**Katedra informatyki i automatyki**

Wydział elektrotechniki i informatyki

**Usługi sieciowe w biznesie**

Dokumentacja projektu

**Temat: Aplikacja internetowa wspierająca prace warsztatu samochodowego**

|  |  |
| --- | --- |
| **Imię Nazwisko**  **Nr. gr laboratoryjnej** | **Data oddania** |
| Damian Bielecki 4EFZI-P1 | 04-01-2023 |

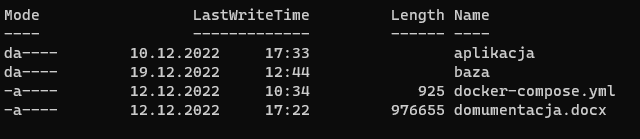
1. **Cel i opis projektu**

Celem projektu jest stworzenie aplikacji internetowej wspomagającej prace warsztatu. Podstawowa funkcjonalność pozwoli na zarządzanie zleceniami, klientami oraz naprawianymi pojazdami. Każdy klienta będzie posiadał specjalny krótki kod pozwalający na podgląd stanu zlecenia i jego kosztów. Dodatkowo serwis będzie implementował podstawowe funkcjonalności do zarządzania pracownikami oraz ich stanowiskami.

Strona serwerowa będzie zbudowana w oparciu o język php i framework symfony. Część wizualna opiera się o szablon adminLTE. Dane będą zapisywane w bazie danych MySQL. Aby uprościć uruchamianie i zarządzanie aplikacją na serwerze zostaną wykorzystywane kontenery dockera.

1. **Uruchomienie projektu**
2. Opis budowania kontenerów

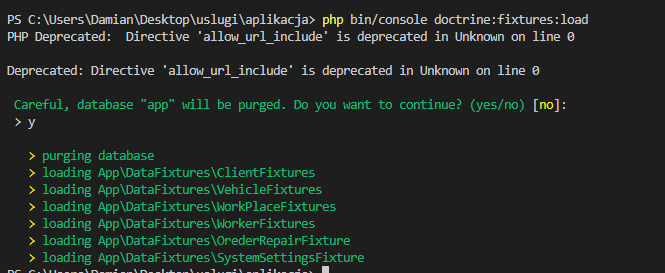
W folderze głównym projektu znajduje się folder aplikacja zawierający cały program oraz plik docker-compose.yml odpowiadający za uruchomienie wszystkich odpowiednich kontenerów. Taki podział pozwala na logiczne oddzielenie wszystkich usług i w przyszłości wystarczy stworzyć oddzielny katalog dla kolejnego kontenera np. serwer cache z zapisanymi ustawieniami.



Rysunek . Struktura folderów

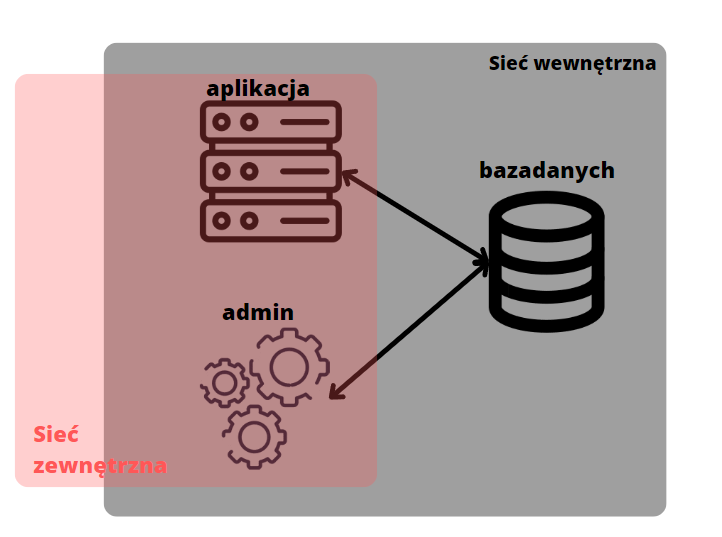
Folder aplikacja zawiera wykonany projekt. Baza zawiera wygenerowane automatycznie dane pozwalające na szybkie przetestowanie aplikacji bez ingerowania w system. W środowisku produkcyjnym baza danych powinna przechowywać dane na odpowiednio przygotowanych wolumenach w dokerze co zwiększy szybkość (bez kopiowania ich z lokalnego folderu). Dane mogą być wygenerowane ponownie poprzez wprowadzenie polecenia do kontenera z aplikacją:

$ php bin/console doctrine:fixtures:load –-env dev



Rysunek . Wygenerowane dane

Kontenery połączone są w następujący sposób:



Rysunek . Połączenia pomiędzy kontenerami

Jak widać istnieją trzy kontenery i dwie sieci. Wszystkie kontenery połączone są w sieci wewnętrznej natomiast kontenery „aplikacja” i „admin” są równocześnie w sieci pozwalającej na łączność z zewnątrz. Opis poszczególnych kontenerów:

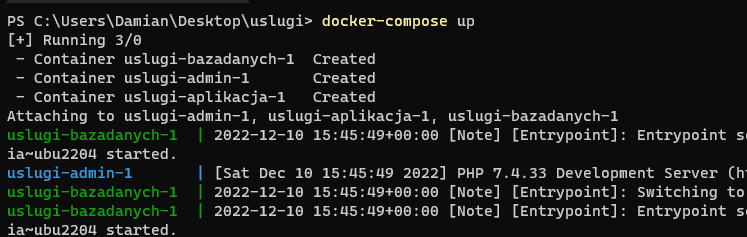
* „aplikacja” – to kontener zawierający całą aplikacje, ma włączone mapowanie portów 8000 na 8000, na których wystawiona jest usługa. Łączy się z kontenerem „bazadanych”. Bazuje na obrazie budowanym w folderze aplikacja.
* „admin” - to kontener bazujący na obrazie phpmyadmin pozwalający na dostęp i modyfikowanie bazy danych przydatny podczas rozwoju aplikacji, domyślnie nie jest uruchamiany
* „bazadanych” – bazuje na obrazie mariadb, jako jedyny kontener jest tylko w sieci wewnętrznej, posiada połączony volumen podpięty pod folder danych bazy (\var\lib\mysql).

Obraz aplikacji zbudowany jest z dwóch warstw. Pierwsza warstwa bazuje na oficjalnym obrazie symfony w wersji 6.1. Druga warstwa odpowiada za kopiowanie kodu aplikacji do folderu w kontenerze.

1. Uruchomienie aplikacji

Wszystkie kontenery uruchamiamy będąc w folderze głównym projektu przy pomocy polecenia w cmd:

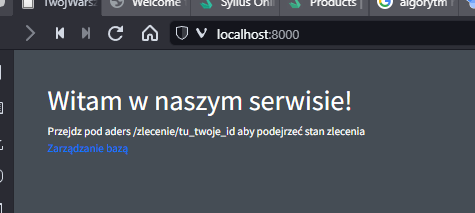
$ docker-compose up



Wszystkie potrzebne obrazy zostaną pobrane i zbudowane. Po uruchomieniu się wszystkich kontenerów aplikacja będzie dostępna poprzez adres:

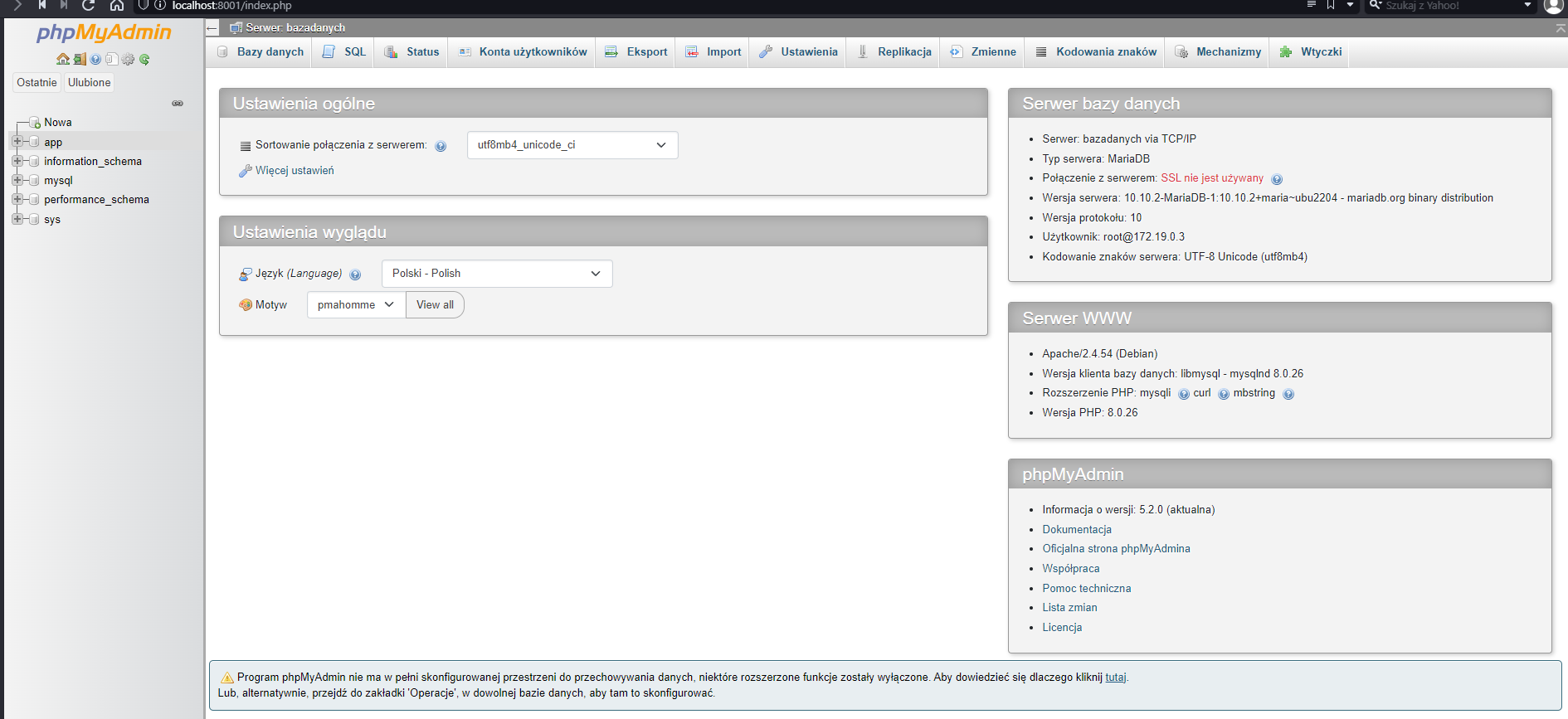
<http://localhost:8000/>

Uruchomiona witryna:

****

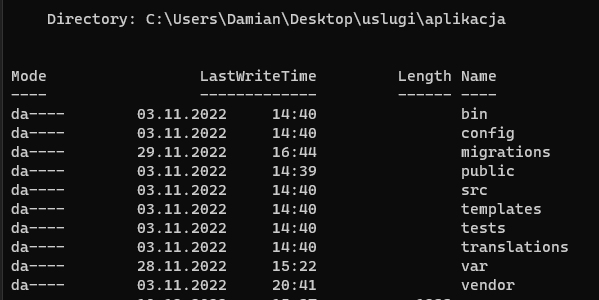
Jeżeli kontener do zarządzania bazą danych jest uruchomiony to możemy się z nim połączyć przez:

<http://localhost:8001/>

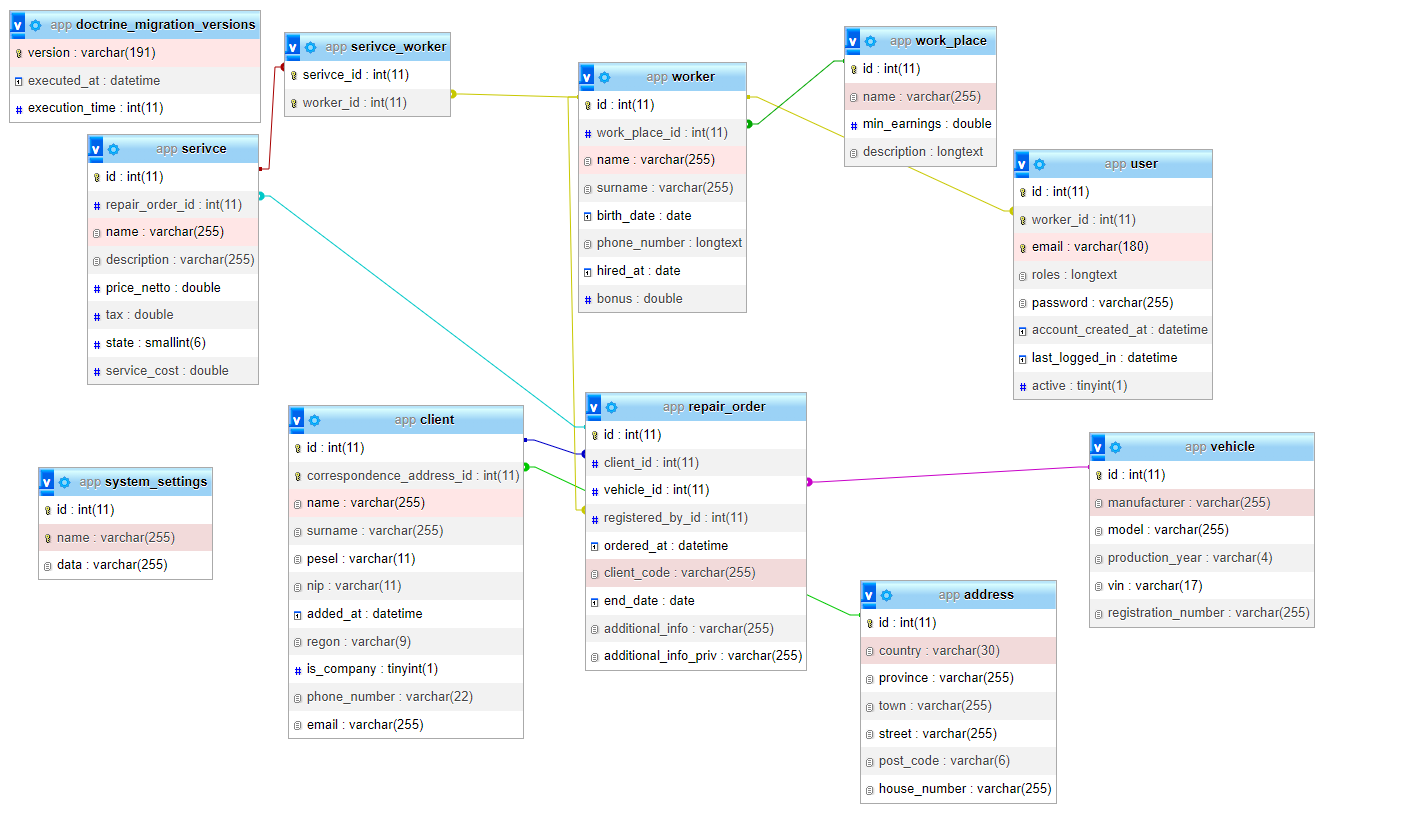
****

1. **Architektura aplikacji**

Aplikacja została napisana przy użyciu frameworka i wzorca MVC. Poniżej znajduje się struktura folderów aplikacji:



* Bin – zawiera pliki wykonywalne freamworka
* Config – wszystkie pliki konfiguracyjne
* Migrations – zawiera migracje bazy danych
* Public – folder ze statycznymi danymi jak grafiki, muzyka itp.
* Src – pliki źródłowe aplikacji
* Templates – szablony warstwy widoku
* Tests – testy aplikacji
* Transations – zawiera pakiety językowe
* Var – dane tymczasowe
* Vendor – dane zewnętrznych bibliotek

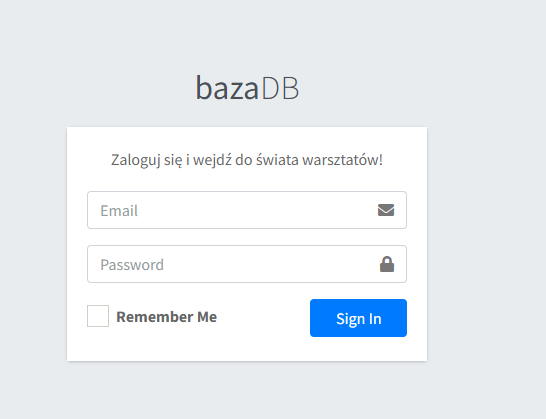
****

Rysunek . Schemat ERD

****

Rysunek . Tabele w bazie danych

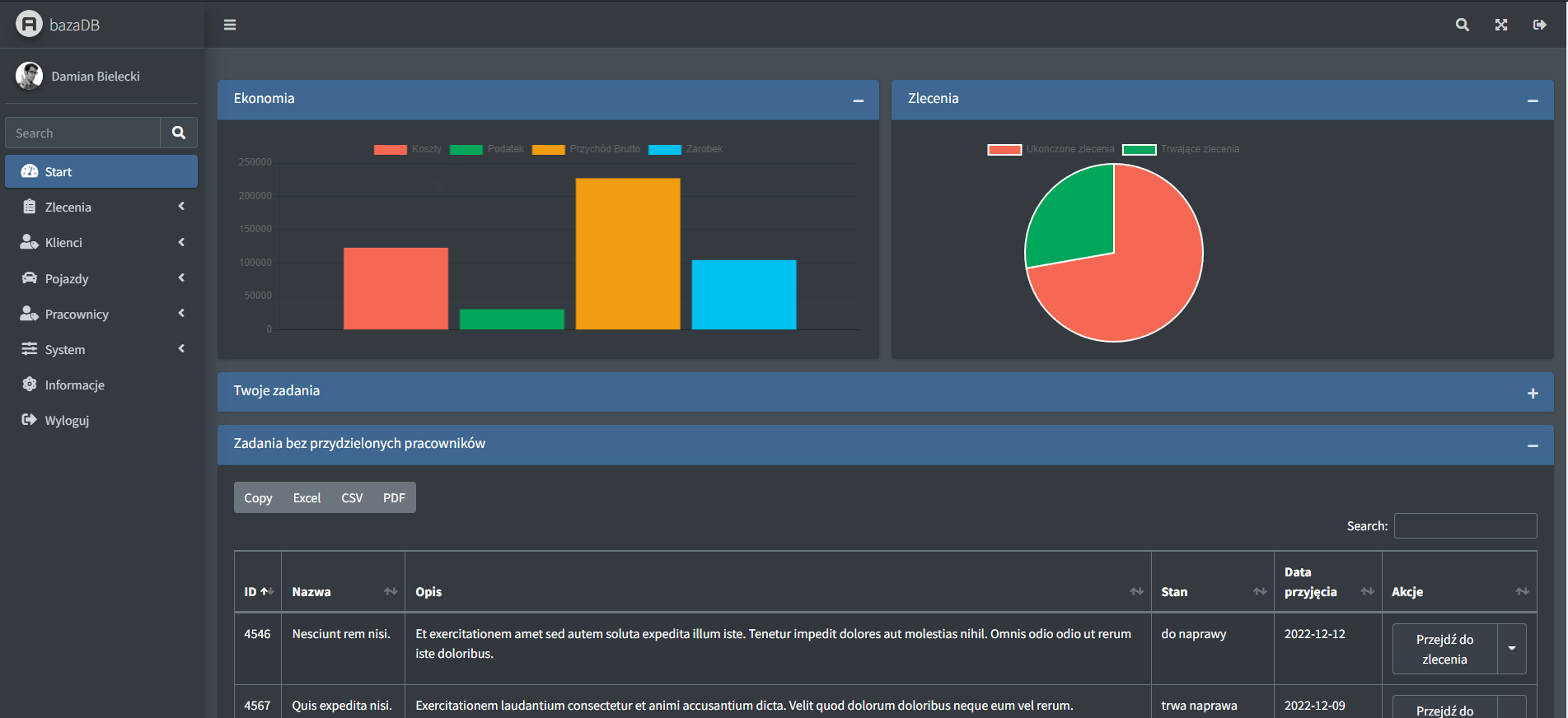
1. **Przedstawienie aplikacji** 
   1. **Logowanie do panelu zarządzania**

****

Rysunek . Panel logowania

Istnieją dwa domyślne konta pozwalające na zalogowanie się do aplikacji:

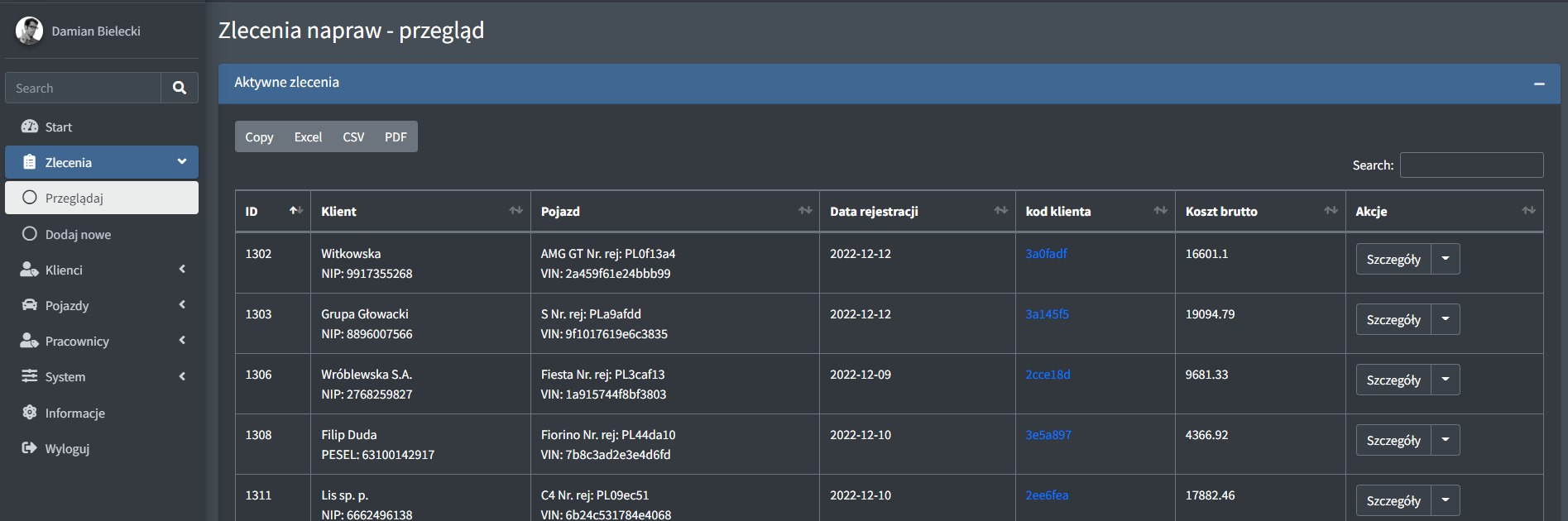
* Login: [damian@test.com](mailto:damian@test.com) hasło: admin
* Login: [marbor@test.pl](mailto:marbor@test.pl) hasło: marbor
  1. **Panel startowy**



