

POLITECHNIKA ŚWIĘTOKRZYSKA

Wydział Elektrotechniki, Automatyki i Informatyki

Konrad Niżnik

Numer albumu: 92887

System zarządzania warsztatem samochodowym wspomagany przez AI

**Praca dyplomowa
na studiach I-go stopnia
na kierunku Informatyka**

Promotor pracy dyplomowej:

dr inż. Mariusz Wiśniewski

Nazwa Katedry, w której promotor jest zatrudniony

Katedra Systemów Informatycznych

Kielce 2025

Zatwierdzam:

Rok akademicki: 2024/25

PRODZIEKAN
ds. Studenckich i Dydaktyki
Wydział Elektrotechniki, Automatyki i Informatyki
.....
Prodziekan ds. studenckich i dydaktyki
dr inż. Barbara Jakubowska

ZADANIE NA PRACĘ DYPLOMOWĄ STUDIÓW PIERWSZEGO STOPNIA

Wydano studentowi:

Konrad Niżnik

nr albumu: 92887

I. Temat pracy:

System zarządzania warsztatem samochodowym wspomagany przez AI AI-powered car repair shop management system

II. Cel pracy:

Celem pracy jest wykonanie aplikacji internetowej, realizującej zadania wspomagające pracę warsztatu samochodowego. Spodziewa się, że aplikacja będzie dostarczała funkcje zgodne z jej przeznaczeniem oraz umożliwiała zarządzanie użytkownikami (personalem). Zakres wykorzystania sztucznej inteligencji pozostaje w uznaniu Dyplomanta. Technologie wykonania frontendu i backendu można dobrać dowolnie.

III. Plan pracy (zakres pracy):

1. Omówienie technologii wykorzystanych do realizacji projektu.
2. Analiza systemowa i projekt techniczny.
3. Wykonanie i prezentacja aplikacji celowej projektu.
4. Testy jednostkowe, integracyjne.

IV. Uwagi dotyczące pracy: brak.

V. Termin oddania pracy: zgodnie z Regulaminem Studiów.

VI. Konsultant:

Opiekun merytoryczny

Promotor pracy dyplomowej

dr inż. Mariusz Wiśniewski

KIEROWNIK
Katedry Systemów Informatycznych
Wydziału Elektrotechniki, Automatyki i Informatyki
.....
dr hab. inż. Roman Olszowski, prof. PŚk
(podpis)

.....
(podpis)

Temat pracy dyplomowej celem jej wykonania otrzymałem(am):

Kielce, dnia 02.06.2025 r.

Konrad Niżnik
.....
czytelny podpis studenta

Kielce, dnia 24.06.2025

Konrad Nizimik 92887
Imię i nazwisko studenta, nr albumu
ul. Pasmysłowa 4/48, Kraków 30-701
Adres zamieszkania
Studia pierwszego stopnia, niestacjonarne
Studia pierwszego/drugiego* stopnia, forma studiów stacjonarno/niestacjonarno
Informatyka, Systemy Informatyczne
Kierunek, zakres
dr inż. Mariusz Klimiński
Promotor pracy dyplomowej

OŚWIADCZENIE

Przedkładając w roku akademickim 2024/25. promotorowi pracy dyplomowej studiów pierwszego/drugiego* stopnia, powołanemu przez Dziekana Wydziału Elektrotechniki, Automatyki i Informatyki Politechniki Świętokrzyskiej, pracę dyplomową pod tytułem: System zarządzania warsztatem samochodowym wspomagany przez AI

oświadczam, że:

- 1) przedstawiona praca dyplomowa została opracowana przeze mnie samodzielnie, stosownie do wskazówek merytorycznych opiekuna pracy,
- 2) przy wykonywaniu pracy dyplomowej wykorzystano materiały źródłowe, w granicach dozwolonego użytku wymieniając autora, tytuł pozycji i miejsce jej publikacji,
- 3) praca dyplomowa nie zawiera żadnych danych, informacji i materiałów, których publikacja nie jest prawnie dozwolona,
- 4) przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego/stopnia naukowego w wyższej uczelni,
- 5) niniejsza wersja pracy jest identyczna z treścią elektroniczną w systemie Archiwum Prac Dyplomowych.

Przyjmuję do wiadomości, że w przypadku ujawnienia w mojej pracy dyplomowej, stanowiącej podstawę nadania tytułu zawodowego, przypisania sobie przeze mnie autorstwa istotnego fragmentu lub innych elementów cudzego utworu lub ustalenia naukowego, rektor, w drodze decyzji administracyjnej, stwierdzi nieważność dyplomu.

Zostałem uprzedzony:

- 1) o odpowiedzialności karnej wynikającej z art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t. j. Dz. U. z 2022 r. poz. 2509 ze zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”
- 2) o odpowiedzialności dyscyplinarnej na podstawie art. 307 ust. 1 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (t. j. Dz. U. z 2022 r. poz. 574, ze zm.): „Student podlega odpowiedzialności dyscyplinarnej za naruszenie przepisów obowiązujących w uczelni oraz za czyn uchybiający godności studenta.”

Konrad Nizimik
Czytelny podpis studenta

*niepotrzebne skreślić

Załącznik Nr 2
do Antyplagiatowej procedury sprawdzania prac dyplomowych
i elektronicznej archiwizacji prac dyplomowych w Politechnice Świętokrzyskiej
wprowadzonej Zarządzeniem Rektora Nr 21/23

Kielce, dnia 24.06.2025.....

Kamrad Nikiel 92887
Imię i nazwisko studenta, nr albumu
ul. Przemysłowa 4/48, Kraków 3D-FD1
Adres zamieszkania
Student pierwszego stopnia, niestacjonarne
Studia pierwszego stopnia, formy studiów stacjonarnie/stacjonarne
Informatyka, Systemy Informatyczne
Kierunek, zakres
dr. inż. Mariusz Kisiński
Promotor pracy dyplomowej

OŚWIADCZENIE AUTORA PRACY

Zgodnie z ustawą z dnia 4 lutego 1994r. o prawie autorskim i prawach pokrewnych
(t.j. Dz. U. 2022 poz. 2509), wyrażam zgodę na udostępnianie mojej pracy dyplomowej dla celów naukowych
i dydaktycznych.

Kamrad Nikiel
Czytelny podpis studenta

Tytuł pracy w języku polskim

System zarządzania warsztatem samochodowym wspomagany przez AI

Streszczenie

Celem projektu było stworzenie systemu informatycznego, który pozwoli na efektywne zarządzanie usługami w warsztacie samochodowym, wykorzystując sztuczną inteligencję. Opracowana aplikacja internetowa daje klientom możliwość rezerwacji usług oraz interakcji z chatbotem AI, który pomaga w doborze właściwej usługi na podstawie opisanego problemu. Pracownicy mogą obsługiwać rezerwacje, aktualizować postępy napraw oraz kontaktować się z klientami. Rolą administratora jest zarządzanie użytkownikami oraz ofertą dostępnych usług. System zaprojektowano w architekturze klient-serwer, wykorzystując technologie takie jak React, Spring Boot oraz PostgreSQL.

Słowa kluczowe: - system informatyczny, warsztat samochodowy, sztuczna inteligencja, chatbot, Spring Boot, React

AI-powered car repair shop management system

Summary

The aim of this thesis was to develop an information system for managing car workshop services with the support of artificial intelligence. The implemented web application allows customers to book services and interact with an AI-powered chatbot that recommends suitable services based on described vehicle issues. Employees can manage reservations, update repair statuses, and communicate with clients. Administrators control the service offerings and user roles. The system is built using a client-server architecture with React, Spring Boot, and PostgreSQL technologies.

Keywords: - information system, car workshop, artificial intelligence, chatbot, Spring Boot, React

Spis treści

1. Wstęp	13
2. Przegląd podobnych aplikacji dostępnych na rynku	14
2.1. Warsztat24	14
2.2. Inter Cars eSOW	14
2.3. Podsumowanie analizy istniejących rozwiązań	14
3. Analiza problemu i wymagania systemowe	16
3.1. Funkcje dostępne przed zalogowaniem	16
3.2. Funkcje klienta po rejestracji i zalogowaniu	16
3.3. Funkcje pracownika warsztatu	17
3.4. Funkcje administratora	17
3.5. Wsparcie sztucznej inteligencji	18
3.6. Integracje i bezpieczeństwo	18
4. Przegląd technologii	19
4.1. Spring Framework (Java)	19
4.2. React (JavaScript)	19
4.3. Tailwind CSS	20
4.4. PostgreSQL	21
4.5. PayU – system płatności online	23
4.6. ChatGPT (OpenAI API)	23
5. Projekt systemu	25
6. Implementacja systemu	27
6.1. Panel klienta	27
6.1.1. Widok główny panelu klienta	27
6.1.2. Wybór terminu wizyty	28
6.1.3. Wybór usług warsztatowych	29
6.1.4. Podsumowanie koszyka i wybór metody płatności	30
6.1.5. Proces płatności integracja z PayU	31
6.1.6. Podgląd i zarządzanie rezerwacjami	32
6.1.7. Edycja danych klienta	35
6.1.8. Diagnozowanie usterki za pomocą AI	37
6.2. Panel pracownika	40
6.2.1. Zamówienia do obsługi	41
6.2.2. Edytuj dostępne terminy	42

6.2.3. Edytuj dostępne terminy	44
6.3. Panel administratora	45
6.3.1. Zarządzanie użytkownikami	45
6.3.2. Dodawanie nowego użytkownika	48
6.3.3. Zarządzanie usługami	49
6.4. Funkcje ogólnodostępne	51
6.4.1. Strona główna	51
6.4.2. Usługi	51
6.4.3. Kontakt	52
6.4.4. O nas	52
7. Testowanie aplikacji	53
7.1. Rodzaje testów w procesie wytwarzania oprogramowania	53
7.2. Zakres wykonanych testów aplikacji	54
8. Podsumowanie	56
Bibliografia	57
Załączniki	59

1. Wstęp

W obecnych czasach trudno jest sobie wyobrazić przedsiębiorstwo, które nie korzysta z technologii informacyjnych do organizacji codziennych zadań. Tak samo jest w sektorze motoryzacyjnym – warsztaty samochodowe również coraz chętniej sięgają po rozwiązania, które umożliwiają optymalizację pracy, poprawiają komunikację z klientem oraz lepsze planowanie napraw.

Niestety, wiele z dostępnych systemów na rynku to skomplikowane i drogie platformy, które niekoniecznie są odpowiednie dla małych warsztatów. Zwykle są one przeładowane funkcjami, które są rzadko używane, albo, przeciwnie, nie posiadają podstawowych narzędzi, jak chociażby szybka rezerwacja wizyt online. Dodatkowo, brakuje w nich integracji z nowoczesnymi technologiami, takimi jak sztuczna inteligencja. Te niedoskonałości zainspirowały mnie do stworzenia własnego systemu – czytelnego, łatwego w obsłudze, ale i nowoczesnego.

W rezultacie powstała aplikacja internetowa, która wspiera zarówno klientów, jak i pracowników warsztatu. Umożliwia ona z jednej strony szybkie umawianie usług, sprawdzanie postępu naprawy oraz kontakt z obsługą. Z drugiej strony pozwala pracownikom zarządzać grafiką pracy, edytować zlecenia oraz przeglądać historię wizyt. Całość opiera się na rolach użytkowników – klient, pracownik oraz administrator – co pozwala każdemu widzieć tylko te informacje, które są dla niego istotne.

Cennym dodatkiem w tym projekcie jest integracja z AI – chatbot bazujący na GPT-4 pozwala klientowi opisać problem z pojazdem prostym językiem, a system automatycznie dopasowuje odpowiednią usługę. To prawdziwe ułatwienie, szczególnie dla tych, którzy nie mają wiedzy na temat mechaniki, a chcą jedynie zgłosić usterkę i umówić się na naprawę.

W kolejnych rozdziałach szczegółowo opisano, jak przebiegał cały proces tworzenia systemu – od zaplanowania funkcjonalności, przez dobór technologii, aż po testy i wdrożenie. Celem tej pracy było stworzenie praktycznego rozwiązania, które mogłoby znaleźć zastosowanie w rzeczywistym warsztacie.

2. Przegląd podobnych aplikacji dostępnych na rynku

Na rynku dostępnych jest wiele programów komputerowych, które wspierają zarządzanie warsztatami samochodowymi. Różnią się one zakresem funkcji, poziomem zaawansowania technologicznego, modelem licencjonowania oraz grupą docelowych użytkowników (małe warsztaty, autoryzowane stacje obsługi, sieci serwisowe). W tym rozdziale przedstawiono przegląd kilku popularnych rozwiązań, które można znaleźć na polskim rynku.

2.1. Warsztat24

Warsztat24 to platforma internetowa stworzona z myślą o niewielkich i średnich warsztatach samochodowych. Umożliwia ona zarządzanie harmonogramem napraw, organizowanie bazy klientów, rejestrowanie części zamiennych oraz realizację operacji finansowych. System funkcjonuje w modelu SaaS, co umożliwia użytkowanie aplikacji przez przeglądarkę, bez potrzeby instalacji na komputerze. Co więcej, program zapewnia współpracę z popularnymi hurtowniami motoryzacyjnymi, wystawianie faktur, tworzenie raportów oraz obsługę plików JPK. W wersji podstawowej możliwości dostosowania interfejsu oraz funkcjonalności są ograniczone, co może być problematyczne dla warsztatów potrzebujących bardziej spersonalizowanych rozwiązań.[1]

2.2. Inter Cars eSOWA

System eSOW to rozwiązanie proponowane przez Inter Cars, skierowane do warsztatów współpracujących z tą siecią dystrybucji części. Umożliwia ono kompleksowe zarządzanie zleceniami serwisowymi, zamawianie części oraz dostęp do katalogów produktów i aktualnych cen. Kluczową zaletą tego systemu jest pełna integracja z bazą danych Inter Cars, co znacznie ułatwia proces zamówień i przyspiesza realizację napraw. System funkcjonuje w trybie online, co pozwala na korzystanie z niego z różnych miejsc, jednak jego pełne możliwości są dostępne przy stałej współpracy z Inter Cars.[2]

2.3. Podsumowanie analizy istniejących rozwiązań

Analizując dostępne rozwiązania, można dostrzec, że większość z nich oferuje podstawowe funkcje związane z warsztatem, takie jak:

- obsługa zleceń,
- utrzymywanie bazy danych klientów i pojazdów,
- zarządzanie magazynem części,
- wystawianie faktur i rozliczeń,
- tworzenie raportów.

