



POLITECHNIKA POZNAŃSKA

WYDZIAŁ AUTOMATYKI, ROBOTYKI I ELEKTROTECHNIKI
Instytut Robotyki i Inteligencji Maszynowej



Praca dyplomowa inżynierska

AUTOMATYZACJA PROCESU PLANOWANIA WYDATKÓW NA USŁUGI IT W VW POZNAŃ Z WYKORZYSTANIEM MICROSOFT POWER PLATFORM

Remigiusz Wolniak, 151192

Michał Gajdzis, 151066

Promotor
dr hab. inż. Piotr Kaczmarek

POZNAŃ 2025

Zastrzeżenie dotyczące treści pracy dyplomowej

Niniejsza praca inżynierska zawiera treści, informacje itp. udostępnione przez spółkę Volkswa-
gen Poznań Sp. z o.o. z siedzibą przy ulicy Warszawskiej 349, 61-060 w Poznaniu, mogące stanowić
tajemnice przedsiębiorstwa tej spółki i mogące być wykorzystane wyłącznie dla potrzeb napisania
niniejszej pracy. Wobec powyższego niedozwolone jest wykorzystywanie całości lub części niniejszej
pracy, a także udostępnianie całości lub części pracy komukolwiek jak również kopiowanie, powie-
lanie, publikowanie itp. bez pisemnej zgody spółki Volkswagen Poznań Sp. z o.o. – zastrzeżenie to
nie ma zastosowania do przypadku udostępnienia niniejszej pracy nauczycielom akademickim w
celu oceny, recenzji i obrony ww. pracy. Podmioty, które naruszą powyższy zakaz ponoszą odpo-
wiedzialność odszkodowawczą wobec spółki Volkswagen Poznań Sp. z o.o.

Tutaj będzie skan karty pracy dyplomowej.

Spis treści

1	Wstęp	1
2	Podstawy teoretyczne	3
2.1	Struktura procesu	3
2.1.1	Gromadzenie danych dotyczących ofert usługodawców	3
2.1.2	Przygotowanie danych	3
2.1.3	Przebieg Iteracji	4
2.2	Wykorzystane technologie	5
2.2.1	Skrypty pakietu Office	5
2.2.2	SharePoint	5
2.2.3	Power Automate	6
2.2.4	Power Apps	6
3	Architektura rozwiązania	8
3.1	Założenia projektowe	8
3.1.1	Systematyzacja danych	8
3.1.2	Archiwizacja danych	9
3.1.3	Interfejs przyjazny dla użytkownika	9
3.1.4	Użycie pakietu Microsoft 365	9
3.1.5	Optymalizacja	10
3.2	Koncepcja rozwiązania	10
3.2.1	Baza danych	10
	Struktura bazy danych	10
	Atrybuty danych	10
	Model powiązań	11
3.2.2	Dodawanie informacji do bazy danych	11
3.2.3	Interfejs procesu decyzyjnego	12
	Ekran nawigacyjny	12
	Ekran szczegółowy	12
3.2.4	Generowanie raportu	13
4	Implementacja	14
4.1	Utworzenie bazy danych na platformie Sharepoint	14
4.2	Ekran dodawania danych	16
4.2.1	Zapis pliku w chmurze	16
	Przepływ SaveFileAndRunScript	17
4.2.2	Skrypt pakietu Office	19
4.2.3	Walidacja nazw kolumn	20

4.2.4	Integracja z listami SharePoint	21
4.2.5	Uzupełnianie numerów MPK	25
4.3	Ekran startowy aplikacji i przygotowanie danych	26
4.4	Edycja danych	28
4.4.1	Ekran wyboru usługi do edycji	28
4.4.2	Ekran edycji elementu	29
	Porównanie finalnych decyzji z poprzednich lat	29
	Link do instrukcji obsługi	29
	Porównanie tegorocznych indykacji	29
	Formularz do uzupełnienia danych	30
	Podsumowanie	30
4.5	Generowanie raportu	30
4.5.1	Łączenie źródeł danych do formy docelowej	31
4.5.2	Przekazywanie danych do Power Automate	32
5	Rozwinięcie	33
6	Zakończenie	34
	Literatura	35
A	Składanie dokumentu w systemie L^AT_EX	36
A.1	Struktura dokumentu	36
A.2	Akapity i znaki specjalne	36
A.3	Wypunktowania	36
A.4	Polecenia pakietu <code>ppcreefthesis</code>	37
A.5	Rysunki	37
A.5.1	Tablice	38
A.5.2	Przydatne uwagi	38
A.6	Literatura i materiały dodatkowe	39

Rozdział 1

Wstęp

Współczesny świat biznesu stawia coraz większe wymagania wobec przedsiębiorstw, zarówno w zakresie wydajności procesów, jak i precyzji podejmowanych działań. Powszechne metody zarządzania i przetwarzania danych, oparte są na manualnej pracy z wykorzystaniem mało efektywnych narzędzi oraz wymianie informacji w sposób niustandaryzowany. Stają się one niewystarczające w przypadku rosnącej skali operacji oraz wymagań co do szybkości i niezawodności podejmowanych decyzji. W obliczu tych wyzwań coraz większą rolę odgrywają rozwiązania z zakresu automatyzacji biurowej, które umożliwiają oszczędność czasu i zasobów oraz pozwalają na usprawnienie kluczowych procesów organizacyjnych, minimalizując ryzyko błędów ludzkich.

