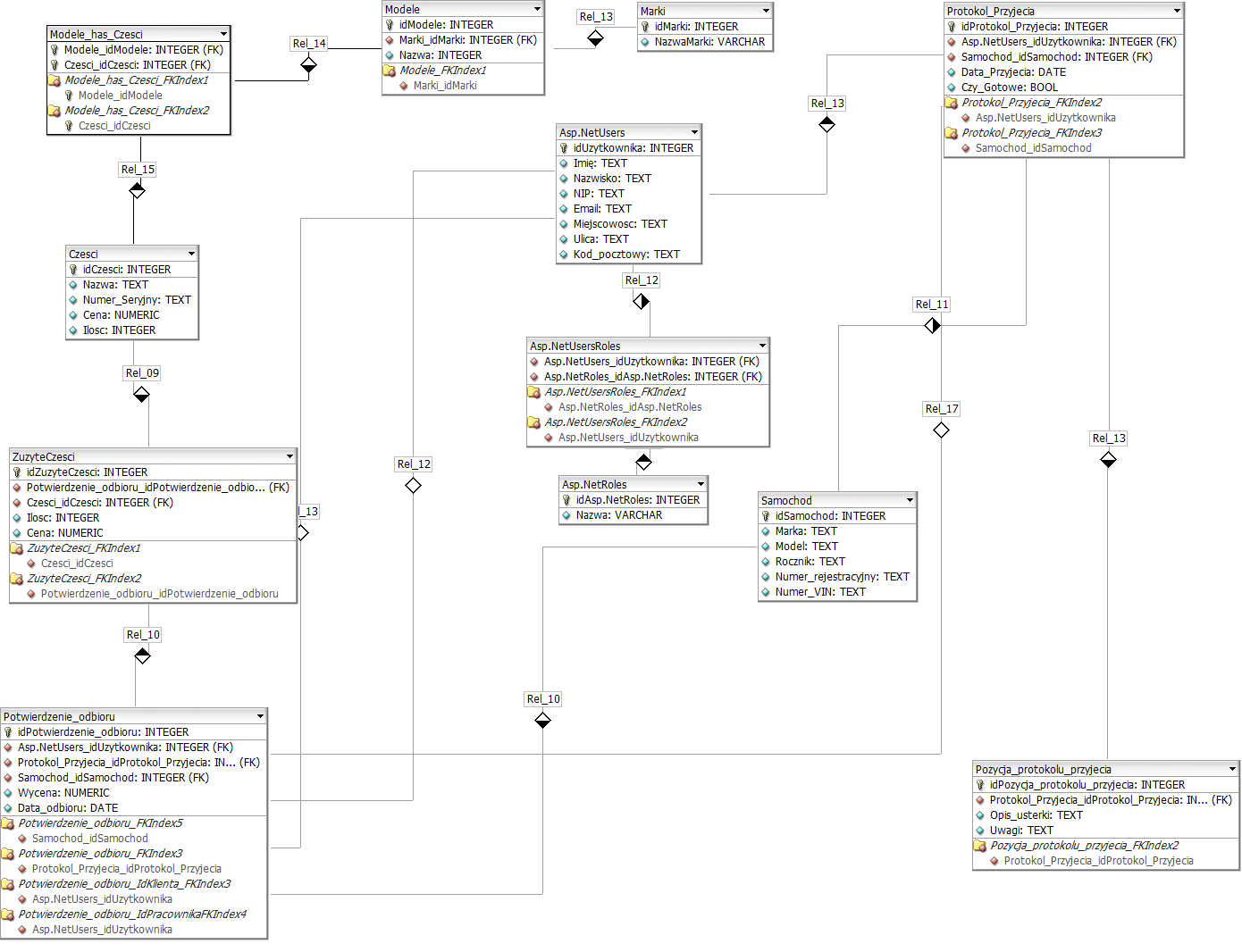
**Projekt z przedmiotu: Bazy Danych**

**Temat:** *Aplikacja do obsługi serwisu samochodowego*

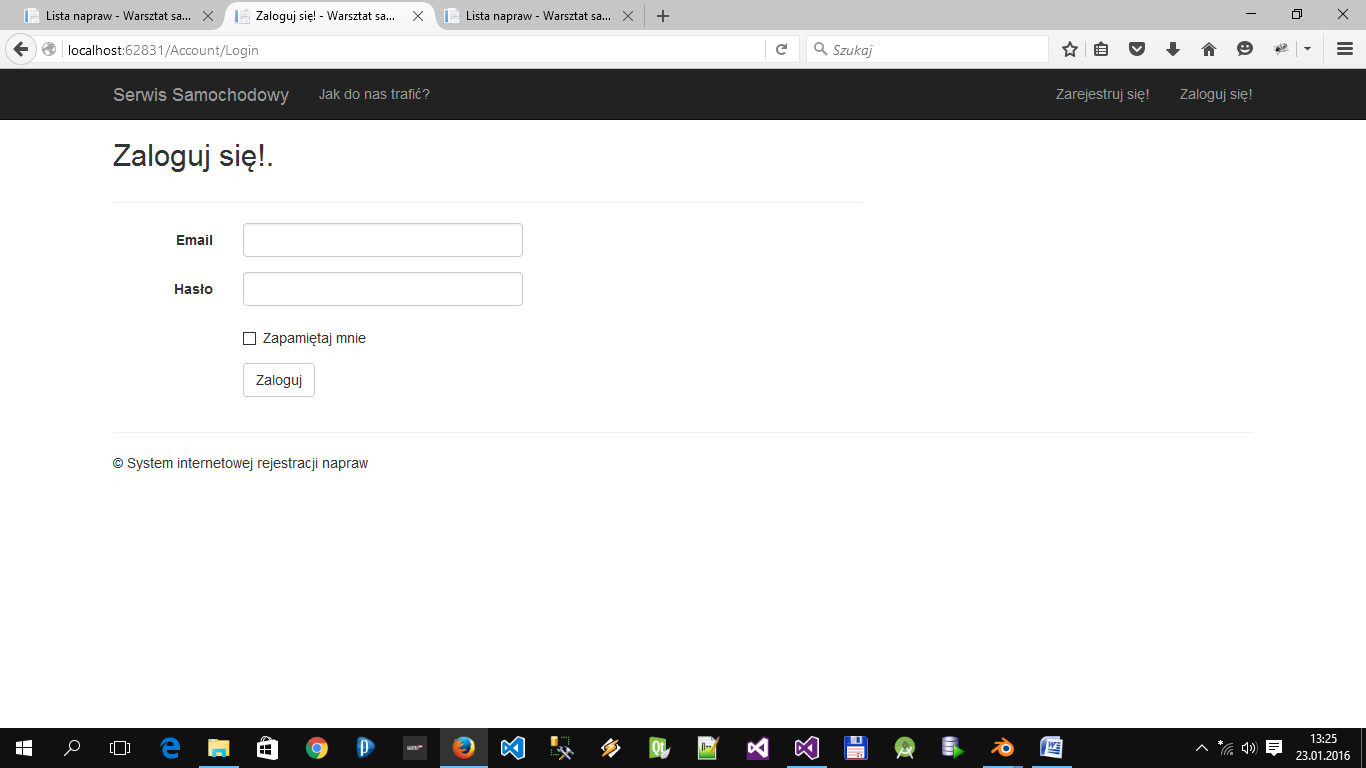
**Wykonali: Łukasz Szęszoł i Damian Naprawa**

