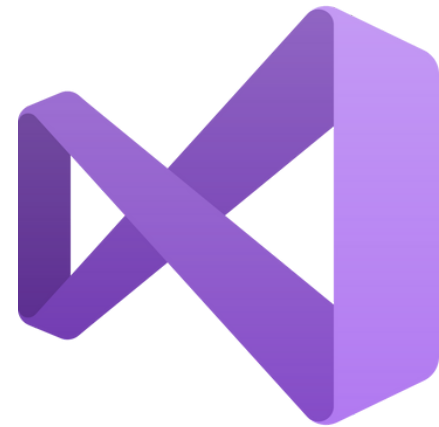


Czyli dlaczego aplikacja bankowa na komputer nie ma szans wypalić?

# Aplikacja bankowa



Autor: Wiktor Szyszka



- Windows Forms – Visual Studio
- Microsoft SQL Server Management Studio



# Użyte technologie

# Opis projektu

Poszczególne sektory projektu



## EKRAN UŻYTKOWNIKA

Ekran wyświetlający obecny stan konta, podstawowe informacje o użytkowniku. Możliwość wykonywania przelewów oraz opcja wylogowania – zamknięcia sesji.



## EKRAN WERYFIKACJI

Weryfikacja sprawdzająca, czy osoba próbująca użyć aplikacji bankowej nie jest robotem. Proste zadanie na zapamiętywanie.



## EKRAN REJESTRACJI

Ekran umożliwiający dodanie nowego użytkownika poprzez wprowadzenie odpowiednich danych.

# Zabezpieczenia użytkowe

## Jak uczynić aplikację przystępną dla każdego?

Najistotniejszą kwestią podczas tworzenia aplikacji użytkowej jest odpowiednie jej przemyślenie, aby jej obsługa nie wymagała zaawansowanej wiedzy. W pewnym sensie to aplikacja powinna prowadzić użytkownika za przysłowiową "rączkę", tak aby ten tego nie odczuł i tak, aby uniknąć popełnienia jakiegokolwiek błędu.



# Sprawdzanie, sprawdzanie jeszcze raz sprawdzanie!

W aplikacji zaimplementowałem warunki konieczne bez spełnienia których użytkownik nie będzie w stanie przejść do dalszego korzystania z aplikacji. Systemami zabezpieczającymi są m.in. sprawdzenie poprawności wprowadzonego hasła w formularzu rejestracji, ograniczenie długości możliwego do wpisania PESELu do 11 cyfr itp.





```
//Warunek - wszystkie pola uzupełnione
if ((name->Length == 0) || (surname->Length == 0) || (login->Length == 0)
    || (password->Length == 0) || (CPassword->Length == 0) || (PESEL->Length == 0)) {
    MessageBox::Show("Proszę wypełnić wszystkie pola",
        "Jedno lub więcej pól pozostały niewypełnione", MessageBoxButtons::OK, MessageBoxIcon::Exclamation);

    return;
}
```