Jednym z obszarów, w którym automatyzacja znajduje zastosowanie, jest zarządzanie usługami IT i powiązanymi kosztami. W dużych organizacjach o rozbudowanej strukturze, konieczność gromadzenia, analizy oraz weryfikacji danych finansowych stanowi poważne wyzwanie. Dzięki wdrożeniu odpowiednich narzędzi, procesy te mogą być prowadzone w sposób uporządkowany i efektywny, umożliwiając jednocześnie bieżącą kontrolę nad wydatkami oraz lepsze planowanie budżetowe.

Ustandaryzowany i zautomatyzowany przepływ informacji ogranicza ryzyko powielania błędów i pozwala na skrócenie czasu potrzebnego na wykonanie poszczególnych zadań. Dodatkowo, wdrożenie automatyzacji zapewnia większą przejrzystość i ułatwia dostęp do informacji każdemu uczestnikowi procesu.

W dobie intensywnej cyfryzacji przedsiębiorstw oraz dynamicznego rozwoju technologii, automatyzacja biurowa staje się konieczna, aby sprostać wymaganiom współczesnego rynku. Odpowiednio zaprojektowane systemy i narzędzia wspierają nie tylko wydajność operacyjną, ale także strategiczne zarządzanie zasobami, umożliwiając rozwój w innych obszarach swojej działalności.

Celem pracy jest opracowanie aplikacji, usprawniającej proces podejmowania decyzji dotyczących zakupu usług IT¹ na najbliższy rok kalendarzowy. Praca została wykonana z wykorzystaniem *Power Platform* oraz *SharePoint*, które są integralną częścią pakietu *Microsoft 365*. Zdecydowano się na wybór tego rozwiązania, ponieważ pozwala ono na prostą integrację między programami wchodzącymi w skład pakietu. Ponadto, każdy z uczestników procesu ma dostęp do wspomnianych serwisów, co pozwala uniknąć dodatkowych kosztów.

DOPISAC:

Struktura pracy jest następująca. W rozdziale 2 przedstawiono przegląd literatury na temat ...

¹ Usługi IT należy rozumieć jako licencje oraz klucze dostępu do używanych systemów informatycznych.

Rozdział 3 jest poświęcony ... (kilka zdań).

Rozdział 4 zawiera ... (kilka zdań) ... itd.

Rozdział X stanowi podsumowanie pracy.

W przypadku prac inżynierskich zespołowych lub magisterskich 2-osobowych, po tych dwóch w/w akapitach musi w pracy znaleźć się akapit, w którym będzie opisany udział w pracy poszczególnych członków zespołu. Na przykład:

Jan Kowalski w ramach niniejszej pracy wykonał projekt tego i tego, opracował ... Grzegorz Brzęczyszczkiewicz wykonał ..., itd.

Rozdział 2

Podstawy teoretyczne

2.1 Struktura procesu

Przedmiotem omawianego procesu jest podjęcie decyzji dotyczących zakupu usług IT w zakładzie Volkswagen Poznań. Proces ten polega na wielokrotnej wymianie uwag dotyczących wcześniej używanego lub nowego oprogramowania między oddziałem Volkswagena w Poznaniu a zakładem z siedzibą w Wolfsburgu.

W wyniku wymiany zdań zapada decyzja o zakupie lub rezygnacji z wybranego produktu. Procedura, zazwyczaj podzielona na cztery indykacje¹, rozpoczyna się wraz z początkiem czerwca i trwa do końca roku.

Efektem podejmowanych działań jest nabycie odpowiedniej ilości potrzebnych uprawnień licencyjnych. Przy podejmowaniu decyzji kluczowymi aspektami są:

- liczba użytkowników danego oprogramowania,
- cena zakupu w porównaniu z rokiem poprzednim,
- określenie, czy dana usługa zostanie w pełni wykorzystana biorąc pod uwagę poprzednie kryteria.

Dotychczas analiza i przetwarzanie danych odbywały się przy użyciu arkuszy kalkulacyjnych programu Excel, a wymiana informacji między jednostkami była realizowana za pomocą wiadomości e-mail.

2.1.1 Gromadzenie danych dotyczących ofert usługodawców

Informacje na temat serwisów są zbierane na początku roku, przed rozpoczęciem cyklu procesu. W tym czasie, prowadzone są rozmowy między menadżerami odpowiedzialnymi za dane rozwiązanie (*BSM*, ang. *Business Service Manager*) a firmami świadczącymi usługi, w celu otrzymania zaaktualizowanych wiadomości związanych z ich produktami. Na podstawie danych od usługodawców oraz menadżerów, powstaje arkusz, który jest przekazywany do zakładu w Poznaniu.