**Diagram ERD**



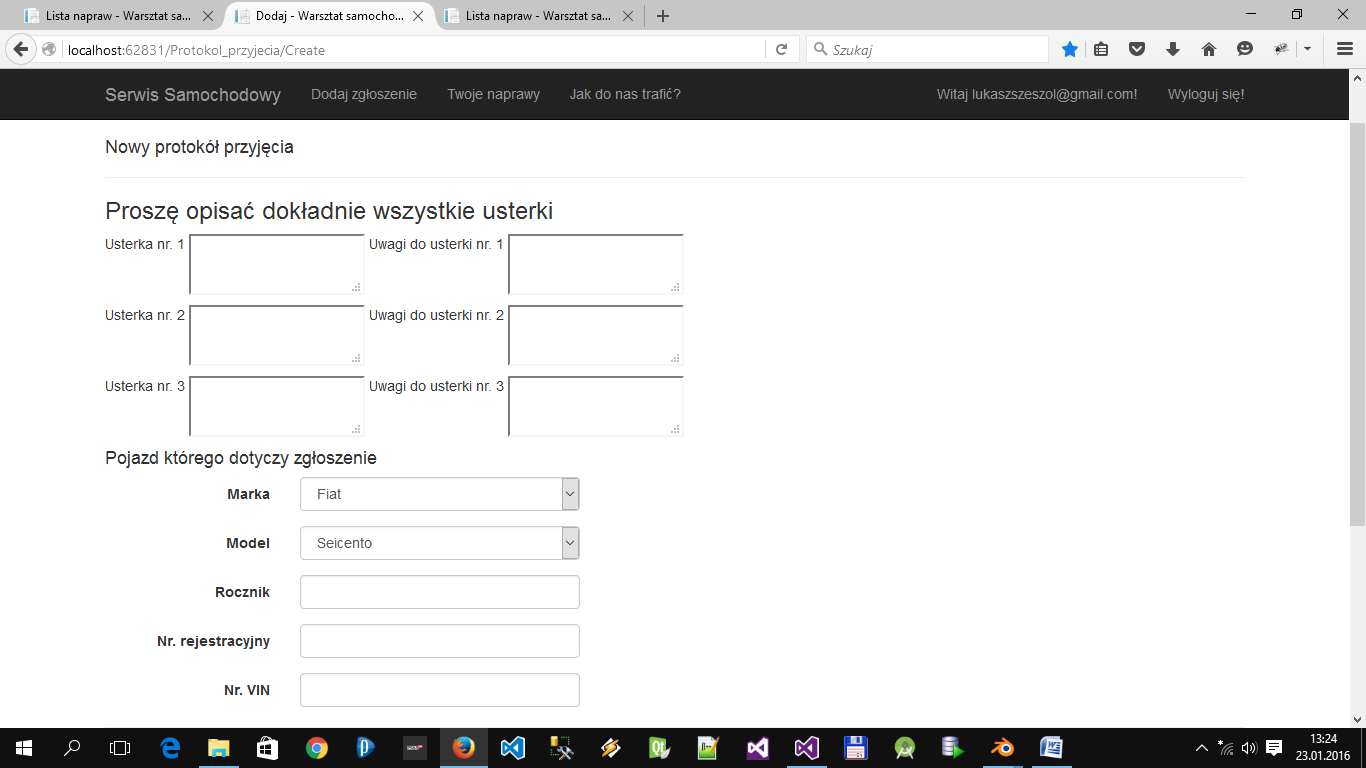
Aplikacja nosi nazwę **SerwisSAM** i służy do przyjmowanie i rozliczania napraw w serwisie samochodowym. W aplikacji występują 2 typy użytkowników. Klienci i Pracownicy serwisu.

