

WYŻSZA SZKOŁA BANKOWA W GDAŃSKU  
WYDZIAŁ FINANSÓW I ZARZĄDZANIA

Dawid Zamorowski  
nr albumu 42449

**APLIKACJA WEBOWA WSPOMAGAJĄCA  
TERMINOWE SERWISOWANIE  
POJAZDÓW OSOBOWYCH**

Praca inżynierska  
na kierunku Informatyka

Praca napisana pod kierunkiem  
prof. dr. inż. Mariusza Mola

Gdańsk 2021

## **Streszczenie**

Temat: Aplikacja webowa wspomagająca terminowe serwisowanie pojazdów osobowych.

Słowa kluczowe: aplikacja webowa, samochody, książka serwisowa pojazdu, baza danych, php.

Celem pracy było stworzenie aplikacji która pozwoliłaby osobom prywatnym jak i serwisom samochodowym terminowo wymieniać płyny oraz części samochodowe. W niniejszej pracy ukazałem proces realizacji tego celu, począwszy od przeanalizowania gotowych rozwiązań, poprzez ustalenie oczekiwań i priorytetów dla nowej innowacyjnej aplikacji, wybór technologii implementacji, implementacje, testowanie, podsumowanie statusu realizacji i przydatności, wyniesienie z realizacji pracy doświadczenia.

## **Abstract**

Topic: Web application supporting the timely servicing of passenger vehicles.

Keywords: web application, cars, vehicle service book, database, php.

The purpose of my project was to create an application to support the timely fluid and car parts replacement for private users and car services. In this project I presented the process of achieving the target(starting by analyzing finished solutions, through setting expectations and priorities for a new innovative application, implementations, testing the implementation summary status and usefulness as well as gaining professional experience due to the implementation of this project.

# Spis treści

<b>Streszczenie.....</b>	<b>2</b>
<b>Abstract.....</b>	<b>2</b>
<b>Oświadczenie o samodzielnym wykonaniu pracy.....</b>	<b>2</b>
<b>Wstęp.....</b>	<b>5</b>
<b>1. Przegląd istniejących rozwiązań.....</b>	<b>7</b>
1.1. Książka serwisowa pojazdów.....	7
1.2. AUTODOC CLUB.....	9
1.3. My Car Service.....	12
<b>2. Projektowanie.....</b>	<b>16</b>
2.1. Priorytety projektowe.....	17
2.1.1. Priorytet projektowy : Elastyczność.....	17
2.1.2. Priorytet projektowy : Kontrola wersji.....	17
2.1.3. Priorytet projektowy : Wieloplatformowość.....	18
2.1.4. Priorytet projektowy : Wieloplatformowość.....	18
2.1.5. Priorytet projektowy : Uwzględnienie użytkownika na wielu smartfonach.....	18
2.2. Diagram przykładów użycia.....	19
2.3. Sposób identyfikowania obiektów.....	19
2.4. Kontrola wersji w bazie danych oraz przesyłanie zmian.....	20
2.5. Diagram konceptualny klas występujących w aplikacji... ..	20
<b>3. Narzędzia.....</b>	<b>22</b>
3.1. HTML5.....	22
3.1.1. HTML : Znaczniki.....	22
3.1.2. HTML : Znacznik HEAD.....	23
3.1.3. HTML : Znacznik BODY.....	24
3.1.4. HTML : Porządkowanie kodu.....	24
3.2. PHP.....	25
3.2.1. PHP : Wady i zalety.....	25
3.3. CSS.....	26
3.3.1. CSS : Struktura reguł.....	26
3.3.2. CSS : Rodzaje czcionek CSS.....	27
3.4. Brackets.....	28
3.5. phpMyAdmin.....	29
3.6. Xampp.....	30
3.6.1. Xampp : Poprawne działanie programu.....	31
3.6.2. Xampp : Baza danych.....	31
3.6.3. Xampp : Ukośniki.....	31
3.7. Hosting.....	32
<b>4. Technologie implementacyjne.....</b>	<b>34</b>
4.1. Usługa internetowa.....	34
4.1.1. Usługa internetowa : XML.....	34
4.1.1.1. XML : Otwieranie pliku.....	35
4.1.2. Usługa internetowa : JSON.....	35
4.1.2.1. JSON : Otwieranie pliku.....	36
4.1.3. Usługa internetowa : SOAP.....	36

4.1.4.	Usługa internetowa : REST.....	37
4.1.5.	Usługa internetowa : Różnice między SOAP a ROST....	37
4.1.5.1.	Różnice między SOAP a ROST : Format.....	38
4.1.5.2.	Różnice między SOAP a ROST : Wydajność rozwiązań.....	38
4.1.5.3.	Różnice między SOAP a ROST : Bezpieczeństwo.	38
4.1.5.4.	Różnice między SOAP a ROST : Rozmiar danych.	38
4.1.5.5.	Różnice między SOAP a ROST : Sposób pracy z danymi.....	39
4.1.5.6.	Różnice między SOAP a ROST : Stanowość.....	39
<b>5.</b>	<b>Interfejs użytkownika.....</b>	<b>41</b>
5.1.	Widok logowania.....	41
5.2.	Widok rejestracji.....	42
5.3.	Widok głównego menu.....	43
5.4.	Widok moich samochodów.....	44
5.5.	Widok dodania samochodu.....	45
5.6.	Widok samochodu.....	46
5.7.	Widok zmiany przebiegu.....	47
5.8.	Widok dodanie serwisu.....	48
5.9.	Widok usunięcia samochodu.....	49
<b>6.</b>	<b>Implementacja.....</b>	<b>50</b>
6.1.	Baza danych.....	50
6.2.	Logowanie.....	51
6.3.	Wylogowanie.....	51
6.4.	Rejestracja.....	52
6.5.	Aktualny użytkownik.....	52
6.6.	Usunięcie danych.....	53
6.7.	Aktualizacja danych.....	53
6.8.	Dodanie elementów.....	54
	<b>Podsumowanie.....</b>	<b>56</b>
	<b>Zawartość płyty.....</b>	<b>56</b>
	<b>Bibliografia i netografia.....</b>	<b>57</b>

## Wstęp

Przedstawiona praca dyplomowa, porusza kilka istotnych problemów z dziedziny programowania (również aplikacji mobilnych). Głównym jej tematem jest przedstawienie procesu tworzenia aplikacji na platformach Windows, Linux, MAC oraz systemy operacyjne smartfonów tj. Android oraz IOS. System, który opisuje w pracy inżynierskiej jest system monitorowania wymian płynów i części samochodowych o nazwie: " Mobilna Fura ".

W dzisiejszych czasach Internet używany jest każdego dnia, przez ogromną ilość populacji. Usprawnia on i ułatwia wiele czynności dnia codziennego. Dzięki niemu, ludzie są w stanie np. przełać pieniądze, znaleźć szybki przepis na obiad czy po prostu sprawdzić pogodę. Wymaga to często, aby człowiek, usiadł przed komputerem czy laptopem, a jak powszechnie wiadomo, łatwiej wziąć do ręki smartfon a niżeli „odpalać” komputer. Dlatego ogromną popularnością cieszą się urządzenia mobilne.

Z moich obserwacji wynika, że serwisy samochodowe najczęściej zapisują w książce serwisowej, przekazywanej później dla klienta, datę ponownego przeglądu bądź kiedy ma nastąpić wymiana płynów / części. W praktyce wygląda tak, że posiadacze samochodów bardzo często zapominają, że zbliża się czas w którym powinni odwiedzić warsztat, ponieważ nie zawsze jest jasno określona data, najczęściej jednak słyszymy od serwisanta, że mamy przyjechać na przegląd np. za 10tyś km. W tym wypadku łatwo jest przeoczyć wymaganą ilość kilometrów, gdyż właściciele nie zawsze zapamiętują stan licznika, a później zastanawiają się czy jest już właściwy czas, aby udać się do serwisu, czy jednak jest za wcześnie.

Aplikacja umożliwi użytkownikowi wpisanie w odpowiednie miejsca danych tj. stan licznika po wizycie u serwisanta oraz datę wykonania wraz z czynnością serwisową. Wiadomo, że każdy kierowca przemierza miesięcznie inny dystans, dlatego nie ma możliwości przypomnienia o planowej wymianie podzespołów.

Aplikacja idealnie sprawdzi się dla właścicieli pojazdów, którzy posiadają więcej niż jeden samochód w rodzinie, ponieważ jest możliwość dodania nieograniczonej liczby pojazdów.

Projekt składa się z 6 rozdziałów, poprzedzonych wstępem i podsumowaniem w zakończeniu.

W niniejszej pracy przejrzę istniejące aplikacje pod kątem samochodowym (rozdział 1), opiszę jak zaprojektowałem aplikację (w rozdziale 2), jak wybrałem narzędzia (rozdział 3) oraz technologie

(rozdział 4), jak wygląda interfejs użytkownika (rozdział 5), oraz jak przebiegała implementacja (rozdział 6).

Całość pracy zakończona została podsumowaniem zawierającym moje spostrzeżenia.

Aplikacja, którą stworzyłem, była testowana przeze mnie, przez osoby z mojej rodziny jak i przyjaciół. Testowana była również w zaprzyjaźnionym serwisie. Wszystkie te osoby były w posiadaniu aplikacji około pół roku.

Pozytywnie zweryfikowano wiele aspektów aplikacji, jak i wskazano sporo nowych dróg. Najczęściej proponowane było umieszczenie rubryki z przypomnieniem o ubezpieczeniu - czyli, kiedy należy zapłacić kolejną ratę ubezpieczenia, lub wykupić nową polisę ubezpieczeniową.

Wybór tematu niniejszej pracy, spowodowany był tym, że od kilku lat interesuję się motoryzacją i już nie raz chciałem połączyć kierunek studiów z pasją. Aplikacja, którą stworzyłem w pełni mnie satysfakcjonuje i wiem, że przydatna będzie nie tylko mi, ale i zdecydowanie większej populacji.

# 1. Przegląd istniejących rozwiązań

W tym rozdziale skupię się na przedstawieniu istniejących rozwiązań używanych przez osoby prywatne, które mają ogólny dostęp do aplikacji mobilnych. Dobór rozwiązań przedstawionych w tym rozdziale wynika z obserwacji z mojego otoczenia oraz z bazy aplikacji w Sklepie Play.

Przedstawię aplikacje Książka serwisowa pojazdów, AUTODOC Club oraz My Car Service. Wszystkie dane wymagane do uzupełnienia w aplikacjach uzupełniane będą na podstawie tego samego modelu pojazdu ( Mitsubishi Colt V 1.3 82KM ).

Szczegółowe omówienie wszystkich cech i możliwości tych aplikacji wykracza poza ramę tej pracy, dlatego część aspektów pominię, np. logowanie użytkownika ( uwierzytelnianie ), instalowanie, konfiguracje, administracje. Zwracać będę uwagę na następujące elementy:

- funkcjonalność aplikacji
- przejrzystość treści
- łatwość w operowaniu menu
- możliwość wyszukiwania i filtrowania ( np. po numerze rejestracyjnym pojazdu )
- możliwość pracy bez potrzeby połączenia z Internetem
- łatwość wprowadzania zmian
- możliwość dodania kolejnych kategorii ( np. przegląd LPG )

Przyglądając się tym istniejącym rozwiązaniom będę je opatrywał również krótkimi uwagami: co można byłoby ulepszyć oraz w jaki sposób - w kontekście planów związanych z aplikacją Mobilna Fura.

Przedstawione poniżej 3 różne systemy monitorowania pojazdu częściowo wykonują te same zadania. Jednakże różnią się one zastosowaną technologią, funkcjami, podejściem do problemu, a przede wszystkim każda z tych aplikacji, w mojej potrzebie wydaje się "niedoprecyzowana".

## 1.1 Książka serwisowa pojazdów

Po uruchomieniu aplikacji mobilnej "Książki serwisowej" ukazuje się 5 kafelek (Pojazdy, Warsztaty, Import i Eksport baz danych oraz Wyjście). Jeżeli nie mamy gotowej bazy danych to automatycznie ilość kafelek niweluje się do 2 (Pojazdy oraz Warsztaty).

Po wybraniu opcji "Pojazdy", istnieje jedyna możliwość dodania niezbędnych informacji o pojeździe tj. marka, model, rok produkcji

itd. Po wpisaniu danych istnieje możliwość zapisania ich. Aplikacja posiada nieograniczoną ilość pojazdów do dodania. W tej kategorii pozostają 2 opcje “Czynności serwisowe” oraz “Ciśnienie w oponach”.

Po wybraniu czynności serwisowych ponownie ukazuje się przycisk Dodaj. Aplikacja nie podpowiada o możliwych do wyboru kategoriach, należy wpisać je ręcznie. Istnieje możliwość samodzielnego wpisania daty, godziny oraz przebiegu. Można również opisać przyczynę bądź przebieg naprawy oraz dodać warsztat, ale pod warunkiem, że jest on dodany przez użytkownika w zakładce “Warsztaty”.

W kategorii “Ciśnienie w oponach” przebieg wygląda identycznie jak w czynnościach serwisowych. Istnieje możliwość dodania rozmiaru opon, ciśnienia w oponach przedniej osi oraz ciśnienia w oponach tylnej osi. W menu głównym znajduje się wspomniana już wcześniej zakładka o nazwie “Warsztat”. Należy tam umieścić nazwę oraz adres jak i numer telefonu do preferowanego warsztatu. Istnieje również możliwość dodania zdjęcia, email ‘a oraz strony WWW.

Główną zaletą aplikacji jest jej przejrzystość, łatwość w obsłudze oraz funkcjonalność. Kolejną zaletą jest możliwość samodzielnego wpisania stanu licznika przy wykonanej usłudze.

Słabą stroną aplikacji jest to, że nie ma możliwości ustawienia przypomnienia o zbliżających się wymianach lub przeglądach. Powoduje to że użytkownik może przeoczyć termin wymiany np. płynów, a co gorsza przegląd. Kolejnym minusem aplikacji jest to, że można samodzielnie wpisać każdą informację dotyczącą pojazdu. Większości użytkowników preferuje rozwijaną listę wyboru. Jeśli jednak nie zawierałaby czynności, którą użytkownik pragnie zamieścić, dobrze byłoby gdyby takową informację mógłby dodać.

Dyskusyjne jest w tej aplikacji, brak możliwości zalogowania się. Z jednej strony, użytkownicy często zapominają haseł bądź Loginów, więc pominięcie tego kroku jest dla pewnej części wygodniejsze. Jednakże z drugiej strony, jeśli przykładowo telefon zostanie zmieniony, a użytkownik pominie przesłanie danych na nowy telefon, wszystkie dane przepadną. Jeśli byłaby możliwość logowania, to nie byłoby zagrożenia, że niezbędne dane zostaną utracone.





*Screenshot 1:* Główne menu w aplikacji Książka serwisowa pojazdów.

## 1.2 AUTODOC CLUB

Po uruchomieniu aplikacji mobilnej, od razu pojawia się przewodnik po aplikacji, który oczywiście można pominąć. Zawiera on również łatwe do zapamiętania animacje, co zachęca bardziej do korzystania z aplikacji.

W następnym kroku istnieje możliwość zarejestrowania/zalogowania się lub kontynuowania jako gość. Ja wybrałem opcję gościa - bez logowania. Po tym kroku użytkownikowi pokazuje się kolejny samouczek, który wyjaśnia krok po kroku działanie aplikacji.

