Adam Kulasiak, Patrycja Wiater

Currency Bank

Bank walutowy

Spis treści

[Opis ogólny 2](#_Toc62914731)

[Stack technologiczny 2](#_Toc62914732)

[Login window: 3](#_Toc62914733)

[Register window: 3](#_Toc62914734)

[Main window: 4](#_Toc62914735)

[Account Opening Window: 5](#_Toc62914736)

[New Transfer Window: 5](#_Toc62914737)

[Exchange Window: 6](#_Toc62914738)

[Inne funkcjonalności: 6](#_Toc62914739)

[Instalacja aplikacji: 6](#_Toc62914740)

[WAŻNE INFORMACJE 10](#_Toc62914741)

[Bibliografia 11](#_Toc62914742)

# Opis ogólny

Aplikacja Currency Bank – Bank Walutowy to narzędzie służące głównie do przechowywania pieniędzy w formie elektronicznej. Mogą być one przechowywane w różnych walutach dzięki automatycznemu przewalutowaniu. Aplikacja umożliwia wypłatę oraz wpłatę wybranej kwoty, przelewy zwykłe oraz międzywalutowe, posiadanie kilku kont w różnych walutach, przelewy wewnętrzne służące zmianie waluty. Konta można bez ograniczeń tworzyć i usuwać. Aplikacja szczególnie może spodobać się osobom które dużo podróżują bądź kupują produkty od lokalnych wytwórców na zagranicznych stronach internetowych. Dzięki zastosowaniu dwujęzycznych tłumaczeń z aplikacji mogą korzystać klienci z różnych części świata.

Poniższy opis podzielony zostanie pod względem funkcjonalności a także będzie zawierał informacje dotyczące strony biznesowej oraz deweloperskiej.

# Stack technologiczny

Aplikacja napisana jest w .NET Core 3.1, jedynie dostarczony instalator napisany jest w .NET Framework 4.7.2 z kilku powodów:

- W instalatorze jest funkcja sprawdzająca czy mamy zainstalowany .NET Core oraz pozwalająca na jego instalację.

- w .NET Core był problem z działaniem Web Forms

Aplikacja korzysta z bazy danych SQLite. Do połączenia z nią od strony C# używany jest Entity Framework Core i podejście Code First.

Komunikacja miedzy aplikacja WPF a backendem odbywa się poprzez API Controllers. Dane przesyłane są w formacie JSON, wykorzystywane są obiekty DTO, aplikacja WPF wykonuje HTTP requesty do pobierania oraz wysyłania danych. Rozdzielona jest wersja kontrolerów od warstwy logiki. W całkiem osobnym projekcie znajdują się rzeczy związane z modelami i bazą danych.

Do mapowania obiektów (np. pomiędzy DTO oraz modelami) użyty jest Auto Mapper.

W osobnym projekcie „Commons” dostępnych jest kilka metod wykorzystywanych w różnych częściach aplikacji. Ten projekt ma dołączone również testy jednostkowe.

Hasła w bazie danych są zahaszowane oraz posolone (wykorzystany jest algorytm dostępny w System.Security.Cryptography.HMACSHA512).

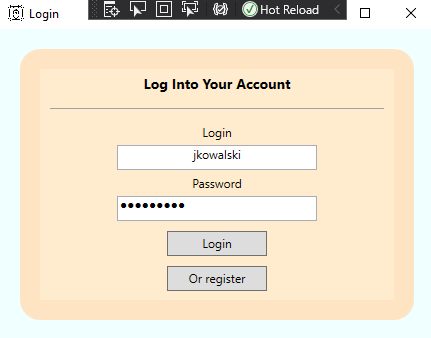
Do uwierzytelniania wykorzystane są JSON Web Tokens. Aplikacja w momencie logowania (jeżeli login oraz hasło są poprawne) zwraca do klienta token dostępu wraz z informacjami jakiego użytkownika dotyczy (te informacje zakodowane są w tokenie). Każde żądanie HTTP wymaga aby w nagłówku był dołączony Bearer Token. Jeżeli go nie będzie, serwer zwroci komunikat „Unauthorized” z racji ze do kontrolerów maja dostęp tylko zalogowani użytkownicy.