Rysunek . Panel startowy po zalogowaniu

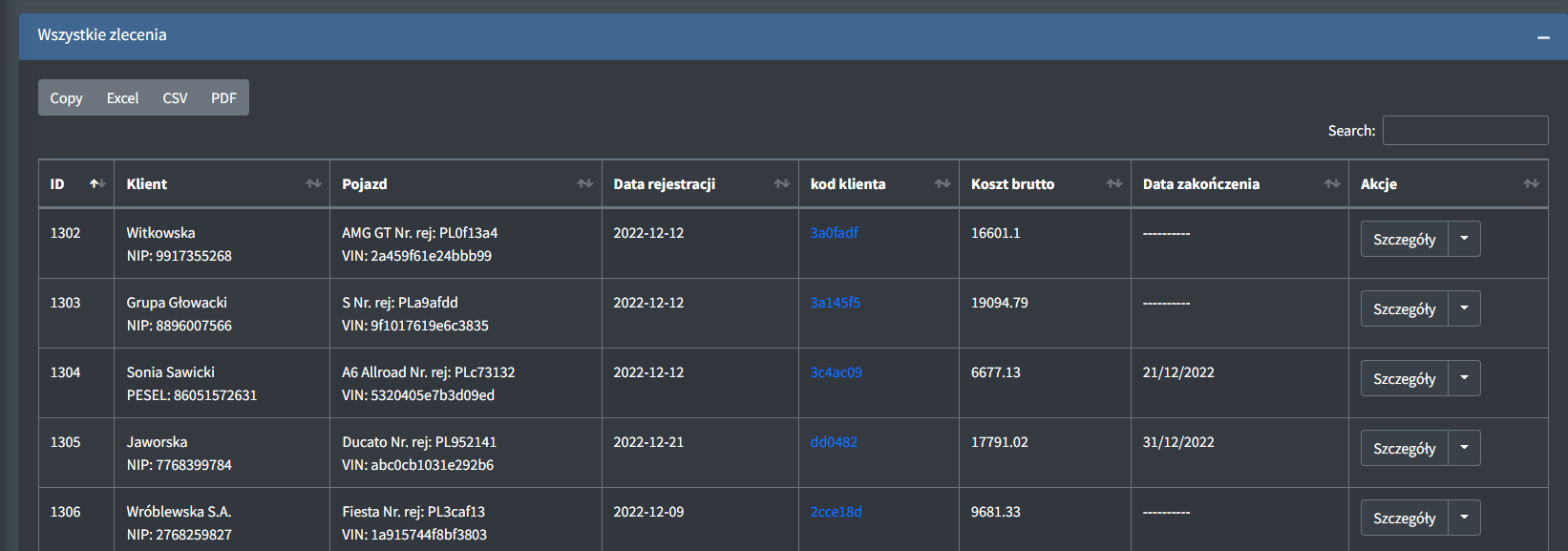
Widok domowy pokazuje użytkownikowi statystyki finansowe liczbę ukończonych i trwających zleceń. Poniżej widoczny jest zwinięty panel z zadaniami, w których uczestniczy zalogowany pracownik. Ostatnie okno pokazuje zadania bez przydzielonych pracowników.

* 1. **Zarządzanie zleceniami**



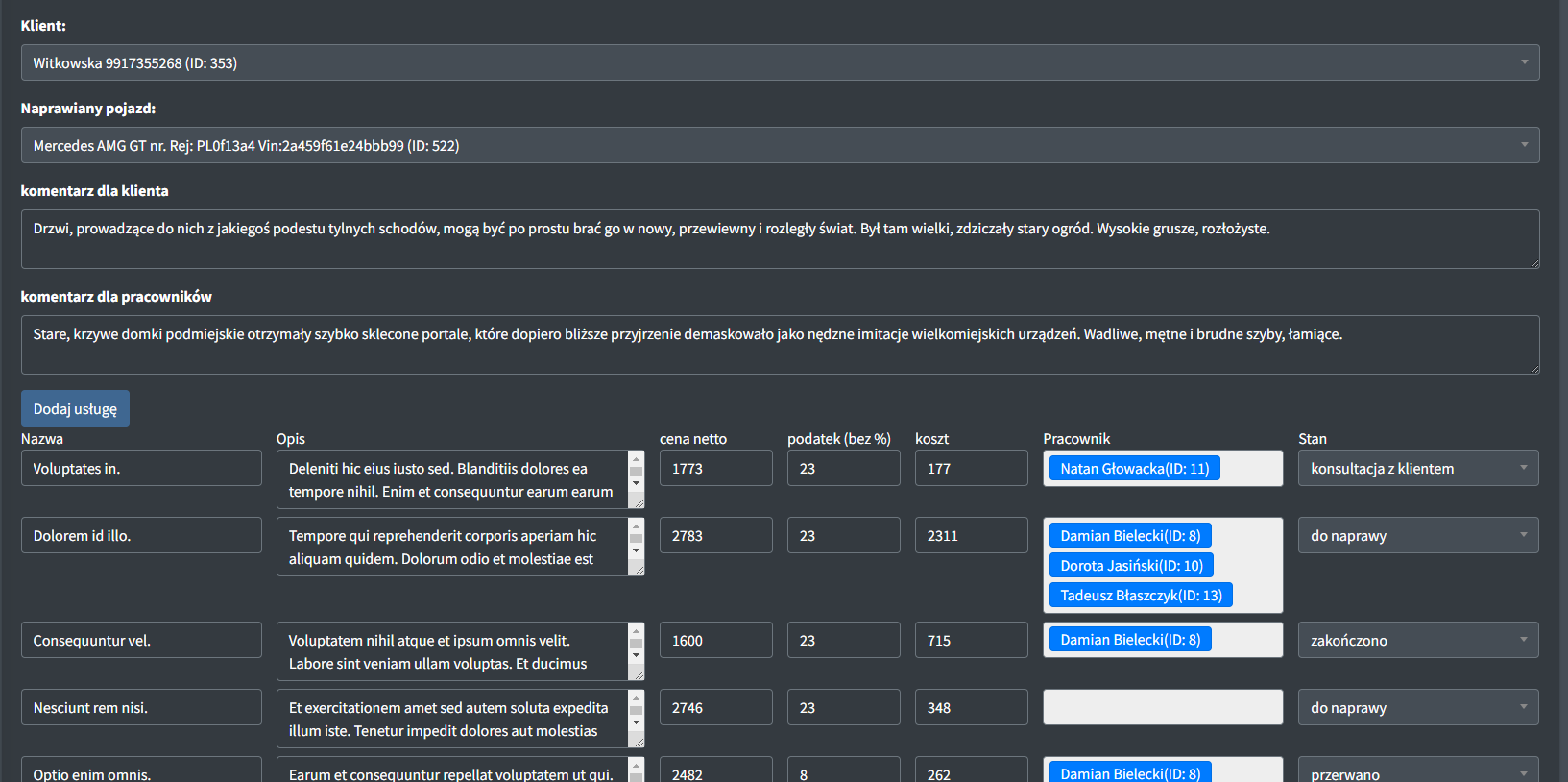
Rysunek . Aktywne zlecenia

Przegląd zleceń wyświetla dwa bloki, aktywne zlecenia i wszystkie zlecenia. Blok aktywne zlecenia wyświetla wszystkie trwające zlecenia. W zależności od typu klienta (klient prywatny lub firma) wyświetlane są różne dane klienta.



