serwer CMMS CSV

Serwer CMMS CSV jest dodatkiem do programu CMMS Maszyna SMART a jego zadaniem jest generowanie w czasie rzeczywistym plików CSV z danymi reprezentującymi podstawowe rejestry systemu.
Generowane są pliki reprezentujące historię RM, harmonogram PM, rejestr wyposażenia oraz kilka innych rejestrów i słowników.

Wybrano pliki CSV ze względu na ich nieustanną popularność i prostotę tworzenia z ich pomocą źródeł danych dla Excela, Power Bi i praktycznie każdego języka programowania.

Dodatkową funkcjonalnością jest generowanie zgłoszeń awarii na podstawie pliku CSV wygenerowanego np. przez system SCADA.

Instalacja

Program dostarczony jest w formie folderu z plikami który należy skopiować na komputer serwer a następnie skonfigurować połączenie z bazą danych SQL w pliku serwer.ini tak jak w programie klieckim.

Następnie należy dokonać szczegółowej konfiguracji w pliku csv_set.ini zwracając szczególną uwagę na foldery csv_data i csv_cmd. Domyślnie są one podkatalogami w katalogu aplikacji ale możemy to zmienić.

Wersja testowa

Na życzenie udostępniamy wersję testową. Wersja testowa:

- pozwala jedynie na generowanie plików ręcznie za pomocą konsoli
- mechanizm zgłaszania awarii działa tylko w trybie testowym z rędznym potwierdzeniem operacji

Rodzaje generowanych plików

System generuje 3 grupy plików CSV oraz pomocnicze pliki słownikowe

- rm1.csv ... rm8.csv — historia zdarzeń RM (awarie, usterki, zgłoszenia etc.)
- pm1.csv ... pm4.csv — historia i plan przeglądów prewencyjnych PM
- eq1.csv ... eq4.csv — listy wyposażenia EQUIPMENTS (maszyny, urządzenia, pojazdy, budynki etc.)
- dici.csv — lista indeksów wyposażenia / maszyn
- dicW.csv — lista wydziałów produkcyjnych
- dicp.csv — lista kategorii zdarzeń RM
- users.csv — lista użytkowników systemu

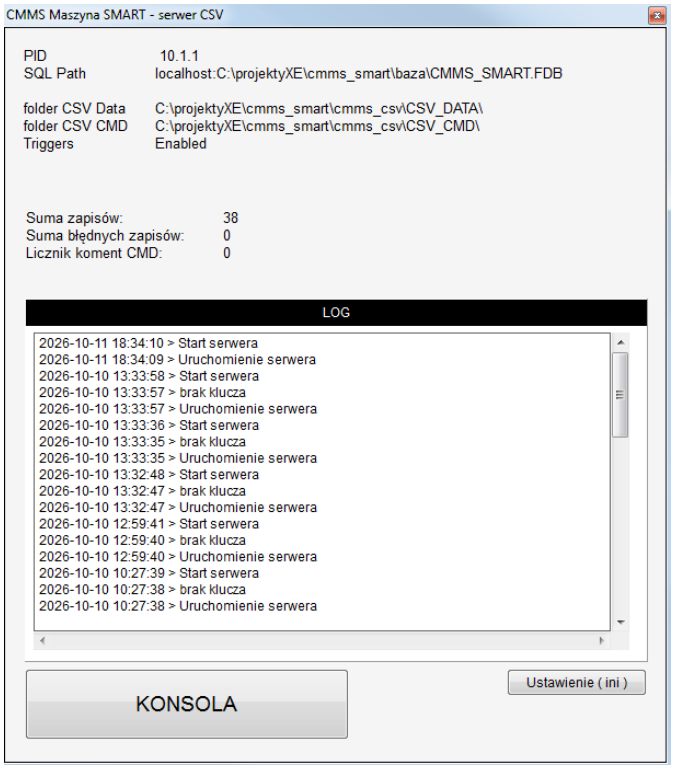
Uzupełnieniem plików CSV jest plik kontekstu i pliki statusu

Plik kontekstu AI_Context.MD stanowi kompletny opis plików, pól, ich znaczenia i konfiguracji
oraz powiązań między danymi.
Plik może wspomóc analizę danych przez modele AI. Plik może np. być wsparciem dla copilota w power Bi

Po każdym zapisie aktualizowane są pliki status.json i status.md które zawierają konfigurację plików i inforamacje o czasie ich utworzenia.

