

**Specyfikacja projektu**  
**Oprogramowanie Systemów Pomiarowych**

**Myjnia samochodowa**

Grupa dziekańska **TI-4** Rok akademicki **2020/21** Semestr **VI**

Skład sekcji:

Kamil Wieczorek

Olaf Kędziora

Maciej Liszka

Wojciech Gołek

## Spis treści

1. Wprowadzenie .....	2
2. Założenia projektu .....	2
2.1. Opis działania .....	2
2.2. Przeznaczenie projektu i możliwości wykorzystania .....	2
3. Oprogramowanie .....	3
3.1. Opis środowiska oraz wykorzystanych pakietów .....	3
3.2. Interfejs użytkownika .....	3
3.3. Schemat blokowy .....	5
3.4. Schemat przepływu danych .....	6
3.5. Opis stanów .....	7
3.6. Opis stworzonych subVI .....	7
3.7. Opis stworzonych kontrolek .....	8
4. Przypadki użycia (use cases) .....	9
4.1. Naciśnięcie "Start" przez użytkownika: .....	9
4.2. Przekazanie informacji nt. mycia do bazy danych (wgląd dla administratora) oraz tworzenie statystyk .....	9
5. Opis błędów .....	10
5.1. Procedury testowe .....	10
5.2. Obsługa błędów .....	11
6. Bibliografia .....	13

## **1. Wprowadzenie**

Dokument ten opisuje sposób działania oraz implementacji projektu oprogramowania myjni samochodowej bazującego na przykładowym zadaniu egzaminacyjnym (CLD). Zadanie zostało rozbudowane o dodatkowe funkcje takie jak obsługa systemu wizyjnego oraz komunikacja z bazą danych.

## **2. Założenia projektu**

### **2.1. Opis działania**

Opisywany sterownik myjni symuluje działanie rzeczywistego systemu sterowania automatycznej myjni samochodowej. Poprzez interakcję użytkownika z elementami panelu czołowego, może on wybierać odpowiednie opcje mycia oraz symulować ruch pojazdu pomiędzy poszczególnymi stacjami myjni. Użytkownik jest na bieżąco informowany o postępach procesu mycia samochodu i czasie jaki upłynął dla każdej z wybranych opcji. Dzięki wykorzystaniu bazy danych (SQLite) oraz systemu wizyjnego możliwy jest dostęp do historycznych danych myjni.

### **2.2. Przeznaczenie projektu i możliwości wykorzystania**

Aplikacja napisana w ramach projektu może znaleźć zastosowanie w automatycznych myjniach samochodowych.

Zbierane przez aplikację informacje przechowywane są w bazie danych i mogą zostać wykorzystane na wiele sposobów. Osoba zarządzająca myjnią ma wgląd w szereg statystyk mogących usprawnić proces zarządzania. Przykładowo, mając wgląd w wybierane przez klientów opcje mycia, menedżer myjni otrzymuje niezbędne informacje na temat popularności poszczególnych opcji.

Wykorzystanie systemu wizyjnego i odczytywanie oraz zapisywanie do bazy danych numerów tablic rejestracyjnych umożliwia przykładowo opracowanie systemu zniżek dla klientów lub wykorzystanie bezgotówkowego systemu płatności, czy też sprzedaż karnetów. Wspomniane rozwiązania zdecydowanie zwiększają atrakcyjność myjni korzystającej z aplikacji, a co za tym idzie mogą zachęcić klientów do jej wyboru.