Do aplikacji logujemy się poprzez poniższy ekran



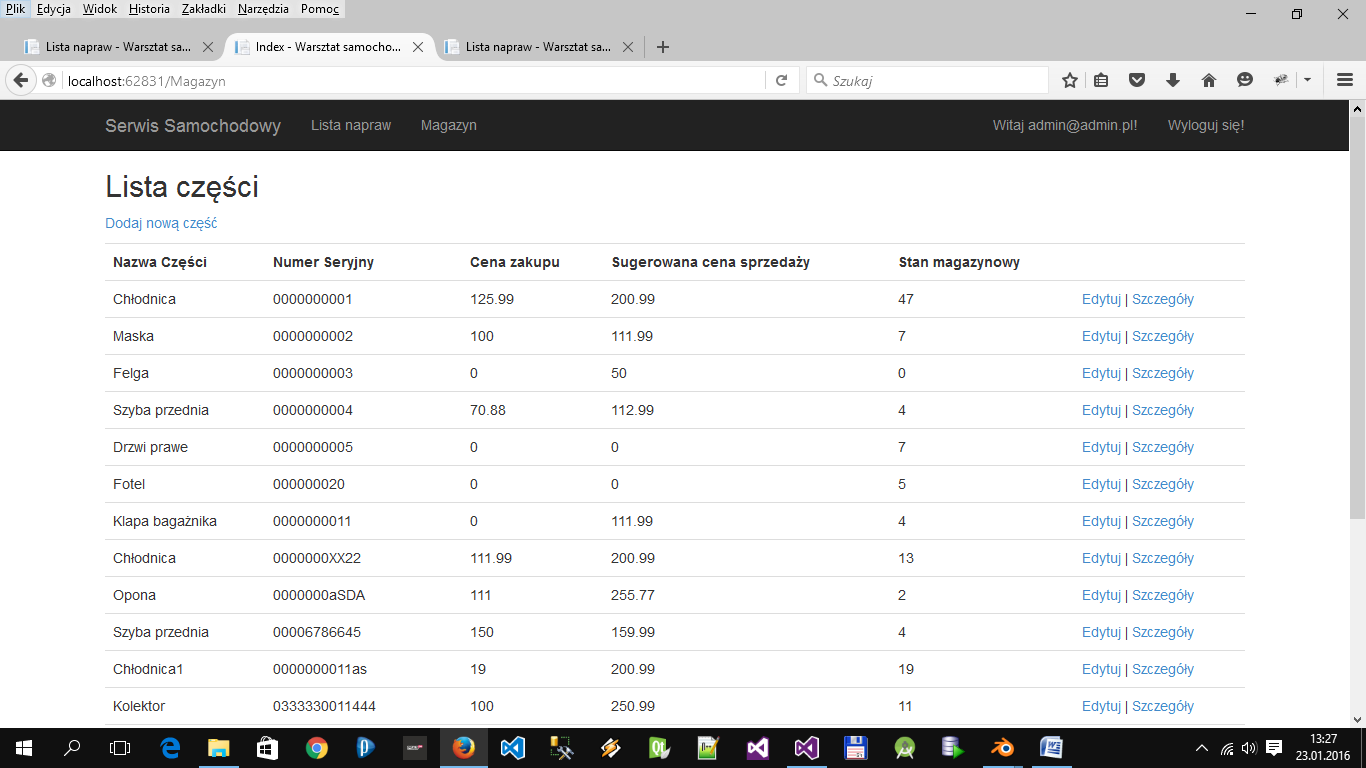
Klienci w pasku głównym mają opcję dodania nowej naprawy po wciśnięciu przycisku „Dodaj zgłoszenie” oraz przeglądaniu swoich dotychczasowych napraw i ich statusu.

*Ekran dodawania zgłoszenia*



Pracownicy mają podgląd na wszystkie dotychczasowe zlecenia napraw jak i na zawartość magazynu części.

