicjując 3.2.6.2 – Logowanie do aplikacji.
3. Aplikacja wyświetla użytkowników zarejestrowanych w systemie.



Rysunek 4: Diagram sekwencji dla przypadku użycia 3.2.6.3 – Wyświetlanie listy użytkowników (scenariusz główny)

3.2.6.4 Edycja danych użytkownika

Korzysta z 3.2.6.2 – Logowanie do aplikacji.

Opis słowny – niniejszy przypadek użycia umożliwia użytkownikowi edycję danych dotyczących jego profilu.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z profilem, wprowadzenie danych, kliknięcie przycisku, zatwierdzenie
Sytuacje wyjątkowe	Błędnie wprowadzone dane
Warunki końcowe	Dane użytkownika zaktualizowane w systemie

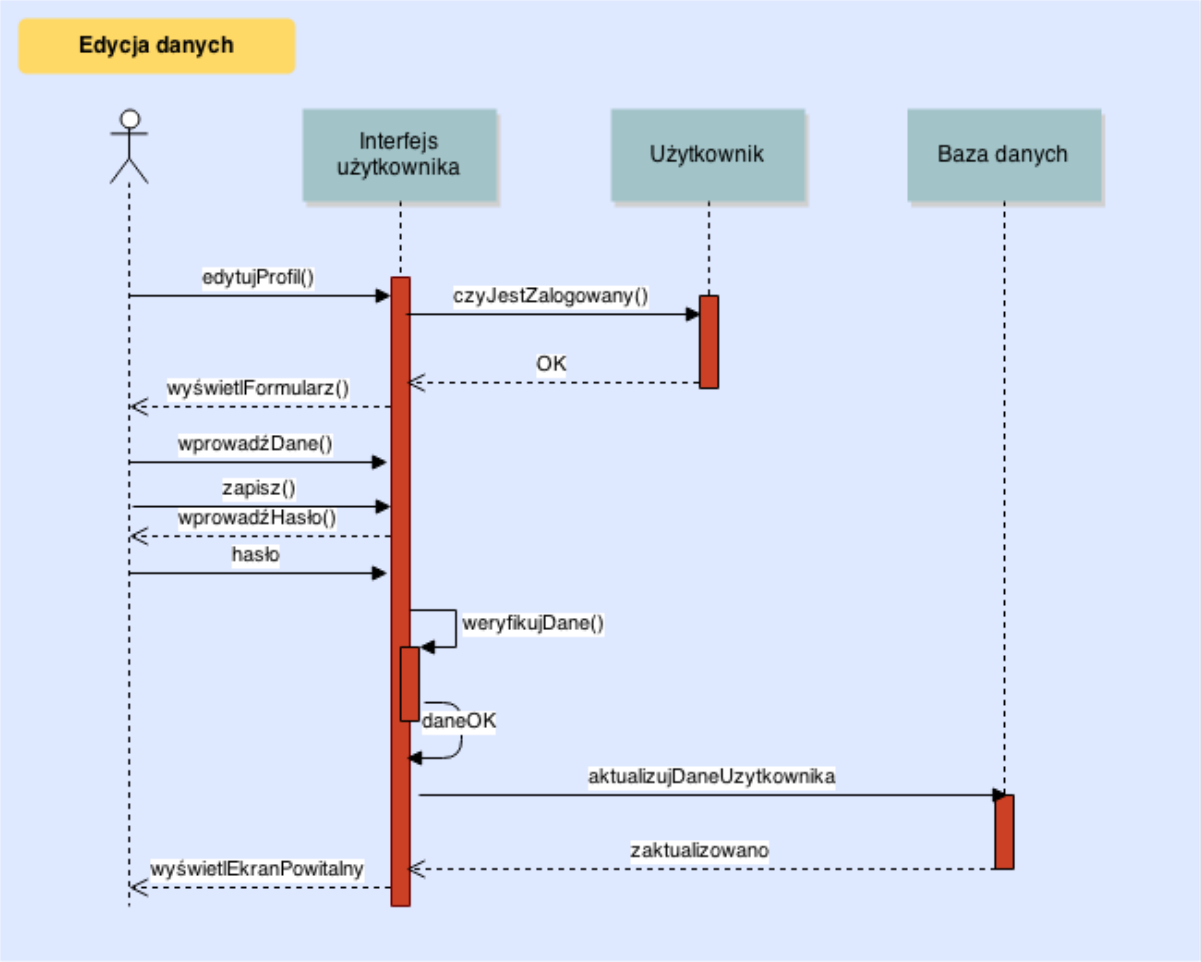
Scenariusz główny:

1. Użytkownik loguje się do aplikacji, tak jak w scenariuszu 3.2.6.2 – Logowanie do aplikacji.
2. Użytkownik otwiera stronę „Profil”.
3. Użytkownik naciska przycisk „Edytuj profil”.
4. Aplikacja wyświetla okno zawierające dane użytkownika.
5. Użytkownik edytuje imię, nazwisko, adres e-mail lub hasło i naciska przycisk „Zapisz”.
6. System prosi użytkownika o potwierdzenie zmiany poprzez wpisanie starego hasła.

7. System weryfikuje wprowadzone dane.
8. Jeżeli dane są prawidłowe, system edytuje dane użytkownika.

Scenariusz alternatywny – niepoprawne dane:

1-7. Jak w scenariuszu głównym.
8. System wyświetla informacje o wykrytym błędzie.
9. Użytkownik poprawia dane i naciska przycisk „Zapisz”.
10. Powrót do kroku 7. ze scenariusza głównego.



Rysunek 5: Diagram sekwencji dla przypadku użycia 3.2.6.4 – Edycja danych użytkownika (scenariusz główny)

3.2.6.5 Usunięcie użytkownika

Korzysta z 3.2.6.2 – Logowanie do aplikacji.

Opis słowny – niniejszy przypadek użycia umożliwia użytkownikowi skasowanie swojego profilu z systemu.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z profilem, wprowadzenie danych, kliknięcie przycisku
Sytuacje wyjątkowe	Błędnie wprowadzone dane
Warunki końcowe	Skasowanie użytkownika z systemu i wylogowanie

Scenariusz główny:

1. Użytkownik loguje się do aplikacji, tak jak w scenariuszu 3.2.6.2 – Logowanie do aplikacji.
2. Użytkownik otwiera stronę „Profil".
3. Użytkownik naciska przycisk „Usuń użytkownika".
4. System prosi użytkownika o potwierdzenie decyzji poprzez wpisanie hasła i naciśnięcie przycisku „Usuń".
5. System weryfikuje poprawność hasła.
6. Jeżeli hasło jest zgodne, aplikacja usuwa informacje o użytkowniku i następuje wylogowanie.

Scenariusz alternatywny – użytkownik nie potwierdza decyzji:

1-5. Jak w scenariuszu głównym.

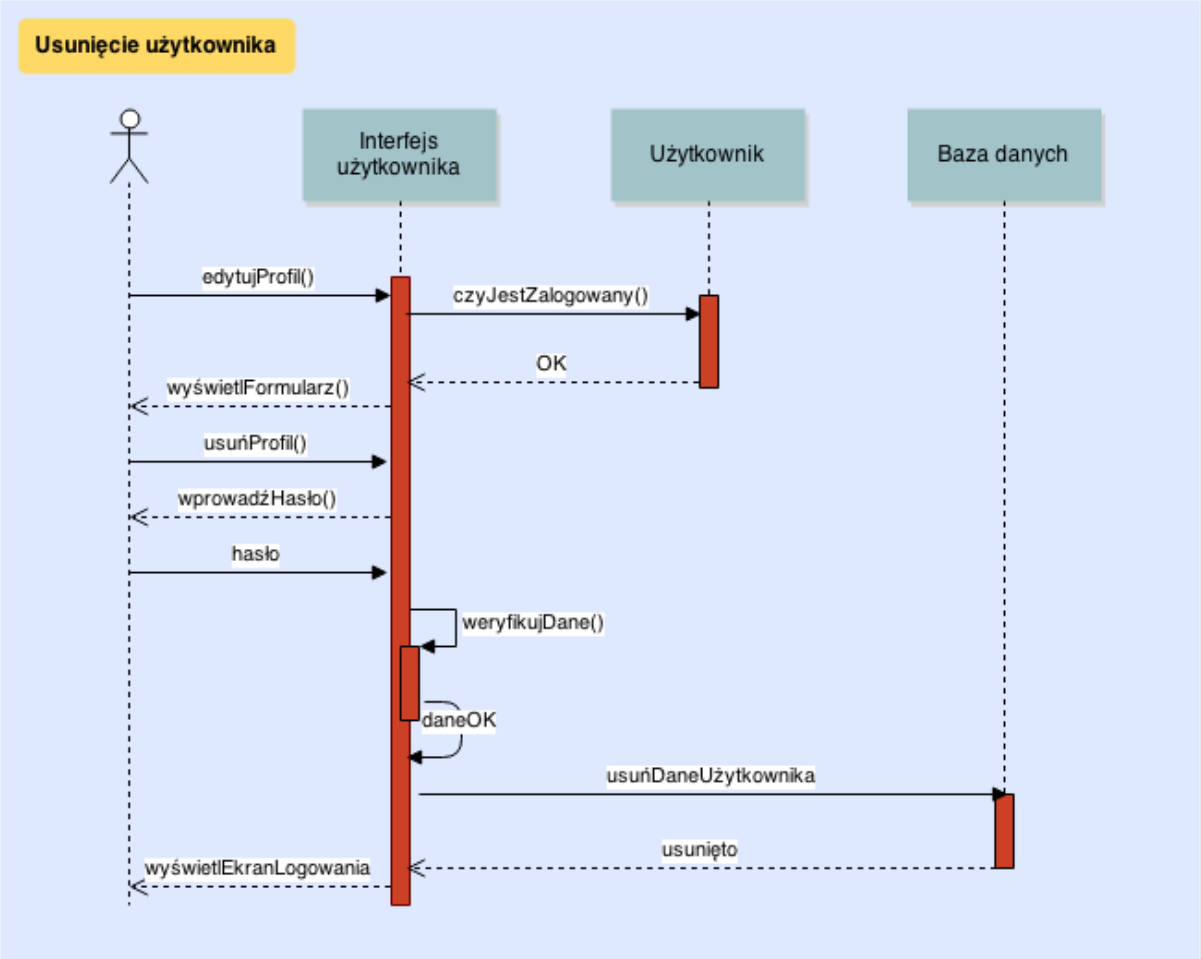
6. Użytkownik nacisnął przycisk „Anuluj".
7. System powraca do strony „Profil".

Scenariusz alternatywny – niepoprawne dane:

1-5. Jak w scenariuszu głównym.

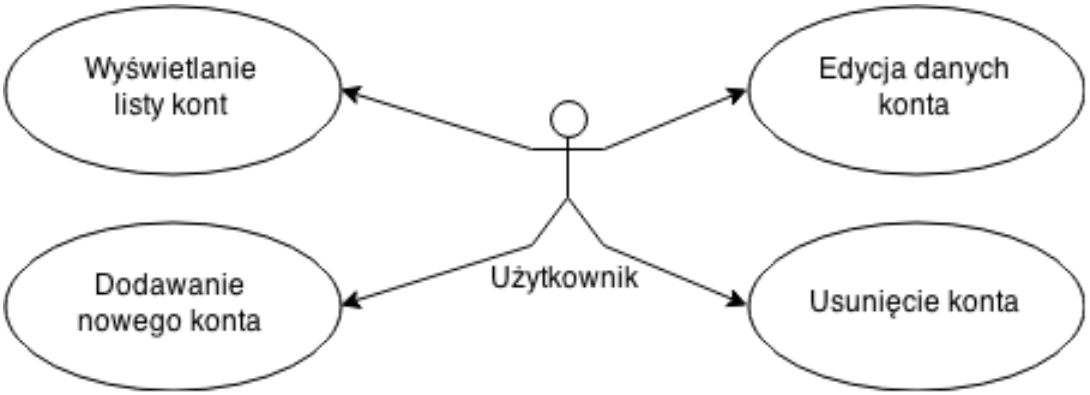
6. System wyświetla informację o wykrytym błędzie.
7. Użytkownik poprawia hasło i naciska przycisk „Usuń".

7. Powrót do kroku 5. ze scenariusza głównego.



Rysunek 6: Diagram sekwencji dla przypadku użycia 3.2.6.5 – Usunięcie użytkownika (scenariusz główny)

3.2.7 Opis przypadków użycia – konta



Rysunek 7: Diagram przypadków użycia związanych z zarządzaniem kontami.

3.2.7.1 Wyświetlanie listy kont

Korzysta z 3.2.6.2 – Logowanie do aplikacji.

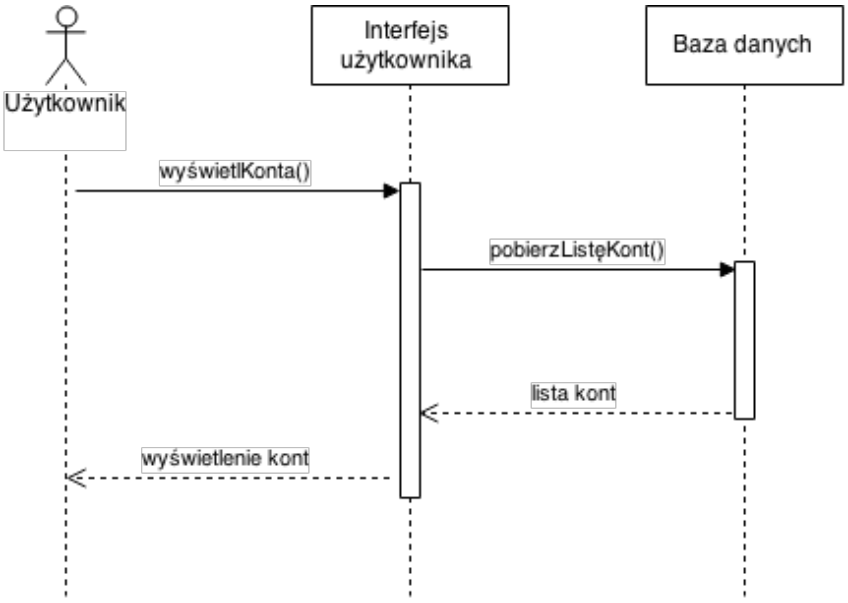
Funkcja generalizująca dla 3.2.7.2, 3.2.7.3 oraz 3.2.7.4.

Opis słowny – niniejszy przypadek użycia umożliwia użytkownikowi wyświetlenie listy kont, które są do niego przypisane w programie. Dzięki temu może on łatwo nimi zarządzać oraz monitorować ich stan.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z kontami
Sytuacje wyjątkowe	Brak
Warunki końcowe	Wyświetlona lista posiadanych kont

Scenariusz główny:

1. Użytkownik otwiera stronę „Konta”.
2. Jeśli użytkownik nie jest zalogowany, aplikacja wymaga zalogowania, inicjując 3.2.6.2 – Logowanie do aplikacji.
3. Aplikacja wyświetla konta należące do zalogowanego użytkownika.



Rysunek 8: Diagram sekwencji dla przypadku użycia 3.2.7.1 – Wyświetlenie listy kont

3.2.7.2 Utworzenie nowego konta

Funkcja specjalizująca dla 3.2.7.1 – Wyświetlanie listy kont.