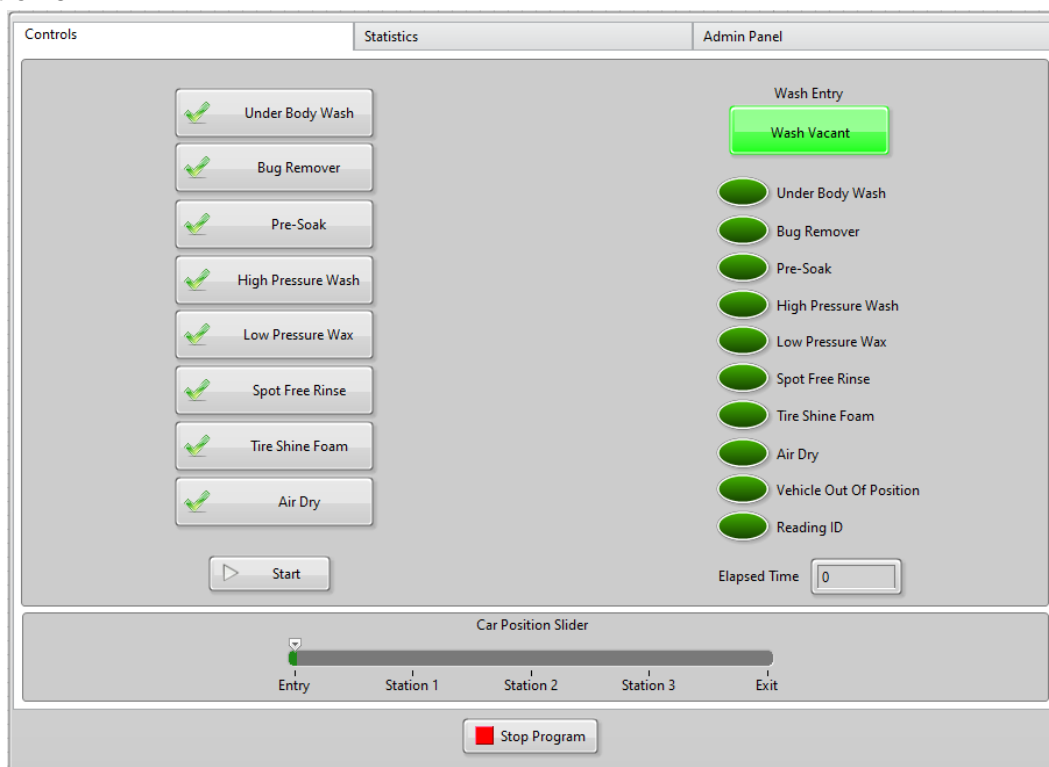
### 3. Oprogramowanie

#### 3.1. Opis środowiska oraz wykorzystanych pakietów

Projekt został wykonany w środowisku LabVIEW. Wykorzystano wersję oprogramowania 2020. Aplikacja oparta jest na maszynie stanów. W implementacji zastosowany został darmowy szablon JKI State Machine dostępny poprzez VIPM (VI Package Manager). Do implementacji wizji maszynowej został wykorzystany pakiet Vision Development Module wraz z Vision Assistant. Dane zbierane były w lokalnej bazie danych opartej o system zarządzania relacyjną bazą danych SQLite. Jej połączenie było możliwe dzięki zastosowaniu pakietu SQLite Library by drjdpowell.