Jednak brakuje nowoczesnych funkcji, takich jak:

- płatności internetowe zintegrowane z systemem,
- chatboty wspomagające kontakt z klientem,
- pełna mobilność i dostępność z różnych urządzeń oraz lokalizacji.

System informatyczny, nad którym pracuję w tej pracy, ma spełniać te potrzeby poprzez integrację z usługami online, elastyczną architekturę internetową oraz wsparcie dla elementów sztucznej inteligencji.

3. Analiza problemu i wymagania systemowe

Współczesne warsztaty samochodowe działają w zmiennym środowisku, w którym konieczne jest jednocześnie zarządzanie obsługą klientów, organizacją harmonogramów pracy, kontrolowaniem postępu napraw oraz gwarantowaniem wysokiej jakości i terminowości świadczonych usług. Choć na rynku dostępnych jest wiele rozwiązań informatycznych, które wspierają te działania, to większość z nich nie spełnia potrzeb mniejszych warsztatów; są one zazwyczaj przestarzałe, zbyt skomplikowane lub po prostu drogie w użytkowaniu. Brakuje również systemów, które kompleksowo i w sposób intuicyjny ułatwiałyby zarządzanie obsługą klienta, począwszy od pierwszego kontaktu, aż do zakończenia naprawy.

W odpowiedzi na te wymagania, tworzony system ma na celu zapewnienie webowego rozwiązania wspomaganego przez sztuczną inteligencję, które będzie odpowiadać rzeczywistym potrzebom użytkowników. Struktura systemu została zaplanowana z uwzględnieniem trzech kluczowych grup użytkowników: klienta, pracownika oraz administratora. Każda z tych ról ma przydzielony różny poziom dostępu i zestaw funkcji dopasowanych do realizowanych zadań.

3.1. Funkcje dostępne przed zalogowaniem

Użytkownik systemu ma możliwość skorzystania z kilku ważnych funkcji jeszcze przed zalogowaniem. Może zobaczyć kompletny wykaz dostępnych usług serwisowych wraz z obowiązującymi cenami, sprawdzić galerię wcześniejszych napraw oraz użyć formularza kontaktowego, aby zadać pytanie lub umówić się na indywidualną konsultację. Taki poziom informacji i możliwości interakcji umożliwia użytkownikowi dokonanie oceny oferty warsztatu jeszcze zanim podejmie decyzję o rejestracji.

3.2. Funkcje klienta po rejestracji i zalogowaniu

Klient, po stworzeniu swojego konta, ma możliwość zalogowania się do systemu za pomocą standardowego formularza (z użyciem tokena JWT) albo konta Google. Po zalogowaniu uzyskuje dostęp do dostosowanego panelu użytkownika, w którym ma możliwość:

- rezerwacji usług w dostępnych terminach,
- oglądania historii swoich rezerwacji oraz statusów ich realizacji,
- anulowania zaplanowanych wizyt,
- przeprowadzania płatności online poprzez system PayU lub wybierania opcji płatności na miejscu,
- zmiany swoich danych osobowych (na przykład danych kontaktowych),
- otrzymywania powiadomień o statusie usługi lub przypomnień dotyczących nadchodzących wizyt.

Dodatkowo, klient ma możliwość korzystania z wbudowanego chatbota, opartego na sztucznej inteligencji (na przykład ChatGPT), który może doradzić w przypadku wątpliwości co do rodzaju potrzebnej usługi. Klient opisuje objawy lub problemy z pojazdem, a chatbot zasugeruje najbardziej odpowiednią usługę do zarezerwowania. To rozwiązanie znacząco poprawia komfort obsługi i eliminuje potrzebę kontaktu telefonicznego.

3.3. Funkcje pracownika warsztatu

Panel pracowniczy warsztatu udostępnia szereg narzędzi, które pozwalają na zarządzanie aktualnymi procesami serwisowymi. Pracownik może:

- sprawdzać oraz modyfikować rezerwacje klientów,
- usuwać lub przenosić rezerwacje,
- aktualizować stany realizowanych napraw (np. "przyjęte", "w toku", "zakończone"),
- zarządzać kalendarzem oraz dostępnymi terminami usług,
- prowadzić komunikację z klientem przez e-mail,
- monitorować historię wykonanych usług przypisanych do danego pojazdu lub klienta.

Taki pakiet narzędzi umożliwi osobie pracującej szybkie dostosowywanie się do zmieniających się okoliczności (np. spóźnienia w dostarczeniu komponentów) oraz zapewnia przejrzystość i punktualność w wykonaniu zadań.

3.4. Funkcje administratora

Administrator dysponuje najszerszym zakresem uprawnień w systemie, który obejmuje zarówno aspekty zarządzania, jak i techniczne. Do jego istotnych obowiązków należą:

- tworzenie, modyfikowanie i usuwanie kont użytkowników (w tym klientów oraz pracowników),
- przydzielanie ról oraz zarządzanie dostępem,
- zmiana i dodawanie nowych usług do oferty warsztatu,
- kompletna edycja informacji użytkowników (w tym dane osobowe i kontaktowe),

Osoba na stanowisku administratora jest odpowiedzialna za administrowanie kontami użytkowników, ustawianie opcji usług oraz kontrolowanie, czy aplikacje działają prawidłowo w aspekcie operacyjnym i organizacyjnym.

3.5. Wsparcie sztucznej inteligencji

System korzysta z komponentów sztucznej inteligencji, w tym zintegrowanego chatbota, który pomaga klientom w podejmowaniu decyzji związanych z wyborem usługi. Chatbot bada podane przez użytkownika informacje o objawach, proponuje odpowiadające usługi i umożliwia szybkie przejście do rezerwacji. W przyszłości planuje się rozszerzenie funkcji sztucznej inteligencji o analizowanie danych z przeszłości, aby przewidywać obciążenie warsztatu, usprawniać harmonogramy oraz wstępnie diagnozować problemy.

3.6. Integracje i bezpieczeństwo

Projektowany system został stworzony z myślą o zapewnieniu bezpiecznego dostępu oraz możliwości przyszłych integracji z różnorodnymi usługami zewnętrznymi. Obecnie oferuje dwa sposoby weryfikacji tożsamości użytkowników: tradycyjne logowanie z użyciem tokenów JWT oraz logowanie przez konto Google. Zarządzaniem tymi procesami zajmuje się biblioteka Spring Security, która umożliwia dokładne przypisanie uprawnień w zależności od roli użytkownika (klient, pracownik, administrator).

Jeśli chodzi o integracje, system jest przystosowany do obsługi płatności online, szczególnie we współpracy z systemem PayU. Klient ma możliwość dokonania płatności przez Internet lub wyboru płatności bezpośrednio w warsztacie.

Chociaż obecnie system nie jest zintegrowany z zewnętrznymi systemami księgowymi ani magazynowymi, jego konstrukcja pozwala na rozwój w tym obszarze. Integracja z API dostawców części oraz systemami ERP stanowi możliwy kierunek rozwoju aplikacji, co w przyszłości pozwoli na dalszą automatyzację i optymalizację procesów serwisowych.

4. Przegląd technologii

W trakcie realizacji projektu zastosowano zestaw innowacyjnych, skalowalnych technologii, które są dobrze opisane, a ich wybór miał na celu zapewnienie dużej dostępności, bezpieczeństwa danych, elastyczności w rozwoju oraz łatwości użytkowania. Zastosowano model architektury klienckiej i serwerowej, z oddzielnymi warstwami dla frontendu, backendu, a także dla komunikacji z bazą danych i zewnętrznymi serwisami. Wszystkie wybrane narzędzia są aktywnie rozwijane przez społeczności open source i wspierane w sposób komercyjny, co wpływa na stabilność rozwiązania oraz jego długotrwałość.

4.1. Spring Framework (Java)

Jednym z kluczowych elementów warstwy backendu jest Spring Framework – popularne narzędzie do programowania w ekosystemie języka Java, które umożliwia budowanie aplikacji o dużej skali, modularnych i łatwych do testowania [3].

W projekcie zastosowano następujące komponenty:

- Spring Boot – umożliwia szybkie uruchamianie aplikacji przy minimalnej konfiguracji, co eliminuje potrzebę radzenia sobie z złożoną strukturą projektu. Umożliwia również tworzenie mikroustug, które są w pełni gotowe do produkcji [4].
- Spring Security – zajmuje się kontrolą dostępu do zasobów systemowych. W projekcie zrealizowano zarówno tradycyjne logowanie z wykorzystaniem tokenów JWT, jak i logowanie przez konto Google (OAuth 2.0), co zapewnia łatwe uwierzytelnianie oraz efektywne zarządzanie sesją użytkownika, uwzględniając jego rolę w systemie (klient, pracownik, administrator) [5].
- Spring Data JPA – upraszcza interakcję z relacyjną bazą danych PostgreSQL, automatycznie generując zapytania SQL bazujące na metodach interfejsów repozytoriów. To podejście znacznie przyspiesza rozwój warstwy danych i ogranicza liczbę błędów [6].
- Spring Web (Spring MVC) – umożliwia tworzenie REST API z pełną kontrolą nad routowaniem, walidacją danych wejściowych oraz globalnym zarządzaniem błędami w aplikacji.

4.2. React (JavaScript)

W części dotyczącej frontendu użyto biblioteki React, opracowanej przez Meta (wcześniej Facebook), która zyskała status jednego z najczęściej wybieranych narzędzi do tworzenia nowoczesnych UI w aplikacjach internetowych. Jej kluczową zaletą jest komponentowa struktura, która pozwala na podział interfejsu na wielokrotnie używane, zamknięte komponenty. Taka metoda poprawia przejrzystość kodu, ułatwia testowanie i konserwację, a także sprzyja szybszemu wprowadzaniu nowych funkcji [7][8].

React funkcjonuje dzięki Virtual DOM – wewnętrznemu modelowi, który monitoruje zmiany w stanie aplikacji i zestawia je z wcześniejszą wersją DOM-u, aktualizując jedynie te części drzewa, które faktycznie uległy zmianie. Taka metoda redukuje operacje na rzeczywistym DOM i znacząco podnosi wydajność aplikacji typu SPA (Single Page Application) [7].

W projekcie React użyto do stworzenia wielu istotnych komponentów aplikacji frontendowej, takich jak:

- Formularze rejestracji i logowania – obsługujące dynamiczną weryfikację danych, logowanie za pomocą JWT oraz Google OAuth,
- Interfejs do zarządzania rezerwacjami – pozwalający użytkownikowi na przeglądanie, edytowanie, dodawanie oraz anulowanie rezerwacji,
- Panele użytkowników – widoki dostosowane do roli użytkownika (klient, pracownik, administrator) wyświetlane w zależności od warunków,
- Widoki usług oraz cenników – dynamicznie pobierane z API backendowego, zrealizowane przy użyciu React hooks (useState, useEffect),
- Responsywny układ – przystosowany do działania na komputerach stacjonarnych, tabletach oraz telefonach[7][8].

W celu zarządzania nawigacją w aplikacji zastosowano bibliotekę React Router, która umożliwia tworzenie systemu tras zgodnych z ideą SPA. Dzięki niej zmiany widoków odbywają się bez przeladowania strony, co zwiększa wygodę użytkowania aplikacji[9].

Główne trasy aplikacji to:

- /login – logowanie dla użytkowników,
- /register – tworzenie nowego konta,
- /client-panel – panel dostępny dla użytkownika,
- /admin-panel – panel dostępny dla administratora,
- /worker-panel – panel dostępny dla pracownika,
- /services – przegląd oferowanych usług w warsztacie.

React Router obsługuje również przekierowania, dynamiczne parametry w ścieżkach (takie jak /service/:id) oraz zarządzanie dostępem (na przykład PrivateRoute z logicznym systemem autoryzacji) [8].

4.3. Tailwind CSS

Do tworzenia interfejsu użytkownika wykorzystano Tailwind CSS - nowoczesny framework oparty na zasadzie utility-first, który charakteryzuje się tym, że klasy przypisywane są bezpośrednio w znacznikach HTML/JSX, zamiast tworzenia oddzielnych reguł CSS. Taki sposób pracy przyspiesza proces tworzenia i modyfikacji komponentów oraz upraszcza strukturę stylów.

Tailwind dysponuje bogatym zestawem klas narzędziowych, które pozwalają precyzyjnie określać różne aspekty wizualne, takie jak kolory, marginesy, padding, wielkości czcionek oraz zaokrąglenia. Wszystkie te parametry mogą być definiowane i zmieniane w pliku konfiguracyjnym, co zapewnia dużą elastyczność oraz spójność wizualną w całym projekcie.

W projekcie Tailwind został wykorzystany między innymi do:

- Tworzenia nowoczesnego i responsywnego interfejsu użytkownika, dostosowującego się do różnych rozmiarów ekranów,
- Budowy układów komponentów z zastosowaniem flexboxa oraz CSS Grid, co umożliwiło przejrzyste rozmieszczenie treści,
- Utrzymywania spójności typografii, kolorów, odstępów oraz kształtów bez konieczności tworzenia oddzielnych klas CSS,
- Szybkiego prototypowania widoków bez konieczności przechodzenia między plikami komponentów a arkuszami stylów.

Tailwind CSS okazał się bardzo pomocny przy wprowadzaniu zmian w projekcie - zmiana stylu danego komponentu zazwyczaj wymagała tylko zmiany jednej lub dwóch klas. Dzięki obszernej dokumentacji oraz aktywnej społeczności, framework jest prosty do nauki i wdrożenia, nawet dla osób początkujących w programowaniu [9].

4.4. PostgreSQL

Do przechowywania informacji w projekcie wykorzystano PostgreSQL – nowoczesny open source'owy system zarządzania bazą danych, który od dłuższego czasu uznawany jest za jedno z najbardziej solidnych i wszechstronnych rozwiązań w obszarze baz danych relacyjnych. PostgreSQL obsługuje standard SQL ANSI/ISO i pozwala na wiele dodatkowych rozszerzeń, które umożliwiają realizację skomplikowanych operacji wykraczających poza zdolności wielu innych systemów w tej kategorii [10].

