*Dokumentacja końcowa programu File Manipulator*

# Opis struktury

Projekt aplikacji został podzielony na bibliotekę podstawową (Core*, z ang. rdzeń*), która posiada w sobie warstwy dostępu do danych oraz logikę biznesową, a także projekt UI, który jest startowym projektem z plikiem wykonywalnym .exe.

Aplikacja wykorzystuje technologię *Windows Presentation Foundation* oraz wzorzec architekturalny *Model-View-ViewModel*, a do komunikacji między warstwami w większości wykorzystuje wzorzec architekturalny obserwatora (składający się z obserwatora i obiektu obserwowalnego) wspomagany przez bibliotekę *Microsoft Reactive Extensions*,  
a także wzorzec komendy, służący do wywoływania akcji oraz weryfikowania czy dana akcja może być wywołana.

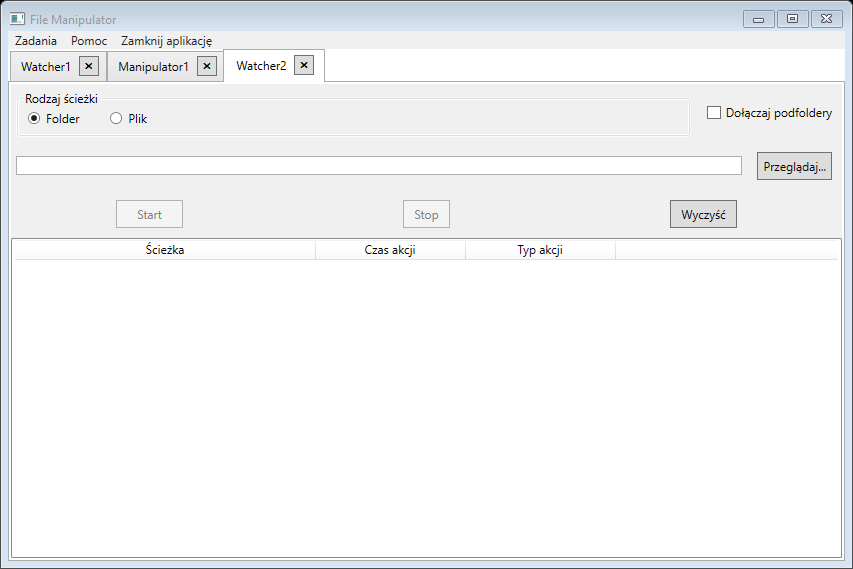
Między widokiem i modelami/view-modelami stosuje się mechanizm spinania danych, który automatycznie synchronizuje widok ze stanem logiki. Do pełnej synchronizacji widoków, potrzebna była implementacja interfejsu *INotifyPropertyChanged* do większości modeli.

Porównując się do architektury MVC – *viewmodele* są wobec tamtej architektury jako kontrolery zarządzające modelami i widokami.

Widoki zostały opisane w języku *XAML* (pochodnym od XML), co jest standardem we frameworku WPF. Daje to bezpieczeństwo zarządzania pamięcią oraz asynchroniczną pracę elementów okna. Wynikiem tego jest lepsza stabilność oraz wspomniana możliwość rozbicia struktury w celu poprawy czytelności.

Biblioteka Core jest w stanie działać niezależnie od obecnego projektu interfejsu, więc jest możliwość wypożyczenia logiki aplikacji do np. konsoli tekstowej lub interfejsu API (co daje możliwość w przyszłości uruchamiania makro poleceń).

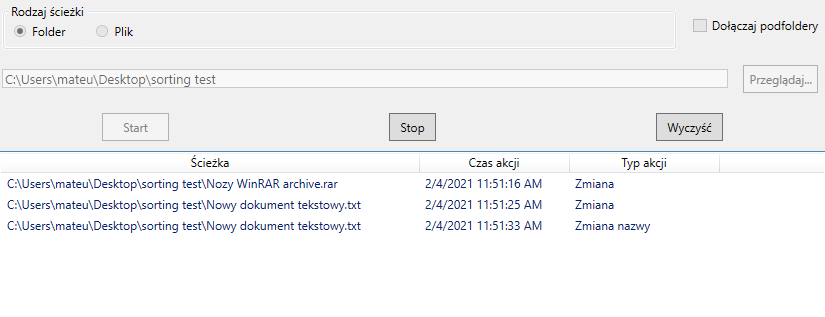
Większość wymagań funkcjonalnych została zaimplementowana więc w projekcie Core, jedynie filtracja katalogów systemowych w manipulatorze została wpisana do zachowania interfejsu graficznego, z uwagi na powiązanie z oknami dialogowymi.



# Opis interfejsu użytkownika

Głównym założeniem przy projektowaniu aplikacji była wielozadaniowość znana choćby z obsługi przeglądarek internetowych. Już na początku stwierdziliśmy, że to muszą być karty.  
Do okna zostało dodane menu z podstawowymi akcjami. W wersji rozwojowej myślimy nad wprowadzeniu interfejsu wstążek znanej z pakietu MS Office co z pewnością ułatwi czytelność.