Okno programu i konsola

Okno programu



W oknie programu widzimy informacje o ścieżkach do bazy danych SQL oraz do katalogu dla plików CSV i katalogu dla pliku poleceń CMD

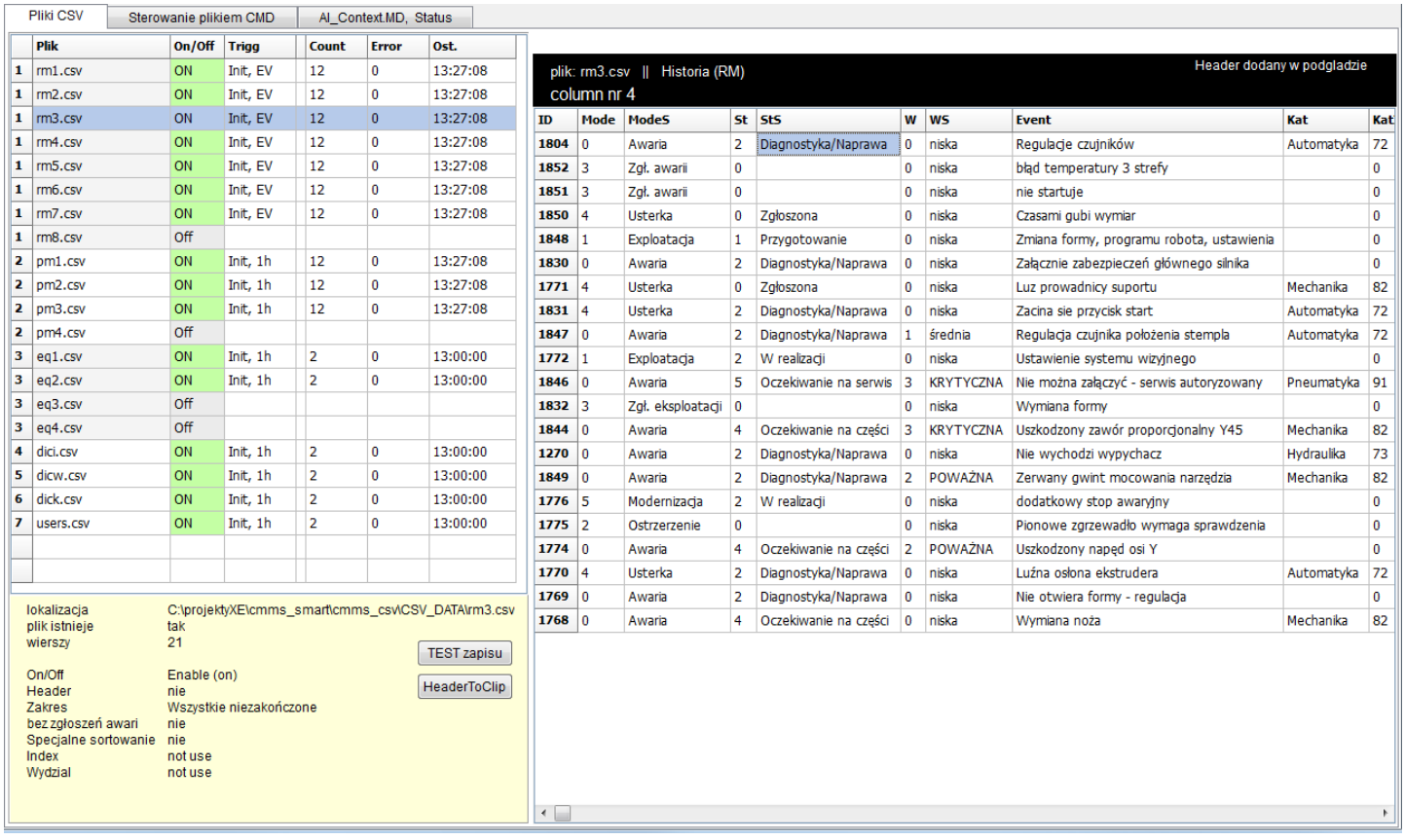
PID określa aktualną wersję programu
Pierwsza cyfra określa numer bazy programu CMMS. Aby zapewnić właściwą pracę cały system musi mieć PID taki sam lub większy.

Dostępny jest też log operacji, szczególnie załączeń i wyłączeń programu.

Przyciskiem [ustawienia] możemy otworzyć edytor pliku CSV_SET.ini ale musimy pamiętać że zmiany będą obowiązywały dopiero po restarcie.

Konsola

Konsola pozwala na śledzenie konfiguracji i pracy systemu oraz na testowe zapisy plików.



Wyzwalacze TRIGG

Wyzwalacz określa kiedy dany plik jest generowany. Wszystkie (aktywne) pliki korzystają z wyzwalacza Init czyli są zapisywane po uruchomieniu programu.
Wyzwalacz EV wykrywa zmiany w bazie danych i jest aktywowany ok minuty po ostatniej zmianie. Korzystają z niego pliki historii i harmonogramu.