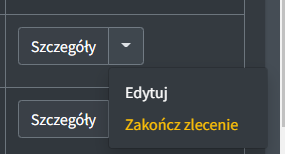
Rysunek . Wszystkie zlecenia

Blok wszystkie zlecenia pokazuje tabele ze wszystkimi wykonanymi zleceniami. Ukończone zlecenia posiadają datę zamknięcia. Pole kod klienta zawiera kod, który może zostać przekazany klientowi tak aby ten miał podgląd na stan wykonanych zadań. Pole jest równocześnie odnośnikiem, który przeniesie nas do widoku klienta.



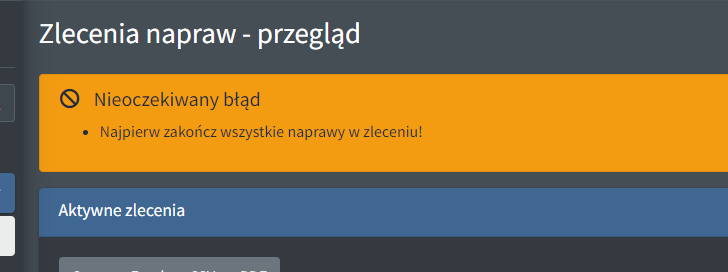
Rysunek . Edycja istniejącego zlecenia

Po wejściu w edycje zlecenia lub dodawanie nowego widzimy formularz pozwalający na wprowadzenie odpowiednich danych. W pierwszej kolejności należy wskazać klienta i naprawiany pojazd. W dalszej kolejności widoczny jest segment zadań, które możemy dodać poprzez przycisk dodaj usługę. Każda usługa ma przypisanych pracowników oraz stan.

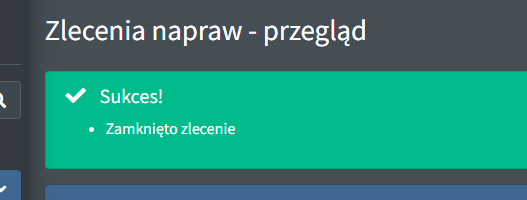


Rysunek . Opcje aktywnego zlecenia

Każde aktywne zlecenie można edytować lub z widoku listy zakończyć. Zakończenie jest możliwe jeżeli stan każdego zadania w zleceniu jest zakończony lub anulowany. Zakończone zlecenia nie mogą być edytowane.