Po przejściu instruktażu pokazuje się możliwość dodania pojazdu. Każda z kategorii posiada listę wyboru. Znalazły się w niej również marki oraz modele bardzo rzadko wstępujące, więc nie byłoby potrzeby, aby użytkownik miał możliwość dodawania kolejnych modeli. Aplikacja już na wstępie wymaga podania przebiegu. Bez tych informacji użytkownik nie jest w stanie przejść do dalszych kroków. Po podaniu niezbędnych informacji samouczek przekazuje użytkownikowi informację, że jest możliwość dodania kolejnego pojazdu.

Aplikacja posiada zakładkę “Historia wydatków” w której to użytkownik może nadzorować wydatki na:

- części samochodowe

- olej
- paliwo
- inne - istnieje możliwość dodania w tym miejscu kategorii takich jak np. mycie, tuning, mandat, jest też miejsce na wpisanie kwoty.

W zakładce “Przypomnienia” użytkownik wybiera kategorie - własne, części samochodowe, olej. Każdą z tych kategorii łączy wpisanie danych z licznika oraz datę przypomnienia.

Kolejną opcją w menu jest “Statystyka”. Po wpisaniu niezbędnych wydatków przez użytkownika uzupełnia wykres. Opcja jest możliwa tylko po zarejestrowaniu konta.

Następnym krokiem są “Poradniki” (tutoriale) użytkownik ma dostęp do licznych wskazówek w formie PDF lub filmów instruktażowych.

W ostatniej pozycji jakim jest opcja “Więcej” znajdują się zalecenia takie jak kiedy należy wymienić filtr kabinowy, płyn hamulcowy, czy pasek rozrządu.

Aplikacja jest bardzo obszerna, mógłbym opisać ją dogłębniej, jednak wykracza to poza ramy tej pracy.

W tym miejscu wskażę wady i zalety aplikacji mobilnej AUTODOC CLUB. Główną zaletą aplikacji mobilnej jest przejrzystość oraz łatwość poruszania się po niej.

Następna zaletą jest zalogowanie się do swojego konta z każdego innego urządzenia.

Kolejnym plusem aplikacji jest zakładka “Inne” w sekcji wydatków, gdzie można dodać opis, kwotę oraz zdjęcie np. z przeglądu instalacji LPG.

Następną zaletą jest “Historia przebiegu”, która zostaje zapisywana wraz z datą, dzięki czemu użytkownik może samodzielnie obliczyć ile kilometrów przemierza rocznie. Ułatwi mu to również oszacowanie co ile miesięcy należałoby wymienić dane płyny, jeśli takowe należy wymieniać przy konkretnym przebiegu licznika.

Przy jakiegokolwiek edycji lub dodaniu czynności, aplikacja prosi o podanie aktualnego stanu licznika. Ułatwiającym pomysłem w aplikacji są podane kategorie, dzięki czemu użytkownik wybiera odpowiednią dla niego opcję, przez co aplikacja staje się w jego oczach korzystniejsza, ponieważ nie musi sam zastanawiać się jak prawidłowo nazywa się dana wymiana.

Następną zaletą jest możliwość ustawienia przypomnień o wymianie danego podzespołu (np. płynu chłodniczego, czy też filtra powietrza) w samochodzie.

W aplikacji jest można również ustawić przypomnienie na konkretny dzień lub przy wymaganym przez serwisanta stanie licznika.

Kolejną bardzo przydatną opcją jest możliwość wglądu w statystyki płatności na poszczególne wydatki związane z pojazdem tj. części samochodowe, oleje, paliwo i wiele innych.

Następną ułatwiającą funkcją są podpowiedzi kiedy i co ile tysięcy kilometrów należałoby wymienić podzespoły w pojeździe.

Aplikacja w zakładce “Moje przypomnienia” zapisuje każde wprowadzone przez użytkownika dane. Istnieje również możliwość wglądu do całej historii przebiegu pojazdu co pozwala na ciągłe monitorowanie wymienianych podzespołów.

Kolejną bardzo pomocną funkcją jest odnośnik do sklepu online z częściami samochodowymi dla wybranego samochodu.

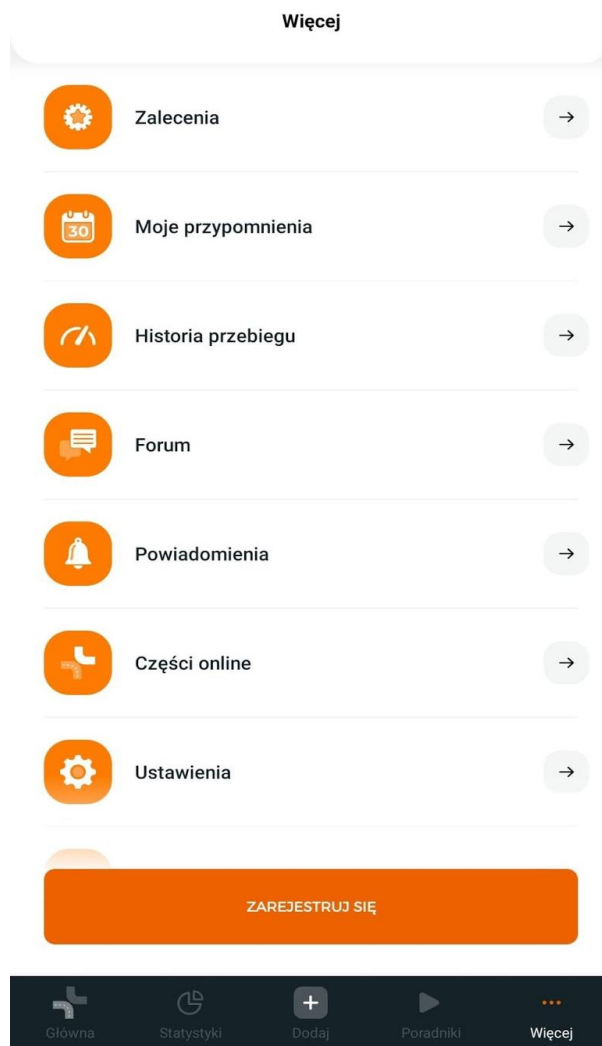
Bardzo często zdarza się, że użytkownik nie jest sam w stanie zdiagnozować co należałoby wymienić lub skontrolować w jego samochodzie, więc twórca aplikacji wzbogacili ją o zakładkę “Forum”. Dzięki temu użytkownik może zadać pytanie na każdy możliwy temat, nie tylko części samochodowych ale też warsztatów, polecanych firm produkujących podzespoły oraz wiele innych.

Aplikacja jest stworzona w bardzo przemyślany sposób. Z trudnością można odnaleźć jej wady. Pierwsze co użytkownik dostrzeże jest to, że na wstępie przy wybraniu pojazdu jest duże prawdopodobieństwo, że wyświetli się błędna wizualizacja co może zastanowić użytkownika czy aby na pewno wybrał właściwy model.

Kolejną utrudniającą wadą jest to że po uzupełnieniu danych o przeprowadzonej wymianie podzespołu nie ma podpowiedzi kiedy teraz należałoby wymienić ją ponownie. Można znaleźć to dopiero w zakładce “Zalecenia”.

Przedostatnią wadą aplikacji jest fakt, że w samochodach starszych (ok. poniżej 2000r.) nie ma podpowiedzi kiedy należałoby wymienić filtry, płyny, lub inne części samochodowe.

Za ostatnią wadę można uznać zbyt wąską gamę instruktaży wymian. To znaczy, że np. Skoda Fabia posiada bardzo dużą ilość poradników, a np. Daewoo Lanos może posiadać jedną bądź żadnej.



*Screenshot 2:* Przykładowa zakładka w aplikacji AUTODOC CLUB.

### 1.3 My Car Service

Po uruchomieniu aplikacji mobilnej “My Car Service” ukazuje się użytkownikowi możliwość zalogowania za pomocą konta Facebook, Google, czy też za pomocą konta e-mail. Program jest zarówno przejrzysty jak i prosty, nie ma w nim niczego skomplikowanego dla użytkownika.

Po zalogowaniu się do aplikacji pokazuje się menu górne w którym można znaleźć przebieg kilometrów samochodu, naprawy, tankowanie paliwa, dokumentację pojazdu czy też ustawienia. W prawym dolnym rogu można dostrzec ikonę “+”, która po naciśnięciu rozwija kolejne możliwości tj. dodanie przebiegu samochodu, naprawy oraz wymiany części, tankowania jak i dokumenty samochodu.

Aplikacja posiada zakładkę historia, która umożliwia wprowadzenie stanu pojazdu, nawet każdego dnia. Pozwoli to także wywnioskować w jakim okresie zrobiło się najwięcej kilometrów czy też podczas ubezpieczania samochodu będzie wiadomo jaki jest roczny przebieg.

W górnym lewym rogu widnieje ikona samochodu, po naciśnięciu ukaże się użytkownikowi aktualnie wybrany samochód, możliwość edytowania jego czy też transfer danych oraz możliwość dodania kolejnego pojazdu do konta klienta.

W zakładce przebieg można zarówno sprawdzić jaki jest aktualny dystans, który pokonał pojazd jak i także ile użytkownik przejeżdża kilometrów w ciągu jednego dnia na podstawie przynajmniej 3 wpisów przebiegu pojazdu.

Kolejną możliwością w menu są wykonane czynności przy samochodzie. Chodzi tutaj o wyświetlenie danych takich jak wymiana podzespołu w pojeździe, naprawa usterki wraz z kosztem przedmiotu czy też usługi mechanika oraz dodanie zdjęcia i przypomnienia za ile kilometrów lub w jakim terminie trzeba wymienić daną rzecz ponownie.

Następnym krokiem jest tankowanie, w którym wyświetlana jest lista tankowań wykonanych w pojeździe z datą jej wykonania, przebiegiem przy jakim było to wykonane czy też kosztem danej usługi. Można wprowadzić także przy tankowaniu czy było ono wykonane do pełnego zbiornika paliwa czy też może nie.

Zakładka dokumenty pozwala na dodawanie informacji wykonanej w pojeździe wraz z kosztami za jej wykonanie oraz datę w którym była ona wykonana.

Ostatnim polem w menu są ustawienia. Znaleźć tam można informacje na temat klienta który jest aktualnie zalogowany w aplikacji jak i również wybrany w danym momencie pojazd z informacją o marce, modelu oraz nazwie jaką nadał mu użytkownik. Można dostrzec tam także możliwość odblokowania wersji PRO, która pozwala na dodawanie i zarządzanie samochodami bez limitu, wyłączenie reklam które mogą denerwować użytkownika czy też przykładowo korzystać z trybu ciemnego aplikacji. Wszystko to oczywiście za opłatą co miesięczną, aby móc korzystać z powyższych udogodnień. Znaleźć można tutaj jeszcze parę innych możliwości, które mógłbym opisać dogłębnie, jednak wykracza to poza ramy tej pracy.

W tym miejscu wskażę wady i zalety aplikacji mobilnej My Car Service.

Główną zaletą aplikacji jest możliwość zalogowania się na trzy sposoby:

- konta Facebook
- konta Gmail
- poczty e-mail

Umożliwia korzystanie z programu na każdym urządzeniu mobilnym na którym ją zainstalujemy.

Kolejną zaletą jest wybranie przez użytkownika na ile dni przed np. wymianą filtra paliwa, czy też wizytą u mechanika, ma dostać powiadomienie o danej czynności oraz o jakiej godzinie ma się ono wyświetlić w przypomnieniach.

Następną zaletą jest możliwość dodania następnego samochodu do listy pojazdów, edytowanie danych o nich w każdym momencie oraz transfer danych na serwer aplikacji o klienta autach.

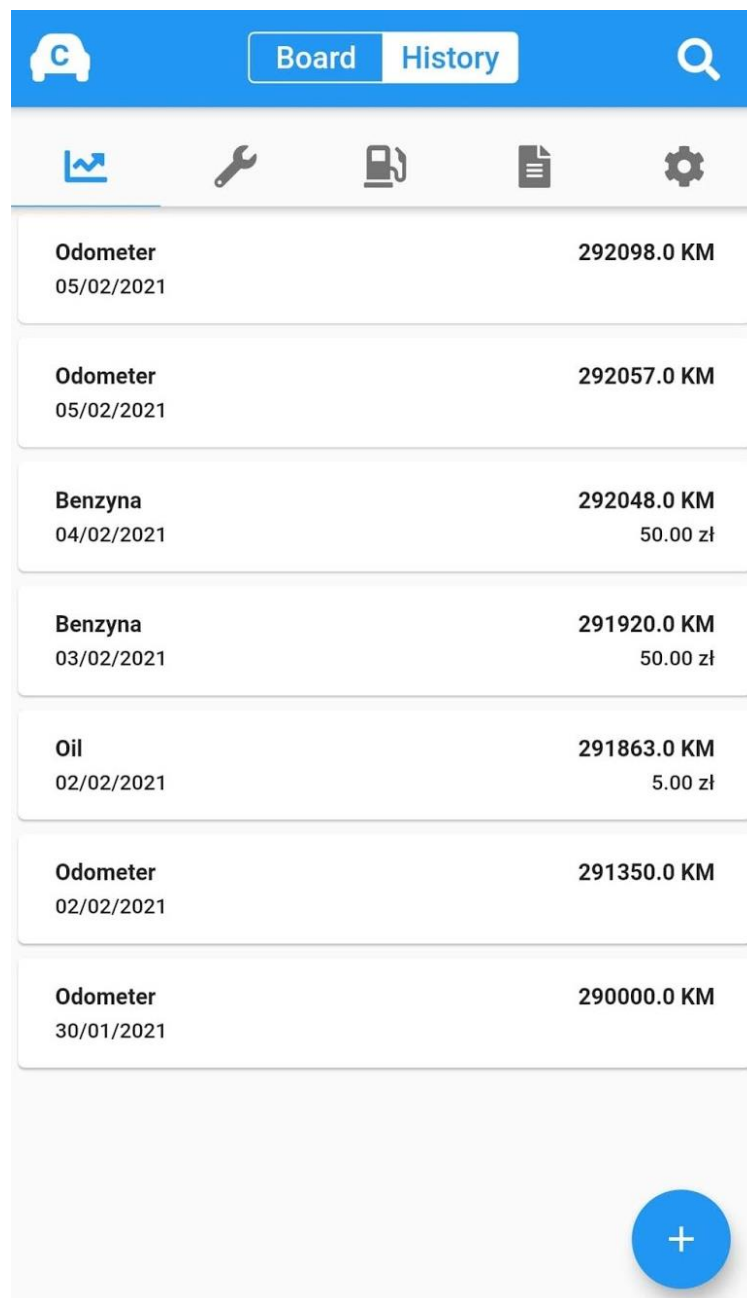
Ostatnią istotną zaletą jest historia przejechanych kilometrów. Użytkownik otrzymuje informację jakiego dnia jaki był posiadany przebieg kilometrów na liczniku pojazdu. Umieszczone są także informacje przy jakim przebiegu samochodu została dokonana wymiana podzespołu w nim czy też wizyty u mechanika. Może to także klientowi ułatwić wywnioskowanie ile przejeździł trasy przez rok od poprzedniego ubezpieczenia pojazdu.

Aplikacja była stworzona przemyślanie, w prosty i przejrzysty sposób dla użytkownika. Lecz można w niej dostrzec liczne wady które mogą zrazić, bądź nawet utrudnić korzystanie z niej, jak przykładowo brak wyboru języka użytkownika z programu. Posiadamy tutaj tylko możliwość korzystania z języka angielskiego. Jest duże prawdopodobieństwo, że osoby które nie potrafią posługiwać się w tym języku, zrezygnują z aplikacji.

