



Wyższa Szkoła Informatyki i Zarządzania
Kolegium Informatyki Stosowanej, Informatyka
Konrad Boroń, w65528

Projekt Szkolenie techniczne 1

Rzeszów, 29.06.2023

1.Opis tematu

Aplikacja Car Dealer jest aplikacją desktopową stworzoną dla salonów samochodowych w celu zarządzania sprzedażą samochodów oraz obsługą transakcji gotówkowych. Aplikacja umożliwia dodawanie, edytowanie i usuwanie informacji o samochodach, klientach, pracownikach, oraz obsługę transakcji gotówkowych.

Link do repo na GitHub: <https://github.com/zlplayer/CarDealer>

2. Wykorzystane technologie

Do wykonania projektu został zainstalowany pakiet Guna.UI2.WinForms jest to pakiet który udostępnia elementy do stworzenia aplikacji desktopowej. Dzięki którym można stworzyć nowoczesny wygląd aplikacji.

3. Wymagania systemowe

- System operacyjny: Windows 10 lub nowszy
- Microsoft .NET Framework 4.8 lub nowszy
- SQL Server (baza danych)

4. Funkcjonalności

1. Zarządzanie samochodami:

- Dodawanie nowych samochodów do bazy danych.
- Edytowanie informacji o istniejących samochodach
- Usuwanie samochodów z bazy danych.

2. Zarządzanie klientami:

- Dodawanie nowych klientów do bazy danych
- Edytowanie informacji o istniejących klientach.
- Usuwanie klientów z bazy danych.

3. Zarządzanie pracownikami:

- Dodawanie nowych pracowników do bazy danych
- Edytowanie informacji o istniejących pracownikach.
- Usuwanie pracowników z bazy danych.

4. Obsługa transakcji gotówkowych:


- Wybieranie samochodu i klienta dla transakcji.
- Dodawanie wybranych samochodów do transakcji gotówkowej.
- Wyświetlanie podsumowania transakcji i obliczanie całkowitej kwoty.
- Potwierdzanie transakcji gotówkowej i aktualizowanie stanu samochodów w bazie danych.


5. Wyszukiwanie:

- Wyszukiwanie samochodów, klientów i pracowników na podstawie różnych kryteriów


5. Baza danych

Aplikacja wykorzystuje bazę danych SQL Server do przechowywania informacji o samochodach, klientach, pracownikach i transakcjach gotówkowych. Baza danych składa się z kilku tabel, takich jak: tbCash, tbCustomer, tbProduct, tbUser. Dane są przechowywane w normalizowanej strukturze, umożliwiającą efektywne zarządzanie informacjami.

tbUser	
	id
	name
	address
	phone
	role
	dob
	password

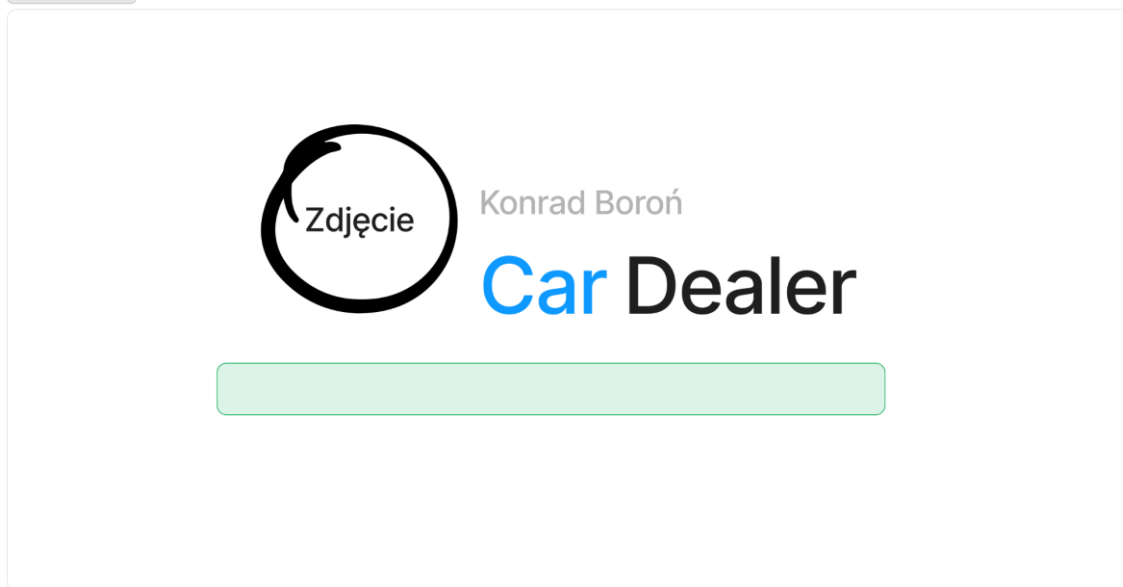
tbProduct	
	pcode
	pname
	ptype
	pcategory
	pqty
	pprice

tbCustomer	
	id
	name
	address
	phone

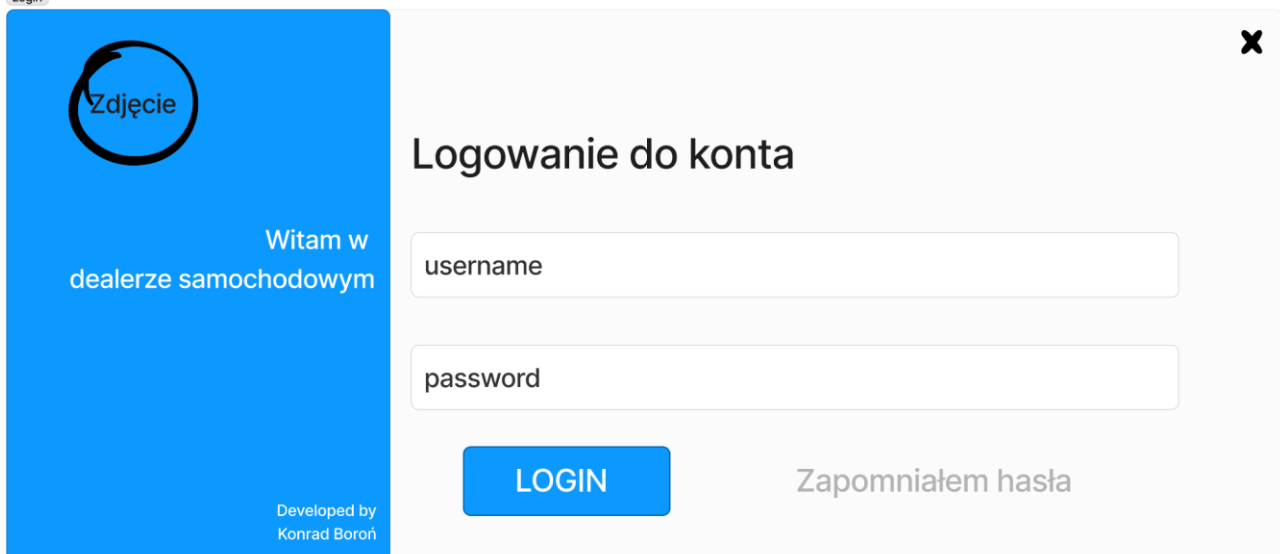
tbCash	
	cashid
	transno
	pcode
	pname
	qty
	price
	total
	cid
	cashier

