

BAZA DANYCH WYPOŻYCZALNI SAMOCHODÓW OSOBOWYCH

Prowadzący: Dr hab. inż. Bogusław Cyganek

Autor: Michał Chrzanowski

Akademia Górniczo-Hutnicza

Kraków (C) 2023

1 Spis treści

1	Spis treści	2
2	Wstęp.....	4
3	Wymagania systemowe.....	5
3.1	Założenia projektowe	5
3.1.1	Przedstawienie pomysłu	5
3.1.2	Wymagania projektowe (minimalne na potrzeby przedmiotu JPO)	5
3.1.3	Podsumowanie założeń projektowych	5
3.2	Wykorzystane środki	5
4	Funkcjonalność.....	6
4.1	Funkcjonalności zrealizowane.....	6
4.2	Funkcjonalności do zrealizowania.....	6
5	Projekt techniczny	7
5.1	Schemat blokowy działania aplikacji	7
5.2	Diagram klas.....	8
5.2.1	Objaśnienie różnic pomiędzy diagramem a plikami w repozytorium.....	8
6	Opis realizacji.....	9
6.1	Środowisko programistyczne	9
6.2	Kompilator.....	9
7	Instrukcja Obsługi	10
7.1	Obsługa programu	10
7.1.1	Okno Główne	10
7.1.2	Okno Rentals, Clients.....	11
7.1.3	Okno Cars.....	11
7.1.4	Dodawanie nowych pojazdów do bazy	12
7.1.5	Wyszukiwanie	12
7.1.6	Czyszczenie wyszukiwania	13
8	GitHub.....	14
9	Bibliografia.....	15

Lista oznaczeń

Skrót	Znaczenie
GUI	Graphical User Interface
MinGW	Minimalist GNU for Windows
GCC	GNU Compiler Collection
SDK	Software Development Kit
STL	Standard Template Library
JPO	Języki Programowania Obiektowego
UML	Unified Modelling Language

2 Wstęp

Dokument dotyczy opracowania bazy danych wraz GUI. Celem jest stworzenie przejrzystej i czytelnej aplikacji pozwalającej pracownikowi na zarządzanie wypożyczalnią samochodów. W dokumencie zostaną opisane wymagania systemowe, przedstawione zostaną funkcjonalności oraz czytelnik zostanie zaznajomiony z obsługą aplikacji.

Projekt jest zorientowany na rozwój w przyszłości stąd nie wszystkie jego składowe, które zostaną opisane w założeniach nie są jeszcze zrealizowane.

Aktualny stan projektu jest zrealizowany na potrzeby przedmiotu Języki Programowania Obiektowego, spójny z minimalnymi wymaganiami o których zostanie wspomniane później.

3 Wymagania systemowe

W tym rozdziale przedstawione zostały wymagania systemowe, które zostały zadane w celu stworzenia bazy danych. Poniżej znajdują się podrozdziały, które precyzują założenia projektowe oraz wykorzystywane środki do realizacji przedstawionych założeń projektowych.

3.1 Założenia projektowe

W tym podrozdziale zostanie przedstawiony opis założeń dla projektu docelowego, wymagań dla projektu, podsumowania założeń projektowych.

3.1.1 Przetawienie pomysłu

Projekt zakłada stworzenie aplikacji dostarczającej system przechowywania danych o samochodach osobowych. Grupą docelową są małe i średnie wypożyczalnie samochodowe. Celem jest skonstruowanie programu z przyjaznym interfejsem graficznym, intuicyjnym w obsłudze oraz kompleksowym zbieraniem wszystkich niezbędnych danych. Aplikacja ma być alternatywą dla firm, które korzystają z wielu programów do przechowywania danych istotnych dla szybkiego i bezpiecznego wynajmu samochodów. Zastosowanie aplikacji ma eliminować rozdrobnienie i ułatwić kontrolę nad stanem faktycznym floty pojazdów.

