

WYŻSZA SZKOŁA INFORMATYKI

I ZARZĄDZANIA

Z SIEDZIBĄ W RZESZOWIE

Projekt

Programowanie

**Car Service – aplikacja mobilna**

**na platformę Android**

|  |  |
| --- | --- |
| Prowadzący:  dr inż. Leszek Gajecki | Wykonawcy:  Olha Voroniak 49146, 4IID-P  Roman Savka 49167, 4IID-P  Ostap Pryima 49141, 4IID-P |
|  |  |

Rzeszów 2014

Spis treści

[1 Postanowienie problemu 3](#_Toc390513522)

[1.1 Wstępne założenia 3](#_Toc390513523)

[1.2 Wykorzystywane technologie 3](#_Toc390513524)

[2 Wymagania funkcjonalne 4](#_Toc390513525)

[2.1 Przypadki użycia 4](#_Toc390513526)

[3 Projektowanie 4](#_Toc390513527)

[3.1 Opracowanie Web Servisu 4](#_Toc390513528)

[3.2 Opracowanie aplikacji mobilnej 5](#_Toc390513529)

[3.2.1 Wprowadzenie wzorców projektowych 5](#_Toc390513530)

[3.2.2 Diagram klas 6](#_Toc390513531)

[4 Podsumowanie 7](#_Toc390513532)

[5 Bibliografia 7](#_Toc390513533)

# Postanowienie problemu

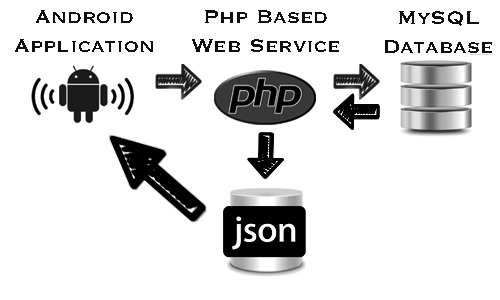
Zaprojektowanie wieloużytkownikowego systemu do ułatwienie komunikacji i współpracy pomiędzy właścicielami samochodów a kierownikami serwisu.

Żeby wszyscy użytkownicy mieli dostęp do wspólnych danych w projekcie powstało dwa obszary do wykonania:

* założenie i opracowanie web serwisu.
* opracowanie aplikacji mobilnej żeby uprzyjemnić korzystanie z serwisu.

## Wstępne założenia

Ogólną architekturę do wykonania postawionego problemu pokazano na rysunku 1.1.



Rysunek 1.1 – Komunikacja aplikacji z bazą danych na web hostingu

Wszystkie dane:

* Dane personalne użytkowników
* Dane o samochodach posiadanych użytkownikami
* Dane o dodanych serwisach;
* Szczegóły komunikacji pomiędzy serwisami a użytkownikami

będą przechowywane w bazie danych na web hostingu.

Etapy komunikacji aplikacji z serwerem:

* wysyłanie aplikacją żądań do php form na serwerze
* opracowanie żądania php, wybieranie danych z bazy
* otrzymanie aplikacją odpowiedzi: json z danymi, które zostały żądane.

## Wykorzystywane technologie

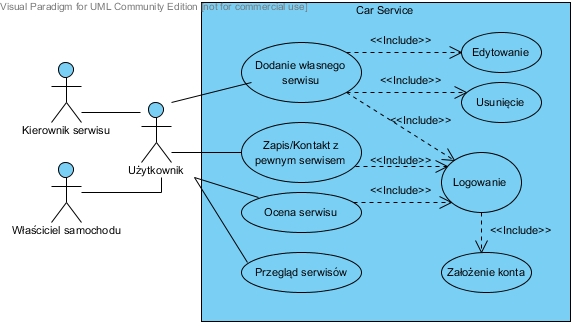
Web hosting: <http://www.hostinger.pl/> (phpMyAdmin 3.5.2.2, MySQL 5.1, PHP 5.3.28)

Aplikacja mobilna może być uruchomiana systemem Android 2.3.3 lub nowszym oraz ekranem o rozdzielczości 320x480 lub większej.

