**Projekt: Program wspierający pracę wypożyczalni pojazdów**

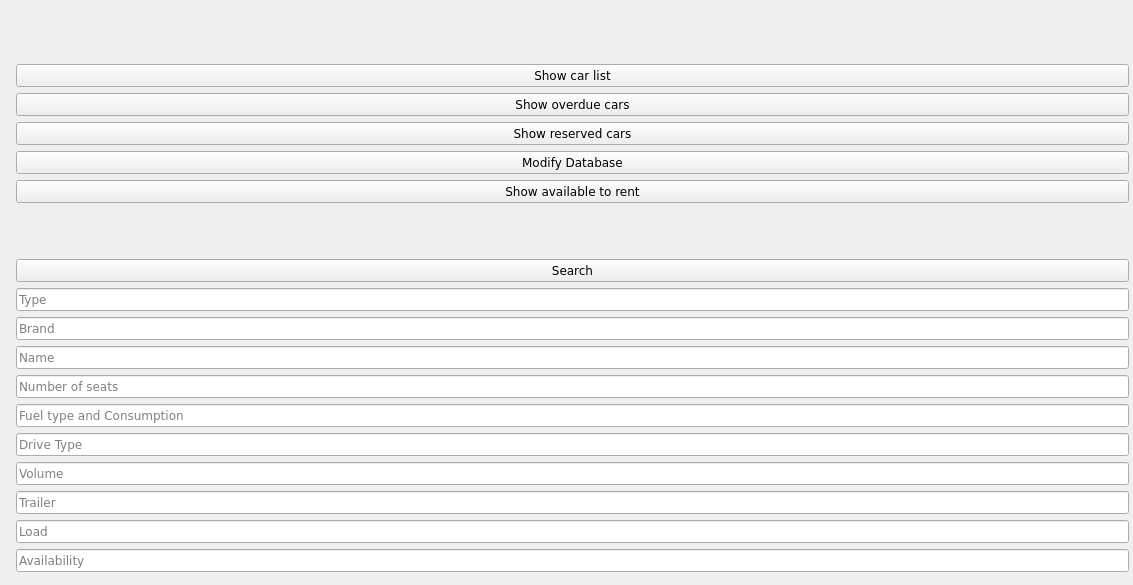
**Daniel Szepietowski**

**Sposób obsługi programu:**

Interfejs programu został zrealizowany przy pomocy QtDesigner, do włączenia programu potrzebne są pliki: car\_rental.py, car\_rental\_io.py, database.py, ui\_car\_rental.py, database\_actions.py , gui.py, dialog.py.

Uruchomienie programu odbywa się poprzez włączenie pliku gui.py

Główny interfejs pozwala na:

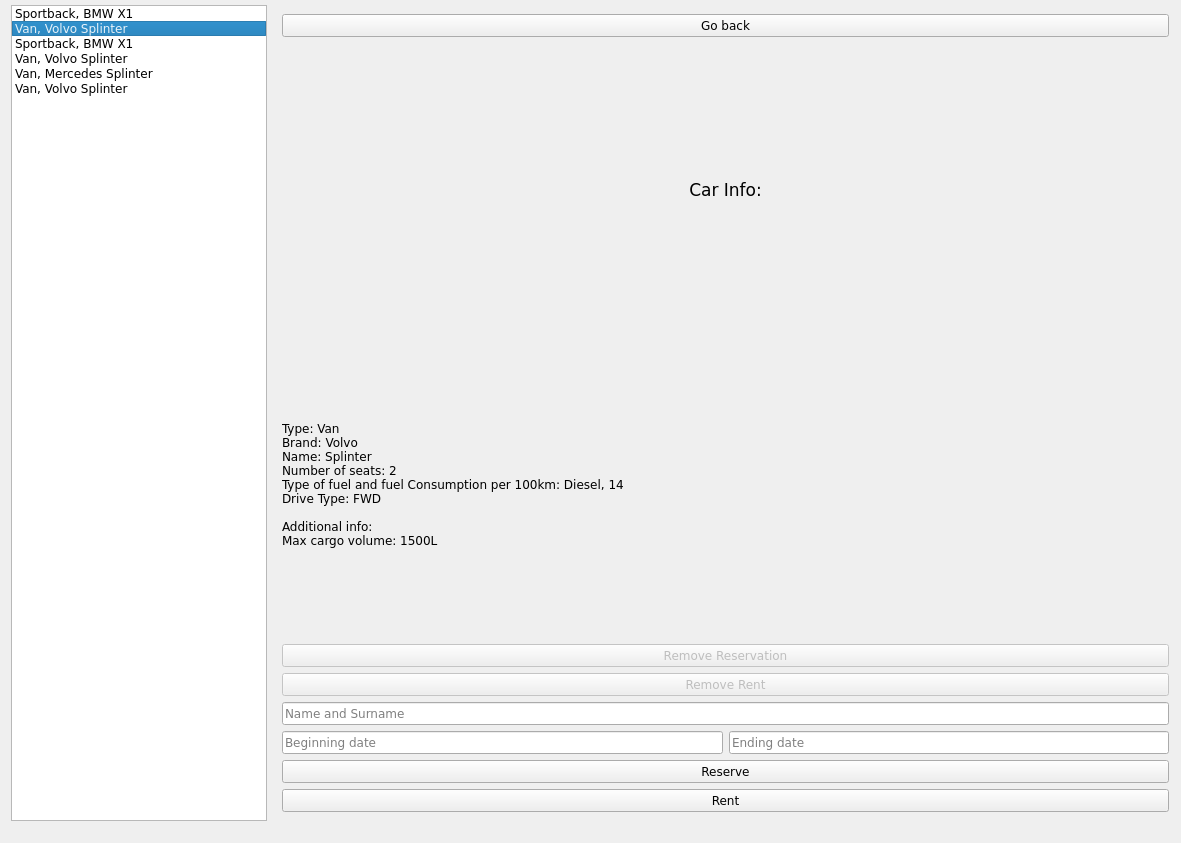


sprawdzenie listy wszystkich pojazdów,

sprawdzenie listy samochodów nieoddanych,

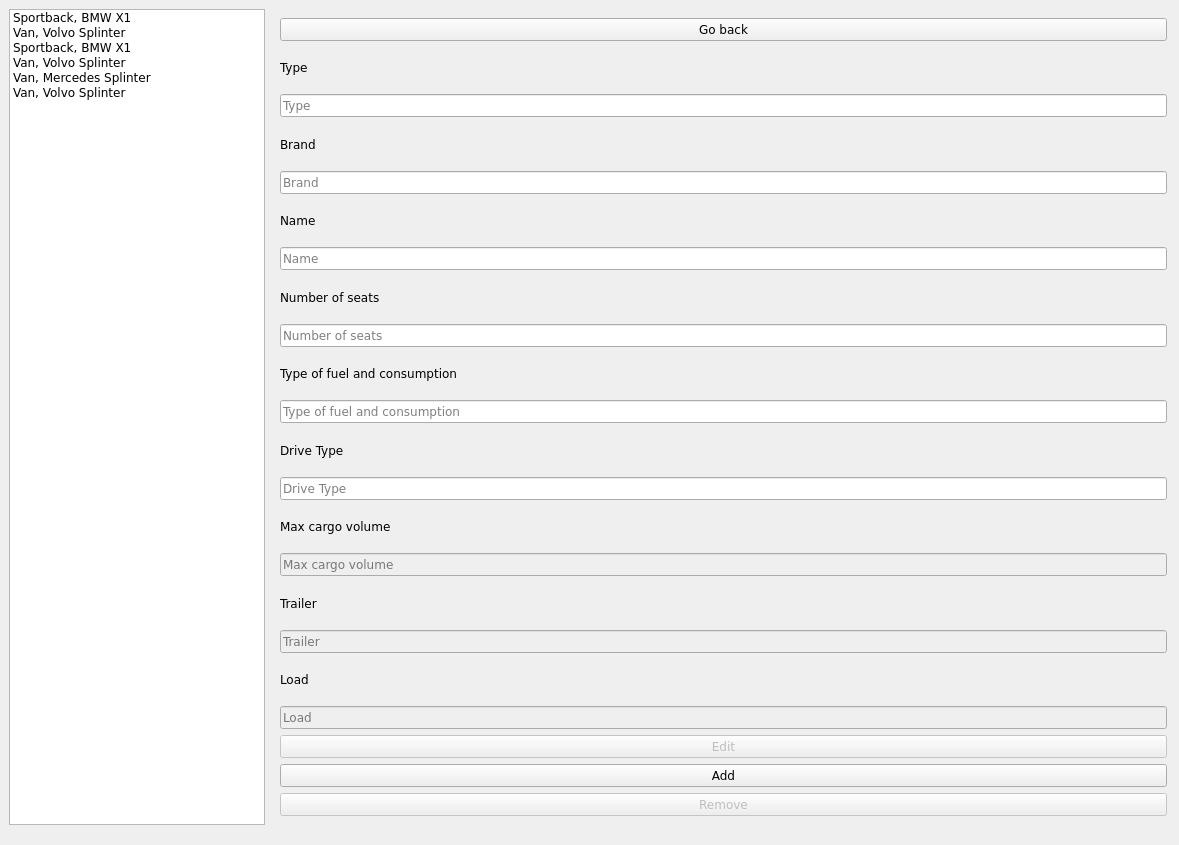
sprawdzenie listy samochodów zarezerwowanych,

sprawdzenie listy pojazdów, które można wypożyczyć.