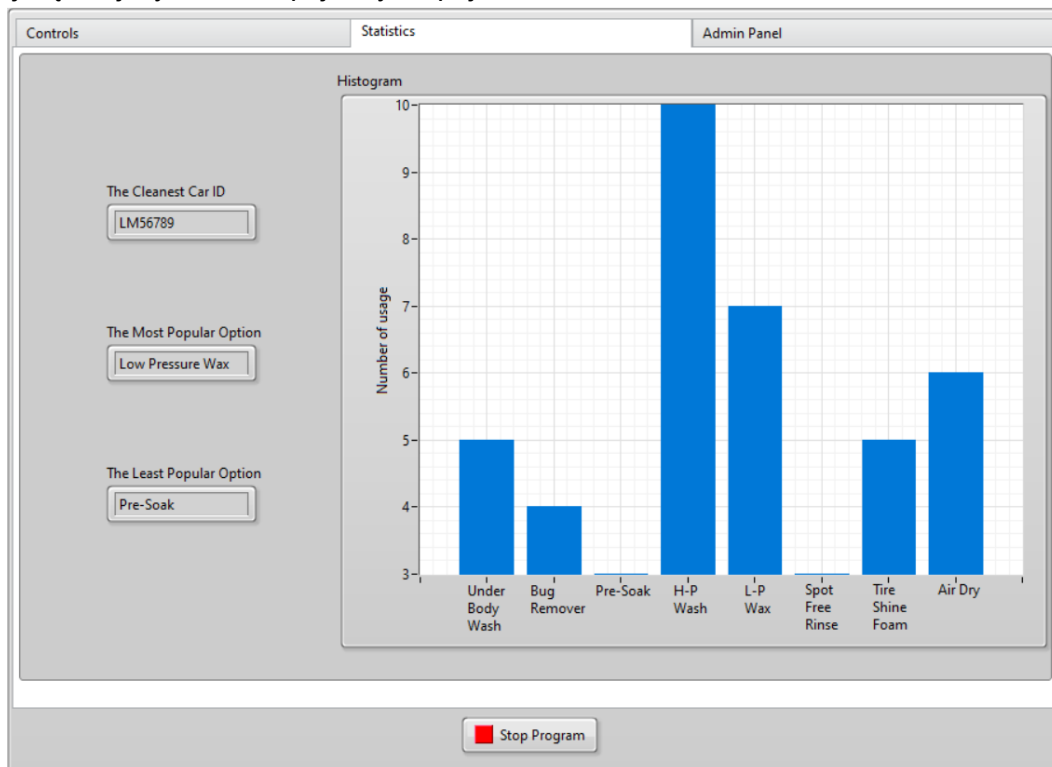
#### 3.2. Interfejs użytkownika

Na interfejs użytkownika składają się trzy przyciski (tab control) pozwalające na swobodne przemieszczanie się pomiędzy trzema stronami panelu czołowego głównej aplikacji. Na pierwszej ze stron znajduje się konsola pozwalająca na wybór interesujących użytkownika opcji mycia pojazdu samochodowego. Ich wybór zostaje potwierdzony przyciskiem "Start", który znajduje się bezpośrednio pod konsolą. Po prawej stronie tej strony panelu znajduje się szereg diod - wskaźników. Na samej górze tej części panelu znajduje się wskaźnik informujący o tym, czy myjnia jest w obecnej chwili zajęta. Pod nim można zauważyć szereg diod informujących o obecnym etapie mycia oraz o ewentualnej konieczności przemieszczenia pojazdu na następną stację. Przeniesienie pojazdu odbywa się w sposób symulowany, poprzez przesunięcie suwaka znajdującego się w dolnej części panelu czołowego. Ustawienie go w odpowiedniej pozycji powoduje upływanie czasu potrzebnego na wykonanie bieżącego cyklu mycia który użytkownik może zobaczyć powyżej suwaka, po prawej stronie.



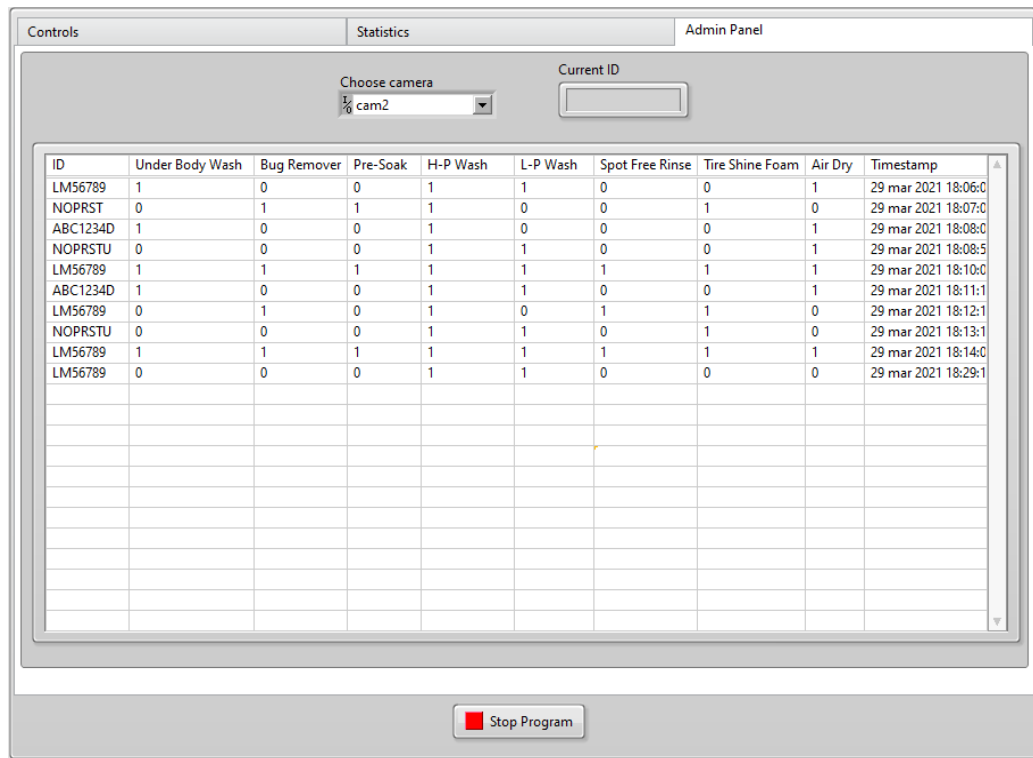
Rys. 1. Pierwsza strona interfejsu użytkownika

Na następnej stronie interfejsu użytkownika znajdują się dane. Po prawej stronie widnieją statystyki dotyczące częstości wykonywania konkretnych czynności (opcji) mycia samochodu. Zaś po lewej stronie została pokazana tablica rejestracyjna samochodu, który najwięcej razy odwiedził myjnię samochodową oraz najchętniej i najczęściej wybierane opcje mycia pojazdów.