System został stworzony z myślą o dużej zgodności z zasadami ACID (Atomiczność, Spójność, Izolacja, Trwałość), co oznacza, że każda zmiana danych jest albo w pełni przeprowadzona, albo nie jest realizowana wcale – zapewniając spójność i integralność danych, nawet w sytuacji awarii. Wbudowane mechanizmy zarządzania transakcjami (rollback, savepoint, poziom izolacji) umożliwiają dokładne nadzorowanie jednoczesnego dostępu wielu użytkowników, co w przypadku systemów z wieloma rolami – takich jak omawiany projekt – jest kluczowym elementem.

W ramach projektu PostgreSQL pełni funkcję głównego repozytorium danych dla:

- kont użytkowników (wraz z ich rolami i danymi personalnymi),
- informacji o dostępnych usługach warsztatu oraz ich cenach,

- kalendarza i historii rezerwacji, wraz z przypisanymi statusami realizacji,
- zdarzeń systemowych i śledzenia aktywności użytkowników.

PostgreSQL wspiera bogaty system typów danych, w tym typy geometryczne, UUID, XML, JSON/JSONB, co pozwala na przechowywanie nie tylko danych relacyjnych, ale również półstrukturalnych i dokumentowych. W kontekście projektu, JSONB wykorzystano m.in. do przechowywania ustawień aplikacyjnych oraz parametrów sesji użytkowników.

Dodatkowe funkcje PostgreSQL, które zostały użyte w aplikacji, to m. in. :

- indeksowanie kluczy podstawowych i obcych – co umożliwia szybkie znalezienie rekordów, np. przypisanych rezerwacji do określonego użytkownika lub usługi,
- wbudowane typy danych związane z czasem i datą – stosowane do zarządzania harmonogramami rezerwacji oraz analizy historii usług w określonym czasie,
- wsparcie dla typów UUID – używanych jako unikalne identyfikatory dla rekordów (np. użytkowników, usług, rezerwacji), co zwiększa bezpieczeństwo i prywatność danych,
- system ról i uprawnień – umożliwiający ograniczenie dostępu do danych tylko dla upoważnionych części aplikacji (np. backend API),
- proste widoki SQL – wykorzystane do logicznego łączenia danych z różnych tabel, co upraszcza proces odczytu (np. zestawienia rezerwacji z informacjami o kliencie i nazwą usługi).

Warto podkreślić, że PostgreSQL pozwala na prostą rozbudowę poziomą – dane mogą być dzielone na partie, a instancje bazy mogą być replikowane w celu zwiększenia zdolności odczytu. System można połączyć z narzędziami takimi jak pgAdmin (interfejs graficzny do zarządzania bazą) oraz PostGIS – rozszerzeniem do obsługi danych przestrzennych, co sprawia, że jest to wszechstronne narzędzie na przyszłość [11].

W warstwie integracyjnej projektu wykorzystano Spring Data JPA, który umożliwia mapowanie danych obiektowych (encje) na relacyjne struktury PostgreSQL, co pozwala programistom na deklaratywne operowanie na danych. Większość zapytań jest generowana automatycznie przez framework, a tam, gdzie potrzebna jest większa precyzja – wykorzystuje się zapytania natywne lub adnotacje JPQL.

PostgreSQL to rozwiązanie, które jest nie tylko bezpieczne i wydajne, lecz także bardzo skalowalne oraz przystosowane do przyszłych integracji, na przykład z mikroserwisami, środowiskami chmurowymi (AWS RDS, Azure PostgreSQL) oraz narzędziami do analizy i raportowania. Wybór tego systemu w omawianym projekcie był podyktowany zarówno funkcjonalnymi wymaganiami, jak i dostępnością doświadczonej pomocy, stabilnością oraz obszerną dokumentacją [12].

4.5. PayU – system płatności online

W systemie wprowadzono opcję dokonywania płatności za usługi warsztatowe, łącząc backend aplikacji z platformą PayU, która jest czołowym dostawcą płatności online w Polsce. Ta integracja znacznie ułatwia proces finalizacji rezerwacji i podnosi standard profesjonalizmu oraz komfortu korzystania z systemu, zwłaszcza z perspektywy użytkownika końcowego – klienta.

Po zalogowaniu się do aplikacji i wybraniu terminu oraz usługi, klient ma możliwość wybrania jednej z dwóch metod płatności:

- Płatność online (PayU) – natychmiastowe przekierowanie do systemu PayU, co umożliwia dokonanie płatności kartą.
- Płatność na miejscu – możliwość uregulowania należności bezpośrednio w warsztacie przy odbiorze pojazdu.

Integracja została przeprowadzona z użyciem REST API PayU – nowoczesnego systemu komunikacji opartego na protokole HTTPS, który umożliwia całkowitą automatyzację procesów płatności: od stworzenia zamówienia, przez skierowanie użytkownika do systemu płatności, aż po otrzymanie informacji o stanie transakcji. Po zakończeniu płatności, system automatycznie aktualizuje status rezerwacji, na przykład jako „opłacona” lub „anulowana”, w zależności od wyniku operacji.

Aby zapewnić pełną zgodność z aktualnymi standardami bezpieczeństwa, wszystkie transakcje płatnicze są ograniczone wyłącznie do platformy PayU. Aplikacja nie gromadzi żadnych danych finansowych użytkowników, takich jak numery kart, co jest zgodne z wytycznymi PCI DSS oraz przepisami unijnymi o ochronie danych osobowych (RODO).

4.6. ChatGPT (OpenAI API)

W celu zwiększenia wygody użytkownika oraz częściowej automatyzacji obsługi klienta, system został wzbogacony o funkcję chatbota opartego na sztucznej inteligencji. Rozwiązanie to wykorzystuje model językowy GPT-4 dostarczany przez firmę OpenAI, który został zintegrowany z aplikacją w celu doradzania klientom przy wyborze odpowiedniej usługi warsztatowej [16].

Z punktu widzenia użytkownika, działanie jest bardzo proste – klient opisuje objawy dotyczące pojazdu, np. „piszczy przy hamowaniu” albo „nie odpala rano”. Chatbot analizuje te informacje i sugeruje odpowiednie usługi, takie jak diagnostyka układu hamulcowego lub kontrola akumulatora i rozrusznika. Dzięki temu klient nie musi ręcznie przeszukiwać listy dostępnych usług ani kontaktować się z obsługą [17].

Technicznie rzecz biorąc, funkcja została oparta o OpenAI API. Aplikacja wysyła odpowiednio sformułowane zapytania z opisem problemu i ewentualną historią rozmowy. Odpowiedź zwracana przez model jest generowana na podstawie dużych zbiorów danych tekstowych i

dostosowana do potrzeb klienta. Backend systemu dba o bezpieczeństwo, m.in. chroniąc klucz API oraz ograniczając liczbę zapytań z jednego konta, by zapobiegać nadużyciom [18].

Po stronie frontendowej zaimplementowano prosty i przejrzysty interfejs czatu. Klient ma możliwość swobodnego wpisywania treści, a odpowiedzi pojawiają się dynamicznie. Dane przekazywane do modelu są ograniczone jedynie do treści rozmowy – nie są przesyłane żadne dane osobowe użytkownika, co pozwala na zachowanie zgodności z przepisami RODO [19].

Zastosowanie tego typu AI w systemie warsztatu samochodowego realnie wpływa na wygodę obsługi – użytkownik szybciej podejmuje decyzje, rzadziej wybiera niewłaściwe usługi, a pracownicy mogą skupić się na realizacji faktycznych zleceń.

5. Projekt systemu

Projekt interfejsu użytkownika stworzono zgodnie z nowoczesnymi zasadami UX/UI, koncentrując się na prostocie, przejrzystości i łatwości w obsłudze. Wykorzystano bibliotekę React, co pozwoliło na tworzenie dynamicznych aplikacji jednostronicowych, a także Tailwind CSS, który gwarantuje responsywny i jednolity wygląd elementów wizualnych.

Układ graficzny został rozdzielony na moduły, które odpowiadają trzem kluczowym funkcjom systemu:

- **Klient**

Dla użytkownika przygotowano panel (ClientPanel.tsx), który umożliwia korzystanie z kluczowych funkcji:

- przechodzenie do dostępnych terminów (AvailableDates.tsx),
- dokonywanie rezerwacji (NewReservation.tsx),
- wybór dostępnych usług (SelectService.tsx),
- zarządzanie swoją historią rezerwacji (UserReservation.tsx),
- aktualizację danych osobowych (ClientProfile.tsx).

Użytkownik ma także dostęp do ogólnodostępnych części aplikacji (About.tsx, Contact.tsx, ServicesList.tsx), które są widoczne bez logowania. Interfejs został zaprojektowany w sposób, który maksymalnie ułatwia użytkownikowi cały proces rezerwacji – od zapoznania się z oferowanymi usługami po dokonanie wyboru i płatność.

- **Pracownik**

Widoki przeznaczone dla pracownika warsztatu można znaleźć w katalogu worker. Obejmują one:

- panel do przeglądania oraz modyfikacji rezerwacji (WorkerPanel.tsx, ManageOrders.tsx),
- aktualizację statusu napraw (WorkerStatusUpdate.tsx, EditOrder.tsx),
- zarządzanie harmonogramem oraz przydzielaniem zadań (WorkerManageAppointments.tsx).

Interfejs został zoptymalizowany, aby umożliwić szybką pracę oraz częste aktualizacje danych, co pozwala pracownikowi na dostęp do ważnych informacji bez potrzeby przechodzenia przez wiele widoków.

- **Administrator**

Panel zarządzania administratorem (AdminPanel.tsx) obejmuje funkcje związane z administrowaniem systemem:

- dodawanie oraz modyfikowanie użytkowników (AddUser.tsx, EditUser.tsx, ManageUsers.tsx),
- zarządzanie usługami (ManageService.tsx),
- analizowanie statystyk systemowych (AdminStats.tsx),
- obsługa rezerwacji (ManageAppointments.tsx).

Te elementy zostały stworzone z myślą o wydajności i klarowności, co pozwala na pełne zarządzanie danymi systemu bez zbędnego obciążania interfejsu.

- **Części wspólne**

Nawigacja (Navbar. tsx) oraz stopka (Footer. tsx) zostały przeniesione do folderu ui. Użycie podejścia opartego na modułach pozwala na ponowne użycie tych komponentów w różnych obszarach aplikacji, co poprawia jednolitość i ułatwia zarządzanie kodem.

6. Implementacja systemu

W tym rozdziale omówiono, jak zrealizowane są kluczowe funkcje systemu z punktu widzenia osoby korzystającej z końcowych usług. Zwrócono uwagę na detale dotyczące interfejsu użytkownika, poszczególne etapy procesów biznesowych oraz współpracę z zewnętrznymi serwisami – w tym z systemem płatności PayU. Procesy te zostały zaprojektowane w taki sposób, aby umożliwić sprawne zarządzanie rezerwacjami zarówno dla klientów indywidualnych, jak i personelu warsztatu.

6.1. Panel klienta

Panel klienta stanowi istotny element systemu, który umożliwia użytkownikowi łatwe korzystanie z usług warsztatu oraz zarządzanie swoimi informacjami i wizytami. Proces ten został rozdzielony na kilka zasadniczych etapów, które przedstawiono poniżej.

6.1.1. Widok główny panelu klienta

Po zalogowaniu użytkownik jest przenoszony do panelu klienta (ClientPanel), gdzie dostępne są cztery kluczowe akcje:

- Utwórz nową rezerwację – prowadzi do procesu zakupu nowej wizyty,
- Moje rezerwacje – pozwala na przegląd i zarządzanie obecnymi oraz zrealizowanymi rezerwacjami,
- Zmień dane – przełącza do sekcji, w której można zaktualizować osobiste informacje klienta.
- Zdiagnozuj usterkę z AI – przekierowuje do wbudowanego chatbota, który na podstawie opisu problemu technicznego (np. dźwięki, objawy, lampki ostrzegawcze) sugeruje odpowiednią usługę warsztatową.

Układ interfejsu bazuje na dużych przyciskach akcji, co zwiększa dostępność i ułatwia nawigację.



Zrzut ekranu nr 1 – widok panelu klienta

6.1.2. Wybór terminu wizyty

Pierwszym krokiem przy tworzeniu rezerwacji jest wybór wolnego terminu (AvailableDates). Terminy są pobierane z backendu w czasie rzeczywistym i prezentowane w przejrzystej siatce. Każda pozycja zawiera dokładną datę i godzinę, a klient może kliknąć dowolny z dostępnych przedziałów. Mechanizm wyboru daty działa dynamicznie – po kliknięciu przycisku Dalej użytkownik przechodzi do widoku wyboru usług.

Nowa rezerwacja

Wybierz dostępny termin:

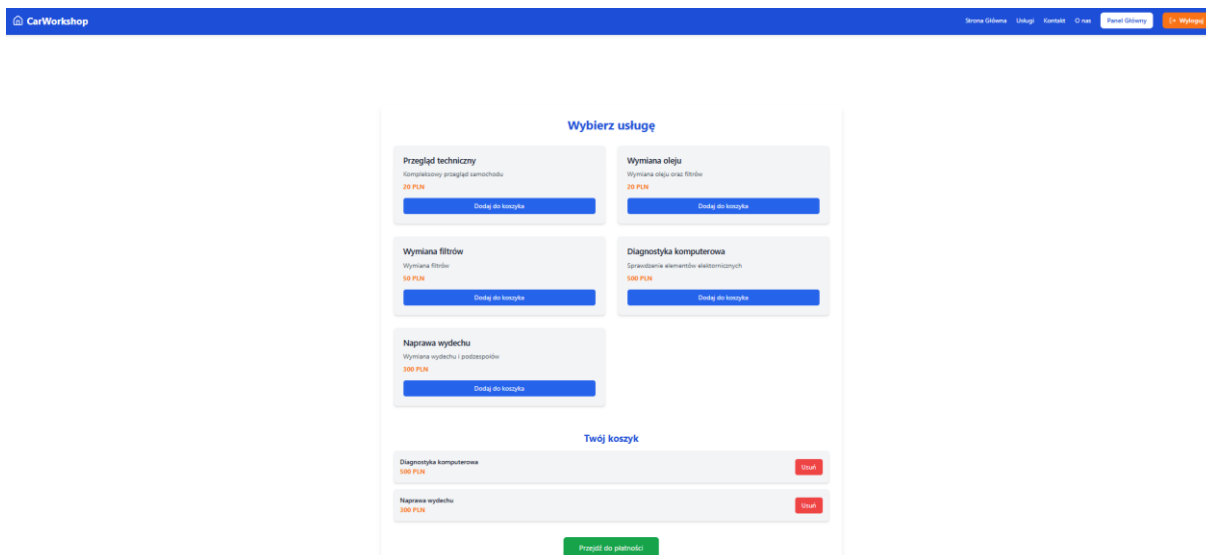
18.06.2025, 21:45:00	27.06.2025, 23:30:00
25.06.2025, 23:00:00	22.06.2025, 23:15:00

[Dalej](#)

Zrzut ekranu nr 2 – widok wyboru terminu usługi

6.1.3. Wybór usług warsztatowych

W kolejnym kroku klient wybiera jedną lub więcej usług, które chce wykonać podczas wizyty. Usługi są prezentowane w postaci responsywnych kart, z nazwą, krótkim opisem oraz ceną. Każda karta posiada przycisk „Dodaj do koszyka”, który dodaje usługę do listy aktualnego zamówienia.



