



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA

Aplikacje internetowe

Dokumentacja projektu strony rezerwacji samochodów

Wykonał:

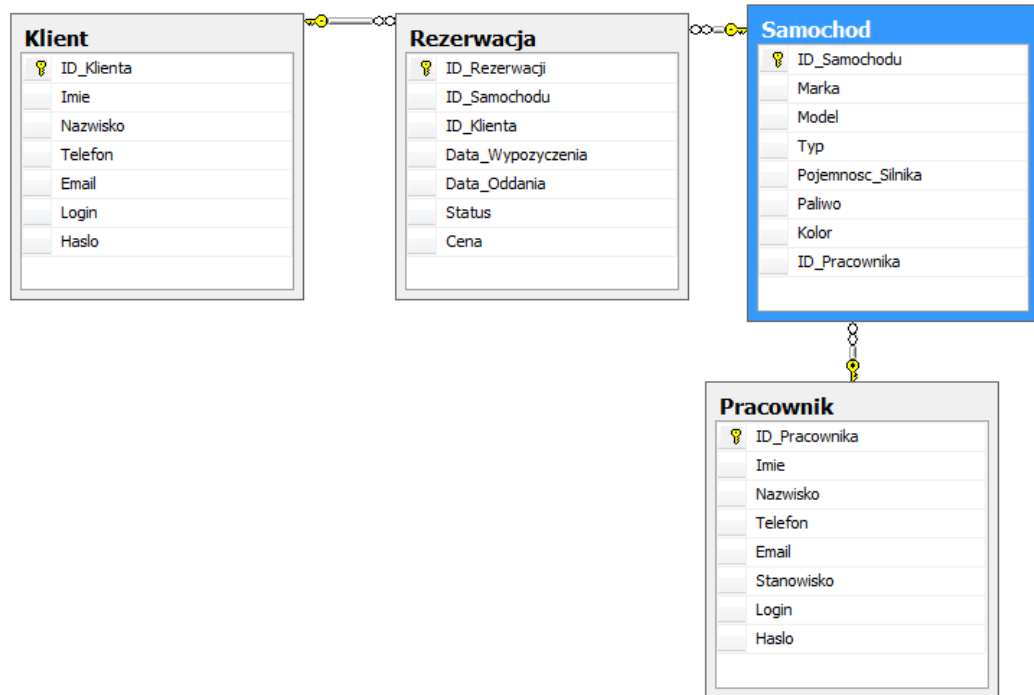
Marcin Kuźniar

grupa L5

1. Projekt

Strona internetowa będzie miała za zadanie zrealizowanie serwisu rezerwacji samochodów z wypożyczalni.

Do tego celu będę wykorzystywał bazę danych, której diagram ERD wygląda jak poniżej:

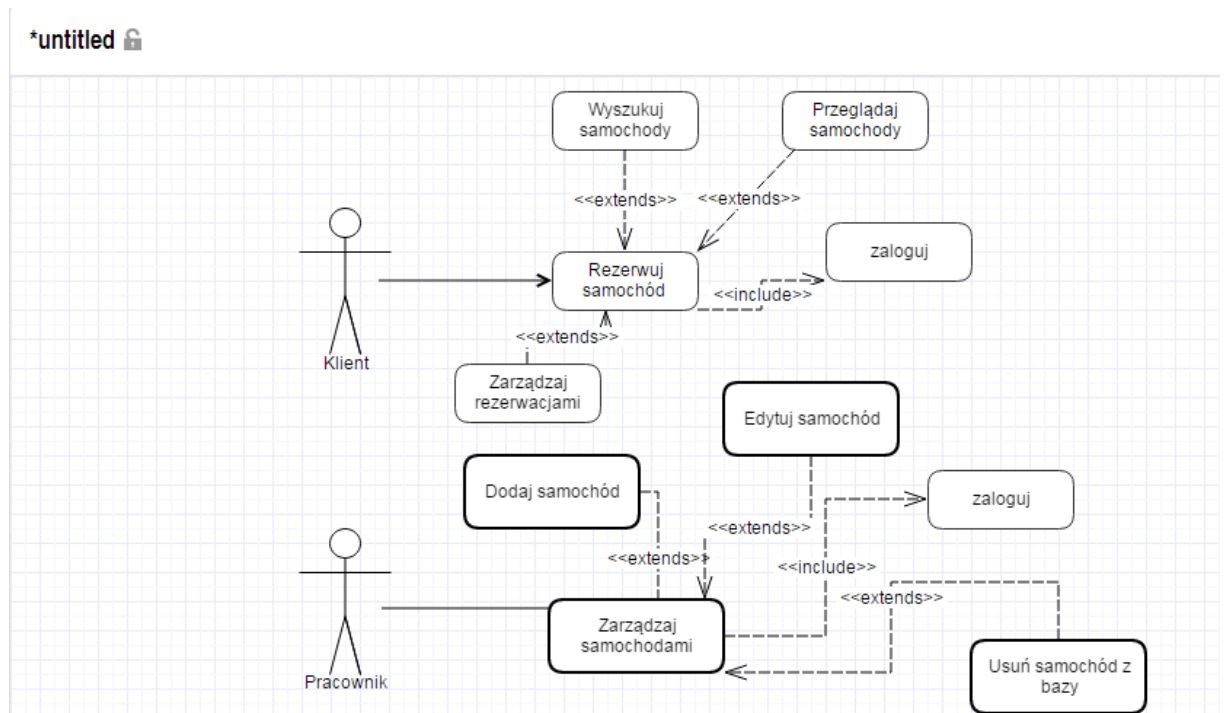


Oczywiście jest to zarys, który może jeszcze ulec niewielkim zmianom.

Planuję na tym etapie, że strona będzie realizować następujące funkcjonalności:

- ✓ pracownik loguje się do systemu (podając login i hasło),
- ✓ pracownik może dodawać kolejne samochody (określając markę, model, typ, pojemność silnika, rodzaj paliwa, kolor, dodawać opis),
- ✓ klient rejestruje się do systemu, loguje się (za pomocą loginu i hasła)
- ✓ klient może przeglądać udostępnione przez serwis samochody,
- ✓ klient może wyszukiwać samochody za pomocą parametrów,
- ✓ klient może dokonywać rezerwacji samochodów,
- ✓ klient może zarządzać swoimi rezerwacjami, usuwać

Poniżej przedstawiono diagram UML:





Oczywiście jest to wersja wstępna, która podczas pracy może podlegać zmianom.


2. Realizacja projektu


- Przygotowanie bazy danych

Realizację projektu rozpocząłem od wygenerowania bazy danych, w której możliwe będzie przechowywanie wszelkich informacji potrzebnych do obsługi mojej aplikacji internetowej. Do tego celu wykorzystałem program SQL Server Management Studio. W związku z tym, że to nie pierwszy projekt dla którego wykorzystuję wymienione wyżej środowisko przeszedłem od razu do dodania nowej bazy o nazwie „**Wyp_samochodow**”. Następnie utworzyłem kolejne cztery tabele: Klient, Pracownik, Samochod, Rezerwacja. Na tym etapie należy określić potrzebne atrybuty oraz ich typ danych, przy pomocy których reprezentowane będą wartości. Jest to bardzo istotne z racji pojawiających się konfliktów w typach danych a późniejsza modyfikacja jest utrudniona. Tutaj należy również wskazać właściwy klucz główny, oraz ustawić jego automatyczną inkrementację. Encje wygenerowanej bazy danych wyglądają następująco:

BAZYL\PROJEKT.W...dow - dbo.Klient			
	Column Name	Data Type	Allow Nulls
	ID_Klienta	numeric(18, 0)	<input type="checkbox"/>
	Imie	nvarchar(50)	<input type="checkbox"/>
	Nazwisko	nvarchar(50)	<input type="checkbox"/>
	Telefon	numeric(18, 0)	<input type="checkbox"/>
	Email	nvarchar(50)	<input type="checkbox"/>
	Login	nvarchar(50)	<input type="checkbox"/>
	Haslo	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

BAZYL\PROJEKT.Wy... - dbo.Pracownik			
	Column Name	Data Type	Allow Nulls
	ID_Pracownika	numeric(18, 0)	<input type="checkbox"/>
	Imie	nvarchar(50)	<input type="checkbox"/>
	Nazwisko	nvarchar(50)	<input type="checkbox"/>
	Telefon	nvarchar(50)	<input type="checkbox"/>
	Email	nvarchar(50)	<input type="checkbox"/>
	Stanowisko	nvarchar(50)	<input type="checkbox"/>
	Login	nvarchar(50)	<input type="checkbox"/>
	Haslo	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

BAZYL\PROJEKT.W...- dbo.Rezerwacja			
	Column Name	Data Type	Allow Nulls
	ID_Rezerwacji	numeric(18, 0)	<input type="checkbox"/>
	ID_Samochodu	numeric(18, 0)	<input type="checkbox"/>
	ID_Klienta	numeric(18, 0)	<input type="checkbox"/>
	Data_Wypozyczenia	date	<input type="checkbox"/>
	Data_Oddania	date	<input type="checkbox"/>
			<input type="checkbox"/>