6. Projekt GUI

Ekran ładowania



Login



Witam w dealerze samochodowym



zdz.profil

Panel

Klienci

Użytkownicy

Samochody

Sprzedaj

Wyloguj

Podpis

Zdjęcie

Dzienna
sprzedaż

0.00

Zegarek

Podsumowanie samochodów

Typ samochodu 1

Typ samochodu 2

Typ samochodu 3

Typ samochodu 4

Witam w dealerze samochodowym



zdj.profil

Podsumowanie samochodów

Typ samochodu 1

Typ samochodu 1

Typ samochodu 1

Typ samochodu 1

Panel

Klienci

Użytkownicy

Samochody

Sprzedaj

Wyloguj

Podpis

Zdjęcie

SALE

Dzienna
sprzedaż

0.00

Zegarek

Rejestracja klienta



Nazwa

Adres

Telefon

Zapisz

Uaktualnij

Wyczyść

+

szukaj

Nazwa	Adres	Telefon
-------	-------	---------

Witam w dealerze samochodowym

zdj.profil

+

szukaj

Nazwa	Adres	Telefon
-------	-------	---------

Panel

Klienci

Użytkownicy

Samochody

Sprzedaj

Wyloguj

Podpis

Zdjęcie

SALE

Dzienna sprzedaż

0.00

Zegarek

Rejestracja klienta



Marka

Model

Typ nadwozia

Ilość

Cena

Zapisz

Uaktualnij

Wyczyść



Marka

Model

Typ nadwozia

Ilość

Cena

Witam w dealerze samochodowym



zdj.profil



szukaj

Marka

Model

Typ nadwozia

ilość

cena

Panel

Klienci

Użytkownicy

Samochody

Sprzedaj

Wyloguj

Podpis

Zdjęcie

Dzienna
sprzedaż

0.00

Zegarek



Nazwa

Model

Typ nadwozia

Cena

Select

Wyszukaj

Zatwierdź

Sprzedaj klient

Section 14



Nazwa

Telefon

Choice

Wyszukaj

Sprzedaj form

+ Produkt

Translaction on: 00000000

Cena całkowita \$: 0.00

Sprzedaj

Nazwa

Ilość

Cena

Całkowity koszt

Nazwa klienta

Kasjer

MainForm sprzedaj

Witam w dealerze samochodowym

+ Produkt Translaction on: 00000000 Cena całkowita \$: 0.00 **Sprzedaj**

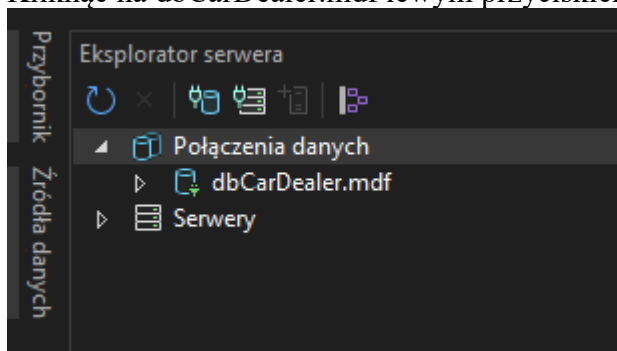
Marka	Ilość	Cena	Całkowity koszt	Nazwa klienta	Kasjer
<div style="display: flex;"> <div style="flex: 1;"> <p>Podpis</p> </div> <div style="flex: 1;"> <p>Zdjęcie</p> </div> <div style="flex: 0.5;"> <p>Dzienna sprzedaż 0.00</p> <p>Zegarek</p> </div> </div>					

Panel
 Klienci
 Użytkownicy
 Samochody
 Sprzedaj
 Wyloguj

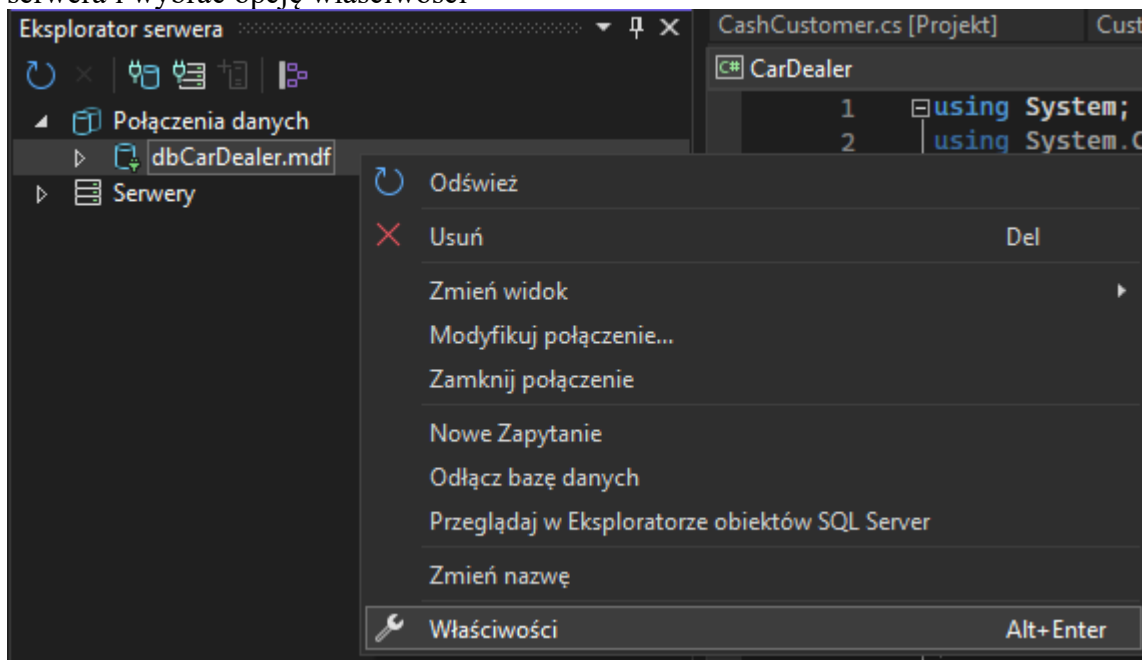
7. Jak uruchomić aplikację

Aby poprawnie uruchomić aplikację należy:

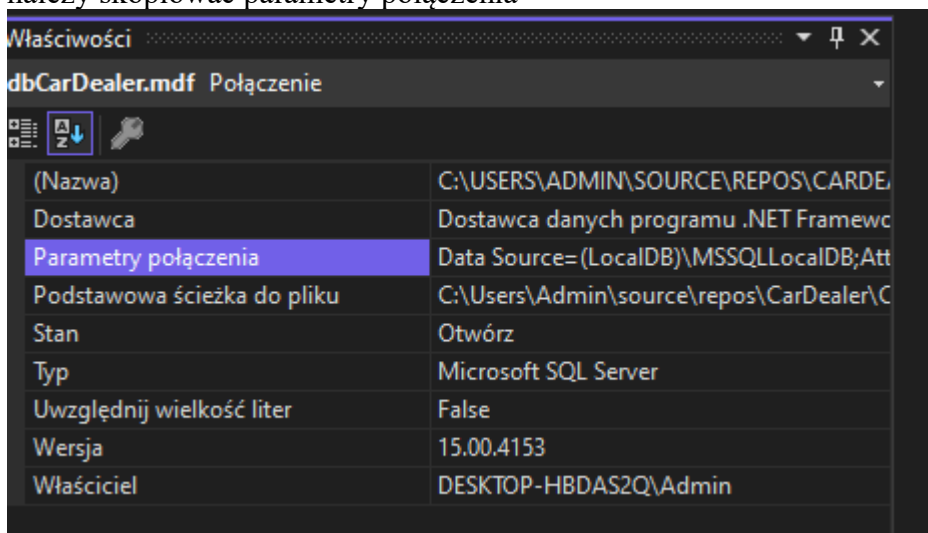
1. Kliknąć na dbCarDealer.mdf lewym przyciskiem myszy włączy się eksplorator serwera



- Następnie należy kliknąć prawym przyciskiem myszy na dbCarDealer.mdf w eksploratorze serwera i wybrać opcję właściwości



- Po kliknięciu na właściwości pojawi się okno właściwości w prawym dolnym rogu i stamtąd należy skopiować parametry połączenia



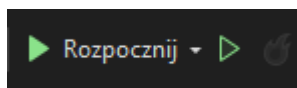
- Następnie należy przejść do DbConnect.cs i wkleić skopiowany tekst do miennej con zamiast PATH(19 linijka)

```
con = @"PATH";
```

- Tak powinna wyglądać poprawne połączenie z bazą

```
17 Odwołania: 13  
18 public string connection()  
19 {  
20     con = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Admin\source\repos\CarDealer\CarDealer\d  
21     return con;  
}
```

- Następnie należy włączyć aplikację przyciskiem Rozpocznij



Przygotowanie Konta użytkowników:

- Konto administratora:
 - Username- admin
 - Password- admin
- Konto Pracownika
 - Username- kasjer
 - Password- kasjer

8. Implementacje

W formularzu CashCustorem są metody takie jak

1. Metoda `txtSearch_TextChanged(object sender, EventArgs e)`
 - Jest wywoływana, gdy zmienia się tekst w polu tekstowym `txtSearch`.
 - Wywołuje metodę `LoadCustomer()`.
2. Metoda `dgvCustomer_CellContentClick(object sender, DataGridViewCellEventArgs e)`
 - Jest wywoływana, gdy kliknięto zawartość komórki w kontrolce `dgvCustomer` (obiekt `DataGridView`).
 - Sprawdza nazwę kolumny dla klikniętej komórki.
 - Jeśli nazwa kolumny to "Choice", wykonuje polecenie SQL UPDATE, które aktualizuje pole "cid" w tabeli "tbCash" na wartość pobraną z komórki drugiej kolumny wiersza klikniętej komórki, gdzie wartość pola "transno" wynosi to samo, co wartość etykiety `cash.lblTransno.Text`.
 - Wywołuje metodę `loadCash()` na obiekcie `cash`.
 - Zamyka bieżące okno formularza (`this.Dispose()`).
3. Metoda `LoadCustomer()`
 - Jest publiczną metodą.
 - Aktualizuje zawartość kontrolki `dgvCustomer` (obiekt `DataGridView`) danymi z bazy danych.
 - Usuwa wszystkie wiersze z `dgvCustomer` (`dgvCustomer.Rows.Clear()`).
 - Tworzy nowe polecenie SQL `SELECT`, które pobiera z bazy danych id, name i phone z tabeli "tbCustomer", gdzie pole "name" zawiera podany tekst z pola tekstowego `txtSearch`.
 - Otwiera połączenie z bazą danych (`cn.Open()`).
 - Wykonuje polecenie SQL (`cm.ExecuteReader()`).
 - W pętli `while` odczytuje dane z obiektu `SqlDataReader` (`dr`) i dodaje je jako nowy wiersz do kontrolki `dgvCustomer`.
 - Zamyka obiekt `SqlDataReader` (`dr.Close()`).
 - Zamyka połączenie z bazą danych (`cn.Close()`).
 - W przypadku wystąpienia błędu, zamyka połączenie z bazą danych, wyświetla komunikat o błędzie za pomocą okna dialogowego `MessageBox.Show()` i zamyka bieżącą instancję formularza.