Aplikacja wykorzystuje zewnętrzne API do wymiany walut dostępne pod adresem <http://api.exchangeratesapi.io/latest/>. Jeżeli klient chce wymienić walutę, aplikacja sprawdza w lokalnej bazie danych czy jest dostępny kurs wymiany dla wybranych walut z danego dnia. Jeżeli jest to zwraca ten z lokalnej bazy, jeżeli nie ma to pobiera z zewnętrznego API, zapisuje w lokalnej bazie i zwraca do klienta.

Po stronie klienta (WPF) zostały wyodrębnione serwisy, które udostępniają metody do obsługi zadań HTTP. Są one wykorzystywane w komponentach. Każde okienko odpowiedzialne jest za inną funkcjonalność. Opis okienek dalej w pliku. Aplikacja dostępna jest w wersji polskiej oraz angielskiej. Wykorzystywane są pliki zasobów w których są słowniki: polski i angielski. W instalatorze możemy ustawić sobie język (opisane w sekcji „Instalacja aplikacji”). Aplikacja na głównym wątku ustawia Culture.Info i korzysta ze słownika odpowiedniego dla danego języka.

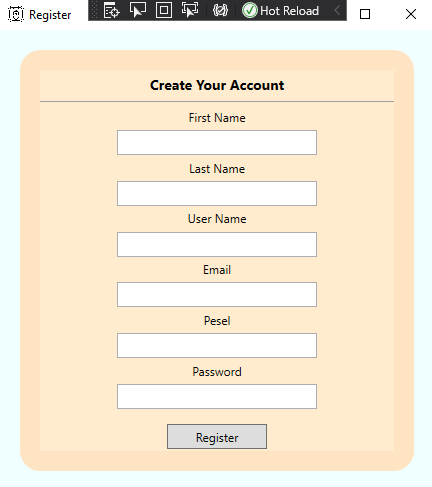
# Login window:

Okienkiem startowym aplikacji jest okienko logowania z którego w razie braku posiadania konta można przejść do rejestracji. Logowanie odbywa się poprzez podanie loginu oraz hasła ustalonego podczas rejestracji. Zalogowany użytkownik uzyskuje dostęp do głównego okna z przeglądem swoich kont.



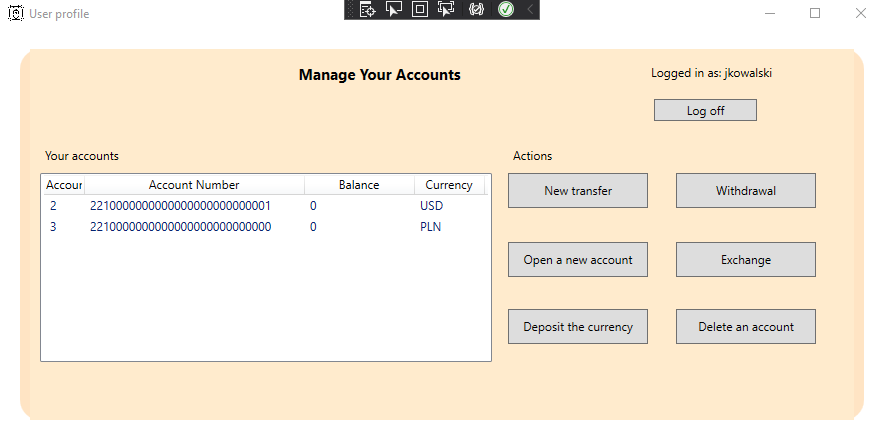
# Register window:

Okienko służące do rejestracji nowego klienta banku. Aby założyć konto należy uprzednio zarejestrować się w aplikacji. Dane potrzebne do rejestracji to - oprócz wspomnianego wyżej loginu i hasła - imię, nazwisko, adres e-mail, pesel. Przy rejestracji system automatycznie zapisuje również datę rejestracji wraz z id nowego klienta.



