



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI

INSTYTUT INFORMATYKI

Projekt dyplomowy

System wspomagania klienta w procesie naprawy i konserwacji samochodu
Customer support system in the process of car repair and maintenance

Autorzy:	<i>Roman Hodzinets, Mateusz Tondos, Adrian Jeleń</i>
Kierunek studiów:	Informatyka
Opiekun pracy:	<i>dr inż. Witold Alda</i>

Kraków, 2022

Streszczenie

Praca „System wspomagania klienta w procesie naprawy i konserwacji samochodu” przedstawia system umożliwiający właścicielom samochodów skrócenie czasu potrzebnego na ich naprawę w warsztacie samochodowym. System wykorzystuje wzorzec typu "odwrócenie sterowania" (ang. Inversion of Control) pomiędzy warsztatem, a jego klientem. W tym modelu klient jest odpowiedzialny za zamieszczenie zlecenia w systemie, a rolą warsztatów jest pozyskanie klienta poprzez złożenie mu najatrakcyjniejszej oferty. W projekcie przewidziano takie elementy jak: złożenie zlecenia, powiązanie usługodawcy ze zleceniem, umówienie spotkania w celu odebrania, a następnie zwrotu samochodu oraz monitorowanie procesu naprawy.

Wynikiem projektu jest sprawny i wygodny dla użytkownika system, działający na komputerach osobistych i urządzeniach mobilnych. Projekt demonstruje potencjał zastosowania systemu informatycznego w modelu odwróconego sterowania dla branży napraw samochodowych za pomocą przykładowej aplikacji przeglądarkowej.

Podczas implementacji rozwiązania zastosowano nowoczesne technologie wytwarzania oprogramowania. Część serwerowa została napisana w języku Java i wykorzystuje framework Spring oraz oprogramowanie bazodanowe PostgreSQL. Część użytkownika powstała w oparciu o języki Typescript / Javascript, HTML i CSS. Wykorzystuje bibliotekę React Query do zarządzania stanem danych serwerowych w aplikacji klienckiej, bibliotekę Formik do formularzy, bibliotekę Material UI do implementacji elementów interfejsu użytkownika oraz bibliotekę React Router do nawigacji.

Słowa kluczowe

Odwrócenie sterowania, branża napraw samochodowych, aplikacja internetowa, Java, Spring, React, TypeScript, JavaScript, PostgreSQL

Summary

Diploma thesis „Customer support system in the process of car repair and maintenance” represents a system that allows car owners to shorten the time required for its repair in the car repair shop. System uses Inversion of Control type of the model between repair shop and the client. In this model client is responsible for placing the order in the system, and it is repair shop’s role to find the client by offering the best deal. In the project following elements were foreseen placing the order, connecting serviceman with the order, making an appointment for a car pickup, following up with the car return and the repair process monitoring.

The result of the project is efficient and convenient system for the user, that works on personal computers as well as on mobile devices. Project demonstrates potential of the information system usage in the inversion of control model for car repair market with help of web application as an example.

During solution implementation modern technologies of software development were used. Server part (backend) was written using Java language combined with Spring framework as well as database software PostgreSQL. User interface part (frontend) was based on Typescript, JavaScript, HTML and CSS languages. React Query library was used for server data state management in client application, Material UI for elements of user interface, Formik library – for forms and React Router – for navigation.

Keywords

Inversion of control, car repair market, web application, Java, Spring, React, TypeScript, JavaScript, PostgreSQL.

Spis treści

1	Wstęp	6
1.1	Charakter i cel pracy	6
1.2	Struktura pracy	7
2	Specyfikacja problemu.....	8
2.1	Opis problemu	8
2.2	Idea rozwiązania	8
3	Koncepcja systemu	10
3.1	Grupy użytkowników aplikacji – podział ze względu na funkcje biznesowe	10
3.2	Wymagania funkcjonalne – ogólny zbiór procesów	10
3.3	Analiza wymagań funkcjonalnych	11
3.4	Specyfikacja informacji przetwarzanych przez moduły interfejsu użytkownika i serwisu	16
3.5	Specyfikacja listy końcówek API potrzebnych do zapewnienia danych widokom ..	18
3.6	Wymagania niefunkcjonalne	20
3.7	Analiza ryzyka	20
4	Projekt i implementacja	24
4.1	Analiza domeny	24
4.1.1	Uprawnienia wykorzystywane w systemie:	24
4.1.2	Kategorie zleceń i dodatkowe opcje dostępne w systemie	24
4.1.3	Dodatkowe opcje systemu, których Klient może oczekiwać od Serwisu w zakresie obsługi zlecenia.....	24
4.1.4	Predefiniowane usługi dostępne w aplikacji:	24
4.1.5	Predefiniowane problemy z samochodem dostępne w aplikacji:.....	25
4.2	Trójwarstwowa architektura systemu	25
4.3	Stos technologiczny i struktura kodu.....	26
4.4	Szczegóły architektury trójwarstwowej.....	29

4.4.1	Warstwa prezentacji	29
4.4.2	Warstwa logiki biznesowej	40
4.4.3	Warstwa bazy danych.....	53
5	Przebieg projektu	58
5.1	Podział pracy	58
5.2	Dziennik projektu	58
5.3	Zastosowane techniki i praktyki	62
6	Podsumowanie	63
6.1	Porównanie specyfikacji systemu z dostarczoną wersją	63
6.2	Ocena przebiegu projektu i produktu końcowego	64
6.3	Plan rozwoju aplikacji	64
7	Słownik pojęć.....	65
8	Spis rysunków	66
9	Spis tabel.....	68
	Bibliografia.....	69

1 Wstęp

1.1 Charakter i cel pracy

Celem pracy jest zademonstrowanie potencjału zastosowania systemu informatycznego w modelu odwróconego sterowania. W takim modelu klient nie poszukuje firmy świadczącej dany typ usługi, za to wystawia ogłoszenie dotyczące zapotrzebowania. Firma wykonująca usługę może złożyć propozycję dla klienta. Ze względu na sposób działania model sprawdzi się na rynkach w których popyt i podaż równoważą się lub występuje nadmiar podaży. Firma mająca nadmiar klientów może nie być zainteresowana ponoszeniem dodatkowego kosztu związanego z poszukiwaniem nowych klientów, co jest narzucone przez model odwróconego sterowania. Kolejnym wymaganiem dotyczącym rynku jest możliwość łatwego i jednoznacznego spisania wymagań dotyczących oczekiwań klienta. Model sprawdzi się w sytuacji, gdy wymaganie jest jasno określone i rezultat nie zależy w istotny sposób od wykonawcy. Przykładem może być firma świadcząca usługi związane z transportem np. w branży taksówkarskiej. Wynik jest łatwy do zweryfikowania w sposób obiektywny – klient jest w miejscu, w którym chciał być. Nie sprawdzi się w sytuacji, gdy zadowolenie klienta z jakości jest ściśle związane z subiektywnym poczuciem estetyki, gustem lub upodobaniami oraz nie daje się w łatwy sposób przedstawić. Przykładem może być branża gastronomiczna. Jedzenie przygotowywane przez konkretną restaurację ma charakter indywidualny (pominąwszy sieciówki) i potrawa pomimo tej samej nazwy w różnych lokalach, będzie smakować inaczej. W tym przypadku opinie innych klientów mogą nie być pomocne, a wybór konkretnej restauracji będzie indywidualny i zależny od gustu.

W naszej pracy została wybrana branża serwisów samochodowych. Spełnia ona wymagania dotyczące równowagi popytu i podaży oraz w obecnych czasach naprawa najczęściej polega na diagnozie i wymianie części na nowe. Wynik jest łatwy do zweryfikowania dla większości typów usterek, a sposób prowadzenia prac wynika z technologii. Rezultat projektu w postaci aplikacji wykorzystującej elementy klasycznej tablicy ogłoszeń oraz modelu odwróconego sterowania powinien być dostępny na komputerach osobistych i urządzeniach mobilnych poprzez przeglądarkę z dostępem do Internetu. Taki model zapewnia wygodne rozwiązania dla klienta indywidualnego. korzystającego z usług serwisów samochodowych, który docelowo będzie czerpał korzyści w postaci wysokiej jakości usług, komfortu wyboru warsztatu samochodowego oraz oszczędności czasu.

1.2 Struktura pracy

Rozdział pierwszy opisuje cel pracy oraz jej strukturę. W drugim rozdziale znajduje się szczegółowa specyfikacja problemu oraz idea rozwiązania. W trzecim rozdziale umieszczono wymagania funkcjonalne i нефункционалне, przedstawiono koncepcję rozwiązania ze szczególnym naciskiem na funkcje biznesowe i przetwarzane informacje oraz analizę ryzyka. Czwarty rozdział opisuje architekturę systemu oraz szczegóły. Piąty rozdział przedstawia przebieg prac, podział zadań oraz zastosowane techniki. W szóstym rozdziale podsumowano rezultat wykonanych prac oraz wskazano potencjalne możliwości dalszego rozwoju systemu. W ostatnich trzech rozdziałach znajdują się spisy rysunków, tabel oraz bibliografia.

2 Specyfikacja problemu

2.1 Opis problemu

Każdy właściciel samochodu będący osobą prywatną potrzebuje:

- sprawdzać stan swojego samochodu,
- wykonywać usługi konserwacyjne,
- na bieżąco naprawiać samochód w celu utrzymania jego prawidłowego działania,
- zależy mu na tym, aby usługa była wykonana w krótkim lub określonym terminie i w dobrej jakości.
- oraz chciałby mieć możliwość uzyskania informacji o postępie prac wykonywanych w ramach usługi.

Aby spełnić te potrzeby, właściciel musi serwisować swój samochód. Aby to zrobić, musi umówić się na wizytę w warsztacie samochodowym, a więc:

- znaleźć listę ofert warsztatów w swojej okolicy,
- ustalić jakość usług świadczonych przez warsztaty na podstawie znalezionych informacji i opinii,
- skontaktować się z wybranymi salonami w celu zapytania o szczegóły oraz ustalenia dostępności terminów i czasu naprawy,
- dostarczyć i odebrać samochód.

W takim tradycyjnym procesie polegającym na aktywnym działaniu klienta indywidualnego, klient musi poświęcić czas na przeprowadzenie poszukiwań w celu wybrania najlepszej oferty spośród ofert wielu serwisów. Dodatkowo, weryfikacja wiarygodności serwisu może być trudna kiedy klient nie ma doświadczenia w naprawianiu samochodów.

2.2 Idea rozwiązania

W proponowanej wizji rozwiązania, klient wykonuje proces wyboru usługi korzystając z przeglądarki. W tym procesie opisuje na stronie internetowej swoje zapotrzebowanie na usługę lub problem związany z samochodem oraz wskazuje możliwy termin rozpoczęcia realizacji zlecenia. Na podstawie tych informacji, klient otrzymuje oferty serwisów samochodowych, które są zainteresowane jego zleceniem. Oferty te mogą zawierać między innymi informacje takie jak przewidywany czas wykonania oraz szacowany koszt usługi, oceny i opinie o serwisie oraz komentarz serwisu odnośnie oferty.

Klient na podstawie zebranych ofert będzie mógł wybrać jedną z nich, a następnie przejść przez proces wykonania usługi będąc wspomaganym przez tę samą stronę internetową. Z punktu widzenia klienta, proces wykonania usługi polegać będzie na:

- przekazaniu samochodu do serwisu w umówionym miejscu i czasie,
- oczekiwaniu na wykonanie usługi przez serwis,
- odebraniu samochodu od serwisu w umówionym miejscu.

Serwisy oferujące usługi klientom będą miały dostęp do:

- listy wszystkich otwartych ofert,
- przygotowania propozycji oferty,
- w przypadku akceptacji oferty – do wprowadzania informacji o procesie naprawy i bieżącej komunikacji z klientem.

Dostęp do funkcji dedykowanych dla serwisów samochodowych będzie możliwy po przejściu przez weryfikację administratora strony internetowej. Administrator strony internetowej za weryfikację i utrzymanie dostępu do aplikacji od każdego serwisu będzie pobierał opłatę abonamentową. Weryfikacja i pobieranie opłaty będą wykonywane poza stroną internetową.

3 Koncepcja systemu

3.1 Grupy użytkowników aplikacji – podział ze względu na funkcje biznesowe

Aplikacja będzie dostępna z poziomu przeglądarki dla dwóch grup użytkowników:

- Grupa użytkowników typu klient indywidualny,
- Grupa użytkowników typu serwis - do której dodawani będą użytkownicy reprezentujący autoryzowane serwisy.