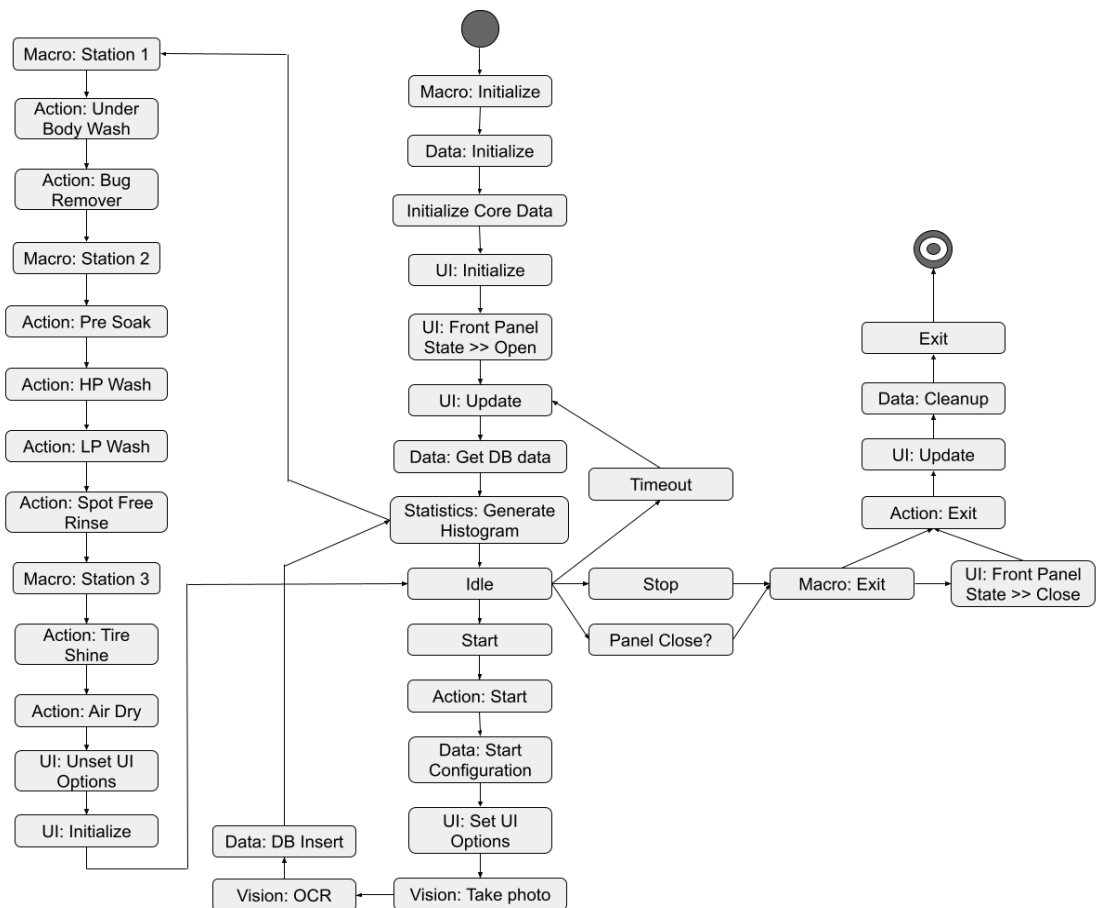
Rys. 2. Druga strona interfejsu użytkownika

Ostatnią opcję stanowi panel administracyjny, można sprawdzić na nim numer tablicy rejestracyjnej pojazdu aktualnie zajmującego myjnię, a także sprecyzować która kamera ma służyć do uzyskania tego numeru. Znajduje się również podgląd obecnego stanu bazy danych. Można odczytać w niej informacje o przeprowadzonych operacjach mycia samochodu, na które składają się: informacja o tablicy rejestracyjnej umytego pojazdu samochodowego, jakie opcje mycia zostały wybrane przez użytkownika oraz czas rozpoczęcia cyklu mycia.