W formularzu CashForm są metody takie jak:

1. Metoda `btnAdd_Click(object sender, EventArgs e)`:
 - Tworzy nową instancję klasy `CashProduct` i przypisuje wartość pola `uname` na podstawie `lblUsername` w głównym formularzu.
 - Wyświetla formularz `CashProduct` jako okno dialogowe.
2. Metoda `btnCash_Click(object sender, EventArgs e)`:

- Tworzy nową instancję klasy `CashCustomer` i wyświetla ją jako okno dialogowe.
 - Jeśli użytkownik potwierdzi spieniężenie produktu poprzez kliknięcie przycisku "Yes" w wyświetlonym oknie dialogowym, wykonuje następujące czynności:
 - Wywołuje funkcję `getTransno()` w celu uzyskania numeru transakcji.
 - Wywołuje funkcję `loadDailySale()` w głównym formularzu (`main`).
 - Pętla `for` iteruje przez wiersze w kontrolce `dgvCash`.
 - Dla każdego wiersza aktualizuje bazę danych, zmniejszając wartość `pqty` w tabeli `tbProduct` o wartość w kolumnie 4 (indeks 3) dla danego wiersza.
 - Czyści kontrolkę `dgvCash`.
3. Metoda `dgvCash_CellContentClick(object sender, DataGridViewCellEventArgs e)`:
- Sprawdza, które pole w kontrolce `dgvCash` zostało kliknięte na podstawie `e.ColumnIndex`.
 - W zależności od klikniętego pola wykonuje odpowiednie czynności:
 - "Delete": Jeśli użytkownik potwierdzi usunięcie gotówki poprzez kliknięcie przycisku "Yes" w wyświetlonym oknie dialogowym, usuwa rekord z tabeli `tbCash` na podstawie wartości `cashid` dla danego wiersza.
 - "Increase": Zwiększa wartość `qty` w tabeli `tbCash` o 1 dla danego wiersza, pod warunkiem, że wartość w kolumnie 4 (indeks 3) jest mniejsza niż wartość zwrócona przez funkcję `checkPqty()` dla `pcode` z kolumny 2 (indeks 1) dla danego wiersza. W przeciwnym razie wyświetla okno dialogowe informujące o braku wystarczającej ilości produktów.
 - "Decrease": Zmniejsza wartość `qty` w tabeli `tbCash` o 1 dla danego wiersza. Jeśli wartość w kolumnie 4 (indeks 3) wynosi 1, zmienia wartość `colName` na "Delete" i przechodzi do etykiety `removeitem`, aby usunąć ten wiersz.
 - Wywołuje funkcję `loadCash()` w celu odświeżenia danych dotyczących gotówki.
4. Metoda `getTransno()`:
- Pobiera aktualną datę i godzinę i tworzy z niej łańcuch znaków w formacie "yyyyMMdd".
 - Wykonuje zapytanie SQL, aby uzyskać numer transakcji dla najnowszego rekordu w tabeli `tbCash`, gdzie `transno` zaczyna się od wartości uzyskanej wcześniej.
 - Jeśli istnieje rekord, pobiera ostatnie 4 cyfry numeru transakcji, zwiększa je o 1 i tworzy nowy numer transakcji.
 - Jeśli nie istnieje żaden rekord, tworzy nowy numer transakcji na podstawie wcześniej utworzonego łańcucha daty i godziny, zwiększając 4 ostatnie cyfry o 1001.
 - Ustawia wartość etykiety `lblTransno` na uzyskany numer transakcji.
5. Metoda `loadCash()`:
- Czyści kontrolkę `dgvCash`.
 - Wykonuje zapytanie SQL, aby pobrać dane dotyczące gotówki z tabeli `tbCash` na podstawie numeru transakcji (`transno`).
 - Dodaje wiersze do kontrolki `dgvCash` na podstawie wyników zapytania.
 - Oblicza sumę wartości z kolumny 5 (indeks 4) i wyświetla ją w etykiecie `lblTotal`.
6. Metoda `checkPqty(string pcode)`:
- Wykonuje zapytanie SQL, aby sprawdzić dostępną ilość produktu (`pqty`) w tabeli `tbProduct` na podstawie kodu produktu (`pcode`).
 - Zwraca wartość `pqty`.

W formularzu `CashProduct` są metody takie jak

1. Metoda `btnCash_Click(object sender, EventArgs e)`:
- Iteruje przez wiersze w kontrolce `dgvProduct`.
 - Dla każdego wiersza sprawdza wartość w kolumnie "Select" (indeks 0).

- Jeśli wartość jest `true`, wykonuje następujące czynności:
 - Tworzy nowe polecenie SQL w celu wstawienia danych do tabeli `tbCash`.
 - Ustawia parametry polecenia na podstawie danych z wiersza i innych wartości, takich jak numer transakcji (`cash.lblTransno.Text`) i nazwa kasjera (`uname`).
 - Otwiera połączenie z bazą danych, wykonuje polecenie `ExecuteNonQuery()` i zamyka połączenie.
 - Wywołuje funkcję `loadCash()` w formularzu `cash`.
 - Zamyka bieżące okno formularza (`this.Dispose()`).
2. Metoda `txtSearch_TextChanged(object sender, EventArgs e)`:
- Wywołuje funkcję `LoadProduct()` w celu wczytania produktów na podstawie wartości wprowadzonej w kontrolce `txtSearch`.
3. Metoda `LoadProduct()`:
- Czyści kontrolkę `dgvProduct`.
 - Wykonuje zapytanie SQL w celu pobrania produktów z tabeli `tbProduct`, które mają nazwę, typ lub kategorię pasujące do wartości wprowadzonej w kontrolce `txtSearch` i ilość większą niż 0.
 - Dodaje wiersze do kontrolki `dgvProduct` na podstawie wyników zapytania.

W formularzu `CustomerForm` są metody takie jak:

1. Metoda `btnAdd_Click(object sender, EventArgs e)`:
 - Tworzy nową instancję klasy `CustomerModule`.
 - Wywołuje metodę `ShowDialog()` na obiekcie `module`, co powoduje wyświetlenie okna dialogowego do dodawania klienta.
2. Metoda `txtSearch_TextChanged(object sender, EventArgs e)`:
 - Wywołuje funkcję `LoadCustomer()` w celu wczytania klientów na podstawie wartości wprowadzonej w kontrolce `txtSearch`.
3. Metoda `dgvCustomer_CellContentClick(object sender, DataGridViewCellEventArgs e)`:
 - Pobiera nazwę kolumny dla klikniętej komórki w kontrolce `dgvCustomer`.
 - Jeśli nazwa kolumny to "Edit":
 - Tworzy nową instancję klasy `CustomerModule`.
 - Ustawia wartości pól w formularzu `module` na podstawie danych z wybranego wiersza w kontrolce `dgvCustomer`.
 - Wyłącza przycisk "Save" (`module.btnSave`) i włącza przycisk "Update" (`module.btnUpdate`) w celu umożliwienia edycji danych klienta.
 - Wywołuje metodę `ShowDialog()` na obiekcie `module`, co powoduje wyświetlenie okna dialogowego do edycji klienta.
 - Jeśli nazwa kolumny to "Delete":
 - Wyświetla okno dialogowe z pytaniem, czy na pewno usunąć rekord klienta.
 - Jeśli odpowiedź to "Yes", wykonuje polecenie SQL do usunięcia klienta z bazy danych.
 - Wyświetla komunikat o pomyślnym usunięciu danych klienta.
 - Wywołuje funkcję `LoadCustomer()` w celu ponownego wczytania danych klientów.
4. Metoda `LoadCustomer()`:
 - Czyści kontrolkę `dgvCustomer`.
 - Wykonuje zapytanie SQL w celu pobrania danych klientów z tabeli `tbCustomer`, których pola `name`, `address` lub `phone` pasują do wartości wprowadzonej w kontrolce `txtSearch`.