Wyzwalacz 1h powoduje odświeżanie danych co godzinę. Musimy wiec mieć na uwadze to że zmiany w liście wyposażenia czy w słowniku muszą trochę poczekać nim zostaną uwzględnione w plikach CSV.

Ostatnim wyzwalaczem jest przycisk [Test zapisu] który pzowala na wygenerowanie pliku w dowolnym momencie np. gdy chcemy sprawdzić efekt zmiany konfiguracji.

Licznik zapisów i błędów zapisu

Do każdego pliku jest przypisany licznik zapisów i licznik błędnych zapisów. Błędny zapis prawie zawsze ma jeden powód. Program próbuje zapisać plik który jest otwarty przez inny program.
W praktyce wygląda to tak: serwer zapisuje plik CSV, my w ten plik „klikamy” otwierając go w Excelu. Stacja próbuje zapisać plik ponownie ale nie może bo … jest otwarty. W takim przypadku stacja przechwytuje ten błąd „na cicho” i zwiększa licznik błędów zapisu.

Konfiguracja – plik csv_set.ini

Cała konfiguracja zawarta jest w pliku ini gdzie mamy sekcję [set] z ustawieniami globalnymi, sekcję [set_cmd] z ustawieniami mechanizmu zapisu zgłoszeń do bazy danych oraz sekcje dla każdego z plików csv.

```
[set]
enable=1        // globalne załączenie / wyłączenie systemu (wyłączone wszystkie triggery)
path_data=      // ścieżka do katalogu w którym będą składowane tworzone pliki CSV.
                // pusty klucz to podkatalog CSV_DATA w katalogu stacji
path_cmd=       // ścieżka do katalogu w którym będą składowane pliki sterujące CSV.
                // pusty klucz to podkatalog CSV_CMD katalogu stacji
```

Uwaga.
Prawdopodobnie będziemy chcieli składować pliki w katalogach udostępnionych w sieci lub na dysku sieciowym. Zaleca się aby był to dysk na komputerze stacji a nie na innej maszynie w sieci aby uniknąć zakłóceń podczas chwilowej utraty dostępu do tych katalogów czego w sieci nie da się całkowicie wykluczyć.

```
Delimiter=;     // csv znak rozdzielający
QuoteChar="     // csv znaki otaczające tekst
Decimal=.       // separator dziesiętny
```

Ustawienia dla wszystkich rodzajów plików:

```
enable=1        // załącza tworzenie i obsługę pliku
```

Każdy plik ma parametr Enable który określa czy ma być obsługiwany (załączony)
Jest też globalny klucz Enable w sekcji [set] którym można włączyć / wyłączyć całą aktywność systemu poprzez wyłączenie wszystkich wyzwalaczy ale pozostawiając możliwość ręcznego wyzwalania zapisu w konsoli.

```
header =1       // określa czy w pliku CSV w pierwszym wierszu mają być nazwy kolumn
```

Każdy zapisywany plik csv może, ale nie musi mieć wiersz z nazwami kolumn. Jeśli u ustawieniach danego pliku wyłączymy nazwy kolumn to zostaną one dodane w podglądzie w konsoli co zostanie wyraźnie zaznaczone.
Plik sterujący dodawaniem zgłoszeń awarii za_cmd.css musi mieć nagłówek.

Konfiguracja plików RM1 … RM8.csv

rm_range=0 // zakres czasu dla którego pobierane są dane
0) bieżący miesiąc
1) bieżący rok
2) rekordy do początku podanego roku (year)
3) wszystkie niezakończone zdarzenia bez kontroli czasu

year=2026 // rok dla rm_range=2

spec_sort=0 // wymusza sortowanie według czasu i typów zdarzenia w kolejności: zgłoszenie, awaria, usterka etc.
// najczęściej stosowany z parametrem rm_range=3

no_fail_note=1 // powoduje pominiecie w zestawieniu zgłoszeń awarii
no_big_text=1 // ogranicza zawartość pól notatnikowych (wielowierszowych) do 100 znaków

index=0 // filtr ograniczający zbiór zdarzeń przypisanych do obiektów z przypisanym ID indeksu
wydzial=0 // filtr ograniczający zbiór zdarzeń przypisanych do obiektów z przypisanym ID wydziału

Konfiguracja plików PM1 … PM4.csv

pm_range=0 // zakres czasu dla którego pobierane są dane
0) bieżący rok - plan/w trakcie/przeterminowane
1) wszystkie plan/w trakcie/przeterminowane
2) wszystkie od początku roku year

year=2026 // rok dla pm_range=2
no_big_text=1 // ogranicza zawartość pól notatnikowych (wielowierszowych) do 100 znaków
index=0 // filtr ograniczający zbiór zadań przypisanych do obiektów z przypisanym ID indeksu
wydzial=0 // filtr ograniczający zbiór zadań przypisanych do obiektów z przypisanym ID wydziału