Zrzut ekranu nr 3 – widok wyboru usługi

Widok ten umożliwia:

- wielokrotny wybór usług,
- podgląd łącznej wartości zamówienia,
- usuwanie pozycji z koszyka.

System zapewnia walidację – klient nie przejdzie do kolejnego kroku bez wyboru co najmniej jednej usługi.

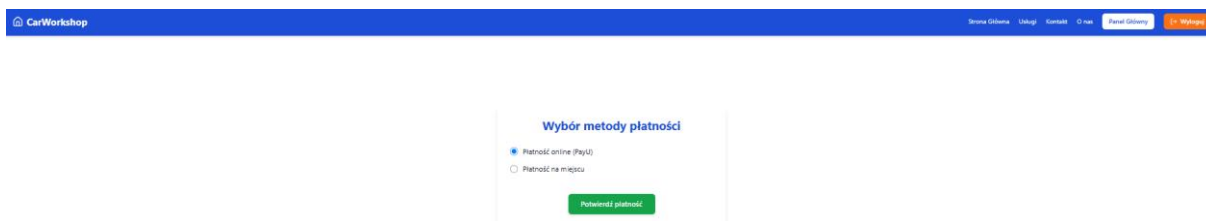
6.1.4. Podsumowanie koszyka i wybór metody płatności

Po zatwierdzeniu wyboru usług użytkownik przechodzi do ekranu wyboru metody płatności.

Dostępne są dwie opcje:

- Płatność online (PayU) – przekierowanie do bramki płatności elektronicznych,
- Płatność na miejscu – deklaracja zapłaty gotówką lub kartą w warsztacie.

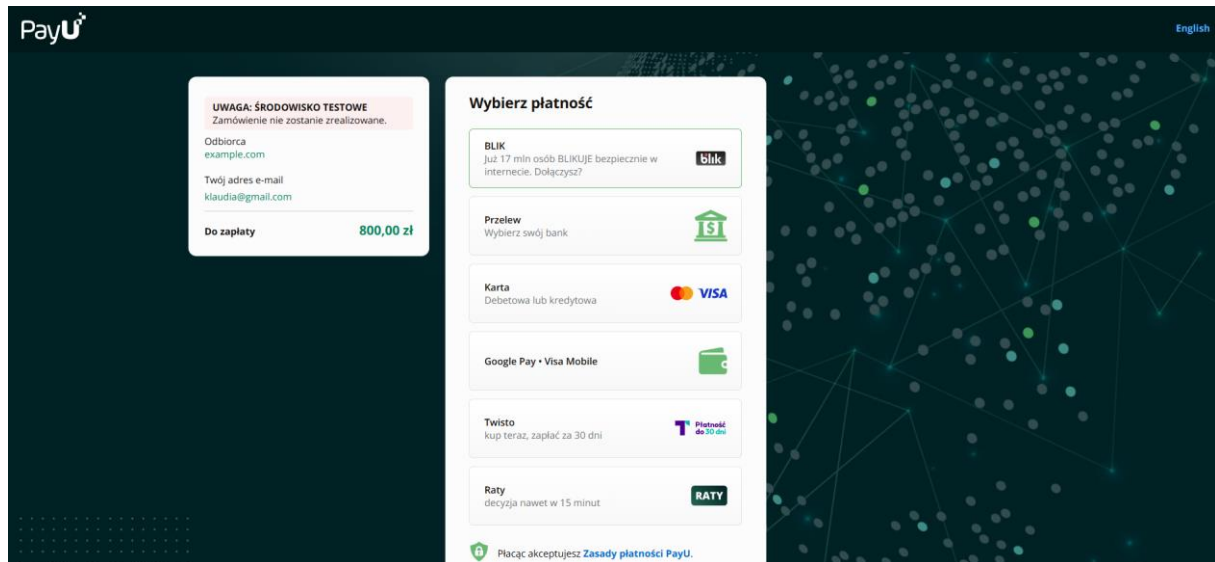
Domyślnie zaznaczona jest płatność online. Po kliknięciu Potwierdź płatność, klient zostaje przekierowany do bramki PayU.



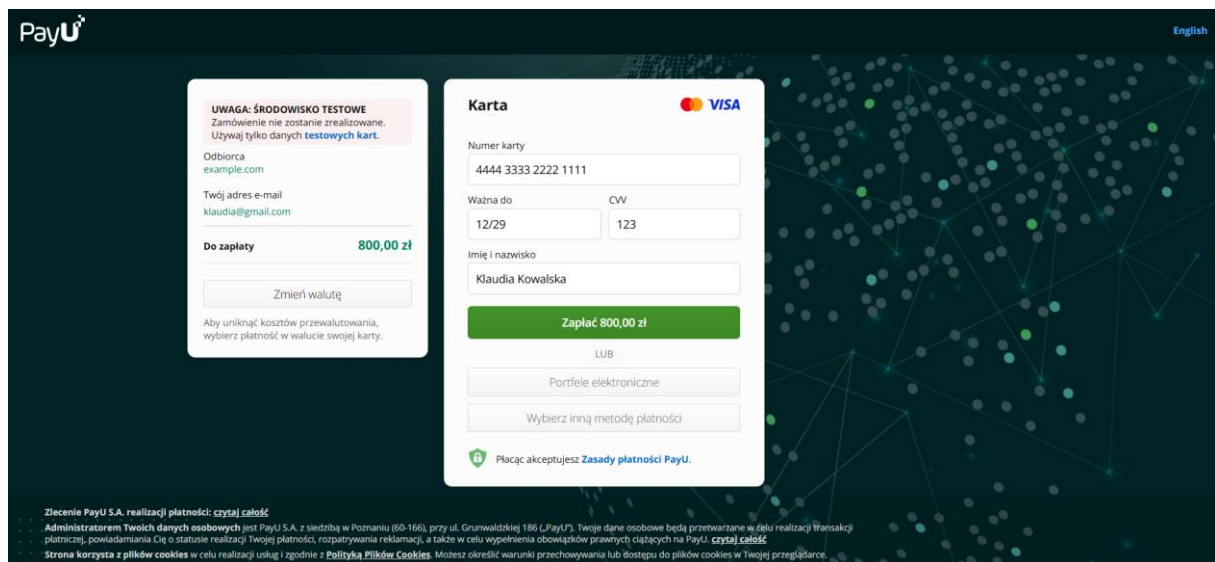
Zrzut ekranu nr 4 – widok wyboru systemu płatności

6.1.5. Proces płatności – integracja z PayU

Na ekranie PayU klient wybiera jedną z metod płatności: BLIK, przelew bankowy, karta, Google Pay, płatności odroczone lub ratalne. Po zakończeniu płatności, system odbiera odpowiedź z API PayU i aktualizuje status rezerwacji. W przypadku sukcesu rezerwacja zostaje oznaczona jako PAID, a użytkownik wraca do systemu z potwierdzeniem. Dla płatności na miejscu – rezerwacja pozostaje ze statusem UNPAID.



Zrzut ekranu nr 5 – widok przekierowania do systemu płatności PayU



Zrzut ekranu nr 6 – widok wprowadzania danych karty płatniczej

Metoda createOrder odpowiada za cały proces tworzenia nowego zamówienia przez klienta. Najpierw system sprawdza, czy użytkownik o podanym loginie istnieje w bazie. Jeśli nie – zgłasza błąd. Następnie pobierany jest wybrany przez klienta termin wizyty. Jeśli taki termin nie istnieje albo jest już zarezerwowany, system przerywa dalsze działanie i informuje o problemie. W przypadku dostępnego terminu zostaje on oznaczony jako zajęty i przypisany do danego

użytkownika. Kolejny krok to pobranie usług, które klient wybrał – na podstawie ich identyfikatorów. Po zebraniu wszystkich potrzebnych informacji tworzony jest obiekt Order, w którym zapisywane są dane klienta, termin, wybrane usługi, sposób płatności oraz statusy – naprawy i płatności. Jeżeli klient wybrał płatność gotówką, zamówienie zostaje oznaczone jako nieopłacone; w pozostałych przypadkach traktowane jest jako opłacone od razu. Na końcu całe zamówienie jest zapisywane w bazie danych i gotowe do dalszej obsługi przez pracowników warsztatu. Cała metoda odzwierciedla realny scenariusz umawiania wizyty przez klienta i tworzy kompletne zlecenie serwisowe.

```
2 usages
30 public Order createOrder(String username, List<Long> serviceIds, Long appointmentId, String paymentMethod) {
31     User user = userRepository.findByUsername(username)
32         .orElseThrow(() -> new UsernameNotFoundException("User not found"));
33
34     AvailableDate availableDate = availableDateRepository.findById(appointmentId)
35         .orElseThrow(() -> new RuntimeException("Termin nie istnieje"));
36
37     if (availableDate.isReserved()) {
38         throw new IllegalArgumentException("Termin już zajęty");
39     }
40
41     availableDate.setReserved(true);
42     availableDate.setUser(user);
43     availableDateRepository.save(availableDate);
44
45     List<WorkshopService> services = serviceRepository.findAllById(serviceIds);
46
47     Order order = new Order();
48     order.setUser(user);
49     order.setAvailableDate(availableDate);
50     order.setServices(services);
51     order.setPaymentMethod(paymentMethod);
52     order.setRepairStatus(RepairStatus.PENDING);
53
54     if ("cash".equalsIgnoreCase(paymentMethod)) {
55         order.setPaymentStatus(PaymentStatus.UNPAID);
56     } else {
57         order.setPaymentStatus(PaymentStatus.PAID);
58     }
59
60     return orderRepository.save(order);
61 }
62
63 1 usage
64 public List<Order> getUserOrders(String username) {
65     User user = userRepository.findByUsername(username)
66         .orElseThrow(() -> new RuntimeException("Użytkownik nie znaleziony"));
67     return orderRepository.findByUser(user);
68 }
```

Zrzut ekranu nr 7 – kod źródłowy dla funkcji tworzenia nowego zamówienia

6.1.6. Podgląd i zarządzanie rezerwacjami

Widok rezerwacji klienta pełni kluczową funkcję w całym systemie, ponieważ umożliwia użytkownikowi kontrolę nad historią i statusem jego wizyt serwisowych. Po wejściu do zakładki „Moje rezerwacje” klient widzi czytelnie przedstawione karty wszystkich utworzonych rezerwacji – zarówno tych oczekujących, jak i zrealizowanych czy anulowanych.

Na każdej karcie znajdują się:

- numer rezerwacji,
- wybrane usługi,
- termin wizyty
- status płatności (np. „Opłacone”, „Nieopłacone”),
- status naprawy (np. „W trakcie naprawy”, „Zakończone”, „Oczekujące”, „Anulowane”).

Całość została zaprojektowana w sposób przejrzysty i intuicyjny – zastosowano kolorowe przyciski oraz ikony, które pozwalają szybko zidentyfikować stan danej rezerwacji. Dzięki temu użytkownik od razu wie, które wizyty są aktywne i jaka jest ich aktualna sytuacja.



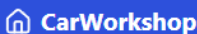
Zrzut ekranu nr 8 – widok rezerwacji użytkownika

W przypadku potrzeby modyfikacji rezerwacji (np. zmiana daty lub usług), użytkownik korzysta z przycisku „Edytuj”, który przekierowuje go do formularza kontaktowego. Tam należy podać swoje dane, numer rezerwacji oraz opisać, czego dotyczy zmiana. Formularz ten wysyła wiadomość e-mail bezpośrednio do pracownika warsztatu, na adres wskazany w zakładce „O nas”.


Z perspektywy zarządzania systemem:

- Edytowanie rezerwacji jest dostępne wyłącznie poprzez kontakt z pracownikiem.
- Bezpieczeństwo i zgodność danych zapewnia fakt, że użytkownik nie zmienia danych bezpośrednio w systemie, a jedynie inicjuje prośbę.
- Po odebraniu wiadomości przez pracownika, możliwe jest ręczne wprowadzenie zmian z poziomu interfejsu pracowniczego.

Takie rozwiązanie nie tylko ogranicza ryzyko błędów, ale także daje większą kontrolę pracownikom nad harmonogramem wizyt.

Strona GłównaUsługiKontaktO nasPanel GłównyWyloguj

Kontakt z warsztatem

 **Edycja rezerwacji:**
Zmiana terminu wizyty jest możliwa **wyłącznie** poprzez kontakt z warsztatem. Prosimy opisać, czego dotyczy zmiana i podać numer rezerwacji.

