



**Politechnika
Śląska**

Wydział Informatyki, Elektroniki i Informatyki

Aplikacja do zarządzania budżetem domowym

TWORZENIE APLIKACJI BAZODANOWYCH

Mateusz Cudzik

Jakub Ferens

Mateusz Górecki

Szymon Maciąg

Kajetan Sommer

Julia Wojciuch

Gliwice

13 czerwca 2023

Spis treści

1	Wstęp	2
2	Określenie wymagań	2
2.1	Wymagania funkcjonalne	2
2.2	Wymagania niefunkcjonalne	2
3	Analiza MoSCoW	3
3.1	Must	3
3.2	Should	3
3.3	Could	3
3.4	Won't	4
4	Scenariusze przypadków użycia	4
4.1	Logowanie	4
4.1.1	Scenariusz główny logowania	4
4.1.2	Scenariusze poboczne logowania	4
4.2	Dodawanie przychodów/wydatków	5
4.2.1	Scenariusz główny dodawania przychodów/wydatków .	5
4.2.2	Scenariusze poboczne dodawania przychodów/wydatków	5
5	Diagram UML	6
6	Schemat bazy danych	7
7	Specyfikacja zewnętrzna	8
8	Specyfikacja wewnętrzna	8
9	Testowanie i uruchamianie	8
10	Weryfikacja osiągniętych efektów względem założeń	8
10.1	Schemat zaimplementowanej bazy danych	8
11	Uwagi i wnioski	9

1 Wstęp

Celem projektu jest stworzenie aplikacji do zarządzania domowym budżetem i jego monitorowania. Warunkiem koniecznym jest obsługiwanie logowania, co umożliwi korzystanie z niej wielu użytkownikom. Aplikacja pozwoli na tworzenie raportów oraz kategoryzowanie przychodów i wydatków z różnych kont.

2 Określenie wymagań

2.1 Wymagania funkcjonalne

- Obsługa logowania
- Kategorie wydatków/przychodów
- Kategorie kont
- Transakcje między profilami
- Generowanie raportów - analiza finansowa
- Przechowywanie skanów paragonów/faktur
- Przechowywanie dłużników
- Informacje przechowywane w bazie danych
- Dodawanie profili członków rodziny do konta (profil dziecka, rodzica itd.)
- Dodanie konta bankowego i operacji na nim
- Obsługa wydatków i przychodów

2.2 Wymagania niefunkcjonalne

- Bezpieczeństwo – okresowe tworzenie kopi zapasowych danych
- Zabezpieczenie profili użytkowników hasłem
- Hierarchia użytkowników – różne poziomy uprawnień/dostępu, ograniczenia dla profili młodszych użytkowników
- Użyteczność – aplikacja z przystępnym i łatwym w obsłudze interfejsem zarówno dla starszych jak i młodszych użytkowników
- Wieloplatformowość – przypadku aplikacji webowej dostępność z różnych urządzeń przy pomocy dowolnego systemu posiadającego przeglą-

darke

- System/Aplikacja przystosowana do łatwego rozwoju, rozbudowy i aktualizacji
- Responsywność – odpowiedź aplikacji na działania użytkownika w określonym czasie (przykładowo do trzech sekund)

3 Analiza MoSCoW

3.1 Must

- Przechowywanie informacji w bazie danych
- Dodawanie wydatków i przychodów
- Generowanie raportów
- Założenie konta i przypisania do niego danych
- Informowanie użytkownika o aktualnym stanie konta, który jest zmieniany wraz z kolejnymi wpisami o przychodach/wydatkach

3.2 Should

- Przypomnienie hasła
- Formularz rejestracji dostępny dla użytkownika
- Dzielenie wydatków i przychodów na kategorie
- Generowanie raportów z podziałem wydatków/przychodów na kategorie
- Operacje zarządzania profilami (dodawanie, usuwanie itd.)