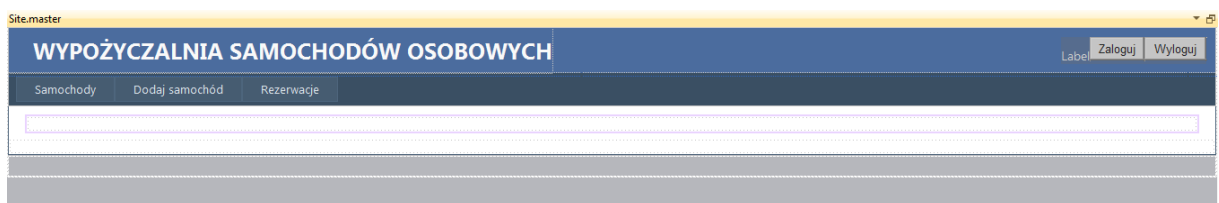
BAZYL\PROJEKT.... - dbo.Samochod			
	Column Name	Data Type	Allow Nulls
	ID_Samochodu	numeric(18, 0)	<input type="checkbox"/>
	Marka	nvarchar(50)	<input type="checkbox"/>
	Model	nvarchar(50)	<input type="checkbox"/>
	Typ	nvarchar(50)	<input type="checkbox"/>
	Rok_Produkcji	numeric(18, 0)	<input type="checkbox"/>
	Pojemnosc_Silnika	numeric(18, 0)	<input type="checkbox"/>
	Paliwo	nvarchar(50)	<input type="checkbox"/>
	Kolor	nvarchar(50)	<input type="checkbox"/>
	Opis	nvarchar(MAX)	<input type="checkbox"/>
	ID_Pracownika	numeric(18, 0)	<input type="checkbox"/>
			<input type="checkbox"/>

Kolejnym krokiem było utworzenie właściwego diagramu związków encji, a więc połączenie encji właściwymi relacjami (klucz główny z kluczem obcym). Dzięki temu w pełni możemy wykorzystywać potencjał bazy przeszukując jej zawartość jednym poleceniem SQL. Diagram związków encji pozostał taki jak podczas pracy nad projektem i został zamieszczony w pierwszym punkcie opisu. Tak przygotowana baza jest już gotowa do użycia i nie wymaga dalszej obsługi.

- Przygotowanie aplikacji

Aplikacja internetowa została napisana w ASP.NET wykorzystując środowisko Microsoft Visual Studio 2010. Podczas pracy wykorzystałem szablon strony dostępny już w programie, który odpowiednio skonfigurowałem a następnie rozszerzyłem o kolejne strony i funkcjonalności. Gotowa aplikacja zbudowana jest z 5 stron: strony rejestracji użytkownika, strony logowania, strony przeglądania dostępnych samochodów, strony dodawania nowych samochodów oraz strony umożliwiającej przeglądanie i zarządzanie rezerwacjami.

Wszystkie strony łączy jeden główny szablon (widoczny cały czas na ekranie) - Site.master, dzięki któremu w każdej chwili możemy dostać się do pożądanej części aplikacji. Na górze umieszczony jest nagłówek informujący na jakiej stronie się znajdujemy, a pod nim kolejne zakładki. Po prawej stronie znajduje się przycisk „Zaloguj” przenoszący na stronę logowania. Obok niego znajduje się etykieta, której zapisane jest powitanie zalogowanego użytkownika. Dla zalogowanego użytkownika natomiast przycisk „Zaloguj” zastąpiony jest klawiszem „Wyloguj”, dzięki któremu możliwe jest opuszczenie aplikacji. Szablon został przedstawiony poniżej:



✓ Strona logowania

Logowanie do systemu dostępne jest w następującej formule:

Samochody

Dodaj samochód

Rezerwacje

ZALOGUJ SIĘ

Podaj swój swoje dane logowania. [Zarejestruj się](#) jeżeli nie posiadasz konta w naszym serwisie.

Dane logowania

Login:

Hasło:

☐ Zapamiętaj mnie

Zaloguj

Strona logowania została w znacznym stopniu niezmieniona w stosunku do tego co oferuje nam przykładowy projekt, z którego skorzystałem. Na górze strony znalazł się nagłówek „Zaloguj się”, pod nim krótki opis, a w nim hiperłącze przenoszące na stronę rejestracji użytkowników nie posiadających konta w serwisie. Poniżej pozostała etykieta, pole tekstowe oraz kontrolka walidacji:

```
<p>  
<asp:Label ID="UserNameLabel" runat="server" AssociatedControlID="UserName">Login:</asp:Label>  
<asp:TextBox ID="UserName" runat="server" CssClass="textEntry"></asp:TextBox>  
<asp:RequiredFieldValidator ID="UserNameRequired" runat="server" ControlToValidate="UserName"  
    CssClass="failureNotification" ErrorMessage="Login jest wymagany." ToolTip="Login jest wymagany."  
    ValidationGroup="LoginUserValidationGroup">*</asp:RequiredFieldValidator>  
</p>
```