3.1.2 Wymagania projektowe (minimalne na potrzeby przedmiotu JPO)

Prowadzący przedmiot załączył następujące wymagania dla stworzenia bazy danych:

- Stworzenie aplikacji bazy danych umożliwiającej przechowywanie informacji o wybranych obiektach
- Dodawanie/usuwanie rekordów
- Szukanie w bazie i wyświetlanie wyników,
- Możliwość zapisywania/odczytu z dysku (np. format *json*).
- Baza powinna posiadać tzw. engine + interfejs użytkownika GUI

3.1.3 Podsumowanie założeń projektowych

Podsumowując dwa poprzednie punkty, wyszczególniono następujące założenia projektowe:

- Opracowanie architektury aplikacji i sposobu komunikacji z użytkownikiem
- Stworzenie aplikacji
- Stworzenie funkcjonalności zdefiniowanych przez wymagania projektowe
- Przeprowadzenie testów działania aplikacji pod kątem wymagań projektowych
- Wybór odpowiednich bibliotek w celu tworzenia GUI
- Wybór odpowiedniego SDK
- Opracowanie dokumentacji technicznej

3.2 Wykorzystane środki

Na podstawie wyżej wymienionych założeń, do realizacji projektu wybrany został język C++ oraz środowisko programistyczne QT. Wybór wskazanego SDK, wynika z posiadania własnej biblioteki do tworzenia GUI. Wykorzystanie środowiska QT, zdeterminowało wykorzystanie aplikacji QT Creator 9.0.1 Community do edycji kodu, debugingu oraz testów. Organizacja plików programu oraz kontrola kompilacji została zapewniona narzędziem qmake.

4 Funkcjonalność

W tym rozdziale czytelnik zostanie zaznajomiony z funkcjonalnościami jakimi dysponuje oraz będzie dysponować aplikacją w przyszłości.

4.1 Funkcjonalności zrealizowane

Stworzona aplikacja realizuje następujące funkcjonalności:

- Dodawanie nowych pojazdów;
- Wyszukiwanie pojazdów;
- Wyświetlanie wyników wyszukiwania pojazdów;
- Zapis bazy pojazdów do pliku *json*;
- Odczyt bazy pojazdów z pliku *json*;

4.2 Funkcjonalności do zrealizowania

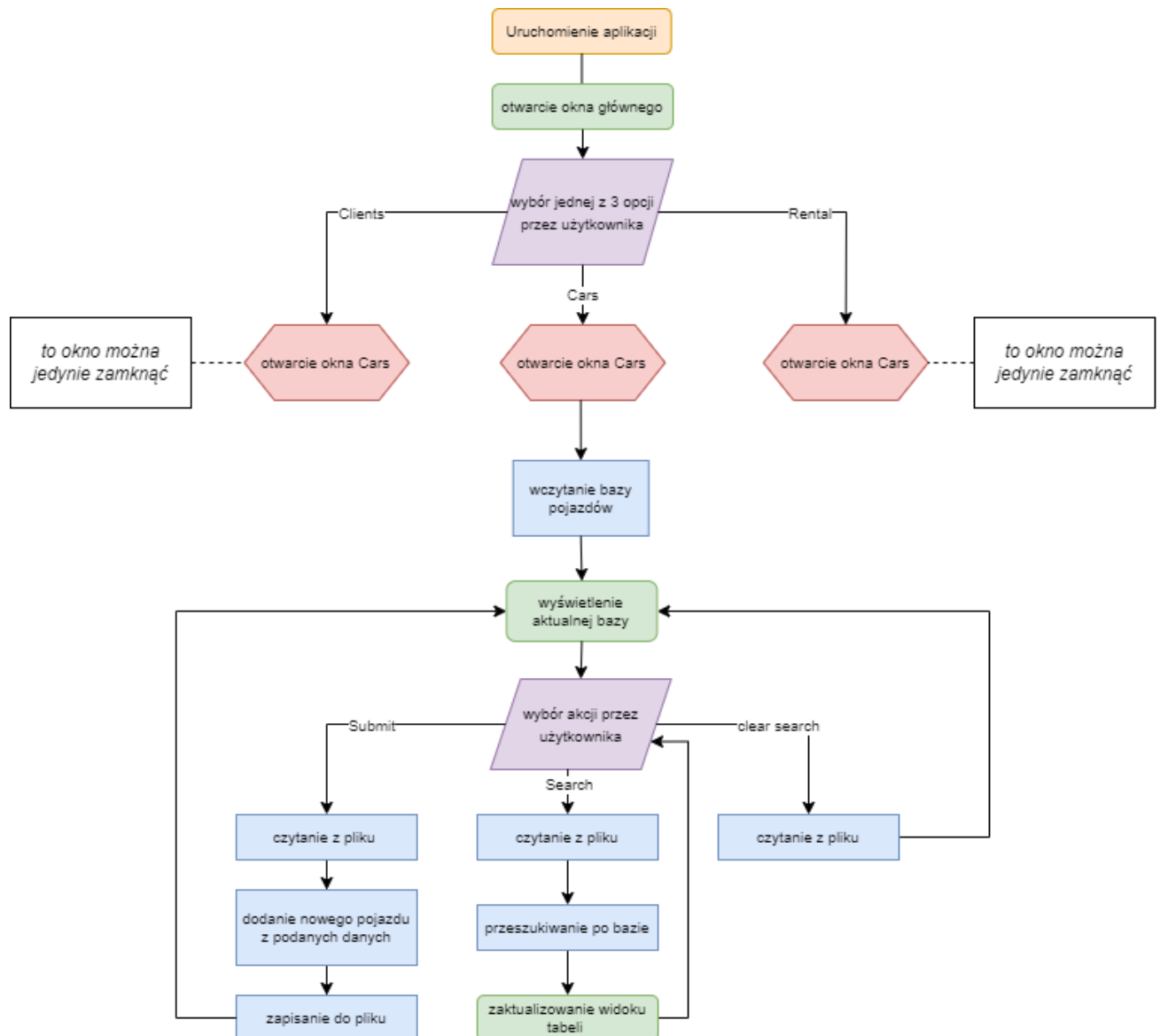
Obecny kształt aplikacji pozwala ją rozwinąć o kolejne funkcjonalności:

- Usuwanie rekordów z niedostępnymi już pojazdami;
- Dodawanie nowych klientów;
- Wyszukiwanie klientów;
- Wyświetlanie wyników wyszukiwania klientów;
- Zapis bazy klientów do pliku *json*;
- Odczyt bazy klientów z pliku *json*;
- Dodawanie rekordów nowy wynajęciem;
- Wyszukiwanie rekordów z wynajęciem;
- Wyświetlanie listy aktualnie trwających wynajmów;
- Zapis bazy z rekordami wynajmów do pliku *json*;
- Odczyt bazy z rekordami wynajmów z pliku *json*;
- Automatyczne obliczanie pełnej ceny wynajmu;
- Automatyczne informowanie klienta mailem o końcu wynajmu;
- Aktualizowanie statusu pojazdu po rozpoczęciu wynajmu na nim
- pokazywanie liczby dostępnych pojazdów;

5 Projekt techniczny

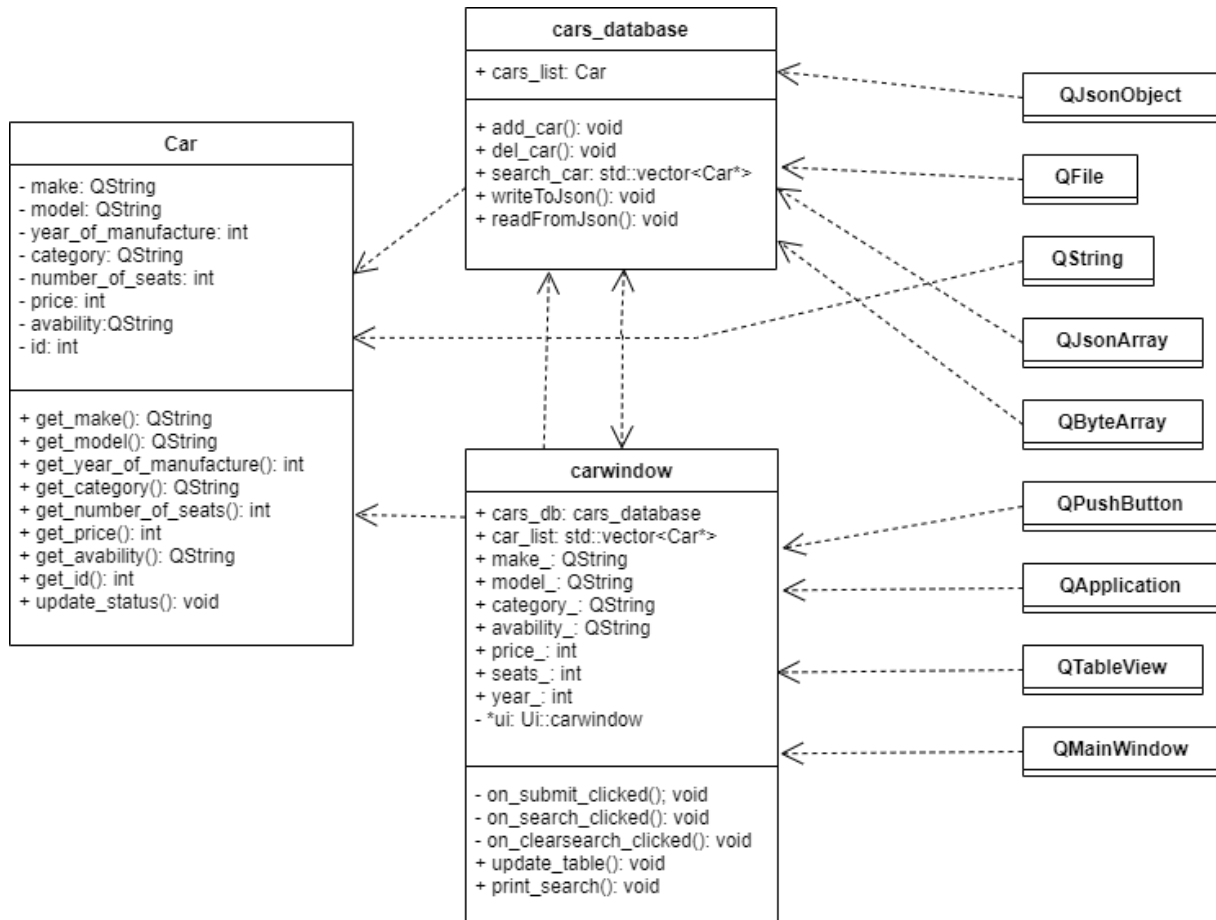
W niniejszym rozdziale czytelnikowi zostaną przedstawione szczegółowe aspekty techniczne realizacji projektu. W poniższych rozdziałach zostaną zaprezentowane schemat blokowy, diagram klas powstałych w projekcie oraz zależności jakie są między nimi.

5.1 Schemat blokowy działania aplikacji



5.2 Diagram klas

Poniżej znajduje się diagram UML opisujący istniejące klasy oraz zależności pomiędzy sobą.



