

Projekt integracja systemów

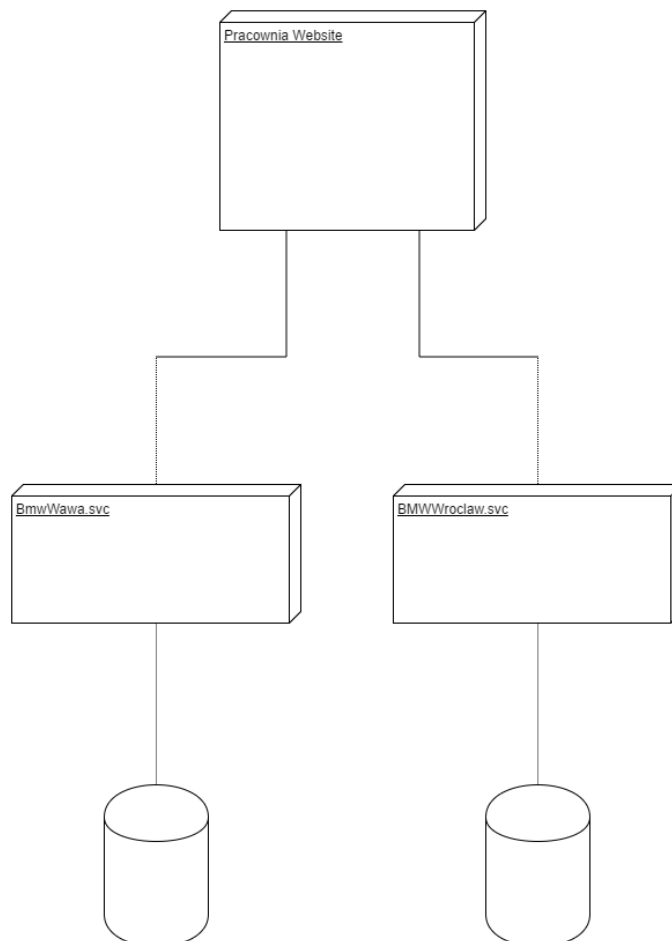
Małgorzata Robak 163765
Adam Gudynowski 164190

Wykorzystane technologie

Do zbudowania aplikacji wykorzystano .NET framework. Aplikacja webowa napisana jest w technologii ASP.NET MVC 5, Webserwisy dostarczające dane to serwisy SOAP napisane przy pomocy technologii WCF. Język programowania to C#. Obie bazy danych funkcjonują na serwerze MySQL.

Opis architektury systemu

System składa się z dwóch webserwisów: BmwWawa.svc oraz BMWWroclaw.svc. Oba serwisy udostępniają informacje o samochodach znajdujących się w dwóch salonach BMW. Trzecim elementem systemu jest Website Pracownia, który pobiera dane z obu webserwisów, a następnie wyświetla je posortowane po cenie w formie tabeli.



Struktura encji

Serwisy

BMWwawa
+ GetCars(): IList<CarWawa>

BMWwroclaw
+ GetCars(): IList<Car>

CarWawa	
NUMBER	Id
VARCHAR	Seria
VARCHAR	Wersja
VARCHAR	Nadwozie
VARCHAR(1)	Silnik
NUMBER	Moc
VARCHAR(1)	Skrzynia
NUMBER	Pojemnosc
NUMBER	Cena
VARCHAR	Kolor

Car	
NUMBER	Id
VARCHAR	Series
VARCHAR	Version
VARCHAR	Body
VARCHAR(1)	FuelType
NUMBER	EngineCapacity
NUMBER	Power
VARCHAR(1)	Gearbox
NUMBER	Price
VARCHAR	Colour

Website

BmwCar	
VARCHAR	Series
VARCHAR	Version
VARCHAR	Body
VARCHAR(1)	FuelType
NUMBER	EngineCapacity
NUMBER	Power
VARCHAR(1)	GearboxType
NUMBER	Price
VARCHAR	Colour
VARCHAR	City

Api Serwisów

XML wygenerowany przy pomocy narzędzia SoapUI

BmwWawa:

<http://localhost:53383/BmwWawa.svc>

Przykładowy request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetCars/>
  </soapenv:Body>
</soapenv:Envelope>
```

Przykładowa odpowiedź (skrótowa)

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetCarsResponse xmlns="http://tempuri.org/">
      <GetCarsResult xmlns:a="http://schemas.datacontract.org/2004/07/BMW_Wawa.DO"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:CarWawa>
          <a:Cena>112500</a:Cena>
          <a:Kolor>Szafirowo czarny</a:Kolor>
          <a:Moc>177</a:Moc>
          <a:Nadwozie>Hatchback / 3dr</a:Nadwozie>
          <a:Pojemnosc>1598</a:Pojemnosc>
          <a:Seria>Seria 1</a:Seria>
          <a:Silnik>66</a:Silnik>
          <a:Skrzynia>77</a:Skrzynia>
          <a:Wersja>120i (177KM) M6</a:Wersja>
        </a:CarWawa>
        <a:CarWawa>
          <a:Cena>135700</a:Cena>
          <a:Kolor>Srebrny Silverstone</a:Kolor>
          <a:Moc>184</a:Moc>
          <a:Nadwozie>Coupe / 3dr</a:Nadwozie>
          <a:Pojemnosc>1997</a:Pojemnosc>
          <a:Seria>Seria 2</a:Seria>
          <a:Silnik>66</a:Silnik>
          <a:Skrzynia>77</a:Skrzynia>
          <a:Wersja>220i (184KM) M6</a:Wersja>
        </a:CarWawa>
      </GetCarsResult>
    </GetCarsResponse>
  </s:Body>
</s:Envelope>
```

BMW Wroclaw:

http://localhost:56854/BMW Wroclaw.svc

Przykładowy request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetCars/>
  </soapenv:Body>
</soapenv:Envelope>
```

Przykładowa odpowiedź (skrótowa)

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetCarsResponse xmlns="http://tempuri.org/">
      <GetCarsResult xmlns:a="http://schemas.datacontract.org/2004/07/BMW_Wroclaw.DAO"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Car>
          <a:Body>Hatchback / 3dr</a:Body>
          <a:Colour>Błękit Estoril</a:Colour>
          <a:EngineCapacity>1499</a:EngineCapacity>
          <a:FuelType>66</a:FuelType>
          <a:GearboxType>77</a:GearboxType>
          <a:Power>109</a:Power>
          <a:Price>94600</a:Price>
        </a:Car>
      </GetCarsResult>
    </GetCarsResponse>
  </s:Body>
</s:Envelope>
```

```

        <a:Series>Seria 1</a:Series>
        <a:Version>116i (109KM) M6</a:Version>
    </a:Car>
    <a:Car>
        <a:Body>MPV / 5dr</a:Body>
        <a:Colour>Jatoba</a:Colour>
        <a:EngineCapacity>1496</a:EngineCapacity>
        <a:FuelType>68</a:FuelType>
        <a:GearboxType>77</a:GearboxType>
        <a:Power>95</a:Power>
        <a:Price>118300</a:Price>
        <a:Series>Seria 2</a:Series>
        <a:Version>214d (95KM) M6</a:Version>
    </a:Car>
</GetCarsResult>
</GetCarsResponse>
</s:Body>
</s:Envelope>

```

Przebieg integracji

Do integracji została wykorzystana biblioteka Automapper.

```

cfg.CreateMap<CarWawa, BmwCar>()
.ForMember(dest => dest.Series, opt => opt.MapFrom(src => src.Seria))
.ForMember(dest => dest.Version, opt => opt.MapFrom(src => src.Wersja))
.ForMember(dest => dest.Body, opt => opt.MapFrom(src => src.Nadwozie))
.ForMember(dest => dest.FuelType, opt => opt.MapFrom(src => src.Silnik))
.ForMember(dest => dest.EngineCapacity, opt => opt.MapFrom(src => src.Pojemnosc))
.ForMember(dest => dest.Power, opt => opt.MapFrom(src => src.Moc))
.ForMember(dest => dest.GearboxType, opt => opt.MapFrom(src => src.Skrzynia))
.ForMember(dest => dest.Price, opt => opt.MapFrom(src => src.Cena))
.ForMember(dest => dest.Colour, opt => opt.MapFrom(src => src.Kolor))
.ForMember(dest => dest.City, opt => opt.MapFrom(src => "Warszawa"));

cfg.CreateMap<Car, BmwCar>()
// propercje o takiej samej nazwie mapowane są automatycznie
.ForMember(dest => dest.City, opt => opt.MapFrom(src => "Wroclaw"));

```

Adres repozytorium

<https://github.com/karambolis8/Pracownia-IAS>

Podział pracy w grupie

Małgorzata Robak

- Przygotowanie danych do baz oraz skryptów bazowych
- Praca koncepcyjna nad architekturą systemu

Adam Gudynowski

- Zaprogramowanie systemu

Wnioski

W niektórych przypadkach zasadne może być postawienie trzeciego webserwisu, który zajmował by się integracją wszystkich źródeł danych, oraz ewentualnym ich zachowaniem w bazie danych. Wtedy website integruje tylko jeden webserwis.