

Politechnika Śląska w Gliwicach
Wydział Automatyki, Elektroniki i Informatyki



Tworzenie aplikacji bazodanowych

Projekt - system wspomagający pracę studiów
car detailerskich

1 Treść zadania

Napisać aplikację wspomagającą pracę studiów car detailerskich. System powinien umożliwić zarządzanie realizacjami, usługami dostępnymi w studiu, obecnymi zasobami oraz sprzętem, kontami a także generowanie raportów dotyczących kosztów i ilości wykonanych usług.

2 Analiza zadania

Zagadnienie przedstawia problem komunikacji aplikacji frontendowej z aplikacją backendową. Do frontendu została wykorzystana biblioteka React.js, natomiast do backendu została wykorzystana biblioteka Node.js

2.1 Struktury danych

Na frontendzie używamy klasowych komponentów Reactowych. Na backendzie stosujemy plik główny app.js który posiada routingi adresów API do poszczególnych plików w katalogu routes. Każdy z plików posiada implementację którejś (lub kilku) z metod: get, post, patch, put, itd.

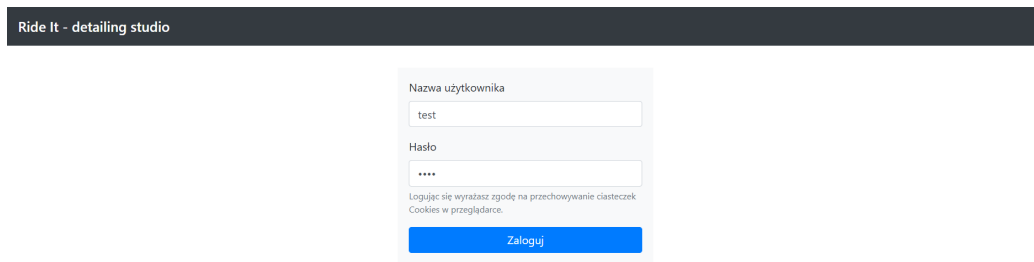
2.2 Algorytmy

Aplikacja frontendowa wymienia dane z backendem każdorazowo przy jakiegokolwiek wprowadzonej zmianie. Dane pobierane są z odpowiednich tabel, zostają odpowiednio połączone i przekazywane dalej.

3 Specyfikacja zewnętrzna

3.1 Okno logowania

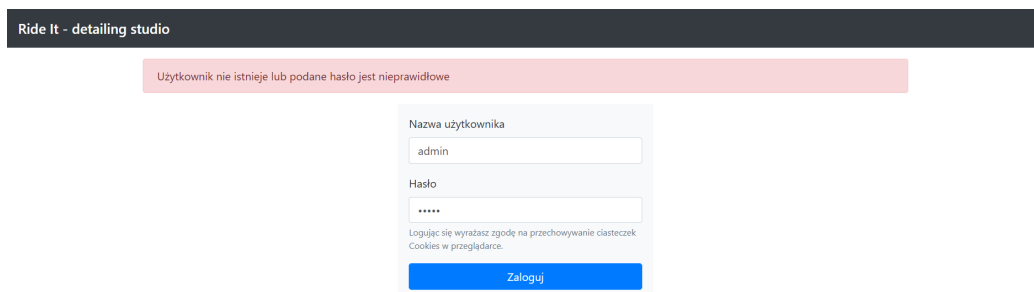
W momencie uruchomienia aplikacji wyświetlane jest strona logowania. Znajduję się tam pole na wpisanie loginu oraz hasła, a także przycisk "zaloguj".



The screenshot shows a login form for 'Ride It - detailing studio'. The form is centered on a dark grey header bar. It contains two input fields: 'Nazwa użytkownika' (Username) with the value 'test' and 'Hasło' (Password) with masked characters '****'. Below the password field is a small text link: 'Logując się wyrażasz zgodę na przechowywanie ciasteczek Cookies w przeglądarce.' At the bottom of the form is a blue button labeled 'Zaloguj'.

Rysunek 1: Strona służąca do logowania.

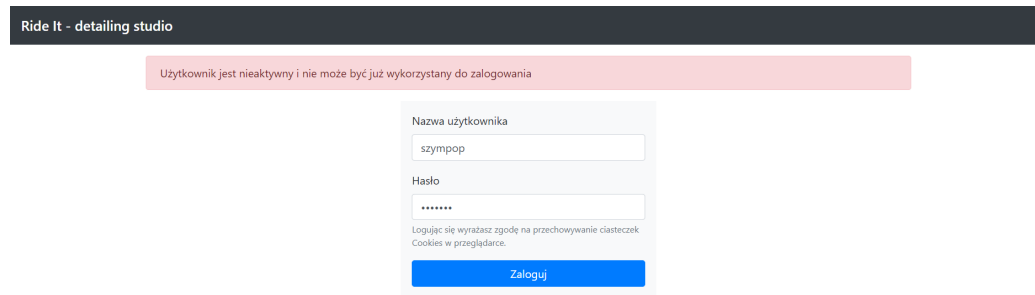
W przypadku wpisania błędnego hasła użytkownik zostaje o tym poinformowany komunikatem: "Użytkownik nie istnieje lub podane hasło jest nieprawidłowe".



This screenshot shows the same login form as Figure 1, but with an error message displayed above it. The error message is 'Użytkownik nie istnieje lub podane hasło jest nieprawidłowe' (User does not exist or the password is incorrect), shown in a pink box. The form fields now contain 'admin' for the username and '*****' for the password. The 'Zaloguj' button remains at the bottom.

Rysunek 2: Strona z komunikatem o błędnym hasle.

W przypadku próby zalogowania się przez użytkownika, którego konto zostało zdezaktywowane zostanie wyświetlony komunikat: "Użytkownik jest nieaktywny i nie może być już wykorzystany do zalogowania".