Na tym etapie nie musimy tak naprawdę nic robić, bo wszystko zostało już wykonane w szablonie. Wystarczy W miejscu, gdzie zapisana jest wyświetlana wiadomość ostrzeżenia „ErrorMessage” wpisać własny tekst jaki chcemy wyświetlić użytkownikowi w przypadku, gdy wartość wpisana w powiązonym polu tekstowym nie przejdzie walidacji. Można także wprowadzać własne nazwy w miejsca „ID”, wynika to z wyłącznie dobrej woli programisty, gdy chce w jakiś sposób uprościć sobie zadanie przy bardziej zaawansowanym programowaniu, nie jest to jednak konieczne. Następnie znajdziemy kolejny zespół etykiety, powiązanego z nim pola tekstowego oraz kontrolki walidacji dla miejsca, w którym wprowadzane jest hasło:

```

<p>
  <asp:Label ID="PasswordLabel" runat="server" AssociatedControlID="Password">Hasło:</asp:Label>
  <asp:TextBox ID="Password" runat="server" CssClass="passwordEntry" TextMode="Password"></asp:TextBox>
  <asp:RequiredFieldValidator ID="PasswordRequired" runat="server" ControlToValidate="Password"
    CssClass="failureNotification" ErrorMessage="Hasło jest wymagane." ToolTip="Hasło jest wymagane."
    ValidationGroup="LoginUserValidationGroup">*</asp:RequiredFieldValidator>
</p>

```

Także i dla tego zbioru modyfikacji uległ wyłącznie parametr „ErrorMessage”. Poniżej zlokalizowana jest kontrolka „CheckBox” wraz z powiązaną z nim etykietą „RememberMeLabel”:

```

<p>
  <asp:CheckBox ID="RememberMe" runat="server"
    oncheckedchanged="RememberMe_CheckedChanged"/>
  <asp:Label ID="RememberMeLabel" runat="server" AssociatedControlID="RememberMe"
    CssClass="inline">Zapamiętaj mnie</asp:Label>
</p>

```

Na samym dole znalazł się najważniejszy punkt tej strony - przycisk „Zaloguj”.

```

<p class="submitButton">
  <asp:Button ID="LoginButton" runat="server" Text="Zaloguj"
    ValidationGroup="LoginUserValidationGroup" onclick="LoginButton_Click"/>
</p>

```

Po kliknięciu na przycisk obsługiwana jest walidacja opisanych pól tekstowych powiązanych z „ValidationGroup=„LoginUserValidationGroup”, oraz metoda „LoginButton_Click”. Do właściwej obsługi walidacji potrzebna jest właściwa klasa, oraz kontrolka:

```

<span class="failureNotification">
  <asp:Literal ID="FailureText" runat="server"></asp:Literal>
</span>
<asp:ValidationSummary ID="LoginUserValidationSummary" runat="server" CssClass="failureNotification"
  ValidationGroup="LoginUserValidationGroup"/>

```

Jednak to także jest już właściwie dodane i skonfigurowane w projekcie, z którego skorzystałem.

Najważniejsza część związana z zrealizowanym przeze mnie logowaniem odbywa się w metodzie:

```
protected void LoginButton_Click(object sender, EventArgs e)
```

Na początku metody tworze połączenie z bazą danych, oraz dwa zapytania SQL:

```

SqlConnection cnn = new SqlConnection("Data Source=BAZYL;Initial Catalog=Wyp_Samochodow;Integrated Security=True");
SqlCommand cmd = new SqlCommand("SELECT * FROM Klient WHERE Login='" + UserName.Text + "' AND Haslo='" + Password.Text + "'", cnn);
SqlCommand cmd2 = new SqlCommand("SELECT * FROM Pracownik WHERE Login='" + UserName.Text + "' AND Haslo='" + Password.Text + "'", cnn);

```

Za pomocą zapytania cmd przeszukuję krotki w tabeli „Klient”, gdzie „Login” i „Hasło” wyglądają identycznie jak w danych wprowadzonych w polach tekstowych strony. Drugie natomiast wykorzystuję do przeszukania tabeli „Pracownik” pod tym samym kątem.

```
SqlDataReader dr;  
SqlDataReader dr2;  
cnn.Open();  
dr = cmd.ExecuteReader();  
dr.Read();  
  
if (dr.HasRows)  
{  
    Session["Login"] = UserName.Text;  
    Session["ID"] = dr["ID_Klienta"].ToString();  
  
    Response.Redirect("About.aspx");  
}  
else  
{  
    BłędneLogowanie.Text = "NIEUDANE LOGOWANIE! Sprawdź wprowadzane dane!!";  
    BłędneLogowanie.Visible = true;  
}  
dr.Close();
```