3.2 Wymagania funkcjonalne – ogólny zbiór procesów

Lista procesów wymaganych do minimalnego działania systemu:

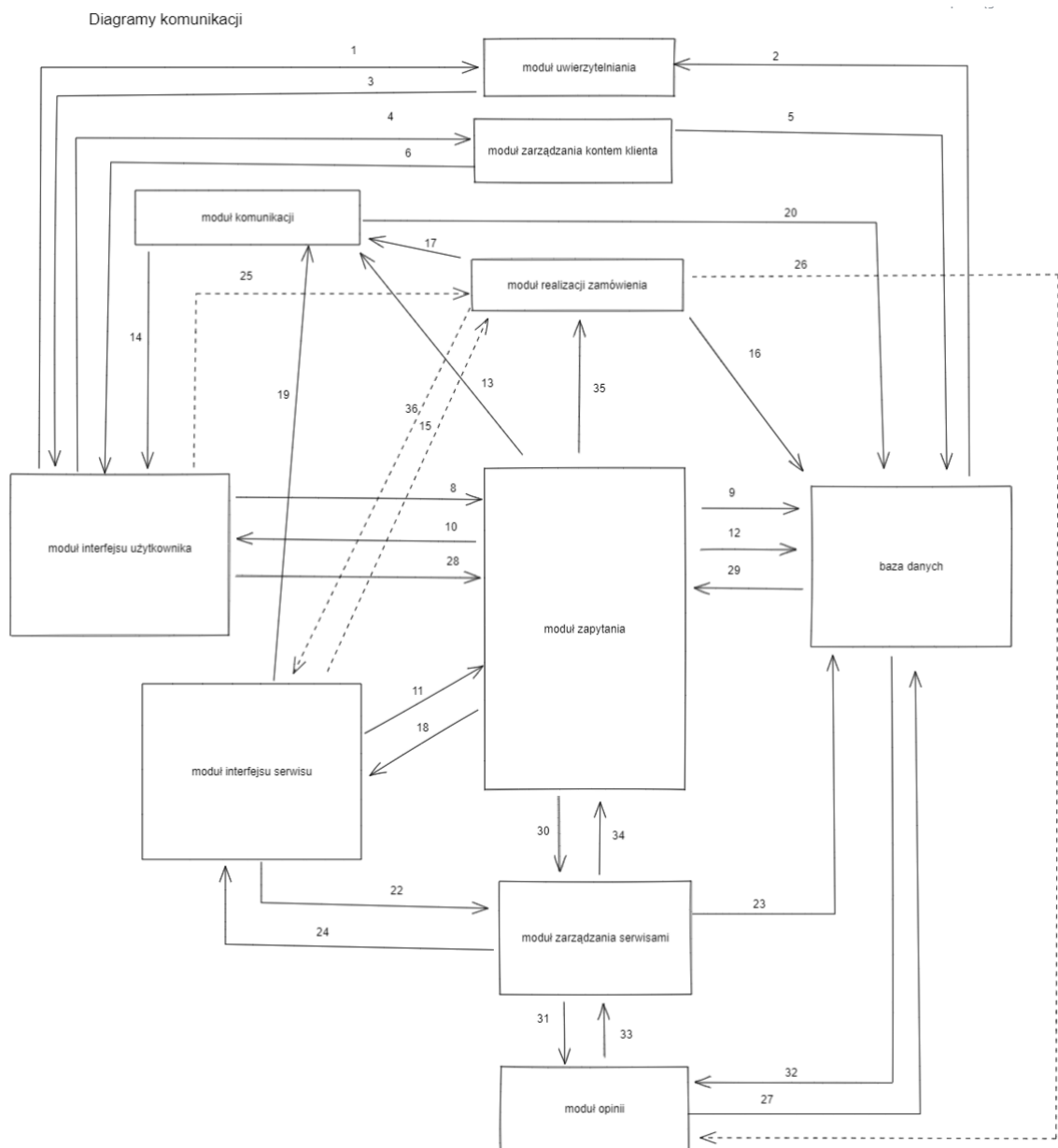
- Logowanie użytkownika (klienta i serwisu),
- Rejestracja użytkownika (klienta),
- Obsługa konta serwisu w systemie,
- Tworzenia zlecenia przez klienta,
- Wyświetlania otwartych zleceń i dodawanie oferty przez serwis,
- Wyświetlanie listy ofert przekazanych przez serwisy,
- Wybór oferty przez użytkownika
- Przekazanie samochodu do serwisu i rozpoczęcia wykonywania usługi,
- Zmiana statusu zamówienia związana z przebiegiem wykonywania usługi,
- Komunikacja klienta i serwisu,
- Zakończenie usługi przez serwis i odbiór samochodu,
- (możliwość rozszerzenia) Wystawienie opinii przez użytkownika.

Funkcjonalności opinii użytkownika nie rozważano w pierwszej wersji tworzonego systemu ze względu na to, że nie stanowi ona wymaganego kroku w celu realizacji zlecenia. Dodatkowo w przypadku małej ilości użytkowników bezpośrednio po rozpoczęciu działania systemu istnieje ryzyko potencjalnego nadużycia ze strony złośliwych użytkowników poprzez wprowadzanie pojedynczych nieuczciwych ocen istotnie wpływające na opinię serwisu. Mogłoby to spowodować zniechęcenie do korzystania z systemu przez serwisy samochodowe oraz użytkowników.

3.3 Analiza wymagań funkcjonalnych

Istotnym elementem zapewniającym prawidłową realizację wymaganych funkcji jest obieg informacji wewnątrz systemu. W celu przejrzystego przedstawienia komunikacji podzielono system na moduły uwzględniając stawiane wymagania funkcjonalne. Następnie zidentyfikowano moduły komunikujące się podczas realizacji poszczególnych funkcji.

Wynik ustalonego przepływu informacji z podziałem na moduły przedstawiono w formie diagramu komunikacji na Rysunek 3.3.1. Wskazane moduły przedstawiają logiczny (abstrakcyjny) podział aplikacji i wraz z informacjami o przekazywanych danych są elementem przygotowawczym do stworzenia listy właściwych widoków, końcówek API oraz struktury bazy danych będących częścią systemu.



Rysunek 3.3.1 Diagram komunikacji

Poniżej przedstawiono szczegółowy opis przepływu informacji pomiędzy modułami podczas realizacji funkcji systemowych:

Logowanie użytkownika

Poprzez moduł interfejsu użytkownika użytkownik przekazuje (1) swój login i hasło do modułu uwierzytelniania.

Moduł uwierzytelniania pobiera (2) z bazy danych aktualnie ustawiony login i hasło i porównuje je z przekazanymi przez użytkownika danymi.

Moduł uwierzytelniania przekazuje (3) do modułu użytkownika wynik logowania wraz z danymi zalogowanego użytkownika.

Rejestracja zleceniodawcy

Moduł interfejsu użytkownika przekazuje (4) modułowi zarządzania kontem klienta dane nowego użytkownika.

Moduł zarządzania kontem klienta sprawdza, czy dany użytkownik istnieje. Jeśli nie, to rejestruje (5) nowego użytkownika w bazie danych.

Moduł zarządzania kontem klienta przekazuje (6) modułowi interfejsu użytkownika wynik rejestracji.

Obsługa konta serwisu w systemie

Moduł interfejsu serwisu przekazuje (22) modułowi zarządzania serwisami aktualne dane dotyczące wybranego serwisu lub nowego serwisu. Moduł zarządzania serwisami zapisuje (23) dane informacyjne dotyczące serwisu w bazie.

Moduł zarządzania serwisami przekazuje (24) modułowi interfejsu serwisu wynik zapisu.

Proces tworzenia zlecenia przez klienta

Moduł interfejsu użytkownika przekazuje (8) modułowi zapytania wypełniony formularz tworzenia zlecenia.

Moduł zapytania rejestruje (9) w bazie danych nowe zlecenie na podstawie przekazanych danych.

Moduł zapytania zwraca (10) użytkownikowi informację o statusie rejestracji zlecenia.

Proces wyświetlania otwartych zleceń przez serwis i dodawania oferty przez serwis

Moduł interfejsu serwisu pobiera (18) listę ofert zapytań o zlecenia. Moduł interfejsu serwisu wysyła (11) wypełniony formularz oferty do modułu zapytania. Moduł zapytania rejestruje w bazie nową ofertę (12).

Moduł zapytania przekazuje (13) modułowi komunikacji informację o nowej ofercie.

Moduł komunikacji wysyła (14) wybranym przez użytkownika kanałem informację o nowej ofercie.

Wyświetlanie listy ofert przekazanych przez serwisy

Moduł interfejsu użytkownika wysyła zapytanie (8) o aktywne oferty dla danego zapytania. Moduł zapytania pobiera (29) informacje o aktualnych zapytaniach z bazy danych. Moduł zapytania wysyła (30) do modułu zarządzania serwisami zapytanie o informacje poszczególnych serwisów. Moduł zarządzania serwisami wysyła (31) zapytanie do modułu opinii. Moduł opinii pobiera (32) wszystkie niezbędne informacje z bazy danych, a następnie przekazuje (33) opinie do modułu zarządzania serwisami. Moduł zarządzania serwisami przekazuje (34) informacje na temat serwisów do modułu zapytania. Moduł zapytania zwraca (10) listę ofert z zapytaniami do modułu interfejsu użytkownika.

Wybór oferty przez użytkownika

Użytkownik wybiera ofertę, a następnie moduł interfejsu użytkownika wysyła (28) informację o wybranej ofercie do modułu zapytania. Moduł zapytania przesyła (35) wszystkie potrzebne informacje do modułu realizacji zamówienia. Moduł realizacji zamówienia zapisuje (16) nowy stan zamówienia w bazie, a następnie wysyła (17) informację o stanie do modułu komunikacji. Moduł komunikacji przekazuje (14) komunikat o statusie zmian do modułu interfejsu użytkownika.

Proces zmiany statusu zamówienia związana z przebiegiem wykonywania usługi

Moduł interfejsu serwisu wprowadza (15) nowy status zamówienia, do modułu realizacji zamówienia. Moduł realizacji zamówienia wprowadza (16) informację o nowym statusie

zamówienia do bazy danych. Moduł realizacji zamówienia przekazuje (17) informację o nowym statusie do modułu komunikacji. Moduł komunikacji wysyła (14) wybranym przez użytkownika kanałem informację o nowym statusie zamówienia.

Proces przekazania samochodu do serwisu i rozpoczęcia wykonywania usługi

Moduł interfejsu serwisu wywołuje proces zmiany statusu zamówienia, zmieniając jego aktualny status na „w trakcie wykonywania”.

Proces komunikacji klienta i serwisu

Moduł interfejsu serwisu wysyła (15) nową wiadomość do modułu realizacji zamówienia. Moduł realizacji zamówienia wprowadza (16) informację o nowej wiadomości do bazy danych. Moduł realizacji zamówienia przekazuje (17) informację o nowej wiadomości do modułu komunikacji. Następnie moduł komunikacji wysyła ją (14) do modułu interfejsu użytkownika. Moduł interfejsu użytkownika wysyła (25) nową wiadomość do modułu realizacji zamówienia. Moduł realizacji zamówienia wprowadza (16) informację o nowej wiadomości do bazy danych.

Proces zakończenia usługi przez serwis i odbiór samochodu przez klienta

Moduł realizacji zamówienia wywołuje proces zmiany statusu zamówienia, zmieniając jego aktualny status na zakończony.

Proces wystawienia opinii przez użytkownika

Moduł interfejsu użytkownika przesyła (25) modułowi realizacji zamówienia opinię po wykonaniu zamówienia. Moduł realizacji zamówienia sprawdza możliwość wystawienia opinii.

Moduł realizacji zamówienia przekazuje (26) opinię do modułu opinii uwzględniając w nim aktualny serwis. Moduł przelicza aktualną ocenę serwisu i zapisuje (27) wyniki wraz z opinią w bazie.

3.4 Specyfikacja informacji przetwarzanych przez moduły interfejsu użytkownika i serwisu

W celu umożliwienia realizacji funkcji biznesowych system komunikuje się z użytkownikiem końcowym poprzez zbieranie i przedstawianie informacji. W Tabeli 3.4.1 i Tabela 3.4.2 przygotowano rodzaj danych przetwarzanych przez moduł interfejsu użytkownika oraz moduł interfejsu serwisu. Przetwarzane informacje przygotowano w oparciu o widoki dostępne dla użytkownika (klient i serwis) ze względu, że będzie on właściwym nadawcą i odbiorcą wszystkich danych. Dane zgromadzone w tabeli w dalszej części zostaną wykorzystane do spójnego przygotowania projektu właściwych widoków, odpowiadających im zapytań API oraz schematu bazy danych.

1) Moduł interfejsu użytkownika:

Tabela 3.4.1 Dane przetwarzane przez poszczególne widoki – moduł interfejsu użytkownika

Nazwa widoku	Przetwarzane dane
Logowanie użytkownika	<ul style="list-style-type: none">• Dane wejściowe: Login,• Dane wyjściowe: Rezultat logowania
Rejestrowanie użytkownika	<ul style="list-style-type: none">• Dane wejściowe: imię, nazwisko, adres email, hasło, numer telefonu• Dane wyjściowe: Rezultat rejestracji
Lista zleceń i ofert	<ul style="list-style-type: none">• Dane wejściowe: Brak• Dane wyjściowe: Lista zleceń utworzonych przez klienta, przyporządkowana lista ofert stworzonych przez serwisy.
Tworzenie zlecenia	<ul style="list-style-type: none">• Dane wejściowe: Dane dotyczące samochodu: marka, model, rok produkcji, rodzaj paliwa, rodzaj skrzyni biegów, czy samochód porusza się samodzielnie Pozostałe dane: adres odbioru, czas odbioru, czy potrzebny samochód zastępczy, rodzaj potrzebnej pomocy / naprawy, dodatkowe informacje.