Opis słowny – niniejszy przypadek użycia umożliwia użytkownikowi dodawanie nowych kont. Dzięki temu ma on możliwość kontrolowania stanu wszystkich posiadanych przez niego fizycznie kont – np. oszczędnościowych lub walutowych oraz definiowania transakcji pomiędzy nimi.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z kontami, kliknięcie przycisku, wprowadzenie danych oraz zatwierdzenie
Sytuacje wyjątkowe	Niepoprawne dane
Warunki końcowe	Nowe konto dodane do systemu

Scenariusz główny:

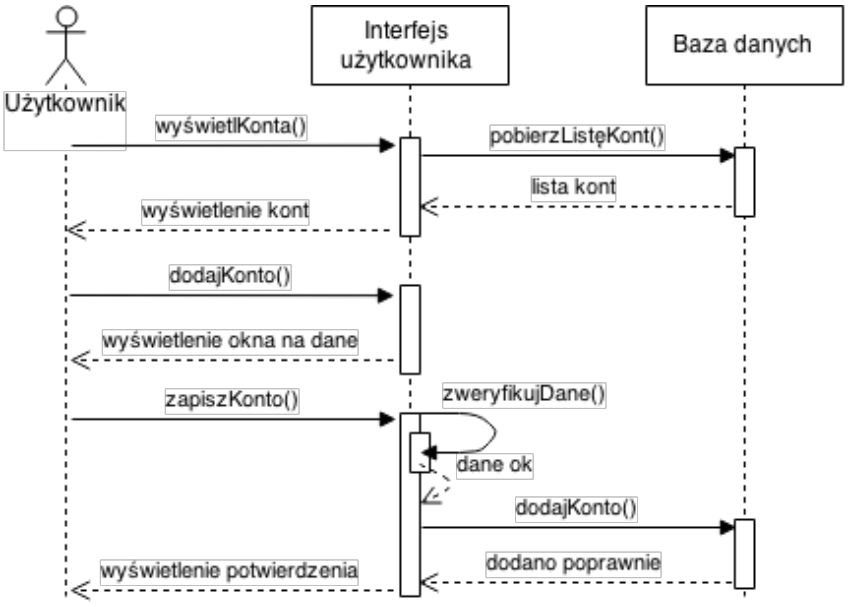
1-3. Jak w funkcji generalizującej 3.2.7.1 – Wyświetlanie listy kont.

4. Użytkownik klika przycisk „Dodaj nowe konto”.
5. Aplikacja wyświetla okno zawierające pola z danymi dla nowego konta.
6. Użytkownik wprowadza nazwę konta, początkowe saldo, wybiera walutę, typ konta i klika przycisk „Dodaj”.
7. System weryfikuje wprowadzone dane (np. czy para użytkownik-nazwa jest unikatowa).
8. Jeśli dane są prawidłowe, system tworzy nowe konto i powiadamia o tym użytkownika.

Scenariusz alternatywny – niepoprawne dane:

1-7. Jak w scenariuszu głównym.

8. System wyświetla informację o wykrytym błędzie.
9. Użytkownik poprawia dane i wciska przycisk „Dodaj”.
10. Powrót do kroku 7 ze scenariusza głównego.



Rysunek 9: Diagram sekwencji dla przypadku użycia 3.2.7.2 – Utworzenie nowego konta

3.2.7.3 Edycja danych konta

Funkcja specjalizująca dla 3.2.7.1 – Wyświetlanie listy kont.

Opis słowny – niniejszy przypadek użycia umożliwia użytkownikowi edycję danych konta. Dzięki temu ma on możliwość aktualizowania danych oraz zapewnienia ich zgodności ze stanem faktycznym – poza systemem.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany i posiada przynajmniej jedno konto
Opis przebiegu interakcji	Wybór strony z kontami, wybór konta, kliknięcie przycisku, wprowadzenie danych oraz zatwierdzenie
Sytuacje wyjątkowe	Niepoprawne dane
Warunki końcowe	Edycja danych konta w systemie

Scenariusz główny:

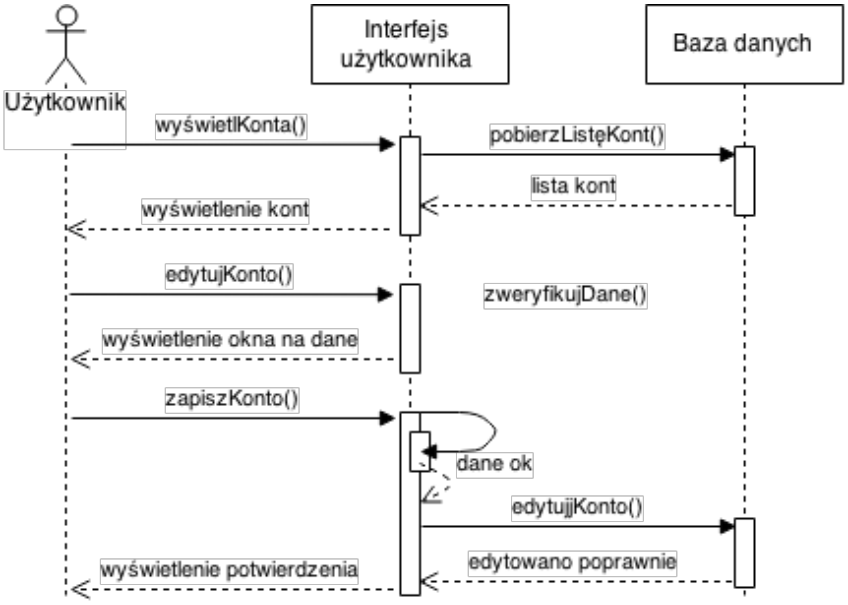
1-3. Jak w funkcji generalizującej 3.2.7.1 – Wyświetlanie listy kont.

4. Użytkownik zaznacza jedno z kont i klika przycisk „Edytuj” obok niego.
5. Aplikacja wyświetla okno zawierające dane zaznaczonego konta.
6. Użytkownik edytuje nazwę konta, walutę lub typ konta i klika przycisk „Zapisz”.
7. System weryfikuje wprowadzone dane (np. czy para użytkownik-nazwa jest unikatowa).
8. Jeśli dane są prawidłowe, system edytuje konto i powiadamia o tym użytkownika.

Scenariusz alternatywny – niepoprawne dane:

1-7. Jak w scenariuszu głównym.

8. System wyświetla informację o wykrytym błędzie.
9. Użytkownik poprawia dane i wciska przycisk „Zapisz”.
10. Powrót do kroku 7 ze scenariusza głównego.



Rysunek 10: Diagram sekwencji dla przypadku użycia 3.2.7.3 – Edycja danych konta

3.2.7.4 Usunięcie konta

Funkcja specjalizująca dla 3.2.7.1 – Wyświetlanie listy kont.

Opis słowny – niniejszy przypadek użycia umożliwia użytkownikowi usunięcie konta. Dzięki temu ma on możliwość ukrycia w systemie kont już nieużywanych lub nieaktywnych.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany i posiada przynajmniej jedno konto
Opis przebiegu interakcji	Wybór strony z kontami, wybór konta, kliknięcie przycisku oraz zatwierdzenie
Sytuacje wyjątkowe	Anulowanie akcji
Warunki końcowe	Usunięcie konta z systemu

Scenariusz główny:

1-3. Jak w funkcji generalizującej 3.2.7.1 – Wyświetlanie listy kont.

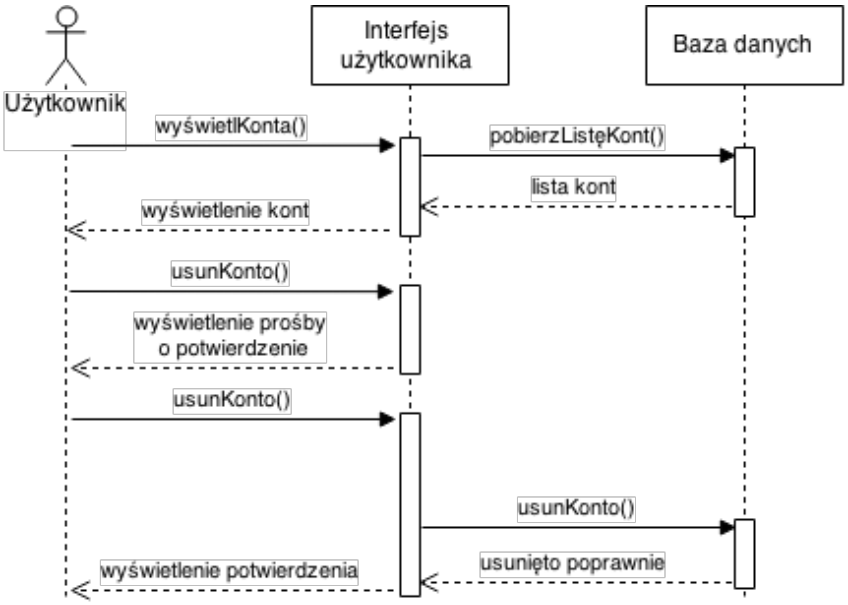
4. Użytkownik zaznacza jedno z kont i klika przycisk „Usuń” obok niego.
5. Aplikacja wyświetla okno potwierdzenia.
6. Użytkownik klika przycisk „Usuń” w wyświetlonym oknie.
7. Aplikacja zamyka okno, usuwa konto i odświeża listę kont.

Scenariusz alternatywny – przerwanie operacji przez użytkownika:

1-5. Jak w scenariuszu głównym.

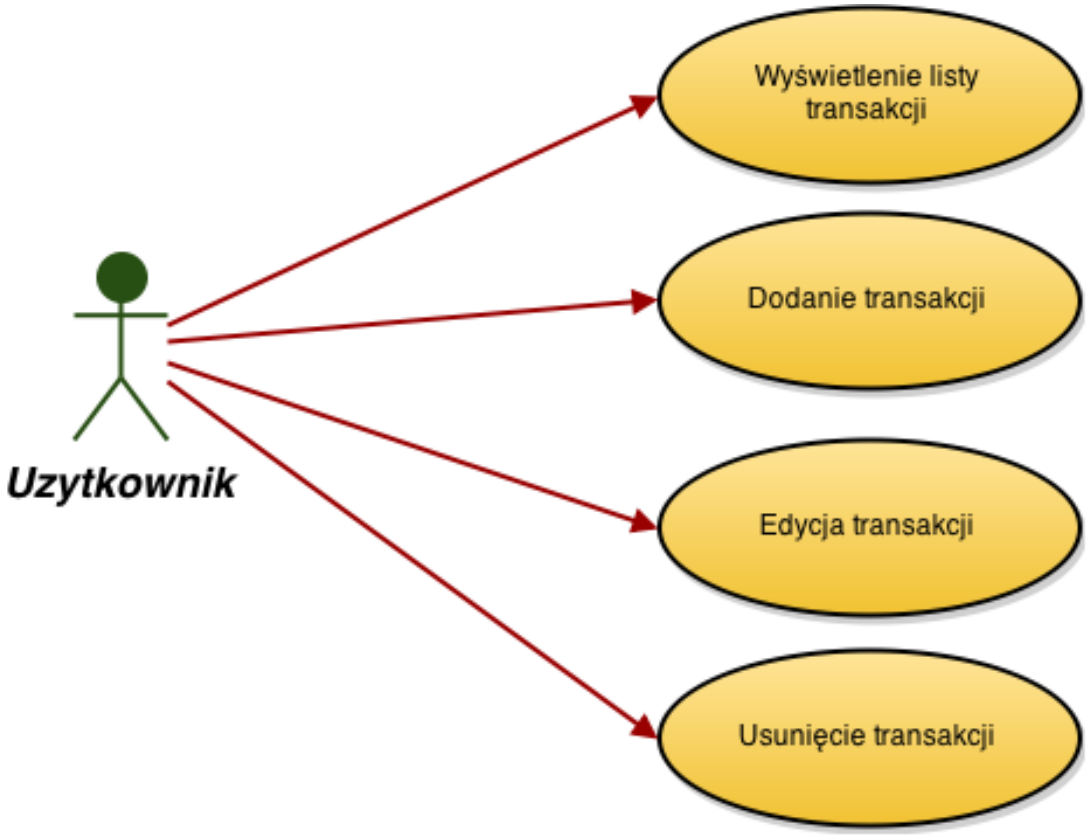
6. Użytkownik wciska przycisk „Anuluj”.

7. Aplikacja zamyka okno.



Rysunek 11: Diagram sekwencji dla przypadku użycia 3.2.7.4 – Usunięcie konta

3.2.8 Opis przypadków użycia – transakcje



Rysunek 12: Diagram przypadków użycia związanych z zarządzaniem transakcjami.

3.2.8.1 Wyświetlanie listy transakcji

Wykorzystuje 3.2.6.2 – Logowanie do aplikacji.
Funkcja generalizująca dla 3.2.8.2, 3.2.8.3 oraz 3.2.8.4.

Opis słowny – użytkownik będzie chciał wyświetlić listę transakcji zarejestrowanych w systemie. Oprócz zwykłej listy zebranych danych, potrzebna jest możliwość filtrowania, co ułatwi użytkownikowi przeglądanie listy.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z transakcjami, wprowadzenie parametrów filtrowania
Sytuacje wyjątkowe	Żądanie filtrowania danych
Warunki końcowe	Wyświetlona lista żądanych transakcji

Scenariusz główny:

1. Użytkownik otwiera stronę „Transakcje”.
2. Jeśli użytkownik nie jest zalogowany, aplikacja wymaga zalogowania, inicjując 3.2.6.2 – Logowanie do aplikacji.

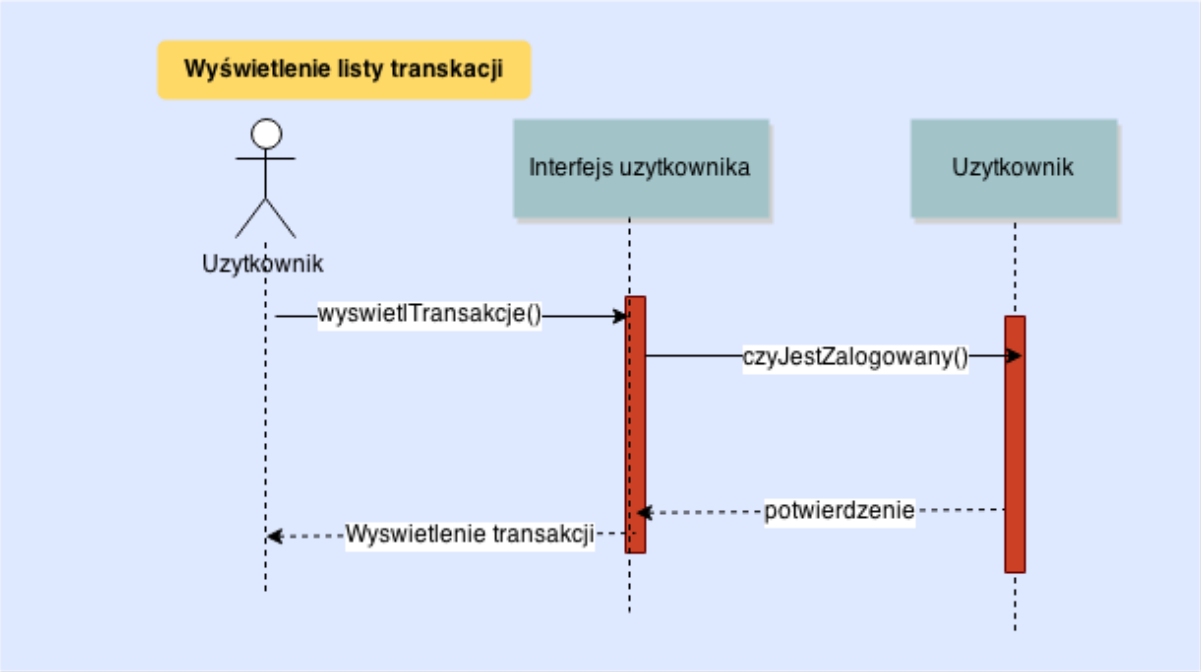
3. Aplikacja wyświetla listę transakcji powiązanych z kontem danego użytkownika.

Scenariusz alternatywny – filtrowanie transakcji:

1-3. Jak w scenariuszu głównym.