Kolejnym utrudnieniem jest brak możliwości wybrania marki pojazdu jak i jego modelu. Klient musi sam wpisać co to jest za samochód oraz nadać mu własną nazwę.

Następną wadą jest brak listy kategorii wykonanych czynności przy aucie. Nie każdy posiada wiedzę, jak nazywa się dany podzespół i nie będzie wiedział jak opisać daną czynność.

Największą wadą jest narzucenie przez autora aplikacji na użytkownika wykonania comiesięcznych płatności, aby móc korzystać z pełnej wersji programu, która rozszerza możliwości o bez limitowe dodawanie i zarządzanie autami, całkowity brak reklam który może zasłaniać treść czy też uniemożliwiać korzystanie z aplikacji, dodanie dla klienta możliwości korzystania z trybu ciemnego, który będzie dla dużej ilości osób przyjemniejszy dla oka.



*Screenshot 3:* Zakładka przebieg kilometrów pojazdu w aplikacji My Car Service.

## 2. Projektowanie

Projektowanie aplikacji webowej jest najważniejszym etapem przy samym jej tworzeniu. W dużej mierze projektowanie łączy się z implementacją ponieważ jak dobrze wiadomo nie zawsze wszystko da się wyobrazić i przewidzieć od razu przy projektowaniu. Początkowe priorytety projektowe są oczywiste od samego początku i zazwyczaj nie ulegają zmianie, z kolei inne stają się zrozumiałe dopiero w trakcie implementacji.

Aplikacja spełniająca wymagania użytkowników cechuje się tym że spełnia wymagania początkowe projektu jak i z łatwością będzie możliwe rozbudowanie go w późniejszym etapie projektowania.

Idealny system informatyczny powstaje według następujących faz:

- Analiza wymagań
- Projektowanie
- Implementacja
- Weryfikacja
- Pielęgnacja

Niekiedy przy projektowaniu aplikacji istnieje potrzeba powrotu rozpatrzenia faz wcześniejszych, aby można było pójść dalej. Przykładem tego są fakty, które można dostrzec dopiero podczas samej implementacji, więc może zdarzyć się powrót do kwestii projektowych, aby uwzględnić te fakty.

Analiza wymagań była całkowicie nieformalna ze względu na brak wymagań, powodem tego było ułatwienie posiadaczom samochodów kontroli nad ich pojazdami jak i stworzenie aplikacji która będzie miała na celu przypominanie im o ważnych wymianach. W początkowym etapie w swojej pracy chciałem stworzyć wersję która będzie spełniała tylko część wymagań, aby sprawdzić trudność realizacji i wykonalność. W pierwszej wersji aplikacja miała być rozszerzona do: panel istniejących pojazdów, baza pomiarów przebiegów, przypomnienia.

Wykonując niniejszą pracę projektowanie było tworzone poprzez wnioski wyciągnięte z prototypowych implementacji.

Implementacja zostanie omówiona w rozdziale 6.



## **2.1. Priorytety projektowe**

Tutaj zostaną przedstawione priorytety projektowe, które zostały wzięte pod uwagę przy tworzeniu aplikacji webowej „Mobilna Fura”. Motywem przewodnim jest stworzenie aplikacji, dzięki której użytkownik w łatwy sposób będzie miał możliwość dodawania wykonanych czynności serwisowych, a także, będzie mógł w każdej chwili sprawdzić jej datę. Najważniejsze jest również stworzenie aplikacji która będzie funkcjonalna jak i przejrzysta. Korzystne było by rozbudowanie tej aplikacji o możliwość podglądu historii przebiegu oraz przypomnień 7 dni przez wymianą istotnych podzespołów. Aplikacja mogłaby być wykorzystywana jako podstawa do dalszego rozwijania projektu pod kontem użycia bazy danych.

Przy tworzeniu tej aplikacji zostały zaczerpnięte rozwiązania przez analizę działających istniejących na rynku aplikacji, wybranie najważniejszych funkcji oraz połączenie z nowymi pomysłami, co pozwoliłoby na stworzenie idealnej aplikacji do serwisowania pojazdów osobowych.

### **2.1.1. Priorytet projektowy: elastyczność**

Przy tworzeniu aplikacji webowych ważne jest aby nie było nic narzucone z góry. Najgorsze są ograniczenia i schematy.

- typ pola – nie powinien być narzucony. Użytkownik powinien sam wprowadzać w pola informacje (liczby oraz treści ), które chce zawrzeć. Pozwala to na swobodę, tym bardziej, jeśli użytkownik nie zna prawidłowej treści tylko jej potoczność lub skróconą nazwę.

- nazewnictwo obiektów – powinna być taka możliwość, lecz bez zrywania ustawień powiązanych z danym obiektem. Takie zjawisko znane jest w językach programowania jak i w bazach danych. Najczęściej jednak sytuacja jest odwrotna – czyli zrywają się ustawienia.

### **2.1.2. Priorytet projektowy : kontrola wersji**

Idealny program powinien udostępniać funkcjonalność kontroli wersji dla danych, którymi manipuluje użytkownik. Każda zmiana, która zachodzi, powinna być rejestrowana, wraz z możliwością wglądu: kto edytował, kiedy oraz jaka była historia w danej komórce.

Dzięki tego typu operacji użytkownik będzie mógł poczuć się zdecydowanie swobodniej podczas nadawanych zmian, ponieważ jeśli wkładnie się błąd, będzie możliwość odzyskania wcześniejszych

informacji. Pozwoli to na poczucie, że użytkownik kontroluje „swoją” aplikację.

### **2.1.3. Priorytet projektowy : brak możliwości działania bez połączenia z Internetem.**

Użytkownik ma możliwość korzystania z aplikacji przy użyciu Wi-Fi, czy też danych komórkowych w swoim telefonie. Dzięki temu posiada dostęp do swojego założonego konta na każdym urządzeniu na jakim tylko zostanie zainstalowana aplikacja, oraz ma możliwość do korzystania z niej w każdym miejscu w którym się znajdzie, oczywiście pod warunkiem posiadania dostępu do sieci. Umożliwia to w czasie rzeczywistym pobieranie na bieżąco danych znajdujących się w bazie oraz aktualizowanie zmian zachodzących w profilu użytkownika.

### **2.1.4. Priorytet projektowy: wieloplatformowość**

Aplikacja powinna mieć możliwość działania na wielu systemach operacyjnych. Użytkownicy często korzystają z Android ‘a czy iOS, dlatego ograniczenie się do jednego systemu operacyjnego, skutkowałoby zmniejszoną ilością zainstalowań. „Mobilna Fura” ma charakter notatnika, więc nie powinna tworzyć barier operacyjnych.

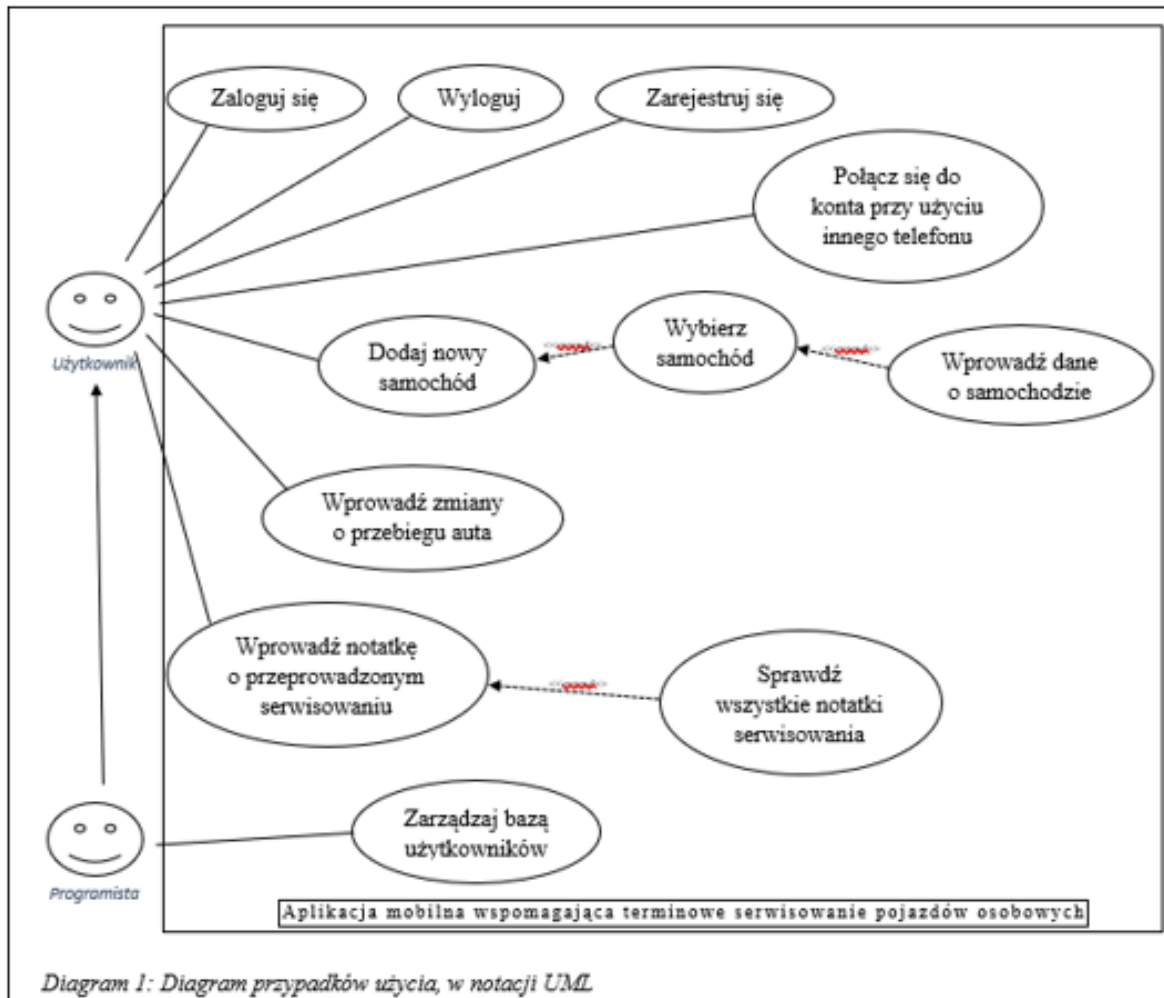
Problemem przy stworzeniu aplikacji wieloplatformowej jest spora ilość różnic pomiędzy systemami operacyjnymi. Dana aplikacja została stworzona w takich sposób, żeby nie kolidować z żadnym systemem komórkowym. System Android i system iOS nie będą posiadać żadnych różnic w swojej funkcjonalności czy też w wyglądzie. Wsparcie dla wieloplatformowości jest również istotnym czynnikiem przy wyborze technologii implementacyjnych (Rozdział 3).

### **2.1.5. Priorytet projektowy: uwzględnienie użytkownika na wielu smartfonach**

Często zdarza się, że użytkownik posiada kilka telefonów – najczęściej dwa ( prywatny oraz służbowy ). Aplikacja powinna mieć możliwość automatycznej synchronizacji danych pomiędzy telefonami. Zwłaszcza: dodawanie pojazdów, zmiany przebiegu, historia wymian podzespołów itp. Dzięki temu użytkownik będzie miał pewność, że nic nie umknie jego uwadze, a dane takie jak przebieg, nie są łatwą czynnością do przypomnienia sobie lub znalezienia w innych zapisach.

## 2.2. Diagram przykładów użycia

W diagramie poniżej zilustrowano zakres funkcjonalności dostępnej dla użytkownika w aplikacji (Diagram 1). Na diagramie zaznaczono możliwości użytkownika oraz programisty.



## 2.3. Sposób identyfikowania obiektów

Użytkownik będzie miał możliwość dodawania obiektów w aplikacji (tj. samochód, serwisowanie). Każdy nowo utworzony obiekt potrzebuje nadania identyfikatora (id), by móc stworzyć połączenia do tego obiektu. Najczęściej używanym identyfikatorami w informatyce są:

- Nazwy nadane przez Użytkownika, identyfikator tego rodzaju został nie wykorzystany z powodu:
  - Nazewnictwo obiektu może być zmienne, gdzie wskazana jest niezmiennosc identyfikatora. Zmiana jego spowodowałaby usunięcie powiązań prowadzących do obiektu

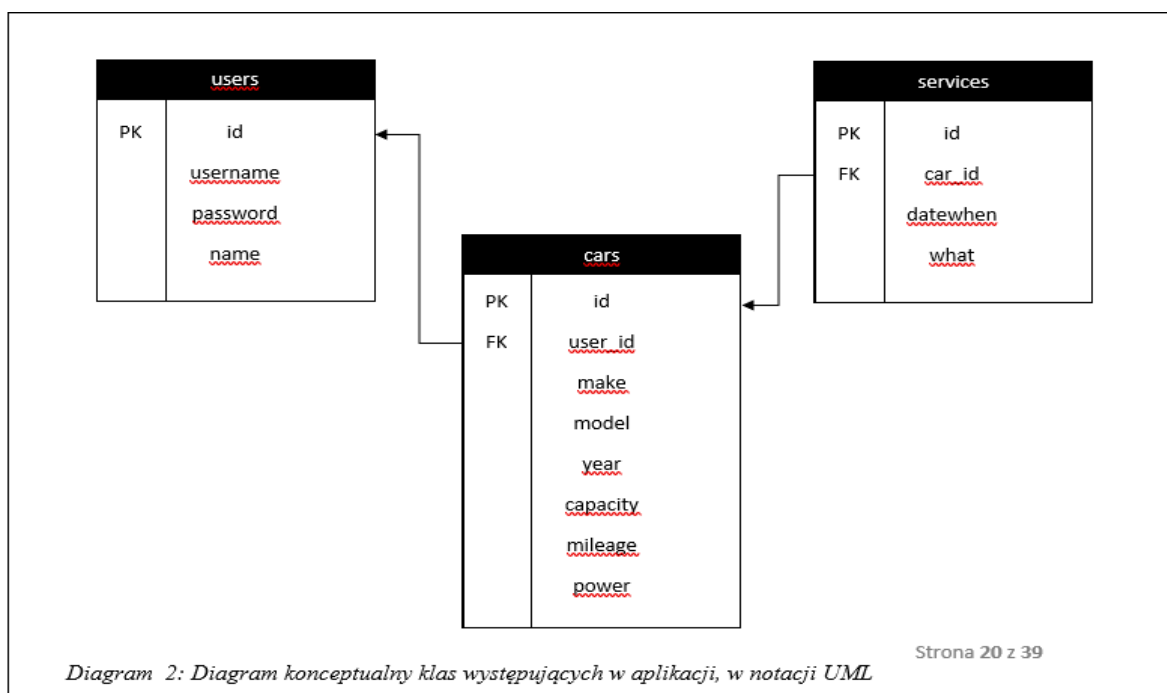
- Takie samo nazewnictwo wykorzystane przez klienta większej liczby niż jednego elementu w systemie spowodowałoby kolizję identyfikatorów.
- Liczby całkowite inkrementowane sekwencyjnie. Zaprojektowana aplikacja funkcjonuje w sposób scentralizowany, dzięki czemu dany rodzaj identyfikatora nie będzie wywoływał kolizji.