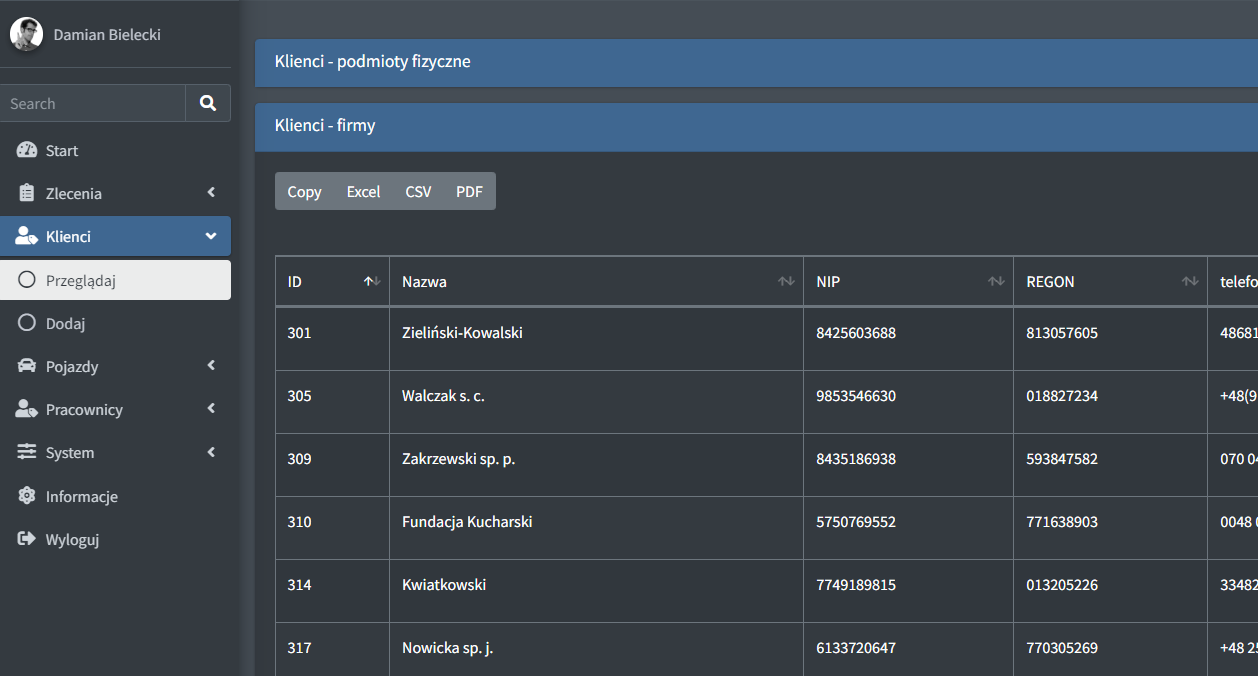


Rysunek . Nieudane zamknięcie zlecenia



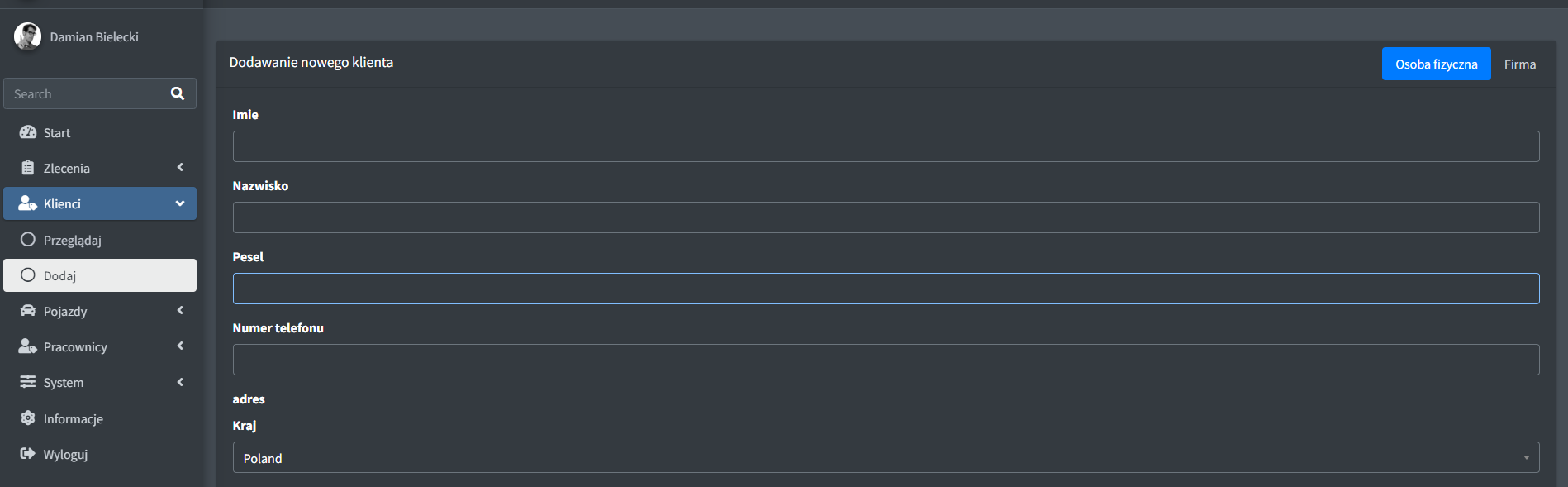
Rysunek . Zakończone zlecenie

* 1. **Zarządzanie klientami i samochodami**

****

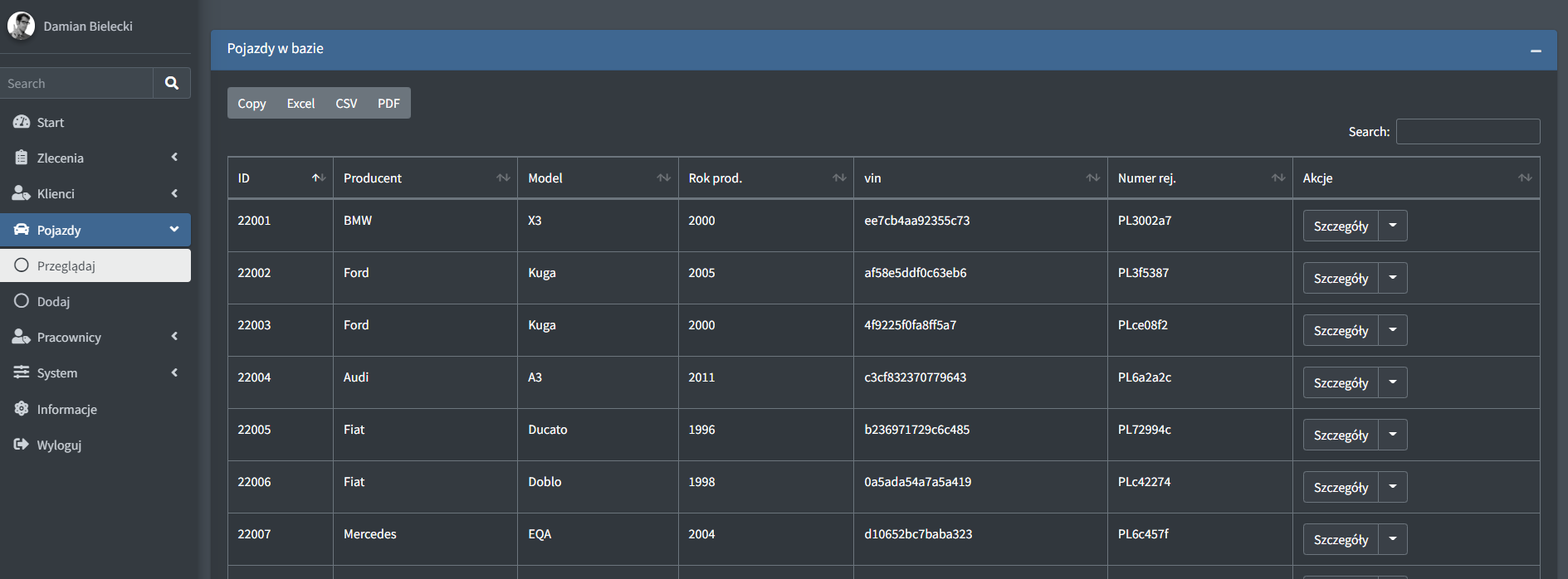
Rysunek . Przegląd klientów

Aplikacja rozróżnia klientów indywidualnych i firmowych. W zależności od typu wyświetlani są w różnych sekcjach. Dane każdego klienta można edytować.

****

Rysunek . Formularz dodawania klientów

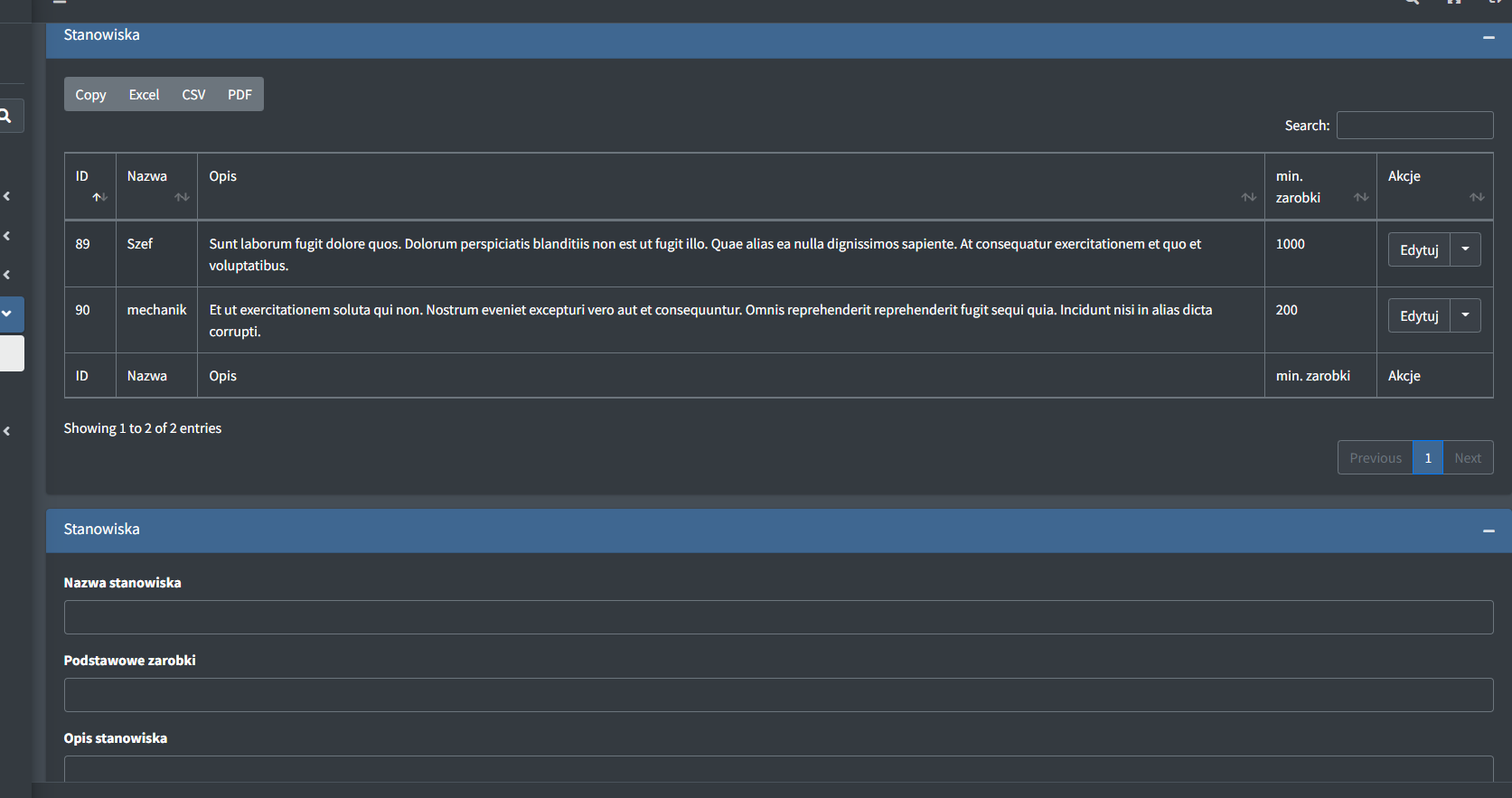
W zależności od przełącznika w prawym górnym rogu formularza wyświetlany jest inny formularz z odpowiednimi danymi. Po kliknięciu przycisku zapisz dane są walidowane i zapisywane w bazie danych.



