



**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

18 czerwca 2023

Spis treści

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

SPIS TREŚCI	2
--------------------	----------

11 Uwagi i wnioski	10
---------------------------	-----------

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 kopii 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-

- darzę,
- 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

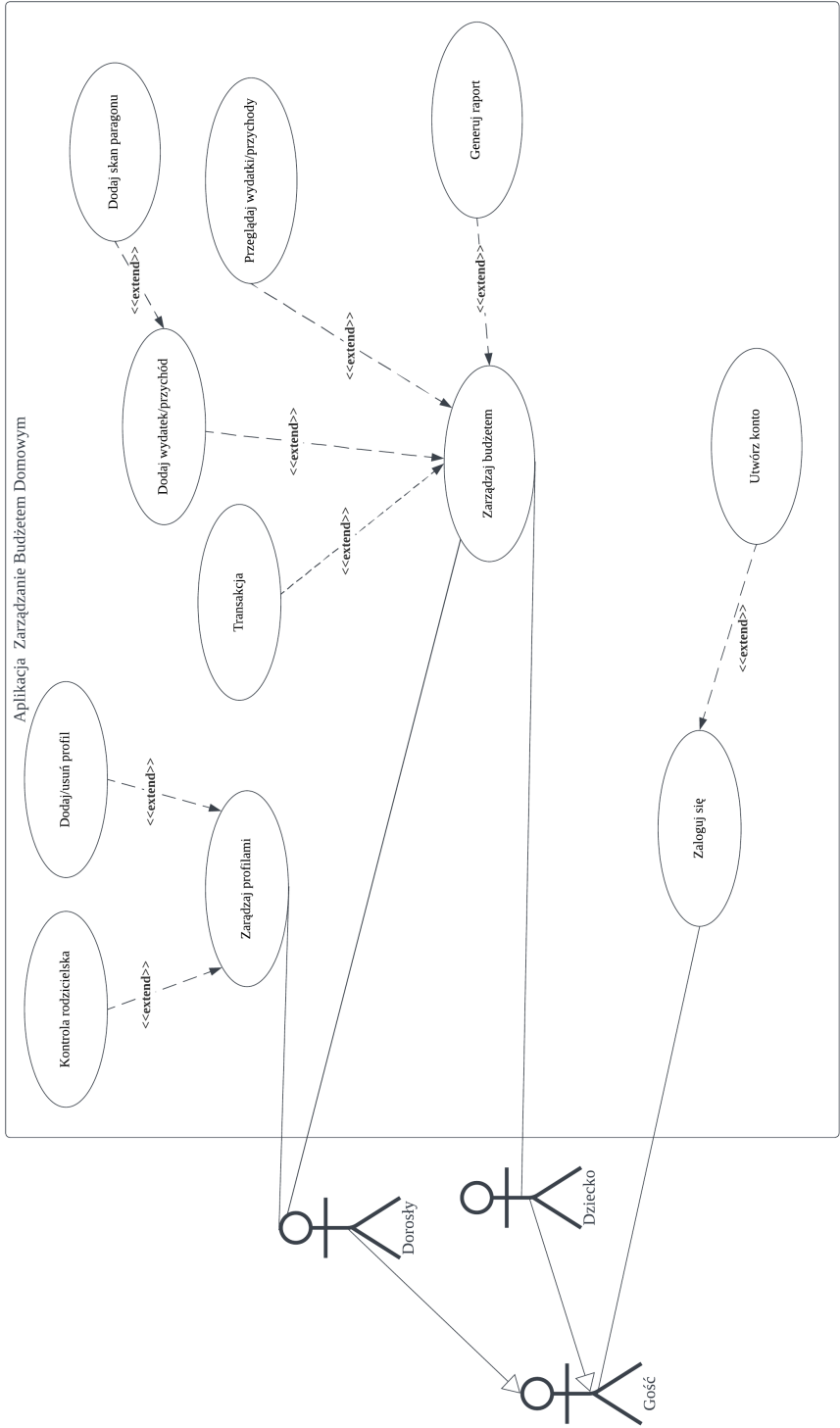
Użytkownik dodaje własną kategorię

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

Użytkownik dodaje wydatek przekraczający saldo

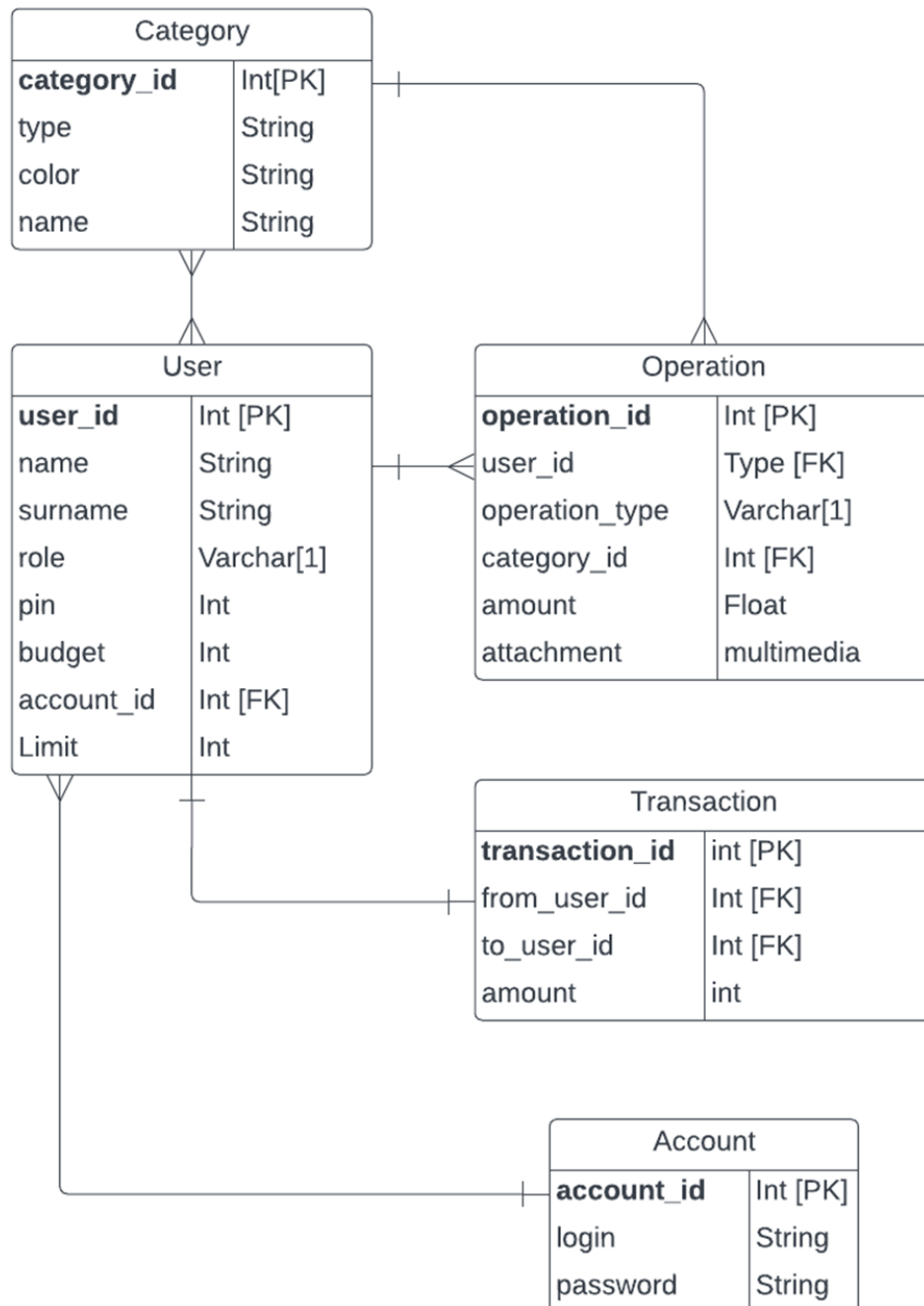
- Użytkownik zostaje ostrzeżony za pomocą powiadomienia przed wykonaniem operacji.
- Użytkownik anuluje lub potwierdza wykonanie transakcji.