	<ul style="list-style-type: none"> Dane wyjściowe: Rezultat tworzenia zlecenia
Wyświetlanie i akceptacja oferty	<ul style="list-style-type: none"> Dane wejściowe: (opcjonalnie) identyfikator akceptowanej oferty Dane wyjściowe: Informacje o serwisie: Nazwa, specjalizacja, adres, email, telefon, opis Informacje o ofercie: dane oferty - notatka zawierająca szczegółowy opis proponowanej oferty, proponowana cena (wstępna) za usługę
Widok komunikacji klienta i serwisu (część użytkownika)	<ul style="list-style-type: none"> Dane wejściowe: (opcjonalnie) wysyłana wiadomość Dane wyjściowe: Skrócony opis informacji o zapytaniu (problem, marka i model pojazdu, proponowana cena za usługę) Informacje o serwisie wykonującym usługę: nazwa, specjalizacja, adres, email, telefon, opis, Lista statusów oraz wiadomości w kolejności chronologicznej (chat).

2) Moduł interfejsu serwisu

Tabela 3.4.2 Dane przetwarzane przez poszczególne widoki – moduł interfejsu serwisu

Nazwa widoku	Przetwarzane dane
Widok otwartych zgłoszeń	<ul style="list-style-type: none"> Dane wejściowe: brak Dane wyjściowe: skrócona lista otwartych zgłoszeń zawierająca nazwę problemu, markę, model i rok produkcji pojazdu, preferowana data i miejsce odbioru.
Widok zgłoszenia	<ul style="list-style-type: none"> Dane wejściowe: brak Dane wyjściowe: Dane dotyczące samochodu – wszystkie dane wprowadzone w widoku tworzenia zlecenia (opcjonalnie) dane zaproponowanej oferty – wszystkie dane wprowadzone w widoku tworzenia oferty

Widok tworzenia oferty	<ul style="list-style-type: none"> • Dane wejściowe: Notatka zawierająca szczegółowy opis proponowanej oferty, proponowana cena (wstępna) za usługę • Dane wyjściowe: rezultat tworzenia oferty
Widok listy napraw	<ul style="list-style-type: none"> • Dane wejściowe: brak • Dane wyjściowe: skrótowy opis wszystkich zaakceptowanych ofert (nazwa świadczonej usługi, pojazd, proponowana cena, adres odbioru samochodu)
Widok komunikacji klienta i serwisu (część serwisu)	<ul style="list-style-type: none"> • Dane wejściowe: (opcjonalnie) wysyłana wiadomość, (opcjonalnie) nowa cena za usługę, (opcjonalnie) nowy status zamówienia, • Dane wyjściowe: Skrócony opis informacji o zapytaniu (problem, marka i model pojazdu, proponowana cena za usługę) Informacje o serwisie wykonującym usługę: nazwa, specjalizacja, adres, email, telefon, opis, Lista statusów oraz wiadomości w kolejności chronologicznej (chat).
Widok zmiany danych serwisu	<ul style="list-style-type: none"> • Dane wejściowe: Adres, opis serwisu, nazwa serwisu, specjalizacja, telefon, email • Dane wyjściowe: aktualne dane serwisu

3.5 Specyfikacja listy końcówek API potrzebnych do zapewnienia danych widokom

W Tabeli 3.5.1 przedstawiono listę końcówek API z podziałem na moduły. Wymieniona lista dotyczy elementów zapewniających dane wymienione w punkcie 3.4. dla odpowiadających im widoków w module interfejsu klienta i serwisu. W tabeli pominięto komunikację, która nie odbywa się poprzez REST API.

Tabela 3.5.1 Lista końcówek API przynależących do poszczególnych modułów

Nazwa modułu	Adres REST API - funkcja
Moduł uwierzytelnienia	POST /user/login – logowanie użytkownika
Moduł zarządzania kontem klienta	POST /user/registration – rejestracja użytkownika GET /user/details – pobranie danych o zalogowanym użytkowniku
Moduł realizacji zamówienia i moduł komunikacji	POST /offer – utworzenie szczegółów oferty GET /offer/{leadId} – pobranie ofert wystawionych dla wybranego zlecenia POST /offer/{offerId}/select – wybranie oferty POST /offer/comment/{offerId} – utworzenie wiadomości do oferty GET /offer/events/{offerId} – pobranie zdarzeń związanych z ofertą GET /offer/my – pobranie oferty wystawione przez zalogowany serwis GET /offer/my/active – pobranie oferty przez serwis, które są w trakcie realizacji POST /offer/price/{offerId} – zmiana ceny oferty przez serwis POST /offer/status/{offerId} – zmiana statusu oferty przez serwis
Moduł zapytania	GET /service – pobranie predefiniowanych usług i potencjalnych problemów GET /carProperty – pobranie predefiniowanych informacji o samochodach POST /lead – utworzenie nowego zlecenia przez klienta GET /lead/my – pobranie wszystkich zleceń stworzonych przez zalogowanego klienta GET /lead/open – pobranie wszystkich oczekujących zleceń przez serwis

Moduł zarządzania serwisem	GET /user/details/service/{serviceId} - pobranie danych o serwisie PUT /user/details/service – aktualizacja danych przez zalogowany serwis
----------------------------	---

3.6 Wymagania niefunkcjonalne

Wymagania wydajności:

- Całkowity czas ładowania się strony powinien być krótszy niż 5 sekund,
- System powinien obsłużyć do 200 użytkowników jednocześnie bez utraty płynności,
- System gotowy do płynnego działania przez 5 lat.

Wymagania niezawodności:

- System powinien być możliwy do odtworzenia po poważnej awarii w przeciągu 24h,
- System gotowy do uruchomienia w trybie 24/7/365,
- Dostępność w skali roku powinna wynosić co najmniej 95%.

Wymagania użyteczności aplikacji:

- Czas szkolenia z obsługi systemu dla serwisu i użytkownika do 4h,
- Strona internetowa musi być sprawna na komputerach osobistych, tabletach oraz urządzeniach mobilnych z dostępem do Internetu.

Wymagania dotyczące standardów i licencji:

- System powinien być przygotowany do komercyjnego użycia z oprogramowaniem nie wymagającym zakupu dodatkowej licencji,
- Produkt powinien być wolny od błędów bezpieczeństwa umieszczonych w projekcie OWASP TOP 10,
- Kod źródłowy HTML, JavaScript / TypeScript [1] i CSS powinien być wolny od błędów i poprawny semantycznie.

3.7 Analiza ryzyka

W celu zwiększenia skuteczności prawidłowego przebiegu realizacji prac wstępnie przeanalizowano ryzyko.

Dla każdego przewidzianego ryzyka zostało oszacowane prawdopodobieństwo wystąpienia problemu w trzystopniowej skali:

- Niskie – sytuacja możliwa jednak nie zależy w istotny sposób od czynników zewnętrznych, a bazując na doświadczeniu w programowaniu, prawdopodobnie nie wydarzy się,
- Średnie – sytuacja zależy od czynników zewnętrznych (sytuacji losowych) lub sytuacja dotyczy niespodziewanych wydarzeń związanych z samą realizacją projektu,
- Wysokie – sytuacja ze względu na posiadane informacje, została wstępnie sklasyfikowana jako prawdopodobna.

Oszacowano również przewidywany wpływ na projekt. Wynik umieszczono w kolumnie „efekt” i podano w trzystopniowej skali:

- Niski (nie wykorzystano) – sytuacja nie zaburzy prawidłowej realizacji projektu. Nie powinna wpłynąć w istotny sposób na czas realizacji projektu.
- Średni – sytuacja może wpłynąć na jakość dostarczonego kodu. Może wpłynąć na wymagania нефункционалне.
- Wysoki – sytuacja może istotnie wpłynąć na realizację podstawowych wymagań funkcjonalnych.

W celu redukcji ryzyka związanego ze zidentyfikowanymi sytuacjami, zaproponowano rozwiązania mające na celu ograniczenie prawdopodobieństwa wystąpienia sytuacji lub zmniejszenia efektu poprzez wcześniejsze wykrycie realizującego się problemu. Metody redukcji ryzyka opisano w Tabeli 3.7.1.

Tabela 3.7.1 Szczegółowe elementy analizy ryzyka

Problem	Prawdopodobieństwo	Efekt	Metoda redukcji ryzyka
Rozbudowanie systemu uniemożliwiające uruchomienie go na pojedynczej maszynie podczas projektowania i eksploatacji	Niskie	Średni	Bieżące optymalizowanie systemu podczas implementacji

Choroba jednego z członków zespołu i zaburzenie postępu prac	Średnie	Wysoki	Komunikacja wzajemnej dyspozycyjności członków zespołu i przewidywanie prac.
Praca w nowym zespole - problemy z organizacją i synchronizacją pracy	Średnie	Średni	Regularne spotkania
Napięty harmonogram prac - zakończenie prac nie może zostać przesunięte w czasie	Wysokie	Wysoki	Regularne spotkania podsumowujące wykonane prace oraz planowanie prac z wyprzedzeniem
Brak precyzyjnie zdefiniowanych kroków biznesowych wymaganych ze względu prawnych i formalnych - możliwość powstania skomplikowanego procesu w celu pełnej realizacji funkcji systemu	Średnie	Średni	Podejście ogólnikowe do problematycznego zagadnienia zostawiające przestrzeń na doprecyzowanie
Brak doświadczenia wszystkich członków w korzystaniu z narzędzi deweloperskich – możliwe problemy organizacyjne natury technicznej.	Wysokie	Średni	Spotkania w celu nauki wykorzystywanych technologii.

Analizując zidentyfikowane ryzyka, największe zagrożenie stwarza napięty harmonogram prac. Brak możliwości przesunięcia terminu zakończenia prac powoduje, że niezrealizowane wymagania funkcjonalne mogą spowodować bezużyteczność produktu – nie będzie on realizował podstawowych funkcji biznesowych. Dlatego konieczne będzie zastosowanie i przestrzeganie regularnych spotkań podsumowujących wykonane prace oraz planowanie prac z wyprzedzeniem. Również istotnym czynnikiem jest choroba jednego z członków zespołu – zespół składa się z niewielkiej liczby osób, które różnią się pod względem znajomości technologii, doświadczenia i przewidywanego zakresu prac w projekcie. Dłuższa niedostępność jednego z członków może spowodować zablokowanie prac pozostałym osobom. W celu

redukcji ryzyka zaproponowano wzajemną komunikację o dyspozycyjności i przewidywanie prac tak by uniknąć dłuższych przestojów w projekcie. Pozostałe metody redukcji ryzyka dla problemów o mniejszej istotności wskazano w tabeli.

4 Projekt i implementacja

4.1 Analiza domeny

4.1.1 Uprawnienia wykorzystywane w systemie:

- a) Użytkownik niezalogowany - może wykonać proces samodzielnej rejestracji konta klienta w systemie, nie ma dostępu do pozostałych modułów,
- b) Klient - użytkownik zalogowany w panelu klienta, zleceniodawca usług w systemie
- c) Serwis - użytkownik zalogowany w panelu serwisu, zleceniobiorca usług w systemie

4.1.2 Kategorie zleceń i dodatkowe opcje dostępne w systemie

- a) Usługa – konkretna usługa, taka jak okresowa diagnostyka / wymiany części (opon, oleju, filtrów),
- b) Problem – naprawy, w których klient może nie znać przyczyny występowania problemu. Np. nierówna praca silnika, świecące się kontrolki ostrzegawcze, problemy ze ściągnięciem pojazdu na jedną stronę.

4.1.3 Dodatkowe opcje systemu, których Klient może oczekiwać od Serwisu w zakresie obsługi zlecenia

- a) Opcja samochód zastępczy - samochód wynajęty krótkoterminowo na czas naprawy
- b) Opcja samochód unieruchomiony – w tym przypadku serwis oprócz właściwej naprawy uwzględni to, że samochód nie jest w stanie samodzielnie się poruszać. Wymagane może być holowanie/laweta lub naprawa na miejscu.