2.1.2 Przygotowanie danych

Otrzymany arkusz kalkulacyjny, zawiera tabelę o strukturze kolumn podobnej do tabeli 2.1. Brakuje w nim jednak informacji kluczowych do rozpoczęcia cyklu. Dlatego pierwszym krokiem

¹indykacja - wstępne głosowanie

jest przygotowanie danych przez osobę nadzorującą proces ze strony oddziału w Poznaniu. Jej zadaniem jest manualne przypisanie numeru określającego miejsce powstawania kosztów, wewnętrznie nazywanego *MPK*. Numer ten definiuje konkretną jednostkę należącą do obszaru IT, która decyduje o zakupie danego produktu. Ponadto dodawana jest kolumna, w której znajduje się wyliczona różnica cen między rokiem obecnym a poprzednim, w celu określenia czy koszt wzrósł lub zmalał. Tak przetworzony plik zostaje umieszczony we wspólnej przestrzeni dyskowej, co umożliwia pozostałym uczestnikom procesu przystąpienie do analizy oraz dalszego przetwarzania zawartych w nim informacji.

TABELA 2.1: Nagłówki kolumn z arkusza kalkulacyjnego z roku 2022

Service group	Service main group	Service sub group	Business Service	ID	Business Service Manager	Unit of Measurement	PL70 2022 PLAN EUR w KVA	QTY	PL71 2023 PLAN EUR w KVA	QTY
---------------	--------------------	-------------------	------------------	----	--------------------------	---------------------	--------------------------	-----	--------------------------	-----

2.1.3 Przebieg Iteracji

W trakcie trwania iteracji rozpatrywane są kluczowe informacje takie jak:

- **Unit of Measurement - trzeba jakoś wytłumaczyć o co to chodzi,**
- decyzja podjęta w roku poprzednim.
- cena oraz ilość użytkowników w roku obecnym,
- cena oraz ilość użytkowników w roku przyszłym,

Po analizie i porównaniu danych z wcześniejszych lat, w arkuszu powstają kolejne kolumny. Ich struktura nie jest określona przez żaden standard, ale zazwyczaj zawierają one:

- Komentarz wewnętrzny,
- Status,
- Komentarz klienta.

Komentarz wewnętrzny nie jest wymagany dla każdego serwisu. Jest on zapisywany w celu skonsultowania decyzji ze współpracownikami.

Status określa wstępną, wymaganą decyzję (Zaakceptowany/Niezaakceptowany).

Komentarz klienta zawiera uzasadnienie podjętej decyzji ze strony Volkswagen Poznań.

Tak uzupełniony arkusz zostaje przekazany pośrednio przez zakład w Wolfsburgu, do zarządu firmy.

Kolejnym etapem jest analiza tych informacji przez wcześniej wymienione podmioty. Ich zadaniem jest konfrontacja podjętej decyzji. Dodawane są kolejne kolumny:

- Komentarz BSM,
- Komentarz K-DES.

Komentarz BSM jest to odpowiedź ze strony menadżera usługi.

Komentarz K-DES (tutaj by się przydało rozszyfrować co to K-DES z niemieckiego) natomiast jest odpowiedzią międzynarodowego zarządu firmy.

Zaaktualizowany plik powraca do Volkswagen Poznań, rozpoczynając tym samym kolejną iterację procesu.

Jak wcześniej wspomniano, proces składa się zazwyczaj z czterech iteracji. Etapem kończącym cykl jest sporządzenie wymaganych dokumentów oraz faktur.

2.2 Wykorzystane technologie

Aby usprawnić przebieg procesu, zabezpieczyć go przed błędami i usystematyzować dane, zdecydowano się na stworzenie aplikacji do jego obsługi. Głównym kryterium przy doborze technologii była powszechna dostępność do powstałego systemu wśród pracowników. Dlatego też zdecydowano się na wykorzystanie komponentów pakietu *Microsoft 365*. Pakiet ten jest bardzo rozbudowany i powszechnie wykorzystywany w firmie Volkswagen. Zawiera on programy pozwalające na stworzenie kompletnego systemu bez konieczności użycia dodatkowych serwisów.

2.2.1 Skrypty pakietu Office

Skrypty pakietu Office pozwalają na automatyzację zadań w arkuszach kalkulacyjnych Excel. Jedną z dostępnych opcji jest funkcja *Action Recorder*, która daje możliwość "nagrania" sekwencji kroków wykonanych przez użytkownika, a następnie przekształca je na skrypt wielokrotnego użytku.

Skrypty pakietu Office dysponują również wbudowanym *edytorem kodu* (ang. *Code Editor*), opartym na języku *TypeScript*, będącym odmianą *JavaScript*. Sam edytor, choć stosunkowo ograniczony, umożliwia zastosowanie konstrukcji niedostępnych w Action Recorder, takich jak instrukcje warunkowe czy pętle.