# Wymagania funkcjonalne

Kierownicy mają możliwość zarejestrować swój serwis w systemie, a właścicieli samochodów mogą zapoznać się z wszystkimi serwisami. Jeżeli kierowca zaloguje się do systemu, to on ma możliwość skontaktować się z wybranym serwisem.

## Przypadki użycia

**

*Rysunek 2.1 – Diagram przypadków użycia*

Jak kierownika serwisu tak i właściciela samochodu system uznaje za użytkownika, ponieważ kierownik serwisu też może być właścicielem samochodu. Użytkownik może przeglądnąć dodane do systemu serwisy, dla wykonania wszystkich innych czynności powinien być zalogowany do systemu (logowanie odbywa się z istniejącego konta albo użytkownik może założyć nowe).

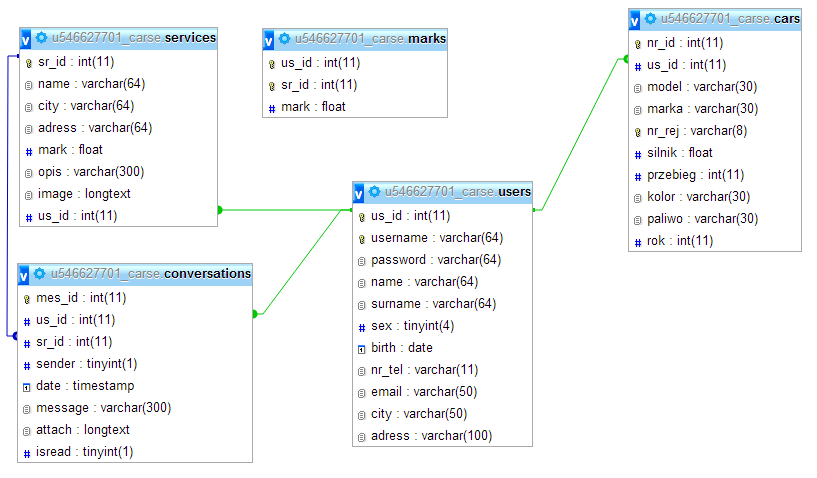
# Projektowanie

## Opracowanie Web Servisu

Tabele i ich powiązania użyte do przechowywania wszystkich danych systemu pokazano na rysunku 3.1.

Do wykonania wszystkich możliwych transakcji przez użytkownika napisano osobne pliki PHP, które wykonują transakcję na bazie i zwracają do aplikacji mobilnej wyniki ich działania.

..\my-car-service\php – ścieżka do php plików w archiwum projektu.