5 Diagram UML



Rysunek 1: Diagram UML

6 Schemat bazy danych



Rysunek 2: Prototypowy schemat bazy danych

7 Specyfikacja zewnętrzna

7.1 Rejestracja

Użytkownik, chcąc skorzystać z aplikacji, musi posiadać własny profil. Formularz rejestracyjny pozwala założyć konto. Celem założenia konta użytkownik musi podać:

- nazwę użytkownika
- adres e-mail
- hasło
- kwotę początkową.

Naciśnięcie przycisku "DODAJ KONTO" skutkuje dodaniem konta do bazy danych.

7.2 Logowanie

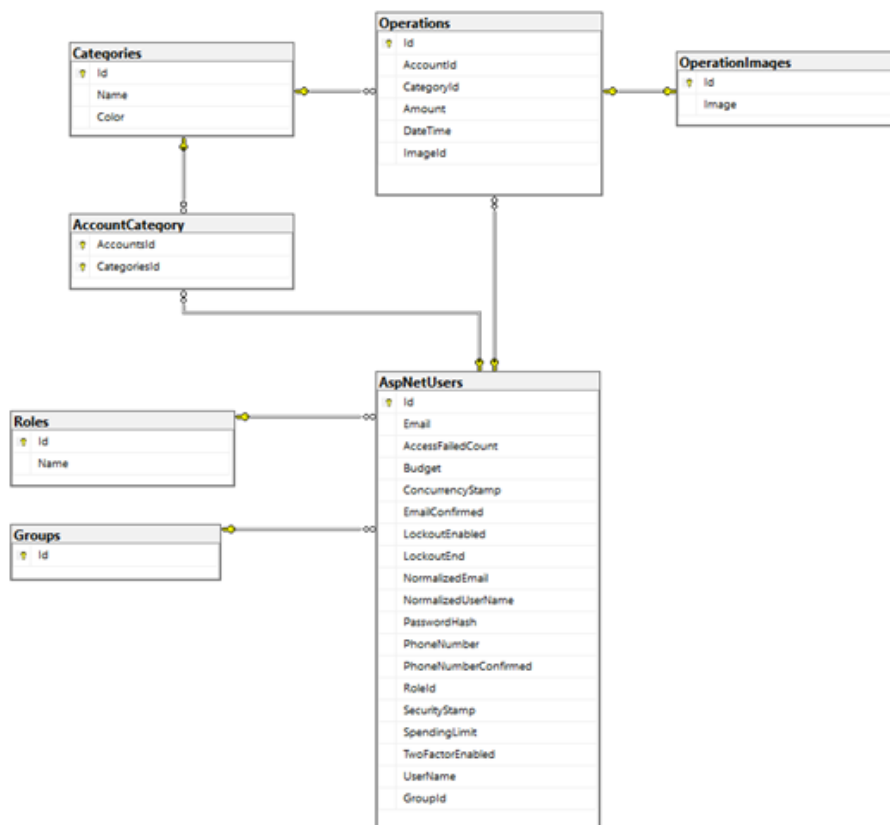
Strona logowania zawiera formularz, w którym należy wprowadzić Login użytkownika oraz hasło. Po wprowadzeniu poprawnych danych oraz naciśnięciu przycisku "LOGIN" użytkownik zostaje przeniesiony do ekranu głównego aplikacji. Jeżeli użytkownik nie posiada konta, może w prosty sposób przejść do formularza rejestracji, klikając w odnośnik "Nie mam jeszcze konta".

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