Ponadto, program Excel pozwala na zapis skryptu w skrószycie. Oznacza to, że każdy użytkownik dysponujący dostępem do pliku uzyskuje również możliwość uruchomienia kodu powiązanego ze skrószkiem, do którego jest on przypisany.

2.2.2 SharePoint

SharePoint to platforma należąca do pakietu Microsoft Microsoft 365, umożliwiająca tworzenie aplikacji webowych, takich jak witryny i strony internetowe. Jej głównym celem jest usprawnienie współpracy zespołowej poprzez dostarczanie narzędzi do publikowania informacji i raportów, które mogą być skierowane do określonych grup odbiorców.

Jednym z kluczowych zastosowań SharePointa jest zarządzanie danymi. Platforma oferuje przestrzeń do przechowywania różnego rodzaju plików, dokumentów i informacji, pełniąc funkcję serwera danych. Dzięki dostępności wbudowanych konektorów² (ang. *connectors*), umożliwia również wykorzystanie przechowywanych danych w procesie tworzenia aplikacji czy witryn.

Istotnym elementem środowiska SharePoint jest możliwość tworzenia list, często nazywanych *listami sharepointowymi*. Listy te mogą być wykorzystywane jako proste bazy danych, które umożliwiają dynamiczne aktualizowanie i synchronizowanie danych w czasie rzeczywistym.

SharePoint oferuje zaawansowane zarządzanie uprawnieniami. Administratorzy mogą precyzyjnie definiować dostęp użytkowników do poszczególnych zasobów witryny co pozwala na skuteczne zabezpieczenie wrażliwych informacji.

² *Konektor* (ang. *connector*) – moduł umożliwiający integrację aplikacji z usługami lub źródłami danych w celu wymiany informacji i synchronizacji systemów.

Platforma jest silnie zintegrowana z innymi usługami pakietu Microsoft 365, takimi jak Teams, Outlook czy OneDrive. Dzięki temu użytkownicy mogą współdzielić dane, pracować nad nimi w czasie rzeczywistym i korzystać z jednego spójnego środowiska pracy.

Ważnym aspektem SharePointa jest możliwość dostosowania wyglądu i funkcjonalności witryn do potrzeb użytkowników. Personalizacja obejmuje m.in. konfigurację interfejsu, dodawanie aplikacji webowych czy tworzenie dedykowanych formularzy.

W kontekście współczesnych modeli pracy, takich jak praca hybrydowa czy zdalna, SharePoint oferuje wsparcie dla użytkowników korzystających z różnorodnych urządzeń. Dostęp do danych jest możliwy za pośrednictwem przeglądarki internetowej oraz aplikacji mobilnych.

2.2.3 Power Automate

Power Automate to narzędzie wchodzące w skład pakietu Microsoft 365, które pozwala na automatyzację procesów biznesowych (*RPA*, ang. *Robotic Process Automation*), ograniczając potrzebę wykonywania powtarzalnych czynności. Umożliwia ono tworzenie przepływów pracy nazywanych *flow* (ang. *przepływ*), opartych na zdarzeniach, które umożliwiają spójną pracę różnych środowisk.

Flow w Power Automate jest odpowiednikiem funkcji w standardowych językach programowania. Na przykład, przepływ może automatycznie wysyłać powiadomienia e-mail po aktualizacji rekordu w SharePoint. Różnica polega na tym, że jest ono tworzone w wizualnym środowisku Low-Code i działa na zasadzie logicznego ciągu akcji wyzwalanych po sobie przez określone instrukcje.

Za pomocą flow można tworzyć własne procesy, które przy odpowiedniej implementacji, dorównują tym znanym z pełnych środowisk kodowych pod względem logiki i efektywności. Do dyspozycji są instrukcje warunkowe, pętle, zmienne, operacje na danych czy integracje z API poprzez konektory.

2.2.4 Power Apps

Power Apps to kolejne środowisko Low-Code, wchodzące w skład pakietu Microsoft 365, które jest dedykowanym rozwiązaniem do tworzenia aplikacji biznesowych. Dzięki intuicyjnemu interfejsowi graficznemu daje możliwość prostej implementacji mechanizmu działania nawet przez osoby bez zaawansowanej wiedzy programistycznej. Jest ona zintegrowana z innymi usługami pakietu Microsoft 365, takimi jak SharePoint czy Power Automate, co rozszerza możliwości stworzonych aplikacji.

Power Apps pozwala na stworzenie spersonalizowanej aplikacji, dostosowanej do motywu organizacji, a przy połączeniu z innymi serwisami daje możliwość tworzenia zaawansowanych rozwiązań, minimalizując przy tym czas potrzebny na ich zaimplementowanie.

Ekran aplikacji, komponowane za pomocą tego rozwiązania, porównywalne są z tymi, które można stworzyć w standardowych środowiskach programistycznych (jak np. JavaScript czy .NET), jednak proces ich tworzenia jest prostszy, ze względu na obecność edytora wizualnego. Umożliwia on korzystanie z gotowych komponentów w aplikacji, takich jak przyciski, pola danych wejściowych, listy, tabele, grafiki etc.