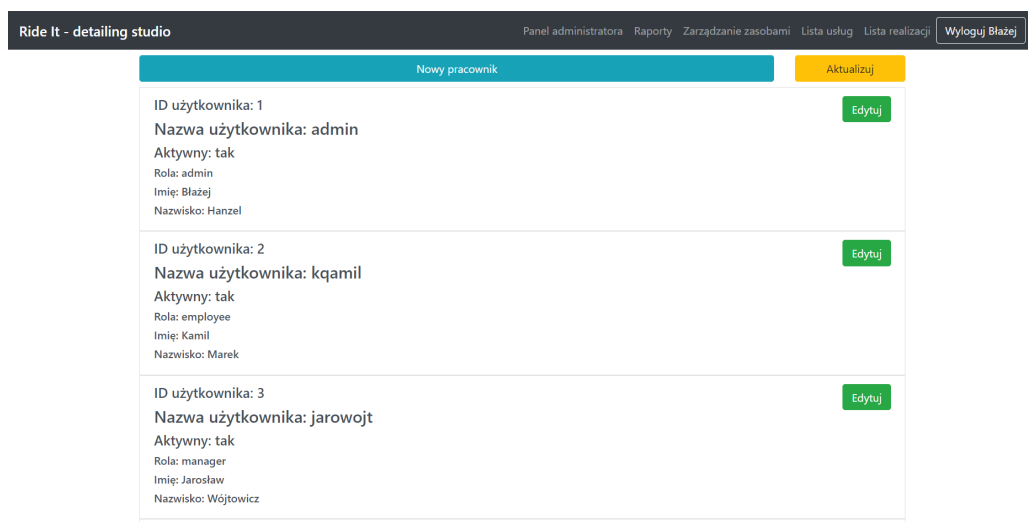
The screenshot shows a web application interface. At the top, a dark header bar contains the text "Ride It - detailing studio". Below the header, a light pink error message box displays the text: "Użytkownik jest nieaktywny i nie może być już wykorzystany do zalogowania". Centered below the message is a white login form with a light gray border. The form contains two input fields: "Nazwa użytkownika" (Username) with the value "szympop" and "Hasło" (Password) with masked characters "*****". Below the password field, there is a small line of text: "Logując się wyrażasz zgodę na przechowywanie ciasteczek Cookies w przeglądarce." At the bottom of the form is a blue button labeled "Zaloguj".

Rysunek 3: Strona z komunikatem o nieaktywnym użytkowniku

Po wpisaniu odpowiedniego loginu, hasła oraz kliknięciu przycisku „Zaloguj” lub klawisza „Enter” przechodzimy do kolejnej strony, która jest zależna od podanych danych. Po zalogowaniu na dane konto pojawia się odpowiedni pasek nawigacyjny, a na jego końcu pojawia się klawisz „wyloguj imię_użytkownika”.

3.2 Konto administratora

Po zalogowaniu się jako administrator ukazuje się panel administratora wraz z paskiem nawigacyjnym administratora.

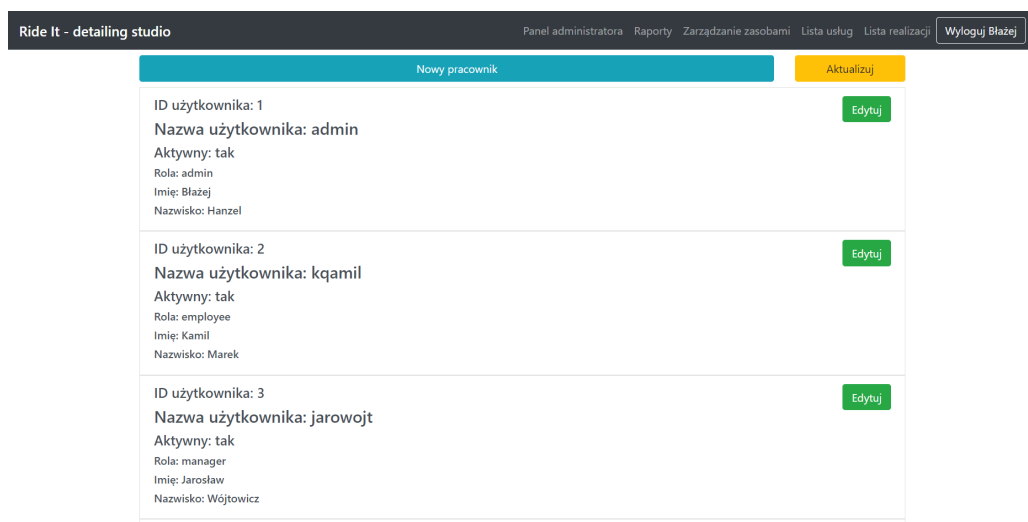


Rysunek 4: Widok konta administratora

Administrator posiada 5 paneli: Panel administratora, Raporty, Zarządzanie zasobami, Lista usług oraz Lista realizacji

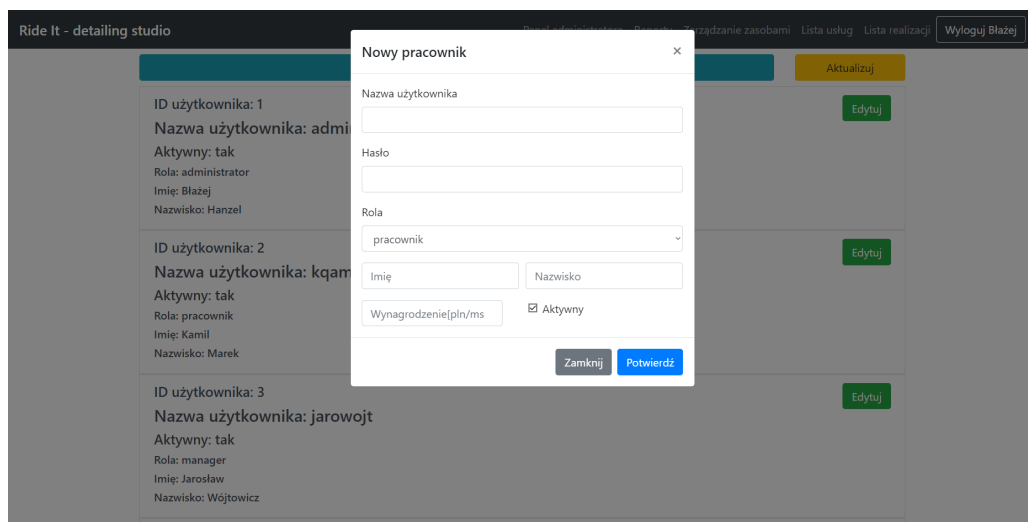
3.2.1 Panel administratora

Po kliknięciu przycisku „Panel administratora” na pasku nawigacyjnym otwiera się strona do zarządzania pracownikami. Wyświetlane zostają przyciski „Nowy pracownik”, „Aktualizuj”, a także lista wszystkich pracowników - aktywnych i nieaktywnych.