Następnie tworzę dwa obiekty klasy SqlDataReader otwieram połączenie z bazą i wywołuję pierwszą komendę SqlCommand, po czym odczytuję kolejne rekordy za pomocą metody Read(). Poniżej w instrukcji warunkowej if sprawdzam czy został odnaleziony rekord z wprowadzonymi przez użytkownika danymi. Jeśli tak do pierwszego parametru Session[„Login”] zapisuję wprowadzony przez użytkownika tekst w polu tekstowym „Login”. Mogłbym zapisać również to co zostało odnalezione w bazie, lecz to ten sam tekst. Do kolejnego parametru Session[„ID”] zapisuję ID_Klienta z rekordu odnalezonego w bazie. Dla takiego sposobu obsługi logowania wystarczyłby mi jeden z parametrów, jednak login został wykorzystany do przywitania użytkownika w części Site.master, natomiast „ID” zalogowanego klienta będę potrzebował podczas zapisu kolejnych danych do bazy danych w aplikacji. Po udanym zalogowaniu następuje przekierowanie na stronę „About.aspx”, na której znajdują się rezerwacje zalogowanego klienta oraz zamknięcie obiektu klasy SqlDataReader. W przypadku błędnego wprowadzenia danych następuje wyświetlenie odpowiedniego komunikatu w etykiecie o ID=BłędneLogowanie umieszczonej na stronie:

```
<p>  
    <asp:Label ID="BłędneLogowanie" runat="server" Text="Label" Visible="False" ForeColor="Red"></asp:Label>  
</p>
```


Jest to osobne powiadomienie nie mające nic wspólnego z walidacją, której komunikaty wyświetlane są w przypadku niewypełnienia jednego lub obu pól podczas wybierania przycisku „Zaloguj”.

Następnie wykonuje identyczny kod przeszukując tabele Pracownik:

```
dr2 = cmd2.ExecuteReader();
dr2.Read();

if (dr2.HasRows)
{
    Session["Login2"] = UserName.Text;
    Session["ID"] = dr2["ID_Pracownika"].ToString();

    Response.Redirect("Dodaj_Samochod.aspx");
}
else
{
    BledneLogowanie.Text = "NIEUDANE LOGOWANIE! Sprawdź wprowadzane dane!!";
    BledneLogowanie.Visible = true;
}
dr2.Close();

cnn.Close();
```

Jednak tutaj pierwszy parametr sesji oznaczyłem jako „Login2”, dzięki czemu będę mógł rozróżnić kto zalogował się do systemu. Różnicą jest również to, że po zalogowaniu pracownika ładowana jest strona „DodajSamochod.aspx”.

Na tym etapie zostało zrealizowane rozróżnienie zalogowanego użytkownika, lecz nie ma możliwości skasowania parametrów sesji. Zostało to wykonane na stronie Site.master, gdzie znajduje się przycisk wyloguj:

```
protected void Wyloguj_Click(object sender, EventArgs e)
{
    Session.RemoveAll();
    Response.Redirect("~/Default.aspx");
}
```

To, który przycisk jest widoczny na ekranie uzależnione jest w metodzie PageLoad strony Site.master w następujący sposób:

```

if (Session["Login"] == null && Session["Login2"] == null)
{
    Wyloguj.Visible = false;
    Zaloguj.Visible = true;
}
else
{
    Wyloguj.Visible = true;
    Zaloguj.Visible = false;
}

```

Także z logowaniem związana jest etykieta powitania, której tekst pobierany jest z każdej strony za pomocą metody:

```

public string LabelLoginText
{
    get
    {
        return LabelLogin.Text;
    }
    set
    {
        LabelLogin.Text = value;
    }
}

```

✓ Strona rejestracji użytkownika

Strona umożliwiająca rejestrację użytkowników prezentuje się w następujący sposób:

Samochody

Dodaj samochód

Rezerwacje

ZAREJESTRUJ NOWE KONTO, ABY MÓC KORZYSTAĆ Z WYPOŻYCZALNI !

Wypełnij poniższy formularz.

Dane użytkownika

Imię:

Nazwisko:

Telefon:

Email:

Login:

Hasło:

Potwierdź hasło:

Status użytkownika.