Dodawanie elementów do ekranów aplikacji odbywa się poprzez przeciąganie ich z biblioteki i upuszczanie w wybranym miejscu. Każdy komponent, może zostać skonfigurowany według potrzeb użytkownika poprzez edycje *właściwości*. Możemy określić między innymi wypełnienie czy pozycję X i Y na ekranie, ale niektóre obiekty mają też unikalne właściwości takie jak *OnSelect*³ dla przycisku.

³OnSelect – określa akcje, które zostaną wykonane po naciśnięciu elementu



RYSUNEK 2.1: Edytor Power Apps

LINK DO OBRAZKA I OPISU ELEMENTÓW: <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/power-apps-studio>

Rysunek 2.1 przedstawia edytor programu. Zawiera on następujące elementy:

1. **Pasek poleceń:** wyświetla inny zestaw poleceń w zależności od wybranego kontrolki.
2. **Akcje aplikacji:** Opcje wyświetlania właściwości, dodawania komentarzy, sprawdzania błędów, udostępniania, podglądu, zapisu lub publikowania aplikacji.
3. **Lista właściwości:** Lista właściwości wybranego obiektu.
4. **Pasek formuł:** Tworzenie lub edycja formuły dla wybranej właściwości z użyciem jednej lub więcej funkcji.
5. **Menu tworzenia aplikacji:** Panel wyboru umożliwiający przełączanie się między źródłami danych oraz wstawianie dodatkowych opcji.
6. **Lista elementów aplikacji:** Pokazuje elementy obecne na ekranie w postaci drzewa.
7. **Płótno/ekran:** Główne płótno do komponowania struktury aplikacji.
8. **Panel właściwości:** Lista właściwości wybranego obiektu.
9. **Ustawienia i wirtualny agent:** Ustawienia aplikacji lub uzyskanie pomocy od wirtualnego agenta.
10. **Selektor ekranu:** Przełączanie się między różnymi ekranami w aplikacji.
11. **Zmiana rozmiaru płótna:** Zmienianie rozmiaru wyświetlanego płótna podczas tworzenia aplikacji.

Rozdział 3

Architektura rozwiązania

Niniejszy rozdział przedstawia architekturę rozwiązania, obejmującą zarówno założenia projektowe, jak i koncepcję opracowywanego systemu. Założenia projektowe określają podstawowe wymagania oraz wytyczne, stanowiąc fundament opracowywanego rozwiązania. Natomiast koncepcja rozwiązania, uwzględniająca zasadnicze założenia, strukturę i logikę działania, stanowiące podstawę do dalszego prowadzenia prac projektowych.

3.1 Założenia projektowe

Założenia projektowe definiują podstawowe wytyczne, które określają zarówno funkcjonalność, jak i wymagania techniczne tworzonego rozwiązania. Obejmują one m.in. systematyzację danych, wykorzystanie platformy Microsoft 365 oraz optymalizację procesów decyzyjnych.

3.1.1 Systematyzacja danych

Jedną z zasadniczych funkcji omawianej aplikacji jest systematyzacja danych. Arkusze kalkulacyjne przesyłane przez oddział w Wolfsburgu nie posiadają ustandaryzowanej struktury, co negatywnie wpływa na ich czytelność oraz czas potrzebny na analizę.

Tabela 3.1 przedstawia różnice w nazwach kolumn na przestrzeni trzech lat.

TABELA 3.1: Zestawienie nagłówków kolumn w latach 2022-2024

	2022	2023	2024
Nazwy kolumn na przestrzeni lat	Service group	Service group	Service group
	Service main group	Service main group	Service main group
	Service sub group	Service sub group	Service sub group
	Business Service	Business Service	Business Service
	ID	ID	ID
	Business Service Manager	Business Service Manager	Business Service Manager
	Unit of Measurement	Unit of Measurement	Resource Unit
		Settlementtype	Settlementtype
	PL70 2022 PLAN EUR w KVA	PL71 2023 PLAN EUR w KVA	PL72 2024 PLAN EUR w KVA
	QTY	QTY	QTY
	PL71 2023 PLAN EUR w KVA	PL72 2024 PLAN EUR w KVA	PL73 2025 PLAN EUR w KVA
	QTY	QTY	QTY

Brak jednolitego formatu danych uniemożliwia również stworzenie spójnej bazy, co ogranicza możliwość ich wykorzystania w systemach automatyzacji procesów biznesowych. Dzięki wdrożeniu omawianego rozwiązania, możliwe jest ujednolicenie danych, pozwalając na ich efektywne zarządzanie i automatyczne przetwarzanie.

3.1.2 Archiwizacja danych

Utworzenie bazy danych gromadzącej informacje o wcześniejszych działaniach realizowanych w ramach projektowanego systemu stanowi istotny element zapewniający ciągłość procesów decyzyjnych. Dzięki systematycznej archiwizacji nowi użytkownicy mogą szybko zapoznać się z przebiegiem procedur i lepiej zrozumieć kontekst dotychczas podejmowanych decyzji. Dostęp do zasobów historycznych nie tylko skraca czas potrzebny na pełne wdrożenie w funkcjonowanie systemu, lecz także usprawnia przetwarzanie danych bieżących.