5.2.1 Objaśnienie różnic pomiędzy diagramem a plikami w repozytorium

Różnice pomiędzy plikami które znajdują się w projekcie, a diagramem widocznym powyżej wynikają, z chęci przedstawienia aktualnie ukończonych i funkcjonalnych klas. Wraz z rozwojem projektu, diagram będzie zastępowany nowymi wersjami z aktualnymi klasami, metodami które zawierają oraz bieżącą siecią połączeń.

6 Opis realizacji

Rozdział ten zaznajomi czytelnika ze szczegółowym opisem środowiska programistycznego, architektura implementacji oraz wykorzystywanych narzędzi

6.1 Środowisko programistyczne

Qt to zestaw narzędzi do tworzenia aplikacji dla różnych platform (Windows, macOS, Linux, iOS i inne). Składa się z bibliotek C++, narzędzi do projektowania interfejsu użytkownika oraz narzędzi do tworzenia aplikacji mobilnych.

Qt Creator to środowisko programistyczne oparte na Qt, które zawiera edytor kodu, narzędzia do debugowania, wizualne narzędzia do projektowania interfejsu użytkownika oraz narzędzia do tworzenia aplikacji mobilnych. Może być używane zarówno z poziomu linii poleceń, jak i z poziomu graficznego interfejsu użytkownika.