Adres warsztatu:
AutoWorkshop Sp. z o.o.
ul. Mechaników 123
00-123 Warszawa

Telefon:
+48 123 456 789

Email:
carworkshop1289@gmail.com

Godziny otwarcia:
Poniedziałek - Piątek: 8:00 - 18:00
Sobota: 9:00 - 14:00
Niedziela: nieczynne

Formularz kontaktowy:

Imię i nazwisko

Adres email

Edycja rezerwacji

Treść wiadomości

Wyślij wiadomość

© 2025 CarWorkshop. Wszelkie prawa zastrzeżone.UsługiKontaktO nas

Zrzut ekranu nr 9 – Formularz kontaktowy do edycji rezerwacji

6.1.7. Edycja danych klienta

Funkcja modyfikacji danych osobowych jest dostępna bezpośrednio w panelu klienta – wystarczy nacisnąć przycisk „Zmiana danych”. Daje to możliwość zaktualizowania ważnych informacji o koncie użytkownika, które mogą się zmieniać podczas korzystania z platformy. Formularz do edycji został stworzony z myślą o jasności oraz łatwości obsługi, aby nawet mniej zaawansowani użytkownicy mogli go bez problemu wypełnić.

Formularz zawiera następujące sekcje:

- Nazwa użytkownika – unikalny login, który identyfikuje osobę w systemie i może być zmieniony, zachowując swoją wyjątkowość,
- Email – adres e-mail, który jest wykorzystywany również do komunikacji i powiadomień,
- Nowe hasło (opcjonalnie) – użytkownik ma możliwość nie wypełniania tego pola, jeśli nie pragnie zmieniać swojego aktualnego hasła,
- Imię i nazwisko – dane osobowe, które identyfikują klienta,
- Telefon – kontakt, z którego korzystają pracownicy serwisu przy ustalaniu spraw związanych z usługami lub terminami,
- Adres – opcjonalna sekcja przeznaczona na dane o miejscu zamieszkania klienta.

Po wprowadzeniu zmian użytkownik ma możliwość potwierdzenia danych poprzez kliknięcie przycisku „Zapisz zmiany”. System sprawdza poprawność informacji oraz przeprowadza odpowiednią weryfikację, na przykład upewnia się, że format adresu e-mail jest prawidłowy. Po dokonaniu zatwierdzenia, informacje są zaktualizowane w bazie danych, a użytkownik otrzymuje informację o pomyślnym zakończeniu operacji.

Funkcjonalność ta została zrealizowana z wykorzystaniem technologii React po stronie klienta (komponent ClientProfile. tsx) oraz przy pomocy Spring Boot i Spring Security po stronie serwera, gdzie dane są aktualizowane poprzez specjalnie opracowany endpoint w kontrolerze ClientController. Wymiana informacji między frontendem a backendem następuje dzięki zapytaniom HTTP w typie PUT, zgodnie z konwencją REST do aktualizacji danych. System zawiera również zabezpieczenia, które pozwalają na edytowanie wyłącznie danych związanych z aktualnie zalogowanym użytkownikiem – jest to możliwe dzięki uwierzytelnieniu przy użyciu JWT (JSON Web Token). Całość zapewnia bezpieczeństwo transakcji i ochronę danych osobowych w zgodzie z najlepszymi praktykami.

Zaktualizuj dane

Nazwa użytkownika

Email

Nowe hasło (opcjonalnie)

Imię

Nazwisko

Telefon

Adres

Zapisz zmiany

Zrzut ekranu nr 10 – widok edycji

Metoda `updateClientProfile` ma na celu zaktualizowanie informacji o użytkowniku przez samego klienta, który jest zalogowany w systemie. Na początku system stara się znaleźć konto w bazie danych, opierając się na podanym loginie. W przypadku braku użytkownika, generowany jest wyjątek sygnalizujący błąd 404 – co oznacza, że użytkownik nie został odnaleziony. Kiedy konto zostaje zidentyfikowane, system dostosowuje jego dane zgodnie z informacjami zawartymi w formularzu – czyli nazwą użytkownika, adresem e-mail, imieniem, nazwiskiem, numerem telefonu oraz adresem. Ponadto, jeśli klient wprowadzi nowe hasło (i pole nie jest puste), to hasło zostanie zaszyfrowane i zaktualizowane w systemie. Na końcu zaktualizowane informacje o użytkowniku są zapisywane w bazie. Cała procedura umożliwia klientowi samodzielne modyfikowanie swojego profilu w sposób bezpieczny i prosty.

```

107 @
108 public User updateClientProfile(String username, UpdateClientRequest request) {
109     User user = userRepository.findByUsername(username)
110         .orElseThrow(() -> new ResponseStatusException(HttpStatus.NOT_FOUND, "User not found"));
111
112     user.setUsername(request.getUsername());
113     user.setEmail(request.getEmail());
114     user.setFirstName(request.getFirstName());
115     user.setLastName(request.getLastName());
116     user.setPhoneNumber(request.getPhoneNumber());
117     user.setAddress(request.getAddress());
118
119     if (request.getPassword() != null && !request.getPassword().isBlank()) {
120         user.setPassword(passwordEncoder.encode(request.getPassword()));
121     }
122
123     return userRepository.save(user);
124 }

```

Zrzut ekranu nr 11 – kod źródłowy funkcji na edycję danych użytkownika

6.1.8. Diagnozowanie usterki za pomocą AI


W sekcji klienta dodano nową praktyczną opcję, która znacznie ułatwia używanie aplikacji – możliwość zidentyfikowania problemu z pojazdem przy pomocy sztucznej inteligencji. Działa to w prosty sposób: użytkownik klika przycisk „Zdiagnozuj usterkę z AI”, wpisuje w formularzu, co się dzieje z jego samochodem (na przykład „zapaliła mi się kontrolka silnika”), a następnie otrzymuje wskazówkę na temat usługi, którą powinien wybrać – może to być na przykład diagnostyka komputerowa.

Funkcja ta wykorzystuje model językowy GPT-4, który potrafi rozumieć naturalny język i w trafny sposób dopasować opisaną usterkę do odpowiedniej usługi. Dzięki temu użytkownik nie musi przejrzeć wszystkich dostępnych opcji – system sam sugeruje najodpowiedniejsze rozwiązanie.

To rozwiązanie oferuje kilka istotnych korzyści:


- Oszczędza czas klienta, ponieważ nie musi zgadywać, co wybrać,
- Zmniejsza ryzyko rezerwacji niewłaściwej usługi,
- Ułatwia pracę pracownikom warsztatu, gdyż część prostych pytań jest obsługiwana automatycznie.

Cały proces przebiega bezpiecznie i zgodnie z zasadami RODO – system nie zbiera danych osobowych, a jedynie treść wprowadzoną przez użytkownika. W przyszłości tę funkcję można by jeszcze rozwijać, na przykład o możliwość przeprowadzania rozmowy etapami czy uwzględnienie historii poprzednich napraw.

 **Diagnostuj problem z pojazdem**

Opis problemu

zapałała mi się kontrolka silnika

 Zdiagnozuj usterkę AI

Sugerowana usługa:

Diagnostyka komputerowa.

Zrzut ekranu nr 12 – widok wspomagania wyboru usługi przez AI

Za integrację systemu z modelem GPT odpowiedzialna jest klasa GptService. Jej rolą jest przekształcenie opisu problemu dostarczonego przez klienta w zapytanie, które następnie jest wysyłane do API OpenAI. Klasa automatycznie zbiera z bazy informacje o dostępnych usługach, dołącza je do treści zapytania i formułuje prośbę o wskazanie najbardziej adekwatnej usługi. Odpowiedź z API zawiera jedynie nazwę usługi, którą system przekazuje użytkownikowi. Dzięki temu klient nie musi znać technicznych terminów – wystarczy, że opíše swój problem, a, aplikacja sama podpowie odpowiednią opcję, wspierając tym samym cały proces rezerwacji.

```

1 package com.workshopapp.workshopservice.service;
2
3 import com.workshopapp.workshopservice.model.WorkshopService;
4 import com.workshopapp.workshopservice.repository.ServiceRepository;
5 import org.springframework.beans.factory.annotation.Value;
6 import org.springframework.http.*;
7 import org.springframework.stereotype.Service;
8 import org.springframework.web.client.RestTemplate;
9
10 import java.util.HashMap;
11 import java.util.List;
12 import java.util.Map;
13 import java.util.stream.Collectors;
14
15 @Service
16 public class GptService {
17
18     1 usage
19     @Value("${openai.api.key}")
20     private String openaiApiKey;
21
22     1 usage
23     private final String OPENAI_URL = "https://api.openai.com/v1/chat/completions";
24
25     2 usages
26     private final RestTemplate restTemplate;
27
28     2 usages
29     private final ServiceRepository serviceRepository;
30
31     no usages
32     public GptService(RestTemplate restTemplate, ServiceRepository serviceRepository) {
33         this.restTemplate = restTemplate;
34         this.serviceRepository = serviceRepository;
35     }
36
37 }

```

Zrzut ekranu nr 13 – widok klasy realizującej funkcje diagnozowania usterki przez AI cz. I

```

31 1 usage
32 public String getSuggestedService(String description) {
33     List<WorkshopService> services = serviceRepository.findByAvailableTrue();
34
35     String serviceList = services.stream().map(s -> s.getName() + ": " + s.getDescription()).collect(Collectors.joining("\n"));
36
37     String prompt = """
38         Jesteś doradcą w warsztacie samochodowym.
39         Na podstawie poniższej listy usług i opisu problemu klienta, zaproponuj jedną najlepiej pasującą usługę (tylko nazwę z listy):
40
41         Lista usług:
42         %s
43
44         Opis klienta:
45         %s
46
47         Odpowiedź sformatuj tylko jako nazwę jednej usługi.
48         """.formatted(serviceList, description);
49
50     HttpHeaders headers = new HttpHeaders();
51     headers.setContentType(MediaType.APPLICATION_JSON);
52     headers.setBearerAuth(openaiApiKey);
53
54     Map<String, Object> body = new HashMap<>();
55     body.put("model", "gpt-3.5-turbo");
56     body.put("messages", List.of(
57         Map.of("role", "system", "content", "Jesteś doradcą w warsztacie samochodowym."),
58         Map.of("role", "user", "content", prompt)
59     ));
60
61     HttpEntity<Map<String, Object>> request = new HttpEntity<>(body, headers);
62     ResponseEntity<Map> response = restTemplate.postForEntity(OPENAI_URL, request, Map.class);
63
64     List<Map<String, Object>> choices = ((List<Map<String, Object>>) response.getBody()).get("choices");
65     Map<String, Object> message = (Map<String, Object>) choices.get(0).get("message");
66     return message.get("content").toString().trim();
67 }
68 }

```

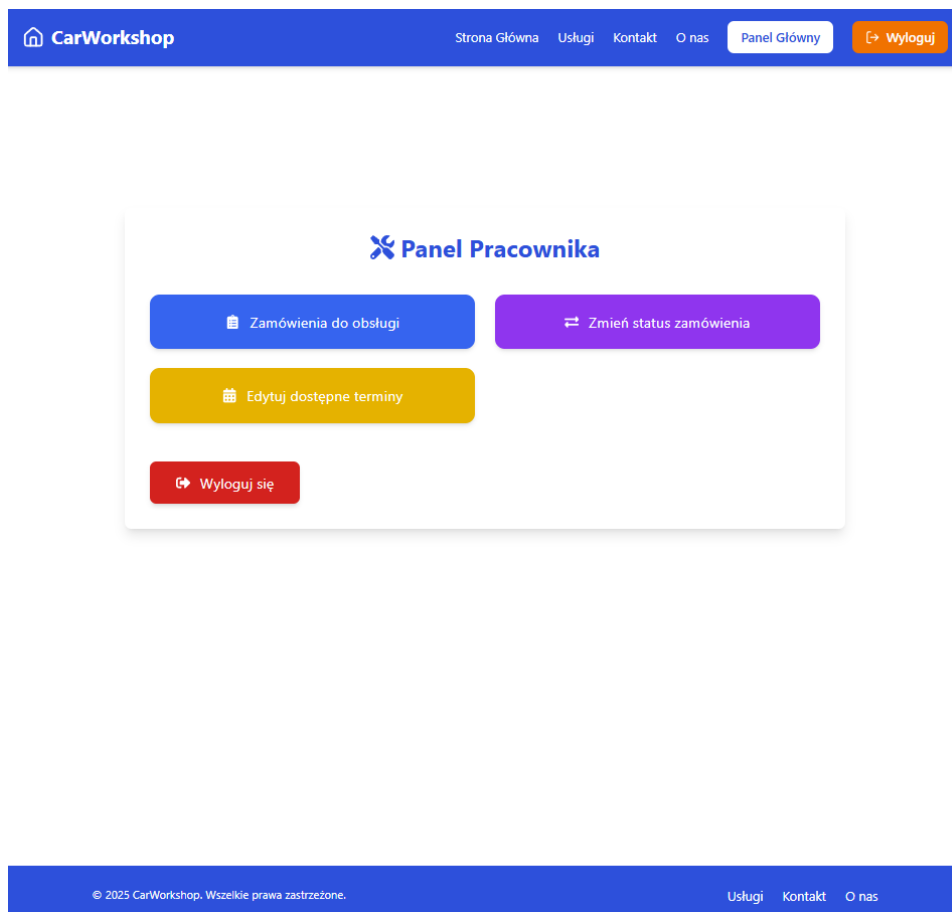
Zrzut ekranu nr 14 – widok klasy realizującej funkcje diagnozowania usterki przez AI cz. II

6.2. Panel pracownika

Panel dla pracowników to przestrzeń, w której zatrudniony w warsztacie może kontrolować swoje codzienne obowiązki. Wszystko zostało zorganizowane w jednym, łatwym do zrozumienia widoku — co pozwala na szybki i komfortowy dostęp do kluczowych funkcji. Po zalogowaniu się pracownik znajduje się na głównym ekranie panelu, gdzie ma do wyboru cztery podstawowe opcje:

- przegląd zamówień czekających na realizację,
- możliwość aktualizacji statusu napraw,
- zmianę dostępnych terminów dla klientów,
- oraz opcję wylogowania.






















Każda z tych opcji została stworzona z myślą o uproszczeniu codziennych zadań mechaników i personelu obsługi. W kolejnych punktach szczegółowo opisano wszystkie funkcje, jakie oferuje panel dla pracowników.