4.1.4 Predefiniowane usługi dostępne w aplikacji:

- Zmiana opon
- Wymiana oleju
- Przegląd
- Wymiana żarówek/światła
- Sprawdzenie poziomu oleju w skrzyni biegów
- Wymiana świec zapłonowych
- Wymiana filtra powietrza
- Wymiana filtra kabinowego
- Kalibracja zbieżności kół
- Serwis oświetlenia

- Wymiana płynu do spryskiwaczy

4.1.5 Predefiniowane problemy z samochodem dostępne w aplikacji:

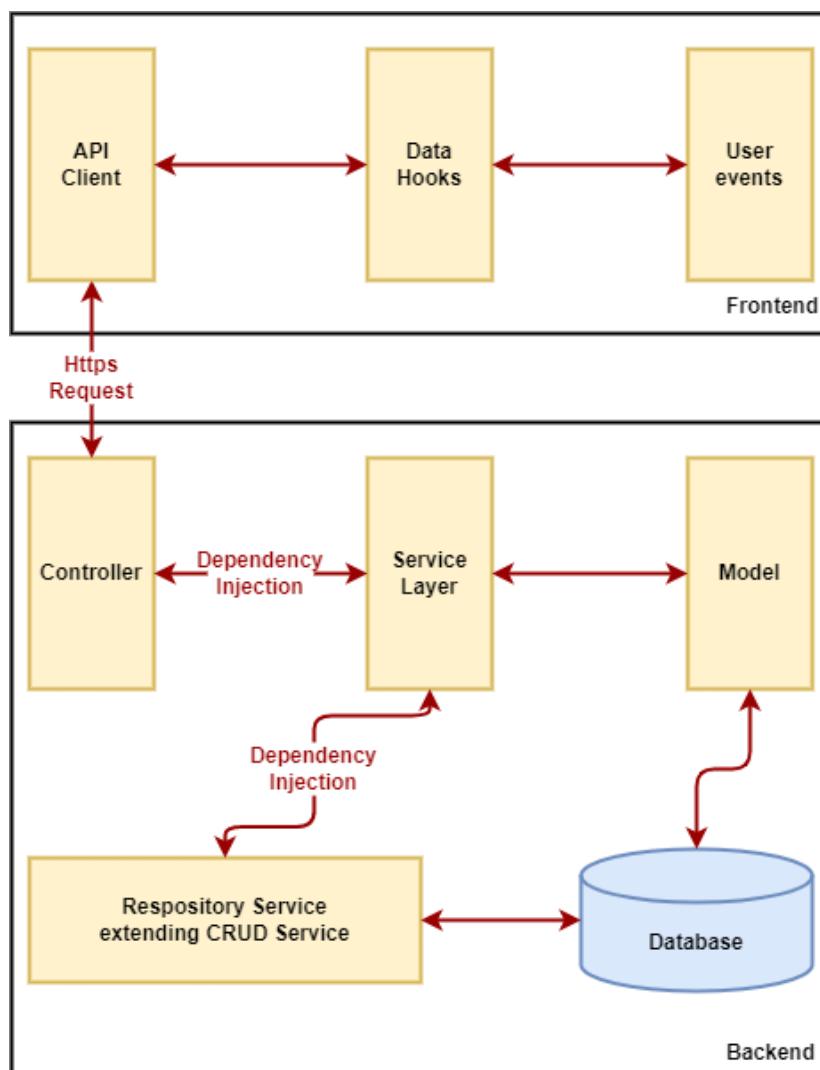
- Problem z kontrolkami
- Problem z zapłonem
- Nierówna praca silnika
- Samochód ściąga na jedną stronę
- Potrzebna diagnostyka elektroniczna
- Problem z amortyzatorami
- Problem ze spalinami

4.2 Trójwarstwowa architektura systemu

System będzie działał w modelu trójwarstwowym wykorzystującym podział na:

- a) **Warstwę kliencką (frontend)** - warstwę prezentacji działającą na przeglądarkach internetowych na urządzenia desktopowych i mobilnych,
- b) **Warstwę serwerową (backend)** - warstwa logiki biznesowej umieszczona na serwerze właściciela aplikacji obsługująca użytkowników (klientów i serwisy) przez sieć Internet,
- c) **Baza danych (database)** - warstwa danych gromadząca i udostępniająca informacje.

Na Rysunek 4.2.1 przedstawiona została ogólna architektura systemu z uwzględnieniem istotnych wewnętrznych modułów.

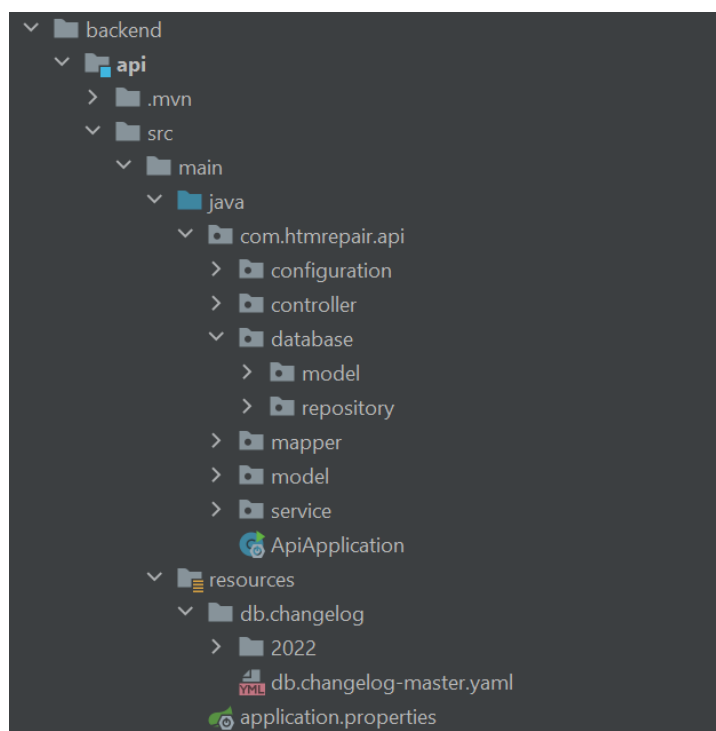


Rysunek 4.2.1 Architektura systemu

4.3 Stos technologiczny i struktura kodu

Warstwa logiki biznesowej została zaimplementowana przy użyciu języka Java [2] oraz frameworka Spring [3]. Do formatowania kodu została wykorzystana biblioteka Lombok [4], która pozwala na pisanie krótszego kodu w bardziej czytelny sposób. Autoryzacja użytkownika przebiega na podstawie tokenu JWT [5] i została zrealizowana poprzez bibliotekę Spring Security.

Rysunek 4.3.1 pokazuje ogólną strukturę kodu źródłowego części serwerowej.



Rysunek 4.3.1 Ogólna struktura kodu źródłowego części serwerowej

Backend został podzielony na następujące katalogi:

- **configuration** – katalog zawierający konfiguracje związane z wykorzystaną biblioteką Spring - na przykład konfiguracja Spring Security [6] – oraz konfiguracje dokumentacji Swagger.
- **controller** – zawiera RESTowe kontrolery endpointów.
- **database** – (data layer), ten katalog zawiera:
 - model – zawiera encje JPA
 - repository – zawiera interfejsy repozytorium dla poszczególnych encji
- **mapper** – zawiera komponenty, które mapują obiekty z bazy danych do DTO.
- **service** – (business layer) zawiera interfejsy oraz ich implementacje z logiką biznesową aplikacji.
- **resources** – katalog przeznaczony do zasobów statycznych, w naszym przypadku to są skrypty do systemu migracji bazy danych oraz właściwości aplikacji.

Warstwa prezentacji została wykonana przy pomocy technologii HTML, CSS i Javascript (Typescript) oraz biblioteki React [7]. Do zarządzania stanem danych pobranych z serwera wykorzystano bibliotekę React Query [8]. Do formularzy wykorzystano bibliotekę Formik [9].

Do implementacji elementów interfejsu użytkownika wykorzystano bibliotekę komponentów Material UI [10]. Do zarządzania nawigacją bibliotekę React Router [11].

Język TypeScript został wybrany z racji na swoją możliwość silnego typowania. Jest to język będący nadzbiorem języka JavaScript, na który ten następnie jest transpilowany. Każda poprawna konstrukcja języka JavaScript jest poprawną konstrukcją języka TypeScript, stąd w niniejszej pracy te pojęcia stosujemy wspólnie.



Rysunek 4.3.2 Ogólna struktura kodu źródłowego warstwy prezentacji

Frontend został podzielony na kilka katalogów zgodnie z Rysunek 4.3.2:

- **public** – katalog, który zawiera domyślny plik serwowany przez przeglądarkę – index.html oraz pliki inne pliki potrzebne przeglądarce i wyszukiwarkom takie jak favicon.ico, robots.txt.
- **api** – katalog reprezentujący warstwę integracji z częścią serwerową
 - **hooks** – zawiera reużywalne hooki do wykorzystania w komponentach biblioteki React.
 - **utils** – zawiera funkcje pomocnicze API, takie jak pobieranie tokena .autentykacji z pamięci lokalnej przeglądarki.
- **components** – są tu komponenty, z których skomponowany jest interfejs części frontendowej. Komponenty podzielone są w podkatalogach:

- **client** – który zawiera komponenty wykorzystywane przez część klienta,
 - **service** – który zawiera komponenty wykorzystywane przez część serwisu,
 - **shared** – który zawiera komponenty współdzielone między dowolnymi dwoma aktorami.
- **contexts** – zawiera konteksty biblioteki React, które umożliwiają kaskadowe propagowanie danych w dół drzewa komponentów. Przykładowym kontekstem z tego katalogu jest `UserContext`, który dostarcza komponentom-dzieciom informację o aktualnie zalogowanym użytkowniku.
 - **generated** – zawiera pliki wygenerowane. Wykorzystaliśmy generator klienta API `swagger-typescript-api` na podstawie DTO upublicznionych przez backend, który zawiera typy oraz dedykowane funkcje wywołujące zapytania dla wszystkich upublicznionych końcówek API.
 - **styles** – zawiera pliki CSS odpowiadające za styl aplikacji.

4.4 Szczegóły architektury trójwarstwowej

4.4.1 Warstwa prezentacji

Na poniższych rysunkach przedstawiono widoki utworzone przy pomocy biblioteki React. Widok to komponent biblioteki React, który pełni funkcję oddzielnej strony. Część modułu interfejsu użytkownika została przedstawiona na rysunkach od Rysunek 4.4.1. do Rysunek 4.4.10. Przedstawiają one funkcje z których może korzystać użytkownik i odpowiadają poszczególnym wymaganiom funkcjonalnym umieszczonym w punkcie 3.2. Obok nazw technicznych widoków (bazujących na kodzie źródłowym) umieszczono polskie nazwy odpowiadające nazwom z Tabela 3.4.1 i Tabela 3.4.2.

- a) **CreateLeadView**: widok tworzenia zlecenia - odpowiada za funkcję tworzenia zlecenia. Ze względów czytelności i estetyki został on podzielony na 3 ekrany. Na pierwszym (Rysunek 4.4.1) użytkownik wprowadza miasto, na drugim (Rysunek 4.4.2) dane dotyczące samochodu, na trzecim (Rysunek 4.4.3) dane dotyczące czasu i miejsca odbioru samochodu. Dodatkowym krokiem jest rejestracja (Rysunek 4.4.4) lub logowanie (Rysunek 4.4.5) w przypadku, gdy użytkownik nie jest jeszcze zalogowany.
- b) **LoginOrRegisterView**: widok logowania i rejestrowanie użytkownika - (Rysunek 4.4.4 oraz Rysunek 4.4.5) odpowiada za funkcję rejestracji i funkcję logowania.

- c) **ShowMyListLeadsView**: widok lista zleceń i ofert (Rysunek 4.4.6, Rysunek 4.4.7, Rysunek 4.4.8) odpowiada za funkcję wyświetlania listy ofert przekazanych przez serwis, dodatkowo odpowiada za nie wskazaną w wymaganiach funkcjonalnych funkcję wyświetlania listy wprowadzonych zleceń oraz statusów zleceń.
- d) **ShowOfferView**: widok wyświetlanie i akceptacja oferty (Rysunek 4.4.9) odpowiada za funkcję wyboru oferty przez użytkownika.
- e) **ClientActivityLogView**: widok komunikacji klienta i serwisu (część kliencka) - (Rysunek 4.4.10) odpowiada za funkcję komunikacji klienta i serwisu.

Część modułu interfejsu serwisu została przedstawiona na rysunkach od Rysunek 4.4.11 do Rysunek 4.4.16. Funkcjonalność logowania dla obu interfejsów jest wspólna. Przedstawione widoki odpowiadają następującym wymaganiom funkcjonalnym:

- a) **ShowServiceListLeadsView** (Rysunek 4.4.11), **ShowServiceLeadView** (Rysunek 4.4.14), **CreateOfferView** (Rysunek 4.4.13): widok otwartych zgłoszeń, widok zgłoszenia, widok tworzenia oferty - odpowiadają za funkcję wyświetlania listy otwartych zleceń i dodawania oferty przez serwis.
- b) **ServiceMyOffersView**: widok listy napraw (Rysunek 4.4.12) odpowiada za funkcję wyświetlania listy aktualnie prowadzonych napraw przez serwis. Funkcja nie została zidentyfikowana w części koncepcyjnej jednak była konieczna do prawidłowego funkcjonowania modułu interfejsu serwisu.
- c) **ServiceActivityLogView**: widok komunikacji klienta i serwisu (część serwisu) (Rysunek 4.4.15) odpowiada za funkcję komunikacji klienta i serwisu, funkcję przekazania samochodu do serwisu i rozpoczęcie wykonywania usługi, funkcję zmiany statusu zamówienia związanego z przebiegiem wykonywania usługi.
- d) **ServiceUpdateDetails**: widok zmiany danych serwisu (Rysunek 4.4.16) odpowiada za funkcję obsługi konta serwisu w systemie.

[Znajdź serwis](#)
[Moje zlecenia](#)
[Nowe zlecenie](#)

KLIENT

Witaj, klient1

WYLOGUJ

Wybierz swoje miasto

KRAKÓW	WARSZAWA	GDAŃSK
--------	----------	--------

Rysunek 4.4.1 Widok pierwszej strony tworzenia zlecenia - CreateLeadPage

[Znajdź serwis](#)
[Moje zlecenia](#)
[Nowe zlecenie](#)

KLIENT

Witaj, klient1

WYLOGUJ

Co potrzebujesz zrobić?