*Ekran wyświetlający listę części*

**

Klient podczas dodawania nowego zgłoszenia naprawy może dodać nowy samochód poprze podanie jego danych lub wybrać taki który już istnieje w systemie poprzez wybranie go z listy rozwijalnej.

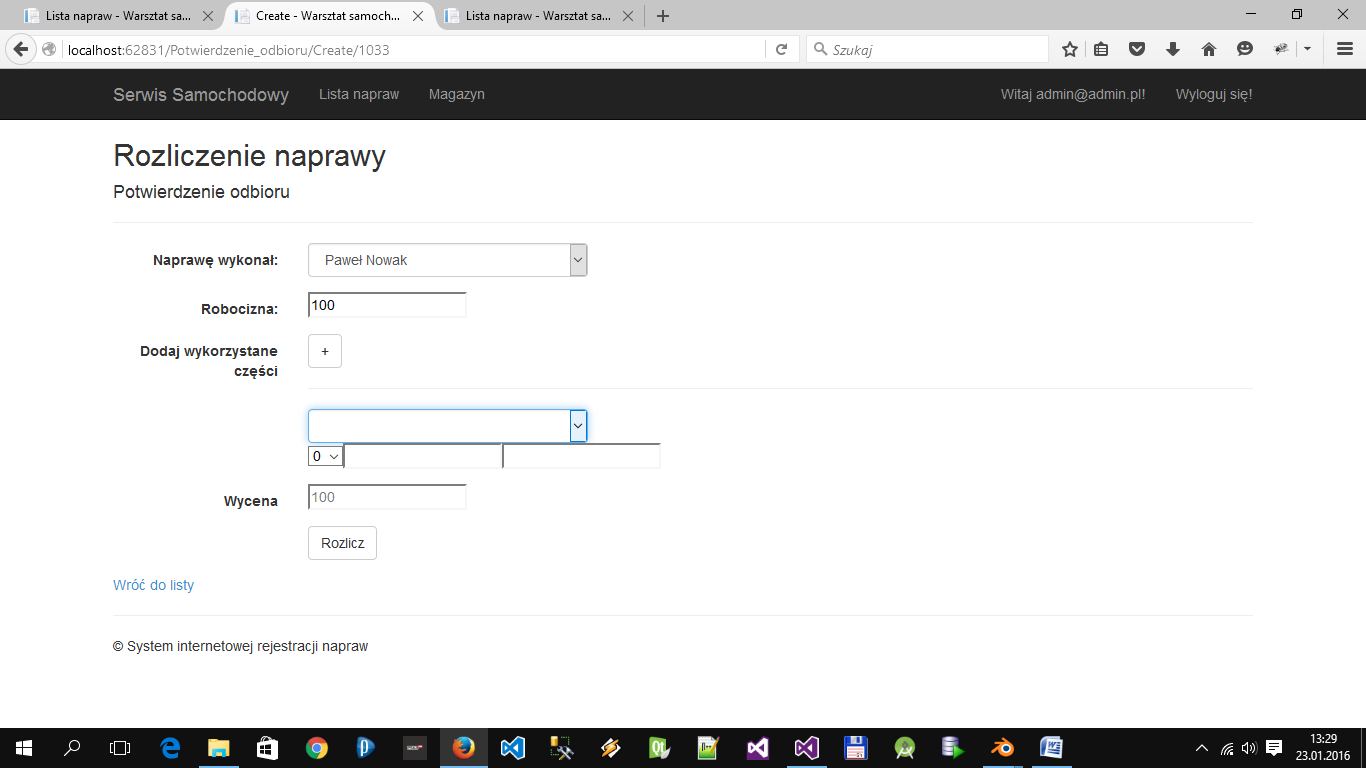
Dodatkowo system nie pozwala dodać ponownie samochodu którego numer VIN i/lub numer rejestracyjny jest już w systemie.