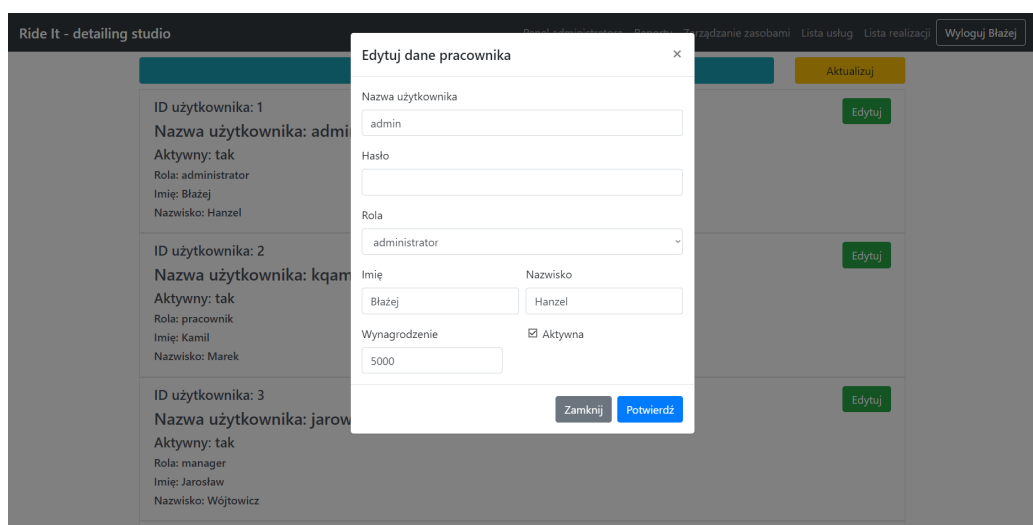
Rysunek 5: Wygląd panelu administratora.

W przypadku kliknięcia przycisku "Nowy pracownik" pojawia nam się okienko, w którym należy podać nazwę użytkownika, hasło, z rozwijalnej listy rolę, imię, nazwisko, wynagrodzenie oraz zaznaczyć czy konto to jest aktywne czy też nie. Okienko to można zamknąć lub potwierdzić dodanie nowego użytkownika odpowiednimi przyciskami.



Rysunek 6: Okno z panelem do dodawania nowego pracownika.

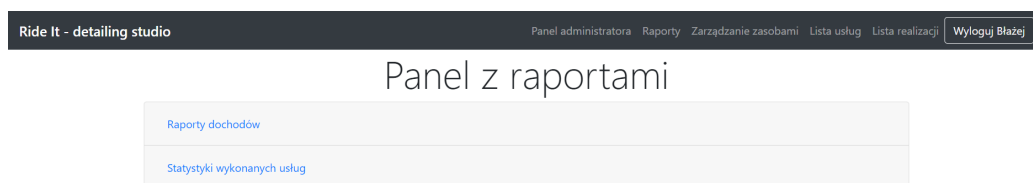
W przypadku kliknięcia przycisku "Aktualizuj" strona odświeży się, dzięki czemu nowo dodany użytkownik pojawi się na wyświetlanej liście. W przypadku kliknięcia przycisku "edytuj", który znajduje się obok poszczególnych użytkowników, możemy edytować wszystkie dane użytkownika. Wyświetla nam się okno do edycji, które wygląda tak samo jak okno do dodawania nowego użytkownika.



Rysunek 7: Okno z panelem do edytowania wybranego użytkownika.

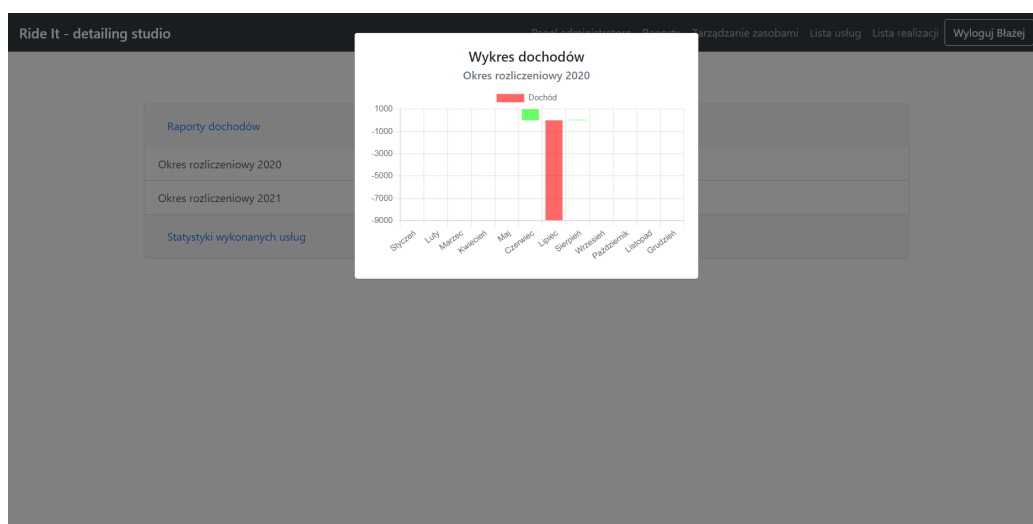
3.2.2 Raporty

Po kliknięciu przycisku „Raporty” otwiera się panel z raportami. Panel ten zawiera raporty dotyczące dochodów oraz statystyki wykonanych usług.