Klient

▼

Utwórz użytkownika

Zawiera ona krótką informację mówiącą o zakresie działań jakie można wykonać na stronie. Zbudowana jest głównie z etykiet oraz powiązanych z nimi pól tekstowych. Służą one do wprowadzania danych niezbędnych do utworzenia nowego użytkownika. Ostatnie pole w postaci listy określa typ użytkownika, a co za tym idzie determinuje zakres funkcjonalności dostępnych na kolejnych stronach. Klawisz „Utwórz użytkownika” służy do uruchomienia polecenia SQL zapisującego dane w określonej encji bazy danych (Klient lub Pracownik w zależności od wybranego statusu użytkownika). Uprzednio jednak następuje sprawdzenie czy wszystkie pola zostały uzupełnione, oraz poprawność powtórzenia hasła. Wszystkie odstępstwa od tego zostaną wyświetlone użytkownikowi w postaci ostrzeżeń po próbie utworzenia użytkownika.

- Podanie imienia jest wymagane.
- Podanie nazwiska jest wymagane.
- Podanie numeru kontaktowego jest wymagane.
- Kod pocztowy jest wymagany.
- Podaj login.
- Podaj hasło.
- Kontrola poprawności - konieczne jest powtórzenie hasła.

Dane użytkownika

Imie:

*

Nazwisko:

*

Walidacja jest tu niezbędna ponieważ w przypadku pozostawienia niezapisanego jednego z pól, u nastąpi błąd podczas zapisu do bazy danych i działanie aplikacji zostanie zatrzymane.

✓ Strona dodawania samochodów

Prezentuje się następująco:

Samochody
Dodaj samochód
Rezerwacje

Podaj dane dotyczące nowego samochodu

Marka:

Model:

Typ:

Rok produkcji:

Pojemność silnika:

Paliwo:

Kolor:

Dodatkowe wyposażenie:

Dodaj samochód

Także i ta strona składa się głównie z etykiet, pól tekstowych oraz list wyboru umożliwiających dodanie samochodu do serwisu. Także i tutaj została zrealizowana walidacja, zabezpieczająca przed przerwaniem pracy serwisu i prawidłowym zapisem pozycji w bazie danych. Jest to jednak pierwsza ze stron, do której dostęp został ograniczony wyłącznie dla zalogowanych użytkowników, dodatkowo musi to być pracownik serwisu (odpowiednie sprawdzenie parametru sesji Login i Login2 w metodzie PageLoad). Po kliknięciu na zakładkę przez niezalogowanego użytkownika zostaje on

automatycznie przeniesiony na stronę logowania, natomiast w przypadku wybrania przez klienta, ładowana jest strona dostępnych samochodów.

✓ Strona dostępnych samochodów

Przeglądanie samochodów dostępne jest na poniższej stronie:

WYPOŻYCZALNIA SAMOCHODÓW OSOBOWYCH

Zaloguj

SamochodyDodaj samochódRezerwacje

PRZEGŁĄDAJ DOSTĘPNE SAMOCHODY!

Wprowadź dane wyszukiwania !

Baza dostępnych samochodów

Marka	Model	Typ	Rok produkcji	Pojemność silnika	Paliwo	Kolor	Opis
Audi	TT	Sport / Coupe	2010	2500	Benzyna	Biały	Pełna opcja.
Audi	A3	Kabriolet	2008	2500	Benzyna	Złoty.	Pełna opcja
Audi	A3	Sedan	2015	3600	Benzyna	Biały	Alu.full opcja
Peugeot	306	Kombi	2014	2000	Diesel	Pomarańczowy	Elektryczne szyby, elektryczne lusterka, wspomaganie kierownicy.
Porsche	Panamera	Sport / Coupe	2014	3500	Benzyna	Czarny	Pełne dodatkowe wyposażenie.
Renault	Laguna	Sedan	2005	3000	Benzyna	Srebrny	Pełna opcja.
Rover	C25	Hatchback	1999	1500	Benzyna	Srebrny	Wspomaganie, elektryczne szyby, podgrzewane lusterka.
Seat	Leon	Kabriolet	2004	2000	Benzyna	Zielony	Elektryczne szyby, wspomaganie kierownicy
Seat	Toledo	Hatchback	2008	1900	Diesel	Czarny	Full opcja

Powyższy widok strony jest dostępny dla niezalogowanych użytkowników korzystających ze strony. Dostępna jest wyłącznie lista dostępnych samochodów, bez dodatkowych informacji czy funkcji. Do tego celu wykorzystałem kontrolkę GridView, do której powiązałem kolejną kontrolkę SQL generującą zapytanie do bazy danych zwracające całą zawartość encji Samochod. Obiekt GridView jest mało efektywnym sposobem prezentacji danych, lecz w pełni spełnia swoją funkcję, mając szereg rozszerzeń potrzebnych przy kolejnych operacjach na danych, stąd chętnie z niej korzystałem.