Po dodaniu nowego zgłoszenia pojawia się ono w zgłoszeniach klienta i ma status niegotowe.

Pojawia się ono także w zgłoszeniach widzianych przez pracowników serwisu. Widnieje obok niego przycisk gotowe.

Pracownik serwisu po zakończeniu naprawy naciska ten przycisk co zmienia status zgłoszenia na gotowe, klient może to zobaczyć po zalogowaniu w swoich naprawach, otrzymuje on również wiadomość email na adres który podał przy rejestracji.

Po kliknięciu przez pracownika serwisu przycisku Gotowe pojawia się przycisk Rozlicz który musimy nacisnąć w momencie odbioru przez klienta samochodu.



Po jego naciśnięciu przechodzimy do ekranu w którym wybieramy pracownika który dokonał naprawy jak i wyceniamy jego pracę, podajemy rodzaj i ilość zużytych części oraz ich ceny.

System automatycznie podpowiada nam części pasujące do danego modelu samochodu i mające niezerowy stan magazynowy.

System nie pozwoli nam także wybrać kilkukrotnie tej samej części ani wydać większej ilości danej części niż aktualnie dysponujemy na magazynie.

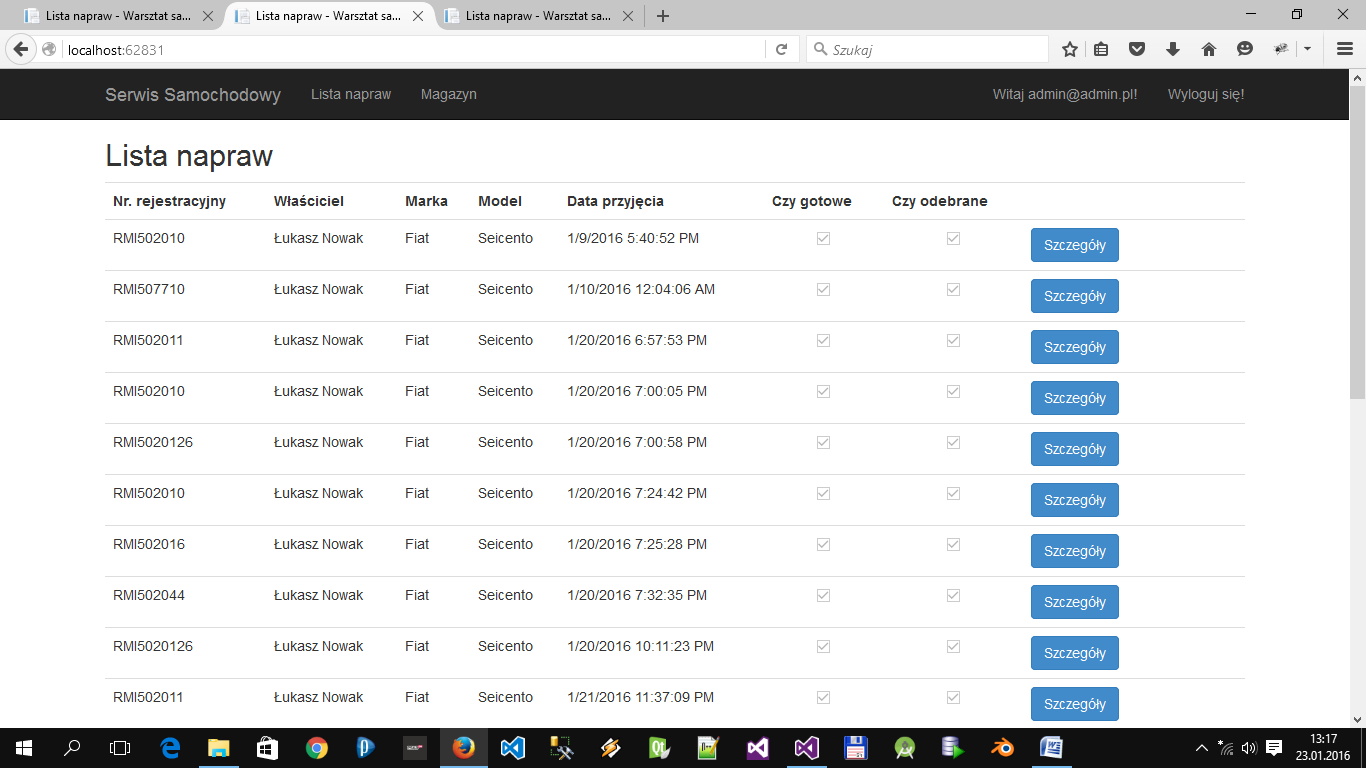
System podpowiada podczas wypełniania protokołu domyślną cenę sprzedaży danej części

System automatycznie podlicza nam i wyświetla dodawane pozycje podczas uzupełniania protokołu.

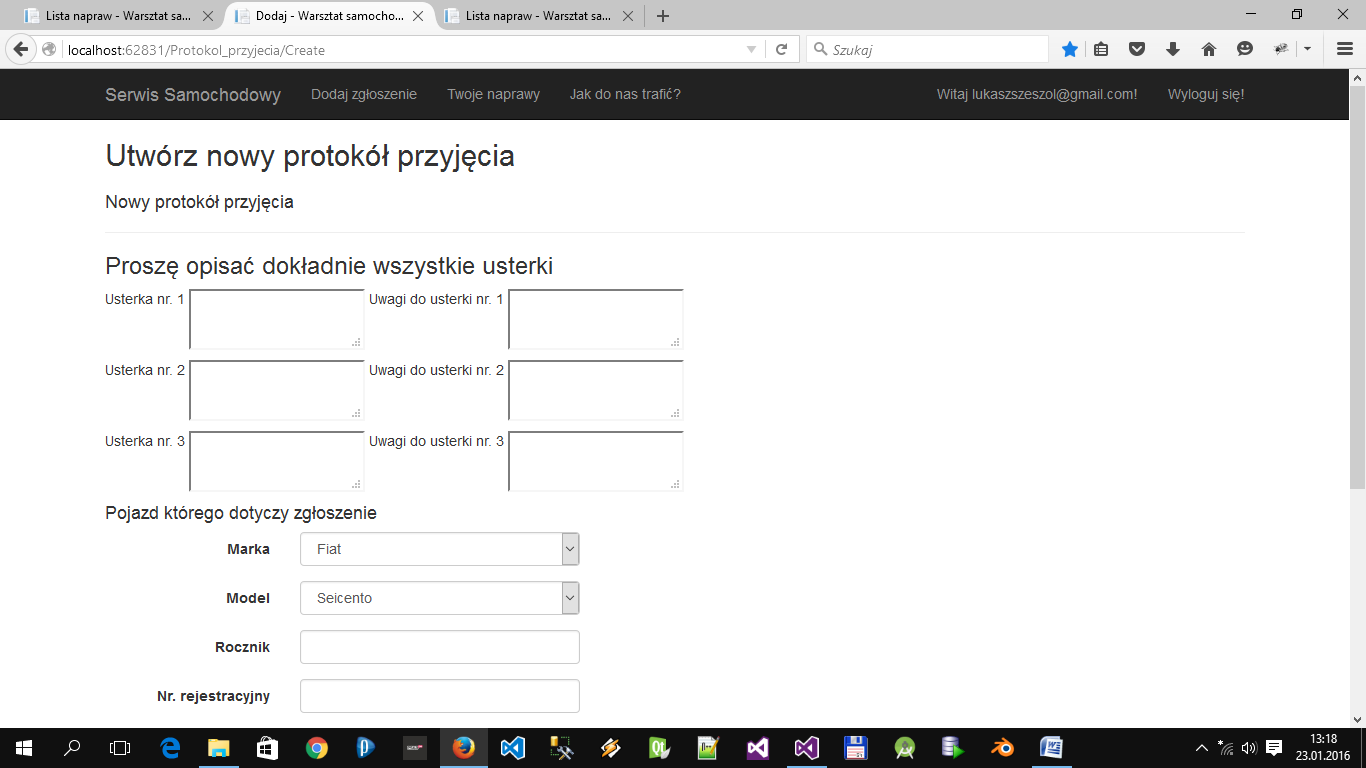
Po rozliczeniu naprawy system automatycznie redukuje stany magazynowe.