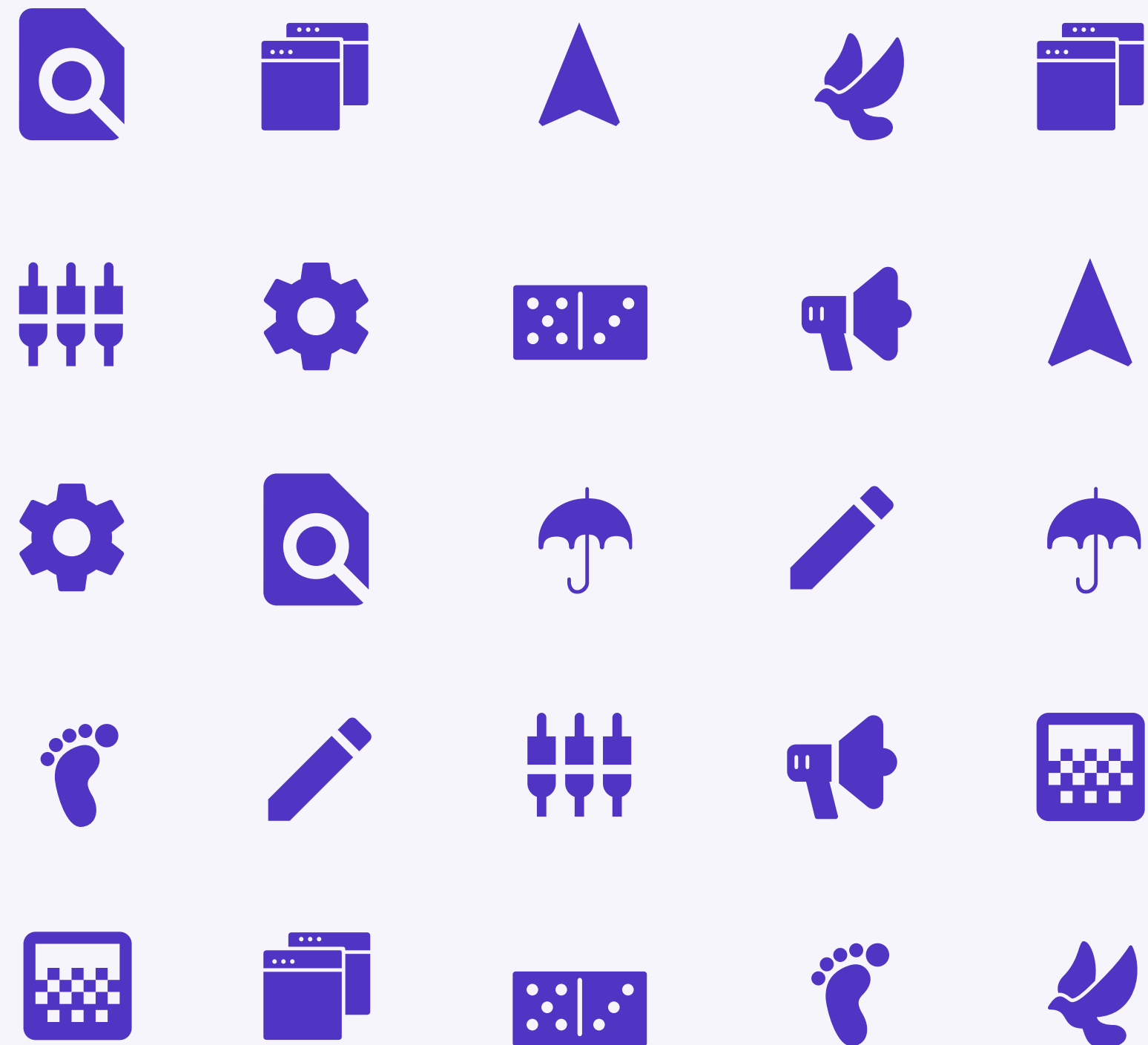
```
//Warunek - identyczne hasła
if (String::Compare(password, CPassword) != 0) {
    MessageBox::Show("Wprowadzono błędne hasła!",
        "Błąd zgodności", MessageBoxButtons::OK, MessageBoxIcon::Warning);

    return;
}
```



# Memory game czyli Captcha 2.0

Dostęp do aplikacji bankowej zabezpieczyłem prostą "minigierką". Gra polega na sparowaniu 8 par tych samych symboli – każda para to 2 te same symbole. Postanowiłem użyć dość niecodziennego rozwiązania, jakim było zastąpienie grafik symboli pojedynczymi literami zapisanymi za pomocą czcionki – Webdings.



## Baza danych

### Największy przyjaciel a może największy wróg?

Zdecydowanie baza danych jest rzeczą, która przysporzył mi najwięcej problemów.

Problemy nie wynikały z złożoności jej działania w kodzie, czy z problemów z utworzeniem odpowiednich tabel. Problem

było zainstalowanie silnika bazy danych.

Winowajcą okazał się system Windows 11, który niestety zapłacił najwyższą cenę próby instalacji silnika. Moje próby naprawienia błędu skończyły się zepsuciem swojego systemu operacyjnego i potrzebą zainstalowania go na nowo.





Czy macie  
jakieś pytania?