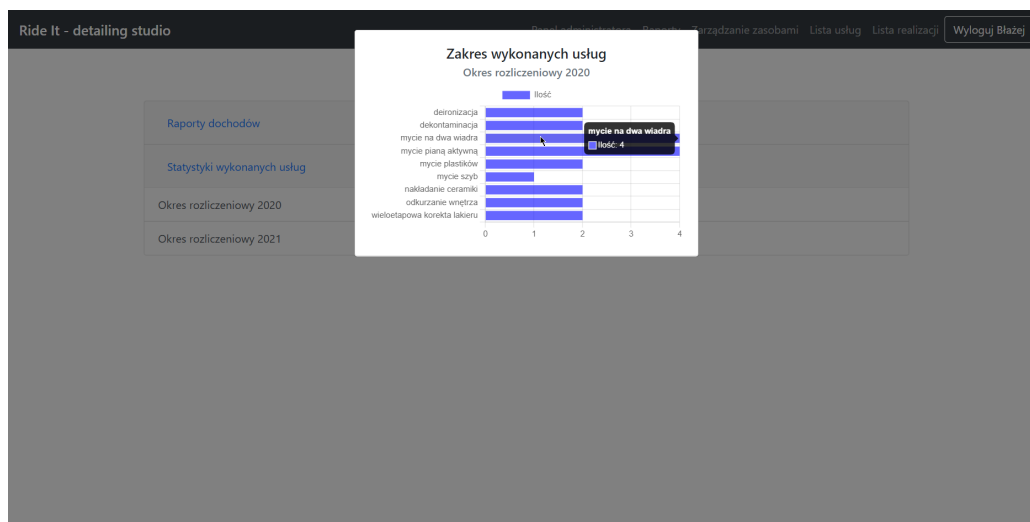
Rysunek 8: Panel z raportami.

W przypadku kliknięcia na pole "Raporty dochodów" rozwija się lista z okresami rozliczeniowymi na dany rok. Po kliknięciu danego pola pojawia się wykres dochodów. Na zielono zaznaczane są miesiące z przychodami dodatnimi, na czerwono z ujemnymi. W momencie najechania na dany słupek wyświetlany jest opis miesiąca i wartość liczbową dochodu. W przypadku gdy klikniemy poza okno, zamyka się ono.



Rysunek 9: Okno z wykresem dochodów na dany rok.

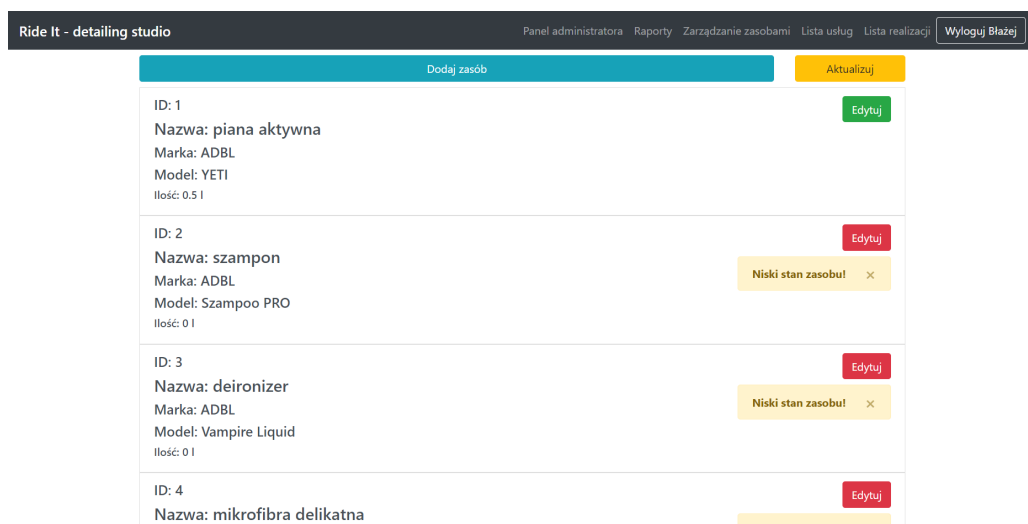
W przypadku kliknięcia na pole "Statystyki wykonanych usług" rozwija się lista z okresami na dany rok. Po kliknięciu danego pola pojawia się wykres ilości wykonania poszczególnych usług. W momencie najechania na dany słupek wyświetlany jest opis jaka to usługa i dokładna wartość liczbowa. Aby zamknąć okno należy kliknąć poza jego pole.



Rysunek 10: Okno z wykresem ilości wykonanych usług.