- Dodaje wiersze do kontrolki `dgvCustomer` na podstawie wyników zapytania.

W formularzu `CustomerModule` są metody takie jak:

1. Metoda `btnSave_Click(object sender, EventArgs e)`:

- Wywoływana po kliknięciu przycisku "Save" (`btnSave`).
- Sprawdza pola formularza za pomocą metody `CheckField()`.
- Jeśli wszystkie pola zostały wypełnione poprawnie (wartość `check` jest ustawiona na `true`):
 - Wyświetla okno dialogowe z pytaniem, czy na pewno zarejestrować tego klienta.
 - Jeśli użytkownik odpowie "Yes", wykonuje polecenie SQL do dodania nowego klienta do bazy danych.
 - Zamyka połączenie z bazą danych.
 - Wyświetla komunikat o pomyślnym zarejestrowaniu klienta.
 - Czyści pola formularza.
 - Wywołuje metodę `LoadCustomer()` w formularzu `CustomerForm` w celu ponownego wczytania danych klientów.

2. Metoda `btnUpdate_Click(object sender, EventArgs e)`:

- Wywoływana po kliknięciu przycisku "Update" (`btnUpdate`).
- Sprawdza pola formularza za pomocą metody `CheckField()`.
- Jeśli wszystkie pola zostały wypełnione poprawnie (wartość `check` jest ustawiona na `true`):
 - Wyświetla okno dialogowe z pytaniem, czy na pewno edytować ten rekord.
 - Jeśli użytkownik odpowie "Yes", wykonuje polecenie SQL do aktualizacji danych klienta w bazie danych.
 - Zamyka połączenie z bazą danych.
 - Wyświetla komunikat o pomyślnym zaktualizowaniu danych klienta.
 - Czyści pola formularza.
 - Wywołuje metodę `LoadCustomer()` w formularzu `CustomerForm` w celu ponownego wczytania danych klientów.
 - Zamyka formularz `CustomerModule`.

3. Metoda `btnCancel_Click(object sender, EventArgs e)`:

- Wywoływana po kliknięciu przycisku "Cancel" (`btnCancel`).
- Czyści pola formularza.

4. Metoda `btnClose_Click(object sender, EventArgs e)`:

- Wywoływana po kliknięciu przycisku "Close" (`btnClose`).
- Zamyka formularz `CustomerModule`.

5. Metoda `CheckField()`:

- Sprawdza, czy pola `txtName`, `txtAddress` i `txtPhone` są puste.
- Jeśli którykolwiek z tych pól jest pusty, wyświetla komunikat o konieczności wypełnienia wszystkich pól.
- W przeciwnym razie ustawia wartość `check` na `true`.

6. Metoda `Clear()`:

- Czyści zawartość pól `txtName`, `txtAddress` i `txtPhone`.
- Włącza przycisk "Save" (`btnSave`) i wyłącza przycisk "Update" (`btnUpdate`).

W formularzu `Dashboard` są metody takie jak:

1. Metoda `extractData(string str)`:

- Przyjmuje parametr `str`, który określa kategorię produktu.

- Wykonuje zapytanie SQL, aby obliczyć sumę **pqty** (ilość produktów) dla danej kategorii (**str**) w tabeli **tbProduct**.
 - Jeśli nie ma danych dla danej kategorii, zwraca wartość 0.
 - Zamyka połączenie z bazą danych.
 - Jeśli wystąpił błąd, wyświetla komunikat z wyjątkiem.
 - Zwraca obliczoną sumę jako wartość.
2. Zdarzenie **Dashboard_Load(object sender, EventArgs e)**:
- Wywoływane po załadowaniu formularza **Dashboard**.
 - Wywołuje metodę **extractData("Coupe")** i przypisuje zwróconą wartość do etykiety **lblCoupe**.
 - Wywołuje metodę **extractData("Sedan")** i przypisuje zwróconą wartość do etykiety **lblSedan**.
 - Wywołuje metodę **extractData("Kombi")** i przypisuje zwróconą wartość do etykiety **lblKombi**.
 - Wywołuje metodę **extractData("Suv")** i przypisuje zwróconą wartość do etykiety **lblSuv**.

W klasie **DbConnect** która służy do podłączenia się do bazy danych są metody takie jak:

- **SqlConnection cn**: Obiekt reprezentujący połączenie z bazą danych.
 - **SqlCommand cm**: Obiekt reprezentujący zapytanie SQL.
 - **string con**: Zmienna przechowująca łańcuch połączeniowy do bazy danych.
2. Metoda **connection()**:
- Zwraca łańcuch połączeniowy do lokalnej bazy danych.
 - Łańcuch połączeniowy zawiera informacje o lokalizacji pliku bazy danych i zabezpieczeniach.
3. Metoda **executeQuery(string sql)**:
- Przyjmuje parametr **sql**, który zawiera zapytanie SQL do wykonania.
 - Nawiązuje połączenie z bazą danych, używając metody **connection()** do uzyskania łańcucha połączeniowego.
 - Tworzy nowy obiekt **SqlCommand** i przypisuje mu zapytanie SQL oraz połączenie.
 - Wykonuje zapytanie SQL przy użyciu metody **ExecuteNonQuery()**.
 - Zamyka połączenie z bazą danych.
 - Jeśli wystąpił błąd, zamyka połączenie i wyświetla komunikat z wyjątkiem za pomocą **MessageBox.Show()**.

W formularzu **LoginForm** są metody takie jak:

1. Metoda **btnClose_Click(object sender, EventArgs e)**:
- Obsługuje kliknięcie przycisku "Zamknij".
 - Wyświetla okno dialogowe z pytaniem, czy użytkownik na pewno chce wyjść z aplikacji.
 - Jeśli odpowiedź to "Tak" (**DialogResult.Yes**), zamyka aplikację za pomocą **Application.Exit()**.
2. Metoda **btnLogin_Click(object sender, EventArgs e)**:
- Obsługuje kliknięcie przycisku "Zaloguj".
 - Sprawdza wprowadzone dane logowania w bazie danych.
 - Wykonuje zapytanie SQL, aby znaleźć użytkownika o podanej nazwie (**txtName.Text**) i hasle (**txtPassword.Text**).
 - Jeśli użytkownik istnieje, pobiera jego nazwę i rolę.
 - Wyświetla powitalne okno dialogowe z nazwą użytkownika.
 - Tworzy instancję klasy **MainForm** i przekazuje nazwę użytkownika i rolę.

- Ukrywa formularz logowania i pokazuje formularz główny (`main.ShowDialog()`).
 - Jeśli użytkownik nie istnieje, wyświetla ostrzeżenie o nieprawidłowej nazwie użytkownika i hasle.
3. Metoda `btnForgot_Click(object sender, EventArgs e)`:
- Obsługuje kliknięcie przycisku "Zapomniałeś hasła".
 - Wyświetla okno dialogowe z informacją, aby skontaktować się ze swoim szefem w przypadku zapomnienia hasła.

W formularzu MainForm są metody takie jak:

1. Metoda `btnClose_Click(object sender, EventArgs e)`:
 - Obsługuje kliknięcie przycisku "Zamknij".
 - Wyświetla okno dialogowe z pytaniem, czy użytkownik na pewno chce wyjść z aplikacji.
 - Jeśli odpowiedź to "Tak" (`DialogResult.Yes`), zamyka aplikację za pomocą `Application.Exit()`.
2. Metody obsługujące kliknięcie przycisków nawigacyjnych (`btnDashboard_Click`, `btnCustomer_Click`, `btnUser_Click`, `btnProduct_Click`, `btnCash_Click`):
 - Wywołują metodę `openChildForm` z odpowiednim formularzem podrzędnym, który ma zostać otwarty.
3. Metoda `btnLogout_Click(object sender, EventArgs e)`:
 - Obsługuje kliknięcie przycisku "Wyloguj".
 - Wyświetla okno dialogowe z pytaniem, czy użytkownik na pewno chce się wylogować.
 - Jeśli odpowiedź to "Tak" (`DialogResult.Yes`), tworzy nową instancję klasy `LoginForm`, ukrywa bieżący formularz i pokazuje formularz logowania.
4. Metoda `MainForm_Load(object sender, EventArgs e)`:
 - Jest wywoływana po załadowaniu formularza głównego.
 - Tworzy kontrolkę `ProgressBar` i dodaje ją do formularza.
 - Tworzy i uruchamia `System.Timers.Timer`, który co sekundę aktualizuje wartość kontrolki `ProgressBar` na podstawie aktualnego czasu.
5. Metoda `Timer_Elapsed(object sender, System.Timers.ElapsedEventArgs e)`:
 - Jest wywoływana przez timer co sekundę.
 - Aktualizuje wartość kontrolki `ProgressBar` zgodnie z aktualnym czasem.
6. Metoda `openChildForm(Form childForm)`:
 - Otwiera formularz podrzędny.
 - Zamyka aktywny formularz podrzędny (jeśli istnieje).
 - Ustawia nowy formularz jako aktywny formularz podrzędny.
 - Ustawia właściwości nowego formularza, takie jak `top-level`, styl obramowania i sposób rozmieszczenia.
 - Ustawia tytuł formularza głównego na tytuł formularza podrzędnego.
 - Dodaje formularz podrzędny do panelu `panelChild` i wyświetla go.
7. Metoda `loadDailySale()`:
 - Ładuje dane dotyczące codziennych sprzedaży.
 - Pobiera aktualną datę w formacie "yyyyMMdd".
 - Wykonuje zapytanie SQL, aby obliczyć sumę wartości sprzedaży z tabeli `tbCash` dla danego dnia.
 - Wyświetla sumę sprzedaży w kontrolce `lblDailySale`.