3.1.3 Interfejs przyjazny dla użytkownika

Dzięki dedykowanemu narzędziu z prostym i intuicyjnym interfejsem, nawigacja po bazie danych jest znacznie łatwiejsza, a problemy związane z używaniem arkuszy kalkulacyjnych zostają wyeliminowane. Przyjazny dla użytkownika interfejs oznacza:

- **Prostotę:** nieskomplikowany układ umożliwia szybkie odnalezienie potrzebnych informacji.
- **Przejrzystość:** dane są zaprezentowane w sposób czytelny, z jasno określonymi polami i etykietami.
- **Przydatne funkcje:** filtrowanie i wyszukiwanie danych wspierają efektywność pracy.

Klarowny układ i czytelność interfejsu pozwalają użytkownikowi skupić się na konkretnej usłudze, co minimalizuje ryzyko pomyłek, takich jak błędne interpretowanie danych lub wybór niewłaściwego wiersza.

Takie podejście nie tylko zwiększa efektywność pracy, ale również poprawia komfort użytkowników, dzięki czemu procesy związane z analizą i zarządzaniem danymi stają się bardziej zrozumiałe i mniej podatne na błędy.

3.1.4 Użycie pakietu Microsoft 365

Wykorzystanie platformy Power¹ w połączeniu z Sharepoint, pozwala na utworzenie w pełni funkcjonalnego rozwiązania, zachowując spójność danych dzięki integracji poszczególnych składników pakietu.

Aby korzystanie z aplikacji było możliwe, użytkownicy muszą mieć dostęp do potrzebnych usług oraz licencji. W przypadku omawianego pakietu, każdy z pracowników, ma do niego dostęp. Pozwala to na uniknięcie dodatkowych kosztów.

Wykorzystany pakiet nie jest dostępny w najbardziej rozbudowanym wariantcie, co wprowadza pewne ograniczenia, ponieważ nie zawiera oprogramowania do tworzenia i zarządzania rozbudowanymi bazami danych o złożonej strukturze (takie możliwości daje między innymi *Microsoft Azure*). Sharepoint pozwala jedynie na utworzenie prostej bazy danych opierającej się o wcześniej opisane listy.

¹Platforma Power (ang. *Power Platform*) – składowa pakietu Microsoft 365, w skład której wchodzi takie programy jak Power Apps, Power Automate czy Power BI.

3.1.5 Optymalizacja

Priorytetem implementowanego rozwiązania jest usprawnienie całego procesu decyzyjnego, co powinno przełożyć się na poprawę efektywności analizy oraz przetwarzania danych przez użytkowników, skracając czas wymagany na podjęcie decyzji.

Dzięki wprowadzeniu mechanizmów automatyzacji biurowej możliwe jest zmniejszenie liczby osób zaangażowanych w realizację procesu, co może przyczynić się do ograniczenia kosztów operacyjnych i lepszej organizacji personelu w przedsiębiorstwie.

3.2 Koncepcja rozwiązania

W niniejszym podrozdziale przedstawiona została koncepcja rozwiązania problemu automatyzacji procesu decyzyjnego. Na podstawie przeprowadzonej analizy wymagań oraz istniejących ograniczeń, zaproponowano kompleksowe podejście do realizacji systemu.

Całość rozwiązania podzielono na cztery główne etapy:

- utworzenie dedykowanej bazy danych,
- opracowanie mechanizmu importu danych z arkuszy kalkulacyjnych,
- przygotowanie formularzy do obsługi procesu,
- automatyzacja generowania raportów.

Przyjęte rozwiązanie ma na celu usprawnienie procesu przy zachowaniu jego dotychczasowej logiki biznesowej. Szczegółowy opis realizacji poszczególnych etapów został przedstawiony w kolejnych podrozdziałach.

3.2.1 Baza danych

Do przechowywania danych wykorzystano listy programu SharePoint. Pomimo tego, że nie jest to dedykowane rozwiązanie bazodanowe, wybór ten podyktowany został wymaganiami integracji z istniejącą infrastrukturą.

Struktura bazy danych

W wyniku analizy danych historycznych zidentyfikowano elementy kluczowe dla procesu indykcji. Na tej podstawie zaprojektowano strukturę składającą się z trzech powiązanych ze sobą list:

- **Lista usług** – zawierająca podstawowe, niezmiennie informacje o serwisach,
- **Lista kwot** – przechowująca dane odnośnie cen i liczbie licencji, które zmieniają się raz do roku,
- **Lista indykacji** – gromadzi informacje w obrębie jednej indykacji.