Strona została w znaczny sposób rozbudowana dla zalogowanych klientów i prezentuje się w następujący sposób:

Wypożyczalnia Samochodów Osobowych
Jesteś zalogowany jako klient! Witaj :tomekk Wyloguj

Samochody
Dodaj samochód
Rezerwacje

PRZEGLĄDAJ DOSTĘPNE SAMOCHODY!

Wprowadź dane wyszukiwania !

Wprowadź markę auta:
Wyszukaj

Baza dostępnych samochodów

Pokaż wszystkie

Id	Marka	Model	Nadwozie	Rocznik	Pojemność	Paliwo	Kolor	Wyposażenie	Data od	Data do	Rezerwuj
6	Seat	Leon	Kabriolet	2004	2000	Benzyna	Zielony	Elektryczne szyby, wspomaganie kierownicy			Rezerwuj
9	Renault	Laguna	Sedan	2005	3000	Benzyna	Srebrny	Pełna opcja.			Rezerwuj
25	Porsche	Panamera	Sport / Coupe	2014	3500	Benzyna	Czarny	Pełne dodatkowe wyposażenie.			Rezerwuj
27	Audi	TT	Sport / Coupe	2010	2500	Benzyna	Biały	Pełna opcja.			Rezerwuj
28	Audi	A3	Kabriolet	2008	2500	Benzyna	Złoty.	Pełna opcja			Rezerwuj
30	Audi	A3	Sedan	2015	3600	Benzyna	Biały	Alu.full opcja			Rezerwuj
32	Rover	C25	Hatchback	1999	1500	Benzyna	Srebrny	Wspomaganie, elektryczne szyby, podgrzewane lusterka.			Rezerwuj

Główny akapit - Baza dostępnych samochodów- zawiera obiekt GridView oraz przycisk „Pokaż wszystkie”. W tym wydaniu GridView jest zdecydowanie bardziej skomplikowany. Oprócz prostego wyświetlania dodałem dodatkowe pola (TemplateField), w których wykorzystałem doinstalowane kontrolki kalendarza AJAX pozwalające wyświetlać datę pożądaną rezerwacji w formie kalendarza. Do tego celu wykorzystałem także przycisk obrazkowy (wyświetlający kalendarz) oraz pole tekstowe wyświetlające wybraną datę. W ostatniej kolumnie znalazł się przycisk „Rezerwuj”, dzięki któremu można zarezerwować wybrany samochód na określony w poprzednich kolumnach czas. Podczas obsługi przycisku uruchamiana jest walidacja pól powiązanych z datą, a także co bardzo ważne przeszukiwana jest baza danych czy nie ma już rezerwacji danego samochodu w podanych przedziałach czasowych. W przypadku pominięcia jednej, lub obu dat podczas wybrania rezerwacji, bądź w przypadku zajęcia samochodu w pożądanym terminie użytkownik jest o tym informowany stosownym komunikatem:

Baza dostępnych samochodów

Pokaż wszystkie

Samochód w tym przedziale czasu został już zarezerwowany ! Prosimy o zmianę terminu lub wybór innego modelu..

Id	Marka	Model	Nadwozie	Rocznik	Pojemność	Paliwo	Kolor	Wyposażenie	Data od	Data do	Rezerwuj
6	Seat	Leon	Kabriolet	2004	2000	Benzyna	Zielony	Elektryczne szyby, wspomaganie kierownicy	2015-12-06	2015-12-17	Rezerwuj

Baza dostępnych samochodów

Pokaż wszystkie

Wprowadź daty !

Id	Marka	Model	Nadwozie	Rocznik	Pojemność	Paliwo	Kolor	Wyposażenie	Data od	Data do	Rezerwuj
6	Seat	Leon	Kabriolet	2004	2000	Benzyna	Zielony	Elektryczne szyby, wspomaganie kierownicy	2015-12-06		Rezerwuj

Na górze w osobnym akapicie znalazł się zbiór elementów umożliwiający wyszukiwanie (wyświetlenie) samochodów danej marki. W jego skład wchodzi etykieta, pole tekstowe oraz przycisk „Wyszukaj”. Po wpisaniu przez użytkownika szukanej marki oraz wybraniu

przycisku wyszukiwania jego oczom wyświetla się osobny GridView. W celu ponownego wyświetlenia pełnego zbioru należy wybrać przycisk „Pokaż wszystkie”.

Dla zalogowanych pracowników utworzyłem kolejny GridView:

Wypożyczalnia Samochodów Osobowych

Jesteś zalogowany jako pracownik! Witaj :andrzej Wyloguj

SamochodyDodaj samochódRezerwacje

Przełączaj dostępne samochody!

Wprowadź dane wyszukiwania !