W formularzu ProductForm są metody takie jak:

1. Metoda `btnAdd_Click(object sender, EventArgs e)`:
 - Obsługuje kliknięcie przycisku "Dodaj".
 - Tworzy instancję formularza `ProductModule` i wyświetla go jako okno dialogowe.
2. Metoda `txtSearch_TextChanged(object sender, EventArgs e)`:
 - Obsługuje zmianę tekstu w polu wyszukiwania.
 - Wywołuje metodę `LoadProduct()` w celu ponownego załadowania produktów z uwzględnieniem aktualnego tekstu wyszukiwania.
3. Metoda `dgvProduct_CellContentClick(object sender, DataGridViewCellEventArgs e)`:
 - Obsługuje kliknięcie na komórkę w tabeli produktów.
 - Sprawdza nazwę kolumny klikniętej komórki.
 - Jeśli nazwa kolumny to "Edit", otwiera formularz `ProductModule` w trybie edycji, pobierając dane wybranego produktu i przekazując je do formularza `ProductModule`.
 - Jeśli nazwa kolumny to "Delete", wyświetla okno dialogowe z pytaniem, czy użytkownik na pewno chce usunąć produkt. Jeśli odpowiedź to "Tak" (`DialogResult.Yes`), wykonuje zapytanie SQL do usunięcia produktu z bazy danych.
 - Na koniec wywołuje metodę `LoadProduct()` w celu odświeżenia listy produktów.
4. Metoda `LoadProduct()`:
 - Ładuje produkty z bazy danych i wyświetla je w tabeli `dgvProduct`.
 - Czyści tabelę `dgvProduct`.
 - Tworzy zapytanie SQL, które wyszukuje produkty na podstawie nazwy, typu i kategorii produktu zgodnie z wartością w polu wyszukiwania `txtSearch`.
 - Otwiera połączenie z bazą danych.
 - Wykonuje zapytanie SQL i odczytuje wyniki do obiektu `SqlDataReader`.
 - W pętli `while` dodaje kolejne wiersze do tabeli `dgvProduct` na podstawie odczytanych danych.
 - Zamyka obiekt `SqlDataReader` i zamyka połączenie z bazą danych.

W formularzu ProductModule są metody takie jak

1. Metoda `btnClose_Click(object sender, EventArgs e)`:
 - Obsługuje kliknięcie przycisku "Zamknij".
 - Zamyka formularz.
2. Metoda `btnSave_Click(object sender, EventArgs e)`:
 - Obsługuje kliknięcie przycisku "Zapisz".
 - Sprawdza poprawność pól danych wywołując metodę `CheckField()`.
 - Jeśli pola danych są poprawne, wyświetla okno dialogowe z pytaniem, czy użytkownik na pewno chce zarejestrować produkt.
 - Jeśli odpowiedź to "Tak" (`DialogResult.Yes`), tworzy zapytanie SQL do dodania produktu do bazy danych na podstawie wprowadzonych danych.
 - Otwiera połączenie z bazą danych, wykonuje zapytanie SQL i zamyka połączenie.
 - Wyświetla komunikat o pomyślnym dodaniu produktu, czyści pola danych, odświeża listę produktów w formularzu `ProductForm`.
 - Metoda `LoadProduct()` zostaje wywołana z formularza `ProductForm`, ponieważ formularz `ProductModule` nie ma bezpośredniego dostępu do tabeli produktów.
3. Metoda `btnUpdate_Click(object sender, EventArgs e)`:
 - Obsługuje kliknięcie przycisku "Aktualizuj".
 - Sprawdza poprawność pól danych wywołując metodę `CheckField()`.
 - Jeśli pola danych są poprawne, wyświetla okno dialogowe z pytaniem, czy użytkownik na pewno chce zaktualizować produkt.

- Jeśli odpowiedź to "Tak" (DialogResult.Yes), tworzy zapytanie SQL do aktualizacji produktu w bazie danych na podstawie wprowadzonych danych.
 - Otwiera połączenie z bazą danych, wykonuje zapytanie SQL i zamyka połączenie.
 - Wyświetla komunikat o pomyślnym zaktualizowaniu produktu.
 - Odświeża listę produktów w formularzu **ProductForm**.
 - Zamyka formularz **ProductModule**.
4. Metoda **btnCancel_Click(object sender, EventArgs e)**:
 - Obsługuje kliknięcie przycisku "Anuluj".
 - Czyści pola danych wywołując metodę **Clear()**.
 5. Metoda **txtQty_KeyPress(object sender, KeyPressEventArgs e)**:
 - Obsługuje zdarzenie KeyPress dla pola **txtQty**.
 - Pozwala wprowadzać tylko cyfry i znaki sterujące (control characters) poprzez ustawienie **e.Handled** na **true** dla innych znaków.
 6. Metoda **txtPrice_KeyPress(object sender, KeyPressEventArgs e)**:
 - Obsługuje zdarzenie KeyPress dla pola **txtPrice**.
 - Pozwala wprowadzać tylko cyfry i kropkę jako separator dziesiętny.
 - Pozwala na wprowadzenie tylko jednego separatora dziesiętnego poprzez sprawdzenie, czy już występuje w tekście.
 7. Metoda **Clear()**:
 - Czyści zawartość pól danych w formularzu.
 - Ustawia indeks kontrolki **cbCategory** na 0.
 - Wyłącza przycisk **btnSave**.
 8. Metoda **CheckField()**:
 - Sprawdza, czy pola danych są poprawnie wypełnione.
 - Jeśli któreś z pól danych jest puste, wyświetla ostrzeżenie.
 - Jeśli wszystkie pola danych są wypełnione, ustawia flagę **check** na **true**.

W formularzu **SplashForm** są metody takie jak

1. Pole **int startPoint**:
 - Przechowuje aktualną wartość postępu.
2. Metoda **timer1_Tick(object sender, EventArgs e)**:
 - Obsługuje zdarzenie Tick timera o nazwie **timer1**.
 - Zwiększa wartość **startPoint** o 2.
 - Ustawia wartość pola **Value** kontrolki **guna2ProgressBar1** na **startPoint**, co aktualizuje postęp na pasku postępu.
 - Jeśli wartość **guna2ProgressBar1.Value** wynosi 100, resetuje wartość na 0, zatrzymuje timer, tworzy formularz **LoginForm**, wywołuje metodę **ShowDialog()** na formularzu **LoginForm** (blokując tymczasowo działanie aplikacji), a następnie ukrywa formularz **SplashForm**.
3. Metoda **SplashForm_Load(object sender, EventArgs e)**:
 - Obsługuje zdarzenie Load formularza.
 - Uruchamia timer o nazwie **timer1**, rozpoczynając animację paska postępu.

W formularzu **UserForm** są metody takie jak

1. Metoda **dgvUser_CellContentClick(object sender, DataGridViewCellEventArgs e)**:
 - Obsługuje zdarzenie kliknięcia na komórkę w kontrolce **dgvUser** (DataGridView).
 - Sprawdza nazwę kolumny dla klikniętej komórki.

- Jeśli nazwa kolumny to "Edit", tworzy formularz `UserModule` i ustawia wartości pól formularza na wartości komórek wiersza związane z użytkownikiem. Następnie wyświetla formularz w trybie edycji.
 - Jeśli nazwa kolumny to "Delete", wyświetla okno dialogowe z pytaniem o potwierdzenie usunięcia rekordu. Jeśli odpowiedź to "Yes", wykonuje zapytanie SQL do usunięcia użytkownika z bazy danych. Wyświetla komunikat potwierdzający usunięcie.
2. Metoda `txtSearch_TextChanged(object sender, EventArgs e)`:
 - Wywołuje metodę `LoadUser()` w celu załadowania użytkowników na podstawie tekstu wprowadzonego w polu wyszukiwania.
 3. Metoda `btnAdd_Click(object sender, EventArgs e)`:
 - Tworzy formularz `UserModule` i wyświetla go w celu dodania nowego użytkownika.
 4. Metoda `LoadUser()`:
 - Załadowuje dane użytkowników do kontrolki `dgvUser` (DataGridView).
 - Czyści wiersze w kontrolce `dgvUser`.
 - Tworzy i wykonuje zapytanie SQL, które pobiera dane użytkowników z bazy danych na podstawie tekstu wprowadzonego w polu wyszukiwania.
 - Odczytuje dane z bazy danych i dodaje je jako nowe wiersze do kontrolki `dgvUser`.