4. Użytkownik wprowadza parametry filtrowania jak nazwa użytkownika, kategoria albo data transakcji.

5. Aplikacja odświeża listę transakcji, wyświetlając tylko te, które pasują do podanych filtrów.



Rysunek 13: Diagram sekwencji dla przypadku użycia 3.2.8.1 – Wyświetlenie listy transakcji (scenariusz główny)

3.2.8.2 Utworzenie nowej transakcji

Funkcja specjalizująca dla 3.2.8.1 – Wyświetlanie listy transakcji.

Opis słowny – jedna z podstawowych funkcjonalności systemu to dodawanie transakcji, których koszty będą sumowane i nadzorowane. Akcje te mają miejsce, gdy użytkownik chce dodać wydatek do systemu.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany oraz posiada przynajmniej jedno konto
Opis przebiegu interakcji	Wybór strony z transakcjami i wciśnięcie dodaj, uzupełnienie danych i zapisanie
Sytuacje wyjątkowe	Wprowadzenie niepoprawnych danych
Warunki końcowe	Dodanie nowej transakcji

Scenariusz główny:

1-3. Jak w scenariuszu generalizującym 3.2.8.1 – Wyświetlenie listy transakcji.

4. Użytkownik wciska przycisk „Dodaj transakcję".

5. Aplikacja wyświetla okno z polami do wprowadzania danych dotyczących transakcji.

6. Użytkownik wprowadza rodzaj transakcji (wpływ/wydatek/transfer), konto od/do, wprowadza wartość, datę, opis i klika przycisk „Dodaj".

7. System weryfikuje wprowadzone dane (np. czy na danym koncie znajduje się wystarczająca ilość pieniędzy).

8. Jeśli dane są poprawne, system tworzy nową transakcję, przelicza ilość pieniędzy pozostałych na koncie i informuje użytkownika.

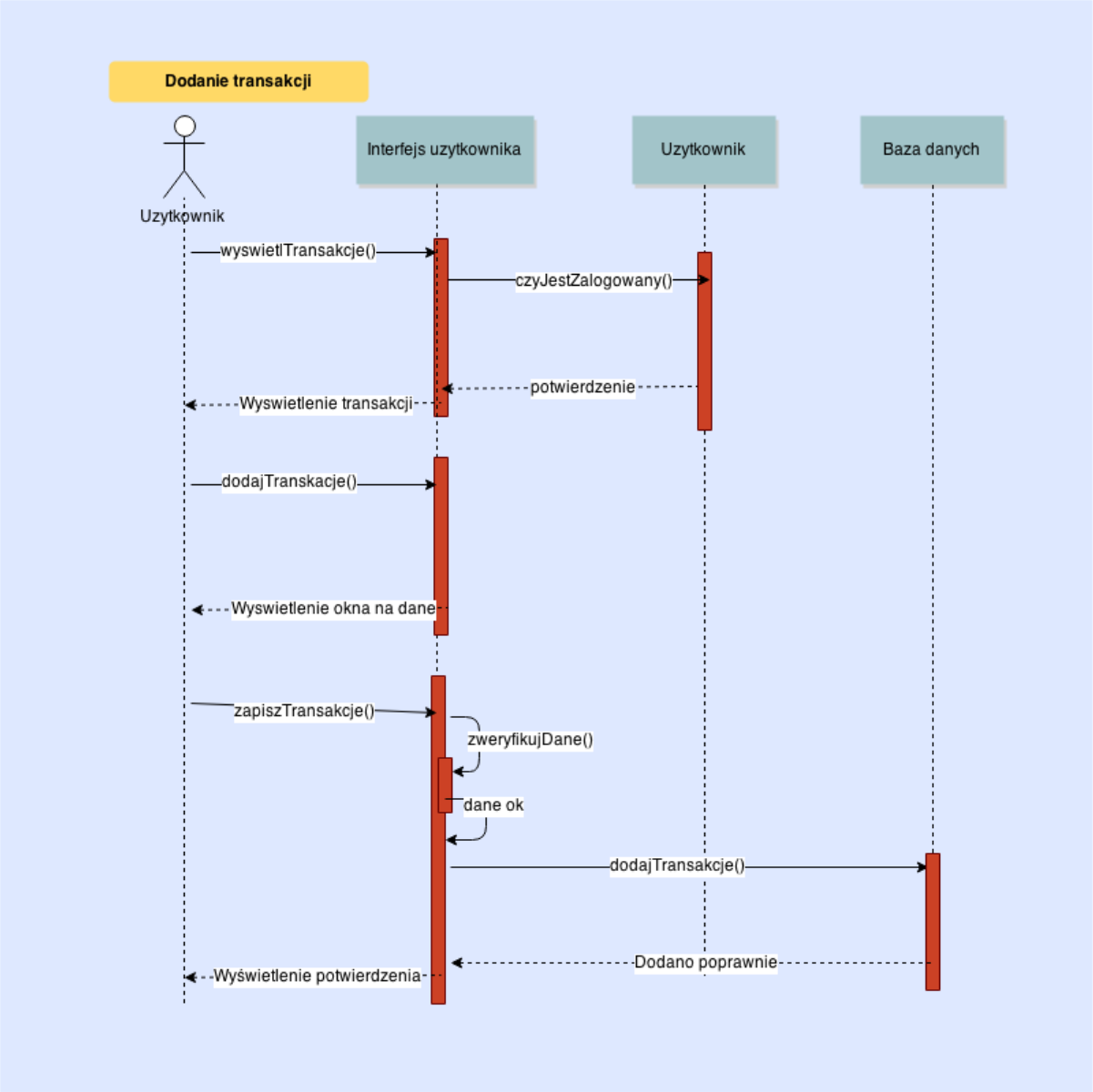
Scenariusz alternatywny – niepoprawne dane:

1-7. Jak w scenariuszu głównym.

8. System wyświetla informację o wykrytym błędzie.

9. Użytkownik poprawia dane i wciska przycisk „Dodaj".

10. Powrót do kroku 7 ze scenariusza głównego.



Rysunek 14: Diagram sekwencji dla przypadku użycia 3.2.8.2 – Utworzenie nowej transakcji (scenariusz główny)

3.2.8.3 Edycja transakcji

Funkcja specjalizująca dla 3.2.8.1 – Wyświetlanie listy transakcji.

Opis słowny – modyfikacja danych dotyczących wybranej transakcji może być przydatna w sytuacji, gdy użytkownik wprowadzi błędne dane w kontekście logicznym, ale które będą poprawnie w sensie formalnym.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany oraz istnieją transakcje w systemie

Opis przebiegu interakcji	Wybór strony z transakcjami, wybór żądanej transakcji, modyfikacja danych i zapisanie
Sytuacje wyjątkowe	Wprowadzenie niepoprawnych danych
Warunki końcowe	Dodanie nowej transakcji

Scenariusz główny:

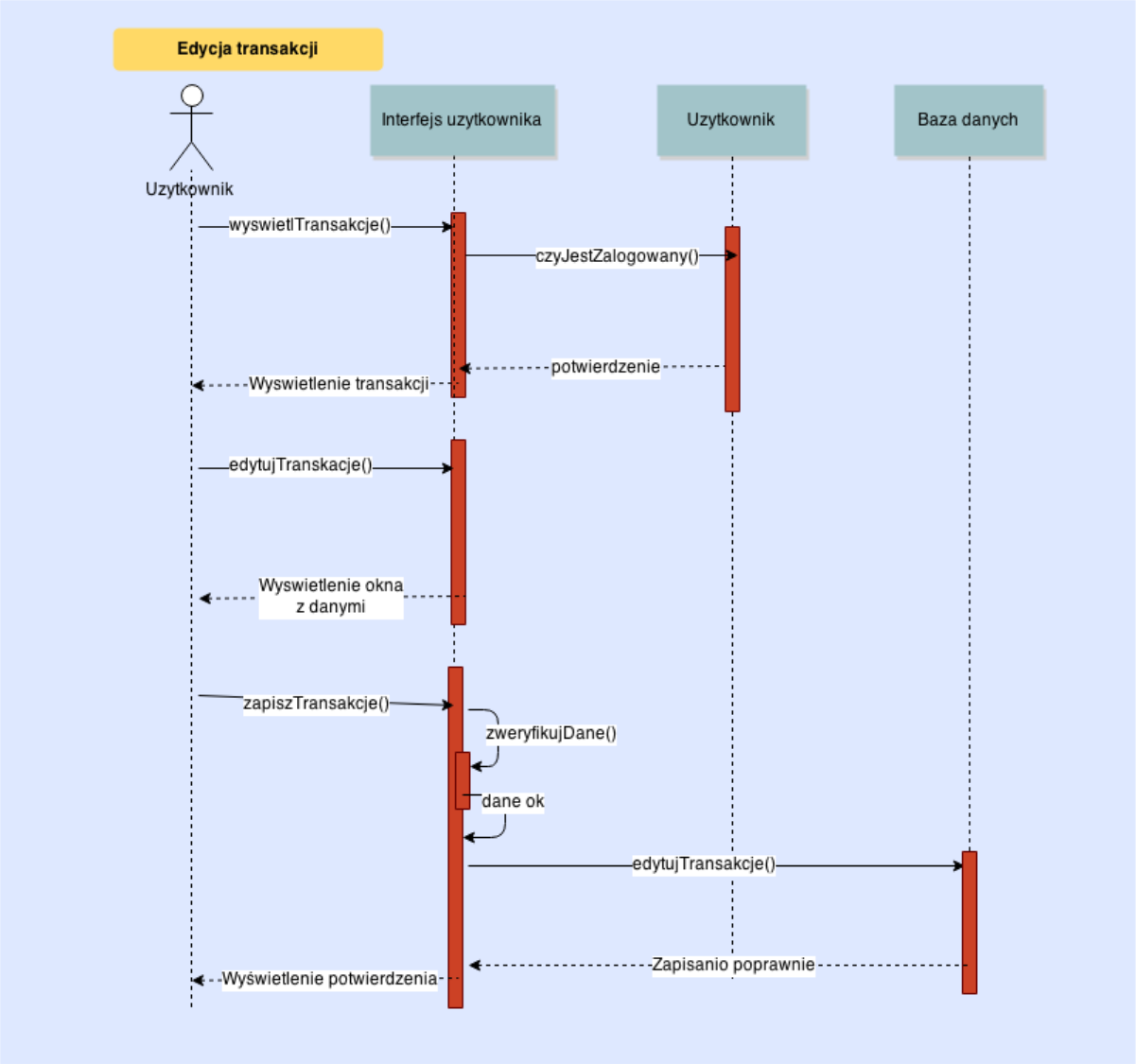
1-3. Jak w scenariuszu generalizującym 3.2.8.1 – Wyświetlenie listy transakcji.

4. Użytkownik wybiera jedną z transakcji i wciska przycisk „Edytuj".
5. Aplikacja wyświetla okno z informacjami o transakcji.
6. Użytkownik modyfikuje dane transakcji (wpływ/wydatek/transfer), konto od/do, wprowadza wartość, datę, opis i klika przycisk „Zapisz".
7. System weryfikuje wprowadzone dane (np. czy na danym koncie znajduje się wystarczająca ilość pieniędzy).
8. Jeśli dane są poprawne, system zapisuje transakcję, przelicza ilość pieniędzy pozostałych na koncie i informuje użytkownika.

Scenariusz alternatywny – niepoprawne dane:

1-7. Jak w scenariuszu głównym.

8. System wyświetla informację o wykrytym błędzie i czeka na poprawkę.
9. Użytkownik poprawia dane i wciska przycisk „Zapisz".
10. Powrót do kroku 7 ze scenariusza głównego.



Rysunek 15: Diagram sekwencji dla przypadku użycia 3.2.8.3 – Edycja transakcji (scenariusz główny)

3.2.8.4 Usuwanie transakcji

Funkcja specjalizująca dla 3.2.8.1 – Wyświetlanie listy transakcji.

Opis słowny – akcje opisane poniżej mają miejsce w przypadku gdy np. użytkownik zwrócił towar którego dotyczyła transakcja, którą już wprowadził.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany oraz istnieją transakcje w systemie
Opis przebiegu interakcji	Wybór strony z transakcjami, wybór żądanej transakcji, usunięcie jej
Sytuacje wyjątkowe	Brak
Warunki końcowe	Usunięcie transakcji

Scenariusz główny:

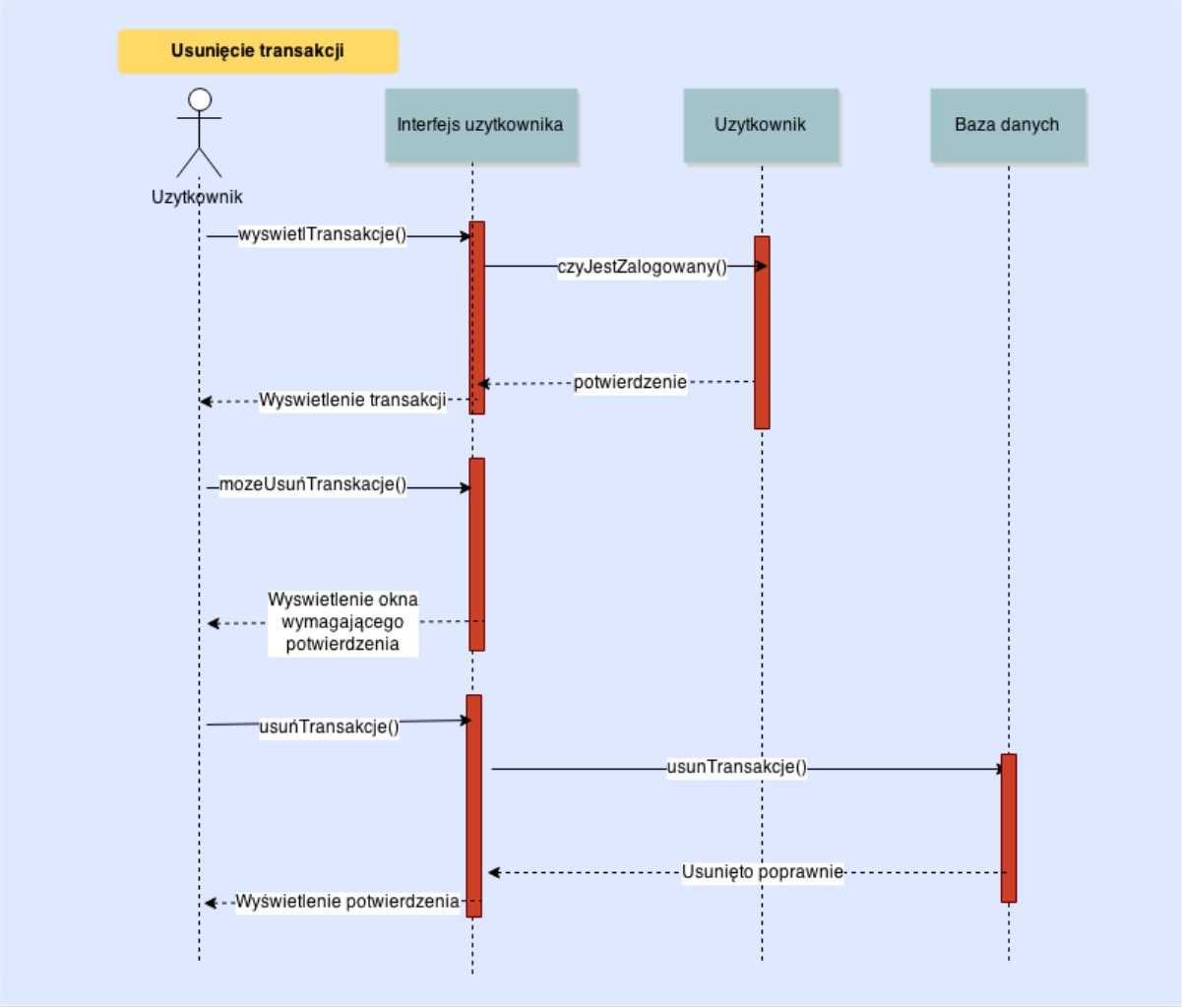
1-3. Jak w scenariuszu generalizującym 3.2.8.1 – Wyświetlenie listy transakcji.