*Rysunek 3.1 – Relacje pomiędzy tabelami w bazie danych.*

## Opracowanie aplikacji mobilnej

### Wprowadzenie wzorców projektowych

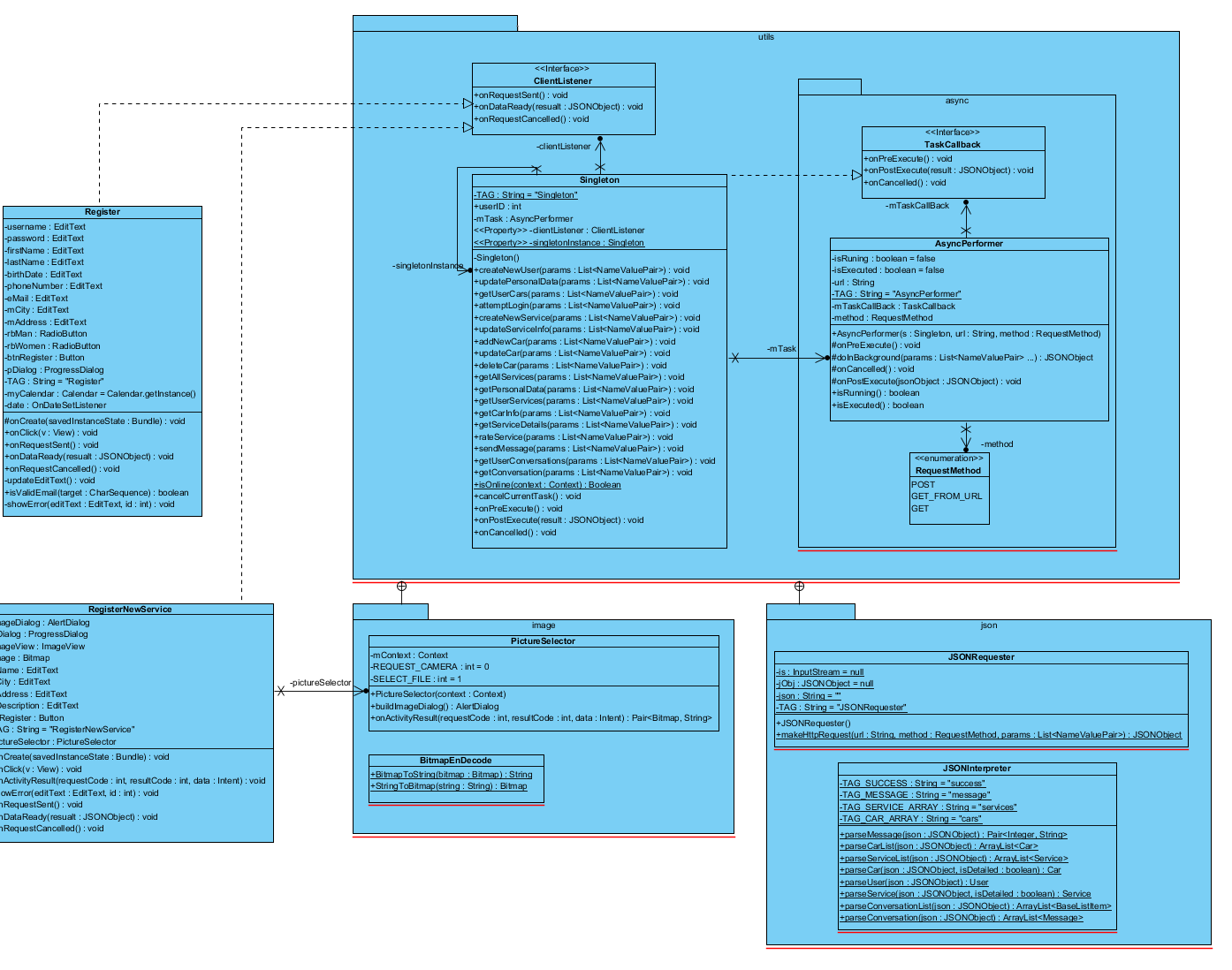
Generalnie aplikacja mobilna składa się pewnej ilości widoków, przy czym każdy widok wykonuje szereg różnych żądań do web serwisu (pobieranie, aktualizacja lub usunięcie pewnych danych). Wszystkie żądania są wykonywane jako http żądanie jednak parametryzowane w różny sposób.

Wzorcem projektowym który nadaje się do takiego problemu jest: **Polecenie (ang. Command, komenda).** Jednym z obszarów zastosowania którego są sytuacje, gdy identyczne polecenia muszą być parametryzowane różnymi danymi w zależności od tego, kto odpowiada za ich obsłużenie.

Wszystkie odpowiedzi web serwisu na różne żądania otrzymywane w postaci JSON. Dla analizowania i interpretowania takich odpowiedzi do obiektów w programie wykorzystano wzorzec **Interpreter.**

Dowolne żądanie może nie zwrócić wyników, po obracaniu smartfonu w tracie jego wykonania. Dla zlikwidowania tego problemu potrzebny obiekt który by istniał nie zależnie od widoku. Zastosowaliśmy dla tego obiekt którzy tworzymy raz na całą aplikację a mianowicie obiekt za wzorcem **Singleton**. Dla poprawnego zawracania wyników żądania zastawaliśmy technikę programowania będąca odwrotnością wywołania funkcji - wywołania zwrotne (ang. Callback).

### Diagram klas



*Rysunek 3.2 – Diagram klas*

Diagram klas na podstawie opisanych wyżej wzorców projektowych pokazano na rysunku 3.2. Na diagramie są tylko dwa przykładowe widoki rejestracji użytkownika i serwisu ( Register i RegisterNewService odpowiednio).

# Podsumowanie

# Bibliografia