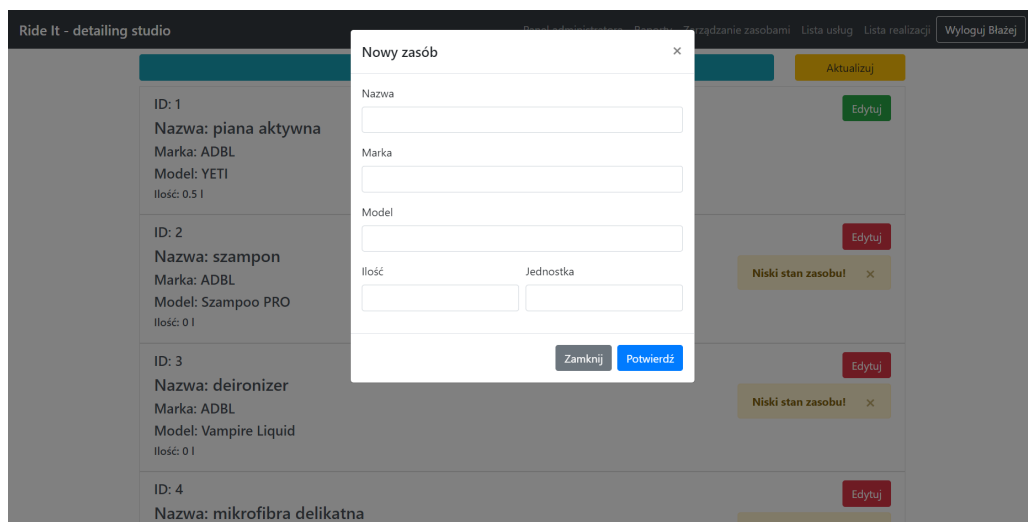
3.2.3 Zarządzanie zasobami

Po kliknięciu przycisku „Zarządzanie zasobami” na pasku nawigacyjnym otwiera się strona do zarządzania zasobami. Wyświetlane zostają przyciski ”Dodaj zasób”, ”Aktualizuj”, a także lista wszystkich zasobów. Każdy zasób ma przycisk edytuj, który w zależności od obecnego stanu, ma inny kolor. W momencie kiedy zasób jest powyżej stanu minimalnego, przycisk ten ma kolor zielony. W przypadku kiedy zasób jest poniżej stanu minimalnego, kolor ten zmienia się na czerwony, a poniżej przycisku pojawia się komunikat o niskim stanie zasobu.



Rysunek 11: Wygląd panelu Zarządzanie zasobami.

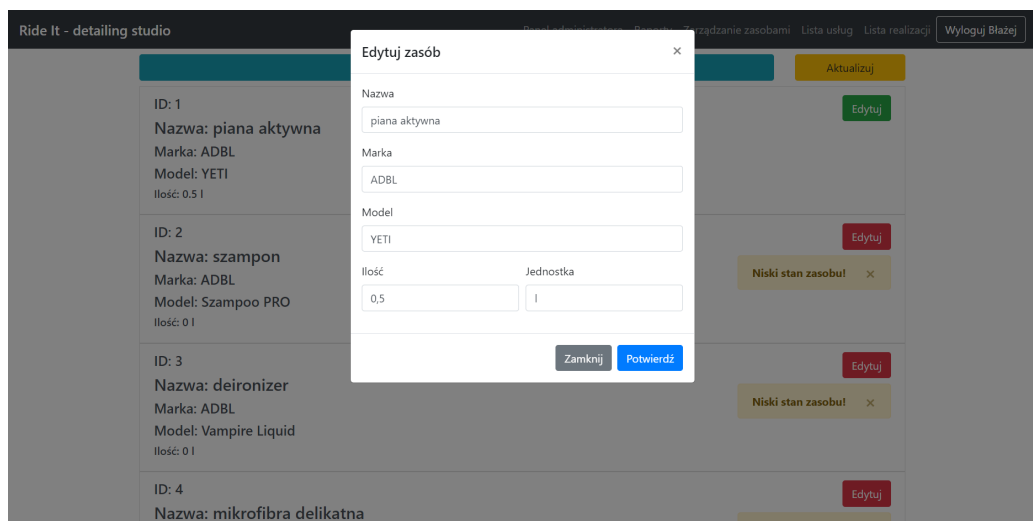
W przypadku kliknięcia przycisku "Dodaj zasób" pojawia nam się okienko, w którym należy podać nazwę zasobu, markę, model, ilość, a także jednostkę. Okienko to można zamknąć lub potwierdzić dodanie nowego użytkownika odpowiednimi przyciskami.



Rysunek 12: Okno z panelem do dodawania nowego zasobu.

W przypadku kliknięcia przycisku "Aktualizuj" strona odświeży się, dzięki

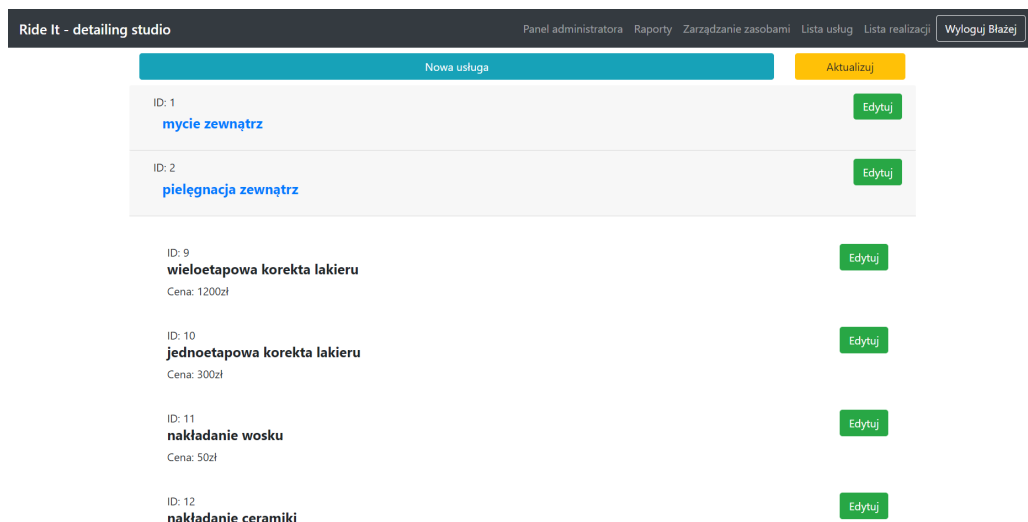
czemu nowo dodany zasób pojawi się na wyświetlanej liście. W przypadku kliknięcia przycisku "edytuj", który znajduje się obok poszczególnych zasobów, możemy edytować wszystkie dane zasobu. Wyświetla nam się okno do edycji, które wygląda tak samo jak okno do dodawania nowego zasobu.



Rysunek 13: Okno z panelem do edytowania wybranego zasobu.