Qt składa się z kilku podstawowych elementów:

- Biblioteki QtCore - zawierające podstawową funkcjonalność, taką jak obsługa plików, kolejek zadań, kontenery i inne narzędzia używane przez pozostałe biblioteki;
- Biblioteki QtGui - zawierające klasy odpowiedzialne za interfejs użytkownika, takie jak okna, przyciski, menu itp;
- Biblioteki QtWidgets - rozszerzenie biblioteki QtGui, które dostarcza dodatkowe kontrolki, takie jak tabele, drzewa, paski narzędzi i inne;
- Narzędzie Qt Creator - środowisko programistyczne umożliwiające tworzenie, edycję i debugowanie aplikacji;
- Narzędzie Qt Designer - pozwalające na projektowanie interfejsu użytkownika.

6.2 Kompilator

MinGW to darmowa implementacja narzędzi GCC dla systemów Windows. MinGW pozwala na kompilowanie programów napisanych w języku C, C++ i Fortran na systemach Windows. Składa się z kompilatorów, bibliotek i narzędzi potrzebnych do budowania aplikacji konsolowych oraz okienkowych. MinGW jest kompatybilne z systemami Windows XP, Vista, 7, 8, 10. MinGW umożliwia tworzenie natywnych aplikacji Windows bez potrzeby używania bibliotek cygwin.dll, dzięki czemu aplikacje są szybsze i mniejsze. MinGW jest zgodny z GNU GPL oraz z innymi licencjami GNU.

7 Instrukcja Obsługi

Projekt został utworzony w środowisku programistycznym QT. Aby móc uruchomić aplikację należy pobrać i zainstalować na lokalnym systemie aplikację QT Creator 9.0.1 (Community).

Uruchomienie aplikacji wymaga od użytkownika wykonania podanych poniżej czynności:

1. Sklonowanie repozytorium na dysk lokalny (link do repozytorium znajduje się [tutaj](#));
2. Uruchomienie programu QT Creator
3. Z menu programu należy wybrać opcję Open Project
4. Wybrać w oknie dialogowym ścieżkę do pliku otwierającego projekt

Przykładowa ścieżka do pliku może wyglądać tak :

"C:\Users\..\studia\semestr 5-6\Języki programowania
obiektowego\projekt\Data_base_car_rent\Data_base_car_rent.pro"

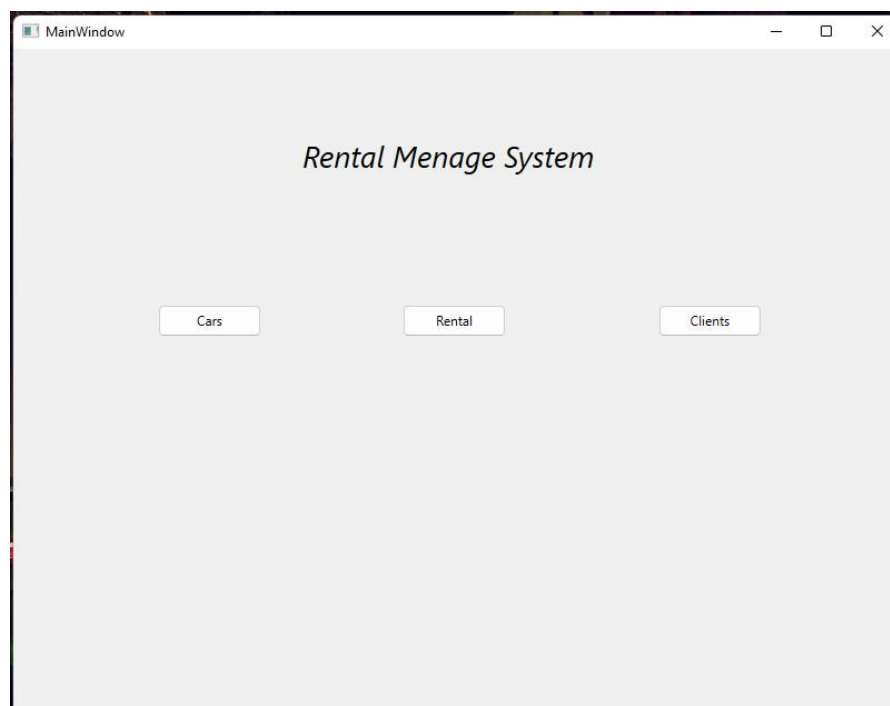
5. Otwarta zostanie cała struktura projektu
6. Należy wybrać z menu w pasku górnym: Budowanie → Uruchom
7. Uruchomiona zostanie aplikacja Rental Menage System

7.1 Obsługa programu

Poniżej czytelnik zostanie zapoznany z obsługą poszczególnych okien ekranu.