## 2.4. Kontrola wersji w bazie danych oraz przesyłanie zmian

Zaprojektowana aplikacja ma posiadać mechanizm do kontroli wersji czyli możliwość do rejestrowania zmian, które zaszły w bazie danych. Przydaje to się do przesyłania (synchronizacji) zmian pomiędzy urządzeniami użytkowników. Niektóre aplikacje odpowiadają za kontrolę wersji, przesyłanie zmian, przechowywanie danych w bazie, oraz wykorzystywanie elementów znajdujących się wewnątrz bazy danych.

## 2.5. Diagram konceptualny klas występujących w aplikacji

Posłużyłem się diagramem konceptualnym klas przy użyciu notacji UML, aby zilustrować zakres typów obiektów, z którymi będzie miał do czynienia użytkownik.



Strona 20 z 39

Jak można zauważyć na powyższym diagramie, każda klasa ma zależne dziedziczenie od innej klasy, tj. „Users” dziedziczy elementy „Cars”, a ta natomiast dziedziczy obiekty „Services”.

## 3. Narzędzia

W danym rozdziale wspomnę o wykorzystanych technologiach narzędziowych, które były wykorzystane w celu stworzenia aplikacji która jest tytułem danej pracy. Pewne z tych narzędzi posiadają charakter instrumentów ułatwiający proces tworzenia oprogramowania, natomiast kolejne są kluczowymi elementami dla całego systemu, które powodują że stworzenie aplikacji byłoby niemożliwe bez nich.

### 3.1. HTML5

HTML (ang. Hypertext Markup Language. Język znaczników służący do tworzenia dokumentów hipertekstowych. Pierwsze powstające strony WWW były oparte tylko na strukturze HTML. Bez niego nie może funkcjonować żadna strona internetowa. Wraz z upływem lat cały projekt się rozwijał – w roku 1990 twórca Tim Berners-Lee opracował i rozesłał do kolegów w CERN pierwszą przeglądarkę internetową o nazwie WorldWideWeb. Po kilku latach sam język i technologie pokrewne rozpoczęły drogę do zapewnienia jednakowej obsługi kodu HTML niezależnie od przeglądarki czy to popularnego niegdyś Internet Explorera aż po obecnie rozpowszechnione Google Chrome.

HTML opisuje całą strukturę informacji, które są zawarte wewnątrz strony internetowej, nadając dobrane znaczenie poszczególnych elementów tekstu formując: nagłówki; akapity; listy; hiperłącza, oraz wprowadza w dokumencie tekstowym obiekty plikowe takie jak multimedia czy też elementy bazy danych jak np. formularze.

#### 3.1.1. HTML: Znaczniki

- HTML – znacznik otwierający dokument – `<html></html>`
- HEAD – opisujący nagłówek dokumentu i najważniejsze informacje o nim – `<head></head>`
- META TITLE – zawierający tytuł strony widoczny w wynikach wyszukiwania, na karcie przeglądarki oraz przekazujący istotne informacje dla robotów o zawartości danej podstrony – `<title></title>`
- BODY – właściwa treść dokumentu zawierająca inne znaczniki z treścią do wyświetlenia – `<body></body>`
- H1-H6 – pełniące funkcje nagłówków strony i opisujące następujące po sobie sekcje danej podstrony. Najistotniejszy z nich jest nagłówek H1. – `<h1></h1>`
- P – paragraf, czyli pojedynczy blok tekstowy – `<p></p>`
- DIV – czyli po prostu blok, za jego pomocą często porządkuje się witrynę – `<div></div>`

- B oraz STONG – służące do pogrubiania poszczególnych fragmentów tekstu – `<b></b>` i `<strong></strong>`
- IMG – pozwalający na umieszczenie obrazów na stronie (choć nie jest to jedyny sposób) – `<img src=""></img>`
- A – czyli hiperłącza, dzięki któremu umożliwia się przejście do innych podstron lub serwisów internetowych. – `<a href=""></a>`

```

<!-- Tak się oznacza
komentarz w HTML - przeglądarka tego nie przetwarza - pozostaje
niewidoczne -->

<!-- Informacja dla przeglądarki o rodzaju dokumentu -->
<!DOCTYPE HTML>

<!-- Znacznik otwierający html i określenie języka-->
<html lang="pl">

<!-- Znacznik otwierający head -->
<head>

<!-- Określenie kodowania znaków - bez szczegółów, bez tego może nie być polskich znaków -->
<meta charset="utf-8">

<!-- Tytuł strony - to jest BARDZO ważne pod SEO a wielu o tym zapomina lub bagatelizuje -->
<title>Super świetny dopasowany SEO semantycznie tytuł podstrony</title>
</head>

<!-- Znacznik otwierający body - właściwa treść dokumentu -->
<body>

<!-- Article - niby nie jest to konieczne, ale jest to dobry zwyczaj by oznaczać fragment kodu, który mógłby stanowić
samodzielną część i być publikowany niezależnie jak treść wpisu, bez menu stopek itp. -->
<article>

<!-- Nagłówek H1 - powinien być jeden i tylko jeden na stronie -->
<h1>Tytuł artykułu lub wpisu, warto w nim dać frazę kluczową</h1>
<p>Tu piszemy jakiś akapit tego tekstu <strong>a tak możemy wyróżnić jakąś frazę, zwrot lub fragment tekstu</strong></p>
<!-- Nagłówek H2 lub dowolny inny, służy się je dla śródtytułów -->
<h2>W ten sposób wyróżnia się śródtytuł - warto je stosować!</h2>
<p>No i piszemy kolejny akapit tekstu, nikt nie lubi patrzeć na ścianę znaków</p>
</article>

<!-- Znacznik zamykający body -->
</body>

<!-- Znacznik zamykający html -->
</html>

```

*Screenshot 4: Budowa języka HTML.*

### 3.1.2. HTML: Znacznik HEAD

Zaczynając pisać dokument HTML najlepiej jest zacząć od znacznika HEAD. Należy rozpocząć tym miejscu ponieważ opisywane są wszystkie ustawienia dokumentu HTML, które są potem widoczne bezpośrednio w kodzie. W obrębie jego znacznika można wykorzystywać takie znaczniki jak:

- `<title></title>` - tytuł podstrony
- `<style></style>` - można dzięki temu dodać unikalne style CSS, które upiększą stronę pisaną w HTML
- `<link rel='stylesheet' href='style.css'>` - w ten sposób wskazać można plik ze stylami CSS które można wykorzystać dla danej strony czy nawet i podstrony

- `<script src="java.js"></script>` - pozwoli to użytkownikowi na dodanie pliku ze skryptami, dzięki czemu może stać się stronie bardziej interakcyjną, pozwalając przykładowo na nietypową obsługę przycisków
- `<meta charset="UTF-8">` - zapewnia to nam obsługę polski znaków

### 3.1.3. HTML: Znacznik BODY

Po zamknięciu HEAD można przejść do znacznika BODY, w którym będzie się znajdowała właściwa treść dokumentu. Innymi słowy to to co będzie widziała osoba pisząca jak i inni użytkownicy, którzy trafią na stronę. Można w niej znaleźć takie znaczniki jak:

- `<p></p>`
- `<div></div>`
- `<h1></h1>`
- `<h2-6></h2-6>`
- `<a></a>`
- `<strong></strong>`
- `<b></b>`
- `<i></i>`
- `<u></u>`
- `<br></br>`
- Etc.

### 3.1.4. HTML: Porządkowanie kodu

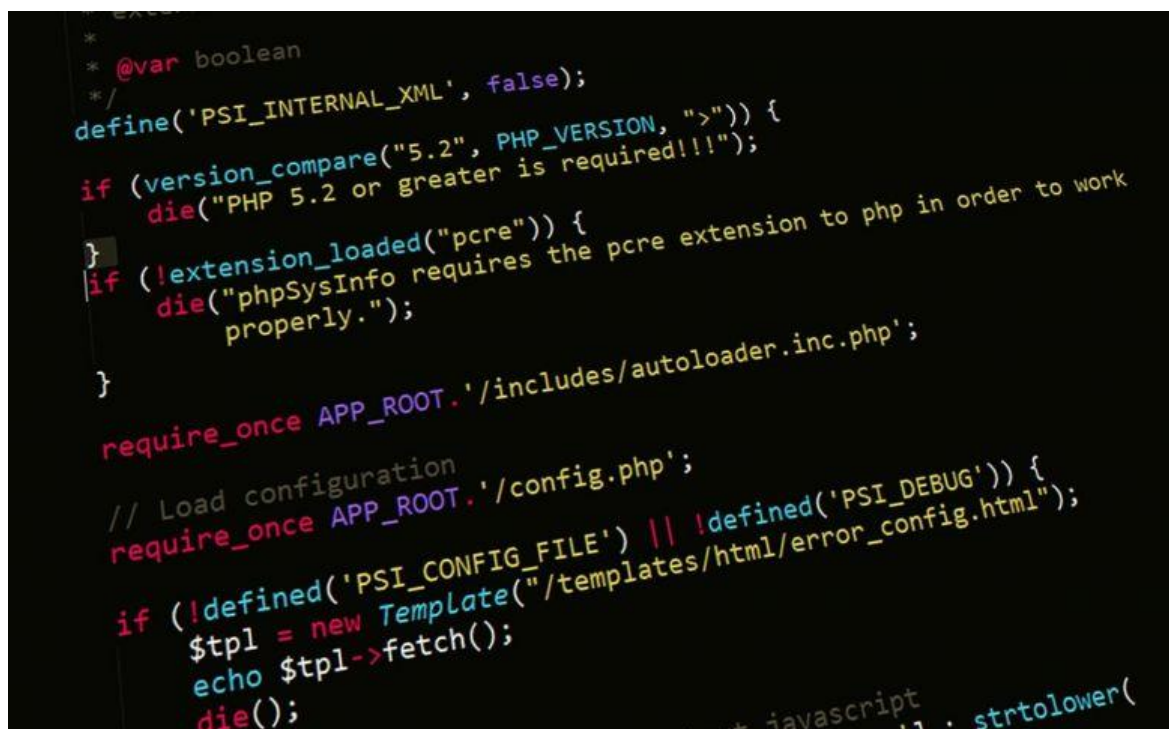
Przy wykorzystaniu kodu HTML wiele jest potrzebne do samodzielnego stworzenia dokumentu. Do rozpoczęcia przygody z tym językiem wystarczy najprostszy edytor tekstowy jakim jest notatnik. Oczywiście to jest, że czym strona internetowa będzie miała być bardziej rozbudowana tym trudniej będzie wykorzystać do tego typu dokumentach.

Warto nie tylko pod kątem wizualnym planować rozkład elementów i treści na stronach internetowych. Należy też zadbać aby semantyka kodu HTML (użytych znaczników) była uporządkowana i sensowna. Pomoże to nie tylko podczas edycji strony i użytkownikom, ale także roboty Google będą lepiej rozumiały co na danej podstronie właściwie się znajduje.



## 3.2. PHP

PHP jest skryptowym językiem programowania, którego można kojarzyć z wygenerowaniem kodu stron internetowych, wykorzystując odpowiedzi przychodzące z baz danych, które za pomocą poleceń są odczytywane stanowiąc część integralną PHP. Powstał on 27 lat temu, ale nadal jest powszechnie wykorzystywany.



```
* @var boolean
*/
define('PSI_INTERNAL_XML', false);

if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work properly.");
}

require_once APP_ROOT.'/includes/autoloader.inc.php';

// Load configuration
require_once APP_ROOT.'/config.php';

if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
}
```

*Ilustracja 1: Przykładowy kod wykorzystujący język PHP.*

Dzięki temu językowi programowania internetowe witryny przestały być strukturami statycznymi, które potrzebowały ręcznej modyfikacji, a stały się dynamicznymi stronami dostosowującymi się (np. internetowa galeria).

Napisanie tego ręcznie w HTML zajęłoby bardzo dużo czasu, a dzięki PHP użytkownik benchmark.pl dostaje żadaną stronę WWW w ułamku sekundy. Gdy do tego dodać fakt, że na jednej stronie się nie kończy, bo to nawet setki inaczej wyglądających stron zależnie od wyboru odwiedzającego tworzony serwis, to widać, że bez automatyzacji nie dałoby rady tego zrobić.

### 3.2.1. PHP – Wady i zalety

Jak wszystko co jest na świecie ma wady i zalety. Łatwość nauki programowania w PHP, tworzenia kodu, a także dowolność

wprowadzenia danych z zewnątrz do generowanej strony, daje ogromne pole do popisu cyberprzestępcom

Główną zaletą PHP, a jednocześnie cechą dla niego charakterystyczną jest jego skalowalność, co oznacza że jest to język, który może zostać wykorzystany przez każdego zarówno osobę która zaczyna swoją przygodę z programowaniem, jak i profesjonalistów w tej dziedzinie, znajdą się dla tego języka zastosowania w ich projektach. Może zostać on wykorzystany do tworzenia małych aplikacji Webowych, stron internetowych, jak i do systemów obsługujących ogromne korporacje, które transformują duże ilości danych. W Internecie można znaleźć mnóstwo poradników oraz materiałów związanych z programowaniem w języku PHP, co pokazuje, że język staje się stosunkowo prostym w opanowaniu przy odpowiednim zaangażowaniu i chęciach.

### 3.3. CSS

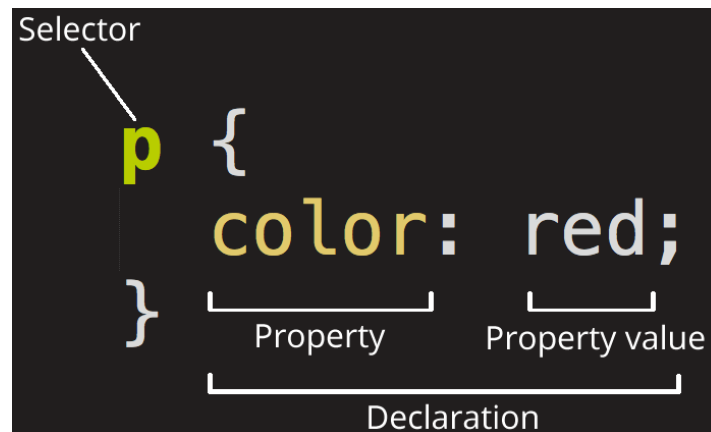
W porównaniu do HTML, CSS nie jest językiem programowania, ale także nie jest on językiem znaczników. Jest on jednak językiem arkuszy stylów co oznacza to, że pozwala zastosować style do elementów w dokumentach HTML. Przykładowo żeby każdy paragraf w stronie HTML zmienił kolor tekstu na czerwony, należy zastosować następujący kod CSS:

A screenshot of a code editor showing a CSS rule. The code is `p { color: red; }`. The text is displayed in a light gray background with a blue vertical bar on the left side of the code area. The code is color-coded: 'p' is blue, '{' is black, 'color:' is red, 'red;' is red, and '}' is black.

*Screenshot 5: Przykładowy styl z arkusza CSS.*

#### 3.3.1. CSS – struktura reguł

Tak jak było wspomniane wcześniej w podpunkcie 3.1. trzeba powiązać z dokumentem HTML, żeby arkusz stylów mógł zadziałać w przeciwnym razie stworzone style nie będą miały żadnego wpływu na to jak wygląda strona.



