

# Team: Cherry Pickers

Zadanie: sendHYBRID.GOV  
Projekt: PUHa

Mateusz Roszkowski,  
Aleksandra Majewska,  
Jan Bronowski,  
Jakub Zacharczuk,  
Mateusz Zagórski,  
Piotr Szmurło

# Frontend - layout



Uploaded Successfully! Upload another?

PDF,TXT,RT...

send\_hybrid\_gov\_18z.pdf

DELETE

✓ CASE\_TYPE valid

✓ DATE valid

✓ DEPARTMENT valid

✓ RECEIVER\_CITY valid

✓ RECEIVER\_NAME valid

✓ RECEIVER\_STREET valid

✓ RECEIVER\_ZIP\_CODE valid

✓ SENDER\_CITY valid

✓ SENDER\_EMAIL valid

✓ SENDER\_EPUAP valid

✓ SENDER\_NAME valid

✓ SENDER\_STREET valid

✓ SENDER\_ZIP\_CODE valid

✓ SIGNATURE valid

✓ UNP valid

# Widok okna logowania

 **SendHybrid**  
Ministerstwo  
Prac

e-mail

password

LOGGING IN

EXPORT CONFIGURATION

SUBMIT

- Frontend pozwala na jednoczesne udostępnienie wielu plików o wszelkich, wymaganych formatach
- Nie wszystkie funkcjonalności aplikacji webowej zostały zintegrowane z API
- Eksport konfiguracji nie został ukończony, natomiast dalszy rozwój zakłada udostępnienie tej opcji jedynie z poziomu konta admina
- Zwykły użytkownik ma dostęp jedynie do udostępniania plików

# Wysłany plik oraz szczegóły weryfikacji zwrócone przez API



SendHybrid

EXPORT CONFIGURATION

LOGGING IN

FILE UPLOAD RECENTS



Uploaded Successfully! Upload another?

PDF, ZIP, TXT, RT...