MAM PROBLEM

WIEM JAKIEJ USŁUGI POTRZEBUJĘ

Z czym potrzebujesz pomocy?

Problem z kontrolkami

Problem z zapłonem

Nierówna praca silnika

Samochód ściąga na jedną stronę

Potrzebna diagnostyka elektroniczna

Problem z amortyzatorami

Problem ze spalinami

Wybierz markę i model samochodu

Marka

Alfa Romeo

Model

159

Wybierz rodzaj silnika

Rodzaj silnika

Benzyna

Wybierz rodzaj skrzyni biegów

Rodzaj skrzyni biegów

Manualna

Wybierz rok produkcji

Rok produkcji

2005

Dodaj szczegóły. Czy masz jakieś spostrzeżenia?

Samochód nie odpala.

← WSTECZ

DALEJ →

Rysunek 4.4.2 Widok drugiej strony tworzenia zlecenia - CreateLeadPage

[Znajdź serwis](#)

[Moje zlecenia](#)
[Nowe zlecenie](#)

KLIENT

Witaj, klient1

WYLOGUJ

Podaj adres pod którym serwis będzie mógł odebrać samochód:

Miasto

Ulica, numer, kod pocztowy

Kraków

ul. Reymona 1

Czas obioru samochodu

Data odbioru

28-08-2022

RANO

WIECZÓR

CAŁY DZIEŃ

Czy potrzebujesz samochodu zastępczego?

TAK

NIE

Czy samochód porusza się samodzielnie?

TAK

NIE

← WSTECZ

WYŚLIJ

Rysunek 4.4.3 Widok trzeciej strony tworzenia zlecenia - CreateLeadPage

[Znajdź serwis](#)

[Moje zlecenia](#)
[Nowe zlecenie](#)

ZALOGUJ

Zarejestruj się

Imię *

Grażyna

Nazwisko *

Kowalska

Adres Email *

grazkaa11@buziaczek.pl

Hasło *

.....

Numer telefonu *

541201214

ZAREJESTRUJ SIĘ

[Masz już konto? Zaloguj się](#)

← WSTECZ

WYŚLIJ

Rysunek 4.4.4 Widok ekranu rejestracji

Znajdź serwis

Moje zlecenia
Nowe zlecenie

ZALOGUJ

Zaloguj się

Adres Email *

cruztom1977@proton.me

Hasło *

.....

ZALOGUJ

Nie masz jeszcze konta? Zarejestruj się

← WSTECZ

WYŚLIJ

Rysunek 4.4.5 Widok ekranu logowania

Znajdź serwis

Moje zlecenia
Nowe zlecenie

KLIENT
Witaj, klient1
WYLOGUJ

Moje zlecenia

	Powód zgłoszenia	Marka	Model	Preferowana data odbioru	Status	
▼	Problem z zapłonem	Aston Martin	567	2022-08-25	Trwa praca	ZOBACZ
▼	Samochód ściąga na jedną stronę	Aixam	A721	2022-08-25	Trwa praca	ZOBACZ
▼	Samochód ściąga na jedną stronę	Acura	RDX	2022-08-25	Poszukiwanie serwisu	
▼	Zmiana opon	Audi	S5	2022-08-25	Poszukiwanie serwisu	
▼	Problem z kontrolkami	Aixam	A721	2022-08-25	Zamknięte	ZOBACZ
▼	Problem z zapłonem	Alfa Romeo	159	2022-09-15	Poszukiwanie serwisu	

Rysunek 4.4.6 Widok listy zleceń - klient

^
Samochód ściąga na jedną stronę
Acura
asdasd
2022-08-25
Poszukiwanie serwisu

Szczegóły

Identyfikator: 8
Adres: Kraków, dgdfdfg
Typ zgłoszenia: Naprawa
Powód zgłoszenia: Samochód ściąga na jedną stronę
Marka: Acura
Model: asdasd
Rok produkcji: 1996
Rodzaj paliwa: Benzyna+LPG
Rodzaj skrzyni biegów: Manualna
Samochód unieruchomiony: Nie
Samochód zastępcze: Tak
Preferowana data odbioru samochodu: 2022-08-25 - rano

Zakład	Lokalizacja	Proponowana cena	Więcej informacji
Serwis Roman złota rączka złoty klucz	Kraków, ul. św. Bonifacego 32	1231	ZOBACZ

Rysunek 4.4.7 Widok szczegółów zlecenia - rozwinięta opcja w menu

^
Zmiana opon
Audi
dfgdffd
2022-08-25
Poszukiwanie serwisu

Szczegóły

Identyfikator: 9
Adres: Warszawa,
Typ zgłoszenia: Usługa
Powód zgłoszenia: Zmiana opon
Marka: Audi
Model: dfgdffd
Rok produkcji: 2000
Rodzaj paliwa: Diesel
Rodzaj skrzyni biegów: Manualna
Samochód unieruchomiony: Nie
Samochód zastępcze: Tak
Preferowana data odbioru samochodu: 2022-08-25 - rano

Zakład	Lokalizacja	Proponowana cena	Więcej informacji
Nie ma jeszcze żadnych ofert			

Rysunek 4.4.8 Widok szczegółów zlecenia - brak ofert

[Znajdź serwis](#)
[Moje zlecenia](#)
[Nowe zlecenie](#)


KLIENT
Witaj, klient1
WYLOGUJ

Podgląd oferty

Nazwa serwisu	Serwis Roman złota rączka złoty klucz, Mechanika samochodów
Adres	Kraków, ul. św. Bonifacego 32
Kontakt	520125412, zlotyroman@gmail.com
Opis serwisu	Naprawimy każdy problem (z samochodem)
Szczegóły zamówienia	asdasd
Przewidywana cena	1231

[POWRÓT](#)
[AKCEPTUJ](#)

Rysunek 4.4.9 Widok szczegółów oferty - możliwość akceptacji

 [Znajdź serwis](#) [Moje zlecenia](#) [Nowe zlecenie](#)

KLIENT

 Witaj, a@b.com

WYLOGUJ

[POWRÓT](#)


Usługa / Problem	Pojazd	Proponowana cena	Adres
Problem z kontrolkami	Austin T3, 1992, Benzyna	1999	Kraków, Krakowska 2, 31-002

Pokaż szczegóły zgłoszenia


Pokaż informacje o Serwisie

Zdarzenia i czat

24-08-2022 06:55:49

Oferta została zaakceptowana 

Serwis wkrótce rozpocznie wykonywanie usługi

 Pracownik podjedzie po samochód pod zadany adres około godziny 10
24-08-2022 07:21:37


Ok, czekam
24-08-2022 07:19:47

Do

Dziękuję
24-08-2022 07:22:25

Do

24-08-2022 07:23:22

Pojazd został odebrany 

Serwisant odebrał pojazd. W każdej chwili możesz zapytać o szczegóły wykonywanej usługi


Proszę obchodzić się ostrożnie z moim samochodem
24-08-2022 09:10:15

Do

Kiedy koniec?
24-08-2022 10:19:26

Do

Napisz do serwisu

WIADOMOŚĆ 

Rysunek 4.4.10 Widok komunikacji klienta i serwisu

[Znajdź serwis](#)
[Naprawy](#)
[Otwarte zlecenia](#)
[Konto](#)

SERWIS Witaj, a [WYLOGUJ](#)

Lista otwartych zgłoszeń

Status	Powód zgłoszenia	Marka i model	Preferowana data odbioru	Miejsce odbioru	
Open Wysłano ofertę	Samochód ściąga na jedną stronę	Skoda Fabia 2005	2022-08-25	Kraków, ul Rydla 17	ZOBACZ
Open	Zmiana opon	Golf 3 2021	2022-08-25	Kraków, ul. Borkowska 1	ZOBACZ
Open	Problem z zapłonem	Alfa Romeo 159 2005	2022-09-15	Kraków, Kawiori 1	ZOBACZ

Rows per page: 5
 1-3 of 3
 |< < > >|

Rysunek 4.4.11 Widok listy oczekujących ofert - serwis

[Znajdź serwis](#)
[Naprawy](#)
[Otwarte zlecenia](#)
[Konto](#)

SERWIS Witaj, a [WYLOGUJ](#)

Moje naprawy

Usługa / Problem	Status	Pojazd	Proponowana cena	Adres	
Problem z kontrolkami	Usługa zakończona	Aixam 567, 2000, Benzyna	1200	Warszawa, ul Gertrudy 51	ZOBACZ
Problem z kontrolkami		Aixam 567, 2000, Benzyna	1423	Warszawa, ul. Kazimierza 132	ZOBACZ
Problem z kontrolkami		Acura asd, 2000, Benzyna	500	Kraków, ul. Władysława 11	ZOBACZ

Rysunek 4.4.12 Widok listy aktywnych napraw – serwis

[Znajdź serwis](#)
[Naprawy](#)
[Otwarte zlecenia](#)
[Konto](#)

SERWIS Witaj, a [WYLOGUJ](#)


Create offer

Co możesz zaoferować klientowi? *

Proponowana cena *

[POWRÓT](#)
[WYŚLIJ](#)

Rysunek 4.4.13 Widok tworzenia oferty – serwis

 [Znajdź serwis](#)

[Naprawy](#)

[Otwarte zlecenia](#)

[Konto](#)

SERWIS

Witaj, a

WYLOGUJ

POWRÓT

Podgląd zgłoszenia

Potrzebna usługa	Samochód ściąga na jedną stronę
Szczegóły	sdfsfd
Marka	Acura
Model	asdasd
Rok produkcji	1996
Paliwo	Benzyna+LPG
Skrzynia biegów	Manualna
Pojazd porusza się samodzielnie	Tak
Adres odbioru	Kraków, dgfdfg
Preferowana data odbioru	2022-08-25
Preferowany czas odbioru	Rano
Samochód zastępczy	Tak

Złożona oferta

Tekst oferty

asdasd

Zaproponowana cena

1231

Status

Open

Rysunek 4.4.14 Widok zlecenia ze złożoną ofertą

[Znajdź serwis](#)
[Naprawy](#)
[Otwarte zlecenia](#)
[Konto](#)

SERWIS

Witaj, ab@b.com

WYLOGUJ

POWRÓT

Usługa / Problem	Pojazd	Proponowana cena	Adres
Kalibracja zbieżności kół	Alfa Romeo T5, 2010, Benzyna+LPG	650	Gdańsk,

Pokaż szczegóły zgłoszenia

Pokaż informacje o Kliencie

Zdarzenia i czat

24-08-2022 12:05:36

Oferta została zaakceptowana

Serwis wkrótce rozpocznie wykonywanie usługi

Chcielibyśmy się umówić na spotkanie

24-08-2022 12:07:23

Proszę przyjechać o godzinie 11

28-08-2022 07:14:02

Ok

24-08-2022 12:14:51

24-08-2022 12:15:31

Pojazd został odebrany

Serwisant odebrał pojazd. W każdej chwili możesz zapytać o szczegóły wykonywanej usługi

27-08-2022 01:01:51

Serwis rozpoczął wykonywanie usługi

Otrzymasz informacje o postępach w realizacji usługi

Naprawiamy koła

27-08-2022 01:26:32

Zmień status usługi:

POJAZD ODEBRANY

USŁUGA ROZPOCZĘTA

USŁUGA ZAKOŃCZONA

POJAZD ZWRÓCONY

Zmień cenę usługi:

650


ZMIEN

Napisz do klienta

NAPRAWA

WIADOMOŚĆ

Rysunek 4.4.15 Widok komunikacji klienta i serwisu - część serwisu

 **Znajdź serwis**

Naprawy

Otwarte zlecenia

Konto

SERWIS

Witaj, a

WYLOGUJ

Zmień swoje dane

Adres *

Kraków, ul. św. Bonifacego 32

O nas - opis serwisu *

Naprawimy każdy problem (z samochodem)

Email *

zlotyroman@gmail.com

Nazwa *

Serwis Roman - złota rączka, złoty klucz

Telefon kontaktowy *

520125412

Specjalizacja serwisu *

Mechanika samochodów

POWRÓT

ZAPISZ

Rysunek 4.4.16 Widok edycji danych serwisu

Każdy widok modułu interfejsu użytkownika został również przystosowany do działania na urządzeniach mobilnych. Na poniższych zrzutach ekranu (Rysunek 4.4.17 i Rysunek 4.4.18) przedstawiono wybrane elementy demonstrujące działanie aplikacji na urządzeniach mobilnych.

