

Projekt aplikacji

Szkielet aplikacji

Podział na moduły:

1. Moduł zarządzania klientami i ich samochodami

Możliwe jest dodawanie nowych klientów i samochodów (dostępnych/obsługiwanych modeli/wersji) z nimi powiązanych oraz wyświetlanie informacji o nich zapisane w bazie.

2. Moduł zarządzania zleceniami od klientów na oferowane usługi.

Przy dodawaniu nowego zlecenia wybierany jest klient, pojazd, rodzaj usługi, jej wstępny koszt oraz ewentualne uwagi od klienta. W trakcie wybierania klienta możliwe jest dodanie nowego co za tym idzie dodanie nowego pojazdu dla klienta, bądź tylko nowego pojazdu dla klienta.

3. Moduł zarządzania wykonaniem zamówionych usług.

Przy ustaleniu realizacji, przypisywany jest pracownik, a następnie używane części do realizacji usługi na podstawie których rośnie koszt wykonanej pracy.

4. Moduł zarządzający pracownikami w warsztacie.

Możliwe jest dodawanie nowych pracowników oraz wyświetlania ich danych. Pracownicy mogą obejmować stanowiska m.in. Mechanik, Lakiernik, Tapicer, Księgowy itp.

5. Moduł zarządzający częściami.

Możliwe jest monitorowanie ilości i rodzaju na dostępnych części, ich przeglądanie oraz wyszukanie części pasujących do danego pojazdu lub części danego typu oraz zamawianie brakujących u dostawców.

6. Moduł zaopatrzeniowy

Możliwe jest przeglądanie ofert dostawców, dodanie dostawców lub ich ofert, składanie zamówień na towary i wprowadzanie nowych dostaw na stan magazynu.

Podział realizacji funkcjonalności

Serwer:

W wszystkich tabelach, gdzie występuje dodany, sztuczny unikalny identyfikator, serwer dba o zachowanie jego unikalności i generuje kolejne identyfikatory dla wstawianych danych

Serwer realizuje następującą funkcjonalność w poszczególnych modułach:

1. Moduł zarządzania klientami i ich samochodami
 - Serwer udostępnia tzw. perspektywy aby umożliwić wygodniejszy dostęp do klientów będących osobami prywatnymi i firmami
2. Moduł zarządzania zleceniami od klientów na oferowane usługi.
3. Moduł zarządzania wykonaniem zamówionych usług.
 - Serwer dba o automatyczne zwiększanie kosztów realizacji usługi, gdy zostaje w niej wykorzystana jakaś część z magazynu
4. Moduł zarządzający pracownikami w warsztacie.
 - Serwer cyklicznie (raz w miesiącu) oblicza wydatki na utrzymanie pracowników oraz przeprowadza wyliczenia składek na ubezpieczenia.
5. Moduł zarządzający częściami.
 - możliwe jest włączenie na serwerze opcji automatycznego zamawiania wybranych części od danego dostawcy, gdy jego stan spadnie poniżej ustalonego poziomu
6. Moduł zaopatrzeniowy
 - po wprowadzeniu zamówienia system komunikuje się z dostawcami i przekazuje im zamówienia

Klient:

Aplikacja kliencka odpowiada za interakcje z użytkownikiem oraz prezentację mu danych w przejrzystej postaci.

Ponad to w poszczególnych modułach aplikacja zapewnia następującą funkcjonalność:

1. Moduł zarządzania klientami i ich samochodami
 - Aplikacja kliencka umożliwia dodanie nowych klientów i samochodów do bazy oraz pozwala na przeglądanie już wprowadzonych klientów i samochodów

2. Moduł zarządzania zleceniami od klientów na oferowane usługi.
 - Aplikacja kliencka umożliwia wystawienie nowego zlecenia wykonania usługi. Ponadto to możliwe jest też dodanie klienta i samochodu.
3. Moduł zarządzania wykonaniem zamówionych usług.
 - Aplikacja kliencka umożliwia zarządzanie pracownikami tj. przypisywanie ich do zadań oraz prowadzenie ewidencji części użytych podczas realizację danej usługi.
4. Moduł zarządzający pracownikami w warsztacie.
 - Aplikacja kliencka umożliwia dostęp do danych osobowych pracowników, ich edycję oraz dodawanie i usuwanie pracowników i stanowisk.
5. Moduł zarządzający częściami.
 - Aplikacja kliencka umożliwia przegląd stanu magazynu oraz parametrów danych części.
6. Moduł zaopatrzeniowy
 - moduł umożliwia przyjmowanie nowych dostaw oraz składanie zamówień na wybrane części u wybranych dostawców. Możliwa jest ponadto edycje oraz dodawanie i usuwanie dostawców oraz ich ofert (cenników)