4. Użytkownik wybiera jedną z transakcji i wciska przycisk „Usuń”.
5. Aplikacja wyświetla okienko wymagające potwierdzenia operacji.
6. Użytkownik potwierdza operację wciskając przycisk „Usuń”.
7. Aplikacja zamyka okienko, usuwa transakcję, przelicza ilość pieniędzy na koncie i odświeża listę transakcji.

Scenariusz alternatywny – przerwanie operacji przez użytkownika:

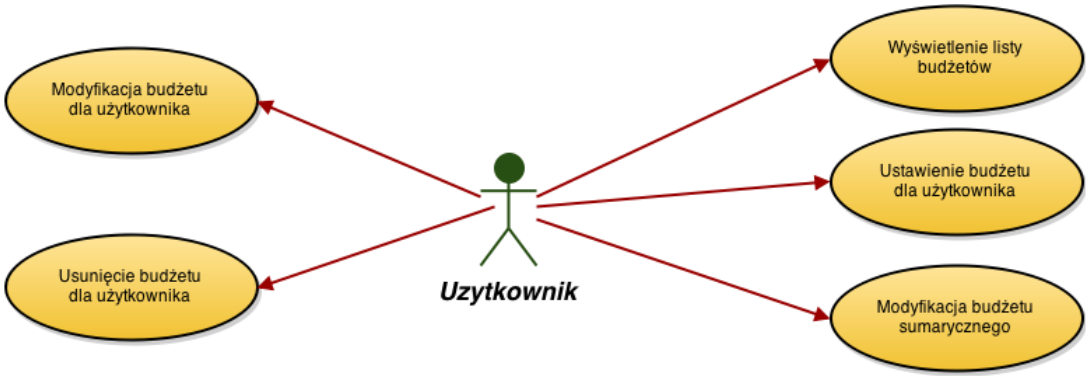
1-5. Jak w scenariuszu głównym.

6. Użytkownik wciska przycisk „Anuluj”.
7. Aplikacja zamyka okno.



Rysunek 16: Diagram sekwencji dla przypadku użycia 3.2.8.4 – Usuwanie transakcji (scenariusz główny)

3.2.9 Opis przypadków użycia – budżet



Rysunek 17: Diagram przypadków użycia związanych z zarządzaniem budżetami.

3.2.9.1 Wyświetlenie budżetów ustawionych w systemie

Wykorzystuje 3.2.6.2 – Logowanie do aplikacji.
Funkcja generalizująca dla 3.2.9.2, 3.2.9.3, 3.2.9.4 oraz 3.2.9.5.

Opis słowny – głównym zadaniem systemu jest nadzorowanie domowego budżetu. Użytkownik będzie chciał przejrzeć aktualny stan budżetów sumarycznych oraz poszczególnych dla każdego z zarejestrowanych użytkowników.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z budżetami
Sytuacje wyjątkowe	Żądanie filtrowania danych
Warunki końcowe	Wyświetlenie żądanych budżetów

Scenariusz główny:

1. Użytkownik otwiera stronę „Budżety”.
2. Jeśli użytkownik nie jest zalogowany, aplikacja wymaga zalogowania, inicjując 3.2.6.2 – Logowanie do aplikacji.
3. Aplikacja wyświetla budżet sumaryczny dla wszystkich użytkowników oraz listę użytkowników z budżetami dla każdego z nich.

Scenariusz alternatywny – filtrowanie wyników:

1-3. Jak w scenariuszu głównym.

4. Użytkownik wprowadza nazwę konta, którego budżet ma zostać wyświetlony i wciska „Enter”.
5. Aplikacja odświeża listę budżetów wyświetlając tylko jeden dla podanego użytkownika.

3.2.9.2 Ustawienie budżetu dla określonego użytkownika

Funkcja specjalizująca dla 3.2.9.1 – Wyświetlanie budżetów.

Opis słowny – celem lepszej kontroli wydatków, użytkownik może chcieć ustawić limit budżetu dla określonego użytkownika, aby ten nie mógł przekroczyć określonej kwoty.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z budżetami, wprowadzenie wymaganych danych, zapisanie nowego budżetu
Sytuacje wyjątkowe	Błędne wprowadzenie danych, przerwanie operacji
Warunki końcowe	Dodanie nowego limitu dla użytkownika

Scenariusz główny:

1-3. Jak w scenariuszu generalizującym 3.2.9.1 – Wyświetlenie budżetów występujących w systemie.

4. Użytkownik wciska przycisk „Dodaj".
5. Aplikacja wyświetla okno z możliwością wyboru konta i wprowadzenia nowego budżetu.
6. Użytkownik wybiera konto, wprowadza nowy budżet i wciska przycisk "Dodaj".
7. Aplikacja weryfikuje poprawność wprowadzonych danych (np. czy liczba nie jest ujemna).
8. Jeśli wartość jest poprawna system zapisuje nową wartość budżetu, zamyka okno i aktualizuje stronę z budżetami.

Scenariusz alternatywny – przerwanie operacji przez użytkownika:

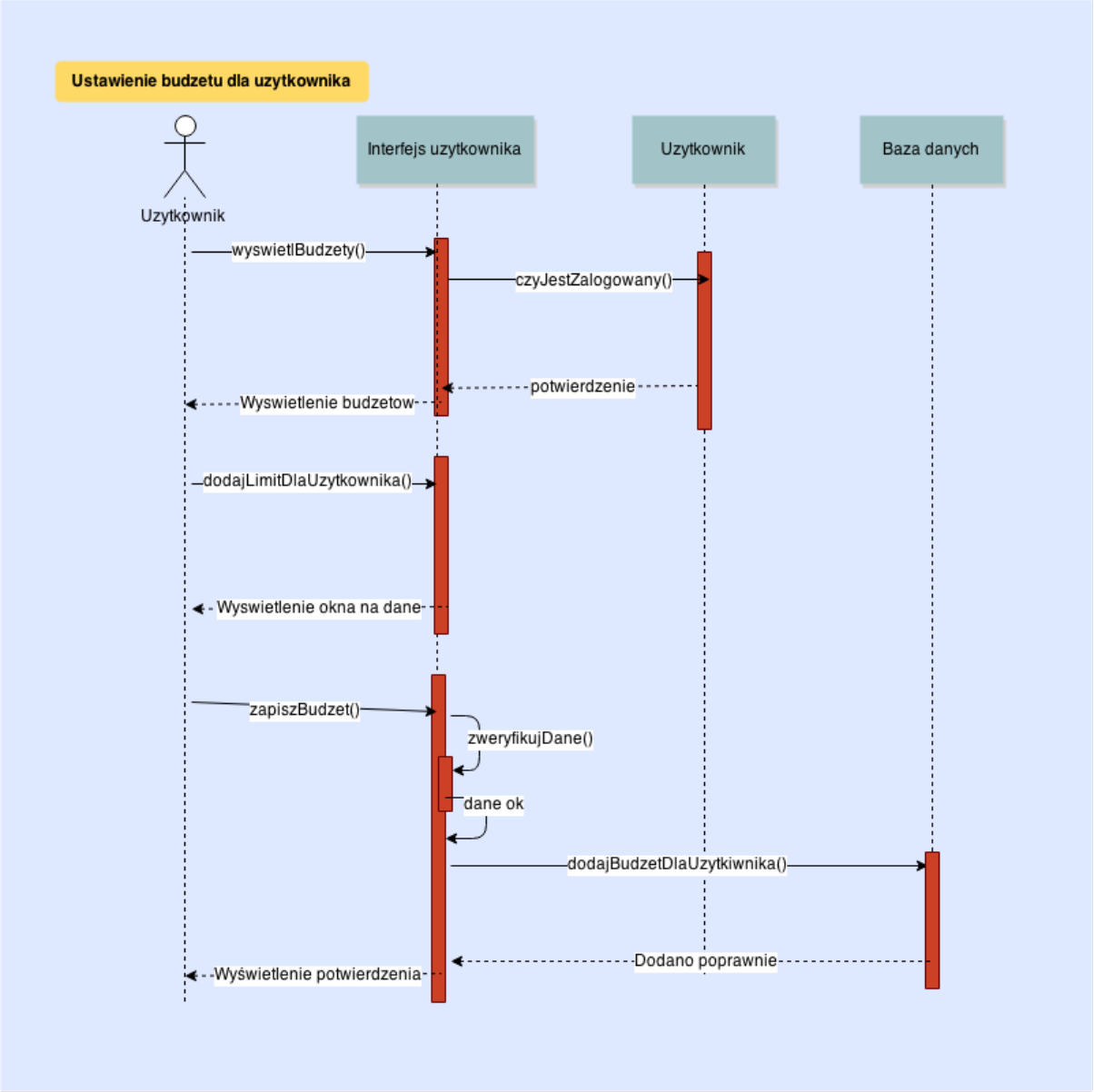
1-5. Jak w scenariuszu głównym.

6. Użytkownik wciska przycisk "Anuluj".
7. Aplikacja zamyka okno.

Scenariusz alternatywny – niepoprawne dane:

1-7. Jak w scenariuszu głównym.

8. System wyświetla informację o niepoprawnych danych.
9. Użytkownik wprowadza nową wartość i wciska przycisk „Dodaj".
10. Powrót do kroku 7 ze scenariusza głównego.



Rysunek 18: Diagram sekwencji dla przypadku użycia 3.2.9.2 – Ustawienie budżetu dla określonego użytkownika (scenariusz główny)

3.2.9.3 Modyfikacja budżetu sumarycznego dla wszystkich użytkowników

Funkcja specjalizująca dla 3.2.9.1 – Wyświetlanie budżetów.

Opis słowny – oprócz ustawiania limitu miesięcznego dla danego użytkownika, kluczowym elementem systemu jest modyfikacja limitu sumarycznego dla wszystkich użytkowników danej aplikacji. Wartość ta musi być zawsze ustawiona, lecz użytkownik może ją dowolnie modyfikować.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z budżetami, wprowadzenie wymaganych danych, zapisanie budżetu

Sytuacje wyjątkowe	Błędne wprowadzenie danych, przerwanie operacji
Warunki końcowe	Zmodyfikowanie budżetu sumarycznego

Scenariusz główny:

1-3. Jak w scenariuszu generalizującym 3.2.9.1 – Wyświetlenie budżetów występujących w systemie.

4. Użytkownik wciska przycisk "Edytuj" obok pola wyświetlającego budżet sumaryczny.
5. Aplikacja wyświetla okno z możliwością wprowadzenia nowego budżetu sumarycznego.
6. Użytkownik wprowadza nowy budżet i wciska przycisk „Zapisz".
7. Aplikacja weryfikuje poprawność wprowadzonych danych (np. czy liczba nie jest ujemna).
8. Jeśli wartość jest poprawna, system zapisuje nową wartość budżetu, zamyka okno i aktualizuje stronę z budżetami.

Scenariusz alternatywny – przerwanie operacji przez użytkownika:

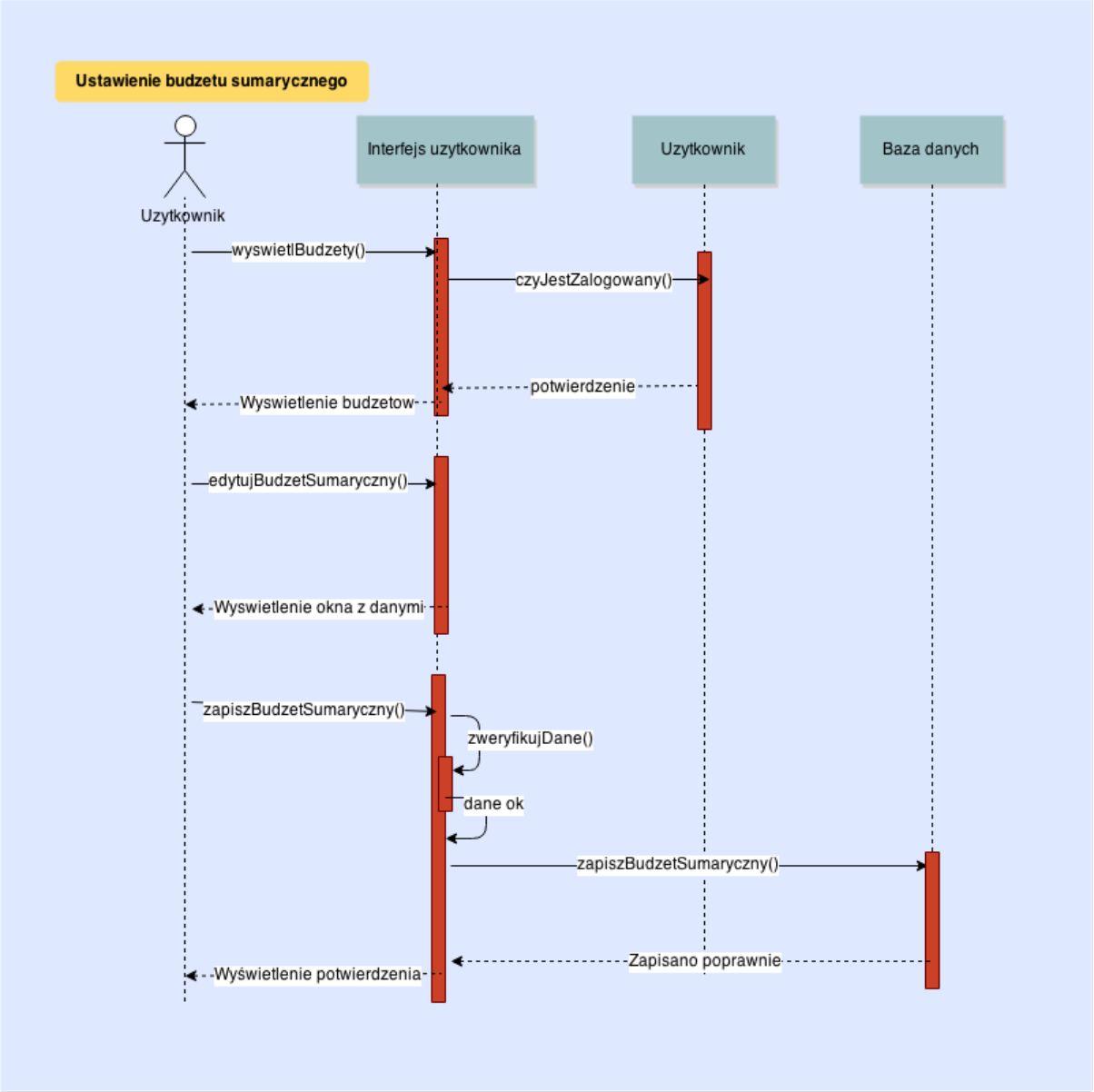
1-5. Jak w scenariuszu głównym.

6. Użytkownik wciska przycisk "Anuluj".
7. Aplikacja zamyka okno.

Scenariusz alternatywny – niepoprawne dane:

1-7. Jak w scenariuszu głównym.

8. System wyświetla informację o niepoprawnych danych.
9. Użytkownik wprowadza nową wartość i wciska przycisk „Zapisz".
10. Powrót do kroku 7 ze scenariusza głównego.



Rysunek 19: Diagram sekwencji dla przypadku użycia 3.2.9.3 – Ustawienie budżetu sumarycznego (scenariusz główny)

3.2.9.4 Modyfikacja budżetu dla określonego użytkownika

Funkcja specjalizująca dla 3.2.9.1 – Wyświetlanie budżetów.