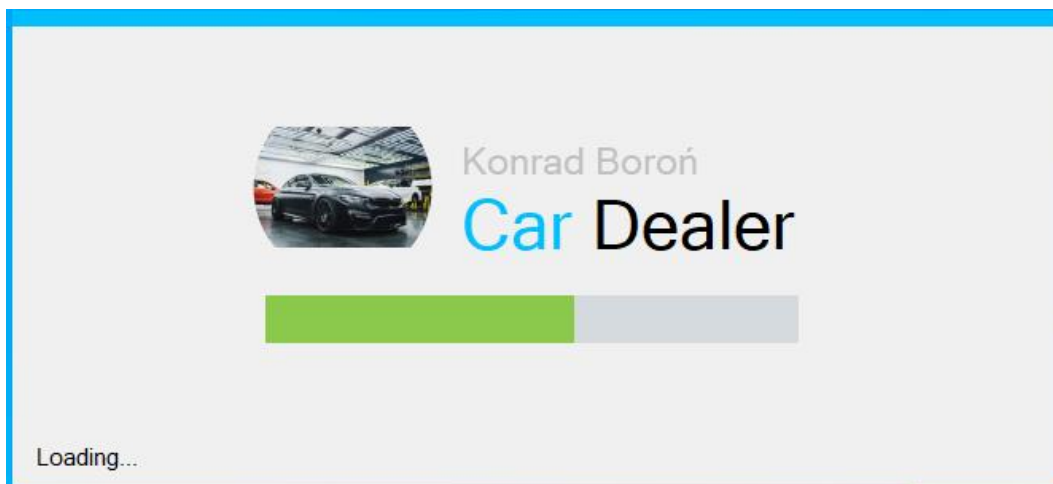
W formularzu `User Module` są metody takie jak:

1. Metoda `UserModule_Load(object sender, EventArgs e)`
 - Zdarzenie wywoływane podczas ładowania formularza `UserModule`.
 - Obecnie nie wykonuje żadnych operacji.
2. Metoda `btnSave_Click(object sender, EventArgs e)`
 - Obsługuje kliknięcie przycisku "Save" (Zapisz) w formularzu `UserModule`.
 - Sprawdza poprawność pól formularza oraz wiek użytkownika.
 - Jeśli dane są poprawne i użytkownik potwierdzi zapis, wywoływane jest polecenie SQL INSERT do dodania nowego użytkownika do bazy danych.
 - Wyświetla komunikat o pomyślnym zapisie użytkownika.
 - Czyści pola formularza i odświeża listę użytkowników w formularzu `UserForm`.
3. Metoda `btnUpdate_Click(object sender, EventArgs e)`
 - Obsługuje kliknięcie przycisku "Update" (Aktualizuj) w formularzu `UserModule`.
 - Sprawdza poprawność pól formularza oraz wiek użytkownika.
 - Jeśli dane są poprawne i użytkownik potwierdzi aktualizację, wywoływane jest polecenie SQL UPDATE do zaktualizowania istniejącego użytkownika w bazie danych.
 - Wyświetla komunikat o pomyślnej aktualizacji danych użytkownika.
 - Czyści pola formularza, odświeża listę użytkowników w formularzu `UserForm` i zamyka formularz `UserModule`.
4. Metoda `btnCancel_Click(object sender, EventArgs e)`
 - Obsługuje kliknięcie przycisku "Cancel" (Anuluj) w formularzu `UserModule`.
 - Czyści pola formularza, przywraca domyślny stan formularza.
5. Metoda `btnClose_Click(object sender, EventArgs e)`
 - Obsługuje kliknięcie przycisku "Close" (Zamknij) w formularzu `UserModule`.
 - Zamyka formularz `UserModule`.
6. Metoda `cbRole_SelectedIndexChanged(object sender, EventArgs e)`
 - Obsługuje zmianę wyboru w ComboBox `cbRole`.
 - W zależności od wybranej roli, dostosowuje wygląd formularza (widoczność pola `txtPass`).
7. Metoda `Clear()`
 - Czyści pola formularza `UserModule`.

- Przywraca domyślne wartości.
8. Metoda **CheckField()**
- Sprawdza poprawność pól formularza i wieku użytkownika.
 - Wyświetla odpowiednie komunikaty ostrzegawcze w przypadku błędów.
 - Ustawia zmienną **check** na **true**, jeśli wszystkie pola są poprawne.
9. Metoda **checkAge(DateTime dateofBirth)**
- Oblicza wiek użytkownika na podstawie daty urodzenia.
 - Zwraca obliczony wiek.

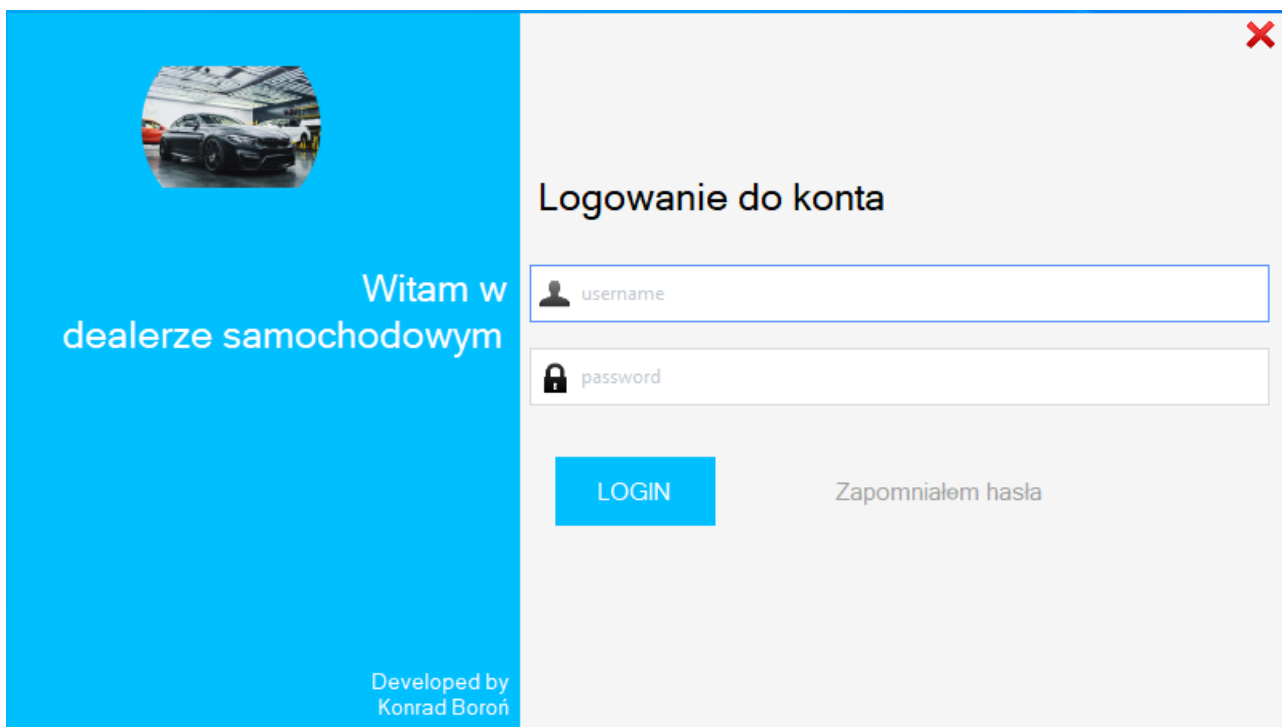
9. Działanie aplikacji (Screeny części okienek i opis):

Ekran ładowania:




Po załadowaniu się paska na ekranie ładowania pojawia możliwość zalogowania do aplikacji


Logowanie:



W Ekranie logowania użytkownik ma możliwość do zalogowania się na swoje konto które zostało stworzone przez administratora lub wyłączyć aplikację.

Panel:

**Konrad Boroń**
Student

**admin**
Administrator

Panel

Klienci

Użytkownik

Samochody

Sprzedaż

Wyloguj

Panel

Car Summary


Coupe
1

Sedan
3

Kombi
2

Suv
4

Projekt wykonał
Konrad Boroń



Dzienny raport

0,00


Dzienna sprzedaż


Całkowita dzienna sprzedaż

02:05:30

Panel jest to karta która otwiera się za każdym razem gdy użytkownik się zaloguje. Po lewej stronie jest menu dzięki któremu użytkownik może przełączać się między kartami. W prawym dolnym rogu znajduje się zegarek i dzienny raport ilości dochodu na dzień.