W obecnej wersji mamy dwa rodzaje zadań możliwych do wykonywania – Watcher oraz Manipulator.



## Watcher

Jest to zadanie nasłuchujące zmiany w wybranym fragmencie systemu plików. W obecnej wersji interfejsu grupuje przechwycone akcje jako Zmianę (treści) oraz Zmianę nazwy. Choć logika rozróżnia bardziej szczegółowo to jednak pojawiły się problemy implementacyjne przy budowie interfejsu. Wersja rozwojowa zostanie znacznie przebudowana.

Narzędzie swoją przydatność pokazuje przy użytkowaniu instalatorów czy podczas prób wykrywania szkodliwego oprogramowania.  
Może być również monitoringiem przesłanych prac zaliczeniowych na serwer.

Obraz zawierający tekst

Opis wygenerowany automatycznie

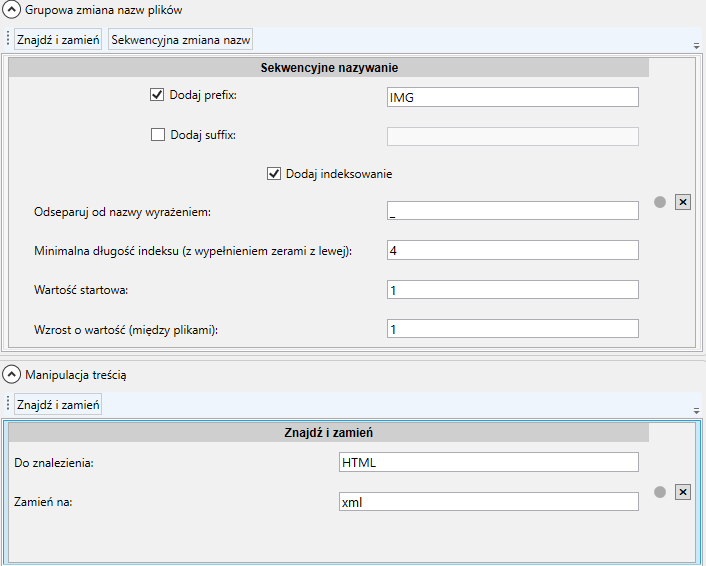
## Manipulator

Wszechstronne narzędzie służące do zarządzania dużymi zbiorami plików.  
Głównym założeniem tego zadania jest możliwość filtracji nazwy pliku oraz treści (na razie obsługuje tylko pliki tekstowe). Filtracja może być wykluczająca dany element (przykładowo filtracja wyrażeń regularnych) oraz sortująca. Co do sortowania – można znaleźć tutaj popularny algorytm sortowania alfanumerycznego używanego choćby w Eksploratorze Windows.  
Następne wersje zostaną rozbudowane o filtrację dat oraz rozmiarów

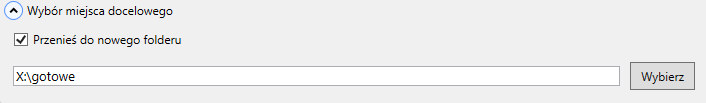
Obraz zawierający tekst

Opis wygenerowany automatycznie

W manipulacjach znajdziemy zamianę wyszukanych słów, czyszczenie nazwy oraz sekwencyjną zmianę nazw.



Ostatnim wyborem pozostaje docelowe miejsce przetworzonego pliku.



**UWAGA! Operacje wykonywane na systemie plików są nieodwracalne i w celu przywrócenia poprzedniego stanu może być potrzebne użycie oprogramowania do odzyskiwania danych.**

Każda operacja posiada kontrolkę pracy, która wskazuje stan wykonania.  
Na końcu wykonywania manipulacji na wybranym zbiorze wszystkie zmiany są finalizowane, a po tej operacji na pasku stanu wyświetli się komunikat o ukończeniu.

# Wdrożenie

Zalecanym środowiskiem uruchomieniowym jest system operacyjny MS Windows 10 w wersji 1909.

Minimalne wymagania to system MS Windows Vista oraz biblioteka .NET Framework 4.6, która domyślnie jest wbudowana w Windows 10.

Wdrożeniowiec musi zapewnić, aby wszystkie dostarczone pliki pozostały w jednym katalogu.

# Pozostałe

Aplikacja zostaje udostępniona na licencji MIT, z otwartym kodem źródłowym znajdującym się w repozytorium GitHub: <https://github.com/mateuszokroj1/FileManipulator>

W planach dalszy rozwój aplikacji – możliwe, że również będzie wersja mobilna na system Android.  
Jest również możliwość stworzenia wersji na platformę Linux i Mac OS, jednak będzie to wymagało przejścia na wersję frameworka .NET 5 oraz ewentualną zmianę technologii interfejsu graficznego.