Przetwarzanie rozproszone w SQL Server

Joanna Marcinkowska

Kraków, 2020

Spis treści

1. Opis problemu

Celem projektu było utworzenie aplikacji, która wykorzystuje przetwarzanie rozproszone w SQL Server. Aplikacja ma umożliwiać przeprowadzenie transakcji rozproszonej, która zostaje zakończona akceptacją albo odrzuceniem. Transakcje mogą być wykonywane między różnymi oddziałami banków.

2. Opis funkcjonalności udostępnianej przez API

Poruszanie się po aplikacji konsolowej odbywa się przy pomocy wyboru odpowiednich klawiszy w menu. Funkcjonalności jakie oferuje to wyszukiwanie i rejestrowanie klientów, wpłacanie i wypłacanie pieniędzy, wykonywanie przelewów oraz wyszukiwanie transakcji.

2.1 Menu główne

W menu głównym jest możliwość wybrania jednej z kilku opcji:

- 1 – znajdowanie klienta po numerze PESEL;
- 2 – rejestrowanie klienta;
- 3 – wypłacanie pieniędzy z banku;
- 4 – wpłacanie pieniędzy na konto;
- 5 – wykonanie przelewu;
- 6 – znajdowanie transakcji dla danego klienta;
- 0 – wyjście z programu.

2.2. Znajdowanie klienta

Po wybraniu tej opcji pojawi się pole w które należy wpisać numer PESEL w celu wyświetlenia informacji o kliencie. Jeżeli numer PESEL będzie niepoprawny, pojawi się błąd. W przeciwnym przypadku zostaną wyświetlone informacje o kliencie tj.: ID, imię, nazwisko, Miasto, numer PESEL, saldo oraz nazwa oddziału do którego należy klient.

2.3 Rejestrowanie klienta

Istnieje możliwość dodania nowego klienta do wybranego przez niego oddziału. Należy wprowadzić imię, nazwisko, numer PESEL, miasto oraz kwotę pieniędzy, która zostanie wpłacona na konto bankowe. Następnie należy wybrać oddział w banku do kutego klient ma zostać przypisany. Dostępne opcje:

- 1 – Oddział w Krakowie;
- 2 – Oddział w Warszawie.

W przypadku wprowadzenia błędnych danych, pojawi się odpowiedni komunikat. Po pomyślnym zarejestrowaniu klienta, zostanie wyświetlony unikalny numer ID.

2.4 Wypłacanie określonej kwoty pieniędzy

Podczas wypłacania pieniędzy należy podać ID osoby wypłacającej oraz kwotę do wypłacenia. Jeżeli wprowadzony numer ID lub wpłacana wypłacana kwota będzie przewyższała saldo dostępne na koncie, pojawi się odpowiednia informacja.

2.5 Wpłacenie określonej kwoty pieniędzy

Aby wpłacić podaną kwotę pieniędzy należy podać numer ID klienta, na którego konto ma zostać wpłacona kwota, oraz kwotę.

2.6 Wykonanie przelewu

Dokonując przelewu między dwoma kontami bankowymi należy podać ID osoby wysyłającej, ID osoby odbierającej przelew oraz kwotę pieniędzy do wysłania. Przelew można wykonywać między kontami w tym samym lub innymi oddziałami banku. Jeżeli wprowadzone informacje będą poprawne, transakcja zostanie zakończona powodzeniem, w przeciwnym razie pojawi się odpowiednia informacja o błędzie.

2.7 Wyświetlanie transakcji

Po wybraniu opcji wyświetlenia transakcji, należy podać ID klienta, dla którego mają zostać wyświetlone transakcje. Jeżeli wprowadzone dane są prawidłowe, zostaną wyświetlone informacje o wszystkich transakcjach, które zostały wykonane na koncie bankowym klienta. Wyświetlone informacje to: kwota, nadawca, odbiorca, data przelewu oraz typ przelewu.

3. Opis typów danych oraz metod udostępnianych w ramach API i opis implementacji

Program składa się z głównej klasy *Program,* klasy pomocniczej *Menu*, która pozwala na wyświetlanie danych, oraz klas *MainBank, Department, Client, DatabaseConnection.*

3.1 Menu

3.1.1 Metody klasy

- FindClient – pobiera informacje od użytkownika, następnie wypisuje dane klienta
- send – pobiera dane potrzebne do wykonania przelewu, następnie wypisuje informacje powodzeniu lub niepowodzeniu transakcji
- withdraw – służy pobieraniu danych potrzebnych do wypłacenia danej kwoty pieniędzy
- deposit – pobiera informacje od użytkownika, potrzebne do wpłacenia zadanej kwoty pieniędzy na konto bankowe
- register – po pobraniu danych z konsoli, wyświetla komunikat o powodzeniu bądź niepowodzeniu przy rejestrowaniu klienta
- showTransaction – pobiera dane od użytkownika, następnie wyświetla informacje o transakcji

3.2 MainBank

Klasa realizuje podstawowe funkcjonalności w centralnym banku. Wykonywane są w niej transakcje rozproszone w metodach które polegają na przesyłaniu danych między wieloma bazami.