Konfiguracja plików PM1 … PM4.csv

no_big_text=1 // ogranicza zawartość pól notatnikowych (wielowierszowych) do 100 znaków
index=0 // filtr ograniczający zbiór obiektów z przypisanym ID indeksu
wydzial=0 // filtr ograniczający zbiór obiektów z przypisanym ID wydziału

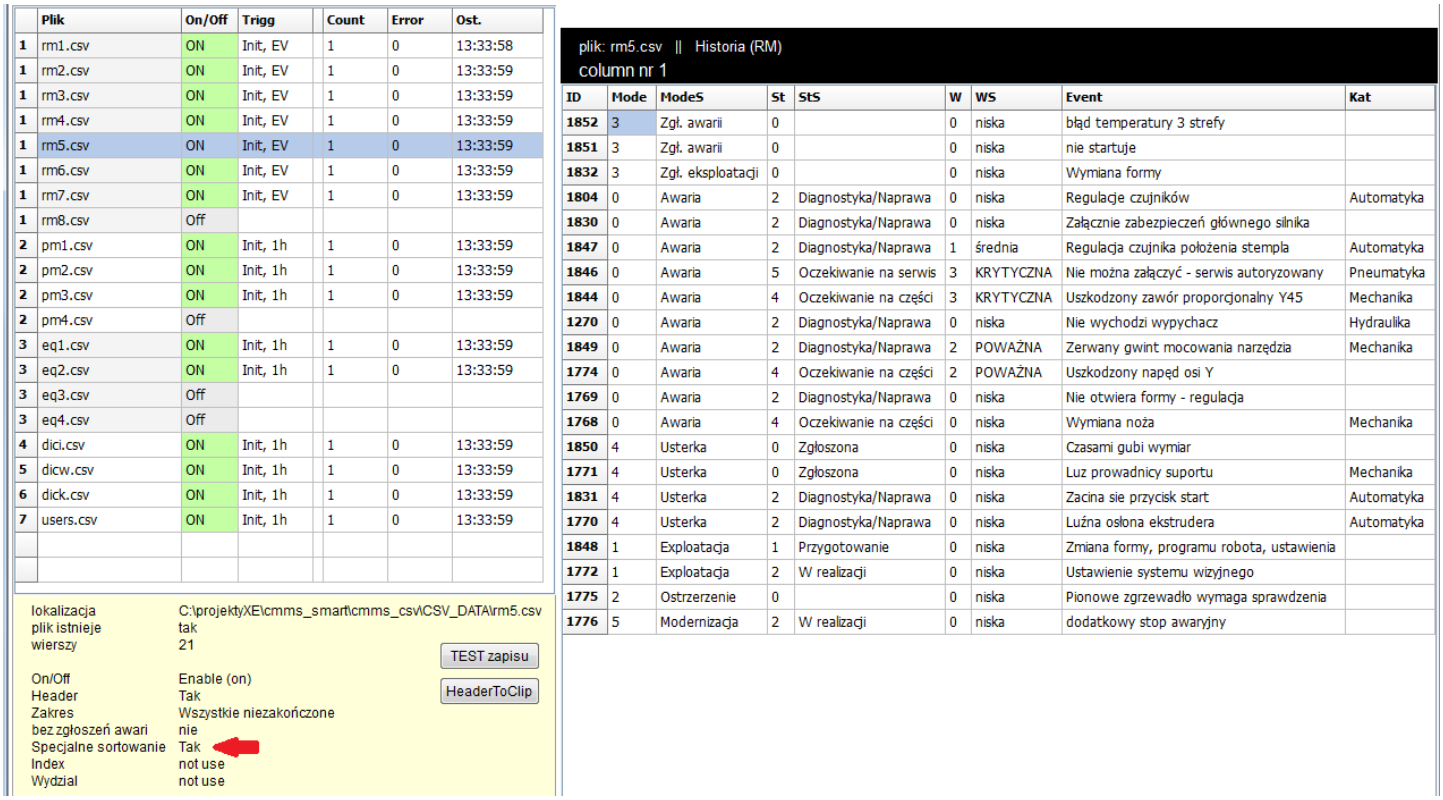
Konfiguracja systemu dodawania zgłoszeń

[set_cmd]
enable=1 // załączenie systemu
test_mode=1 // system w trybie testowym

filtry wydział i index

Każda maszyna ma przypisany wydział i może być przypisana do dowolnej ilości indeksów. Zastosowanie filtrów pozwala na ograniczenie maszyn w plikach eqX.csv oraz zdarzeń i zadań w plikach rmX.csv i pmX.csv do wybranych obszarów produkcyjnych.
Ma to szczególne znaczenie gdy w rejestrach mamy urządzenia pomocnicze i wyposażenie którego nie chcemy mieć w wygenerowanych plikach.