*Ilustracja 2: Struktura reguł w arkuszu CSS.*

Cała struktura kodu CSS jest nazywana listą dyrektyw, czyli inaczej reguł. Oto nazwy jego części:

- Selektor – jest początkiem reguły w elemencie HTML
- Deklaracja – ustala jaką właściwość będzie posiadał element który chcemy wystylizować
- Właściwości – sposób stylizowania danego elementu HTML
- Wartość właściwości – można wybrać jedną wartość z wielu możliwych wyglądów

### 3.3.2 – Rodzaje czcionek CSS

W języku CSS czcionki możemy podzielić na pięć rodzajów, w zależności od wyglądu samej litery.

Wyróżnia się 5 rodzajów czcionki:

- Szeryfowa – serif

stosuje się zazwyczaj do długich bloków tekstu drukowanego. Są one wygodne w czytaniu i nie męczą wzroku

- Bez szeryfowa- sans-serif

Stosuje się je zazwyczaj do długich bloków tekstowych. Proste wykończenie liter ułatwia wyświetlanie ich na wyświetlaczach z mniejszą gęstością pikseli, szczególnie w małym rozmiarze

- O tej samej przestrzeni – monospace

są jednakowej szerokości. Czcionka ta zazwyczaj używana jest do wyświetlania kodu m.in. w edytorach kodu

- Kursywa – cursive

mogą być pochyłe, połączone, bądź być odzwierciedleniem pisma odręcznego

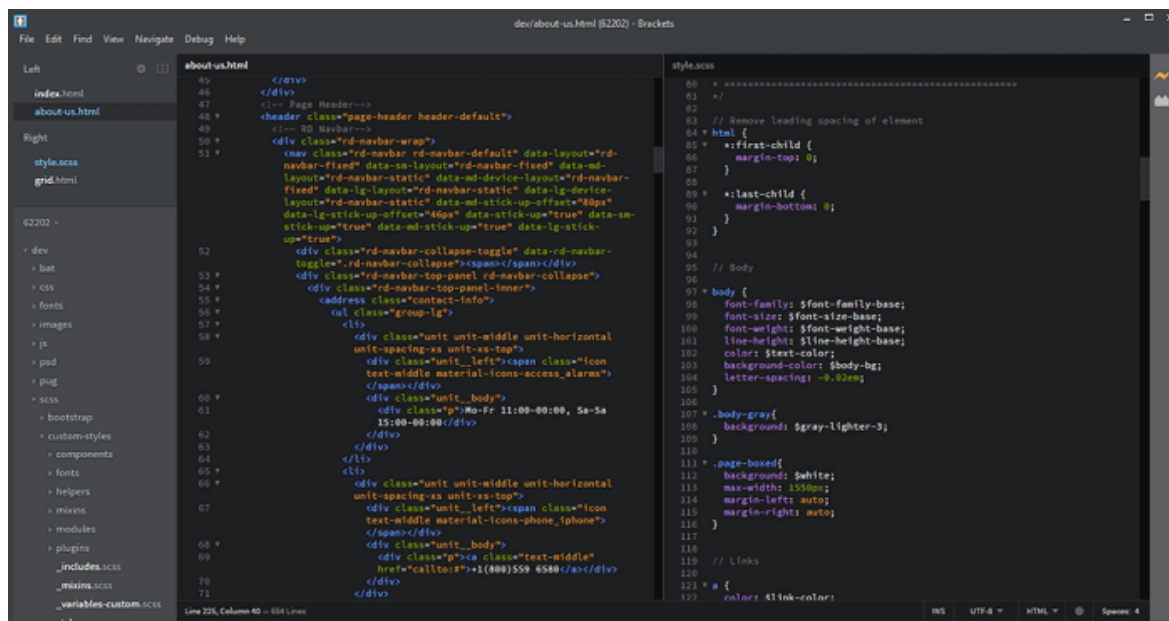
- Fantasy

są zazwyczaj ozdobne i używane jedynie w tytułach, nie nadają się do długich bloków tekstowych



*Ilustracja 3: Rodzaje czcionki w CSS.*

## 3.4. Brackets



*Screenshot 6: Struktura budowy programu Brackets.*

Program oferuje programistom bardzo przejrzysty i czytelny widok kodu. Wykorzystuje ciemne tło i ważne treści oznaczane są kolorami, które w żaden sposób nie zlewają się ze sobą. Brackets daje także możliwość szybkiego tworzenia struktur stron przez wciśnięcie klawisza „TAB” od razu ukazuje się to co jest potrzebne programistom.

Jest on także przyjemnym narzędziem dzięki opcji podglądu pracy w oknie przeglądarki oraz możliwość łatwego przechodzenia między stworzonymi plikami poprzez widok listy po lewej stronie programu, czy też otwarcia dwóch plików jednocześnie obok siebie. Posiada on wbudowany debugger, który na bieżąco pokazuje wykonane błędy w kodzie.

Beckers jest edytorem tekstowym, który idealnie sprawdza się dla programistów, nawet i tych zaczynających dopiero swoją przygodę z pisanem pierwszych programów. Podczas pisania koloryzując składnie podpowiada elementy, które można wprowadzić, tak samo jak i parametry oraz automatycznie domyka każdy nawias, więc nie straszne szukanie we wszystkich liniijkach kodu błędu, „zjedzonej litery”, czy też nie postawionego przecinka czy też nie zamkniętego nawiasu.

Jego ogromną zaletą jest fakt, że występuje w języku polskim jak i to, że jest darmowy.

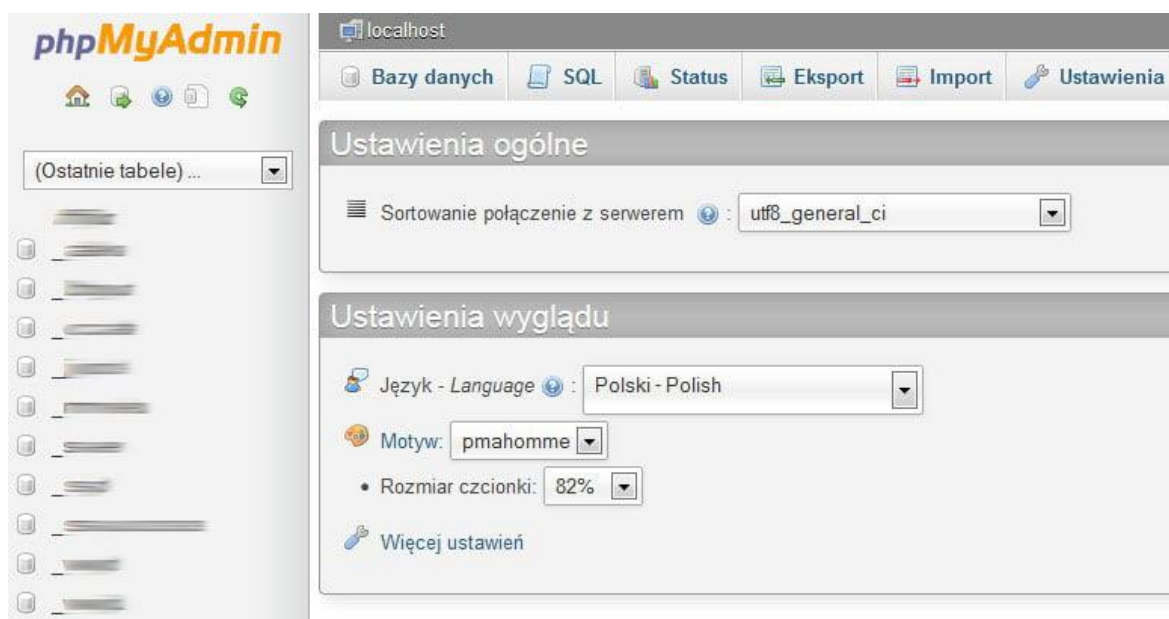
Dany program jest bardzo wartościowym narzędziem, które jest dostępne na platformę Windows, Mac OS oraz Linux, a przede wszystkim jest darmowe.

### **3.5. phpMyAdmin**

Jest to aplikacja którego obsługa odbywa się poprzez stronę WWW i zostało stworzone w celu maksymalnego ułatwienia procesu zarządzania bazami danych MySQL na hostingu. Dzięki phpMyAdmin można w wygodny sposób pracować z bazą danych, tworzyć oraz edytować tabele, rekordy jak i również wykonując kwerendy.

Aplikacja została napisana w języku PHP i dzięki temu umożliwia ona zaawansowane administrowanie bazami danych MySQL, nie trzeba znać zagadnień technicznym czy też komend używanych w wierszu poleceń. Wszystkie powyższe czynności wykonuje program, a nie użytkownik, czyli wspomniana wyżej aplikacja PHP. Jest ona w pełni darmowa i dostępna na licencji GNU (General Public License).

Za pomocą phpMyAdmin można, np. tworzyć lub usuwać tabele lub rekordy w bazie danych, dodawać i kasować relacje oraz edytować całą strukturę i zawartość baz danych MySQL. Narzędzie phpMyAdmin jest często używane do importu i eksportu bazy danych.



*Screenshot 7: Wygląd okna aplikacji phpMyAdmin.*

### 3.6. Xampp

Xampp to prosty do zainstalowania oraz kompletny serwer HTTP gotowy do działania. Jest darmowy oraz przyjazny użytkownikowi wersja serwera Apache. Serwer konfiguruje się natychmiastowo żeby zapewnić najlepsze działanie ze współpracującymi z nim usługami.

XAMPP to idealny obszar do rozległego działania programistów oraz testerów potrzebujących szybkiego i sprawnego przeanalizowania i zweryfikowania konkretnych skryptów na lokalnym komputerze. Programista otrzymuje grupę bibliotek oraz aplikacji które są potrzebne do obsługi i zarządzania witryną WWW.

Znajdą się wśród nich:

- PEAR
- MiniPerl
- mod\_ssl
- OpenSSL
- PHPMyAdmin
- Webalizer
- FileZilla FTP Server
- SQLite
- Zend Optimizar
- Mercury Mail
- Panel kontrolny Xampp'a

Ten program może być udostępniany na czterech platformach: Linux, Windows, OS X oraz Sun Solaris.

Niestety nie pozwala on na udostępnianie bazy w sieci internetowej lecz zamyka się tylko na sieci lokalnej. Przez co posłużył mi przy projektowaniu początkowym aplikacji webowej.

### **3.6.1. Xampp – Poprawne działanie programu**

By projekt w PHP czy MySQL właściwie działał również po przeniesieniu na produkcję, należy wziąć pod uwagę wielkość liter. Wbrew pozorom ma ona ogromne znaczenie dla poprawności pracy. Zaleca się konsekwencję w używaniu wielkości znaków. Kluczowe jest stosowanie niezmiennie identycznego wzoru zapisu – z dużych, bądź małych liter – od początku do końca tak samo.

Dzieje się tak, ponieważ systemy Windows i Linux różnie traktują odwołania do nazw plików pisanych wielkimi i małymi literami. System Windows, będzie działać równie dobrze nawet, jeśli np. nazwy plików są innymi literami niż wskazano to w kodzie.

Problem z jakim mogą spotkać się początkujący programiści ma z kolei związek z bazami danych. Warto pamiętać, aby po przenosinach projektu z serwera lokalnego w odpowiednich plikach konfiguracyjnych projektu przewidzieć odwołania do „nowej” bazy danych na serwerze produkcyjnym.

### **3.6.2. Xampp – Baza danych**

Pliki znajdujące się na lokalnym komputerze nie będą do bazy dopuszczone. Firma hostingowa często domyślnie blokuje dostęp do bazy z hostów zewnętrznych. Jest to polityka dość sensowna, ograniczająca możliwość ataków na bazę. W tym wypadku utrudnia jednak pracę.

### **3.6.3. Xampp – Poprawne działanie programu**

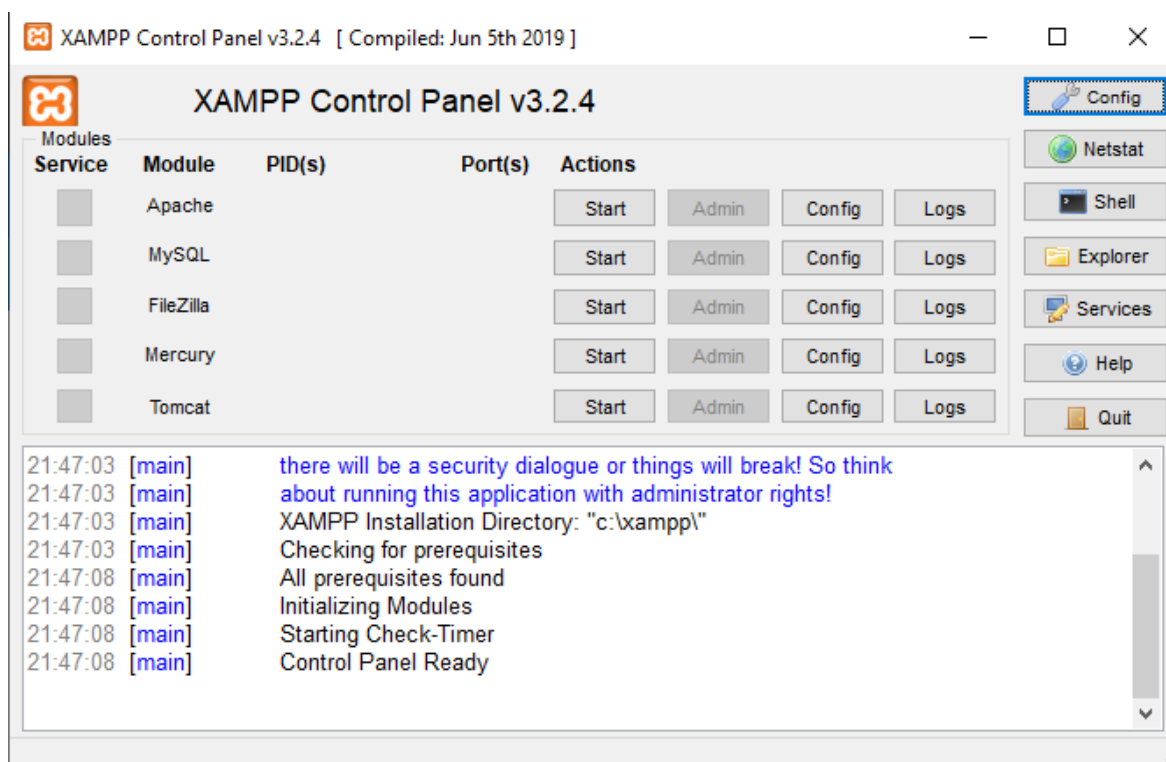
Ukośniki również odgrywają znaczną rolę w XAMPPie. Generalnie występują ich dwa typy –forward slash (/) i back slash (\) – pierwszy z nich oddziela ścieżki w Linuksie i adresach URL, drugi zaś w Windowsie. Slash w jedną bądź w drugą stronę ma zatem zasadnicze znaczenie.

## 1. Forward Slash

- Oddziela elementy adresu URL, np. cyberfolks.pl/blog/wpis.
- Używane do ścieżek w systemie Linux.
- Windows zazwyczaj radzi sobie z prawidłowym interpretowaniem do rozdzielania ścieżek

## 2. Back Slash

- Oddziela foldery w systemach Windows, przykładowo: C:\Windows\
- W programowaniu najczęściej używany jako znak ucieczki, do tzw. escapowania znaków specjalnych.



*Screenshot 8: Interfejs programu Xampp.*

## 3.7. Hosting

Jest to zasób serwerowy, który jest udostępniany przez dostawcę usług internetowych. Dzięki niemu jest umożliwiony dostęp do zawartości. Hosting można nazwać przestrzenią, która znajduje się w



sieci internetowej cały czas. Istnieje także możliwość udostępnienia online na serwerze określonych zasobów.