Zrzut ekranu nr 15 – widok panelu pracownika


6.2.1. Zamówienia do obsługi


Po zalogowaniu się do systemu pracowniczego, pierwszym działaniem, jakie można podjąć, jest przegląd wszystkich bieżących zamówień złożonych przez klientów. Można je znaleźć pod przyciskiem „Zamówienia do obsługi”. Tabela, która się pojawia, zawiera wszystkie istotne dane: kto złożył zamówienie, na kiedy ustalono wizytę, jakie usługi zostały wybrane, czy płatność została już zrealizowana oraz jaki jest obecny stan naprawy (na przykład: oczekuje, w trakcie, zakończone, anulowane). Obok każdego zamówienia znajdują się dwa przyciski: żółty „Edytuj”, który umożliwia modyfikację rezerwacji, oraz czerwony „Usuń”, który pozwala na całkowite skasowanie danego zamówienia. Dzięki temu pracownik ma szybki dostęp do kompletu usług, którymi musi zająć się w danym dniu lub w nadchodzących terminach.

 Zamówienia do obsługi							
ID	Klient	Data	Usługi	Metoda	Status naprawy	Status płatności	Akcje
55	Marcin Tutaj	19.06.2025, 17:15	Wymiana oleju, Wymiana filtrów, Diagnostyka komputerowa	Gotówka	Zakończone	Opłacone	 Edytuj  Usuń
60	Klaudia Kowalska	27.06.2025, 21:30	Diagnostyka komputerowa	Gotówka	Oczekujące	Nieopłacone	 Edytuj  Usuń
57	Klaudia Kowalska	18.06.2025, 18:45	Wymiana oleju, Wymiana filtrów, Diagnostyka komputerowa	Online	W trakcie naprawy	Opłacone	 Edytuj  Usuń
59	Klaudia Kowalska	22.06.2025, 21:15	Diagnostyka komputerowa, Wymiana amortyzatorów (komplet)	Gotówka	Zakończone	Nieopłacone	 Edytuj  Usuń
58	Klaudia Kowalska	18.06.2025, 19:45	Diagnostyka komputerowa, Naprawa wydechu	Online	Anulowane	Opłacone	 Edytuj  Usuń
54	Krzysztof Jarzyna	25.06.2025, 22:00	Diagnostyka komputerowa	Online	Oczekujące	Opłacone	 Edytuj  Usuń
56	Marcin Tutaj	27.06.2025, 22:00	Wymiana oleju, Wymiana filtrów, Diagnostyka komputerowa, Przegląd techniczny	Online	W trakcie naprawy	Opłacone	 Edytuj  Usuń
50	Klaudia Kowalska	18.06.2025, 20:15	Diagnostyka komputerowa	Gotówka	Oczekujące	Nieopłacone	 Edytuj  Usuń
51	Krzysztof Jarzyna	18.06.2025, 14:45	Diagnostyka komputerowa	Gotówka	Zakończone	Opłacone	 Edytuj  Usuń
52	Krzysztof Jarzyna	18.06.2025, 10:00	Diagnostyka komputerowa	Gotówka	Anulowane	Nieopłacone	 Edytuj  Usuń


Zrzut ekranu nr 16 – widok zamówień do obsługi

Po naciśnięciu przycisku „Edytuj” przy wybranym zamówieniu, pracownik otwiera formularz, w którym ma możliwość modyfikacji szczegółów danej wizyty. Na górze wyświetla się lista usług, które można zaznaczyć lub odznaczyć, w zależności od rzeczywistych potrzeb klienta. Wartości umieszczone przy usługach ułatwiają natychmiastową ocenę całkowitego kosztu. Poniżej znajduje się menu rozwijane z dostępnymi terminami – pracownik ma możliwość zmiany daty lub godziny wizyty w razie potrzeby (na przykład po rozmowie z klientem). Na koniec wystarczy kliknąć „Zapisz zmiany”, aby wprowadzone aktualizacje zostały zachowane. Przegląd ten jest prosty, jasny i łatwy w obsłudze – umożliwia sprawne zarządzanie wszystkimi zleceniami bez konieczności przechodzenia przez zbędne etapy.


 **Edytuj zamówienie #55**

 **Wybierz usługi:**

☐ Przegląd techniczny (150.00 zł)
☐ Wymiana oleju i filtrów (250.00 zł)
☐ Diagnostyka komputerowa (100.00 zł)
☐ Naprawa układu wydechowego (350.00 zł)
☐ Wymiana klocków hamulcowych (oś przednia) (180.00 zł)
☐ Wymiana tarcz i klocków hamulcowych (komplet) (450.00 zł)
☐ Serwis klimatyzacji (220.00 zł)
☐ Wymiana świec zapłonowych (100.00 zł)
☐ Wymiana akumulatora (80.00 zł)
☐ Wymiana paska rozrządu (850.00 zł)
☐ Wymiana amortyzatorów (komplet) (600.00 zł)
☐ Wymiana płynu hamulcowego (90.00 zł)
☐ Wymiana żarówek (30.00 zł)
☐ Geometria kół (180.00 zł)
☐ Wymiana opon (z wyważeniem) (120.00 zł)
☐ Czyszczenie DPF/FAP (500.00 zł)
☐ Regeneracja alternatora (400.00 zł)
☐ Przegląd zawieszenia (130.00 zł)
☐ Odgrzybianie klimatyzacji (100.00 zł)
☒ Wymiana oleju (20.00 zł)
☒ Wymiana filtrów (50.00 zł)
☒ Diagnostyka komputerowa (500.00 zł)
☐ Naprawa wydechu (300.00 zł)
☐ Przegląd techniczny (20.00 zł)

 **Wybierz termin:**

25.06.2025, 23:00:00

 **Zapisz zmiany**

Zrzut ekranu nr 17 – widok edycji zamówienia

6.2.2. Edytuj dostępne terminy

Kolejną istotną funkcją, z której mogą korzystać pracownicy, jest opcja ręcznego aktualizowania statusu każdej rezerwacji. Po kliknięciu na przycisk „Zmień status zamówienia” w głównym panelu, użytkownik jest przenoszony do tabeli, w której znajdują się wszystkie złożone zamówienia – podobnie jak w poprzednim widoku. W tabeli, obok podstawowych informacji, takich jak identyfikator rezerwacji, imienia i nazwiska klienta, lista zamówionych usług czy aktualne statusy napraw i płatności, dane są wzbogacone o rozwijane listy, które umożliwiają pracownikom szybkie aktualizowanie tych informacji. W kolumnie „Zmień Naprawę” personel ma

możliwość zmiany etapu realizacji naprawy – dostępne są opcje takie jak „Oczekujące”, „W trakcie”, „Zakończone” oraz „Anulowane”. Z kolei w kolumnie „Zmień Płatność” można łatwo zaznaczyć, czy klient opłacił wizytę.

Aktualizacja Statusów						
ID	Klient	Usługi	Naprawa	Płatność	Zmień Naprawę	Zmień Płatność
55	Marcin Tutaj	Wymiana oleju, Wymiana filtrów, Diagnostyka komputerowa	Zakończone	Opłacone	Zakończone ▼	Opłacone ▼
60	Klaudia Kowalska	Diagnostyka komputerowa	Oczekujące	Nieopłacone	Oczekujące ▼	Nieopłacone ▼
57	Klaudia Kowalska	Wymiana oleju, Wymiana filtrów, Diagnostyka komputerowa	W trakcie	Opłacone	W trakcie ▼	Opłacone ▼
59	Klaudia Kowalska	Diagnostyka komputerowa, Wymiana amortyzatorów (komplet)	Zakończone	Nieopłacone	Zakończone ▼	Nieopłacone ▼
58	Klaudia Kowalska	Diagnostyka komputerowa, Naprawa wydechu	Anulowane	Opłacone	Anulowane ▼	Opłacone ▼
54	Krzysztof Jarzyna	Diagnostyka komputerowa	Oczekujące	Opłacone	Oczekujące ▼	Opłacone ▼
56	Marcin Tutaj	Wymiana oleju, Wymiana filtrów, Diagnostyka komputerowa, Przegląd techniczny	W trakcie	Opłacone	W trakcie ▼	Opłacone ▼
50	Klaudia Kowalska	Diagnostyka komputerowa	Oczekujące	Nieopłacone	Oczekujące ▼	Nieopłacone ▼
51	Krzysztof Jarzyna	Diagnostyka komputerowa	Zakończone	Opłacone	Zakończone ▼	Opłacone ▼
52	Krzysztof Jarzyna	Diagnostyka komputerowa	Anulowane	Nieopłacone	Anulowane ▼	Nieopłacone ▼

Zrzut ekranu nr 18 – widok aktualizacji statusów zamówień

Metoda `updateRepairStatus` pozwala pracownikowi warsztatu zaktualizować status naprawy dla konkretnego zamówienia. Na początku system próbuje odnaleźć zamówienie po jego identyfikatorze – jeśli takie nie istnieje, zgłaszany jest błąd. Gdy zamówienie zostanie znalezione, odczytywany jest nowy status przesłany w żądaniu, przekształcany na wartość enum (czyli jedną z dopuszczalnych opcji, np. „W TRAKCIE” czy „ZAKOŃCZONE”), a następnie przypisywany do danego zlecenia. Na końcu zmiana zostaje zapisana w bazie danych. Dzięki tej metodzie pracownicy mogą na bieżąco informować klientów o aktualnym stanie realizacji usługi.

```

1 usage
92 @ public void updateRepairStatus(Long orderId, UpdateRepairStatusRequest request) {
93     Order order = orderRepository.findById(orderId)
94         .orElseThrow(() -> new RuntimeException("Order not found"));
95
96     RepairStatus status = RepairStatus.valueOf(request.getRepairStatus().toUpperCase());
97     order.setRepairStatus(status);
98     orderRepository.save(order);
99 }
100

```

Zrzut ekranu nr 19 – kod źródłowy funkcji na aktualizację statusu naprawy dla danego zamówienia

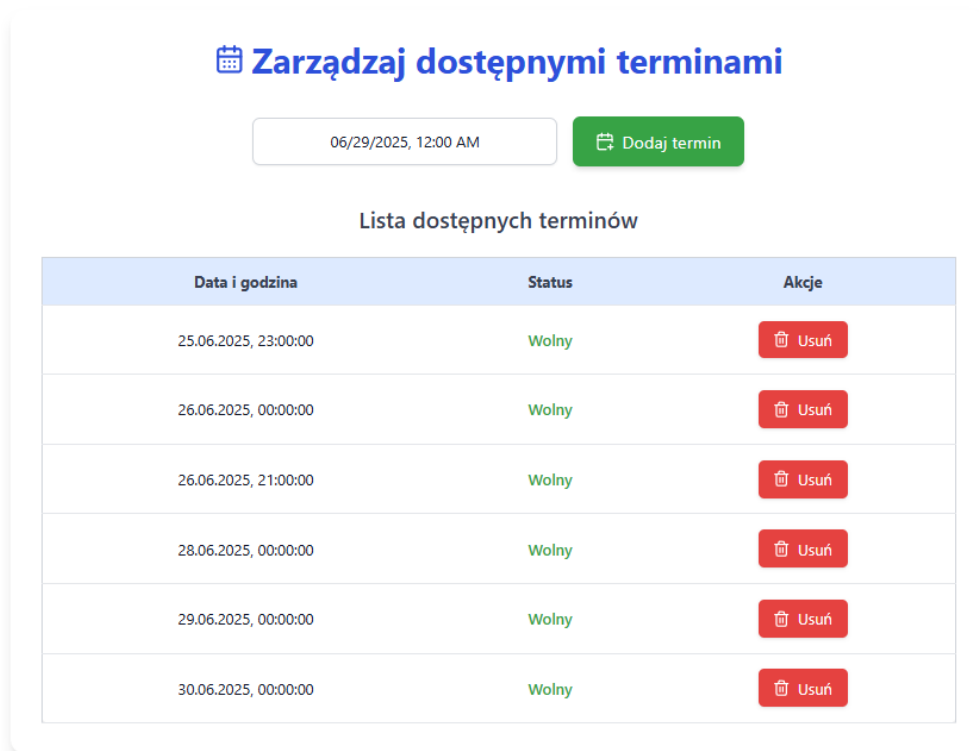
6.2.3. Edytuj dostępne terminy

Ostatnia sekcja w panelu dla pracowników to miejsce, w którym można dodawać oraz usuwać dostępne terminy wizyt. Tutaj określa się, w jakich godzinach klienci będą mieli możliwość rezerwacji wizyt w warsztacie. Na górze umieszczono pole wyboru – klikając je, rozwinie się przejrzysty kalendarz, w którym można wybrać odpowiednią datę oraz godzinę. Po wybraniu terminu wystarczy nacisnąć „Dodaj termin”, a nowa pozycja od razu pojawi się na niższej liście. Poniżej znajduje się tabela z obecnie dostępnymi terminami. Obok każdego terminu widoczna jest konkretna data oraz godzina, a także informacja, że termin jest „Wolny”. Jeśli któryś z terminów jest już zbędny, można go usunąć w dowolnym momencie, naciskając przycisk „Usuń”.

Widok ma kilka istotnych ograniczeń:

- nie ma możliwości dodania terminu wstecznego – system automatycznie uniemożliwia ustawienie daty, która już minęła, co zapobiega błędom.
- Przy próbie dodania terminu, który już istnieje, pojawi się komunikat informujący o podwójnym rekordzie – to zapobiega tworzeniu duplikatów.

Całość została stworzona z myślą o prostocie i funkcjonalności, aby pracownik mógł szybko zarządzać dostępnością bez potrzeby zajmowania się bazą danych ani panelem administratora.



Zrzut ekranu nr 20 – widok panelu zarządzania dostępnymi terminami

6.3. Panel administratora

Panel administratora to przestrzeń, która umożliwia komfortowe zarządzanie kluczowymi elementami systemu. Po zalogowaniu administratora witają prosty ekran startowy, z którego można wybrać jedną z dostępnych sekcji.

Do wyboru są trzy podstawowe opcje:

- Zarządzanie użytkownikami – umożliwia przeglądanie, modyfikowanie oraz usuwanie już istniejących kont.
- Dodaj nowego użytkownika – to szybki sposób na wprowadzenie pracownika lub innej osoby do systemu.
- Zarządzanie usługami – pozwala na aktualizowanie oferty warsztatu: dodawanie nowych usług, a także edytowanie oraz usuwanie już istniejących.