Rys. 3. Trzecia strona interfejsu użytkownika

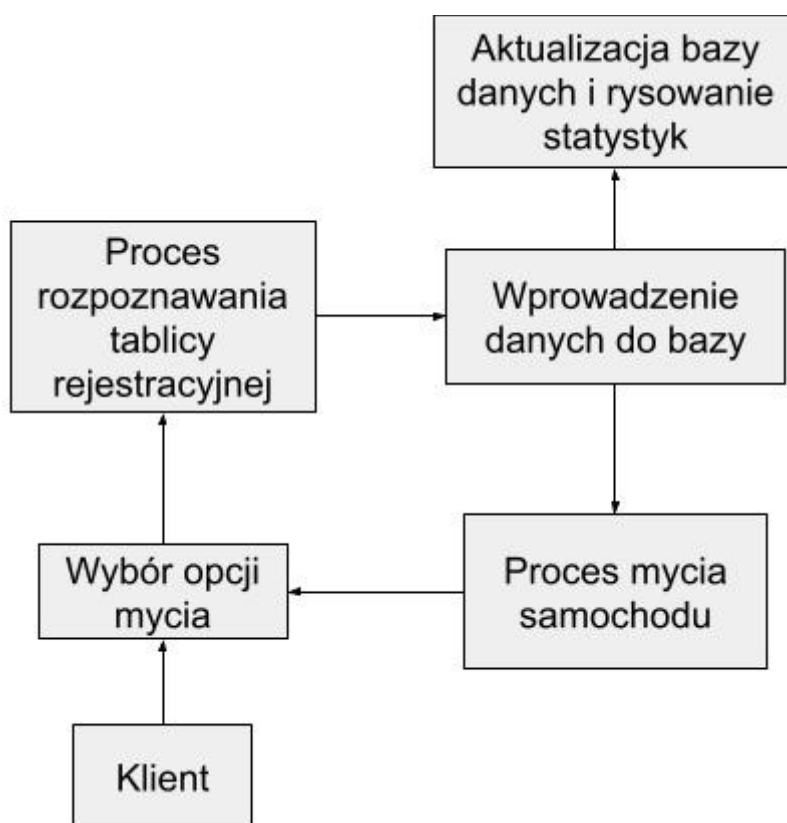
### 3.3. Schemat blokowy



Rys. 4. Schemat blokowy

Przedstawiony na rysunku 4 schemat blokowy programu pokazuje sposób działania programu. Poszczególne bloki odpowiadają konkretnym stanom zaimplementowanym w aplikacji. Program rozpoczyna działanie od inicjalizacji danych oraz podstawowych wartości widocznych na interfejsie użytkownika. Następnie przechodzi do właściwej części pracy, podczas której wykonuje operacje na danych uzyskanych od klienta myjni. Kończąc pracę interfejs jest odświeżany po raz ostatni, a wszelkie referencje oraz połączenia są zamykane.

### 3.4. Schemat przepływu danych



Rys. 5. Schemat przedstawiający przepływ danych

Na rysunku 5 widoczny jest schemat przepływu danych badanej aplikacji. Przedstawia on ogólny zarys sposobu przepływu danych w aplikacji. Źródłem danych w aplikacji zawsze jest klient, który wybiera opcje mycia. Oprócz wybranych opcji zapisywany jest również numer tablicy rejestracyjnej klienta. Dane te wprowadzane są do bazy danych, a następnie na ich podstawie przeprowadzany jest proces mycia samochodu. Informacje wpisane do bazy są również wykorzystywane do tworzenia statystyk. Dane pomiędzy poszczególnymi stanami przekazywane są za pomocą rejestrów przesuwanych zaimplementowanych w wykorzystanym szablonie.

### 3.5. Opis stanów

1. "Event Structure", "Idle" - stan zawierający podstawową strukturę zawartą w JKI State Machine. Odpowiada za obsługę stanu bezczynności w aplikacji, a także zapewnienie obsługi podstawowych przycisków z panelu czołowego aplikacji (np. start). Dokładniejszy opis obsługi przedstawiony został w rozdziale 4 na przykładzie działania programu po naciśnięciu przycisku start.
2. "Core" - ta grupa stanów odpowiada za obsługę nieobsługiwanego stanu (w razie gdy się pojawią np. przez podanie złej nazwy przy wywołaniu stanu), bezpieczne wyłączenie aplikacji w przypadku zamknięcia okna a także za obsługę błędów.
3. "Data" - ta grupa stanów odpowiada za zarządzanie danymi.
4. "UI" - ta grupa stanów odpowiada za panel użytkownika, między innymi za inicjalizację, blokowanie określonych funkcji (przycisków) i odświeżanie wyświetlanych wartości.
5. "Macro" - ta grupa stanów odpowiada za wywołanie w odpowiedniej kolejności stanów z grupy "Action" oraz za wywołanie stanów przy uruchamianiu i wyłączaniu aplikacji.
6. "Action" - ta grupa stanów odpowiada za włączenie trybów mycia wybranych przez użytkownika.
7. "Statistics" - ta grupa stanów odpowiada za tworzenie histogramu oraz analizę danych statystycznych.
8. "Vision" - ta grupa stanów odpowiada za analizę obrazu z kamery, zrobienie zdjęcia tablicy rejestracyjnej, a następnie odczytanie z niego znaków.