Hosting określa się jako część serwera, która jest przeznaczona do konfiguracji i kontroli przez jednego użytkownika. Można nazwać go także hostowaniem, czyli udostępnianiem w sieci internetowej zasobów użytkownika.

## **4. Technologie implementacyjne**

W niniejszym rozdziale przedstawione zostaną kwestie technologiczne, które zostały wykorzystane w celu stworzenia aplikacji. Wzięte pod uwagę zostały kwestie tj.:

- Kryteria przydatności danej technologii
- Dostępność
- Bogactwo funkcjonalności
- Poziom nowoczesności
- Zaawansowanie techniczne
- Popularność
- Stabilność
- Wydajność

### **4.1. Technologie implementacyjne: Usługa internetowa**

Usługa internetowa ( ang. Web Service ) to właściwość systemu teleinformatycznego polegająca na udostępnianie za pomocą sieci, danych uporządkowanych w określonej strukturze.

Usługa sieciowa jest w istocie składnikiem oprogramowania, niezależnym od platformy sprzętowej oraz implementacji, dostarczającym określonej funkcjonalności. Dane przekazywane są zazwyczaj za pomocą protokołu HTTP i z wykorzystaniem XML.

Na bazie usług sieciowych można konstruować systemy rozproszone i aplikacje internetowe. Aplikacje te komunikują się przez sieć komputerową, z wykorzystaniem odpowiednich protokołów dostępu zdalnego.

Jest to niezależny moduł programistyczny, umożliwiający funkcjonalność tj. wymiana danych. Najczęściej oparte są o protokół HTTP w modelu użytkownik-serwer.

Na przestrzeni wielu lat ukształtowały się pewne formaty danych do komunikacji między aplikacjami. Opisane zostaną tutaj dwa z nich: XML(ang. Extensible Markup Language) oraz JSON (ang. JavaScript Object Notation).

#### **4.1.1. Usługa internetowa – XML**

XML to nic innego jak zwykłe pliki tekstowe, które używają niestandardowych znaczników do opisu struktury i innych funkcji dokumentu.

To język znaczników stworzony przez World Wide Web Consortium (W3C) w celu zdefiniowania składni kodowania dokumentów, które zarówno ludzie, jak i maszyny mogą czytać. Czyni to poprzez użycie znaczników, które definiują strukturę dokumentu, jak również sposób przechowywania i transportu dokumentu.

Najprawdopodobniej najłatwiej porównać go z innym językiem znaczników, z którym możesz się zapoznać - HTML używanym do kodowania stron internetowych. Używa on krótkich kodów, które opisują format treści na stronie internetowej. Na przykład następujący prosty kod HTML używa znaczników, aby niektóre słowa były pogrubione lub pisane kursywą. Zasadniczo HTML jest językiem, który koncentruje się na prezentacji treści, podczas gdy XML jest dedykowanym językiem opisu danych używanym do przechowywania danych.

#### **4.1.1.1. XML – otwieranie pliku**

Istnieje kilka sposobów bezpośredniego otwarcia pliku XML. Możesz je otwierać i edytować dowolnym edytorem tekstu, przeglądając je za pomocą dowolnej przeglądarki internetowej lub korzystać ze strony internetowej, która umożliwia przeglądanie, edytowanie.

#### **4.1.2. Usługa internetowa – JSON**

JSON to otwarty format zapisu struktur danych. Jego przeznaczeniem jest najczęściej wymiana danych pomiędzy aplikacjami. JSON składa się z par atrybut – wartość oraz typów danych tablicowych. Notacja JSON’a jest zbieżna z obiektami w języku JavaScript.

Jego zaletą jest popularność, prostota użytkowania, zwiezłość - a jako że dane są zapisywane do tekstu – po sformatowaniu czytelne dla ludzi. Został stworzony w 2001r.

JSON może być alternatywą dla XML lub CSV. Pliki JSON zapisujemy z rozszerzeniem .json a typem MIME dla plików .json jest „application/json”.

Wartościami w JSON nie mogą być: ciąg znaków, liczba, obiekt JSON (zagnieżdżony), tablica, wartość prawda/fałsz (zapis małymi literami true albo false), null.

##### 1.1. Zalety JSON

- prostota i czytelność
- lekkość – tekst formatu JSON zawiera mniej znaków, ponieważ nie przechowuje nadmiarowych informacji,

- natywny dla JavaScript – szybsze parowanie w stosunku do innych formatów, które wymagają kosztownych transformacji,
- popularny i rozpowszechniony – co powoduje możliwość wykorzystania go w wielu językach i systemach.

#### 1.2. Wady JSON

- mniejsza powszechność od XML – bo JSON to stosunkowo młody wynalazek,
- większy rozmiar – w porównaniu do formatów binarnych,
- nie można używać komentarzy

### 4.1.2.1. JSON – otwieranie pliku

Aby mieć możliwość otwarcia pliku JSON należy mieć zainstalowany, któryś z poniższych programów, w zależności, jaka platforma systemowa jest używana.



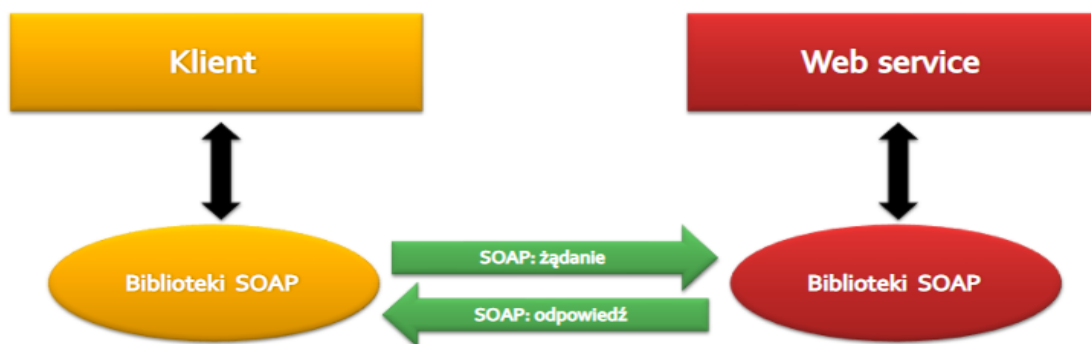
*Ilustracja 4: Programy pozwalające otworzyć format JSON.*

### 4.1.3. Usługa internetowa – SOAP

SOAP (ang. Simple Object Transfer Protocol) to jak sama nazwa wskazuje, protokół komunikacyjny, który został stworzony z inicjatywy Microsoftu po to, aby umożliwić komunikację pomiędzy aplikacjami, używając języka XML. Protokół powstał w 1998 roku, odpowiadając na potrzeby firmy Microsoft. Jednak jego przydatność docenia się także dzisiaj. Nadal często używany jest ze względu na

swoją standaryzację, gwarancję bezpieczeństwa oraz prostą obsługę, umożliwiającą kontrolę zawartości przekazywanych dalej danych.

SOAP jest standardem W3C, którego głównym celem było zastąpienie bardziej specyficznych protokołów komunikacyjnych (RPC), których wykorzystanie może być ograniczone poprzez zapory sieciowe lub inne zabezpieczenia.



*Ilustracja 5 : Struktura SOAP*

#### **4.1.4. Usługa internetowa – REST**

REST (ang. Representational State Transfer) jest stylem architektonicznym i wzorcem narzucającym dobre praktyki, który umożliwia definiowanie formatu przesyłanych danych. Stanowi też element standaryzacji protokołu HTTP.

REST został stworzony przez Roya w 2000 roku. Z racji swojej elastyczności, szybkości działania i prostoty konstrukcji jest bardzo chętnie używany przez programistów. W przeciwieństwie do SOAP, REST nie jest protokołem. Można określić go jako usługę – serwis komunikujący się na relacji klient – serwer.

#### **4.1.5. Usługa internetowa : Różnice między SOAP a Rost**

Mimo że zarówno SOAP i REST są usystematyzowanymi strukturami danych, istnieją pomiędzy nimi zasadnicze różnice. Dotyczą one kilku parametrów, które należy wziąć pod uwagę, wybierając najlepsze rozwiązanie. Należy do nich format, wydajność, bezpieczeństwo, rozmiar danych, sposób pracy z danymi, stanowść i zastosowanie.

#### **4.1.5.1. Różnice między SOAP a ROST : Format**

Jeśli ważną rolę stanowi duża liczba formatów, należałoby zdecydować się na REST. Dzięki niemu dane można wymieniać w formacie JSON, XML, HTML a także jako zwykły tekst. Daje więc Ci to duże pole manewru.

#### **4.1.5.2. Różnice między SOAP a ROST : Wydajność rozwiązań**

REST ze względu na swój mały rozmiar wymaga mniejszej przepustowości łącza niż SOAP. Co więcej, dla takiej samej ilości danych wymaga mniejszej mocy obliczeniowej. Dzięki temu działa szybciej, nawet o kilkadziesiąt procent, od SOAP.

#### **4.1.5.3. Różnice między SOAP a ROST : Bezpieczeństwo**

Jeśli chodzi o najbezpieczniejszy protokół to zdecydowanie jest to SOAP. Dzięki niemu otrzymuje się wsparcie dla WS-Security, który jest rozszerzeniem elementów bezpieczeństwa, zapewniającym dołączanie tokenów do komunikatów, oferującym poufność oraz integralność. Co więcej, wybierając SOAP, zyskuje się też budowaną logikę, która wspomaga komunikację o błędach w integracji.

#### **4.1.5.4. Różnice między SOAP a ROST : Rozmiar danych**

W zakresie rozmiaru danych oba rozwiązania również znacząco różnią się pomiędzy sobą. SOAP wymusza format XML oraz określone definicje w dokumencie. Wszystko to powoduje narzut na przesyłane zapytanie.

Natomiast formaty dostępne dla REST, czyli wspomniany już np. JSON, mimo że są ustandaryzowane, w przypadku rozmiaru są zdecydowanie bardziej elastyczne i optymalne.

#### **4.1.5.5. Różnice między SOAP a ROST : Sposób pracy z danymi**

Jeśli chodzi o dane, rozwiązanie SOAP umożliwia udostępnianie elementów logiki aplikacji jako usługi. Są one definiowane zgodnie ze standardem WSLD, który został opracowany przez IBM oraz Microsoft. Dzięki temu to plik definiuje, które informacje i w jaki sposób można wydobyć z danego serwisu.

W przypadku REST dane traktowane są jako zasoby. Dane zasoby są więc opisane w dokumentacji serwisu.

#### **4.1.5.6. Różnice między SOAP a ROST : Stanowość**

W zakresie stanowości również istnieją spore różnice. W przypadku SOAP jego działanie może być zarówno bezstanowe, jak i z wykorzystaniem sesji.

Z kolei, jeśli chodzi o REST to jednym z jego założeń jest komunikacja bezstanowa. Konsekwencją tego jest brak tworzenia i przechowywania sesji po stronie serwera. Dzięki temu w łatwy sposób można powiększyć i rozbudować system, a także go unowocześnić.

### **ZASTOSOWANIE ROZWIĄZAŃ**

Patrząc pod kątem funkcjonalności, w zasadzie każdą usługę sieciową można oprzeć na SOAP i REST. Jednak ze względu na wydajność i wygodę rozwiązań, o wiele częściej wykorzystuje się RESTful API niż SOAP.

Okazuje się, że nawet niektóre istniejące już webserwisy oparte o rozwiązanie SOAP, są przepisywane na REST. Także pomimo próby zachowania ich obecnej funkcjonalności działania. Dlaczego tak się dzieje? Z praktyki wynika, że zastosowanie SOAP jest zdecydowanie bardziej wyspecjalizowane i może ono mieć przewagę w aplikacjach, w których najważniejszym parametrem jest bezpieczeństwo. Wykorzystuje się to np. w usługach finansowych oraz w bramkach płatności.

Po przeanalizowaniu różnic dotyczących SOAP i REST wciąż nie jest możliwe wybranie najlepszego rozwiązania. Nawet słabnąca popularność usług opierających się o protokół SOAP i licznych zalet REST nie są wystarczającymi argumentami. Jak więc dokonać

właściwego wyboru? Przede wszystkim powinieneś wziąć pod uwagę rodzaj wdrażanej aplikacji. Mimo że w przeważającej części będzie to prawdopodobnie REST, może okazać się, że SOAP jest również bardzo przydatny. Jeśli jednak będziesz mieć wątpliwości, zasięgnij opinii specjalistów!



## 5. Interfejs użytkownika

W tym rozdziale zostały przedstawione screenshot'y aplikacji webowej „Mobilna Fura”. Aplikacja ta została stworzona tak, aby była przejrzysta oraz prosta w rozumieniu. Ma ona na celu stworzenie idealnego notesu dla kierowcy/właściciela pojazdu. Aplikacja ta w późniejszym etapie zostanie rozbudowana o dodatkowe zakładki tj. „Przypomnienia”, „Historia przebiegu” oraz zostanie dodana baza czynności serwisowych. Docelowo aplikacja ma również zawierać pełną bazę marek oraz modeli pojazdów osobowych.

### 5.1. Widok logowania



*Screenshot 9:  
Widok ekranu  
logowania  
użytkownika*



*Screenshot 10:  
Efekt nie  
wprowadzenia  
żadnych danych*



*Screenshot 11:  
Efekt  
wprowadzenia  
błędnych danych*

W widoku ekranu logowania aplikacji (Screenshot 9) użytkownik widzi dwa pola do wprowadzania danych oraz dwa przyciski. Pierwsze pole „Login” odpowiada za wprowadzenie nazwy użytkownika który znajduje się już w bazie danych aplikacji, natomiast drugie pole „Hasło” jest odpowiedzialne za wprowadzenie hasła które jest dopisane do konta użytkownika.

Pierwszy przycisk „Zaloguj” odpowiada za sprawdzenie czy dane w polach „Login” i „Hasło” zgadzają się z jakimkolwiek elementem bazy, jeżeli wszystko jest poprawnie, to przycisk przekierowuje do Głównego Menu, w przeciwnym wypadku wyświetla się komunikat (Screenshot 11) o nie pasujących elementach bazy danych użytkowników. W przypadku, jeśli użytkownik nie wprowadzi, w przynajmniej jedno pole, prawidłowych danych pojawi się komunikat nad danym polem, z informacją, że należy ją uzupełnić (Screenshot 10) w przeciwnym wypadku nie zostanie on zalogowany. Natomiast drugi przycisk „Zarejestruj” się przekierowuje do widoku rejestracji.

## 5.2. Widok rejestracji



Rejestracja

Login:

Hasło:


Imię:

Nazwisko:

Zarejestruj się

Logowanie

*Screenshot 12:*  
*Widok ekranu rejestracji*



Rejestracja

Login: Uzupełnij login

Hasło: Uzupełnij hasło

Imię: Uzupełnij imię

Nazwisko: Uzupełnij nazwisko

Zarejestruj się

Logowanie

*Screenshot 13:*  
*Efekt wprowadzenia błędnych danych*

Pierwszy widok (Screenshot 12) pozwala klientowi na zarejestrowanie się do aplikacji. Znajdziemy w tym miejscu cztery pola do wprowadzania danych oraz dwa przyciski funkcyjne.

Pierwsze z pól pozwala na dodanie do bazy, nazwy użytkownika, kolejne na nadanie hasła zabezpieczające konto. Trzecie oraz czwarte pole pozwala na wprowadzenie informacji o kliencie takie jak imię oraz nazwisko dzięki czemu wiadomo do jakiego użytkownika jest przypisane konto.