Opis słowny – system musi pozwolić również modyfikować wcześniej ustawiony budżet dla poszczególnych użytkowników, gdyż limity te mogą się zmieniać np. gdy dziecko dostanie większe kieszonkowe.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór strony z budżetami, wprowadzenie wymaganych danych, zapisanie budżetu

Sytuacje wyjątkowe	Błędne wprowadzenie danych, przerwanie operacji
Warunki końcowe	Zmodyfikowanie budżetu użytkownika

Scenariusz główny:

1-3. Jak w scenariuszu generalizującym 3.2.9.1 – Wyświetlenie budżetów występujących w systemie.

4. Użytkownik wciska przycisk „Edytuj” obok pola wyświetlającego budżet dla określonego użytkownika.
5. Aplikacja wyświetla okno z możliwością wprowadzenia nowego budżetu.
6. Użytkownik wprowadza nowy budżet i wciska przycisk „Zapisz”.
7. Aplikacja weryfikuje poprawność wprowadzonych danych (np. czy liczba nie jest ujemna).
8. Jeśli wartość jest poprawna, system zapisuje nową wartość budżetu, zamyka okno i aktualizuje stronę z budżetami.

Scenariusz alternatywny – przerwanie operacji przez użytkownika:

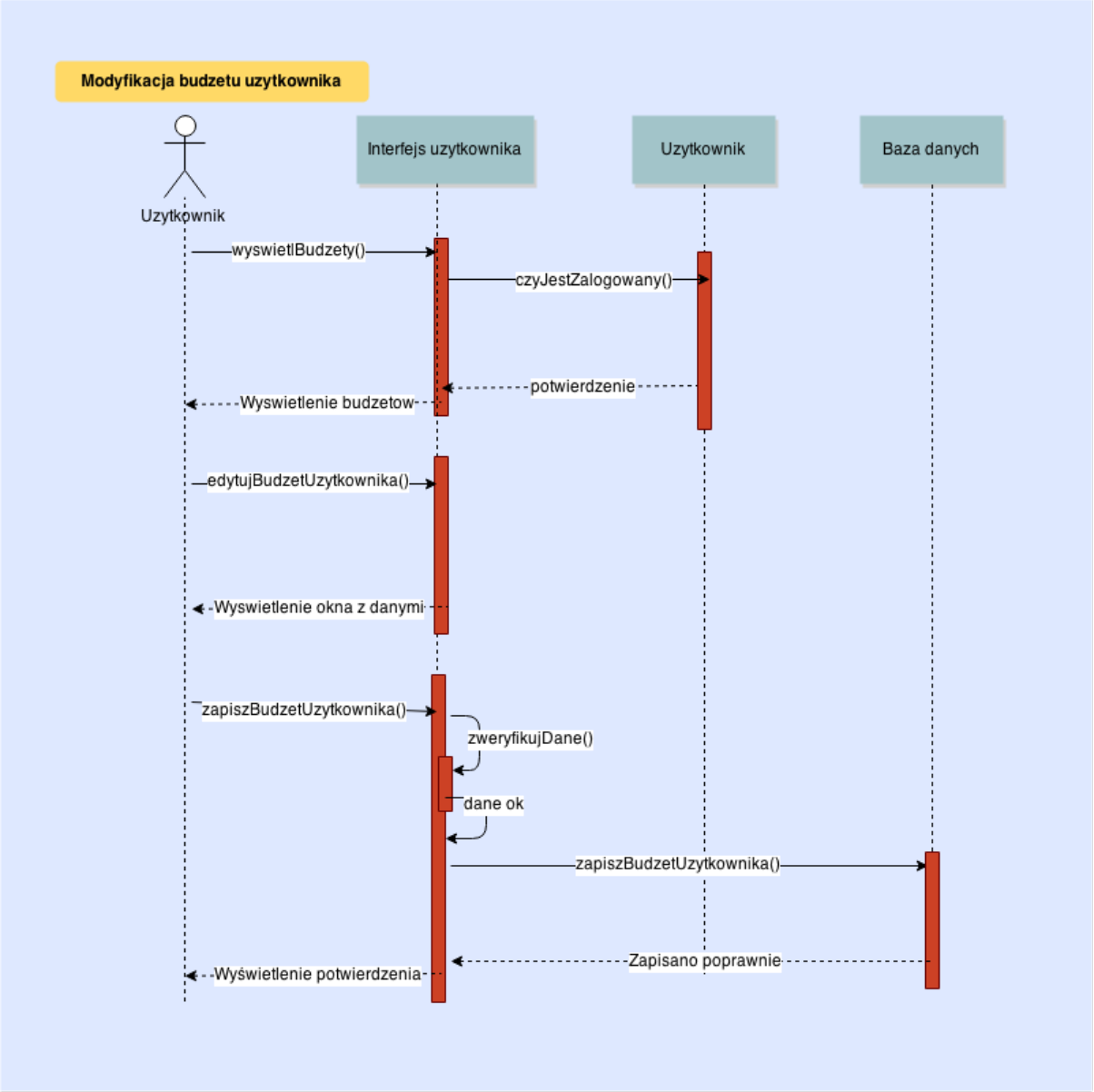
1-5. Jak w scenariuszu głównym.

6. Użytkownik wciska przycisk „Anuluj”.
7. Aplikacja zamyka okno.

Scenariusz alternatywny – niepoprawne dane:

1-7. Jak w scenariuszu głównym.

8. System wyświetla informację o niepoprawnych danych.
9. Użytkownik wprowadza nową wartość i wciska przycisk „Zapisz”.
10. Powrót do kroku 7 ze scenariusza głównego.



Rysunek 20: Diagram sekwencji dla przypadku użycia 3.2.9.4 – Modyfikacja budżetu dla określonego użytkownika (scenariusz główny)

3.2.9.5 Usunięcie budżetu dla określonego użytkownika

Funkcja specjalizująca dla 3.2.9.1 – Wyświetlanie budżetów.

Opis słowny – akcje te będą wykonane, gdy np. użytkownik nie będzie dłużej korzystał z podanej aplikacji lub limit miesięczny przestał go dotyczyć. Poza tym, polityka ustalania budżetów może ulec zmianie – na przykład zamiast ustawiania budżetów na poszczególne osoby, zacznie obowiązywać tylko budżet sumaryczny.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany oraz istnieje limit ustawiony na użytkownika

Opis przebiegu interakcji	Wybór strony z budżetami, usunięcie wybranego budżetu
Sytuacje wyjątkowe	Przerwanie operacji
Warunki końcowe	Usunięcie budżetu użytkownika

Scenariusz główny:

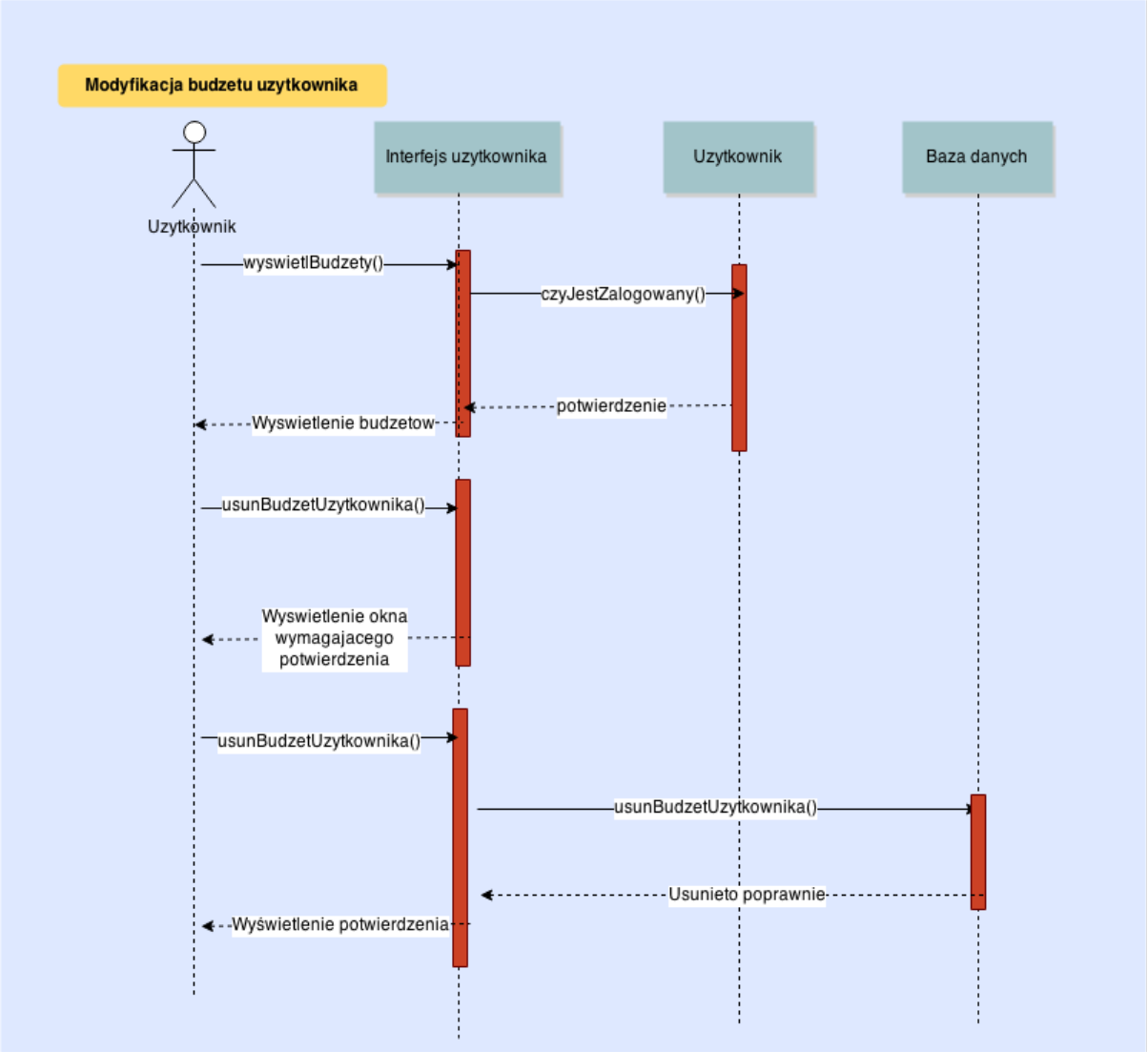
1-3. Jak w scenariuszu generalizującym 3.2.9.1 – Wyświetlenie budżetów występujących w systemie.

4. Użytkownik wciska przycisk „Usuń” obok pola wyświetlającego budżet dla określonego użytkownika.
5. Aplikacja wyświetla okno wymagające potwierdzenia operacji.
6. Użytkownik potwierdza operację wciskając przycisk „Usuń”.
7. System usuwa budżet dla określonego użytkownika, zamyka okno i odświeża stronę z budżetami.

Scenariusz alternatywny – przerwanie operacji przez użytkownika:

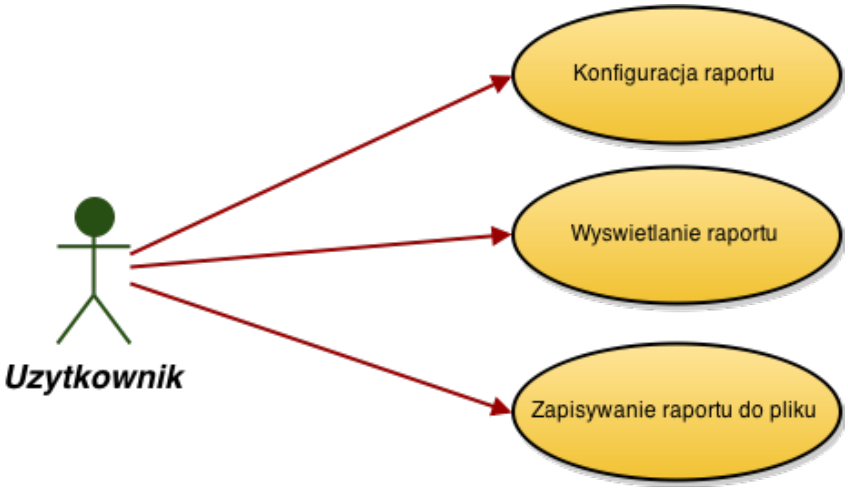
1-5. Jak w scenariuszu głównym.

6. Użytkownik klika przycisk „Anuluj”.
7. Aplikacja zamyka okno.



Rysunek 21: Diagram sekwencji dla przypadku użycia 3.2.9.5 – Usunięcie budżetu dla określonego użytkownika (scenariusz główny)

3.2.10 Opis przypadków użycia – raporty



Rysunek 22: Diagram przypadków użycia związanych z raportami.

3.2.10.1 Konfiguracja raportu

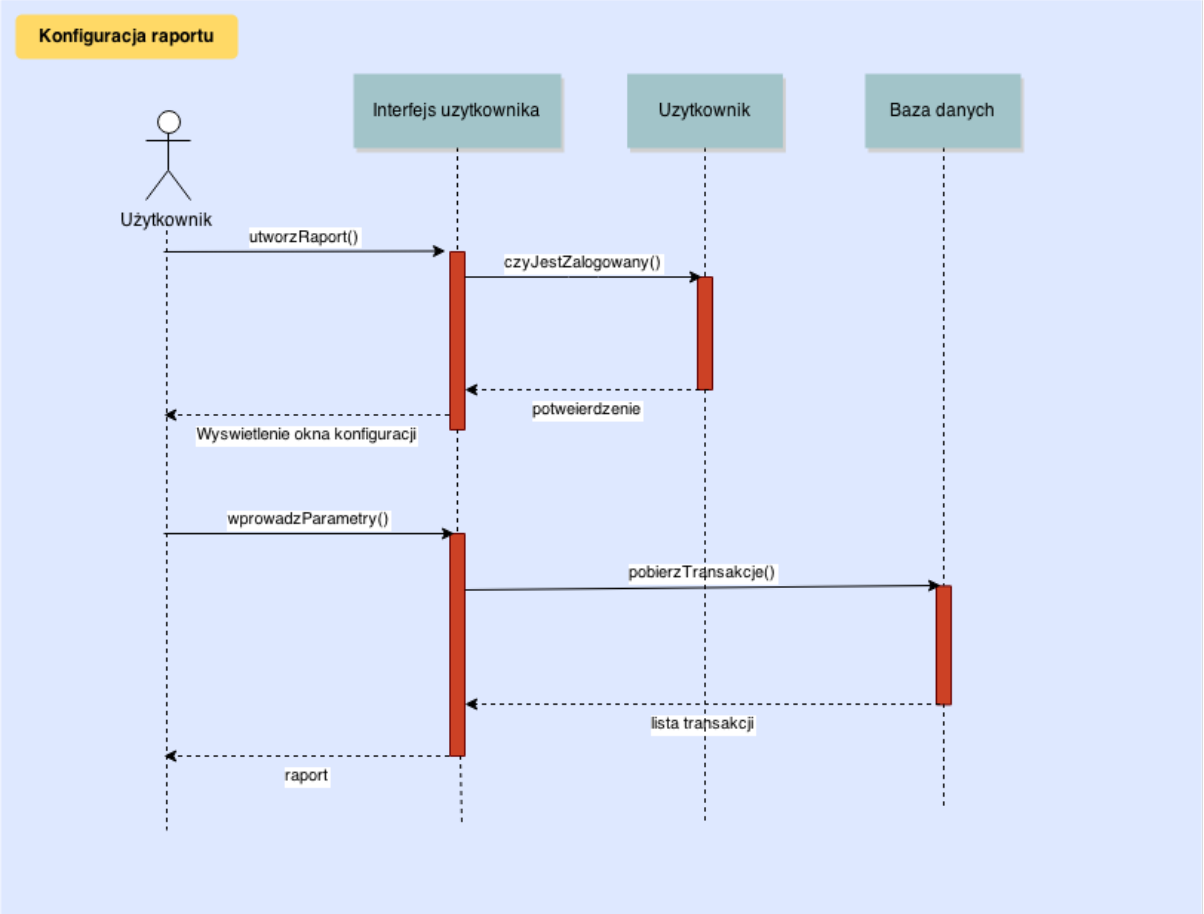
Wykorzystuje 3.2.6.2 – Logowanie do aplikacji.
Funkcja generalizująca dla 3.2.10.2 oraz 3.2.10.3.