Rysunek . Przegląd pojazdów w bazie

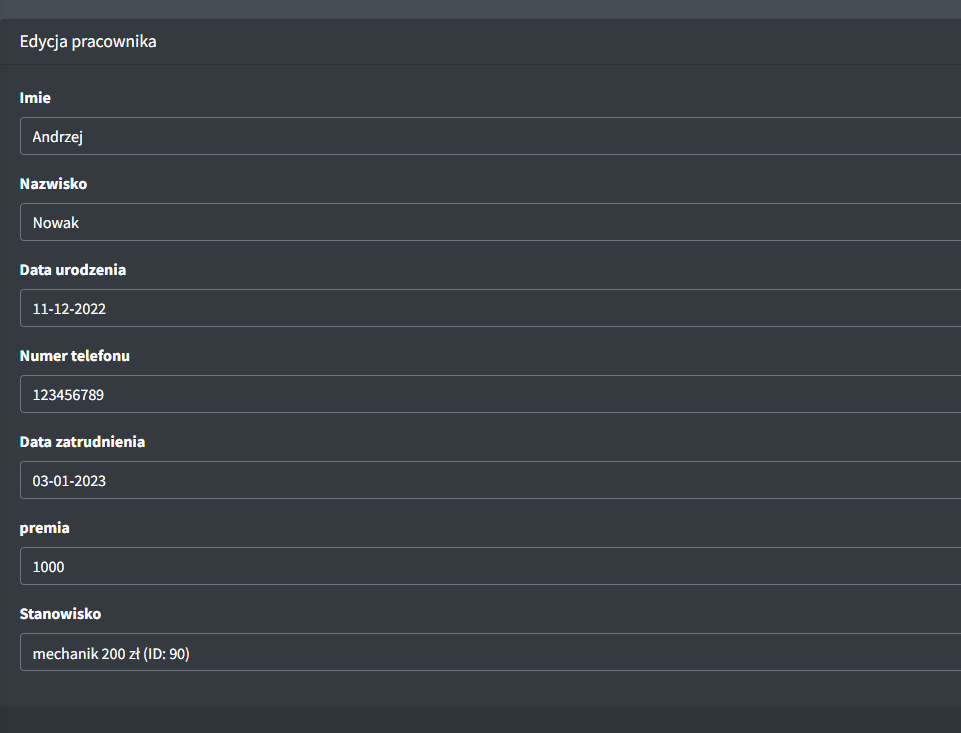
W przeciwieństwie do klientów podgląd pojazdów nie posiada żadnych szczególnych podziałów a dodawanie nowego pojazdu do bazy to zwykły formularz z kilkoma polami tekstowymi.

* 1. **Zarządzanie pracownikami i ich kontami**



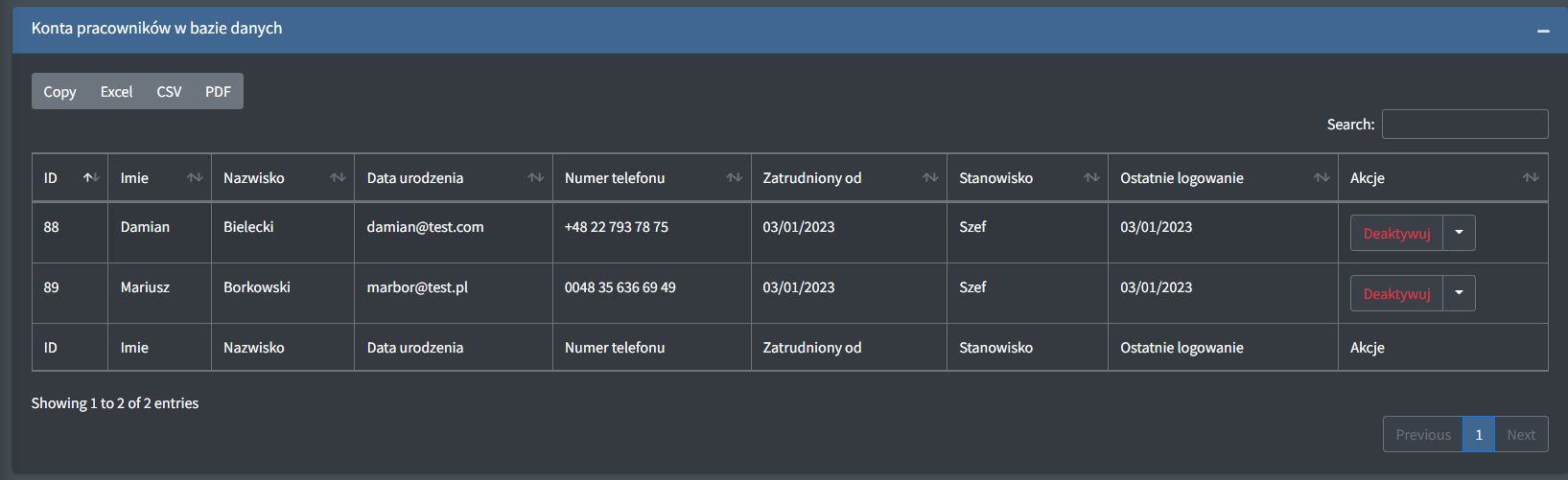
Rysunek . Stanowiska w firmie

Każdy utworzony pracownik ma przypisane stanowisko z opisem oraz minimalną pensją.



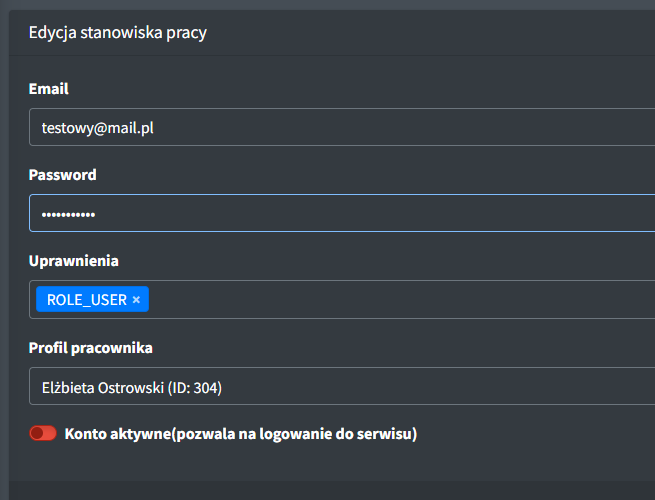
Rysunek . Profile pracowników

W tym momencie zarządzanie pracownikami jest bardzo proste i pozwala jedynie na ogólny podgląd jego zarobków oraz zadań w firmie. Pracownik posiada jedynie pole premia, podstawowe zarobki pobierane są z obejmowanego stanowiska. Profile pracowników służą również do tworzenia kont w serwisie. Zarządzanie kontami użytkowników znajduję się w sekcji Ustawienia -> Konta