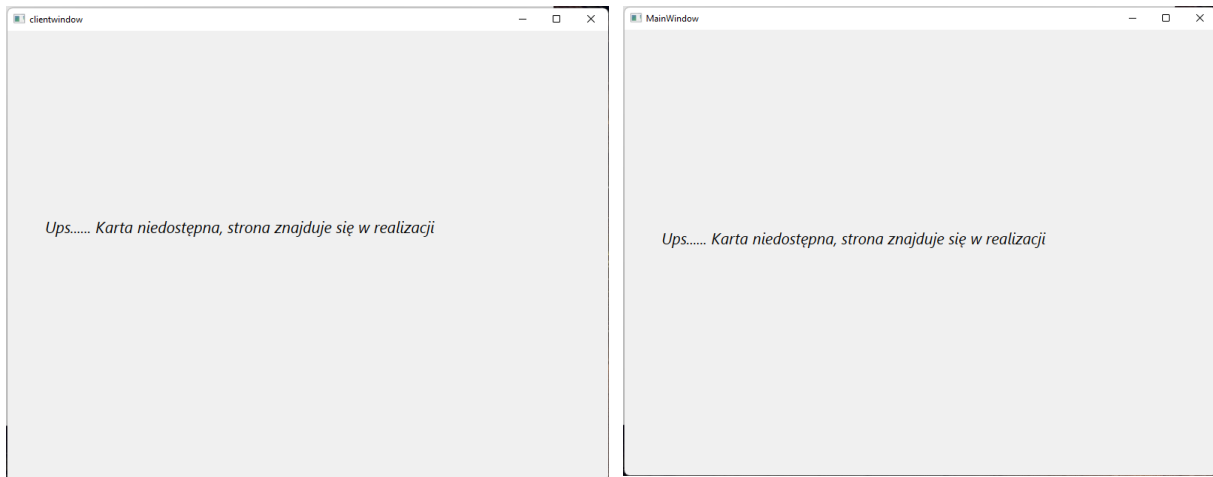
7.1.1 Okno Główne

Po uruchomieniu aplikacji ukazuje się okno główne. Dostępne są do wyboru trzy opcje. Otwarcie bazy pojazdów pod przyciskiem Cars, tworzenie nowego wypożyczenia pod przyciskiem Rental, dodawanie i przegląd klientów pod przyciskiem Clients.



7.1.2 Okno Rentals, Clients

Na aktualnym etapie projektu, opcje te są niedostępne, ale zostały dodane w celu informowania użytkownika o funkcjach które zostaną dodane w przyszłości



7.1.3 Okno Cars

Poniżej zaprezentowane jest przykładowe okno Cars zawierające widok aktualnej bazy danych pojazdów, oraz interfejs z polami pobieranymi od użytkownika.

The image shows a screenshot of the 'carwindow' application. It features a search form at the top with fields for 'make', 'model', 'category', and 'year of production', followed by a 'submit' button. Below these are fields for 'price/per day', 'number of seats', and 'avability', followed by 'search' and 'clear search' buttons. At the bottom, there is a table displaying car data.

	id	category	avability	make	model	number_of_seats	price	sa
1	1674501662	S	unavaiaible	Audi	RS3	5	740	20
2	1674501623	S	unavaiaible	BMW	M3 Competition	5	800	20
3	1674501545	S	avaiaible	Mercedes-Benz	CLA 4S AMG	5	600	20
4	1674501459	S	avaiaible	Ford	Mustag	5	550	20

7.1.4 Dodawanie nowych pojazdów do bazy

Aby dodać nowy pojazd do bazy należy uzupełnić wszystkie pola, po czym zatwierdzić przyciskiem SUBMIT. Jeśli dane zostaną poprawnie uzupełnione to nowo dodany pojazd wyświetli się na górze listy. Poniżej pokazano przykład dodawania pojazdu marki Renault

The screenshot shows a web application window titled 'carwindow'. It contains a form with the following fields:

- make:** Renault
- model:** Megane RS
- category:** S
- year of production:** 2012
- price/per day:** 400
- number of seats:** 5
- availability:** available

Buttons: submit, search, clear search.