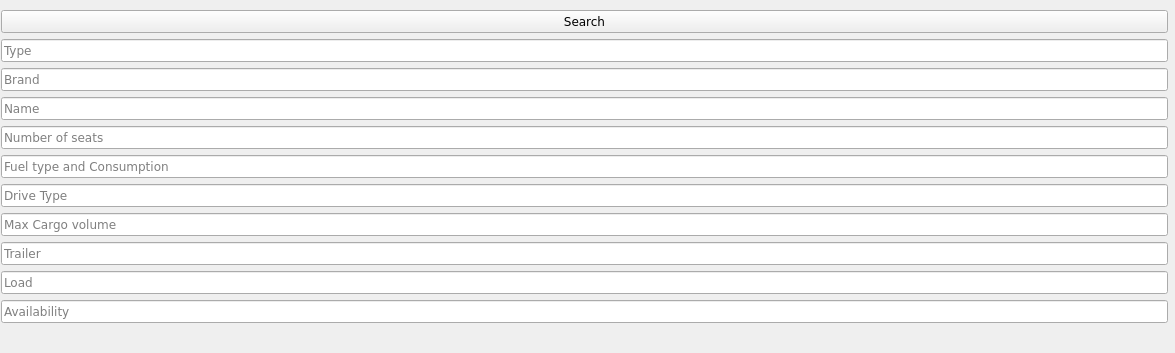
Wszystkie powyższe funkcje realizowane są poprzez powyższe menu, w którym przedstawiane są informacje dotyczące pojazdów oraz ich rezerwacji i wynajmu.

modyfikację bazy danych,



Użytkownik po kliknięciu na obiekt może usunąć go z bazy, bądź edytować. Otrzymuje również możliwość dodania pojazdu o wybranych przez siebie parametrach.

wyszukiwanie.



Wyszukiwanie przyjmuje parametry:

Type – Obsługiwany każdy typ, dostępne typy specjalne: Truck, Van.

Brand – Marka pojazdu

Name – nazwa pojazdu, modelu

Number of seats – liczba miejsc w samochodzie

Fuel type and Consumption – typ paliwa i spalanie na 100km

Drive Type – Typ napędu pojazdu

Volume – Wartość dostępna dla pojazdów typu Van i Truck

Trailer – przyjmuje wartości True i False, dostępne dla typu Truck

Load – Maksymalne obciążenie pojazdu, dla klasy Truck

Availability – przyjmuje datę(w formacie day.month.year), w której chcemy znaleźć wolny samochód.

**Budowa programu:**

Program złożony jest z plików car\_rental.py, car\_rental\_io.py, database.py, ui\_car\_rental.py, database\_actions.py, test\_car\_rental.py , gui.py, dialog.py.

car\_rental.py składa się z klas tworzących program(Rezerwacja, Wynajęcie, Samochód, Ciężarówka, Van) , oraz funkcji im przypisanych, stworzone są tam również obsługiwane błędy(InvalidDate, InvalidDateRange)

car\_rental\_io.py obsługuje plik „database.json”, który jest tekstową bazą danych programu. Potrafi z niego wyczytywać i wpisywać do niego wartości(funkcje write\_to\_json, read\_from\_json oraz wspomagającą cars\_as\_dict). Korzysta z modułu car\_rental.py aby zapisywać do klas bądź z nich odczytywać.

test\_car\_rental.py - zawiera testy dla wszystkich innych plików

database.py – obsługuje błędy przy korzystaniu z plików i wywołuje funkcje z car\_rental\_io.py, posiada funkcje read\_from\_database, write\_to\_database

database\_actions.py zawiera funkcje pracujące na bazie danych, bądź podanych listach. Są to m.in. wyświetlenie listy samochodów spóźnionych w oddaniu, wyszukiwanie pojazdów, znajdowanie pojazdów zarezerwowanych i niezarezerwowanych, wspomaga dodawanie i usuwanie obiektów z bazy danych. Operuje głównie na funkcjach z database.py

gui.py jest plikiem odpowiedzialnym za interfejs graficzny, zawiera funkcje obsługujące działania na guzikach i polach. Wspierany jest przez ui\_car\_rental.py oraz dialog.py. Zawiera w sobie klasy Dialog i CarRentalWindow. Wykorzystuje praktycznie wszystkie powyższe moduły oraz importuje z bibliotek PySide2.

Funkcje i klasy poszczególnych plików opisane są dokładniej wewnątrz nich.