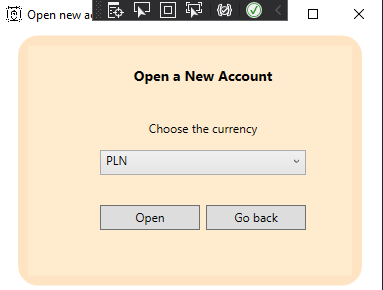
# Main window:

Główne okienko aplikacji to przegląd funkcjonalności oraz posiadanych kont. Poszczególne przyciski pozwalają na skorzystanie z operacji jakie oferuje bank. Tabela umieszczona po lewej stronie okna przedstawia dane o posiadanych kontach: id, numer, balans oraz walutę.



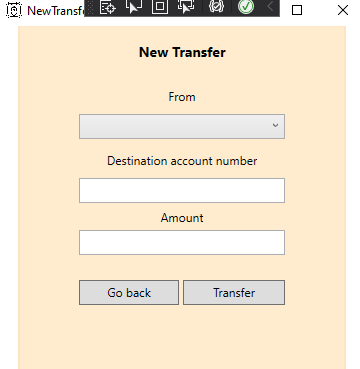
# Account Opening Window:

Okno służące obsłudze akcji dodawania nowego konta. Wymagane dane to wyłącznie waluta w jakiej klient chce otworzyć nowe konto. Okienko posiada przycisk umożliwiający anulację operacji. Pozostałe dane przekazywane są automatycznie na podstawie zalogowanego klienta.



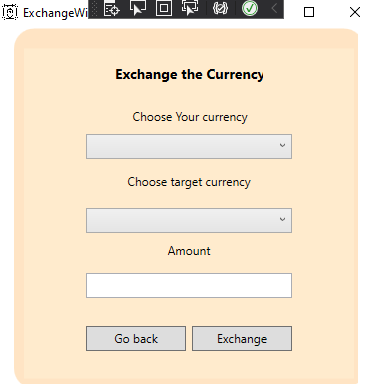
# New Transfer Window:

Okno służące obsłudze przelewów. Lista rozwijana umożliwia wybór konta z którego klient chce wykonać przelew. Standardowo należy podać numer konta odbiorcy przelewu oraz kwotę. Jeżeli konto odbiorcy jest w innej walucie niż nadawcy, kwota zostaje automatycznie przewalutowana na podstawie wartości pobieranej poprzez WebAPI z oficjalnej strony Narodowego Banku Polskiego.



# Exchange Window:

Okno służące obsłudze przewalutowania kwoty przez danego klienta. Do wykonania tej akcji klient musi posiadać co najmniej dwa konta – konto z którego zamierza przewalutować daną kwotę oraz konto w walucie docelowej. Listy rozwijane zapobiegają błędom polegającym na pomyłkach przy podawaniu kont. Standardowo zastosowano również przycisk umożliwiający opuszczenie okienka skutkujące anulacją operacji.



# Inne funkcjonalności:

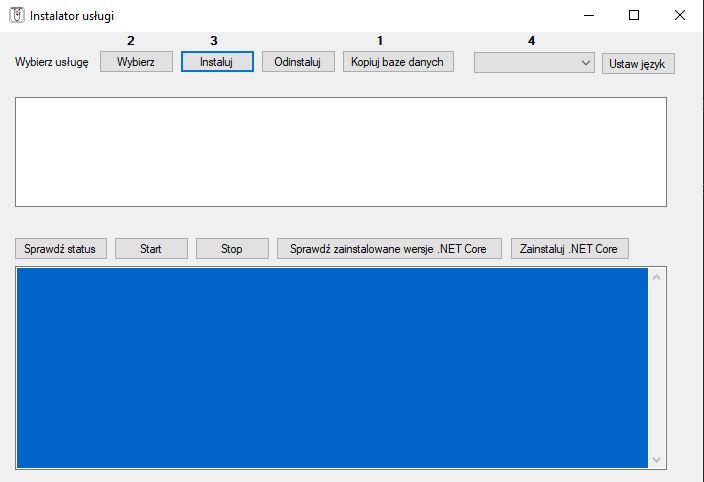
- usunięcie konta – system automatycznie blokuje próby usunięcia konta którego balans nie wynosi 0,