Specjalne sortowanie



	Plik	On/Off	Trigg	Count	Error	Ost.
1	rm1.csv	ON	Init, EV	1	0	13:33:58
1	rm2.csv	ON	Init, EV	1	0	13:33:59
1	rm3.csv	ON	Init, EV	1	0	13:33:59
1	rm4.csv	ON	Init, EV	1	0	13:33:59
1	rm5.csv	ON	Init, EV	1	0	13:33:59
1	rm6.csv	ON	Init, EV	1	0	13:33:59
1	rm7.csv	ON	Init, EV	1	0	13:33:59
1	rm8.csv	Off				
2	pm1.csv	ON	Init, 1h	1	0	13:33:59
2	pm2.csv	ON	Init, 1h	1	0	13:33:59
2	pm3.csv	ON	Init, 1h	1	0	13:33:59
2	pm4.csv	Off				
3	eq1.csv	ON	Init, 1h	1	0	13:33:59
3	eq2.csv	ON	Init, 1h	1	0	13:33:59
3	eq3.csv	Off				
3	eq4.csv	Off				
4	dici.csv	ON	Init, 1h	1	0	13:33:59
5	dicw.csv	ON	Init, 1h	1	0	13:33:59
6	dick.csv	ON	Init, 1h	1	0	13:33:59
7	users.csv	ON	Init, 1h	1	0	13:33:59

plik: rm5.csv || Historia (RM)
column nr 1

ID	Mode	ModeS	St	StS	W	WS	Event	Kat
1852	3	Zgł. awarii	0		0	niska	błąd temperatury 3 strefy	
1851	3	Zgł. awarii	0		0	niska	nie startuje	
1832	3	Zgł. eksploatacji	0		0	niska	Wymiana formy	
1804	0	Awaria	2	Diagnostyka/Naprawa	0	niska	Regulacje czujników	Automatyka
1830	0	Awaria	2	Diagnostyka/Naprawa	0	niska	Załącznie zabezpieczeń głównego silnika	
1847	0	Awaria	2	Diagnostyka/Naprawa	1	średnia	Regulacja czujnika położenia stempla	Automatyka
1846	0	Awaria	5	Oczekiwanie na serwis	3	KRYTYCZNA	Nie można załączyć - serwis autoryzowany	Pneumatyka
1844	0	Awaria	4	Oczekiwanie na części	3	KRYTYCZNA	Uszkodzony zawór proporcjonalny Y45	Mechanika
1270	0	Awaria	2	Diagnostyka/Naprawa	0	niska	Nie wychodzi wypychacz	Hydraulika
1849	0	Awaria	2	Diagnostyka/Naprawa	2	POWAŻNA	Zerwany gwint mocowania narzędzia	Mechanika
1774	0	Awaria	4	Oczekiwanie na części	2	POWAŻNA	Uszkodzony napęd osi Y	
1769	0	Awaria	2	Diagnostyka/Naprawa	0	niska	Nie otwiera formy - regulacja	
1768	0	Awaria	4	Oczekiwanie na części	0	niska	Wymiana noża	Mechanika
1850	4	Usterka	0	Zgłoszona	0	niska	Czasami gubi wymiar	
1771	4	Usterka	0	Zgłoszona	0	niska	Luz prowadnicy suportu	Mechanika
1831	4	Usterka	2	Diagnostyka/Naprawa	0	niska	Zacina sie przycisk start	Automatyka
1770	4	Usterka	2	Diagnostyka/Naprawa	0	niska	Luźna osłona ekstrudera	Automatyka
1848	1	Eksploatacja	1	Przygotowanie	0	niska	Zmiana formy, programu robota, ustawienia	
1772	1	Eksploatacja	2	W realizacji	0	niska	Ustawienie systemu wizyjnego	
1775	2	Ostrzerzenie	0		0	niska	Pionowe zgrzewadło wymaga sprawdzenia	
1776	5	Modernizacja	2	W realizacji	0	niska	dodatkowy stop awaryjny	

Dla podglądanego pliku pokazującego wszystkie niezakończone zdarzenia załączono opcję specjalnego sortowania. Normalnie sortowanie jest wg kolumny TA.
Tu mamy zdarzenia w kolejności: zgłoszenie awarii, zgłoszenie eksploatacji, awarie, usterki, eksploatacja, ostrzeżenia i na koniec modernizacja. Dopiero drugim poziomem sortowania jest kolumna TA

Taki sam porządek sortowania mają niektóre aplikacje mobilne, np. aplikacja do wyświetlania zdarzeń na telewizorach.