Klienci:

**Konrad Boroń**
Student

**admin**
Administrator

Klienci

Panel

Użytkownik

Samochody

Sprzedaż

Wyloguj


Klienci

+

Wyszukaj

No	ID	Nazwa	Adres	Telefon
1	2	Konrad Boroń	Siedlanka	111222333
2	3	Adam Kowalski	Kraków 34	5654322456
3	6	Dominik Kozak	Warszawa 56	777888999

Projekt wykonał
Konrad Boroń



Dzienny raport

0,00

Dzienna sprzedaż

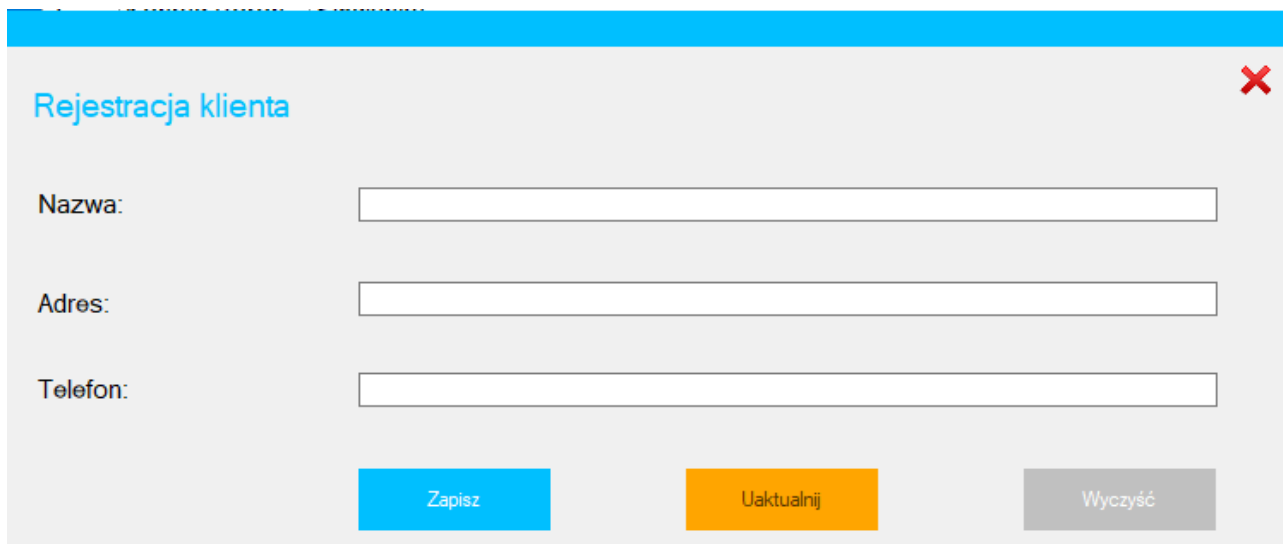
Całkowita dzienna sprzedaż

02:16:42

Klienci jest to karta w której wyświetlani są wszyscy dodani klienci. W tej karcie również można dodać nowego klienta klikając w plusik. Klikając w plusik pojawi się moduł do dodawania klienta.

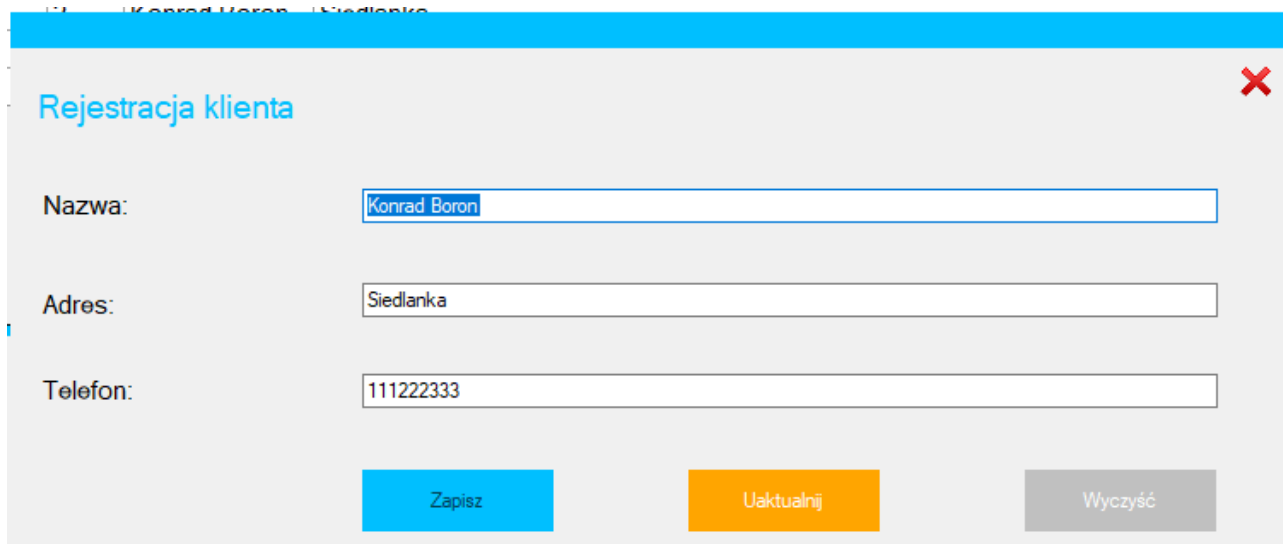
Również klikając w długopis można edytować dane klienta. A klikając w kosz można usuwać usuwać dany rekord.

Po kliknięciu w plusik:




The screenshot shows a web form titled "Rejestracja klienta" (Client Registration) with a red close button in the top right corner. The form contains three input fields: "Nazwa:" (Name), "Adres:" (Address), and "Telefon:" (Phone). Below the fields are three buttons: "Zapisz" (Save) in blue, "Uaktualnij" (Update) in orange, and "Wyczyść" (Clear) in grey. The form is currently empty.

Po kliknięciu w długopis:




The screenshot shows the same "Rejestracja klienta" form, but now it contains pre-filled data. The "Nazwa:" field is filled with "Konrad Boron", the "Adres:" field is filled with "Siedlanka", and the "Telefon:" field is filled with "111222333". The buttons "Zapisz", "Uaktualnij", and "Wyczyść" remain at the bottom.

Użytkownik:



Konrad Boroń
Student



admin
Administrator

Panel

Klienci

Użytkownik

Samochody

Sprzedaj


Wyloguj

Użytkownicy

+

No	ID	Nazwa	Adres	Telefon	Rola	Data urodzenia	Hasło
1	1	admin	Wadowice	096541234	Administrator	01.03.2001	admin
2	6	kasjer	Krakow	67887654567	Cashier	01.02.2000	kasjer

Projekt wykonał
Konrad Boroń



Dzienny raport

0,00

SALE

Dzienna sprzedaż

Całkowita dzienna sprzedaż

02:19:07

Do zakładki użytkownik ma dostęp tylko administrator. Ponieważ ta karta służy do dodawania nowego użytkownika po kliknięciu w plusik wyświetli się okno dodawania nowego użytkownika. W zakładce użytkownik wyświetla się tabela z dodanymi użytkownikami. W tej karcie po kliknięciu w długopis również można edytować dane. A po kliknięciu w koszyk dany rekord jest usuwany.

Po kliknięciu w plusik przez administratora w karcie użytkownik

Rejestracja Użytkownika

Nazwa:

Adres:

Telefon:

Rola :

Cashier

Data urodzenia:

piątek .23 czerwca 2023

Hasło:

Zapisz

Uaktualnij

Wyczyść

Po kliknięciu w długopis przez administratora w karcie użytkownik

Rejestracja Użytkownika

Nazwa:

admin

Adres:

Wadowice

Telefon:

096541234

Rola :

Administrator

Data urodzenia:

czwartek . 1 marca 2001

Hasło:


Zapisz

Uaktualnij

Wyczyść

Samochody:

Konrad Boroń
Student


admin
Administrator

Panel

Klienci

Użytkownik

Samochody

Sprzedaj

Wyloguj


Samochody

+

Wyszukaj

No	Pcode	Marka	Model	Typ nadwozia	Ilość	Cena	
1	2	Fiat	126p	Coupe	1	15000,00	
2	5	BMW	E46	Coupe	1	23000,00	
3	7	Honda	TypeR	Sedan	3	500000,00	
4	8	BMW	M3	Kombi	2	630000,00	
5	9	BMW	X5M	Suv	4	556000,00	
6	10	BMW	Seri 3	Sedan	2	50000,00	
7	11	Volvo	S60	Sedan	3	15000,00	
8	12	Audi	A4	Kombi	2	30000,00	
9	13	Jaguar	F-Type	Coupe	1	205000,00	

Projekt wykonał
Konrad Boroń



Dzienny raport

0,00

SALE

Dzienna sprzedaż

Całkowita dzienna sprzedaż

03:16:56

W karcie samochody w tabeli wyświetlane są dostępne samochody do sprzedaży. Po kliknięciu w plusik można dodać nowy samochód do sprzedania. Po kliknięciu w długopis można edytować dane dotyczące wybranego samochodu. Po kliknięciu w kosz można usunąć wybrany samochód.