3.2.1 Pola klasy

- databaseName – przechowuje nazwę bazy centralnego banku

3.2.2 Metody klasy

- searchClient – zwraca listę z danymi klienta
- checkClientID – sprawdza czy klient o podanym ID istnieje w bazie
- sendMoney – wykonuje operację wysyłania przelewu między różnymi kontami bankowymi
- getDepartment – zwraca oddział banku do którego należy klient
- checkDepartment – sprawdza ilość oddziałów do których należy klient
- getOtherDepartment – znajduje drugi oddział dla klienta
- withdrawMoney – wykonuje transakcję wypłacenia danej kwoty pieniędzy
- depositMoney – wykonuje transakcję wpłacenia danej kwoty pieniędzy
- getDepartmentID – znajduje ID danego oddziału
- registerClient – zapisuje klienta do bazy
- getTransaction – wyszukuje transakcji dla danego klienta
- checkPesel – sprawdza poprawność wprowadzonego numeru PESEL
- randomID – zwraca losowy numer ID
- departmentID – zwraca ID i nazwę oddziału

3.3 Department

Klasa pomagająca w realizacji transakcji rozproszonych w zależności od wybranego oddziału.

3.3.1 Metody klasy

- sendMoneyDepartment – wykonuje przesłanie pieniędzy dla danego oddziału
- sendSameDepartment – wykonuje przesłanie pieniędzy, gdy oddział nadawcy i odbiorcy jest taki sam
- saveTransaction – zapisuje dane transakcji
- transaction – zapisuje informacje o transakcji w bazie
- setSender – aktualizuje stan konta osoby wysyłającej przelew
- setReciver - aktualizuje stan konta osoby odbierającej przelew
- withdrawMoneyDepartment – wykonuje wypłatę pieniędzy dla podanego klienta
- depositMoneyDepartment – wykonuje wpłatę pieniędzy dla podanego klienta
- registerClient – rejestruje klienta w bazie
- getTransaction – pobiera informacje o wykonanych transakcjach

3.4 Client

Klasa przechowująca dane o klientach.

3.4.1 Pola klasy

- ID – przechowuje ID klienta
- Name – przechowuje imię klienta
- Surname – przechowuje nazwisko klienta
- City – przechowuje miasto klienta
- Balance – przechowuje informacje o saldzie klienta
- PESEL – przechowuje informacje o numerze PESEL klienta

3.5 DatabaseConnection

Klasa odpowiedzialna za łączenie się z bazą.

3.5.1 Pola klasy

- mainConnection – przechowuje informacje potrzebne do połączenia z bazą danych

3.5.2 Metody klasy

- connectToDatabase – wykonuje łączenie z bazą w zależności od nazwy bazy

4. Testy jednostkowe

Zostało wykonane 15 testów jednostkowych sprawdzających poprawność utworzonych metod. Dla niektórych funkcji wykonanie testów nie było możliwe, ponieważ ich działanie polegało na wypisywaniu danych. Utworzone testy są dołączone w folderze *Tests*.

5. Podsumowanie i wnioski

Aplikacja została utworzona w celu ułatwienia wykonywania operacji między wieloma bazami danych. Do tego celu zostały wykorzystane transakcje rozproszone, które umożliwiają wykonywanie działań na kilku zasobach. Transakcja może zostać zakończona zatwierdzeniem albo odrzuceniem. Aby transakcja mogła zostać zaakceptowana, każdy z uczestników musi zagwarantować, że zmiana danych będzie trwała. Jeżeli jeden uczestnik transakcji nie będzie w stanie zagwarantować trwałości, transakcja zakończy się odrzuceniem i wszystkie zmiany zostaną wycofane.

Zrealizowanie transakcji było możliwe dzięki strukturze *System.Transactions*, która jest w pełni zintegrowana z ADO.NET, oraz dzięki klasie *TransactionScope*. Użyta została metoda *Complete,* która jest wywoływana tylko wtedy gdy w transakcji nie zostały rzucone wyjątki.

6. Literatura

- https://docs.microsoft.com/pl-pl/dotnet/framework/data/adonet/system-transactions-integration-with-sql-server
- https://newton.fis.agh.edu.pl/~antek/index.php?sub=db2_doc
- https://docs.microsoft.com/pl-pl/dotnet/framework/data/adonet/distributed-transactions