Budowa plików CSV

pliki RM1 .. RM8.CSV

```
0.  || ID       // ID zdarzenia (klucz główny)
1.  || Mode     // Tryb (rodzaj) zdarzenia (kod liczbowy)
2.  || ModeS    // Tryb (rodzaj) zdarzenia tekstowo
3.  || St       // Status zdarzenia (kod liczbowy)
4.  || StS      // Status zdarzenia tekstowo
5.  || W        // Waga zdarzenia (kod liczbowy)
6.  || WS       // Waga zdarzenia tekstowo
7.  || Event    // Skrócony opis zdarzenia
8.  || Kat      // Kategoria zdarzenia ( np. Automatyka, Mechanika, Elektryka etc. )
9.  || KatID    // ID słownika kategorii
10. || TA       // Zgłoszenie zdarzenia (Data i godzina)
11. || TB       // Rozpoczęcie obsługi zdarzenia (Data i godzina)
12. || TE       // Zakończenie obsługi zdarzenia (Data i godzina)
13. || Twait    // Czas oczekiwania na interwencję
14. || Tserv    // Czas obsługi
15. || Twork    // Czas pracy
16. || Tsusp    // Czas zawieszenia
17. || Ttotal   // Czas całkowity
18. || Sec      // Czas od TA w sekundach
19. || OBID     // ID Obiektu
20. || OBName   // Nazwa obiektu/maszyny
21. || OBSign   // Oznaczenie
22. || OBCode   // Symbol
23. || OBNf     // Numer fabryczny
24. || OBNe     // Numer ewidencyjny
25. || OBDept   // Wydział
26. || OBABC    // Krytyczność ABC
27. || Opis     // Pełen opis zdarzenia (zgłoszenia)
28. || OpisPD   // Opis podjętych działań
29. || TypA     // Typ awarii
30. || typE     // Typ eksploatacji
31. || Koszt    // Szacunkowy koszt bez części.
32. || KosztC   // Koszt części wynikający z rozchodu części
```

```
33. || UID      // ID pracownika odpowiedzialnego (zgłaszającego)
34. || USER     // Pracownik odpowiedzialny (nazwisko i imię)
35. || Ex       // 1: zgłoszenie awarii; 2: zgłoszenie eksploatacji
```

pliki PM1 .. PM4.CSV

```
0.  || ID       // ID zadania (klucz główny)
1.  || St       // Status zadania (kod liczbowy)
2.  || StS      // Status zadania tekstowo
3.  || Task     // Skrócony opis zadania
4.  || Pr       // Priorytet
5.  || PTime    // Przewidywany czas pracy
6.  || Wz       // Wymaga zatrzymania (1: Tak, 0: Nie)
7.  || TA       // Zaplanowano (Data)
8.  || TB       // Rozpoczęcie (Data)
9.  || TE       // Zakończenie (Data)
10. || Days     // Za ile dni lub o ile dni przeterminowane
11. || Tstop    // Szacowany czas przestoju maszyny
12. || TWork    // Szacowany czas pracy (roboczogodziny)
13. || Koszt    // Szacunkowy koszt bez części.
14. || KosztC   // Koszt części wynikający z rozchodu części
15. || OBID     // ID Obiektu / Maszyny
16. || OBName   // Nazwa obiektu/maszyny
17. || OBSign   // Oznaczenie maszyny
18. || OBCode   // Symbol
19. || OBNf     // Numer fabryczny maszyny
20. || OBNe     // Numer ewidencyjny maszyny
21. || OBDept   // Wydział produkcyjny
22. || OBABC    // Krytyczność maszyny w klasyfikacji ABC
23. || Opis     // Pełen opis zadania
24. || OpisP    // Opis podsumowania (podjęte działania)
25. || UID      // ID pracownika odpowiedzialnego
26. || Usr      // Pracownik odpowiedzialny (nazwisko i imię)
```

pliki EQ1 .. EQ4.CSV

```
0.  || ID       // ID Obiektu
1.  || Nazwa    // Nazwa obiektu/maszyny
2.  || Sign     // Jednoznaczne, krótkie oznaczenie
3.  || Sybol    // Symbol, np. symbol fabryczny
4.  || SOpis    // Krótki opis
5.  || Typ      // Typ - numer id słownika typów wyposażenia
6.  || TypS     // Nazwa typu wyposażenia w słowniku
7.  || Nrfab    // Numer fabryczny
8.  || Nrewid   // Numer ewidencyjny
9.  || UDT      // Numer UDT
10. || ABC      // Krytyczność ABC
11. || Wp       // Wydział produkcyjny
12. || WpS      // Wydział produkcyjny symbol
13. || Lok      // Lokalizacja
14. || Lpr      // Linia produkcyjna
15. || Rok      // rok produkcji
16. || Stech    // Stan techniczny
17. || Uwagi    // Opis/Uwagi
18. || Prod     // Producent
19. || Count    // roboczogodziny
```

pliki pomocniczy DICI.CSV

```
0.  || ID       // ID ( używany dla filtru indexu )
1.  || Index    // Nazwa indexu
```

pliki pomocniczy DICW.CSV

```
0.  || ID       // ID ( używany dla filtru wydizału )
1.  || Wydział  // Nazwa wydziału
2.  || Kod      // Kod literowy wydziału
```