Sposób realizacji wymuszenia integralności danych

Serwer:

Część serwerowa odpowiada za zapewnienie więzów unikalności tj. aby wybrane wartości atrybutów nie mogły wystąpić w danym kontekście więcej niż dana liczba razy. Ponadto serwer zapewnia istnienie odpowiadających sobie elementów np. samochód nie może istnieć w aplikacji jeśli nie posiada właściciela. Aby zapewnić podane więzy w definicji bazy danych dodano odpowiednie więzy na pola poszczególnych tabel, a także uwzględniono te więzy w krotności związków pomiędzy tabelami oraz ich dobrowolności lub obligatoryjności. Decyzję o wymuszeniu tych więzów integralności zostały podyktowane wydajnością – nie można tego zrobić wydajniej w aplikacji klienckiej gdyż wymagało by to kilku odwołań do bazy danych oraz prostotą implementacji takiego rozwiązania.

Klient:

Aplikacja kliencka odpowiada za więzy integralności w zakresie poprawności formatowania wprowadzanych danych, np. zapewnia poprawną składnię adresu e-mail lub NIP. Kontrola tych więzów odbywa się poprzez użycie odpowiednich wyrażeń regularnych. Sprawdzanie formatu danych jest wykonywane w aplikacji klienckiej, gdyż błędny format danych jest najczęściej wynikiem pomyłki użytkownika, która zostanie za chwile poprawiona i zostanie podjęta kolejna próba wprowadzenia danych. Taka charakterystyka operacji wymaga dużej interaktywności aplikacji i szybkiego

sprawdzania poprawności formatu wprowadzanych danych, dlatego powinna być zaimplementowana jak najbliżej użytkownika czyli w aplikacji klienckiej.

Propozycje indeksów

Po analizie funkcjonalności aplikacji zauważono, iż często wykonywane będą następujące operacje, które można przyspieszyć dodając odpowiednie indeksy:

- wyszukanie wszystkich parametrów danej wersji samochodu – indeks na polu *vversion_id* w tabeli *Vehicle_Parameter*
- wyszukanie wszystkich parametrów danej części – indeks na polach *catalog_number* i *name* w tabeli *Part_Parameter*
- wyświetlanie wszystkich wersji wybranego modelu samochodu – indeks na polach *model* i *brand* w tabeli *Vehicle_Version*
- wyszukanie wszystkich pojazdów danego klienta – indeks na polu *client_id* w tabeli *Vehicles*
- wyszukanie wszystkich produktów (konkretnych części) pasujących jako pewna część do pewnego modelu samochodu – indeks na polu *part_number* w tabeli *Part*
- wyszukanie wszystkich sztuk części danego towaru - indeks na polach *catalog_number* i *name* w tabeli *Part_Instance*

Opis transakcji

Wybrany moduł:

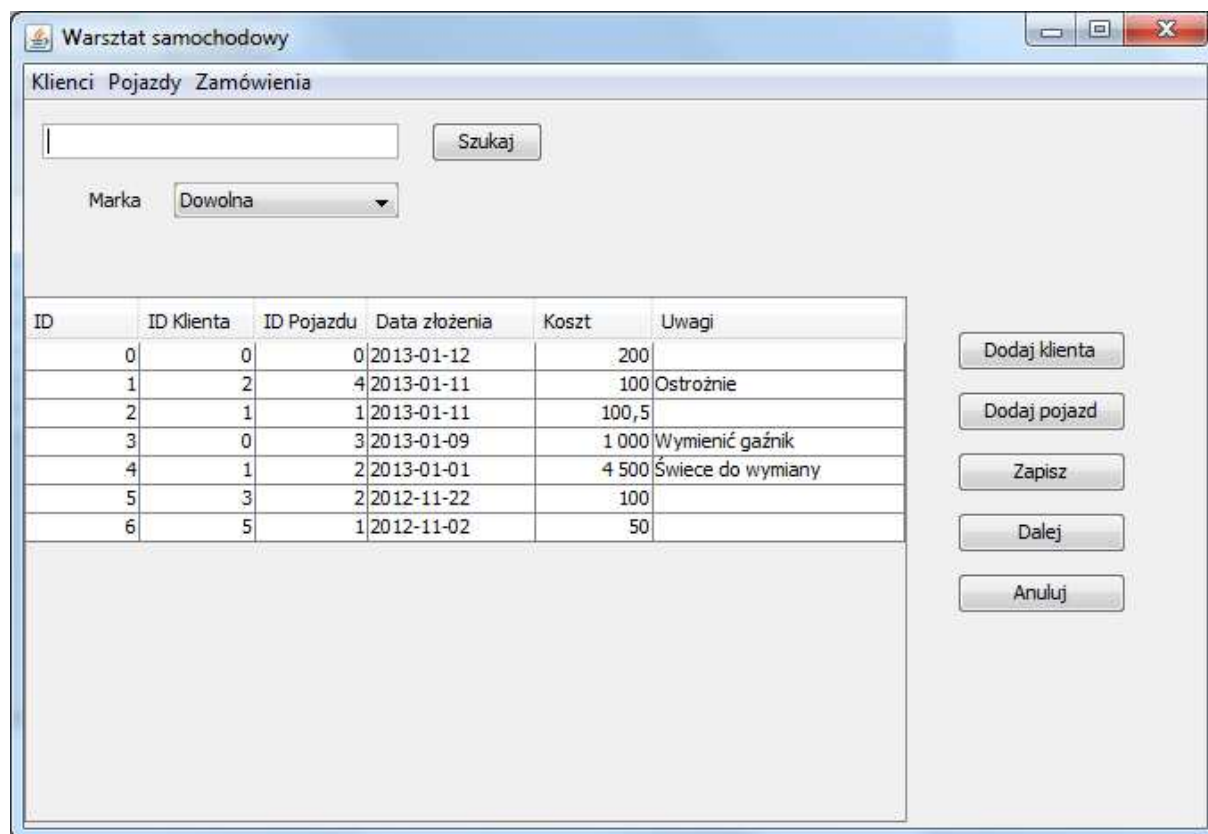
2. Moduł zarządzania zleceniami od klientów na oferowane usługi

Opis

Transakcja rozpocznie się w momencie wybrania opcji dodania nowego zamówienia. Polegać będzie na wyborze klienta z listy klientów bądź dodanie całkowicie nowego do bazy danych. Następnie zostanie wybrany pojazd wybranego klienta lub zostanie dodany całkowicie nowy pojazd - obligatoryjne w przypadku dodania klienta w danej transakcji. Kolejnym krokiem będzie wybór oferowanej usługi z domyślnym kosztem i / lub zmianie wstępnego kosztu usługi oraz sporządzeniu uwag do zamówienia. Finalnym ekranem będzie wyświetlenie wybranych danych celem zatwierdzenia transakcji. Będzie możliwość anulowania transakcji na każdym z etapów.

Projekt GUI oraz hierarchia menu

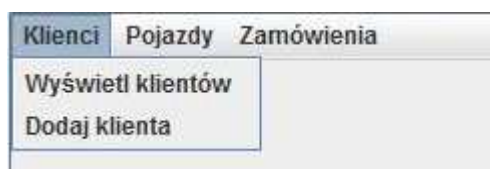
Graficzny interfejs użytkownika został zaprojektowany w programie NetBeans z użyciem standardowego, dobrze znanego dla użytkownika wyglądu w stylu Java.



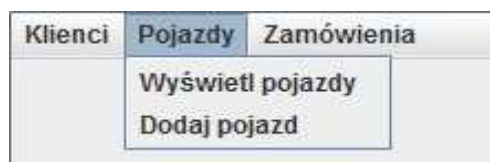
Obsługa interfejsu jest bardzo intuicyjna.

Pasek menu został podzielony na rozwijane menu. Każdy z nich odnosi się do innego bytu, który jest obsługiwany w tym module.

Menu klienci:



Menu pojazdy:



Menu zamówienia:



Przy pomocy tych menu możemy przejść do paneli umożliwiających nam bezpośrednio wykonanie wszystkich operacji, jakie są dla niego dostępne w tym module, lub też bezpośrednio do operacji dodania nowego elementu danego typu.