Baza dostępnych samochodów

ID	Marka	Model	Typ	Rok produkcji	Pojemność silnika	Paliwo	Kolor	Opis	ID pracownika	Usuń
27	Audi	TT	Sport / Coupe	2010	2500	Benzyna	Biały	Pełna opcja.	1	Usuń
28	Audi	A3	Kabriolet	2008	2500	Benzyna	Żółty.	Pełna opcja	1	Usuń
30	Audi	A3	Sedan	2015	3600	Benzyna	Biały	Alu.full opcja	1	Usuń
35	Peugeot	306	Kombi	2014	2000	Diesel	Pomarańczowy	Elektryczne szyby, elektryczne lusterka, wspomaganie kierownicy.	3	Usuń
25	Porsche	Panamera	Sport / Coupe	2014	3500	Benzyna	Czarny	Pełne dodatkowe wyposażenie.	1	Usuń
9	Renault	Laguna	Sedan	2005	3000	Benzyna	Srebrny	Pełna opcja.	1	Usuń
32	Rover	C25	Hatchback	1999	1500	Benzyna	Srebrny	Wspomaganie, elektryczne szyby, podgrzewane lusterka.	1	Usuń
6	Seat	Leon	Kabriolet	2004	2000	Benzyna	Zielony	Elektryczne szyby, wspomaganie kierownicy	1	Usuń
36	Seat	Toledo	Hatchback	2008	1900	Diesel	Czarny	Full opcja	3	Usuń

Nie ma dostępnego trybu wyszukiwania, natomiast ma możliwość zarządzania samochodami - dokładnie ich usuwania. Jest to realizowane za pomocą klawisza „Usuń” znajdującego się w ostatniej kolumnie tabeli. Operacja usunięcia samochodu wymagała jednak wcześniejszego usunięcia wszelkich powiązań w bazie danych, a więc w moim przypadku wszystkich rezerwacji na dany pojazd. Zrealizowałem to właśnie w ten sposób inaczej pracownik musiałby najpierw odnaleźć i usunąć każdą rezerwację teraz robi to razem z autem.

✓ Strona z rezerwacjami

Strona zarządzania rezerwacjami jest ostatnią przygotowaną przeze mnie stroną i dla klienta prezentuje się następująco:

Wypożyczalnia Samochodów Osobowych

Jesteś zalogowany jako klient! Witaj :tomekk Wyloguj

Samochody

Dodaj samochód

Rezerwacje

Przeglądaj rezerwacje.

Id	Marka	Model	Data wypożyczenia	Data zwrotu	Usuń
27	Seat	Leon	2015-12-10	2015-12-19	Usuń
29	Audi	A3	2015-12-29	2015-12-31	Usuń
34	Porsche	Panamera	2015-12-29	2015-12-31	Usuń
35	Rover	C25	2015-12-31	2016-01-13	Usuń

Także i ta strona ma ograniczony dostęp. Niezalogowany użytkownik nie ma do niej dostępu, przy próbie wejścia w tę zakładkę zostaje przeniesiony na stronę logowania. Dla zalogowanego klienta utworzyłem GridView z listą jego rezerwacji. Zapytanie SQL jest

tutaj dość skomplikowane ponieważ pobiera dane z aż trzech tabel naraz. W ostatniej kolumnie znalazł się przycisk „Usuń” pozwalający na usunięcie pożądanej rezerwacji przez użytkownika systemu.

Wypożyczalnia Samochodów Osobowych

Jesteś zalogowany jako pracownik! Witaj :andrzej Wyloguj

Samochody

Dodaj samochód

Rezerwacje

Przeglądaj rezerwacje.

ID rezerwacji	ID samochodu	ID klienta	Data wypożyczenia	Data oddania	Usuń
27	6	1	2015-12-10 00:00:00	2015-12-19 00:00:00	Usuń
29	30	1	2015-12-29 00:00:00	2015-12-31 00:00:00	Usuń
34	25	1	2015-12-29 00:00:00	2015-12-31 00:00:00	Usuń
35	32	1	2015-12-31 00:00:00	2016-01-13 00:00:00	Usuń
36	28	2	2016-01-18 00:00:00	2016-01-28 00:00:00	Usuń
37	32	2	2016-01-25 00:00:00	2016-01-29 00:00:00	Usuń
38	6	2	2016-01-12 00:00:00	2016-01-22 00:00:00	Usuń

Zalogowany pracownik również ma możliwość usuwania pojedynczych rezerwacji, jednak w jego przypadku dostępne ma informacje bardziej „surowe”. Jednak funkcjonalność została zachowana jak dla klienta.

- Podsumowanie

Aplikacja została wykonana zgodnie z założonym projektem. Zostały w niej zawarte wszystkie funkcjonalności oraz dodatkowe, których realizacja okazała się niezbędna lub wynikała z chęci poprawy działania strony.