send\_hybrid\_gov\_18z.pdf

DELETE

✓ CASE\_TYPE valid

✓ DATE valid

✓ DEPARTMENT valid

✓ RECEIVER\_CITY valid

✓ RECEIVER\_NAME valid

✓ RECEIVER\_STREET valid

✓ RECEIVER\_ZIP\_CODE valid

✓ SENDER\_CITY valid

✓ SENDER\_EMAIL valid

✓ SENDER\_EPUAP valid

✓ SENDER\_NAME valid

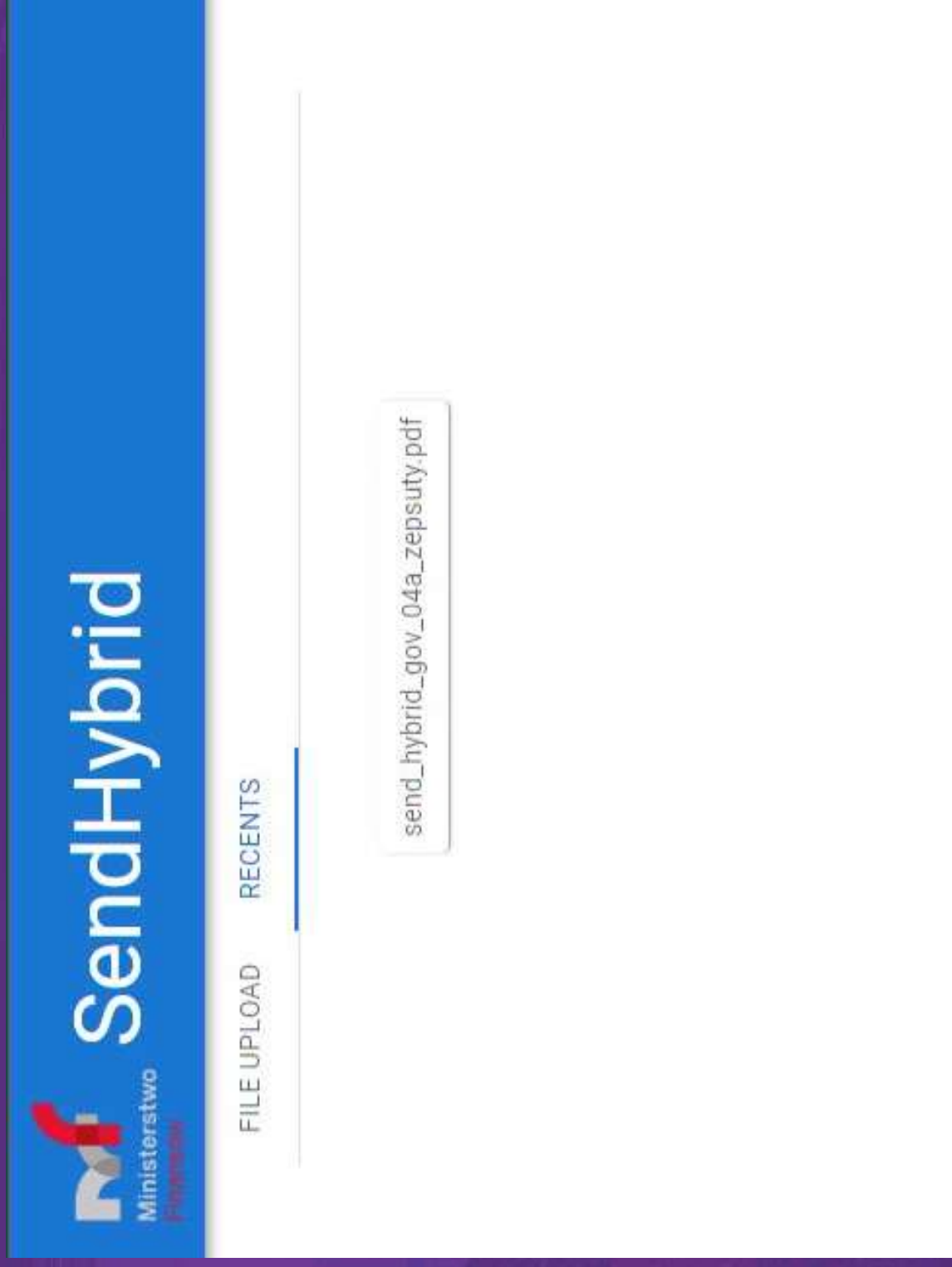
✓ SENDER\_STREET valid

✓ SENDER\_ZIP\_CODE valid

✓ SIGNATURE valid

✓ UNP valid

# Widok ostatnio weryfikowanych plików





# Wczytywanie MetaDanych z pliku

- Aby wczytywać tekst z pliku PDF zastosowaliśmy bibliotekę PyPDF2
- Wczytujemy stopkę oraz inne dane Odbiorcy i Nadawcy z pierwszej strony, a z ostatniej strony wczytujemy dane podpisu elektronicznego
- W czasie wytwarzania oprogramowania chcieliśmy użyć biblioteki Spacy, która wykorzystuje NER (Named-entity recognition) do znajdowania znaczenia słów korzystając z Machine Learning'u. Niestety baliśmy się przekroczyć wymagań systemowych, a same dokumenty są w wystarczający sposób pobodne, aby "scrapowanie danych" wykonać ręcznie
- czytywane Metadane:
  - Osoba podpisująca się elektronicznie;
  - Nadawca: adres mailowy, adres, ePUAP, Nazwa/Imię,
  - Data
  - UNP
  - Sprawa: rodzaj sprawy, znak sprawy
  - Odbiorca: Nazwa/Imię, Adres

# Weryfikacja pliku

- W systemie mamy plik z rozszerzeniem .json, który określa jak powinien zostać zweryfikowany PDF
- Na samym początku weryfikowana jest nazwa pliku i jest zmieniana (niektóre nazwy nie mogły się utworzyć w programie)
- Następnie weryfikujemy długość PDFa oraz jego metadane
- Weryfikacja podpisu elektronicznego, która informuje o dwóch stronach podpisu oraz jego dacie - funkcjonalność nie dodana do zbioru metadanych (Podgląd na GUI)
- Łatwość konfiguracji wymagań PDFa poprzez zmianę pliku .json przez rolę “admin”

# API

- REST
- Wykorzystuje OpenAPI Swagger
- Możliwość testowania z poziomu przeglądarki
- Wspiera wysyłanie, odbieranie plików i zmianę konfiguracji



## Dalszy rozwój funkcjonalności

- Tworzenie pliku PDF z raportem wgranego pliku - aktualnie jest zwracana informacja z raportu na stronie
- Automatyczna edycja pisma - aktualnie edytowana jest nazwa pliku, a możliwe jest rozszerzenie na edycję linii grubości linii/wielkości marginesu

# Dalszy rozwój backend

- Bezstanowość - dodanie integracji z bazą danych bądź cache, np. Redis. Umożliwi to skalowanie backendu wg potrzeb (wiele instancji backendu)
- Uwierzytelnienie po stronie backendu
- Dodanie RBAC (Role Based Access Control) - możliwość wywołania danego endpointu będzie możliwa w zależności od roli użytkownika
- Istnienie możliwości dodania deskryptorów K8s