3.3 Could

- Potwierdzenie rejestracji mailem
- Edycja informacji o koncie (nazwy użytkownika, hasła, itd.)
- Ustawianie cyklicznych/stałych wydatków/przychodów
- Transakcje między profilami
- Przechowywanie skanów paragonów/faktur
- Definiowanie własnych, niestandardowych kategorii

3.4 Won't

- Weryfikacja Captcha
- Przechowywanie informacji o dłużnikach
- Powiadomienia o przekroczonym budżecie

4 Scenariusze przypadków użycia

4.1 Logowanie

4.1.1 Scenariusz główny logowania

- Przypadek rozpoczyna się, gdy niezalogowany użytkownik wejdzie na stronę
- Użytkownik wpisuje swój login oraz hasło
- System sprawdza poprawność danych
- Użytkownik zostaje przeniesiony do panelu wyboru profilu
- Użytkownik wybiera profil.
- Wyświetlony zostaje panel sterowania budżetem
- Użytkownik zostaje zalogowany

4.1.2 Scenariusze poboczne logowania

Konto nie istnieje

- Użytkownik zostaje przeniesiony do formularza rejestracji
- Użytkownik wprowadza swoje dane
- System sprawdza poprawność danych
- Konto zostaje utworzone

Wybrany profil jest chroniony

- Użytkownik wpisuje PIN
- System sprawdza poprawność danych
- W przypadku wprowadzenia poprawnego kodu pin scenariusz się kończy, w przeciwnym razie użytkownik jest informowany o błędnym kodzie PIN, po kilku błędnych próbach nakładana jest czasowa blokada

Wybrany profil jest profilem dziecka

- Użytkownikowi wyświetlone zostaje uproszczone GUI

4.2 Dodawanie przychodów/wydatków

4.2.1 Scenariusz główny dodawania przychodów/wydatków

- Zalogowany użytkownik decyduje się dodać przychód/wydatek na panelu sterowania budżetem
- Użytkownik wpisuje kwotę, nazwę własną operacji oraz wybiera jej kategorię
- Operacja zostaje uwzględniona w budżecie