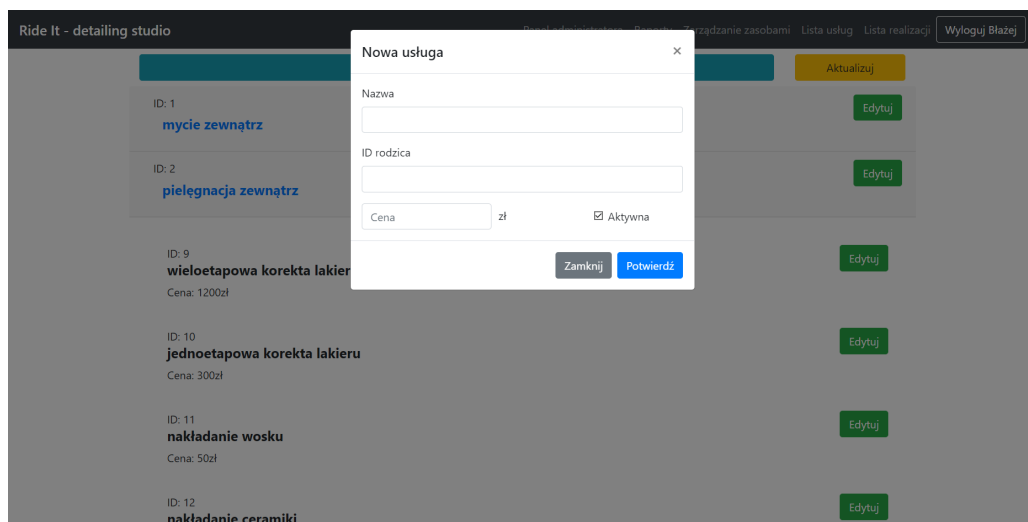
3.2.4 Lista usług

Po kliknięciu przycisku „Lista usług” na pasku nawigacyjnym otwiera się strona do zarządzania usługami. Wyświetlane zostają przyciski ”Nowa usługa”, ”Aktualizuj”, a także lista wszystkich zasobów w postaci kategorii z rozwijanymi podkategoriami. Każda usługa ma przycisk edytuj.



Rysunek 14: Wygląd panelu z listą usług.

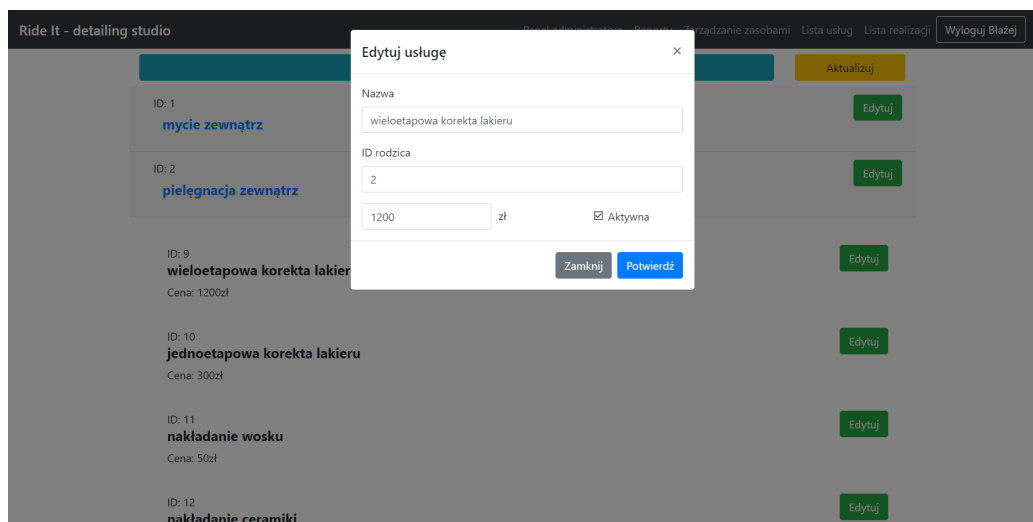
W przypadku kliknięcia przycisku "Nowa usługa" pojawia nam się okienko, w którym należy podać nazwę usługi, id rodzica, cenę, a także czy usługa jest aktywna. Okienko to można zamknąć lub potwierdzić dodanie nowej usługi odpowiednimi przyciskami.



Rysunek 15: Okno z panelem do dodawania nowej usługi.

W przypadku kliknięcia przycisku "Aktualizuj" strona odświeży się, dzięki

czemu nowo dodana usługa pojawi się na wyświetlanej liście. W przypadku kliknięcia przycisku "edytuj", który znajduje się obok poszczególnych usług, możemy edytować wszystkie dane usługi. Wyświetla nam się okno do edycji, które wygląda tak samo jak okno do dodawania nowej usługi.



Rysunek 16: Okno z panelem do edytowania wybranej usługi.

3.2.5 Lista realizacji

Po kliknięciu przycisku „Lista realizacji” na pasku nawigacyjnym otwiera się strona do zarządzania listą realizacji. Wyświetlane zostają przyciski ”Nowa realizacja”, a także lista wszystkich realizacji. Każda realizacja ma przycisk ”Szczegóły”, który po kliknięciu pokazuje listę usług przypisanych do danej realizacji. Każda realizacja, która nie została zakończona na klikalny przycisk ”Zakończ” oraz jej tło ma kolor niebieski.

The screenshot shows a web application interface for 'Ride It - detailing studio'. At the top, there is a navigation bar with links: 'Panel administratora', 'Raporty', 'Zarządzanie zasobami', 'Lista usług', 'Lista realizacji', and a 'Wyloguj Błazej' button. The main content area is titled 'Nowa realizacja' and contains three entries for car detailing services. Each entry has a light blue header with the title 'Nowa realizacja'.