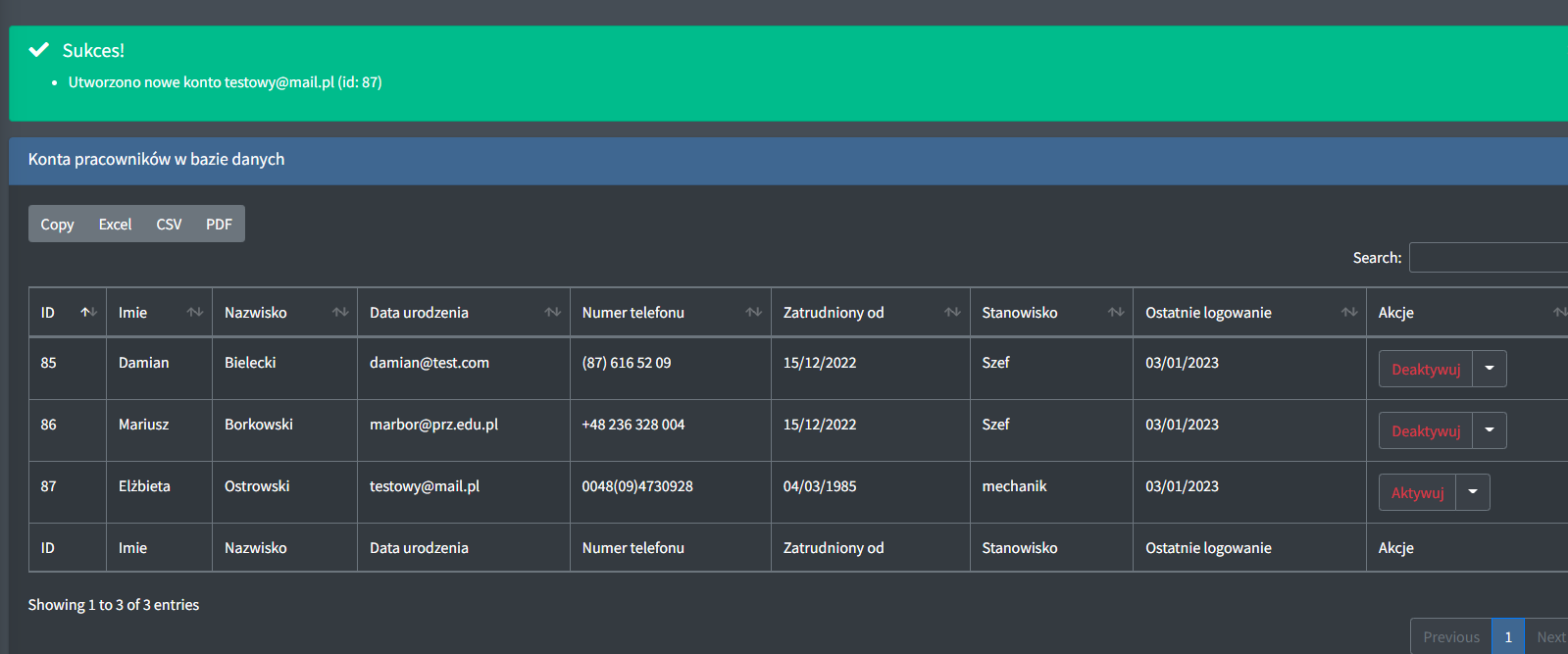
Rysunek . Wszyscy pracownicy w bazie

Panel ten wyświetla wszystkich utworzonych pracowników w bazie.



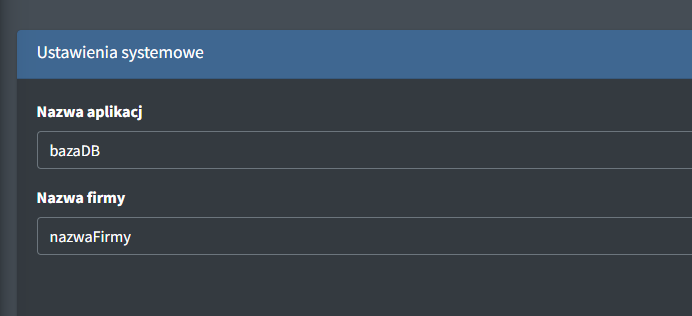
Rysunek . Edycja stanowiska pracy

Dodanie pracownika polega na podaniu jego maila(aktualnie nie wykorzystywany, w przyszłości reset hasła, powiadomienia itp.) hasła, uprawnień, oraz istniejącego profilu pracownika. Domyślnie konto jest wyłączone.



Rysunek . Dodany pracownik

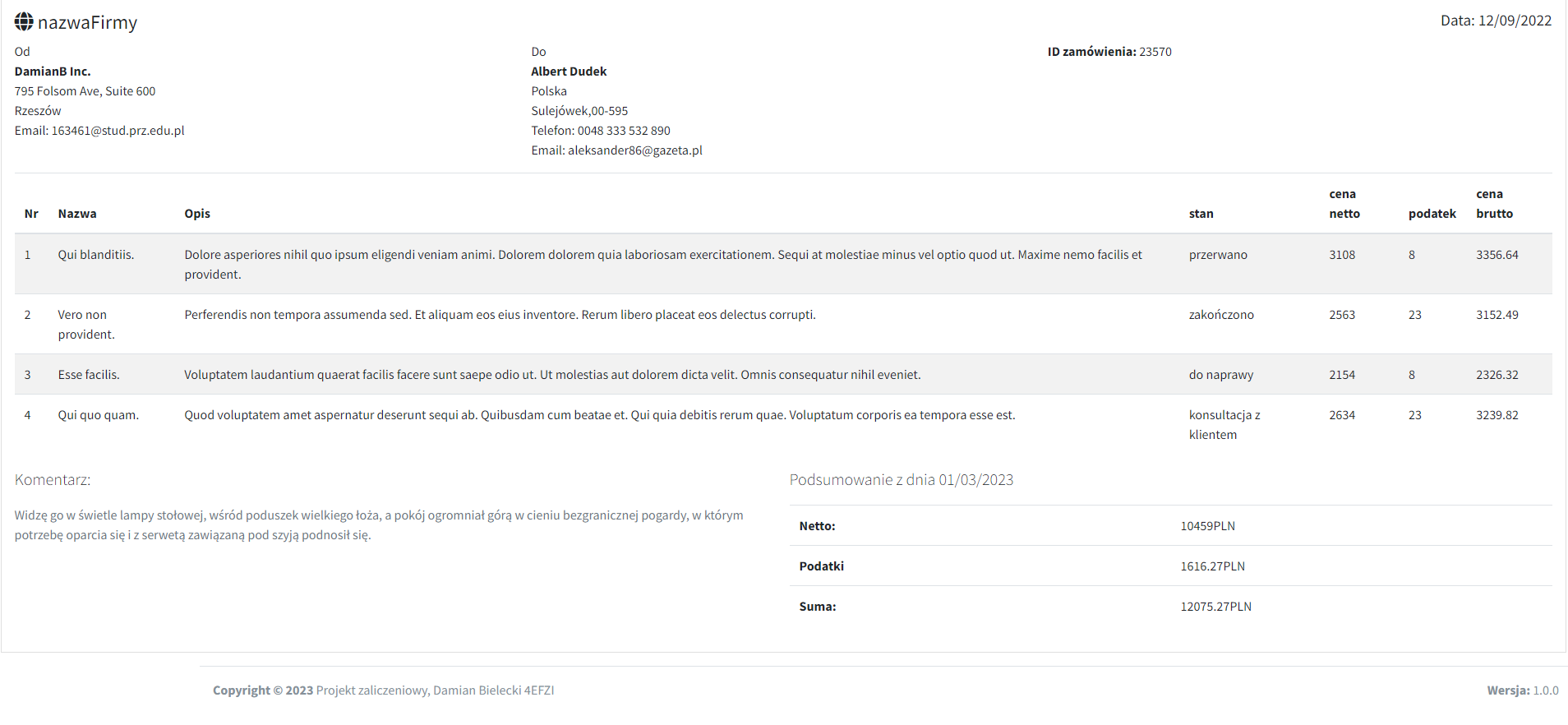
* 1. Ustawienia



Rysunek . Zmiana danych w aplikacji

W aplikacji został zaimplementowany prosty system ustawień. Na aktualnym etapie aplikacja jest bardzo prosta a więc nie posiada jeszcze wiele danych do zapisywania.

* 1. Podgląd zlecenia przez klienta



Rysunek . Podgląd zlecenia przez klienta

Każdy klient otrzymuje wygenerowany automatycznie unikalny krótki kod, który pozwala mu na podgląd stanu jego zlecenia. Klient widzi każde dodane zadanie, jego stan oraz koszt. Dodatkowo do dyspozycji pracowników jest pole „komentarz” pozwalający na dopisanie uwag o danym zleceniu.

1. **Podsumowanie**