### 3.6. Opis stworzonych subVI

1. DB\_Create.vi - zadaniem tego subVI jest utworzenie bazy danych zawierającej dane myjni na wskazanej przez użytkownika ścieżce, uwzględnia również inne możliwości które zostały dokładniej opisane w rozdziale 5.2.
2. DB\_Close.vi - zadaniem tego subVI jest zamknięcie połączenia z bazą danych
3. DB\_GetMostFrequentID.vi - zadaniem tego subVI jest wyciągnięcie z bazy danych numeru rejestracyjnego, który występuje najczęściej w zapisanych danych.
4. DB\_InsertData.vi - zadaniem tego subVI jest przekazanie numeru rejestracyjnego oraz trybów mycia wybranych przez użytkownika do bazy danych.



5. DB\_SelectData.vi - zadaniem tego subVI jest wyciągnięcie wszystkich numerów rejestracyjnych oraz odpowiadających im trybów mycia z bazy danych.
6. SB\_SelectSumStats.vi - zadaniem tego subVI jest określenie ilości występowania poszczególnych trybów mycia w bazie danych.
7. Timer.vi - zadaniem tego subVI jest zliczanie zadanego czasu, po jego upływie jedno z wyjść zostaje ustawione na logiczną jedynkę. Możliwe jest zatrzymanie zliczania czasu oraz zresetowanie stoperta.
8. Take\_photo.vi - zadaniem tego subVI jest odnalezienie dopasowania do przykładowego obrazu (fragment tablicy rejestracyjnej z literkami PL) w obrazie z kamery. Po odnalezieniu dopasowania, czyli po ukazaniu się tablicy rejestracyjnej w obiektywie kamery, wykonane jest zdjęcie, które następnie zostanie przekazane do innego subVI w celu przetworzenia wykonanego zdjęcia.
9. OCR.vi - zadaniem tego subVI jest interpretacja oraz odczytanie znaków ze zdjęcia wykonanego przy pomocy subVI Take\_photo.vi. Optyczne rozpoznawanie znaków odbywa się dzięki wykonanemu wcześniej treningowi OCR, którego efekt został zapisany w pliku o rozszerzeniu .abc. Po rozpoznaniu znaków, zadaniem tego subVI jest przepisanie rozpoznanych znaków z formy graficznej na formę tekstową.

### **3.7. Opis stworzonych kontrolek**

- 3.7.1. Car wash Data.ctl - definicja typu zawierająca elementy sterujące myjnią.
- 3.7.2. Car Wash Entry Console.ctl - definicja typu zawierająca tryby mycia
- 3.7.3. Car Wash Led.ctl - definicja typu zawierająca wskaźniki led dla aktualnych opcji oraz jeden wskaźnik, aby sprawdzić, czy czytany jest identyfikator samochodu.

## **4. Przypadki użycia (use cases)**

### **4.1. Naciśnięcie "Start" przez użytkownika:**

- 4.1.1. Kontrolki na panelu czołowym informują użytkownika o dostępności myjni.
- 4.1.2. Użytkownik dokonuje wyboru trybów mycia.
- 4.1.3. Użytkownik naciska przycisk "Start".
- 4.1.4. Użytkownik wjeżdża na myjnię.
  - Kontrolka informuje użytkownika o odczytywaniu znaków z tablicy rejestracyjnej.
- 4.1.5. Użytkownik korzystając z suwaka przekazuje informację o obecnym położeniu samochodu (Symulacja przejazdu samochodu przez myjnię).
  - Kontrolka informuje użytkownika, że samochód jest na niewłaściwej stacji. Proces mycia wstrzymany.
  - Kontrolki informują użytkownika który tryb mycia jest wykonywany w danej chwili.
- 4.1.6. Użytkownik korzystając z suwaka przekazuje informację o obecnym położeniu samochodu (Symulacja przejazdu samochodu przez myjnię).
  - Kontrolka informuje użytkownika, że samochód jest na niewłaściwej stacji. Proces mycia wstrzymany.
  - Kontrolki informują użytkownika który tryb mycia jest wykonywany w danej chwili.
- 4.1.7. Po wyjechaniu z myjni panel czołowy się resetuje i jest gotowy do następnego użycia