Znajdź serwis

Moje zlecenia
Nowe zlecenie
KLIENT

Witaj, klient1
WYLOGUJ

POWRÓT

Usługa / Problem	Pojazd	Proponowana cena	Adres
Problem z kontrolkami	Aixam 567, 2000, Benzyna	6767567	Warszawa

Pokaż szczegóły zgłoszenia

Pokaż informacje o Serwisie

Zdarzenia i czat

25-08-2022 11:14:06

Pojazd został odebrany

Serwisant odebrał pojazd. W każdej chwili możesz zapytać o szczegóły wykonywanej usługi

qweqwe
25-08-2022 11:14:08

Napisz do serwisu

Rysunek 4.4.17 Widok komunikacji klienta i serwisu na telefonie

Znajdź serwis

Moje zlecenia
Nowe zlecenie
KLIENT

Witaj, klient1
WYLOGUJ

Moje zlecenia

Powód zgłoszenia	Marka	Model
Problem z zapłonem	Aston Martin	567
Samochód ściąga na jedną stronę	Aixam	Minicar
Problem z kontrolkami	Aixam	Couple
Problem z kontrolkami	Acura	Test
Problem z kontrolkami	Acura	Test
Potrzebna diagnostyka elektroniczna	Aixam	Minicar
Samochód ściąga na jedną stronę	Acura	Test

Rysunek 4.4.18 Widok listy zleceń – klient

4.4.2 Warstwa logiki biznesowej

Warstwa logiki biznesowej została wykonana w części backend. W celu komunikacji z warstwą prezentacji wykorzystano REST API. Poniżej w Tabeli 4.4.1 Endpoint logowania do Tabeli 4.4.17 zostały przedstawione wszystkie endpointy, które wykorzystuje aplikacja wraz ze szczegółami dotyczącymi parametrów wejściowych (*parametr query* i *parametr body*) i wyjściowych (*response*) oraz autoryzacji. Różnymi kolorami oznaczono metody w zależności od typu (GET, POST, PUT).

Tabela 4.4.1 Endpoint logowania

POST /user/login	Autoryzacja
	<brak>
Parametry query	Parametry body
<brak>	<pre>{ "email": "string", "password": "string" }</pre>
Response	
<pre>{ "details": { "email": "string", "firstName": "string", "id": 0, "lastName": "string", "phoneNumber": "string", "type": "Service" }, "token": "string" }</pre>	

Tabela 4.4.2 Endpoint rejestracji

POST /user/registration	Autoryzacja
	<brak>
Parametry query	Parametry body
<brak>	<pre>{ "email": "string", "firstName": "string", "lastName": "string", "password": "string", "phoneNumber": "string", "type": "Service" }</pre>
Response	
<pre>{ "details": { "email": "string", "firstName": "string", "id": 0, "lastName": "string", "phoneNumber": "string", "type": "Service" }, "token": "string" }</pre>	

Tabela 4.4.3 Endpoint pobierania danych użytkownika

GET /user/details	Autoryzacja
	User/Service
Parametry query	Parametry body
<brak>	<brak>
Response	
<pre>{ "email": "string", "firstName": "string", "id": 0, "lastName": "string", "phoneNumber": "string", "type": "Service" }</pre>	

Tabela 4.4.4 Endpoint zmiany danych serwisu

PUT /user/details/service	Autoryzacja
	Service
Parametry query	Parametry body
<brak>	<pre>{ "address": "string", "description": "string", "email": "string", "name": "string", "phoneNumber": "string", "specialization": "string" }</pre>
Response	
<pre>{ "address": "string", "description": "string", "email": "string", "name": "string", "phoneNumber": "string", "specialization": "string" }</pre>	

Tabela 4.4.5 Endpoint pobierania danych serwisu

GET /user/details/service/{serviceId}	Autoryzacja
	User/Service
Parametry query	Parametry body
<brak>	<brak>
Response	
<pre>{ "address": "string", "description": "string", "email": "string", "name": "string", "phoneNumber": "string", "specialization": "string" }</pre>	

Tabela 4.4.6 Endpoint pobierania predefiniowanych usług

GET /service	Autoryzacja
	<brak>
Parametry query	Parametry body
<brak>	<brak>
Response	
<pre>[{ "id": 0, "name": "string", "type": "Problem" }]</pre>	

Tabela 4.4.7 Endpoint dodania zlecenia

POST /lead	Autoryzacja
	User
Parametry query	Parametry body
<brak>	<pre>{ "address": "string", "brandId": 0, "canMove": true, "description": "string", "fuelId": 0, "gearboxId": 0, "model": "string", "preferredDate": "2022-07-07", }</pre>

	<pre>"preferredDayPart": "FirstHalf", "replacementNeeded": true, "serviceId": 0, "year": 0 }</pre>
Response	
<pre>{ "address": "string", "brand": { "id": 0, "name": "string" }, "canMove": true, "createdUserId": 0, "description": "string", "fuel": { "id": 0, "name": "string" }, "gearbox": { "id": 0, "name": "string" }, "id": 0, "model": "string", "preferredDate": "string", "preferredDayPart": "FirstHalf", "replacementNeeded": true, "service": { "id": 0, "name": "string", "type": "Problem" }, "status": "Canceled", "year": 0 }</pre>	

Tabela 4.4.8 Endpoint pobierania usług klienta

GET /lead/my	Autoryzacja
	User
Parametry query	Parametry body
<brak>	<brak>
Response	
<pre>{ "address": "string", "brand": { "id": 0, "name": "string" }, "canMove": true, "createdUserId": 0, "description": "string", "fuel": { "id": 0, "name": "string" }, "gearbox": { "id": 0, "name": "string" }, "id": 0, "model": "string", "preferredDate": "string", "preferredDayPart": "FirstHalf", "replacementNeeded": true, "service": { "id": 0, "name": "string", "type": "Problem" }, "status": "Canceled", "year": 0 }</pre>	

```

"canMove": true,
"createdUserId": 0,
"description": "string",
"fuel": {
  "id": 0,
  "name": "string"
},
"gearbox": {
  "id": 0,
  "name": "string"
},
"id": 0,
"model": "string",
"preferredDate": "string",
"preferredDayPart": "FirstHalf",
"replacementNeeded": true,
"service": {
  "id": 0,
  "name": "string",
  "type": "Problem"
},
"status": "Canceled",
"year": 0
}

```

Tabela 4.4.9 Endpoint pobierania otwartych usług

GET /lead/open	Autoryzacja
	Service
Parametry query	Parametry body
<brak>	<brak>
Response	
<pre> { "address": "string", "brand": { "id": 0, "name": "string" }, "canMove": true, "createdUserId": 0, "description": "string", "fuel": { "id": 0, "name": "string" }, "gearbox": { "id": 0, "name": "string" }, "id": 0, "model": "string", "preferredDate": "string", </pre>	

```

"preferredDayPart": "FirstHalf",
"replacementNeeded": true,
"service": {
  "id": 0,
  "name": "string",
  "type": "Problem"
},
"status": "Canceled",
"year": 0
}

```

Tabela 4.4.10 Endpoint pobierania predefiniowanych właściwości samochodów

GET /carProperty	Autoryzacja
	<brak>
Parametry query	Parametry body
<brak>	<brak>
Response	
<pre> { "brand": [{ "id": 0, "name": "string" }], "fuel": [{ "id": 0, "name": "string" }], "gearbox": [{ "id": 0, "name": "string" }] } </pre>	

Tabela 4.4.11 Endpoint tworzenia oferty

POST /offer	Autoryzacja
	Service
Parametry query	Parametry body
<brak>	<pre>{ "leadId": 0, "notes": "string", "price": 0 }</pre>
Response	
<pre>{ "createDate": "string", "createdUserId": 0, "id": 0, "lead": { "address": "string", "brand": { "id": 0, "name": "string" }, "canMove": true, "createdUserId": 0, "description": "string", "fuel": { "id": 0, "name": "string" }, "gearbox": { "id": 0, "name": "string" }, "id": 0, "model": "string", "preferredDate": "string", "preferredDayPart": "FirstHalf", "replacementNeeded": true, "service": { "active": true, "id": 0, "name": "string", "type": "Problem" }, "status": "Canceled", "year": 0 }, "notes": "string", "price": 0, "serviceDetails": { "address": "string", "description": "string", "email": "string", "name": "string", "phoneNumber": "string", </pre>	

```

    "specialization": "string"
  },
  "status": "Canceled"
}

```

Tabela 4.4.12 Endpoint pobierania oferty dla zlecenia

GET /offer/{leadId}	Autoryzacja
	User
Parametry query	Parametry body
<brak>	<brak>
Response	
<pre> { "offers": [{ "createDate": "string", "createdUserId": 0, "id": 0, "lead": { "address": "string", "brand": { "id": 0, "name": "string" }, "canMove": true, "createdUserId": 0, "description": "string", "fuel": { "id": 0, "name": "string" }, "gearbox": { "id": 0, "name": "string" }, "id": 0, "model": "string", "preferredDate": "string", "preferredDayPart": "FirstHalf", "replacementNeeded": true, "service": { "active": true, "id": 0, "name": "string", "type": "Problem" }, "status": "Canceled", "year": 0 }, "notes": "string", "price": 0, "serviceDetails": { </pre>	

<pre> "address": "string", "description": "string", "email": "string", "name": "string", "phoneNumber": "string", "specialization": "string" }, "status": "Canceled" }] } </pre>

Tabela 4.4.13 Endpoint pobierania ofert wystawionych przez serwis

GET /offer/my, GET /offer/my/active	Autoryzacja
	Service
Parametry query	Parametry body
<brak>	<brak>
Response	
<pre> { "offers": [{ "createDate": "string", "createdUserId": 0, "id": 0, "lead": { "address": "string", "brand": { "id": 0, "name": "string" }, }, "canMove": true, "createdUserId": 0, "description": "string", "fuel": { "id": 0, "name": "string" }, "gearbox": { "id": 0, "name": "string" }, "id": 0, "model": "string", "preferredDate": "string", "preferredDayPart": "FirstHalf", "replacementNeeded": true, "service": { "id": 0, "name": "string", "type": "Problem" }, },], } </pre>	

```

        "status": "Canceled",
        "year": 0
    },
    "notes": "string",
    "price": 0,
    "status": "Canceled"
}
]
}

```

Tabela 4.4.14 Endpoint wybierania oferty przez klienta

PUT /offer/{offerId}/select	Autoryzacja
	User
Parametry query	Parametry body
<brak>	<brak>
Response	
<pre> { "createDate": "string", "createdUserId": 0, "id": 0, "lead": { "address": "string", "brand": { "id": 0, "name": "string" }, "canMove": true, "createdUserId": 0, "description": "string", "fuel": { "id": 0, "name": "string" }, "gearbox": { "id": 0, "name": "string" }, "id": 0, "model": "string", "preferredDate": "string", "preferredDayPart": "FirstHalf", "replacementNeeded": true, "service": { "active": true, "id": 0, "name": "string", "type": "Problem" }, "status": "Canceled", "year": 0 } } </pre>	

```

},
"notes": "string",
"price": 0,
"serviceDetails": {
  "address": "string",
  "description": "string",
  "email": "string",
  "name": "string",
  "phoneNumber": "string",
  "specialization": "string"
},
"status": "Canceled"
}

```

Tabela 4.4.15 Endpoint dodawania komentarza do oferty

POST /offer/comment/{offerId}	Autoryzacja
	Service
Parametry query	Parametry body
<brak>	<pre> { "comment": "string", "isStep": false } </pre>
Response	
<pre> { "id": 14, "type": "ServiceComment", "content": "string", "createdDate": "25-08-2022 10:57:03" } </pre>	

Tabela 4.4.16 Endpoint pobierania historii aktywności oferty

GET /offer/events/{offerId}	Autoryzacja
	User, Service
Parametry query	Parametry body
<brak>	<brak>
Response	

```
{
  "events": [
    {
      "content": "string",
      "createdDate": "string",
      "id": 0,
      "type": "CarReturned"
    }
  ]
}
```

Tabela 4.4.17 Endpoint zmiany ceny oferty

POST /offer/price/{offerId}	Autoryzacja
	Service
Parametry query	Parametry body
<brak>	{ "price": 0 }
Response	
<pre>{ "createDate": "string", "createdUserId": 0, "id": 0, "lead": { "address": "string", "brand": { "id": 0, "name": "string" }, "canMove": true, "createdUserId": 0, "description": "string", "fuel": { "id": 0, "name": "string" }, "gearbox": { "id": 0, "name": "string" }, "id": 0, "model": "string", "preferredDate": "string", "preferredDayPart": "FirstHalf", "replacementNeeded": true, "service": { "active": true, "id": 0, "name": "string", "type": "Problem" }, }, }</pre>	

```

    "status": "Canceled",
    "year": 0
  },
  "notes": "string",
  "price": 0,
  "serviceDetails": {
    "address": "string",
    "description": "string",
    "email": "string",
    "name": "string",
    "phoneNumber": "string",
    "specialization": "string"
  },
  "status": "Canceled"
}

```

4.4.3 Warstwa bazy danych

Do wykonania aplikacji została wykorzystana baza danych PostgreSQL [12]. Jest to popularna obiektowo-relacyjna baza danych. Do komunikacji z bazą danych z warstwy logiki biznesowej wykorzystano JPA, czyli Java Persistence API. Umożliwiło to znaczne uproszczenie obsługi danych poprzez mapowanie modelu obiektowego Java na model relacyjny bazy danych.