4.2.2 Scenariusze poboczne dodawania przychodów/wydatków

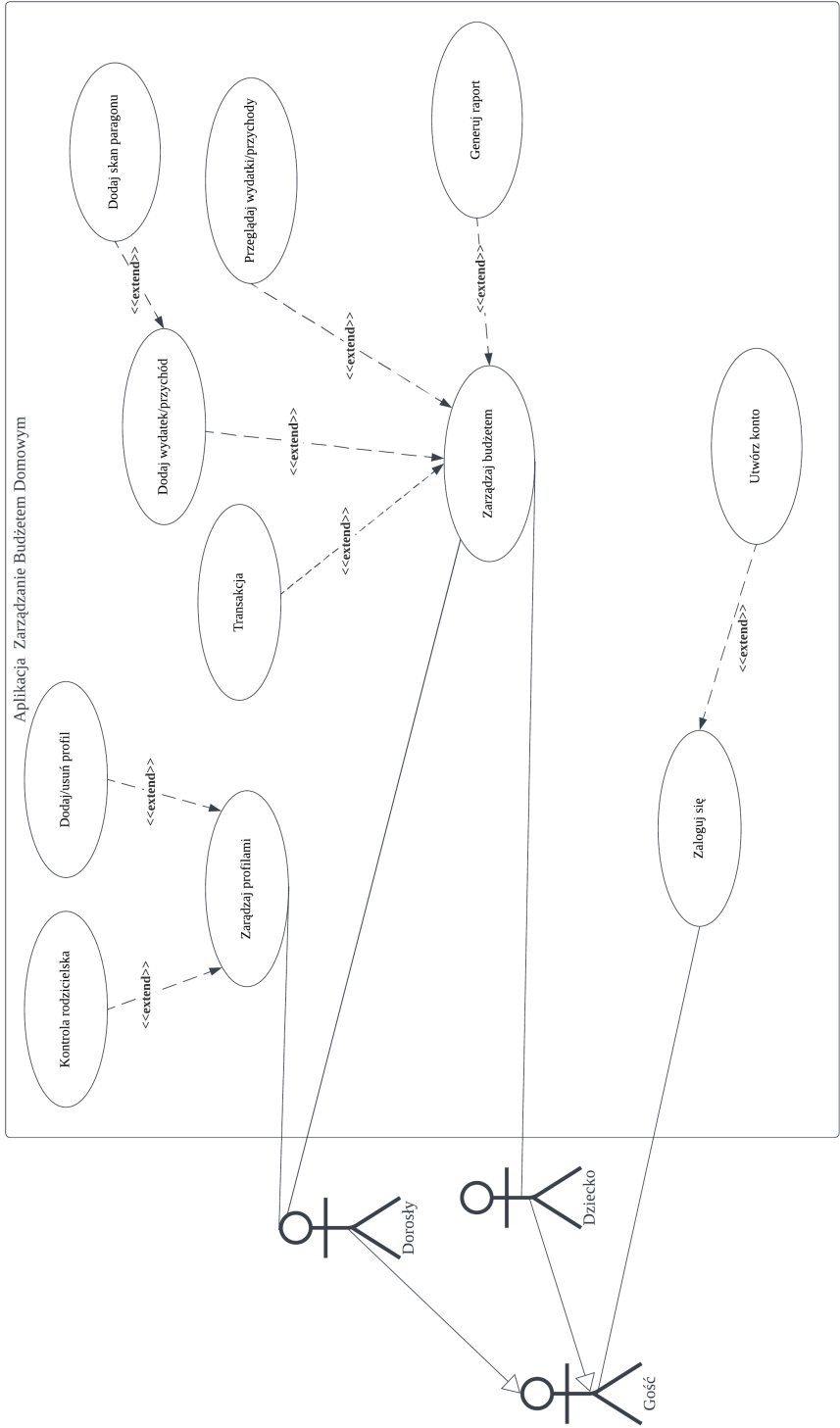
Użytkownik dodaje własną kategorię

- Użytkownik podaje nazwę i wybiera kolor
- Kategoria zostaje dodana do listy wszystkich kategorii

Użytkownik dodaje wydatek przekraczający saldo

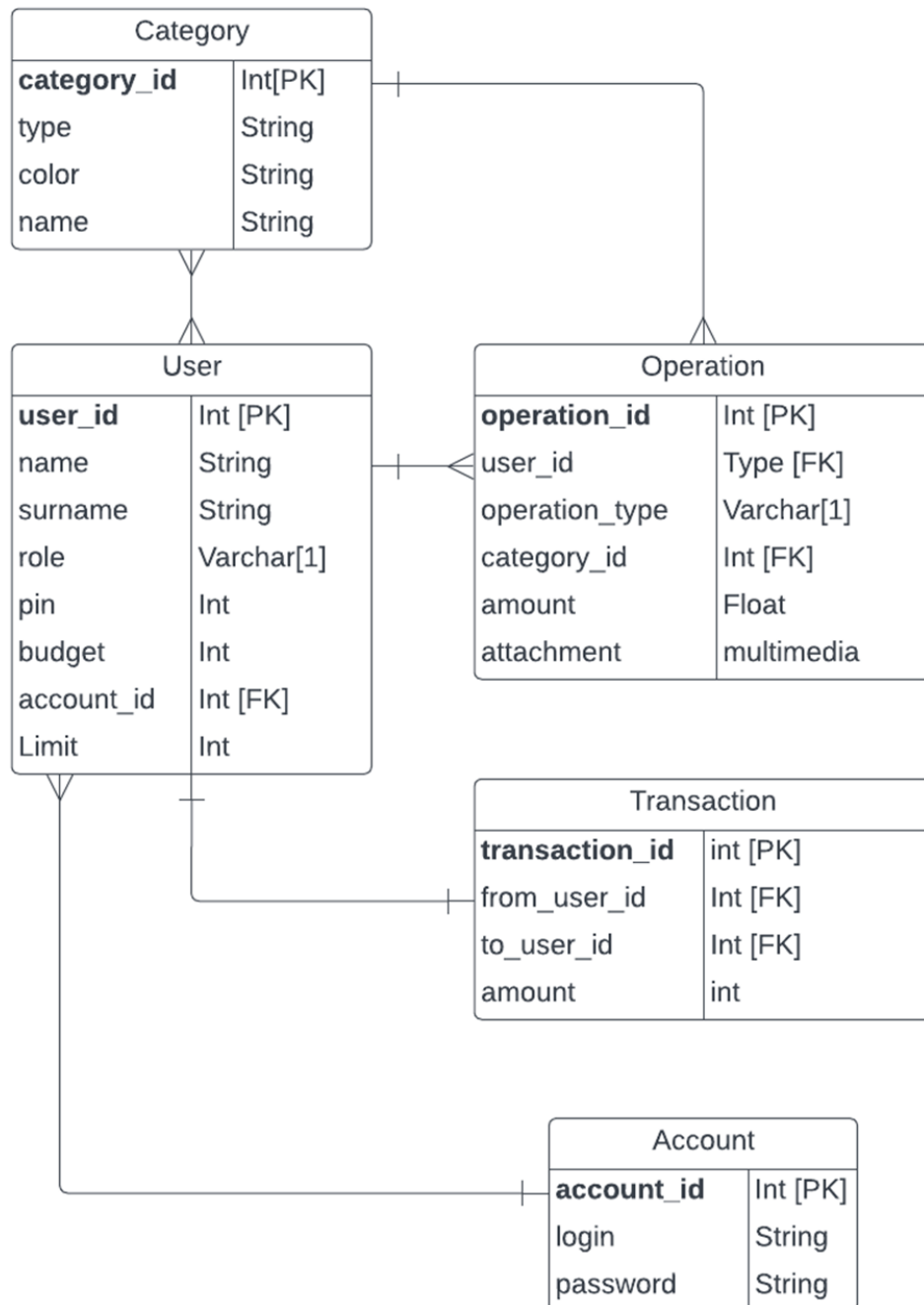
- Użytkownik zostaje ostrzeżony za pomocą powiadomienia przed wykonaniem operacji
- Użytkownik anuluje albo potwierdza wykonanie transakcji