### **4.2. Przekazanie informacji nt. mycia do bazy danych (wgląd dla administratora) oraz tworzenie statystyk**

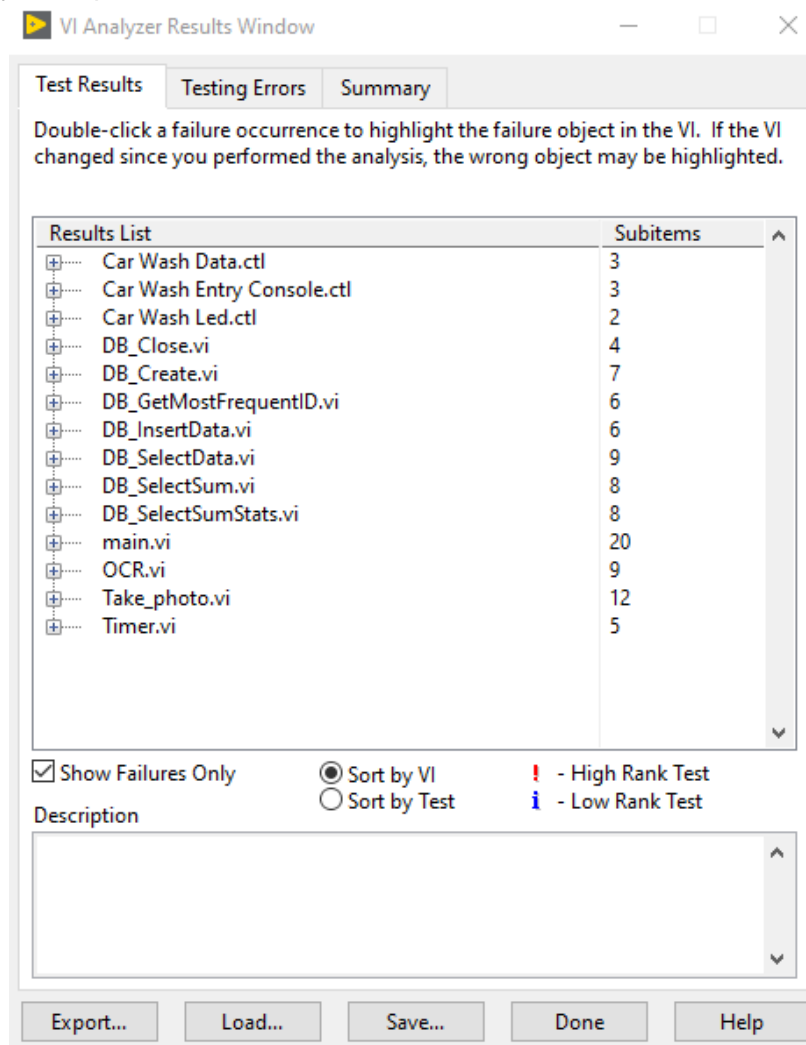
- 4.2.1. Po wjeździe samochodu na myjnię, kamera wykonuje zdjęcie tablicy rejestracyjnej a następnie rozpoznaje znaki z tablicy i przetwarza je na tekst.
- 4.2.2. Numery tablic rejestracyjnych oraz wybrane przez użytkownika tryby mycia zapisywane są do bazy danych. W bazie danych widnieje również informacja na temat czasu rozpoczęcia mycia.
- 4.2.3. Informacje z bazy danych są przetwarzane w celu prezentacji statystyk.
- 4.2.4. Na podstawie danych zapisanych w bazie tworzony jest histogram wyboru trybów mycia. Wyświetlane są numery tablicy rejestracyjnej pojazdu, który najczęściej odwiedza myjnię oraz informacje na temat najczęściej i najrzadziej wybieranych trybów mycia pojazdu.

## 5. Opis błędów

### 5.1. Procedury testowe

#### 5.1.1. VI Analyzer

W celu przetestowania aplikacji pod kątem błędów oraz niezgodności z ogólnie przyjętymi praktykami programowania w LabVIEW został użyty VI Analyzer. Podczas testowania aplikacji wyniknęły pewne błędy które łatwo było ominąć przy manualnym ich poszukiwaniu.



Rys. 6. Rezultat działania narzędzia VI Analyzer

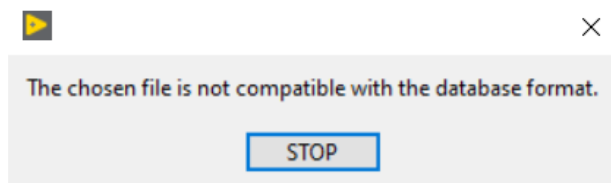
Na rysunku 6 widoczny jest zrzut ekranu prezentujący działanie narzędzia VI Analyzer, jego wynik jest również dostępny w folderze "Documentation" projektu. W ostatecznej wersji udało się wyeliminować wszystkie rażące błędy wskazane przez program. Pozostały głównie te nie wpływające na zachowanie programu.

#### 5.1.2. Trace Execution

Działanie aplikacji poddano testowi przy pomocy narzędzia Trace Execution. Log z działania narzędzia załączono do reszty plików dokumentacji w folderze Documentation.