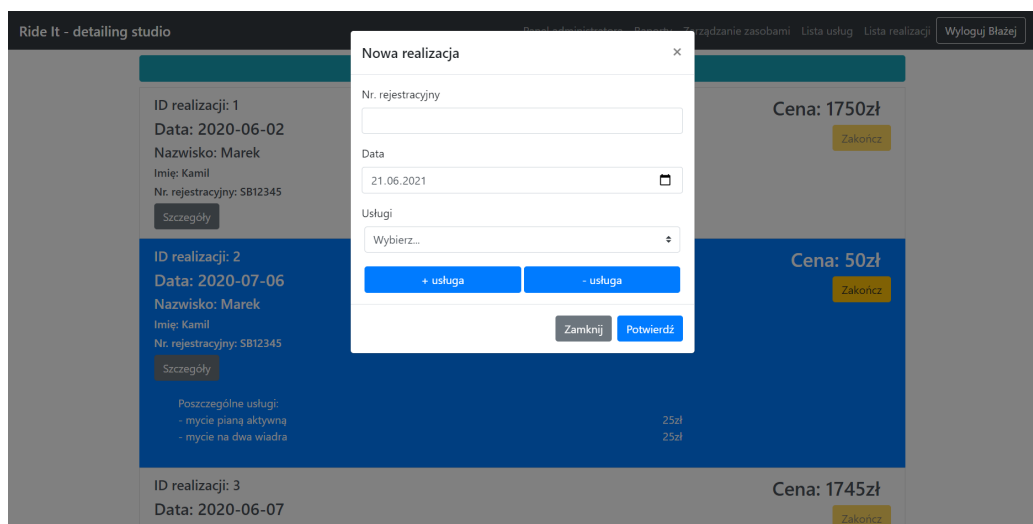
ID realizacji	Data	Nazwisko	Imię	Nr. rejestracyjny	Cena	Akcja
1	2020-06-02	Marek	Kamil	SB12345	1750zł	Zakończ
2	2020-07-06	Marek	Kamil	SB12345	50zł	Zakończ
3	2020-06-07				1745zł	Zakończ

Each entry also includes a 'Szczegóły' button. The second entry (ID 2) has a detailed list of services and their prices:

Poszczególne usługi	Cena
- mycie pianą aktywną	25zł
- mycie na dwa wiadra	25zł

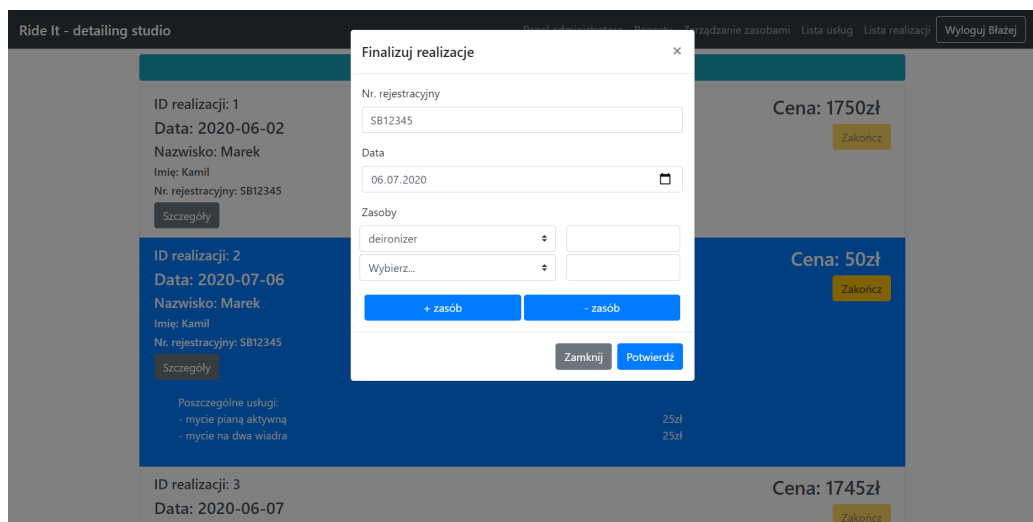
Rysunek 17: Wygląd panelu z listą realizacji.

W przypadku kliknięcia przycisku "Nowa realizacja" pojawia nam się okienko, w którym należy podać numer rejestracyjny samochodu, datę oraz wszystkie usługi, które zostaną wykonane na pojeździe. Usługi wybiera się z listy rozwijalnej, nowe pole na usługi dodaje się przyciskiem "+ usługa", a w przypadku niepotrzebnego pola na usługi usuwa się przyciskiem "- usługa". Okienko to można zamknąć lub potwierdzić dodanie nowej realizacji odpowiednimi przyciskami.



Rysunek 18: Okno z panelem do dodawania nowej realizacji.

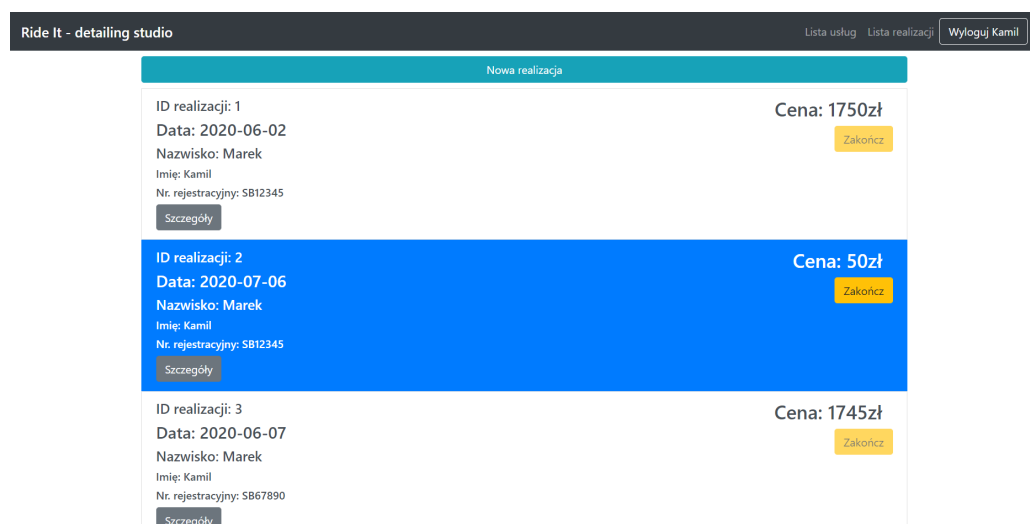
W przypadku kliknięcia przycisku "Zakończ", który znajduje się obok poszczególnych realizacji, możemy zakończyć daną realizację. Wyświetla nam się okno do finalizacji, które wygląda podobnie jak okno do dodania nowej realizacji, jednak zamiast podawania usług, podajemy wszystkie zasoby zużyte przy realizacji.