26

Po kliknięciu w plusik w karcie samochodu

Rejestracja samochodu

Marka:

Model:

Typ nadwozia:

Coupe

Ilość:

Cena:

Zapisz

Uaktualnij

Wyczyść

Po kliknięciu w długopis w karcie samochodu

Rejestracja samochodu

Marka:

Model:

Typ nadwozia:

Coupe

Ilość:


Cena:

Zapisz


Uaktualnij

Wyczyść

Sprzedaj:



Konrad Boroń
Student



admin
Administrator

Panel

Klienci

Uzytkownik

Samochody

Sprzedaj

Wyloguj

Sprzedaj

+

Produkt


Translaction no: 20230623100

Cena całkowita \$: 0,00

Sprzedaj

No	Pcode	Nazwa	Ilość	Cena	Całkowity	Nazwa klienta	Kasjer
----	-------	-------	-------	------	-----------	---------------	--------

Projekt wykonał
Konrad Boroń



Dzienny raport

0,00

SALE

Dzienna sprzedaż

Całkowita dzienna sprzedaż

02:19:37

W karcie sprzedaj jest pusta tabela, lecz po kliknięciu w plusik i zaznaczeniu wybranego samochodu i zatwierdzeniu wyboru zostanie dodany i będzie wyświetlał się w tabeli.


Lista samochodów możliwych do dodania po kliknięciu w plusik:


No	Pcode	Nazwa	Model	Typ nadwozia	Cena	Select
1	2	Fiat	126p	Coupe	15000,00	<input checked="" type="checkbox"/>
2	5	BMW	E46	Coupe	23000,00	<input checked="" type="checkbox"/>
3	7	Honda	TypeR	Sedan	500000,00	<input type="checkbox"/>
4	8	BMW	M3	Kombi	630000,00	<input type="checkbox"/>
5	9	BMW	X5M	Suv	556000,00	<input type="checkbox"/>
6	10	BMW	Seri 3	Sedan	50000,00	<input type="checkbox"/>
7	11	Volvo	S60	Sedan	15000,00	<input type="checkbox"/>
8	12	Audi	A4	Kombi	30000,00	<input checked="" type="checkbox"/>
9	13	Jaguar	F-Type	Coupe	205000,00	<input type="checkbox"/>
10	14	Mercedes-benz	CLA	Sedan	150000,00	<input type="checkbox"/>

Search here

Zatwierdź

Sprzedaj pod dodaniu samochodów:

**Konrad Boroń**
Student


admin
Administrator

Panel

Klienci

Użytkownik

Samochody

Sprzedaj

Wyloguj

Sprzedaj

+

Produkt


Transaction no: 20230623100

Cena całkowita \$: 68 000,00

Sprzedaj

No	Pcode	Nazwa	Ilość	Cena	Całkowity	Nazwa klienta	Kasjer		
1	2	Fiat	1	15000,00	15000,00		admin	↑	↓
2	5	BMW	1	23000,00	23000,00		admin	↑	↓
3	12	Audi	1	30000,00	30000,00		admin	↑	↓

Projekt wykonał
Konrad Boroń



Dzienny raport

0,00

SALE

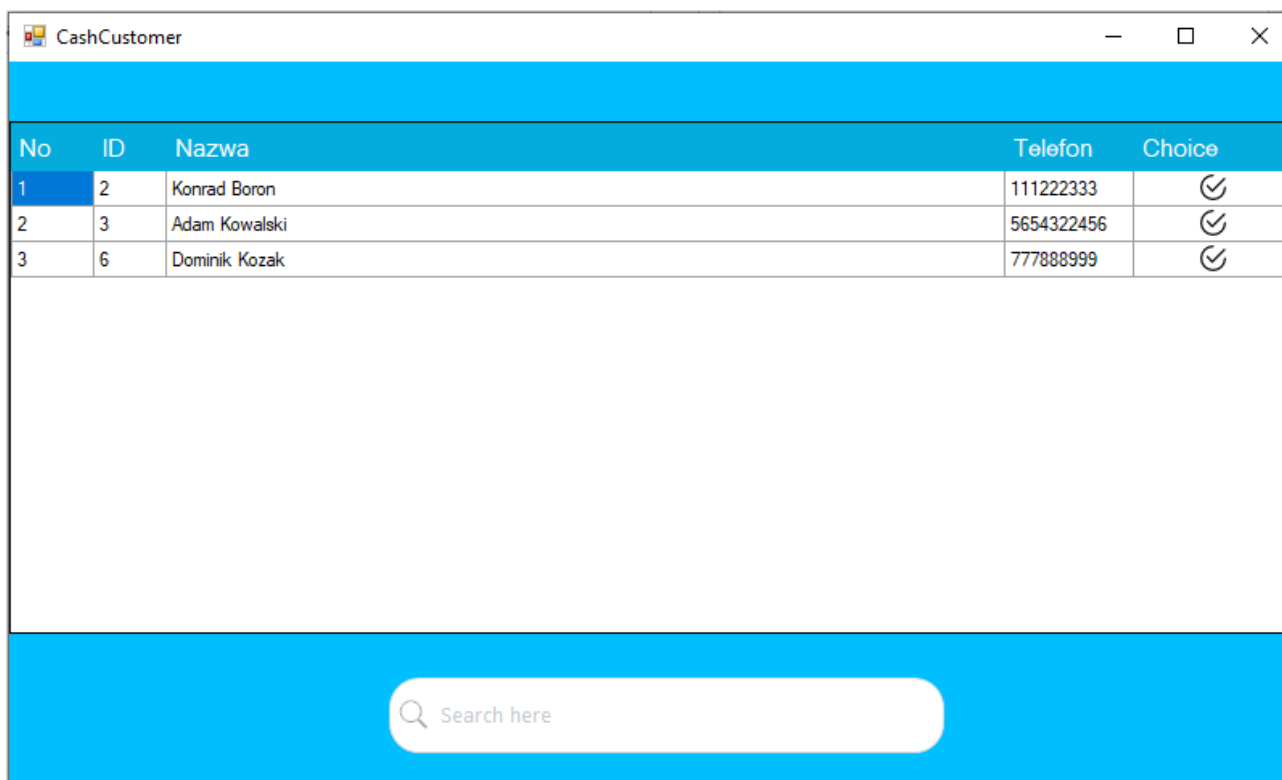
Dzienna sprzedaż

Całkowita dzienna sprzedaż

03:36:13

Po dodaniu samochodów do tabeli przez sprzedawcę można zwiększyć ilość sprzedawanego samochodu lub zmniejszyć. W przypadku gdy samochodu danego typu będzie tylko jeden w bazie pojawi się informacja że do sprzedania jest tylko jeden samochód. A w przypadku gdy użytkownik będzie chciał zejść poniżej jednego samochodu pojawi się informacja o usunięciu samochodu z list sprzedaży. Klikając w koszyk jest możliwość od razu usunąć sprzedawany samochód z tabeli sprzedawanych samochodów. po prawej stronie jest informacja ile będą kosztowały wszystkie samochody które klient chce kupić. A po kliknięciu w przycisk sprzedaj zostanie wyświetlona informacja któremu klientowi chce się samochód sprzedać po wybraniu klienta samochody zostaną sprzedane a baza zaktualizowana o mniejszą ilość wybranych samochodów. Po sprzedaży samochodów w codziennym raporcie kwota zostanie zaktualizowana o kwotę ze sprzedaży samochodów.

Po kliknięciu w przycisk sprzedaj w karcie sprzedaj



The screenshot shows a window titled "CashCustomer" with a blue header and footer. The main area contains a table with the following data:

No	ID	Nazwa	Telefon	Choice
1	2	Konrad Boron	111222333	☑
2	3	Adam Kowalski	5654322456	☑
3	6	Dominik Kozak	777888999	☑

Below the table is a large white search bar with a magnifying glass icon and the text "Search here".

Zaktualizowany codzienny raport o dziennej sprzedaży



10. Wykorzystanie źródeł

Guna.UI2.WinForms- https://www.youtube.com/watch?v=PbrJJlqqiGg&ab_channel=GunaUI

Dokumentacja o tworzeniu bazy danych- <https://learn.microsoft.com/pl-pl/visualstudio/data-tools/create-a-simple-data-application-by-using-adonet?view=vs-2022&tabs=csharp>

<https://www.youtube.com/@MyCodeSpace1/featured>

Ikony: <https://www.flaticon.com/>