

Dokumentacja projektowa

Temat projektu:

Aplikacja do zarządzania wydatkami i
przypomnieniami o samochodzie

Skład sekcji:

Mirosław Ściebura

Temat projektu:

Tematem projektu było stworzenie aplikacji w środowisku .NET, będącej klientem systemu przechowującego informacje o samochodach oraz wydatkach związanych z nimi.

Wymagania:

Aplikacja jest częścią „frontendową” systemu. Backendem tutaj jest serwer z którym komunikuje aplikacja kliencka, będąca aplikacją okienkową systemu Windows. Aplikacja kliencka korzysta z referencji do metod zewnętrznych, które realizują miały logikę systemu (odwołanie do elementów przechowywanych w bazie danych).

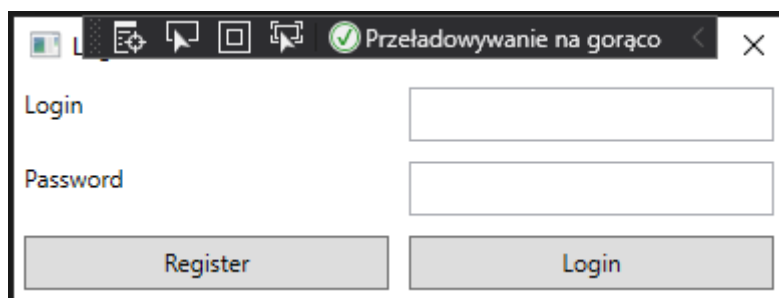
Technologie:

Aplikacja została utworzona w oparciu o .NET Core. Korzystanie z systemu odbywa się tutaj z użyciem wystawionego API (odwołanie do konkretnych metod np. /Car/Read, /Reminder/Update), które zwraca informacje w postaci JSON. Aplikacja jest utworzona w oparciu o WPF. Powiązanie kontrolki z konkretnymi obiektami pobranymi poprzez API było oparte o bindowanie danych z użyciem ItemsSource oraz DataContext wraz z podaniem w pliku XAML odpowiednich nazw właściwości. Do przesyłu danych wykorzystałem również pakiet Newtonsoft.Json, który pozwolił mi na przesył, serializację i deserializację obiektów w postaci JSON.

Działanie aplikacji:

Po uruchomieniu pojawia się ekran logowania. Po wpisaniu loginu i odpowiedniego hasła możemy zalogować się do aplikacji. W przypadku błędu wyświetlany jest odpowiedni komunikat w postaci MessageBox. Poprzez przycisk „Register” można zarejestrować nowego użytkownika. Po zalogowaniu następuje przejście do głównego okna aplikacji, które podzielone jest na trzy zakładki: „Cars”, „Expenses” i „Reminders”. Każda z nich jest odpowiedzialna za odczyt, aktualizację, tworzenie nowych oraz usuwanie informacji. Wydatki i przypomnienia są powiązane z konkretnym samochodem, dlatego aby je utworzyć na karcie „Cars” musi być wybrany konkretny samochód. W innym przypadku przyciski „Create”, „Update” i „Delete” są wyłączane. Dla samochodów waliduje się czy „First Registration Date” jest wcześniej niż „Purchase Date”, Czy aktualny przebieg nie jest mniejszy od przebiegu w dniu zakupu, czy daty mają poprawny format (yyyy-MM-dd) oraz czy przebiegi nie są ujemne. Dla „Reminders” czy data jest poprawna, czy nie jest z przeszłości oraz czy przebieg nie jest mniejszy od aktualnego przebiegu samochodu. Dla „Expenses” walidowana jest data, czy data nie jest z przyszłości, czy przebieg nie jest mniejszy od przebiegu przy zakupie, czy cena nie jest ujemna oraz czy wszystko oprócz „Place” i „Info” jest wypełnione. W przypadku „Car” wszystkie pola są obowiązkowe, natomiast w przypadku „Reminders” jedynie „Info” (oraz „IsChecked”, jednak to pole jest CheckBoxem).

Zrzuty ekranu:



MainWindow Przeładowywanie na gorąco

Cars Expenses Reminders

Actual id 3

Manufacturer Peugeot

Model 206

First Registration Date 2000-06-28T00:00:00

Actual Mileage 88000

Vin YV1LW7902Y2696846

Purchase Date 2006-05-15T00:00:00

Purchase Mileage 30000

Licence Plate SO5336G

3

Create

Update Delete

MainWindow Przeładowywanie na gorąco

Cars Expenses Reminders

Actual id 6

Date 2020-09-08T00:00:00

Place BP

Mileage 85000

Price 40.98

Info Benzyna gaz

Type info FUEL

6

Create

Update Delete

MainWindow

Przeładowywanie na gorąco

Cars

Expenses

Reminders

Actual id

6

Date

2022-03-04T00:00:00

Milleage

135000

Info

Carefully

Is checked

☐ Is checked

6

Create

Update

Delete