Below the form is a table with the following data:

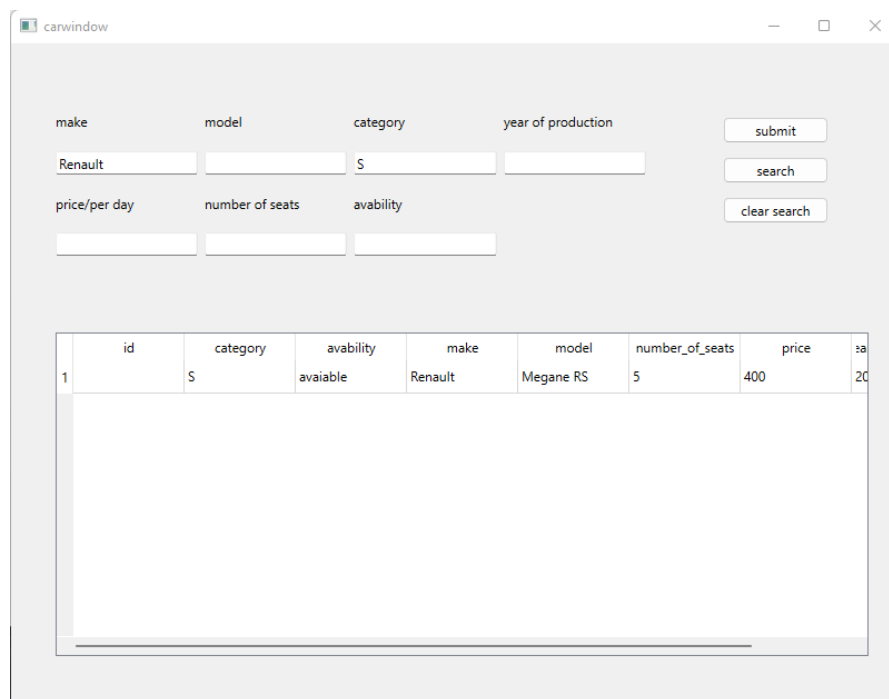
	id	category	availability	make	model	number_of_seats	price	sa
1	1674501662	S	unavailable	Audi	RS3	5	740	20
2	1674501623	S	unavailable	BMW	M3 Competition	5	800	20
3	1674501545	S	available	Mercedes-Benz	CLA 45 AMG	5	600	20
4	1674501459	S	available	Ford	Mustag	5	550	20

A tutaj zaktualizowany widok tabeli po dodaniu ostatniego elementu

	id	category	availability	make	model	number_of_seats	price	sa
1	1674501928	S	available	Renault	Megane RS	5	400	20
2	1674501662	S	unavailable	Audi	RS3	5	740	20
3	1674501623	S	unavailable	BMW	M3 Competition	5	800	20
4	1674501545	S	available	Mercedes-Benz	CLA 45 AMG	5	600	20
5	1674501459	S	available	Ford	Mustag	5	550	20

7.1.5 Wyszukiwanie

Aby wyszukać czy dany pojazd znajduje się w bazie danych należy podać parametry po których użytkownik chciałby wyszukiwać. Nie trzeba podawać wszystkich wartości. Po wpisaniu zadanych parametrów należy wcisnąć przycisk SEARCH. W tabeli wyświetlone zostaną pasujące elementy.