Całość została stworzona w sposób, który sprawia, że obsługa panelu jest intuicyjna i szybka. Przycisków jest bardzo dobrze widocznych, a cały układ jest klarowny i uporządkowany.

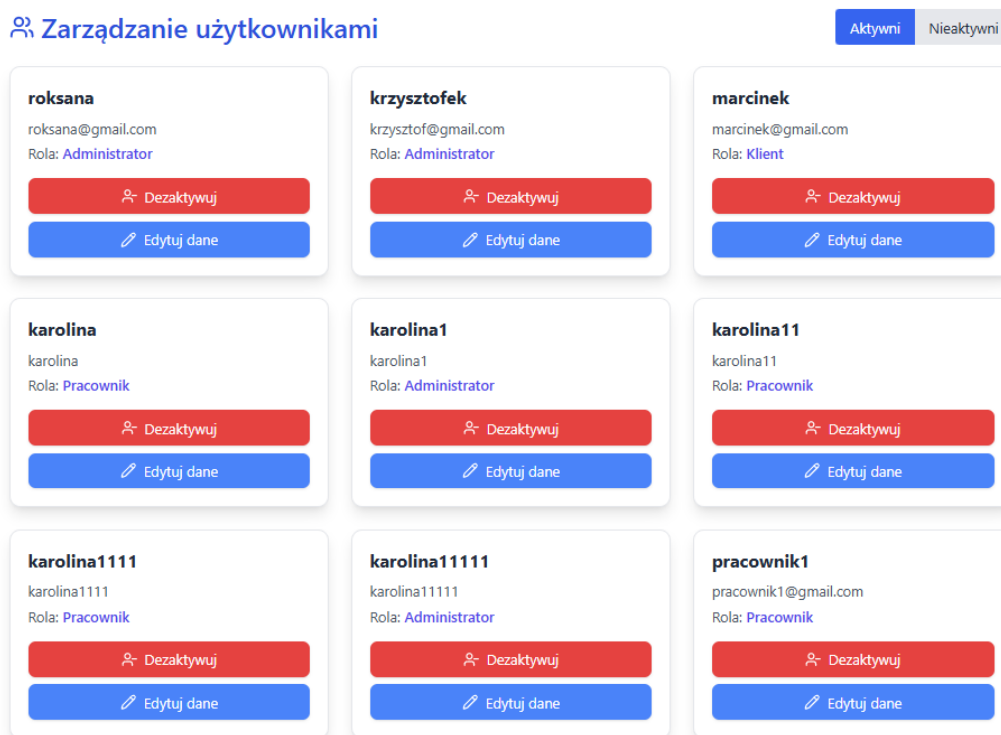
6.3.1. Zarządzanie użytkownikami

W obszarze zarządzania kontami użytkowników, administrator ma pełną kontrolę nad wszystkimi zapisanymi w systemie użytkownikami. Interfejs jest podzielony na dwie sekcje: aktywni i nieaktywni, co ułatwia znalezienie konkretnego konta.

Każdy użytkownik prezentowany jest na przejrzystej karcie, która zawiera jego podstawowe informacje – nazwę użytkownika, adres e-mail oraz przypisaną rolę (na przykład administrator, pracownik, klient). Główne funkcje dostępne w tej sekcji to:

- Dezaktywacja konta – za pomocą przycisku „Dezaktywuj”, administrator ma możliwość wyłączenia konta użytkownika, który nie powinien już mieć dostępu do systemu (na przykład po zakończeniu współpracy). Należy pamiętać, że użytkownicy nie są usuwani z bazy danych! Zamiast tego przechodzą na listę użytkowników nieaktywnych, co umożliwia zachowanie historii ich aktywności oraz danych kontaktowych.
- Ponowna aktywacja konta – w zakładce „Nieaktywni” można przywrócić konto do użytku, klikając przycisk „Aktywuj ponownie”. Jest to bardzo pomocne, na przykład w przypadku powrotu pracownika po przerwie.
- Edycja informacji – każda osoba może być edytowana w momencie, gdy kliknie się „Edytuj dane”. Wówczas rozwija się formularz, gdzie można zmienić takie dane jak: imię, nazwisko, e-mail, numer telefonu, adres i rola. Można także zmienić login, jeśli zajdzie taka potrzeba.

Taki system zarządzania kontami jest znacznie bezpieczniejszy niż permanentne usuwanie danych – pozwala na zachowanie historii, unikanie przypadkowych strat informacji oraz łatwiejsze zarządzanie dostępem do systemu.



Zrzut ekranu nr 21 – widok zarządzania użytkownikami

Zarządzanie użytkownikami

Aktywni

Nieaktywni

krzysztof111 krzysztof111 Rola: Administrator <div>Aktywuj ponownie</div> <div>Edytuj dane</div>	worker1 worker1@example.com Rola: Pracownik <div>Aktywuj ponownie</div> <div>Edytuj dane</div>	kwiatkowski.adriaan adrian.kwawski@carworhop.pl Rola: Administrator <div>Aktywuj ponownie</div> <div>Edytuj dane</div>
kwiatkowski.adrian adrian.kwiawski@carworhop.pl Rola: Administrator <div>Aktywuj ponownie</div> <div>Edytuj dane</div>		

Zrzut ekranu nr 22 – widok użytkowników nieaktywnych

Edytuj użytkownika

Nazwa użytkownika

roksana

 Email

roksana@gmail.com

Imię

roksana

Nazwisko


roksana

 Telefon

111222333

 Adres


Cambridge

 Role użytkownika:

☐ Klient

☐ Pracownik

☒ Administrator

 Zapisz zmiany

Zrzut ekranu nr 23 – widok zarządzania dostępnymi terminami

6.3.2. Dodaj nowego użytkownika

W tej części administrator ma możliwość ręcznego dodania nowego użytkownika do systemu – na przykład pracownika, klienta lub innego administratora. Formularz zawiera wszystkie istotne pola wymagane do założenia konta oraz przypisania odpowiedniej roli.

Pola formularza:

- Nazwa użytkownika – unikalna nazwa, która służy do logowania.
- Adres e-mail – e-mail, który można używać do komunikacji oraz resetowania hasła.
- Hasło – hasło do logowania. Wprowadzone hasło jest odpowiednio szyfrowane na serwerze.
- Imię / Nazwisko – pełne imię i nazwisko użytkownika.
- Numer telefonu – dobrowolny kontakt.
- Adres – adres zamieszkania lub kontaktowy.
- Rola użytkownika – wpływają na dostęp do określonych funkcji programu

Role użytkownika:

- KLIENT – konto klienta, które ma dostęp do usług i rozpoczyna rezerwacje.
- PRACOWNIK – konto pracownika, które ma dostęp do panelu obsługi zamówień.
- ADMIN – konto z pełnymi uprawnieniami do zarządzania użytkownikami, usługami oraz systemem.

Po wypełnieniu wszystkich pól i kliknięciu przycisku „Zapisz użytkownika”, informacje są przesyłane do backendu, a nowe konto zostaje natychmiast dodane do bazy danych.

Ten formularz znacznie upraszcza wprowadzanie nowych pracowników lub szybkie tworzenie kont testowych podczas pracy nad aplikacją.

Metoda `createUser` służy do tworzenia nowego użytkownika w systemie, zazwyczaj przez administratora. Najpierw sprawdza, czy podana nazwa użytkownika lub adres e-mail nie są już zajęte – jeśli tak, przerywa działanie i zgłasza odpowiedni błąd. Następnie weryfikuje, czy zostały przypisane jakiejkolwiek role – bez tego konto nie może zostać utworzone. Jeśli wszystko jest w porządku, tworzony jest nowy obiekt użytkownika, w którym hasło zostaje zakodowane, a pozostałe dane – takie jak imię, nazwisko, telefon czy adres – są przypisywane z formularza. Gotowy użytkownik zostaje zapisany w bazie danych i od tej pory może korzystać z systemu zgodnie ze swoją rolą.


```

1 usage
80 @ ~
public User createUser(CreateUserRequest request) {
81 ~
    if (userRepository.findByUsername(request.getUsername()).isPresent()) {
82 ~
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "Username already taken");
83 ~
    }
84 ~
    if (userRepository.findByEmail(request.getEmail()).isPresent()) {
85 ~
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "Email already registered");
86 ~
    }
87 ~
    if (request.getRoles() == null || request.getRoles().isEmpty()) {
88 ~
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "User must have at least one role");
89 ~
    }
90 ~
    User user = new User(
91 ~
        request.getUsername(),
92 ~
        request.getEmail(),
93 ~
        passwordEncoder.encode(request.getPassword()),
94 ~
        request.getFirstName(),
95 ~
        request.getLastName(),
96 ~
        request.getPhoneNumber(),
97 ~
        request.getAddress(),
98 ~
        request.getRoles()
99 ~
    );
100 ~
    return userRepository.save(user);
101 ~
}
102 ~
103 ~
104 ~
105 ~
106 ~

```

Zrzut ekranu nr 24 – kod źródłowy funkcji na dodawanie nowego użytkownika

6.3.3. Zarządzanie usługami

W tej sekcji administrator może w pełni kontrolować dostępne w systemie usługi. Wszystkie usługi są prezentowane na przejrzystych kartach, zawierających nazwę, cenę, opis oraz informację o dostępności dla klientów.

Dostępne funkcje:

- Dodanie nowej usługi - po naciśnięciu zielonego przycisku „+ Dodaj nową usługę” wyświetli się formularz, w którym można wpisać nazwę, cenę, opis i zaznaczyć, czy usługa ma być natychmiastowo dostępna. Umożliwia to administratorowi łatwe rozszerzenie oferty bez potrzeby modyfikacji kodu.
- Edycja istniejącej usługi - każdą z usług można modyfikować – zmieniając nazwę, opis, cenę lub stan dostępności. Wystarczy jedynie kliknąć przycisk „Edytuj”, a system otworzy prosty formularz z już wypełnionymi danymi. Jest to bardzo wygodne, na przykład przy zmianie cennika lub terminologii.
- Usuwanie usługi - usługę można usunąć z systemu, lecz tylko w przypadku, gdy nie ma na nią aktywnej rezerwacji przez żadnego użytkownika. Jeśli usługa została już zarezerwowana przez klienta, próba jej usunięcia zostanie zablokowana. W takim wypadku, jeśli chcemy „usunąć” usługę z oferty, wystarczy odznaczyć opcję „Dostępna”

w formularzu edycji. To spowoduje, że klienci nie będą mogli jej wybrać, zachowując jednak dane o wcześniejszych rezerwacjach.

Ten system jest elastyczny i bezpieczny – umożliwia rozwój oferty przy jednoczesnym zachowaniu integralności danych i historii zamówień.

Zarządzanie usługami

+ Dodaj nową usługę

Przegląd techniczny Cena: 150 PLN Opis: Kompleksowa kontrola stanu technicznego pojazdu – obowiązkowy przegląd rejestracyjny. Dostępna: Tak <div>Usuń</div> <div>Edytuj</div>	Wymiana oleju i filtrów Cena: 250 PLN Opis: Wymiana oleju silnikowego oraz filtrów: oleju, powietrza i kabinowego – zalecane co 10–15 tys. km. Dostępna: Tak <div>Usuń</div> <div>Edytuj</div>	Diagnostyka komputerowa Cena: 100 PLN Opis: Szybka analiza błędów silnika i innych układów za pomocą interfejsu OBD2. Dostępna: Tak <div>Usuń</div> <div>Edytuj</div>
Naprawa układu wydechowego Cena: 350 PLN Opis: Usunięcie nieszczelności, wymiana tłumika lub elementów układu wydechowego. Dostępna: Tak <div>Usuń</div> <div>Edytuj</div>	Wymiana klocków hamulcowych (oś przednia) Cena: 180 PLN Opis: Demontaż zużytych i montaż nowych klocków hamulcowych na osi przedniej. Dostępna: Tak <div>Usuń</div> <div>Edytuj</div>	Wymiana tarcz i klocków hamulcowych (komplet) Cena: 450 PLN Opis: Wymiana tarcz oraz klocków na jednej osi – gwarancja bezpieczeństwa hamowania. Dostępna: Tak <div>Usuń</div> <div>Edytuj</div>

Zrzut ekranu nr 25 – widok zarządzania dostępnymi usługami

Edytuj usługę

Nazwa usługi:

Przegląd techniczny

Cena:

150

Opis:

Kompleksowa kontrola stanu technicznego pojazdu – obowiązkowy przegląd rejestracyjny.

☒ Dostępna

Anuluj

Zapisz

Zrzut ekranu nr 26 – okno do edycji usługi

Dodaj nową usługę

Nazwa usługi:

Cena:

Opis:

☐ Dostępna

Anuluj

Zapisz

Zrzut ekranu nr 27 – okno dodawania nowej usługi

6.4. Funkcje ogólnodostępne

Aplikacja oferuje różne publiczne podstrony, z których mogą korzystać wszyscy użytkownicy – zarówno ci posiadający konto, jak i ci, którzy dopiero zamierzają się zarejestrować. Te obszary mają na celu zaprezentowanie oferty warsztatu, zachęcenie do rejestracji oraz umożliwienie prostego kontaktu.

6.4.1. Strona główna

Strona główna działa jak wizytówka warsztatu. Natychmiast wita odwiedzających hasłem „Twoje Auto w Najlepszych Rękach” oraz zachęca do zapoznania się z ofertą za pomocą przycisku „Dowiedz się więcej”. Poniżej znajduje się szybki przegląd dostępnych usług oraz kilka argumentów, dla których warto wybrać ten warsztat – nowoczesne urządzenia, wykwalifikowani mechanicy, krótkie terminy. Na stronie znajduje się także zaproszenie do rejestracji lub logowania, co umożliwia użytkownikowi szybkie założenie konta lub umówienie wizyty.

6.4.2. Usługi

W sekcji „Usługi” widoczna jest aktualna lista propozycji oferowanych przez warsztat. Każda z usług jest prezentowana na oddzielnej karcie, która zawiera:

- nazwa,
- krótki opis,
- oraz cenę w złotych.

Użytkownik nie musi się logować, aby mieć dostęp do cennika. Dzięki temu może od razu zorientować się, czego się spodziewać i jakie usługi są dostępne. Lista jest na bieżąco aktualizowana – administrator ma możliwość w każdej chwili zmienić ceny lub dodać nowe usługi.