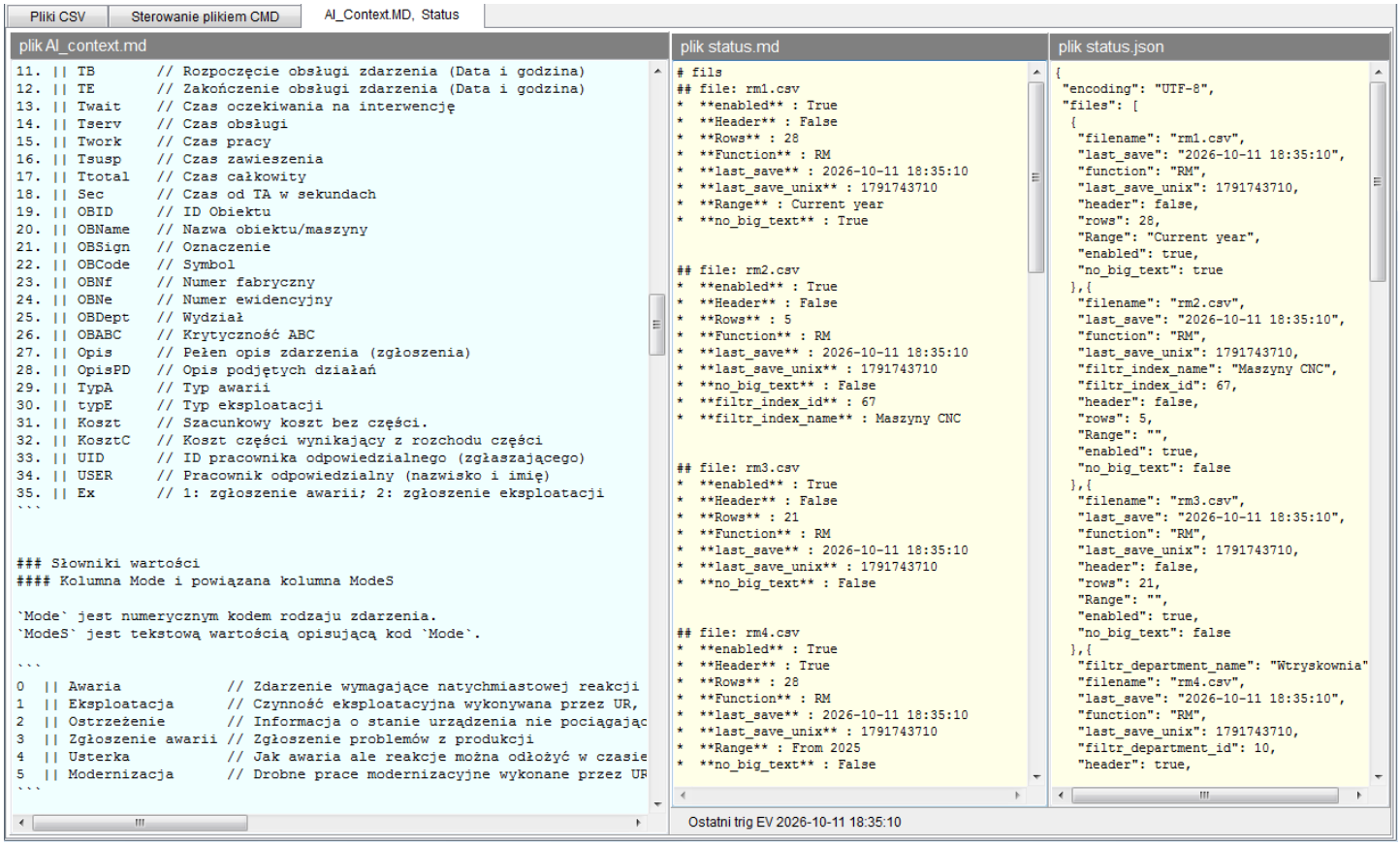
pliki pomocniczy DICK.CSV

```
d0.  || ID         // ID
1.  || Kategoria  // Nazwa kategorii
```

pliki pomocniczy USERS.CSV

```
0.  || ID        // ID
1.  || Nazwisko  // Nazwa kategorii
2.  || Imię      // Nazwa kategorii
3.  || Funkcja   // funkcja pełniona w strukturze przedsiębiorstwa
4.  || Profil    // określa rolę i poziom uprawnień w systemie
```

Pliki AI_Context.MD i pliku statusu



Często spotykamy się z taką oto narracją że wystarczy dostarczyć modelowi AI dostatecznie dużo danych a on zajdzie korelacje miedzy nimi. I tak może być jeśli dane są jednorodne i / lub bardzo dobrze zdefiniowane.
Niestety dane w systemie CMMS Maszyna SMART wprowadzane są przez ludzi, często po fakcie.