- wpłata – możliwość zasilenia dowolnego konta poprzez podanie kwoty oraz wybór ID z listy rozwijanej,

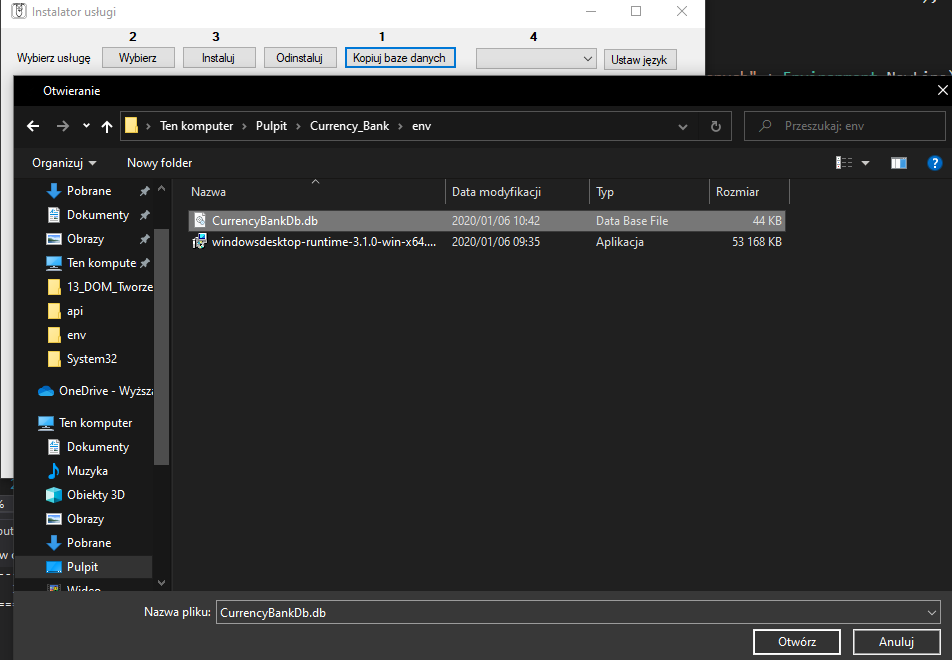
- wypłata – możliwość wypłaty dowolnej kwoty (w granicach posiadanego balansu na danym koncie) z dowolnego konta wybieranego za pomocą listy rozwijanej pobierającej dane kont zalogowanego klienta.

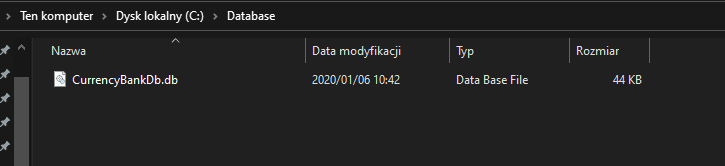
# Instalacja aplikacji:

Do aplikacji dołączony jest instalator do części backendowej oraz przygotowujący bazę danych napisany własnoręcznie. Jest to w dużej mierze nakładka na wiersz poleceń i PowerShell jednak poprawia wygodę korzystania.