5 Diagram UML



Rysunek 1: Diagram UML

6 Schemat bazy danych



Rysunek 2: Prototypowy schemat bazy danych

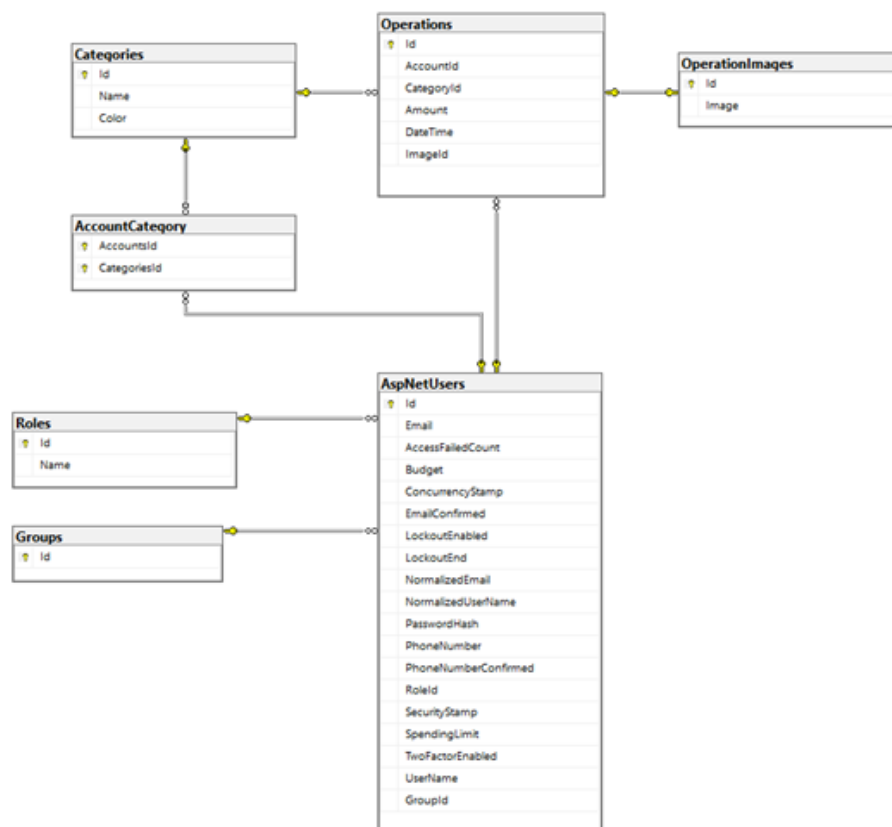
7 Specyfikacja zewnętrzna

8 Specyfikacja wewnętrzna

9 Testowanie i uruchamianie

10 Weryfikacja osiągniętych efektów względem założeń

10.1 Schemat zaimplementowanej bazy danych



Rysunek 3: Wykorzystany schemat bazy danych

11 Uwagi i wnioski