Rysunek 19: Okno z panelem do finalizowania wybranej realizacji.

3.3 Konto pracownika

Po zalogowaniu się jako pracownik ukazuje się lista realizacji wraz z paskiem nawigacyjnym pracownika.

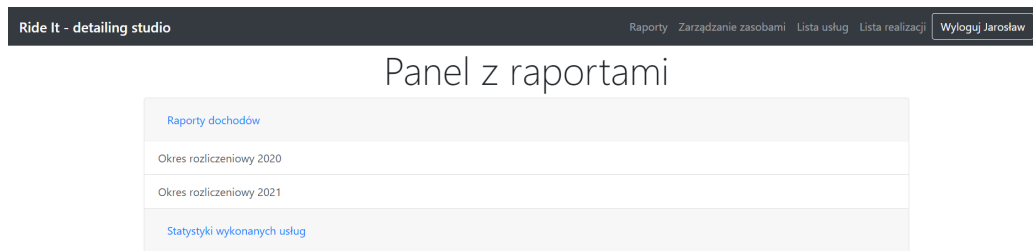


Rysunek 20: Widok konta pracownika

Pracownik posiada 2 panele: Lista usług oraz Lista realizacji. Panele te wyglądają identycznie jak w przypadku konta administratora, jedyną różnicą jest brak możliwości edycji listy usług przez pracownika.

3.4 Panel managera

Po zalogowaniu się jako manager ukazuje się panel z raportami wraz z paskiem nawigacyjnym managera.



Rysunek 21: Widok konta mamagera

Manager posiada 4 panele: Raporty, Zarządzanie zasobami, Lista usług oraz Lista realizacji. Panele te wyglądają identycznie jak w przypadku konta administratora wraz z możliwością edycji.

4 Specyfikacja wewnętrzna

4.1 Frontend

Aplikacja z poziomu frontendu jest podzielona na kilka plików. W pliku:

- `frontend/src/components/EmployeeList.js` - znajduje się kod źródłowy dotyczący wyglądu panelu administratora.
- `frontend/src/components/LoginForm.js` - znajduje się kod źródłowy dotyczący wyglądu panelu logowania.
- `frontend/src/components/navbar.js` - znajduje się kod źródłowy dotyczący wyglądu paska nawigacji.
- `frontend/src/components/ProfitReport.js` - znajduje się kod źródłowy dotyczący wyglądu wykresów z raportami.
- `frontend/src/components/RegistryList.js` - znajduje się kod źródłowy dotyczący listy realizacji.

- `frontend/src/components/ResourceList.js` - znajduje się kod źródłowy dotyczący wyglądu listy zasobów.
- `frontend/src/components/ServicesList.js` - znajduje się kod źródłowy dotyczący wyglądu listy usług.
- `frontend/src/components/StatisticsReport.js` -
- `frontend/src/panels/ReportPanel.js` - znajduje się opis widoku panelu z raportami oraz kod, który pobiera dane o istniejących raportach i umożliwia ich sprawdzenie.
- `frontend/src/App.js` - znajduje się komponent aplikacji, który służy jako root wszystkich komponentów.
- `frontend/src/index.js` - znajduje się komenda startująca Reacta.

4.2 Backend

Aplikacja z poziomu backendu jest podzielona na kilku plików:

- `backend/modules/database.js` - znajduje się definicja połączenia z bazą danych
- `backend/modules/role-check.js` - znajduje się algorytm sprawdzający czy użytkownik posiada wymaganą rolę do dostępu do odpowiedniego panelu.
- `backend/routes/employee-list.js` - posiada obsługę endpointów dotyczących listy użytkowników.
- `backend/routes/login.js` - posiada obsługę endpointów dotyczących logowania.
- `backend/routes/logout.js` - posiada obsługę endpointów dotyczących wylogowania.
- `backend/routes/reports.js` - posiada obsługę endpointów dotyczących raportów.
- `backend/routes/resource-list.js` - posiada obsługę endpointów dotyczących listy zasobów.
- `backend/routes/service-history.js` - posiada obsługę endpointów dotyczących listy realizacji.

- `backend/routes/services – list.js` - posiada obsługę endpointów dotyczących listy usług.
- `backend/app.js` - główny plik startowy aplikacji serwera, w którym definiujemy parametry oraz podpinamy routingi.

5 Testowanie

Aplikacja była testowana ręcznie przez uruchamianie każdego panelu, każdego przycisku. Wykonane zostały próby zalogowania poprawnymi danymi, niepoprawnymi danymi, a także poprawnymi danymi na nieaktywne konta. Każdy panel został przetestowany w każdy możliwy sposób. Dane były dodawane i edytowane. Dane były dodawane zarówno z poziomu aplikacji, jak i ze strony bazy danych. Wszystko komunikowało się ze sobą poprawnie, tzn. dane dodane w aplikacji pojawiały się w bazie danych, tak samo dane dodane w bazie danych pojawiały się w aplikacji.

6 Wnioski

Aplikacja wspomagająca pracę studiów car detailerskich jest aplikacją złożoną. Należało odpowiednio zaprojektować wszystkie panele w sposób czytelny i prosty w obsłudze. Aplikacja wymagała poprawnej komunikacji między frontendem, a backendem. Wielu z nas nie miało do czynienia z pisaniem aplikacji po stronie frontendu jak i backendu, co wydłużało czas projektowania. Wielu z nas dzięki temu zaznała się z językiem "JavaScript", rozwiązaniami frontendowymi jak React, a także Node.js jako backend. Największym wyzwaniem okazała się poprawna obsługa danych, które znajdowały się w wielu tabelach połączonych ze sobą, przez co zapytania stawały się rozległe i skomplikowane.