The screenshot shows a web application window titled "carwindow". It contains a search form with the following fields and buttons:

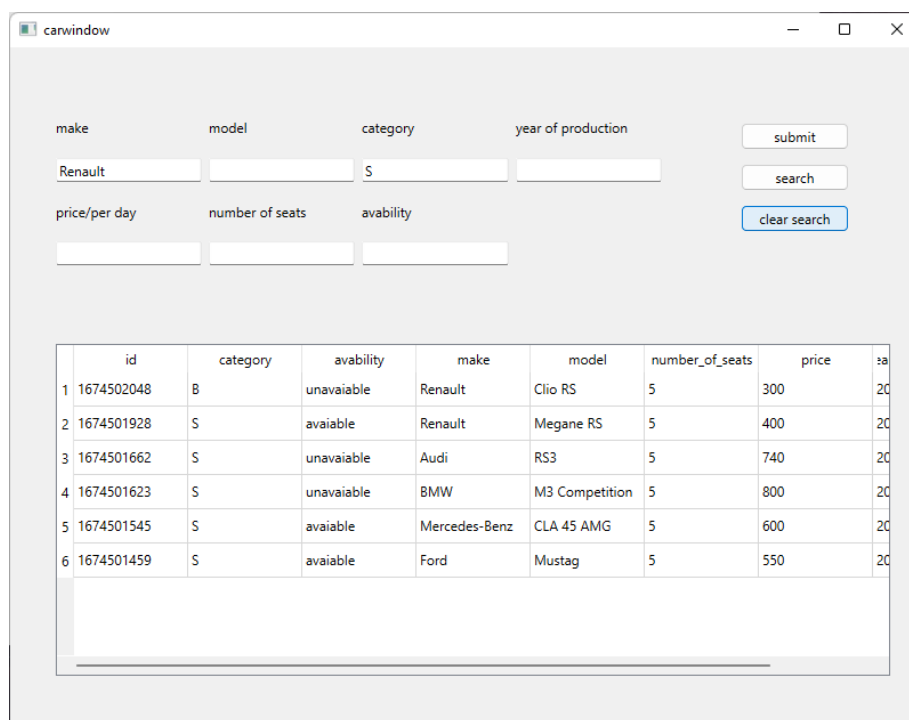
- make: Renault
- model:
- category: S
- year of production:
- submit button
- search button
- clear search button
- price/per day:
- number of seats:
- availability:

Below the form is a table with the following data:

	id	category	availability	make	model	number_of_seats	price	sa
1		S	avaible	Renault	Megane RS	5	400	20

7.1.6 Czyszczenie wyszukiwania

Wciśnięcie przycisku CLEAR SEARCH, powoduje ponowne wyświetlenie wszystkich dostępnych pojazdów



The screenshot shows the same web application window as before, but with the "clear search" button highlighted in blue. The table now displays six rows of data:

	id	category	availability	make	model	number_of_seats	price	sa
1	1674502048	B	unavaible	Renault	Clio RS	5	300	20
2	1674501928	S	avaible	Renault	Megane RS	5	400	20
3	1674501662	S	unavaible	Audi	RS3	5	740	20
4	1674501623	S	unavaible	BMW	M3 Competition	5	800	20
5	1674501545	S	avaible	Mercedes-Benz	CLA 45 AMG	5	600	20
6	1674501459	S	avaible	Ford	Mustag	5	550	20

8 GitHub

Repozytorium projektu zostało zamieszczone w systemie GitHub. Tam też będą aktualizowane wszelkie zmiany dotyczące rozwoju pozostałej części klas oraz ich funkcjonalności.

W repozytorium znajdują się wszystkie pliki dostępne w projekcie. Niektóre pliki zamieszczone są w formie sygnalizacyjnej, które pokazują pomysły na rozwój projektu. Każdy plik zawiera komentarz wraz informacją o wersji lub w fazie w której aktualnie się znajduje.

W repozytorium dostępna jest również ta dokumentacja techniczna, która będzie uaktualniana na wraz z realizacją kolejnych etapów projektowych.

Link do repozytorium: https://github.com/mchrzanow/jpo_project.git

9 Bibliografia

- [1] Bogusław Cyganek: Introduction to Programming with C++ for Engineers, John Wiley & Sons Ltd 2021
- [2] <https://wiki.qt.io>
- [3] <https://blogprogramisty.net/komendy-git-a-ktore-nalezy-znac/>
- [4] <https://github.com/Ellie-Y/CarRentalSystem.git>
- [5] <https://thecodeprogram.com/how-to-use-json-data-with-qt-c->
- [6] <https://www.youtube.com/@VoidRealms>
- [7] <https://www.samouczekprogramisty.pl/podstawy-uml/>