Pierwszy przycisk „Zarejestruj się” umożliwia po wprowadzeniu poprawnie powyższych 4 pól, dodanie do bazy danych użytkownika kolejnego konta.

Jeżeli jednak któreś z pól nie zostanie wypełnione, wyświetli się nad danym miejscem komunikat (Screenshot 13) o uzupełnienie danych. Natomiast pola „Login” oraz „Hasło” są dodatkowo zabezpieczone i sprawdzają długość wprowadzonych liter w tych polach i jeżeli są krótsze niż 3 znaki, to obramowanie zmienia kolor na czerwony co jednocześnie uniemożliwia poprawne założenie konta użytkownikowi.

Przycisk „Logowanie” pozwala na wrócenie do widoku logowania.

### 5.3. Widok głównego ekranu



*Screenshot 14: Widok ekranu głównego aplikacji*

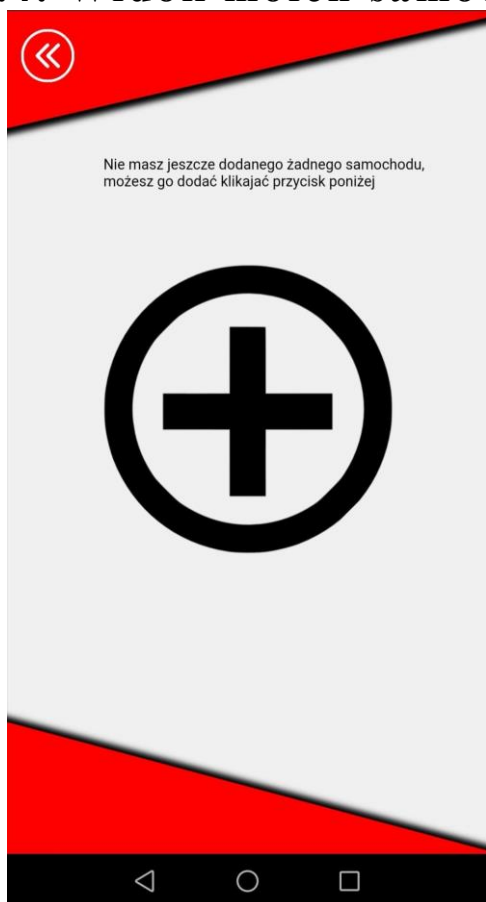
Dany widok (Screenshot 14) ukaże się użytkownikowi po uprzednim, poprawnym zalogowaniu się do aplikacji.

W prawym górnym rogu, widać logo aplikacji, razem obok niego przywitanie aktualnie zalogowanego użytkownika, wykorzystując wcześniej podane przy rejestracji jego imię wraz z nazwiskiem.

Są dwie możliwości funkcyjnych przycisków. Pierwsza to „Moje samochody”, która po naciśnięciu przekierowuje klienta do widoku moich samochodów, która zostanie opisana w kolejnym podpunkcie.

Natomiast drugi przycisk „Wyloguj” ma za zadanie wylogowanie z aktualnie wykorzystywanej sesji użytkownika i powrót do widoku logowania.

#### 5.4. Widok moich samochodów



*Screenshot 15:  
Pierwsze zalogowania się*



*Screenshot 16:  
Widok zakładki „Moje  
Samochody”*

Widok „Moich samochodów” jest zależny od jednej rzeczy - czy użytkownik wprowadził przynajmniej jeden pojazd do bazy danych pojazdów, jeżeli nie, ujrzy na ekranie widok (Screenshot 15)

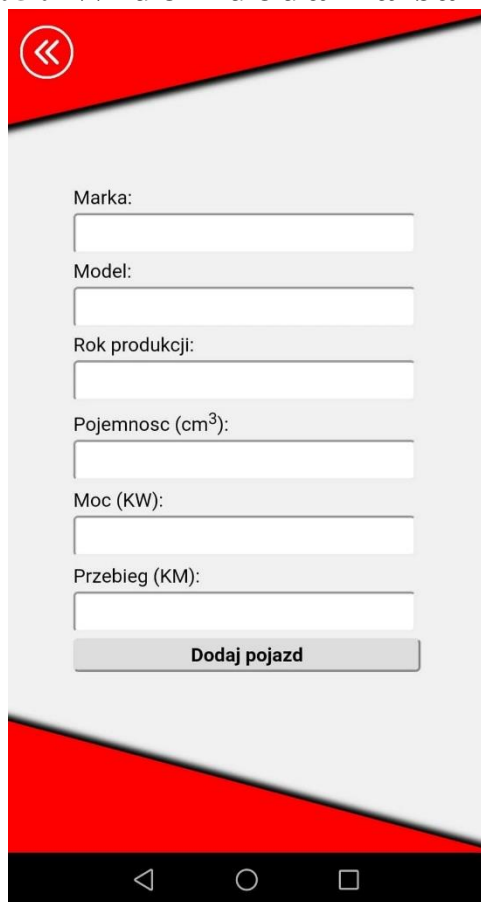
informujący o braku wprowadzonych pojazdach do bazy. Duży przycisk w kształcie plusa znajdujący się w okręgu, pozwoli na przekierowanie użytkownika do widoku dodania samochodu.

W przeciwieństwie do klienta który nie miał wprowadzonego jeszcze żadnego samochodu, ukazać się może drugi widok (Screenshot 16). Wyświetli on listę wszystkich wprowadzonych przez niego samochodów wraz z danymi, które zamieścił przy dodawaniu do bazy, z możliwością przejścia do widoku wybranego samochodu.

Nad listą znajduje się identyczny przycisk, jak w przypadku użytkownika, który nie posiada jeszcze żadnego pojazdu wprowadzonego do swojego konta. Pozwala on na przejście w zakładkę dodawania samochodu.

W tym widoku znajduje się także przycisk „przewijania wstecz”, który po wciśnięciu przekierowuje użytkownika z powrotem do głównego widoku.

## 5.5. Widok dodania samochodu

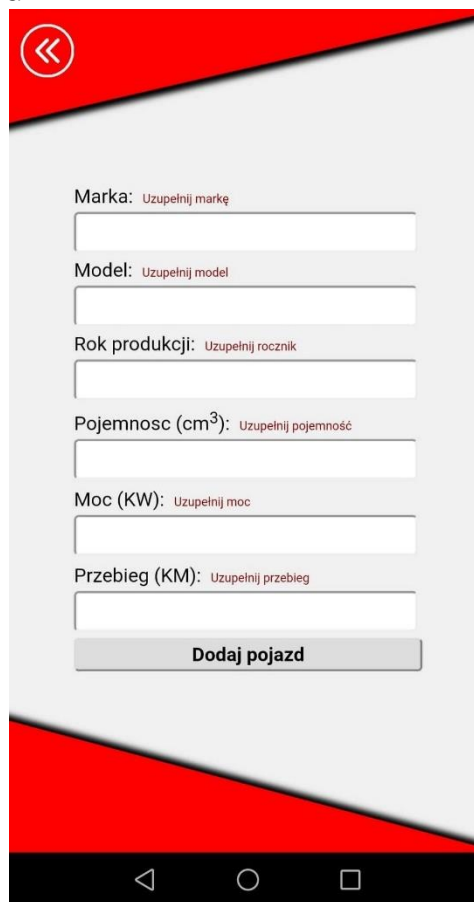


Formularz dodania samochodu z polami tekstowymi:

- Marka:
- Model:
- Rok produkcji:
- Pojemność (cm<sup>3</sup>):
- Moc (KW):
- Przebieg (KM):

Przycisk: Dodaj pojazd

*Screenshot 17:  
Widok ekranu dodania  
samochodu*



Formularz dodania samochodu z polami tekstowymi:

- Marka: Uzupełnij markę
- Model: Uzupełnij model
- Rok produkcji: Uzupełnij rocznik
- Pojemność (cm<sup>3</sup>): Uzupełnij pojemność
- Moc (KW): Uzupełnij moc
- Przebieg (KM): Uzupełnij przebieg

Przycisk: Dodaj pojazd

*Screenshot 18:  
Brak wprowadzenia danych*

W kolejnym widoku (Screenshot 17) dodania samochodu, użytkownikowi ukaże się sześć pól wraz z dwoma przyciskami funkcyjnymi. W samym centrum znajdują się pola, w które należy wpisać dane o samochodzie, którego należy dodać do konta zalogowanego użytkownika. Wprowadzić w tym miejscu trzeba takie informacje jak: marka, model, rok produkcji, pojemność silnika, moc kW czy przebieg.

Nie wprowadzenie danych choćby w jedno pole istotnych informacji, skutkuje wyświetleniem się nad danym miejscem komunikatu (Screenshot 18) o nie wprowadzeniu danych we wszystkich wymaganych miejscach. Bez tych danych nie będzie można dodać samochodu do listy pojazdów użytkownika.

Przycisk „Dodaj pojazd” wprowadza dane podane we wszystkich sześciu polach do bazy samochodów, tylko jeśli dane są wprowadzone poprawnie. W tym widoku znajduje się także przycisk cofnięcia do poprzedniego widoku. Wygląd i położenie tego przycisku jest taki sam jak w poprzednim podpunkcie (5.4).

## 5.6. Widok samochodu



*Screenshot 19: Widok samochodu*

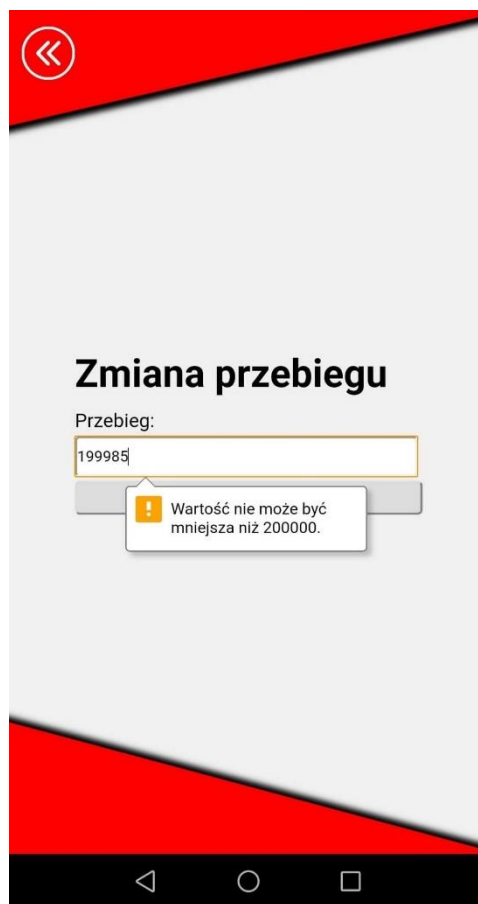
W zakładce „Widok samochodu” (Screenshot 19) znajdują się wszystkie dane na temat aktualnie wybranego samochodu użytkownika, które zostały wprowadzone do bazy. Dostrzec można tutaj za wyjątkiem danych wyświetlanych na ekranie, jeszcze cztery przyciski.

Pierwszy z nich odpowiada za przekierowanie do zmiany przebiegu. Kolejny z nich za przekierowanie do dodania czynności serwisowych (pod danymi o samochodzie). Będzie on wyświetlał wszystkie wprowadzone przez użytkownika informacje tj. kiedy został wykonany serwis oraz co zostało naprawione. Trzeci przycisk jest odpowiedzialny za przejście do widoku usunięcia pojazdu. Czwarty z nich, czyli już ostatni przycisk odpowiada za cofnięcie do poprzedniego widoku (został on opisany już w podpunkcie 5.4).

## 5.7. Widok zmiany przebiegu



*Screenshot 20:*  
*Widok ekranu zmiany przebiegu*



*Screenshot 21:*  
*Wprowadzenie błędnych danych*

W danym widoku (Screenshot 20), znajdują się trzy elementy, o odmiennych funkcjach. Pierwszy z nich jest polem w którym podany jest aktualny przebieg pojazdu. Został on pobrany z bazy danych

samochodów. Można go zaktualizować, tylko i wyłącznie po wprowadzeniu wartości wyższych niż aktualny przebieg – pozwoli to uniknąć błędnego wpisania przebiegu.

Jeśli jednak użytkownik wpisze mniejszy przebieg pojazdu, niż aktualny, wyświetli się mu komunikat o błędnie uzupełnionym polu (Screenshot 21).

Kolejnym elementem danego widoku jest przycisk „Aktualizuj przebieg”, którego funkcja odpowiada za pobranie aktualnie wpisanego przebiegu i porównanie go z przebiegiem istniejącym już w bazie. Jeśli pole zostanie uzupełnione poprawnie to w bazie przebiegów, będzie widniał najnowszy, zaktualizowany przebieg.

## 5.8. Widok dodania serwisu



*Screenshot 22:  
Widok dodawania serwisu*



*Screenshot 23:  
Efekt nie wprowadzenia danych*

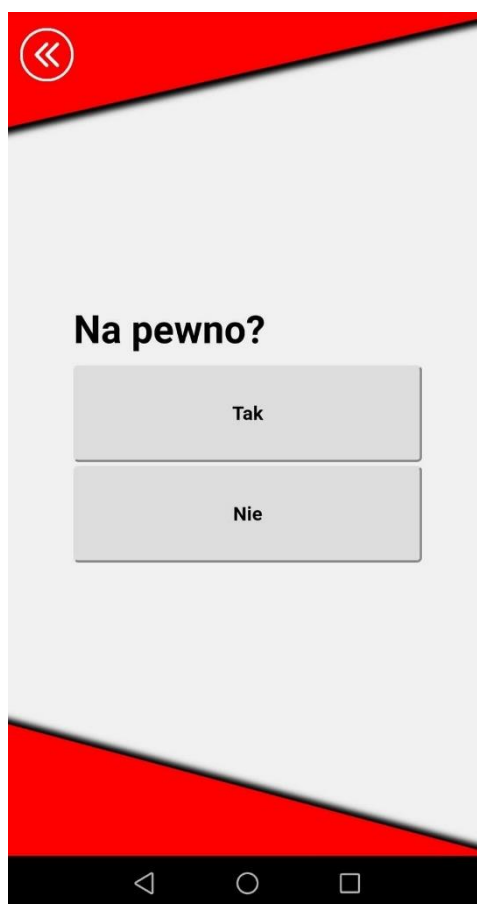
Kolejny widok przedstawia (Screenshot 22) możliwość dodania serwisowania pojazdu. Znajduję się tam wybór daty wykonanej czynności, którą wprowadza użytkownik. Poniżej wyboru daty znajduje



się pole, w które można wpisać wykonaną czynność, można tam wpisać np. samodzielna wymiana filtra powietrza, czy też wykonanie przeglądu w autoryzowanym serwisie.

Z przycisków funkcyjnych do dyspozycji jest m.in. „Dodaj serwis”, który wprowadza dane o każdym serwisowaniu pojazdu do bazy i wyświetla te wszystkie informacje w zakładce „Widok samochodu.”