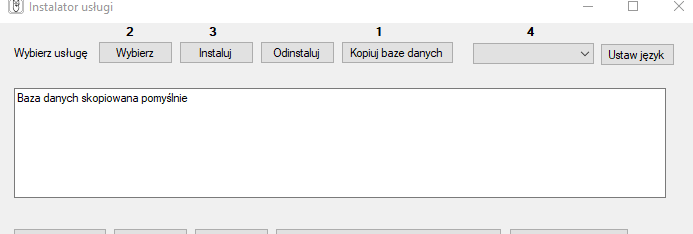


1. Po kliknięciu przycisku „Kopiuj bazę danych” otworzy się okienko dialogowe. Wybieramy zaczynając od głównego folderu projektu: env\CurrencyBankDb.db. Po wybraniu pliku i kliknięciu otwórz, na dysku C zostanie utworzony folder „Database” do którego zostanie skopiowana dostarczona baza danych

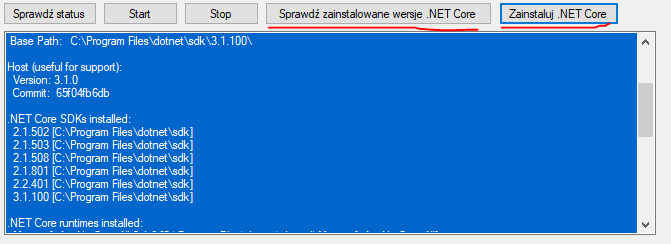


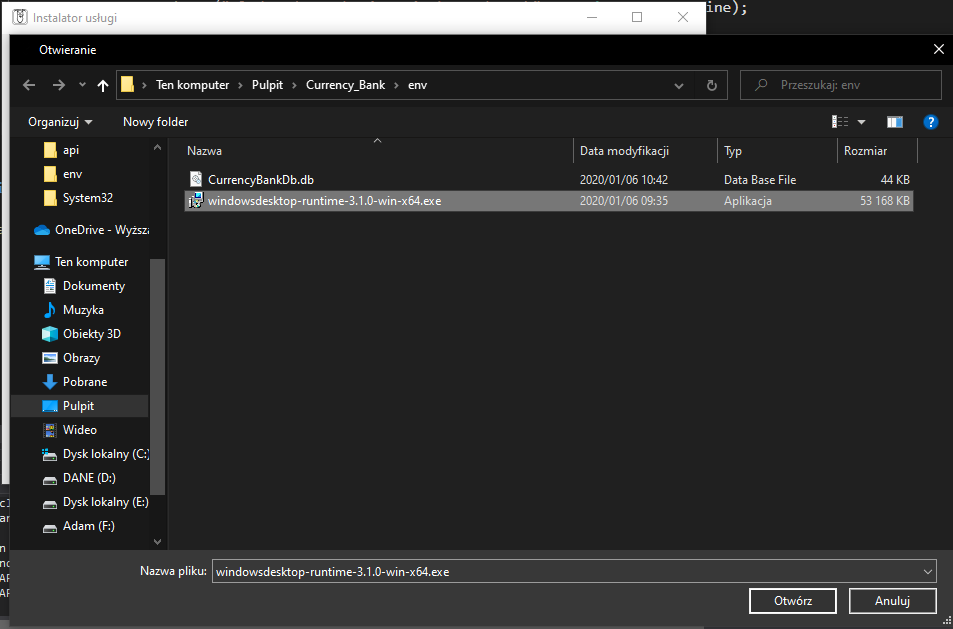


Jeżeli operacja się udała w oknie instalatora zobaczymy komunikat: „Baza danych skopiowana pomyślnie”.

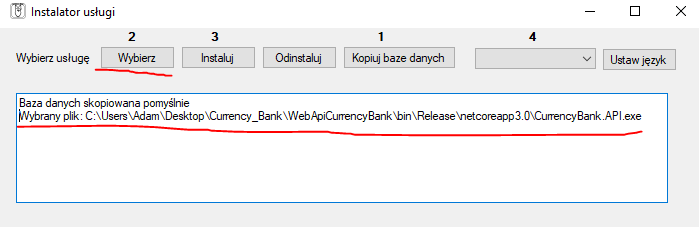


1. Instalator dostarcza funkcję, która sprawdza czy mamy zainstalowany .NET Core. Możemy to zrobić klikając na przycisk „Sprawdź zainstalowane wersje .NET Core”. W niebieskim polu zobaczymy listę zainstalowanych wersji (o ile jakieś są). Jeżeli nie ma, klikamy w przycisk „Zainstaluj .NET Core”, otworzy się okienko dialogowe, w katalogu env (tym samym co w punkcie 1) załączona jest instalka do wersji 3.1.0, możemy ją wybrać i zainstalować.