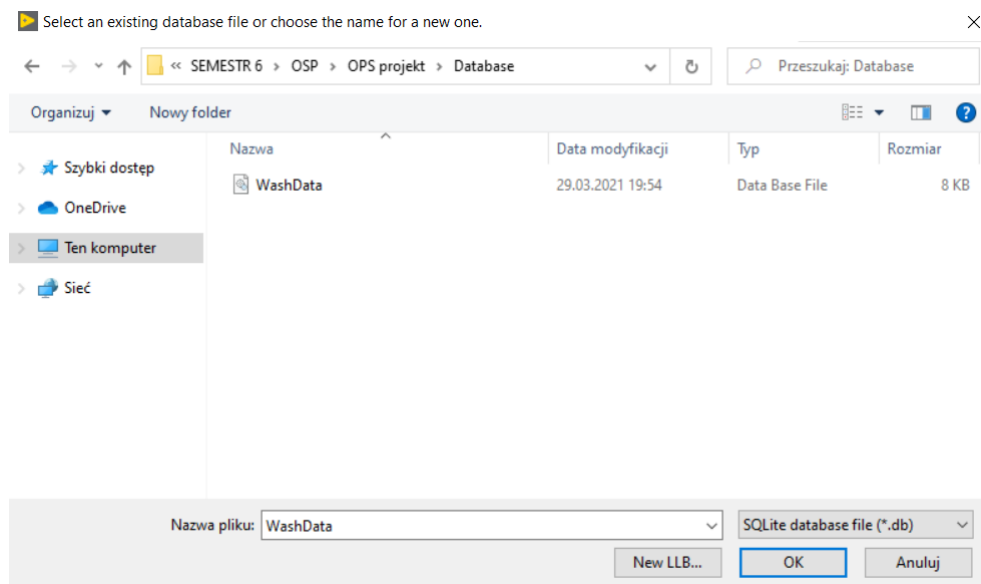
## 5.2. Obsługa błędów

W stworzonej aplikacji zaimplementowano mechanizm obsługi zdarzeń wyjątkowych w szczególności błędów, których wystąpienie zmienia prawidłowy przebieg działania aplikacji. Podjęcie akcji następuje po nieplanowanym działaniu użytkownika takim jak wybór pliku do wczytania/nadpisania bazy danych z rozszerzeniem innym niż dopuszczalne tj. “.db”. Wówczas aplikacja nie utworzy połączenia z bazą danych, a użytkownikowi wyświetlona zostanie odpowiednia informacja z możliwością zatrzymania aplikacji.



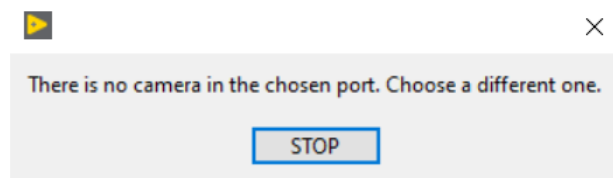
Rys. 7. Informacja dla użytkownika o błędnym wyborze pliku bazy danych

Anulowanie opcji wyboru bazy danych spowoduje utworzenie domyślnego, pustego pliku o rozszerzeniu “.db” i nazwie zawierającej obecną datę np. “WashData\_2021.04.14\_14-36-00”.



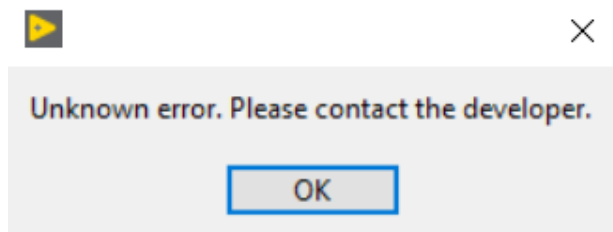
Rys. 8. Okno wyboru pliku bazy danych

Do poprawnego działania aplikacji konieczne jest podłączenie urządzenia rejestrującego obraz (kamera). W tym celu w zakładce “Admin Panel” konieczne jest wybór podłączonego urządzenia. Wybranie nieistniejącego urządzenia wygeneruje błąd dlatego konieczne jest zatrzymanie aplikacji, a także poinformowanie użytkownika o błędnym wyborze i zalecenie podłączenia poprawnie działającego sprzętu.



*Rys. 9. Informacja dla użytkownika o błędnym wyborze urządzenia rejestrującego obraz*

Cały system jest odporny na inne nieokreślone działania, co za tym idzie nie zaburzają one prawidłowego działania aplikacji. W przypadku wystąpienia nieznanego dotąd błędu użytkownik zostanie poproszony o ewentualny kontakt z twórcami.



*Rys. 10. Informacja dla użytkownika o nieznanym dotąd błędzie*

## 6. Bibliografia

- [1] <https://github.com/vieczorkamil/CarWash>
- [2] <https://bitbucket.org/drijdpowell/sqlite-labview/src/master/>
- [3] <https://www.ni.com/pl-pl/support/documentation.html>

Data sprawdzenia dostępu do stron pod powyższymi adresami: 19.04.2021