W celu uzyskania spójności bazy danych u wszystkich członków zespołu wykorzystana została biblioteka Liquibase [13]. Przykładowy skrypt Liquibase można zobaczyć na Rysunek 4.4.19.

```

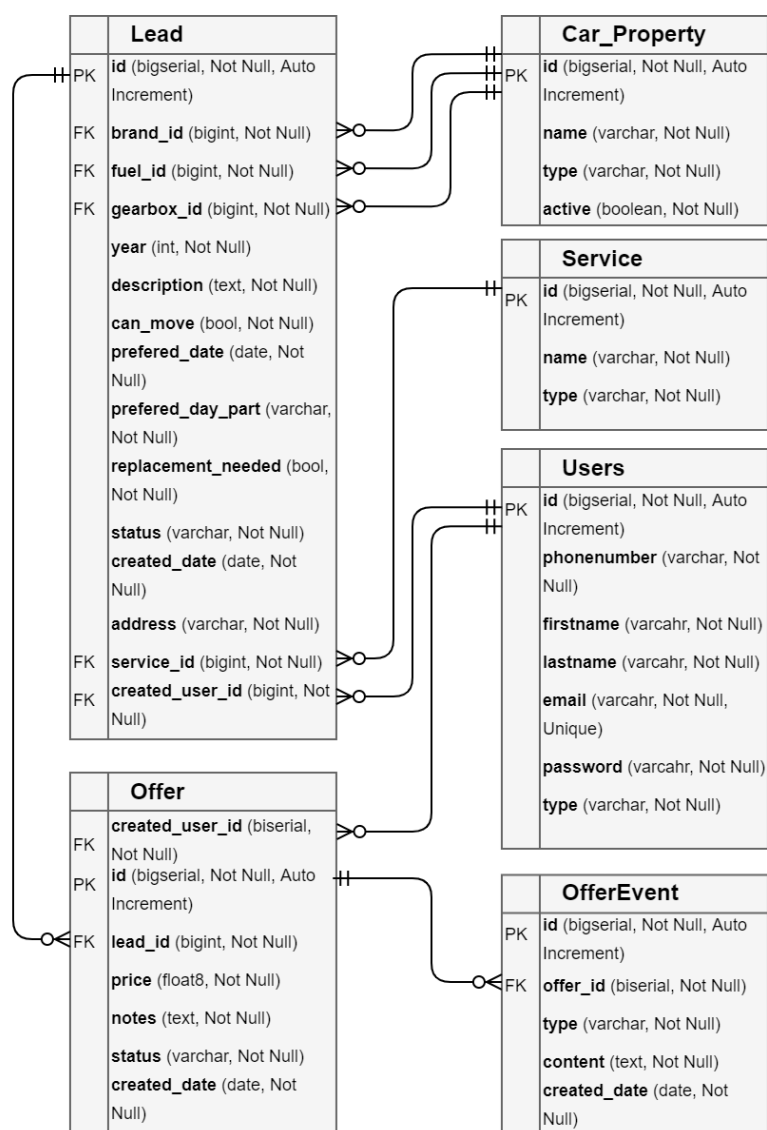
--liquibase formatted sql
--changeset hodzinets:7-insert_car_property_fuel

INSERT INTO car_property(type, name)
VALUES ('Fuel', 'Benzyna')
      , ('Fuel', 'Benzyna+LPG')
      , ('Fuel', 'Diesel')
      , ('Fuel', 'Elektryczny')
      , ('Fuel', 'Hybryda');

```

Rysunek 4.4.19 Skrypt Liquibase

Rysunek 4.4.20 przedstawia ogólny schemat bazy danych, tabelę oraz relacje między nimi.



Rysunek 4.4.20 Diagram bazy danych

Poniżej zostały przedstawione opisy poszczególnych tabel wraz z rysunkami prezentującymi techniczne szczegóły poszczególnych pól.

	Users
PK	id (bigserial, Not Null, Auto Increment) phonenummer (varchar, Not Null) firstname (varcahr, Not Null) lastname (varcahr, Not Null) email (varcahr, Not Null, Unique) password (varcahr, Not Null) type (varchar, Not Null)

Rysunek 4.4.21 Struktura tabeli Users

Users (Rysunek 4.4.21) - tabela z użytkownikami systemu. Są to kliencie zarówno serwisy.

- id - identyfikator
- phonenummer - numer komórkowy
- firstname - imię
- lastname - nazwisko
- email - adres email
- password - hash hasła
- type - typ użytkownika
 - User - zwykły użytkownik (klient)
 - Service - serwis naprawy samochodów

	Service
PK	id (bigserial, Not Null, Auto Increment) name (varchar, Not Null) type (varchar, Not Null)

Rysunek 4.4.22 Struktura tabela Service

Service (Rysunek 4.4.22) - predefiniowana lista możliwych usług

- id - identyfikator
- name - nazwa usługi (w języku polskim)
- type - typ usługi
 - Service
 - Problem

	Lead
PK	id (bigserial, Not Null, Auto Increment)
FK	brand_id (bigint, Not Null)
FK	fuel_id (bigint, Not Null)
FK	gearbox_id (bigint, Not Null)
	year (int, Not Null)
	description (text, Not Null)
	can_move (bool, Not Null)
	preferred_date (date, Not Null)
	preferred_day_part (varchar, Not Null)
	replacement_needed (bool, Not Null)
	status (varchar, Not Null)
	created_date (date, Not Null)
	address (varchar, Not Null)
FK	service_id (bigint, Not Null)
FK	created_user_id (bigint, Not Null)

Rysunek 4.4.23 Struktura tabeli Lead

Lead (Rysunek 4.4.23) - tabela zleceń

- id - identyfikator
- address - address odbioru samochodu
- service_id - typ naprawy. Klucz obcy do tabeli “Service”
- created_user_id - użytkownik tworzący zlecenie. Klucz obcy do tabeli “Users”
- description - opis problemu/samochodu
- can_move - czy samochód jeżdżący, czy potrzebuje holowania/lawety
 - true - jeżdżący
 - false - potrzebuje holowania/lawety
- preferred_date - preferowany dzień odbioru samochodu
- replacement_needed - czy jest potrzebny samochód zastępczy
- status - status zlecenia
 - Open
 - Canceled
 - Hired - została wybrana oferta
 - Finished

- created_date - data dodania zlecenia
- brand_id - marka pojazdu. Klucz obcy do tabeli “Car_Properties”
- fuel_id - typ paliwa. Klucz obcy do tabeli “Car_Properties”
- gearbox_id - typ skrzyni biegów. Klucz obcy do tabeli “Car_Properties”
- year - rocznik pojazdu

	Offer
FK	created_user_id (bserial, Not Null)
PK	id (bigserial, Not Null, Auto Increment)
FK	lead_id (bserial, Not Null)
	price (float8, Not Null)
	notes (text, Not Null)
	status (varchar, Not Null)
	created_date (date, Not Null)

Rysunek 4.4.24 Struktura tabeli Offer

Offer (Rysunek 4.4.24) - oferty od zleceniobiorców

- id - identyfikator
- created_user_id serwis tworzący ofertę. Klucz obcy do tabeli “Users”
- lead_id - zlecenie do którego była wystawiona oferta. Klucz obcy do tabeli “Lead”
- price - cena
- notes - wiadomość powitalna od zleceniobiorcy
- status - status oferty

- created_date - data dodania oferty

	OfferEvent
PK	id (bigserial, Not Null, Auto Increment)
FK	offer_id (biseial, Not Null)
	type (varchar, Not Null)
	notes (text, Not Null)
	created_date (date, Not Null)

Rysunek 4.4.25 Struktura tabeli Offer Event

OfferEvent (Rysunek 4.4.25) - wydarzenia i komentarze

- id - identyfikator
- offer_id - oferta, której dotyczy wydarzenie. Klucz obcy do tabeli Offer

Offer

- type - typ właściwości
 - OfferSelected
 - CarTaken
 - InProgress
 - ServiceFinished
 - CarReturned
 - PriceChanged,
 - ServiceStep,
 - UserComment,
 - ServiceComment

- content - komentarz
- created_date - data wydarzenia

	Car_Property
PK	id (bigserial, Not Null, Auto Increment)
	name (varchar, Not Null)
	type (varchar, Not Null)
	active (boolean, Not Null)

Rysunek 4.4.26 Struktura tabeli CallProperty

CarProperty (Rysunek 4.4.26) - predefiniowane właściwości samochodu (takie jak typ paliwa, typ skrzyni biegów, marka pojazdu)

- id - identyfikator
- name - nazwa właściwości
- type - typ właściwości
 - Brand
 - Fuel
 - Gearbox

- active - status właściwości (false jeżeli ta właściwość już nie wykorzystywana w aplikacji)

5 Przebieg projektu

5.1 Podział pracy

Projekt powstawał w zespole trzyosobowym. Praca została podzielona bazując na umiejętnościach każdego z członków zespołu.

Roman Hodzinets – implementacja części backendowej aplikacji, przygotowanie jej architektury, implementacja systemu uwierzytelnienia oraz bazy danych.

Mateusz Tondos – implementacja części frontedowej aplikacji

- Projekt architektury części frontendowej
- Implementacja zarządzania sesją użytkownika oraz widoków logowania i rejestracji
- Implementacja kroku pierwszego i drugiego tworzenia zlecenia
- Implementacja widoku komunikacji klienta i serwisu

Adrian Jeleń - Implementacja części frontendowej aplikacji

- Implementacja kroku trzeciego tworzenia zlecenia
- Implementacja widoku otwartych zleceń
- Implementacja wyświetlania listy ofert
- Implementacja listy moich zleceń

5.2 Dziennik projektu

Data: 10.2021

Opis: Spotkanie inicjujące projekt: ustalenie składu zespołu, ustalenie głównego tematu pracy, przeanalizowanie innowacyjności i praktyczności projektu.

Data: 11.2021

Opis: Uszczegółowienie zakresu i tematu pracy.

Opracowanie szkicu dokumentacji zawierającej opis problemu i wizję rozwiązania.

Ustalenie ogólnego harmonogramu prac na pierwszy semestr realizacji projektu.

Opracowanie pierwszej wersji dokumentacji zawierającej opis problemu i wizję rozwiązania.

Konsultacja projektowa i podsumowanie rezultatu dotychczasowych prac.

Ustalenie najbliższych działań i podział odpowiedzialności w projekcie.

Przygotowanie dokumentu zawierającego analizę ryzyka realizacji projektu.

Data: 12.2021

Opis: Ustalenie kolejnych działań i odpowiedzialności w projekcie.

Utworzenie drugiej wersji dokumentacji zawierającej opis problemu i wizję rozwiązania uwzględniając rekomendacje otrzymane na konsultacjach projektowych.

Utworzenie szkicu dokumentacji koncepcji systemu.

Spotkanie z promotorem. Ustalenie wymagań projektowych i współpracy. Ustalenie zarysu harmonogramu prac w kolejnym semestrze.

Data: 01.2022

Opis: Opracowanie dalszej części dokumentacji koncepcji systemu:

opracowanie diagramu komunikacji i dekompozycji na moduły.

Konsultacja projektowa i podsumowanie rezultatu dotychczasowych prac.

Zebranie rekomendacji dotyczących dokumentacji.

Zastosowanie rekomendacji dotyczącej koncepcji systemu:

przygotowanie nowej wersji diagramu komunikacji.

Data: 02.2022

Opis: Wprowadzenie usprawnień do dokumentacji koncepcyjnej:

edycja modułu powiadomień,

dodanie nowych procesów.

Spotkanie z promotorem. Konsultacja prac wykonanych w bieżącym semestrze i omówienie najbliższych działań.

Data: 03.2022

Opis: Przygotowanie szkicu dokumentacji specyfikacyjnej:

Ustalenie głównych założeń i wykonanie analizy domeny.

Data: 04.2022

Opis: Konsultacja wykonanych prac. Utworzenie grupy projektowej w systemie dostawcy kontroli wersji. Ustalenie najbliższych działań.

Spotkanie zespołu. Opracowanie pierwszej wersji dokumentacji specyfikacyjnej:

wprowadzenie poprawek do dokumentacji specyfikacyjnej

ustalenie szczegółów implementacji architektury trójwarstwowej

Utworzenie repozytorium w systemie kontroli wersji. Początek implementacji.

Omówienie wymagań niefunkcjonalnych

Utworzenie konfiguracji docker-compose i bazy postgres. Konfiguracja części serwerowej. Utworzenie konfiguracji liquibase, JPA oraz dodanie tabeli użytkowników oraz implementacja autoryzacji.

Data: 05.2022

Opis: Inicjalizacja części klienckiej systemu. Utworzenie formularza tworzenia zapytania ofertowego. Konfiguracja komunikacji z częścią serwerową.

Prace nad implementacją obsługi zleceń.

Konsultacja wykonanych prac z promotorem. Ustalenie najbliższych działań.

Prace nad dokumentacją deweloperską, przygotowanie diagramu struktury bazy danych.

Data: 07.2022

Opis: Implementacja możliwości dodania oferty dla serwisu.