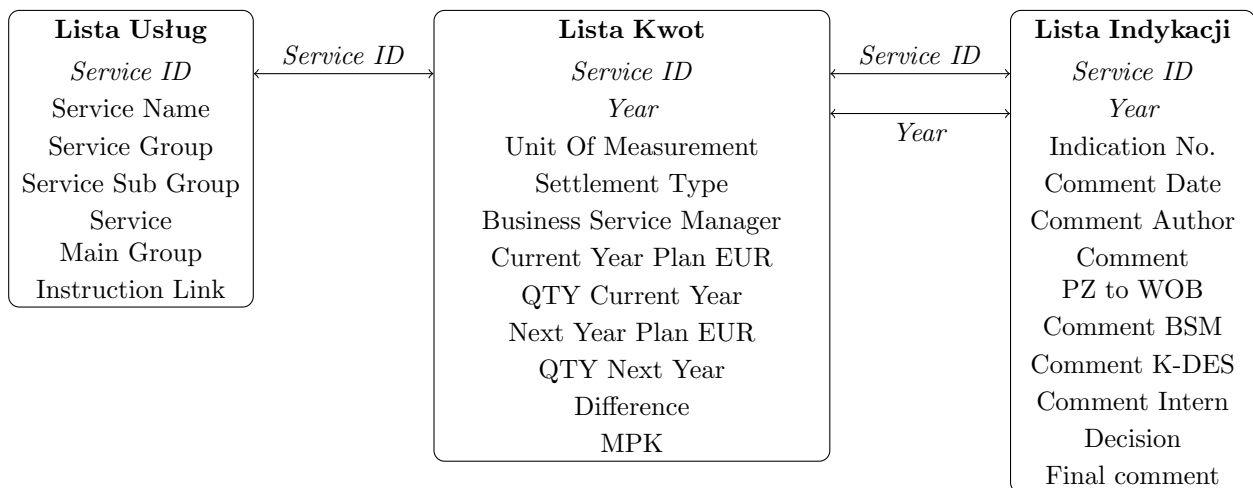
Atrybuty danych

Na podstawie analizy wymagań oraz dotychczasowego procesu, zdefiniowano następujący zestaw atrybutów, które powinna zawierać baza danych:

- Service group
- Service main group
- Service sub group
- Business Service
- Instruction link
- ID
- Business Service Manager
- Unit Of Measurement
- Settlement Type
- Current Year Plan EUR
- Quantity Current Year
- Next Year Plan EUR
- Quantity Next Year
- Year
- MPK
- Difference
- Indication Number
- Comment Intern
- Comment Date
- Comment Author
- Comment PZ to WOB
- Comment BSM
- Comment K-DES
- Decision
- Final comment

Powyższy zestaw atrybutów został opracowany na podstawie analizy danych historycznych z poprzednich lat (przedstawionych w Tabeli 3.1). Wybrane pola reprezentują najczęściej występujące informacje w procesie indykacji, uzupełnione o dodatkowe atrybuty niezbędne do efektywnego funkcjonowania procesu, takie jak pola komentarzy czy decyzji.

Model powiązań



RYSUNEK 3.1: Schemat relacji między listami.

Model danych przedstawiony na Rysunku 3.1 został zaprojektowany z uwzględnieniem następujących założeń:

- *Lista usług* pełni rolę centralnego rejestru serwisów, zawierając ich podstawową charakterystykę,
- *Lista kwot* umożliwia śledzenie zmian w wymiarze finansowym na przestrzeni lat,
- *Lista indykacji* przechowuje historię procesu decyzyjnego wraz z towarzyszącymi komentarzami i ustaleniami.

3.2.2 Dodawanie informacji do bazy danych

Po ustaleniu struktury danych wykorzystywanych przez system, kolejnym etapem jest określenie sposobu importu informacji z arkuszy kalkulacyjnych do bazy danych. Postanowiono wyko-

rzystać program Power Automate w celu automatyzacji tego procesu. Jednakże z uwagi na dużą rozbieżność danych wymaga on asysty użytkownika.

W celu dostosowania danych do struktury bazy, zaplanowano zaimplementowanie formularza walidacyjnego dla nazw kolumn. System pobiera nazwy istniejących kolumn z arkusza i umożliwia ich mapowanie z wykorzystaniem predefiniowanej listy nagłówków z list SharePoint.

Po uporządkowaniu struktury, użytkownik określa rok oraz numer indykacji dla importowanego arkusza. Następnie dane przekazywane są do *flow* w programie *Power Automate*, który przypisze je do odpowiednich list w bazie danych, jednocześnie zapobiegając duplikacji rekordów.

Interfejs jest dodatkowo wyposażony w formularz służący do przypisywania numerów *MPK* nowym serwisom. Jest to kluczowy element, ponieważ numer *MPK* determinuje obszar odpowiedzialny za obsługę danej usługi. Dla serwisów występujących w poprzednich latach, system automatycznie przypisuje istniejące numery *MPK*, redukując ilość danych do wprowadzenia. Jednocześnie zachowana będzie możliwość modyfikacji wcześniej przypisanych numerów.