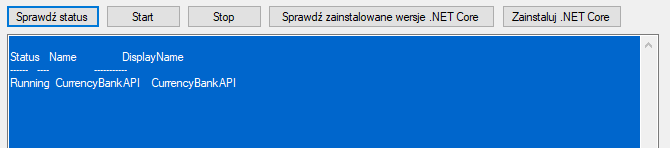
1. Następnie można przystąpić do instalacji programu, który działa w tle jako usługa systemu Windows. Aplikacje możemy również uruchomić z pliku exe klikając na niego dwukrotnie, wówczas odpali nam się jako aplikacja konsolowa. Aby zainstalować jako usługę systemu Windows klikamy na przycisk „Wybierz”, przechodzimy do ścieżki „Currency\_Bank\WebApiCurrencyBank\bin\Release\netcoreapp3.0” i wybieramy plik o nazwie CurrencyBank.API.exe. Klikamy „Otwórz”. Dostaniemy komunikat jaki plik wybraliśmy

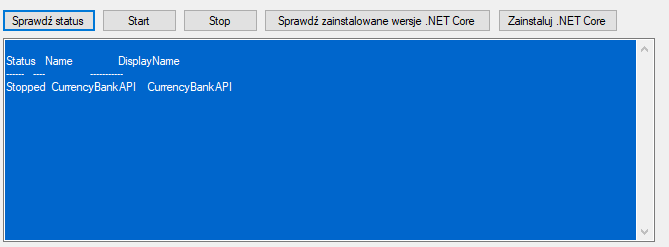


Następnie klikamy „Instaluj”, wyskoczy nam okienko z pytaniem czy chcemy zezwolić na uruchomienie aplikacji, zatwierdzamy (cmd wymaga uprawnień administratora aby utworzyć i zainstalować usługę).

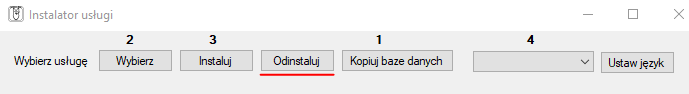
1. Mamy 3 przyciski do zarzadzania usługą: Sprawdź status, który sprawdza czy usługa jest zainstalowana oraz jaki ma stan: „Running” albo „Stopped”. Start, który uruchamia usługę oraz Stop, który ją zatrzymuje. Po kliknięciu w wyżej wymienione przyciski dostaniemy komunikat w niebiskim oknie instalatora.







1. Po zakończeniu korzystania z aplikacji możemy łatwo odinstalować usługę klikając w przycisk „Odinstaluj”. Usługa zostanie usunięta z systemu.



1. Z poziomu instalatora możemy również ustawić język aplikacji: polski lub angielski. Wybieramy z listy rozwijanej i klikamy przycisk „Ustaw język”. W folderze C:\Database zostanie utworzony plik lang.txt z którego korzysta aplikacja WPF.

# WAŻNE INFORMACJE

Aplikacja działa na porcie 5000, do poprawnego działania aplikacji wymagane jest aby port ten był nieużywany. Aplikacja przechowuje pliki niezbędne do działania w folderach C:\Database oraz C:\temp (skrypty .bat które instalator tymczasowo tworzy i później wykonuje). Po zakończeniu oceniania projektu, proszę odinstalować usługę (punkt 5), można tez usunąć foldery C:\Database oraz C:\temp.

# Bibliografia

* <https://docs.microsoft.com/pl-pl/aspnet/core/?view=aspnetcore-3.1>
* <https://docs.microsoft.com/pl-pl/dotnet/framework/>
* <https://sqlite.org/docs.html>
* <https://docs.automapper.org/en/stable/Getting-started.html>
* <https://jwt.io/introduction>
* <http://api.exchangeratesapi.io/latest/>