Data: 08.2022

Opis: Dodanie opisu serwisu

Dodanie możliwość wyboru oferty,

Implementacja wybrania oferty przez klienta

Implementacja możliwości komunikacji pomiędzy klientem a serwisem

Dodanie możliwości zmiany statusu oferty, jej ceny oraz opisu postępowania prac naprawczych

5.3 Zastosowane techniki i praktyki

Elementami na które warto zwrócić uwagę to zastosowane techniki i praktyki. Umiejętny dobór i ich wykorzystanie pozwoliło lepiej wykorzystać dostępny czas. W tym celu zastosowaliśmy techniki i narzędzia umożliwiające łatwą komunikację i organizację pracy w zespole, oprogramowanie i biblioteki umożliwiające automatyzację niektórych procesów i łatwą synchronizację środowisk pomiędzy członkami zespołu. Zastosowanie wszystkich elementów przyczyniło się do sprawnego wykonania projektu, uniknięcia problemów organizacyjnych i jednocześnie umożliwiło zachowanie wysokiej jakości kodu. Wybrane elementy na które warto zwrócić uwagę to:

- **Agile** – technika zwinnego tworzenia oprogramowania polegająca na tworzeniu małych przyrostów. Połączona z dobrą komunikacją w zespole umożliwiła szybką implementację oprogramowania z możliwością wprowadzania zmian w trakcie pracy nad projektem.
- **Code review** – wykorzystana technika polegała na regularnym sprawdzaniu aktualizacji kodu wykonanych przez pozostałych członków zespołu. Pozwoliła na zminimalizowanie potencjalnych problemów związanych z poprawnością implementacji oraz ujednolicenie stylistyki kodu.
- **Burza mózgów** – technika która ułatwiła podejmowanie trafniejszych decyzji związanych z całym projektem na etapie tworzenia założeń, jak i rozwiązywania dalszych problemów związanych z projektem i implementacją systemu.
- **Docker** – narzędzie służące do łatwego utworzenia identycznego środowiska na którym była uruchomiona aplikacja w trakcie tworzenia oprogramowania u wszystkich członków zespołu. Umożliwiło to uniknięcie problemów związanych z różnicą pomiędzy rodzajem systemu lub wersją stosowanych produktów. Ułatwiła łatwe i automatyczne odtworzenie środowiska w przypadku problemów.
- **Git** – narzędzie do zarządzania aktualizacjami kodu. Pozwoliło na spójne i przejrzyste wprowadzanie zmian w kodzie. Ułatwiło proces code review i bieżącą synchronizację kodu pomiędzy członkami zespołu.

6 Podsumowanie

6.1 Porównanie specyfikacji systemu z dostarczoną wersją

Efektem naszej pracy jest system trójwarstwowy składający się z aplikacji klienckiej, serwerowej oraz bazy danych. Powstały system zapewnia kluczowe funkcjonalności jakich potrzebują aktorzy systemu aby wykonać i odebrać usługę, to jest: funkcjonalności złożenia zlecenia, wyświetlania otwartych zleceń, tworzenia, wyświetlania i akceptowania ofert, powiązania usługodawcy ze zleceniem, umówieniem spotkania w celu odebrania i zwrotu samochodu oraz realizacji usługi wspieranej obustronną komunikacją w ramach widoku czatu. Dodatkowo założono, że serwis również może oczekiwać podglądu swoich ofert i ich aktualnych statusów, co domknęło w systemie kompletność dostępu do danych wcześniej wprowadzonych do systemu przekraczając wymagania. Zaimplementowano również możliwość zmiany ceny usługi, co nie było zdefiniowane w specyfikacji.

W systemie nie zrealizowano kilku funkcjonalności niekoniecznych dla wykonania i odebrania usługi.

Nie zrealizowano funkcjonalności wystawiania serwisowi opinii użytkownika jako niekrytycznej dla procesu wykonania i odebrania zamówienia. Decyzję o niezrealizowaniu podjęto również dlatego, iż w przypadku małej ilości użytkowników bezpośrednio po rozpoczęciu działania systemu istnieje ryzyko potencjalnego nadużycia ze strony złośliwych użytkowników poprzez wprowadzanie pojedynczych nieuczciwych ocen istotnie wpływające na opinię serwisu. Mogłoby to spowodować zniechęcenie do korzystania z systemu przez serwisy samochodowe oraz użytkowników.

Nie zaimplementowano modułu komunikacji, który miał odpowiadać za wysyłanie do klienta i serwisu powiadomień wybranym kanałem, między innymi powiadamianie użytkownika, gdy pojawiały się nowe oferty na wystawione zlecenie oraz serwisu, gdy klient zaakceptował ofertę. Ta funkcjonalność również nie mieściła się w wizji projektu przedstawionej w punkcie 1.1 Charakter i cel pracy.

6.2 Ocena przebiegu projektu i produktu końcowego

Jesteśmy zadowoleni z przebiegu projektu. Powstały system spełnia założenia, które zdefiniowaliśmy jako cel projektu, niejednokrotnie je przekraczając. W trakcie trwania projektu nauczyliśmy się organizacji wielomiesięcznej pracy w małym zespole oraz tworzenia projektu na wspólnie z dokumentami opisującymi projekt i implementację.

6.3 Plan rozwoju aplikacji

Powstały system ma potencjał rozwoju wszcz – poprzez rozszerzenie domeny na inne branże, w których zastosowany wzorzec odwrócenia sterowania może być korzystny dla konsumenta – oraz rozwoju w głąb, poprzez dopracowanie doświadczenia serwisu i klienta. Przykładowymi funkcjonalnościami, które mogłyby usprawnić system są:

- Modyfikacja danych konta klienta
- Wysyłanie powiadomień do aktorów systemu podczas wystąpienia zmian, którymi zainteresowani byłiby aktorzy
- Dodawanie i wyświetlanie opinii klientów o serwisie
- Możliwość wysyłania zdjęć / filmów jako załączników do informacji o przebiegu procesu naprawy przez serwis

7 Słownik pojęć

Klient – aktor systemu będący zleceniodawcą.

Serwis – aktor systemu będący zleceniobiorcą.

Zlecenie – Zapytanie ofertowe zawierające informacje o potrzebach klienta w stosunku do jego pojazdu. Na otwarte zlecenie serwisy mogą składać oferty.

Oferta – Propozycja wykonania zlecenia, skierowana do klienta przez serwis, zawierająca notatkę powitalną oraz wycenę usługi.

Rest API – styl architektury, który opisuje interfejs pomiędzy warstwą prezentacji i logiki.

Endpoint API – element udostępniany na zewnątrz warstwy logiki, identyfikowany poprzez przypisany adres dostępowy.

Komponent – element interfejsu użytkownika, który może być wielokrotnie wykorzystany.

Widok – Unikatowa zawartość strony znajdująca się pod adresem dostępowym, tworzona przez zespół komponentów.

8 Spis rysunków

Rysunek 3.3.1 Diagram komunikacji	12
Rysunek 4.2.1 Architektura systemu.....	26
Rysunek 4.3.1 Ogólna struktura kodu źródłowego części serwerowej	27
Rysunek 4.3.2 Ogólna struktura kodu źródłowego warstwy prezentacji	28
Rysunek 4.4.1 Widok pierwszej strony tworzenia zlecenia - CreateLeadPage	31
Rysunek 4.4.2 Widok drugiej strony tworzenia zlecenia - CreateLeadPage	31
Rysunek 4.4.3 Widok trzeciej strony tworzenia zlecenia - CreateLeadPage	32
Rysunek 4.4.4 Widok ekranu rejestracji	32
Rysunek 4.4.5 Widok ekranu logowania	33
Rysunek 4.4.6 Widok listy zleceń - klient	33
Rysunek 4.4.7 Widok szczegółów zlecenia - rozwinięta opcja w menu.....	34
Rysunek 4.4.8 Widok szczegółów zlecenia - brak ofert	34
Rysunek 4.4.9 Widok szczegółów oferty - możliwość akceptacji	34
Rysunek 4.4.10 Widok komunikacji klienta i serwisu.....	35
Rysunek 4.4.11 Widok listy oczekujących ofert - serwis	36
Rysunek 4.4.12 Widok listy aktywnych napraw – serwis.....	36
Rysunek 4.4.13 Widok tworzenia oferty – serwis	36
Rysunek 4.4.14 Widok zlecenia ze złożoną ofert	37
Rysunek 4.4.15 Widok komunikacji klienta i serwisu - część serwisu.....	38
Rysunek 4.4.16 Widok edycji danych serwisu	39
Rysunek 4.4.17 Widok komunikacji klienta i serwisu na telefonie	40
Rysunek 4.4.18 Widok listy zleceń – klient.....	40
Rysunek 4.4.19 Skrypt Liquibase	53
Rysunek 4.4.20 Diagram bazy danych.....	54
Rysunek 4.4.21 Struktura tabeli Users	55

Rysunek 4.4.22 Struktura tabela Service	55
Rysunek 4.4.23 Struktura tabeli Lead	56
Rysunek 4.4.24 Struktura tabeli Offer	56
Rysunek 4.4.25 Struktura tabeli Offer Event	57
Rysunek 4.4.26 Struktura tabeli CallProperty	57

9 Spis tabel

Tabela 3.4.1 Dane przetwarzane przez poszczególne widoki – moduł interfejsu użytkownika	16
Tabela 3.4.2 Dane przetwarzane przez poszczególne widoki – moduł interfejsu serwisu	17
Tabela 3.5.1 Lista końcówek API przynależących do poszczególnych modułów	19
Tabela 3.7.1 Szczegółowe elementy analizy ryzyka	21
Tabela 4.4.1 Endpoint logowania	41
Tabela 4.4.2 Endpoint rejestracji	41
Tabela 4.4.3 Endpoint pobierania danych użytkownika	42
Tabela 4.4.4 Endpoint zmiany danych serwisu	42
Tabela 4.4.5 Endpoint pobierania danych serwisu	43
Tabela 4.4.6 Endpoint pobierania predefiniowanych usług	43
Tabela 4.4.7 Endpoint dodania zlecenia	43
Tabela 4.4.8 Endpoint pobierania usług klienta	44
Tabela 4.4.9 Endpoint pobierania otwartych usług	45
Tabela 4.4.10 Endpoint pobierania predefiniowanych właściwości samochodów	46
Tabela 4.4.11 Endpoint tworzenia oferty	47
Tabela 4.4.12 Endpoint pobierania oferty dla zlecenia	48
Tabela 4.4.13 Endpoint pobierania ofert wystawionych przez serwis	49
Tabela 4.4.14 Endpoint wybierania oferty przez klienta	50
Tabela 4.4.15 Endpoint dodawania komentarza do oferty	51
Tabela 4.4.16 Endpoint pobierania historii aktywności oferty	51
Tabela 4.4.17 Endpoint zmiany ceny oferty	52

Bibliografia

- [1] Microsoft, „TypeScript Documentation,” [Online]. Available: <https://www.typescriptlang.org/docs/>. [Data uzyskania dostępu: 28 08 2022].
- [2] Oracle, „Java Documentation,” [Online]. Available: <https://docs.oracle.com/en/java/>. [Data uzyskania dostępu: 28 08 2022].
- [3] Pivotal, Inc., „Spring Framework Documentation,” 14 07 2022. [Online]. Available: <https://docs.spring.io/spring-framework/docs/current/reference/html/>. [Data uzyskania dostępu: 28 08 2022].
- [4] The Project Lombok Authors, „Project Lombok,” [Online]. Available: <https://www.projectlombok.org/features/all>. [Data uzyskania dostępu: 28 08 2022].
- [5] Auth0 Inc., „Introduction to JSON Web Tokens,” [Online]. Available: <https://jwt.io/introduction>. [Data uzyskania dostępu: 28 08 2022].
- [6] VMware, Inc., „Spring Security Get Started,” [Online]. Available: <https://docs.spring.io/spring-security/reference/index.html>. [Data uzyskania dostępu: 28 08 2022].
- [7] Meta Platforms, Inc., „React Get Started,” [Online]. Available: <https://reactjs.org/docs/getting-started.html>. [Data uzyskania dostępu: 28 08 2022].
- [8] T. Linsley, „React Query Get Started,” [Online]. Available: <https://react-query-v3.tanstack.com/overview>. [Data uzyskania dostępu: 08 28 2022].
- [9] Formium, Inc, „Formik Get Started,” [Online]. Available: <https://formik.org/docs/overview>. [Data uzyskania dostępu: 28 08 2022].
- [10] Material UI SAS., „Material UI Getting Started,” [Online]. Available: <https://mui.com/material-ui/getting-started/overview/?fbclid=IwAR3IjY3DDJFlqPBsQuItINSyJ3Ybv2ua1jI7wTUIO5ozhViGdwgcP8VECvs>. [Data uzyskania dostępu: 28 08 2022].

- [11] Remix Software Inc., „Welcome to React Router,” [Online]. Available: <https://reactrouter.com/en/main>. [Data uzyskania dostępu: 28 08 2022].
- [12] The PostgreSQL Global Development Group, „PostgreSQL 14.5 Documentation,” [Online]. Available: <https://www.postgresql.org/docs/current/>. [Data uzyskania dostępu: 28 08 2022].
- [13] Liquibase Inc., „Liquibase Documentation,” [Online]. Available: <https://docs.liquibase.com/home.html>. [Data uzyskania dostępu: 28 08 2022].
- [14] acacode, „swagger-typescript-api,” [Online]. Available: <https://github.com/acacode/swagger-typescript-api>. [Data uzyskania dostępu: 28 08 2022].