Opis słowny – istotną funkcją systemu jest generowanie raportów wizualizujących dane zebrane w bazie. Użytkownik powinien mieć możliwość określenia budżetów i zakresu transakcji, których raport ma dotyczyć.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany
Opis przebiegu interakcji	Wybór opcji raportu
Sytuacje wyjątkowe	Brak
Warunki końcowe	Brak

Scenariusz główny:

1. Użytkownik otwiera stronę „Raport”.
2. Jeśli użytkownik nie jest zalogowany, aplikacja wymaga zalogowania, inicjując 3.2.6.2 – Logowanie do aplikacji.
3. Aplikacja wyświetla pola do wprowadzania opcji raportu
4. Użytkownika wprowadza parametry raportu, takie jak zakres uwzględnionych transakcji (budżet sumaryczny/budżet indywidualny/wszystkie transakcje użytkownika), zakres dat, rodzaj raportu (tabela/wykres).



Rysunek 23: Diagram sekwencji dla przypadku użycia 3.2.10.1 – Konfiguracja raportu (scenariusz główny)

3.2.10.2 Wyświetlanie raportu

Funkcja specjalizująca dla 3.2.10.1 – Wyświetlanie konfiguracji raportu.

Opis słowny – użytkownik powinien mieć możliwość wyświetlenia raportu w oknie aplikacji.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany, parametry raportu są wybrane
Opis przebiegu interakcji	Wyświetlenie raportu
Sytuacje wyjątkowe	Pusty raport
Warunki końcowe	Brak

Scenariusz główny:

1-4. Jak w scenariuszu generalizującym 3.2.10.1 – Wyświetlanie konfiguracji raportu.

5. Użytkownik wciska przycisk „Wyświetl raport”.

6. Aplikacja wyświetla okno z raportem w formie zgodnej z konfiguracją.

7. Użytkownik wciska przycisk „Zamknij”.

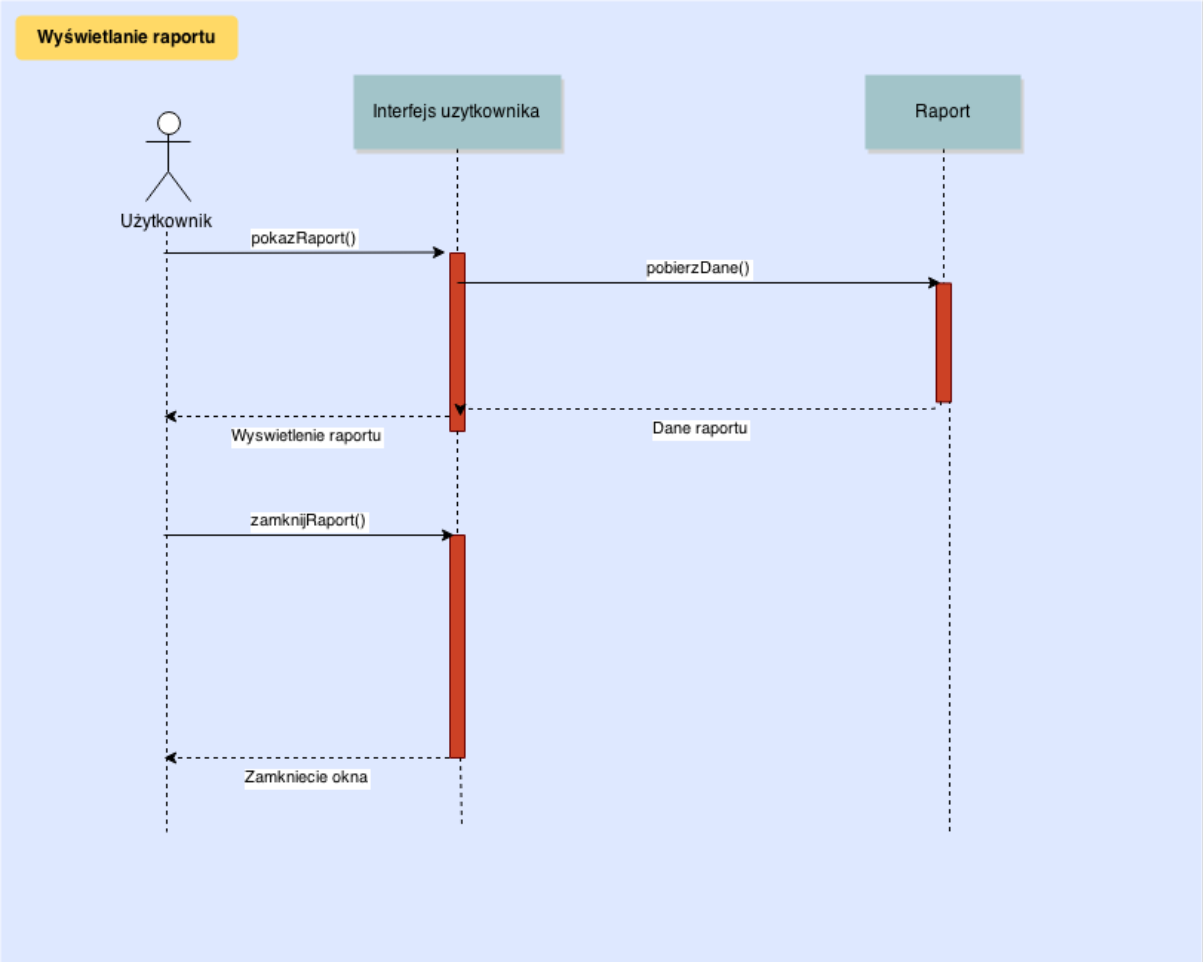
8. Aplikacja zamyka okno.

Scenariusz alternatywny – brak transakcji w raporcie

1-5. Jak w scenariuszu głównym.

6. Aplikacja wyświetla informację o pustym raporcie.

7. Powrót do kroku 1. scenariusza głównego.



Rysunek 24: Diagram sekwencji dla przypadku użycia 3.2.10.2 – Wyświetlanie raportu (scenariusz główny)

3.2.10.3 Zapisywanie raportu do pliku

Funkcja specjalizująca dla 3.2.10.1 – Wyświetlanie konfiguracji raportu.

Opis słowny – aplikacja powinna umożliwiać również zapisanie raportu do pliku o określonym formacie.

Aktor	Użytkownik
Warunki początkowe	Użytkownik jest zalogowany, parametry raportu są wybrane

Opis przebiegu interakcji	Wybór katalogu i nazwy pliku, zapis raportu
Sytuacje wyjątkowe	Pusty raport, przerwanie operacji
Warunki końcowe	Raport zapisany do pliku

Scenariusz główny:

1-4. Jak w scenariuszu generalizującym 3.2.10.1 – Wyświetlanie konfiguracji raportu.

5. Użytkownik wciska przycisk „Zapisz raport”.
6. Aplikacja wyświetla okno wyboru katalogu i nazwy pliku.
7. Użytkownik wybiera katalog i nazwę pliku i wciska przycisk „Zapisz”.
8. Aplikacja zapisuje raport w odpowiednim formacie do wybranego pliku i zamyka okno.

Scenariusz alternatywny – brak transakcji w raporcie

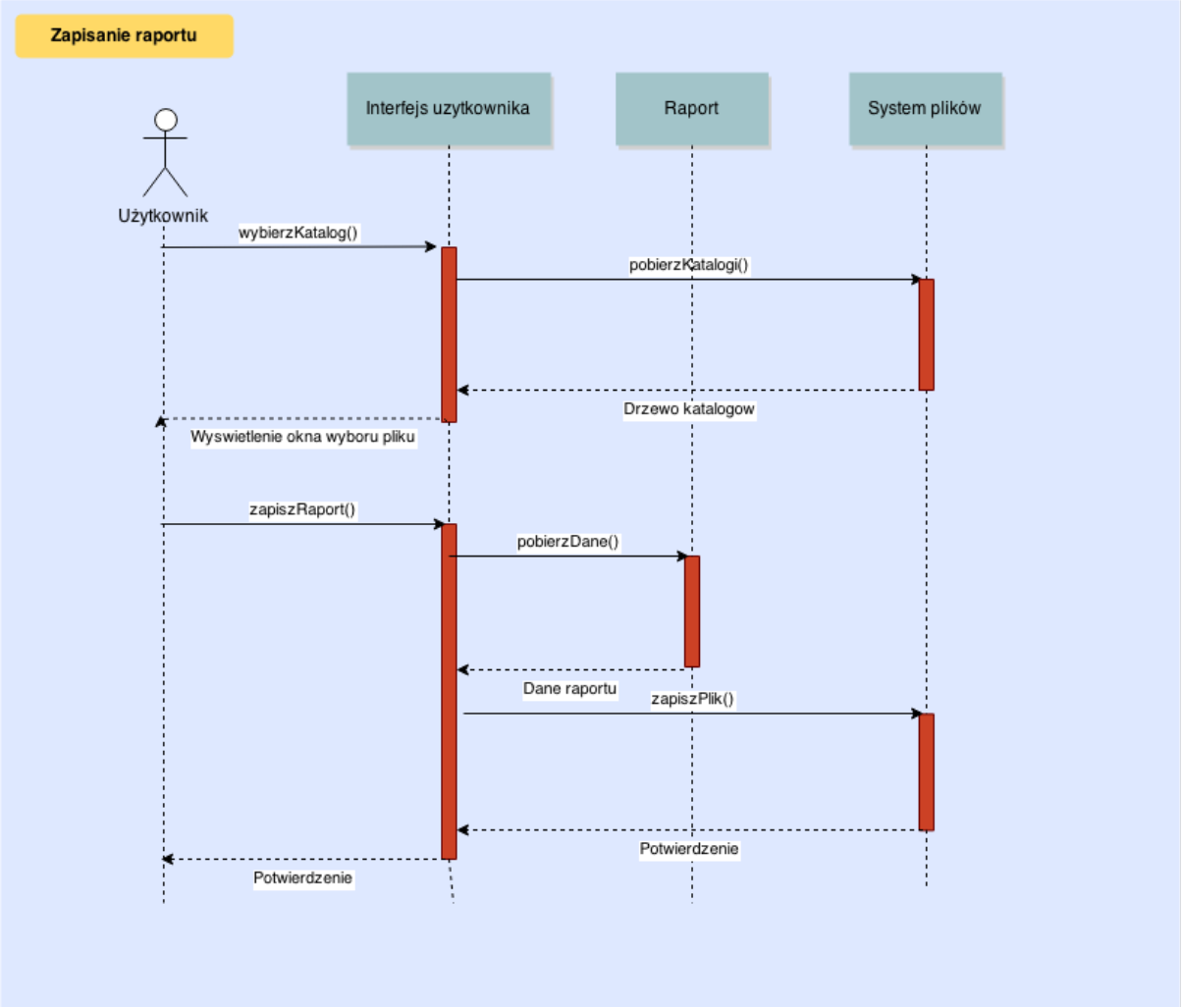
1-5. Jak w scenariuszu głównym.

6. Aplikacja wyświetla informację o pustym raporcie.
7. Powrót do kroku 1. scenariusza głównego.

Scenariusz alternatywny – przerwanie operacji przez użytkownika

1-6. Jak w scenariuszu głównym.

7. Użytkownik wciska przycisk „Anuluj”.
8. Aplikacja zamyka okno.
9. Powrót do kroku 2. ze scenariusza głównego.



Rysunek 25: Diagram sekwencji dla przypadku użycia 3.2.10.3 – Zapisywanie raportu do pliku (scenariusz główny)

3.3 Wymagania niefunkcjonalne

3.3.1 Bezpieczeństwo

3.3.1.1 Ograniczenia dostępu

Priorytet: Wysoki

Złożoność: Niska

Aplikacja nie może pozwalać niezalogowanym użytkownikom na dostęp do danych. Użytkownicy zalogowani mogą mieć dostęp wyłącznie do transakcji własnych oraz transakcji w budżetach zbiorczych. Ponadto użytkownicy mają możliwość edycji wyłącznie własnych transakcji.

3.3.1.2 Zabezpieczenie przed nieautoryzowanym odczytem danych

Priorytet: Wysoki

Złożoność: Średnia

Wszelkie dane przechowywane przez aplikację muszą być zabezpieczone przed odczytem i modyfikacją bez pośrednictwa aplikacji. Dane nie powinny być przechowywane w formacie jawnym.

3.3.1.3 Transakcyjność

Priorytet: Wysoki

Złożoność: Średnia

Aplikacja nie może pozwalać na realizację niekompletnych operacji. W wypadku wystąpienia błędu podczas realizacji operacji dane powinny zostać przywrócone do stanu sprzed rozpoczęcia operacji. Nie powinien być możliwy jednoczesny odczyt i zapis na tych samych danych.

3.3.2 Użytkowanie

3.3.2.1 Wieloinstancyjność

Priorytet: Średni

Złożoność: Niska

Aplikacja powinna pozwalać wielu użytkownikom na jednoczesne zalogowanie i korzystanie z jej funkcji. Nie powinna jednak pozwalać na wiele jednoczesnych sesji tego samego użytkownika.

3.3.3 Wygląd

3.3.3.1 Pomoc kontekstowa

Priorytet: Niski

Złożoność: Niska

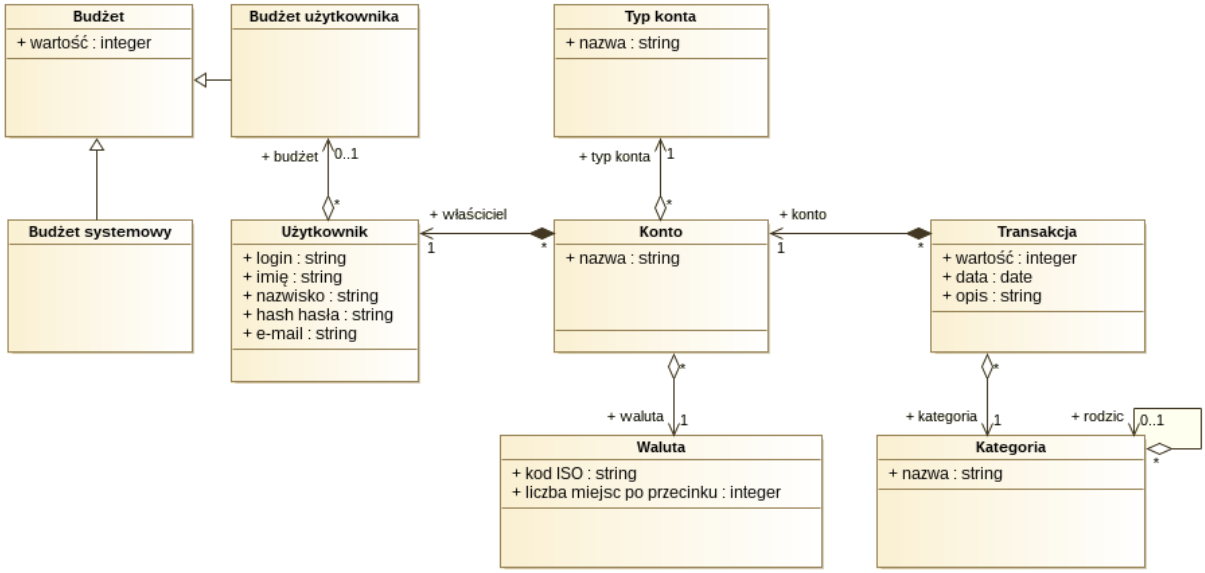
Aplikacja powinna zawierać mechanizm wyświetlania opisów dostępnych opcji wyjaśniających ich przeznaczenie i zastosowanie.