Pracowników ma również możliwość edycji stanów magazynowych, dodawania nowych części edycji cen itp. po wejściu w zakładkę magazyn.

Wygląd aplikacji z perspektywy zalogowanego pracownika serwisu.



Wygląd aplikacji z perspektywy zalogowanego klienta.



Aplikacja powstała w języku C#, przy użyciu platformy .NET.

Używaliśmy także Javascriptu, JQuery ,CSS w wersji 3.0 oraz HTML w wersji 5.

Dzięki Javascriptowi i JQuery uzyskaliśmy metody asynchroniczny sposób obsługi żądań nie ma potrzeby przeładowywania całej strony przy każdym żadaniu.

Użyliśmy wzorca projektowego MVC 5.0 rozwijanego przez firmę Microsoft.

**ASP.NET MVC** – platforma aplikacyjna do budowy [aplikacji internetowych](https://pl.wikipedia.org/wiki/Aplikacja_internetowa) opartych na wzorcu Model-View-Controller (MVC) oparta na technologii [ASP.NET](https://pl.wikipedia.org/wiki/ASP.NET).

Do najważniejszych cech platformy aplikacyjnej ASP.NET MVC należy:

* Wykorzystanie silnika ASP.NET pozwalającego na wykorzystanie wielu komponentów infrastrukturalnych oferowanych przez tę platformę, takich jak mechanizmy zarządzania stanem aplikacji, mechanizmy uwierzytelniania i autoryzacji, profile, cache, health monitoring itd.
* [Model programistyczny (API)](https://pl.wikipedia.org/wiki/Application_Programming_Interface) platformy jest mocno oparty na [interfejsach](https://pl.wikipedia.org/wiki/Interfejs_%28programowanie_obiektowe%29), pozwalając na łatwą rozbudowę, dorabianie i testowanie poszczególnych komponentów (obejmuje to również wsparcie dla podejścia TDD).
* Elastyczny mechanizm mapowania adresów [Uniform Resource Locator](https://pl.wikipedia.org/wiki/Uniform_Resource_Locator) pozwalający na łatwą budowę aplikacji według wzorca [Representational State Transfer (REST)](https://pl.wikipedia.org/wiki/Representational_State_Transfer), wykorzystujących czytelną dla użytkowników strukturę adresów URL. Mechanizm ten jest również bardziej przyjazny silnikom wyszukiwarek internetowych [Search Optimization Engine (SEO)](https://pl.wikipedia.org/wiki/Optymalizacja_dla_wyszukiwarek_internetowych).

# BIBLIOGRAFIA:

# J.Freeman ASP.NET MVC 4. Zaawansowane programowanie

# T.Rak, K.Żydzik C# 6.0 i MVC 5. Tworzenie nowoczesnych portali internetowych