6.4.3. Kontakt

Zakładka „Kontakt” zawiera wszelkie istotne informacje dotyczące warsztatu:

- lokalizacja,
- numer telefonu,
- adres e-mail,
- godziny pracy (z uwzględnieniem dni w tygodniu).

Po prawej stronie można znaleźć formularz kontaktowy, który pozwala na wysłanie wiadomości bezpośrednio przez stronę. To praktyczna opcja dla tych, którzy chcą uzyskać dodatkowe informacje bez konieczności telefonowania.

6.4.4. O nas

Strona „O nas” to przestrzeń, gdzie warsztat może się zaprezentować i opisać swój zakres działalności. Znajdziesz tu krótki opis historii firmy oraz część z punktami, które przedstawiają, czemu warto skorzystać z ich propozycji – w tym m. in. indywidualne traktowanie klientów, gwarancja jakości czy zniżki dla lojalnych klientów. Na dole tej strony umieszczono także galerię z wykonanych napraw – na przykład zdjęcia wymienionego silnika lub malowania pojazdów. To ilustruje faktyczne zdolności zespołu i wzmacnia zaufanie.

7. Testowanie aplikacji

Tworzenie programów to coś więcej niż tylko pisanie kodu; to także nieprzerwana kontrola jego poprawności. Nawet najlepiej zaplanowana struktura może zawierać wady, które mogą nie być dostrzegalne od razu. Z tego powodu testowanie odgrywa kluczową rolę w procesie rozwijania aplikacji – obecnie traktowane jako integralna część inżynierii oprogramowania, a nie jedynie jako końcowy etap projektu.

Testy pozwalają na dostrzeganie błędów już na początkowych fazach rozwoju, co znacznie obniża wydatki związane z ich późniejszym usuwaniem. Mają również funkcję dokumentacyjną – pokazują, jak system powinien działać w określonych warunkach. W środowisku Agile lub DevOps testy są automatycznie uruchamiane przy każdym nowym wdrożeniu (CI/CD), co umożliwia utrzymanie wysokiej jakości kodu przez cały okres realizacji projektu. [19][20].

7.1. Rodzaje testów w procesie wytwarzania oprogramowania

Testy można klasyfikować w różne grupy – w oparciu o ich cel, zakres oraz moment realizacji:

Testy jednostkowe (unit tests)

To najprostsza forma testowania – pojedyncza funkcja, metoda lub element. Ich celem jest weryfikacja, czy dana jednostka działa poprawnie w pełnej izolacji. Tego rodzaju testy są szybkie do uruchomienia i zwykle towarzyszą pisaniu kodu – zgodnie z filozofią Test-Driven Development (TDD) [19].

Testy integracyjne (integration tests)

Sprawdzają, czy różne moduły systemu współdziałają ze sobą w odpowiedni sposób. W aplikacjach frontendowych przykładami mogą być testy komponentów, które łączą się z interfejsem API i zajmują się przetwarzaniem danych użytkownika. W systemach backendowych testy łączą funkcje kontrolera, serwisu i repozytorium [21].

Testy systemowe i E2E (end-to-end)

To rozbudowane testy całego systemu w warunkach zbliżonych do tych produkcyjnych. Symulują realistyczne scenariusze użytkowników, takie jak proces rejestracji, składania zamówienia czy edytowania danych osobowych. W tym przypadku testowane jest całe spektrum aplikacji – od interfejsu użytkownika, przez API, aż do bazy danych [20].

Testy regresyjne

Po każdej zmianie w kodzie może wystąpić ryzyko, że coś zostanie przypadkowo uszkodzone. Testy regresyjne pomagają wychwycić takie przypadki, ponieważ są uruchamiane automatycznie po każdej aktualizacji kodu [22].

Testy wydajnościowe, bezpieczeństwa i użyteczności

W bardziej skomplikowanych projektach bada się także aspekty niefunkcjonalne. Obejmują one m. in. szybkość reakcji systemu pod obciążeniem, odporność na ataki oraz łatwość obsługi interfejsu użytkownika [20].

7.2. Zakres wykonanych testów aplikacji

W trakcie realizacji projektu inżynierskiego wykonano testy dla dwóch kluczowych elementów aplikacji, które dotyczą różnych etapów testowania – jednostkowego i integracyjnego.

AddUser – testy jednostkowe

Komponent AddUser służy do zakupu nowego użytkownika i zawiera formularz z różnymi polami tekstowymi oraz opcją wyboru ról. Dla tego komponentu opracowałem testy jednostkowe, które miały na celu sprawdzenie:

- wyświetlania wszystkich pól w formularzu (np. nazwa użytkownika, email, hasło itd.),
- prawidłowego działania mechanizmu wprowadzania danych do każdego z pól,
- odpowiedzi checkboxów przypisujących role użytkownikom (KLIENT, PRACOWNIK, ADMIN).

Testy zostały przeprowadzone przy użyciu frameworka Vitest oraz biblioteki React Testing Library [23]. Skupiły się na wewnętrznej logice komponentu – bez konieczności symulowania działania backendu – i były realizowane w całkowitej izolacji od pozostałej części aplikacji.

UserReservations – testy integracyjne

Drugi testowany element to UserReservations, który ma na celu pobranie rezerwacji zalogowanego użytkownika i prezentację ich w formie kart. Dodatkowo umożliwia użytkownikowi edytowanie oraz usuwanie rezerwacji. W tym przypadku zastosowano testy integracyjne, ponieważ element ten współpracuje z zewnętrzną usługą (orderService) i przetwarza informacje o użytkowniku. Zakres testów integracyjnych obejmował:

- sytuację, gdy użytkownik nie ma żadnych rezerwacji (widok pusty),
- poprawne pokazanie jednej lub więcej rezerwacji z danymi pobranymi z mockowanego API,
- przeprowadzenie operacji usunięcia rezerwacji (w tym potwierdzenie przez użytkownika i aktualizacja widoku).

Te testy pozwoliły zweryfikować, że element poprawnie współdziała z warstwą usług oraz odpowiednio reaguje na zachowania użytkownika w kontekście rzeczywistych sytuacji [23].

Testowanie aplikacji obejmowało zarówno testy jednostkowe, jak i integracyjne. W ramach testów jednostkowych sprawdzono, czy formularze działają prawidłowo, natomiast testy

integracyjne skupiły się na interakcji komponentów z zewnętrznymi usługami oraz ich reakcjach na działania użytkowników.

Wykorzystanie nowoczesnych narzędzi do testowania, takich jak Vitest i Testing Library, pozwoliło na stworzenie jasnych i efektywnych testów, które nie tylko potwierdzają poprawność kodu, ale również służą jako dokumentacja. Dzięki temu zyskano większą pewność, że aplikacja będzie prawidłowo funkcjonować również po wprowadzeniu nowych zmian lub aktualizacji.

8. Podsumowanie

Celem tej pracy było zaprojektowanie i stworzenie nowoczesnego systemu informatycznego do zarządzania warsztatem samochodowym, zintegrowanego z sztuczną inteligencją. Opracowana aplikacja internetowa umożliwia klientom łatwe rezerwowanie usług oraz interakcję z chatbotem AI, który dopasowuje odpowiednią usługę na podstawie opisu problemu. Pracownicy mają możliwość efektywnego zarządzania zleceniami, harmonogramem i komunikacją z klientami. Administratorzy natomiast posiadają pełną kontrolę nad systemem, kontami użytkowników oraz ofertą dostępnych usług.

System został zaprojektowany w architekturze klient-serwer, korzystając z technologii React, Spring Boot oraz PostgreSQL. Został on zintegrowany z zewnętrznymi usługami, takimi jak PayU, który obsługuje płatności, oraz ChatGPT, wspierający klientów poprzez sztuczną inteligencję. Interfejs stworzono zgodnie z zasadami nowoczesnego UX/UI, dostosowując go do trzech kluczowych ról w systemie: klienta, pracownika oraz administratora.

Podczas realizacji projektu przeprowadzono analizę dostępnych na rynku rozwiązań i znaleziono ich niedostatki, głównie brak intuicyjności, minimalną dostępność nowoczesnych funkcji oraz niewystarczające wsparcie dla małych warsztatów. Nowo stworzony system miał na celu wypełnienie tej luki, oferując narzędzie, które jest funkcjonalne, proste w użytkowaniu oraz nowoczesne.

Warto zaznaczyć, że prace nad tym systemem umożliwiły nie tylko poznanie i praktyczne zastosowanie nowoczesnych technologii webowych, ale także oszacowanie rzeczywistych potrzeb biznesowych w motoryzacji. Projekt był również sposobnością do stawienia czoła wyzwaniom związanym z bezpieczeństwem, integracją z zewnętrznymi API oraz testowaniem oprogramowania w realnym środowisku.

Na obecnym etapie system oferuje pełną funkcjonalność operacyjną, chociaż jego struktura była przygotowana z myślą o przyszłym rozwoju. W przyszłości planowane jest wprowadzenie takich funkcji jak moduł zarządzania magazynem części zamiennych, system ocen dla klientów, zaawansowana analityka AI do prognozowania obciążenia warsztatu oraz integracja z systemami księgowymi i ERP. Rozważane jest także stworzenie wersji mobilnej aplikacji, co zwiększyłoby jej dostępność dla użytkowników, oraz wprowadzenie prostego modułu CRM wspierającego relacje z klientami.

Podsumowując, projekt osiągnął zamierzone cele i stanowi mocny fundament do dalszego rozwoju – zarówno od strony technologicznej, jak i praktycznej w kontekście rzeczywistej pracy warsztatu samochodowego.

Bibliografia

1. Warsztat24 – opis systemu i funkcji. Źródło: <https://warsztat24.com>
2. Inter Cars – eSOW. Źródło: <https://intercars.com.pl/pl/oprogramowanie>
3. Johnson, R. et al. Spring Framework Documentation. Pivotal Software, 2024.
4. Walls, C. Spring Boot in Action. Manning Publications, 2021.
5. Walls, C. Spring Security in Action. Manning Publications, 2021.
6. Bauer, C., King, G. Java Persistence with Hibernate. Manning Publications, 2020.
7. Abramov, D., Clark, A. React – Official Documentation. Meta Platforms, 2024.
<https://reactjs.org>
8. React Router Team. React Router Documentation. Remix Software Inc., 2024.
<https://reactrouter.com>
9. Tailwind Labs. Tailwind CSS Documentation. <https://tailwindcss.com/docs>
10. Momjian, B. PostgreSQL: Introduction and Concepts. Addison-Wesley Professional, 2023.
11. PostgreSQL Global Development Group. PostgreSQL 16 Documentation. 2024.
<https://www.postgresql.org/docs/>
12. Krosing, H., Etsuro, M. PostgreSQL Server Programming. Packt Publishing, 2022.
13. PayU. PayU Developer Documentation – REST API. 2024.
<https://developers.payu.com/en/>
14. PayU Poland. Bezpieczne płatności online – dokumentacja techniczna. PayU S.A., 2023.
<https://www.payu.pl>
15. PCI Security Standards Council. PCI DSS Quick Reference Guide, Version 4.0. 2022.
<https://www.pcisecuritystandards.org>
16. OpenAI. ChatGPT – Oficjalna strona produktu. <https://openai.com/chatgpt>
17. OpenAI. GPT-4 Technical Overview. 2023. <https://openai.com/research/gpt-4>
18. OpenAI API Documentation. <https://platform.openai.com/docs>
19. OpenAI. Polityka prywatności i przetwarzanie danych. <https://openai.com/policies>
20. Beck, K. Test-Driven Development: By Example. Addison-Wesley, 2002.
21. Crispin, L., Gregory, J. Agile Testing: A Practical Guide for Testers and Agile Teams. Addison-Wesley, 2009.
22. Freeman, S., Pryce, N. Growing Object-Oriented Software, Guided by Tests. Addison-Wesley, 2009.
23. Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.

24. Dokumentacja React Testing Library: <https://testing-library.com/docs>

Spis załączników

Zrzut ekranu nr 1 – Widok główny panelu klienta	28
Zrzut ekranu nr 2 – Widok wyboru terminu usługi	29
Zrzut ekranu nr 3 – Widok wyboru usługi	30
Zrzut ekranu nr 4 – Wybór metody płatności	30
Zrzut ekranu nr 5 – Przekierowanie do systemu PayU	31
Zrzut ekranu nr 6 – Formularz danych karty płatniczej	31
Zrzut ekranu nr 7 – Kod źródłowy dla funkcji tworzenia nowego zamówienia	32
Zrzut ekranu nr 8 – Widok rezerwacji użytkownika	33
Zrzut ekranu nr 9 – Formularz kontaktowy do edycji rezerwacji	34
Zrzut ekranu nr 10 – Widok edycji danych klienta	36
Zrzut ekranu nr 11 – Kod źródłowy funkcji na edycję danych użytkownika	37
Zrzut ekranu nr 12 – Widok wspomagania wyboru usługi przez AI	38
Zrzut ekranu nr 13 – Kod źródłowy klasy do diagnozowania usterki przez AI cz. I	39
Zrzut ekranu nr 14 – Kod źródłowy klasy do diagnozowania usterki przez AI cz. II	39
Zrzut ekranu nr 15 – Widok panelu pracownika	40
Zrzut ekranu nr 16 – Widok zamówień do obsługi	41
Zrzut ekranu nr 17 – Widok edycji zamówienia	42
Zrzut ekranu nr 18 – Widok aktualizacji statusów zamówień	43
Zrzut ekranu nr 19 – Kod źródłowy funkcji na aktualizację statusu naprawy dla danego	44
Zrzut ekranu nr 20 – Widok panelu zarządzania dostępnymi terminami	45
Zrzut ekranu nr 21 – Widok zarządzania użytkownikami	46
Zrzut ekranu nr 22 – Widok użytkowników nieaktywnych	47
Zrzut ekranu nr 23 – Widok zarządzania dostępnymi terminami	47
Zrzut ekranu nr 24 – Kod źródłowy funkcji na dodawanie nowego użytkownika	49
Zrzut ekranu nr 25 – Widok zarządzania dostępnymi usługami	50
Zrzut ekranu nr 26 – Okno do edycji usługi	50
Zrzut ekranu nr 27 – Okno dodawania nowej usługi	51