Jednym z rozdziałów opisujących pliki historii jest rozdział opisujący korelacje czasowe pomiędzy czasami TA, TB i TE oraz częste błędy i zaniechania przy tworzeniu danych przez użytkowników.

Plik może być bardzo użyteczny przy współpracy agenta Cpilot z Powier Bi i to za równo przy analizie zaimportowanych danych jak i na samym etapie tworzenia modelu danych.

Program generuje też plik statusu w formatach MD i json z opisem plików, ich ustawień i czasu ostatniego zapisu.

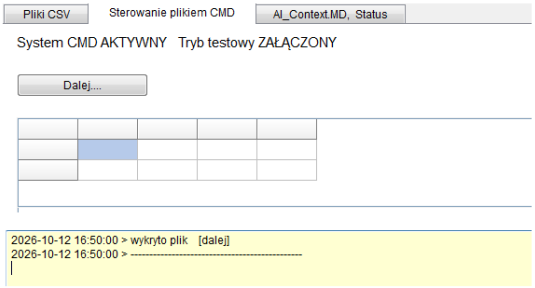
Dodawanie zgłoszeń awarii plikiem cmd

Koncepcja jest bardzo prosta. Program obserwuje folder CSV_CMD. Kiedy wykryje w tym folderze plik za_cmd.csv odczytuje go, odczytuje zawarte w nim parametry, wpisuje zgłoszenie awarii / eksploatacji do bazy danych i go kasuje. Po czym czeka na następny plik.

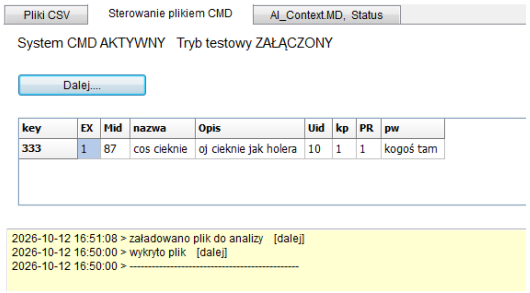
tryb testowy

W trybie testowym po każdym etapie operacji program czeka na potwierdzenie przyciskiem [Dalej]

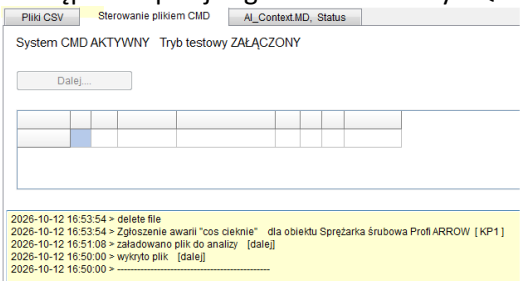
Po wykryciu pliku czeka na potwierdzenie tego faktu



Następnie ładuje plik do tabeli tak aby można było ocenić jego poprawność



Następnie zapisuje zgłoszenie do bazy SQL i kończy cykl oczekując na następny



Budowa pliku za_cmd.csv

```
0.  || key     // Pole o wartości 333 - wymagane do procedowania pliku
1.  || Ex      // Rodzaj zgłoszenia:  1- zgłoszenie awarii; 2-zgłoszenie eksploatacji
2.  || Mid     // ID obiektu / maszyny
3.  || Nazwa   // krótki opis zgłoszenia
4.  || Opis    // Opis zgłoszenia bez znaków końca linii
5.  || Uid     // ID osoby zgłaszającej. Może być 0
6.  || Kp      // kogo powiadomić ( liczbowy kod )
7.  || Pr      // priorytet ( liczbowy kod )
8.  || Pw      // tekstowy opis kogo powiadomione
```

Plik musi mieć wiersz z nazwami kolumn.

Kolumna Kp

```
0  || Nie podano
1  || Mechanik
2  || Elektryk
3  || Automatyk
4  || Inżynier procesu
5  || dział IT
```

Kolumna Pr

```
0  || Niski priorytet – maszyna pracuje
1  || Średni priorytet – potrzebne pomoc w krótkim czasie
2  || Zdarzenie krytyczne – potrzebna natychmiastowa pomoc
```

Wzór pliku za_cms:

key;EX;Mid;Nazwa;Opis;Uid;kp;PR;pw
333;1;87;nazwa zdarzenia;opis zdarzenia;1;1;2;jan kowalski