## 5.9. Widok usunięcia samochodu



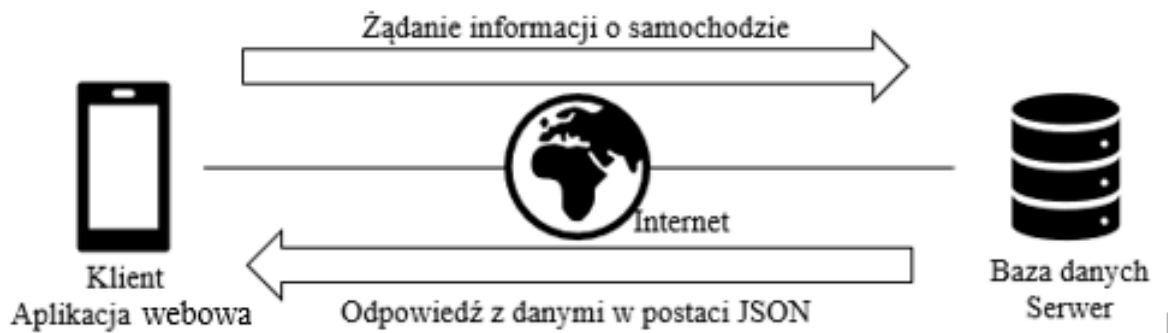
*Screenshot 24: Widok ekranu usunięcia pojazdu aplikacji*

Ostatni widok aplikacji (Screenshot 24) posiada dwa funkcyjne przyciski. Pierwszy „Tak” odpowiada za usunięcie aktualnie wybranego pojazdu z bazy danych wraz ze wszystkimi zamieszczonymi danymi oraz z dodanymi serwisami.

Natomiast przycisk „Nie”, anuluje funkcje usunięcia samochodu, zachowa dane nie naruszając ich i wróci do poprzedniego widoku.

## 6. Implementacja

W danym rozdziale zostanie opisana implementacja tytułowej aplikacji pod względem strony klienta, będącej aplikacją webową, jak i serwerową. Rozwiązania ukazane opierają się zarówno na technologiach, jak i narzędziach opisanych w poprzednich rozdziałach. Klient w tym rozumieniu jest aplikacją webową, przy użyciu Internetu przesyła zapytanie do serwera (serwer z bazą danych) aby otrzymać informacje o samochodzie. W tym celu wykorzystałem PHP do stworzenia warstwy aplikacji, natomiast MySQL wykorzystano jako bazę danych systemu.



*Diagram 3: Diagram obrazujący implementację serwera*

### 6.1. Baza danych

```
<?php
$link = new mysqli("mysql.cba.pl", "root", "root", "mysql");
if ($link -> connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
    exit();
}
?>
```

*Screenshot 25: Kod połączenia z baza MySQL*

W powyższym kodzie zapisane jest połączenie z bazą danych serwera wykorzystując język PHP. Został utworzony plik config.php w celu stworzenia połączenia między aplikacją webową, a bazą danych. Było to dla mnie cenne, ponieważ wykorzystując jakiekolwiek zapytania do bazy, czy też wysłanie elementów do niej, pozwoliło mi to stworzyć prostą zmienną związaną z połączeniem do bazy.

## 6.2. Logowanie

```
if(empty($username_err) && empty($password_err)){  
    $sql = "SELECT * FROM users WHERE username='$username'";  
    if($results = mysqli_query($link,$sql)){  
        if(mysqli_num_rows($results) == 1){  
            $row = mysqli_fetch_assoc($results);  
            $param_username = $row["username"];  
            if($param_username === $username){  
                if(md5($password) === $row["password"]){  
                    $_SESSION["loggedin"] = true;  
                    $_SESSION["userid"] = $row['id'];  
                    $_SESSION["user"] = $row['name']." ".$row['surname'];  
                    header("location: main.php");  
                }  
            }  
        }  
    }  
}
```

*Screenshot 26: Kod logowania się*

Dany kod pobiera wprowadzone dane przez użytkownika do konkretny pól, po czym porównuje je z elementami znajdującymi się w bazie danych. Jeżeli elementy będą do siebie pasować użytkownik zaloguje się do aplikacji. Natomiast przy niepowodzeniu zalogowania klient dostanie informacje o błędnym loginie, bądź hasle.

## 6.3. Wylogowanie

```
<?php  
session_start();  
  
$_SESSION = array();  
  
if (ini_get("session.use_cookies")) {  
    $params = session_get_cookie_params();  
    setcookie(session_name(), '', time() - 42000,  
        $params["path"], $params["domain"],  
        $params["secure"], $params["httponly"]  
    );  
}  
session_destroy();  
header("location: login.php");  
?>
```

*Screenshot 27: Kod wylogowania się*

Podany kod umożliwia wylogowania się z aktualnej sesji, w której w danym momencie przebywa użytkownik, a co za tym idzie pozwala mu na wylogowanie z swojego konta.

## 6.4. Rejestracja

```
if(empty($username_err) && empty($password_err) && empty($surname_err) && empty($name_err)){  
    $password = md5($password);  
  
    $sql_check = "SELECT * from users where username='$username'";  
    $result=mysqli_query($link,$sql_check);  
    if (mysqli_num_rows($result) == 0)  
    {  
  
        $sql = "INSERT INTO users (username,password,name,surname) VALUES ('$username','$password','$name','$surname')";  
  
        if($result = mysqli_query($link,$sql)){  
  
            $registered = "Konto utworzone, możesz się teraz zalogować";  
            header("location: login.php");  
        }  
    }  
}
```

*Screenshot 28: Kod rejestracji*

Kod rejestracji pozwala po wprowadzeniu wszystkich danych, na dodanie nowego użytkownika do bazy danych. Oczywiście sprawdzając, czy wprowadzona przez niego nazwa nie jest już przypadkiem przez inne konto zajęta.

## 6.5. Aktualny użytkownik

```
<?php  
  
session_start();  
error_reporting(E_ALL & ~E_NOTICE);  
  
if(isset($_SESSION['loggedin'])){  
  
    }  
else{  
  
        header("location: login.php");  
    }  
}  
?>
```

*Screenshot 29: Kod aktualnie zalogowanego użytkownika*

Wyżej podany kod umożliwia na wyświetlenie danych takich jak: imię i nazwisko, które zostały wprowadzone przy wykonaniu rejestracji, użytkownika który jest w danym momencie zalogowany.

## 6.6. Usunięcie danych

```
<?php

session_start();
//error_reporting(E_ALL & ~E_NOTICE);

if(isset($_SESSION['loggedin'])){
    require_once "config.php";

    if(isset($_SESSION['selectedcar'])){

        $carid = $_SESSION["selectedcar"];

        if(isset($_POST["yes"])){

            $sql1 = "DELETE FROM cars where id=$carid";
            $sql2 = "DELETE FROM services where car_id=$carid";

            $result1 = $link->query($sql1);
            $result2 = $link->query($sql2);

            header("location: select_car.php");
        }
    }
}
```

*Screenshot 30: Kod usunięcia samochodu*

W powyższym kodzie mamy możliwość usunięcia danych z bazy, w tym przypadku będzie to pojazd osobowy. Przy usunięciu go otrzymamy najpierw zapytanie, czy jesteśmy pewni wykonania danej czynności. Zgadzając się na to, użytkownik straci wraz z samochodem, wszystkie serwisowania jakie były w nim przeprowadzone.

## 6.7. Aktualizacja danych

```
if($_SERVER["REQUEST_METHOD"] == "POST"){

    if(empty(trim($_POST["mileage"]))) {
        $mileage_err = "Uzupełnij przebieg";
    } else {
        $mileage = trim($_POST["mileage"]);
    }

    if(empty($mileage_err)){

        $carid = $_SESSION["selectedcar"];

        $sql = "UPDATE cars SET mileage='$mileage' WHERE id='$carid'";

        if($results = mysqli_query($link,$sql)){

            $_SESSION["selectedcar_mileage"] = $mileage;
            header("location: display_car.php");
            $mileage_err = "";
        }
    }
}
```

*Screenshot 31: Kod aktualizacji przebiegu*

Dany kod odpowiada za sprawdzenie wprowadzonych danych i porównanie go z elementem (przebieg) z bazy danych. W tym przypadku, jeżeli użytkownik nie poda wartości wyższej od aktualnego przebiegu pojazdu, to nie zostanie ten przebieg zaktualizowany.

## 6.8. Dodawanie elementów

```
if($_SERVER["REQUEST_METHOD"] == "POST"){

    if(empty(trim($_POST["date"]))){
        $date_err = "Uzupełnij datę";
    } else{
        $date = trim($_POST["date"]);
    }

    if(empty($_POST["what"])){
        $what_err = "Uzupełnij czynność";
    } else{
        $what = $_POST["what"];
    }

    if(empty($what_err) && empty($date_err)){

        $carid = $_SESSION["selectedcar"];
        $sql = "INSERT INTO services (car_id,datewhen,what) VALUES ('$carid','$date','$what')";

        if($results = $link->query($sql)){

            header("location: display_car.php");
```

*Screenshot 32: Kod dodania serwisowania*

```
if($_SERVER["REQUEST_METHOD"] == "POST"){

    require_once "config.php";
    $userid = $_SESSION["userid"];

    if(empty($make_err) && empty($model_err) && empty($year_err) && empty($capacity_err)
        && empty($mileage_err) && empty($power_err)){

        $sql = "INSERT INTO cars (user_id,make,model,year,mileage,power,capacity)
            VALUES ('$userid','$make','$model','$year','$mileage','$power','$capacity')";

        if($results = mysqli_query($link,$sql)){

            header("location: select_car.php");

        }else{

            $make_err = "Błąd bazy danych";

        }

    }
```

*Screenshot 33: Kod dodania samochodu*

Oba wyżej podane kody, odpowiadają za wpisanie nowego elementu do bazy danych. W pierwszym przypadku (Screenshot 32) element ma wprowadzać notatki serwisowe dla konkretnego samochodu i wyświetlać je w formie listy. Natomiast w drugim (Screenshot 33),

elemencie będzie nowy pojazd, który zostanie przypisany do aktualnie zalogowanego profilu i będzie wyświetlany w formie listy.

## **Podsumowanie**

Zainspirowany zamiłowaniem oraz wiedzą zdobytą w zakresie motoryzacji, postanowiłem w pewien sposób połączyć pasję z swoim kierunkiem kształcenia. Co sprawiło powstanie tego projektu oraz poszukiwań jak najlepszych rozwiązań w danej dziedzinie. Przy pomocy nakreślonych cech, zaprojektowałem i zaimplementowałem, aplikację „Mobilna Fura” – zatem cel mojej pracy został osiągnięty. Aplikacja została udostępniona użytkownikom do testów i korzystania z niej. Sam także z niej korzystam.

Przeszedłem dość długą drogę, aby osiągnąć postawione sobie priorytety projektowe realizowane na wielu płaszczyznach. Wszystkie rozwiązania techniczne udało mi się zrealizować zgodnie z planem projektowym.

Należy pamiętać, że aplikacja która została opracowana, jest dopiero prototypem, może posiadać błędy. Jestem przekonany, iż aplikacja ma spory potencjał do dalszego jej rozwoju.

## **Zawartość płyty**

- Biblioteki/
- Kody\_zrodlowe/ - kod źródłowy aplikacja Mobilna Fura
- Wersja\_elektroniczna\_pracy\_pisemnej/
- Strona/



## **Bibliografia i netografia**

### **Indeks diagramów**

Diagram 1: Diagram przypadków użycia, w notacji UML.....	19
Diagram 2: Diagram konceptualny klas występujących w aplikacji, w notacji UML.....	20
Diagram 3: Schemat obrazujący implementacje serwera.....	50

### **Indeks ilustracji**

Ilustracja 1: Przykładowy kod wykorzystując język PHP.....	25
Ilustracja 2: Struktura reguł w arkuszu CSS.....	27
Ilustracja 3: Rodzaje czcionki w CSS.....	28
Ilustracja 4: Programy pozwalające otworzyć format JSON.....	36
Ilustracja 5: Struktura SOAP.....	37

### **Indeks screenshotów**

Screenshot 1: Główne menu w aplikacji Książka serwisowa pojazdów.....	9
Screenshot 2: Przykładowa zakładka w aplikacji AUTODOC CLUB.....	12
Screenshot 3: Zakładka przebieg kilometrów pojazdu w aplikacji My Car Service.....	15
Screenshot 4: Budowa języka HTML.....	23
Screenshot 5: Przykładowy styl z arkusza CSS.....	26
Screenshot 6: Struktura budowy programu Brackets.....	28
Screenshot 7: Wygląd okna aplikacji phpMyAdmin.....	30
Screenshot 8: Interfejs programu Xampp.....	32
Screenshot 9: Widok ekranu logowania użytkownika.....	41

Screenshot 10: Efekt nie wprowadzenia żadnych danych.....	41
Screenshot 11: Efekt wprowadzenia błędnych danych.....	41
Screenshot 12: Widok ekranu rejestracji.....	52
Screenshot 13: Efekt wprowadzenie błędnych danych.....	52
Screenshot 14: Widok ekranu głównego aplikacji.....	43
Screenshot 15: Pierwsze zalogowanie się.....	44
Screenshot 16: Widok zakładki „Moje Samochody”.....	44
Screenshot 17: Widok dodania samochodu.....	45
Screenshot 18: Brak wprowadzenia danych.....	45
Screenshot 19: Widok samochodu.....	46
Screenshot 20: Widok ekranu zmiany przebiegu.....	47
Screenshot 21: Wprowadzenie błędnych danych.....	47
Screenshot 22: Widok dodania serwisu.....	48
Screenshot 23: Efekt nie wprowadzenia danych.....	48
Screenshot 24: Widok ekranu usunięcia pojazdu aplikacji.....	49
Screenshot 25: Kod połączenia z bazą MySQL.....	50
Screenshot 26: Kod logowania się.....	51
Screenshot 27: Kod wylogowania się.....	51
Screenshot 28: Kod rejestracji.....	52
Screenshot 29: Kod aktualnie zalogowane użytkownika.....	52
Screenshot 30: Kod usunięcie samochodu.....	53
Screenshot 31: Kod aktualizacji przebiegu.....	54
Screenshot 32: Kod dodania serwisowania.....	54
Screenshot 33: Kod dodania samochodu.....	55

1. <https://xampp-windows.softonic.pl/>
2. <https://en.wikibooks.org/>
3. <https://pl.wikipedia.org/>

4. <https://www.proformat.pl/blog/> - wykorzystanie PHP
5. <https://www.w3schools.com/php/> - PHP tutorial
6. Andrew Hunt, David Thomas; „Pragmatyczny programista, od czeladnika do mistrza”; Wydawnictwo WNT; Warszawa; 2002; 354 strony
7. <https://stackoverflow.com/> - strona z pytaniami i odpowiedziami dla programistów
8. Bruegge Bernd, Inżynieria oprogramowania w ujęciu obiektowym: UML, wzorce projektowe i Java; Gliwice; Helion; 2011, ISBN 978-83-246-2872-8
9. <https://searchapparchitecture.techtarget.com/definition/SOAP> - definicje protokołu SOAP
10. Ferreira, Otavia: Semantic Web Services: A RESTful Approach, Proceedings of IADIS International Conference WWW/Internet; 2009
11. Sacha, Krzysztof: Inżynieria Oprogramowania; Wydawnictwo Naukowe PWN; 2010
12. <http://mobilnafura.cba.pl/> - strona aplikacji
13. <https://play.google.com/store/apps/details?id=com.macios.zek.vehicleservicebook/> - link do aplikacji „Książka Pojazdu”
14. <https://play.google.com/store/apps/details?id=com.de.auto.doc.club/> - link do aplikacji „AUTODOC CLUB”
15. <https://play.google.com/store/apps/details?id=com.mycarservice.mycarservice/> - link do aplikacji „My Car Service”