4 Model analityczny

Celem niniejszego rozdziału jest zdefiniowanie, jak wyglądać będzie architektura tworzonego systemu. Aby to osiągnąć, w rozdziale załączone zostały diagramy wykonane zgodnie ze standardem UML, które stanowią wizualną reprezentację architektury systemu oraz pozwalają na łatwiejszą analizę stanu projektu.

4.1 Diagram klas

Przedstawiony poniżej diagram klas reprezentuje wszystkie wykorzystywane przez Zleceniodawcę elementy składające się na cały system. Diagram ten jest kluczowy przede wszystkim dla deweloperów oraz innych osób zajmujących się bezpośrednio wytwarzaniem oprogramowania, tym niemniej powinien zostać zatwierdzony także przez przedstawicieli Zleceniodawcy – diagram klas jest bowiem punktem łączącym – z jednej strony wyobrażenie klienta o podziale funkcjonalności, a z drugiej decyzje projektowe podjęte przez zespół zajmujący się implementacją.



Rysunek 26: Diagram klas systemu.

Diagram klas obrazuje zależności (agregacje, kompozycje, relacje dziedziczenia) pomiędzy poszczególnymi klasami na tyle szczegółowo, by osoby nieposiadające wykształcenia informatycznego i nieznające metod programowania obiektowego mogły zrozumieć zasadę podziału bez szczegółowych wyjaśnień. Wszystkie atrybuty czy operacje ważne z punktu widzenia Zleceniodawcy, które mogą mieć wpływ na ocenę projektu zostały umieszczone na diagramie. Poniżej umieszczony został opis każdej z klas.

Użytkownik reprezentuję osobę korzystającą z aplikacji. Klasa ta zawiera istotne informacje potrzebne głównie do uwierzytelniania, takie jak login oraz hash hasła. Pozostałe pola to na przykład imię i nazwisko, czy adres e-mail.

Konto jest klasą reprezentującą konto pieniężne (np. oszczędnościowe, osobiste, czy walutowe), które należy do danego użytkownika. Klasa ta posiada pola, takie jak nazwa, typ, waluta

oraz bieżące saldo.

Typ konta jest klasą wydzieloną z klasy Konto. Utworzenie tej klasy pozwoli na ograniczenie wartości wprowadzanych w polu typ konta.

Waluta jest kolejną klasą wydzieloną z klasy Konto. Utworzenie tej klasy pozwoli na ograniczenie wartości wprowadzanych w polu waluta. Ponadto przechowuje ona liczbę miejsc po przecinku, na jakie zezwala dana waluta, a więc pozwoli na walidację wartości pieniężnych wprowadzanych przez użytkownika do systemu.

Transakcja reprezentuje przepływ pieniędzy między kontami. Istnieją trzy typy transakcji – przychody (pieniądze pochodzące z konta poza systemem, które trafiają na jedno z kont użytkownika), koszty (pieniądze pochodzące z jednego z kont użytkownika, które trafiają na konto poza systemem), lub przelew (pieniądze transferowane pomiędzy dwoma kontami, oba należące do tego samego użytkownika). Klasa ta zawiera także inne pola, takie jak data, opis (opcjonalnie), czy wartość pieniężna.

Kategoria reprezentuje etykiety, które mogą być przypisane do transakcji. Podobnie jak klasy Waluta oraz Typ konta, jest to klasa wydzielona z innej klasy, w tym przypadku z Transakcji. Zawiera tylko pole przechowujące nazwę. Kategorie tworzą strukturę drzewa – jest to dozwolone poprzez rekurencyjną relację jeden-do-wielu.

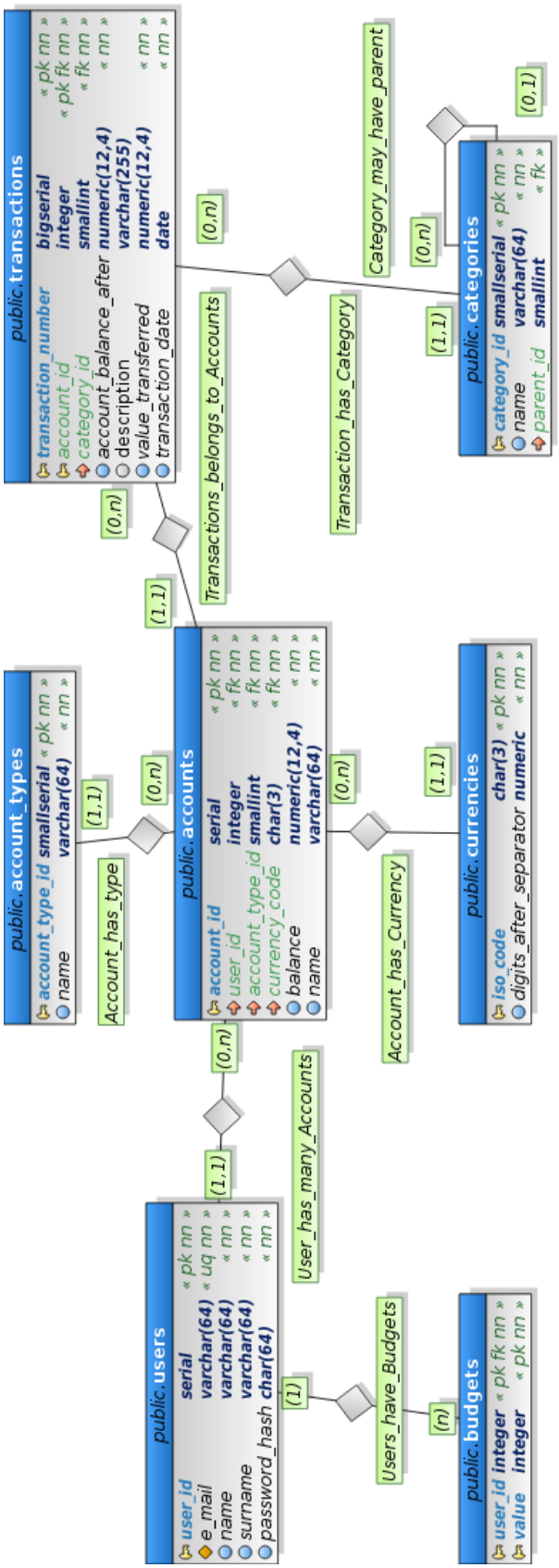
Budżet jest klasą abstrakcyjną reprezentującą budżet, czyli ograniczenie nakładane na użytkownika lub cały system. Zawiera ona tylko jedno pole – wartość ograniczenia. Istnieją dwie klasy dziedziczące po klasie Budżet: Budżet użytkownika oraz Budżet systemowy, odpowiednio reprezentują one ograniczenia nałożone na konkretnego użytkownika, lub cały system.

4.2 Model bazy danych

Ważnym elementem tworzonego systemu informatycznego jest baza danych przechowująca wszystkie informacje, które powinny być w niej trwale. Jak opisano w części dotyczącej sprzętu, wszystkie informacje powinny być przechowywane redundantnie, na przynajmniej dwóch niezależnych od siebie macierzach dyskowych. Pozwoli to na uniknięcie problemów związanych z ewentualnym awariami i chwilowymi przerwami w dostępności.

W czasie tworzenia projektu bazy danych opierano się przede wszystkim na stworzonym wcześniej diagramie klas. Większość klas została zamieniona na odpowiadające im tabele w bazie danych. W przypadku relacji wiele-do-wielu konieczne było stworzenie dodatkowych tabel łączących. Enumeracje zostały zamienione na osobne tabele tylko w uzasadnionych przypadkach.

Poniżej zaprezentowano model bazy danych wykorzystywanej w tworzonym systemie.



Rysunek 27: Model bazy danych systemu.

Poniżej znajdują się opisy każdej z tabel w bazie danych wraz z wyjaśnieniem poszczególnych kolumn w każdej z nich.

4.2.0.2 Users

tabela przechowująca dane o użytkownikach systemu.

Nazwa atrybutu	Opis	Typ danych	Ograniczenia
user_id	PRIMARY KEY	serial	NOT NULL, UNIQUE
e_mail	adres elektroniczny użytkownika	varchar(64)	NOT NULL, UNIQUE
name	imię użytkownika	varchar(64)	NOT NULL
surname	nazwisko użytkownika	varchar(64)	NOT NULL
password_hash	hash hasła użytkownika	char(64)	NOT NULL

4.2.0.3 Accounts

tabela przechowująca dane o kontach znajdujących się w systemie.

Nazwa atrybutu	Opis	Typ danych	Ograniczenia
account_id	PRIMARY KEY	serial	NOT NULL, UNIQUE
user_id	FOREIGN KEY	integer	NOT NULL
account_type_id	FOREIGN KEY	smallint	NOT NULL
currency_code	FOREIGN KEY	char(3)	NOT NULL
balance	aktualny stan konta	numeric(12,4)	NOT NULL
name	nazwa konta	varchar(64)	NOT NULL

4.2.0.4 Account types

tabela przechowująca możliwe typy kont dostępne w systemie.

Nazwa atrybutu	Opis	Typ danych	Ograniczenia
account_type_id	PRIMARY KEY	smallint	NOT NULL, UNIQUE
name	nazwa typu konta	varchar(64)	NOT NULL

4.2.0.5 Currencies

tabela przechowująca dane o walutach dostępnych w systemie.

Nazwa atrybutu	Opis	Typ danych	Ograniczenia
iso_code	PRIMARY KEY	char(3)	NOT NULL, UNIQUE
digits_after_separator	liczba miejsc po przecinku w danej walucie	numeric	NOT NULL

4.2.0.6 Transactions

tabela przechowująca dane o transakcjach.

Nazwa atrybutu	Opis	Typ danych	Ograniczenia
transaction_number	PRIMARY KEY	bigserial	NOT NULL
account_id	FOREIGN KEY, PRIMARY KEY	serial	NOT NULL
category_id	FOREIGN KEY	smallint	NOT NULL
value_transferred	ilość pieniędzy, jaka została przesłana	numeric(12,4)	NOT NULL
transaction_date	data transakcji	date	NOT NULL
description	opcjonalny opis	varchar(255)	-
account_balance_after	kontrolna kolumna zawierająca stan konta po transakcji (świadoma denormalizacja)	numeric(12,4)	NOT NULL

4.2.0.7 Categories

tabela przechowująca dane o kategoriach transakcji zdefiniowanych w systemie. Kategorie mogą tworzyć strukturę drzewiastą, stąd rekurencyjny FOREIGN KEY.

Nazwa atrybutu	Opis	Typ danych	Ograniczenia
category_id	PRIMARY KEY	smallserial	NOT NULL
parent_id	FOREIGN KEY – opcjonalny rodzic	smallserial	-
name	nazwa kategorii	varchar(64)	NOT NULL

4.2.0.8 Budgets

tabela przechowująca dane o budżetach.

Nazwa atrybutu	Opis	Typ danych	Ograniczenia
user_id	FOREIGN KEY	integer	NOT NULL
value	wartość budżetu	integer	NOT NULL

5 Rozwiązania projektowe

5.1 Środowisko

Obecnie coraz większa część systemów informatycznych realizowana jest w postaci aplikacji z dostępem z poziomu przeglądarki internetowej. Taki zabieg pozwala na bardzo łatwy dostęp do systemu z poziomu praktycznie dowolnego urządzenia posiadającego dostęp do internetu. Innymi kluczowym czynnikiem jest duża łatwość w dystrybucji takiego rozwiązania i w związku z tym klient również zdecydował się na zastosowanie takiego podejścia.

Środowiskiem pracy dla użytkowników tworzonego systemu będzie przeglądarka internetowa. Dzięki temu klienci będą mogli korzystać z dostarczonego systemu zarówno przy użyciu komputera osobistego jak i urządzenia mobilnego. Rozwiązanie takie jak dedykowane aplikacje mogą być przydatne na niektórych rodzajach urządzeń, jednak tworzenie ich na wszystkie możliwe rynki (stacjonarne, mobile itp.) stanowiłoby duże wyzwanie i spowodowało znaczące przekroczenia zarówno budżetu jak i harmonogramu.

Zdecydowano się na wsparcie następujących rodzajów przeglądarek:

1. Google Chrome (od wersji 23 wzwyż)
2. Mozilla Firefox (od wersji 21 wzwyż)
3. Safari (od wersji 6 wzwyż)
4. Opera (od wersji 15 wzwyż)
5. Internet Explorer (od wersji 10 wzwyż)

Pozostałe przeglądarki także powinny poprawnie prezentować stronę internetową sklepu, jednak wsparcie dla nich nie jest wymaganiem, a co za tym idzie, dla przeglądarek tych nie będą przeprowadzane testy.

Wygląd strony internetowej powinien być taki sam (z różnicami maksymalnie 0.04% zawartości) dla każdej przeglądarki internetowej. Ewentualne różnice wynikające na przykład z różnicy w formatach monitorów czy ich wielkości powinny być obsługiwane przez mechanizmy wewnętrzne.

Ewentualne aplikacje wspomagające korzystanie ze sklepu (na przykład zdobywające coraz większą popularność aplikacje na urządzenia mobilne) nie znajdują się w fazie analizy w niniejszym projekcie, ewentualnie mogą zostać stworzone w czasie rozbudowy i utrzymywania systemu. Aby pozostawić możliwość tego rodzaju rozszerzeń należy zadbać o odpowiedni protokół komunikacyjny uniezależniający działanie serwerów aplikacyjnych i bazy danych od klienta, który dostarcza dane i polecenia.

5.2 Architektura

Jedną z głównych decyzji w czasie procesu projektowania tworzonego systemu jest wybór odpowiedniej architektury. Nasza aplikacja będzie wykorzystywała architekturę trójwarstwową (*ang.* three-tier architecture), ponieważ jest to najpopularniejsza architektura stosowana w aplikacjach dostępnych z poziomu przeglądarki internetowej. Architektura trójwarstwowa jest przykładem

aplikacji typu klient-serwer. Dzieli się ją na trzy warstwy, dzięki czemu proces tworzenia i nanoszenia poprawek jest łatwiejszy. Co więcej, daje to możliwość zredukowania zależności pomiędzy modułami programu. Architektura składa się z następujących warstw:

- **warstwa danych** (*ang.* data tier) – odpowiada za trwałe przechowywanie danych, udostępnia interfejs do ich pobierania i modyfikacji. Realizowana za pomocą relacyjnej bazy danych.
- **warstwa logiki biznesowej** (*ang.* business logic tier) – odpowiada za logikę całego systemu, koordynuje działania odpowiadające operacjom na kontach i użytkownikach. Znajduje się na serwerze aplikacyjnym. Pełni rolę pośrednika pomiędzy pozostałymi warstwami.
- **warstwa prezentacji** (*ang.* presentation tier) – odpowiada za prezentowanie aplikacji na komputerze użytkownika, zbieranie jego akcji i przekazywanie ich do warstwy logiki biznesowej.

Wybór architektury trójwarstwowej pociąga za sobą technologie w jakie zostaną zastosowane. Do stworzenia interfejsu użytkownika wykorzystane zostaną technologie wykorzystywane w projektowaniu stron internetowych, czyli HTML i CSS. Aby poprawić User Experience, można również założyć wykorzystanie języka JavaScript oraz technologii AJAX. Warstwa logiki, czyli tzw. backend zostanie stworzony przy użyciu języka Java, ze względu na wydajność, łatwość we wprowadzaniu zmian, oraz rozbudowaną dokumentację. Dodatkowo wykorzystany zostanie framework webowy SpringMVC. Jako system zarządzania bazą danych (*ang.* DBMS) postanowiono wykorzystać PostgreSQL.

Rysunek 28: Architektura trójwarstowa, z zaznaczonymi wybranymi technologiami.

5.3 Sprzęt

Wdrożenie systemu wymagać będzie zapewnienia odpowiedniej infrastruktury sprzętowej. W celu zapewnienia skalowalności będzie ona podzielona na serwer aplikacyjny i serwer bazodanowy. Jako że obsługa systemu od strony użytkownika będzie opierać się o interfejs webowy kompatybilny z!ogromną większością komputerów osobistych wyposażonych w przeglądarkę internetową, wykorzystanie dodatkowych stacji klienckich nie będzie konieczne.

Biorąc pod uwagę takie czynniki jak cena i wydajność, a także niezawodność i głośność pracy, zdecydowano się na wybór komputerów serwerowych z oferty firmy *Dell*.

5.3.1 Serwer aplikacyjny

Jako serwer obsługujący warstwę logiki wybrano *Dell PowerEdge R220* ze względu na korzystny stosunek wydajności do ceny.

Rysunek 29: Dell PowerEdge R220

Procesor	Intel® Celeron® G1820 2.7GHz
Pamięć RAM	4 x 4GB UDIMM, 1600 MT/s, Low Volt, Single Rank, x8 Data Width
Dysk	500GB 7.2k RPM SATA 6Gbps Entry 3.5in Cabled Hard Drive
Zasilanie	250W (80+ Silver); auto-ranging (100V–240V)

5.3.2 Serwer bazy danych

Baza danych obsługiwana będzie przez serwer *Dell PowerEdge R320*.

Rysunek 30: Dell PowerEdge R320

Procesor	Intel® Pentium® 1403 v2 2.60GHz
Pamięć RAM	4 x 4GB UDIMM, 1600 MT/s, Low Volt, Single Rank, x8 Data Width
Dysk	500GB 7.2k RPM SATA 6Gbps Entry 3.5in Cabled Hard Drive
Zasilanie	350W (80+ Silver); auto-ranging (100V–240V)

5.3.3 Szafa rack

Oba serwery wyposażone są w obudowy typu *rack 19"* i są kompatybilne z wszelkiego rodzaju szafami wykorzystującymi ten standard. Biorąc pod uwagę niewielkie rozmiary i solidność wykonania, sugerowane jest wykorzystanie szafy serwerowej *Signati 12U*.

Rysunek 31: Signati 12U

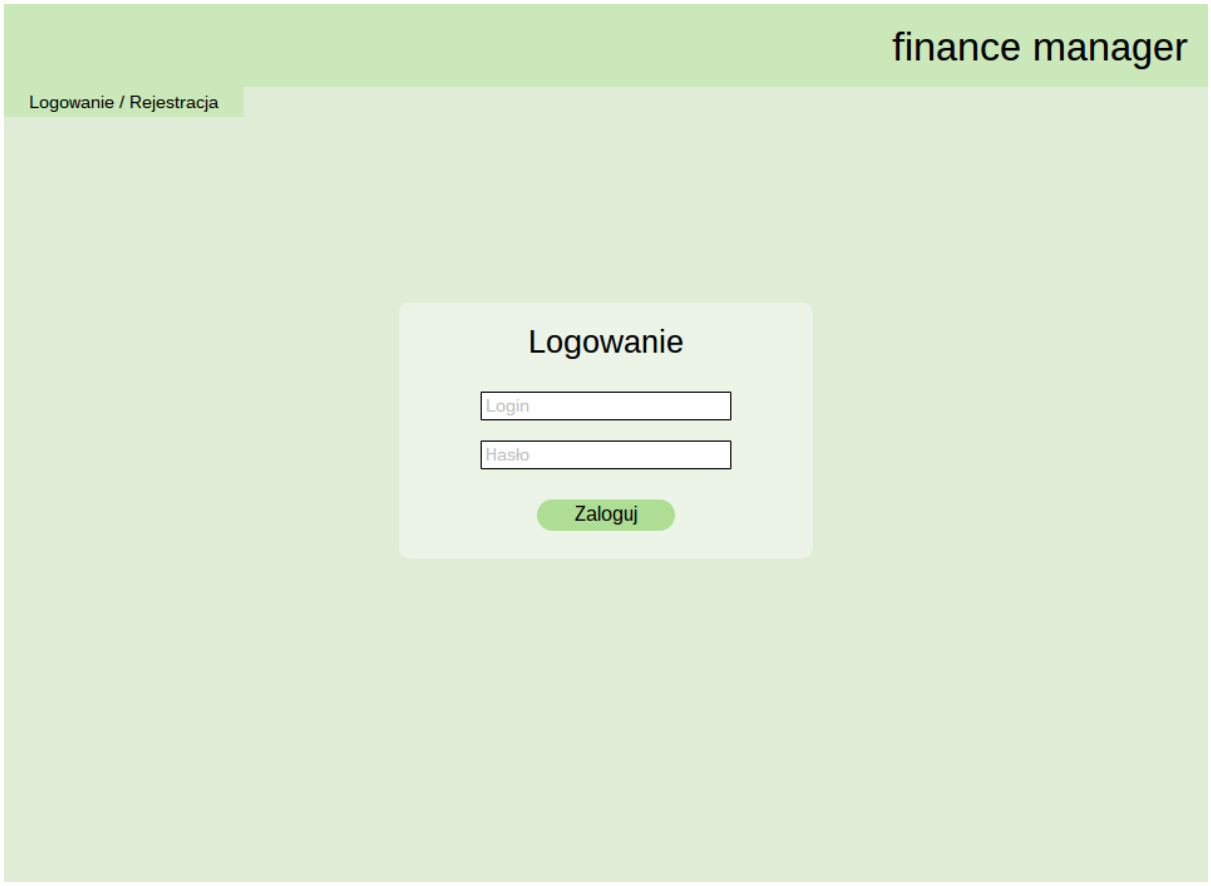
6 Interfejs użytkownika

W dzisiejszych czasach interfejs użytkownika graficzny użytkownika stanowi niezwykle istotny element każdego systemu informatycznego. Powinien on łączyć w sobie dwie cechy: użyteczność i atrakcyjność. Ważne jest uzyskanie odpowiedniego balansu pomiędzy tymi cechami. Skupienie się wyłącznie na użyteczności prawdopodobnie odbije się negatywnie na atrakcyjności i odwrotnie.

Dodatkowo interfejs powinien nawiązywać do aktualnych trendów w tej dziedzinie. Nawet najładniejsza i łatwa w obsłudze aplikacja, która powstała kilka lat temu nie będzie dorównywała aplikacjom stworzonym na przestrzeni ostatnich miesięcy. Wynika to głównie ze zmiany podejścia do tworzenia interfejsów. Obecnie projektowane interfejsy muszą często dobrze przezentować się zarówno na komputerach stacjonarnych, jak i urządzeniach mobilnych. Wymusza to stosowanie prostych kontrolek, oraz ograniczanie ilości informacji do niezbędnych dla działania.

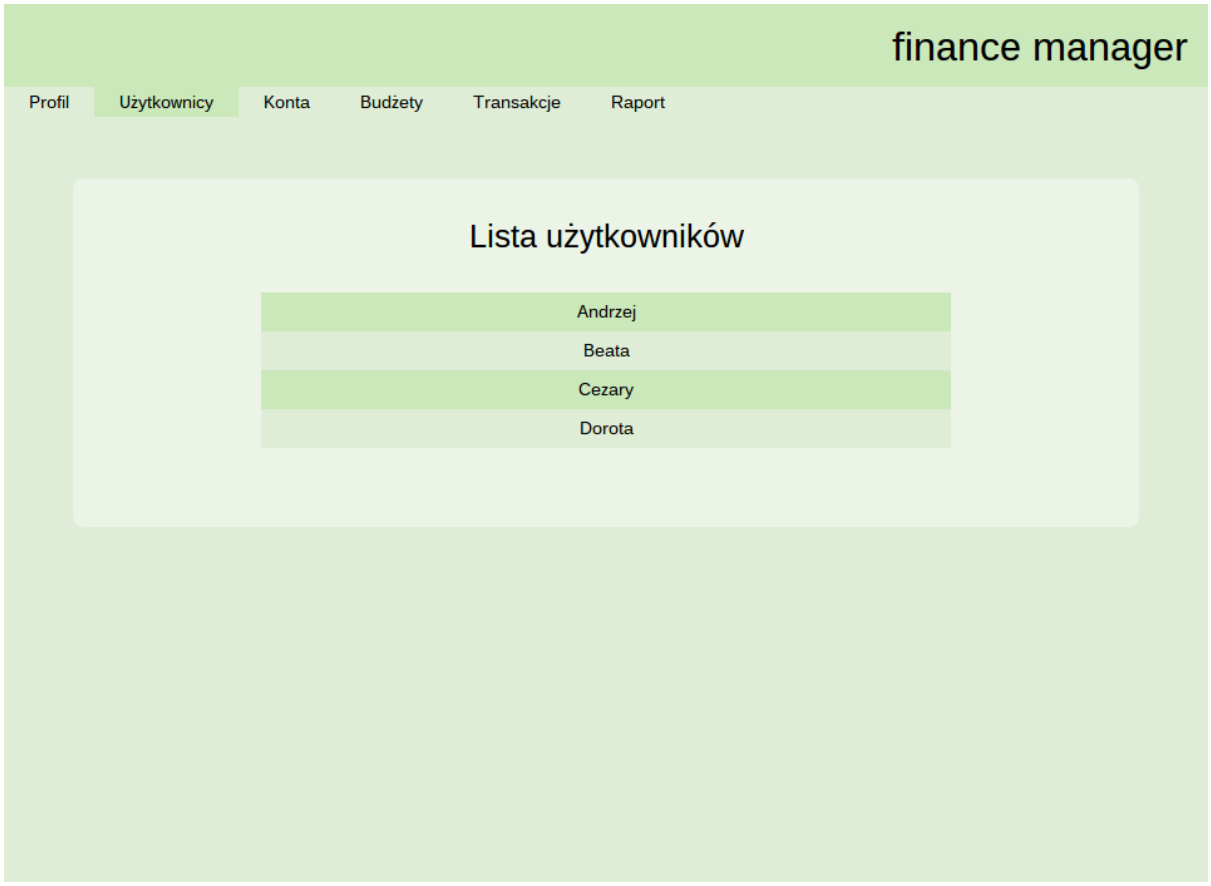
Obecny trend wskazuje, że za najładniejsze uznawane są aplikacje minimalistyczne (np. material design). Przedstawiony poniżej prototyp interfejsu został stworzony z myślą o użyteczności, jednak również jego atrakcyjność można uznać za nawiązującą do panującej mody.

Należy pamiętać, że jest to jedynie prototyp i nie prezentuje on wszystkich funkcjonalności, ani widoków jakie będzie posiadał przyszły system, a jedynie narzuca pewną konwencje, która powinna zostać zachowana przy tworzeniu pozostałych elementów.



Rysunek 32: Prototyp widoku logowania.

Strona startowa jaką zobaczy użytkownik prezentuje się jak na rysunku 32. Z tego poziomu każdy użytkownik posiadający w systemie konto będzie mógł zalogować się do swojego prywatnego panelu. Jeżeli system odwiedzi użytkownik, który swojego konta w systemie nie posiada, będzie on musiał przejść do strony rejestracji, klikając odpowiedni odnośnik na banerze znajdującym się pod logo aplikacji.

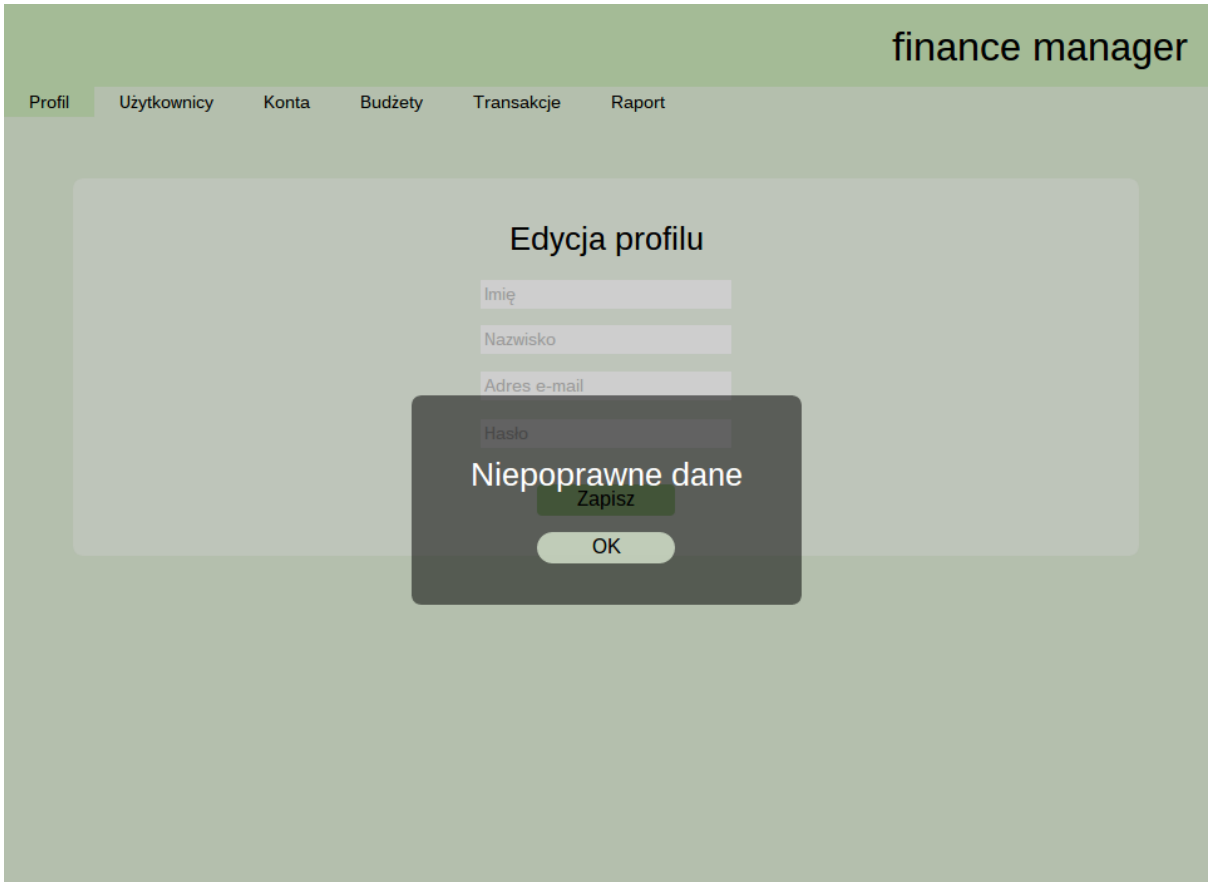


Rysunek 33: Prototyp widoku z listą użytkowników.

Wszystkie pozostałe prezentowane widoki zakładają, że użytkownik pomyślnie zalogował się do systemu. Na rysunku 33 znajduje się widok z listą użytkowników, tak aby każdy mógł sprawdzić, kto również posiada konto w systemie.

Rysunek 34: Prototyp widoku z listą transakcji.

Rysunek 34 prezentuje widok transakcji. Z tego poziomu użytkownik może dokonywać przeglądu dokonanych wydatków, lub uzyskanych przychodów. Po zaznaczeniu odpowiedniej transakcji, uaktywniają się opcje edycji i usuwania zaznaczonej transakcji. Dzięki temu można kontrolować przypadkowo wstawione operacje, lub takie które nie mają znaczącego wpływu na stan konta. Dodakowo, co może się okazać szczególnie przydatne w przypadku gdy lista transakcji urośnie do większych rozmiarów, istnieje możliwość sortowania transakcji według jednego z trzech kluczy.



Rysunek 35: Prototyp widoku pozwalającego na edycję danych użytkownika, z wyświetlonym komunikatem o błędzie.

Ostatni rysunek nr 35 pokazuje widok edycji danych użytkownika. Na pierwszym planie wyświetlony jest komunikat o błędzie, aby zaprezentować w jaki sposób można powiadamiać użytkownika o występujących problemach.

Pozostałe widoki utworzone zostaną z zachowaniem wyznaczonych styli i standardów.