3.2.3 Interfejs procesu decyzyjnego

Interfejs obsługi procesu decyzyjnego został podzielony na dwa współpracujące ze sobą ekrany. Takie rozwiązanie pozwala na zachowanie przejrzystości prezentowanych informacji przy jednoczesnym zapewnieniu dostępu do wszystkich niezbędnych funkcjonalności.

Ekran nawigacyjny

Pierwszy ekran pełni rolę panelu nawigacyjnego, prezentującym kluczowe informacje dotyczące usługi:

- Service Name – nazwa serwisu,
- Service ID – unikalny identyfikator usługi,
- MPK – numer określający miejsce powstawania kosztów,
- Decision – aktualny status decyzji.

Ponadto użytkownicy będą mieli możliwość filtrowania i wyszukiwania serwisów według następujących kryteriów:

- wyszukiwanie serwisów względem ID,
-
- wyszukiwanie serwisów względem nazwy,
- filtrowanie według przypisanych numerów MPK,
- filtrowanie według statusu decyzji (*Accepted*, *Not Accepted*, *No Status*).

Ekran szczegółowy

Po wybraniu serwisu z listy, użytkownik zostaje przekierowany do ekranu szczegółowego, który składa się z trzech głównych sekcji:

- **Podgląd danych historycznych** – prezentuje on zarówno ogólne informacje o serwisie zbierane na przestrzeni lat, jak i szczegóły dotyczące poszczególnych indykacji.

- **Formularz decyzyjny** – zestaw pól do wprowadzenia informacji o bieżącej indykacji. Składają się na niego:
 - rok (wartość domyślna: bieżący),
 - numer indykacji (wartość domyślna: kolejny wolny numer),
 - autor (wartość domyślna: zalogowany użytkownik),
 - komentarze (wewnętrzny, BSM, K-DES),
 - decyzja (wartość domyślna: poprzednia decyzja).

3.2.4 Generowanie raportu

Niby spoko, ale jakoś do pizdy się niektóre zdania czyta. Trzeba to jakoś poprawić. Jakoś tak zbyt ogólnie Ostatnim etapem cyklu obsługi procesu jest generowanie raportu, który będzie gotowy do odesłania w celu kontynuacji konsultacji z zakładem w Wolfsburgu. W ramach tego etapu zaplanowano rozwiązanie wykorzystujące dane przechowywane na listach sharepointowych, co umożliwi wydajne zarządzanie informacjami niezbędnymi do stworzenia raportu.

W dedykowanym oknie aplikacji użytkownik ma możliwość wyboru odpowiedniego roku oraz etapu (numer indykacji). Na podstawie tych informacji, system przetworzy podane kryteria, aby zgromadzić odpowiednie dane z różnych źródeł.

Kolekcja² danych, utworzona na podstawie wybranych kryteriów, łączy dane z trzech list sharepointowych. Dzięki temu możliwe jest skonsolidowanie danych w jedną, spójną strukturę, która zawierać będzie wszystkie niezbędne informacje do sporządzenia raportu. **TO TRZEBA ROZWINĄĆ W IMPLEMENTACJI BO MOGĄ SIĘ POJAWIĆ JAKIE MECHANIZMY (JA NA PRZYKŁAD NIE WIEM XD):** Mechanizmy wyszukiwania umożliwią powiązanie identyfikatorów usług z odpowiadającymi im rekordami, co zapewni dokładność i kompletność zgromadzonych danych.

Dodatkowo, w tym samym oknie aplikacji użytkownik ma dostęp do podglądu zgromadzonych danych w formie tabeli. Umożliwi to weryfikację poprawności i kompletności informacji przed finalnym wygenerowaniem raportu. Po zatwierdzeniu danych, system przekaże zgromadzoną kolekcję do Power Automate, gdzie zostaną one zmodyfikowane, w celu do przygotowania raportu w odpowiednim formacie (tj. w formacie arkusza kalkulacyjnego *Excel*) do odesłania.

²Kolekcja (Power Apps) – tymczasowy zbiór danych, przechowywanych lokalnie w aplikacji, umożliwiający zarządzanie rekordami podczas jej działania.

Rozdział 4

Implementacja

W niniejszym rozdziale przedstawiono szczegóły techniczne wdrożonego rozwiązania oraz przybliżono aspekty implementacyjne systemu, obejmujące wykorzystanie platformy Microsoft Power Platform – w szczególności Power Apps – do budowy interfejsu użytkownika oraz Power Automate do automatyzacji procesów biznesowych. Ponadto, omówiono integrację z platformą SharePoint oraz implementację skryptów usprawniających pracę z pakietem Microsoft Office.

W ramach analizy technicznej szczegółowo opisano wszystkie komponenty systemu oraz sposób ich integracji. Szczególną uwagę poświęcono mechanizmom przepływu danych, automatyzacji procesów oraz implementacji logiki biznesowej w środowisku Low-Code.

4.1 Utworzenie bazy danych na platformie Sharepoint

Zgodnie z koncepcją, baza danych utworzona została w środowisku SharePoint. Pierwszym krokiem jest stworzenie strony dedykowanej temu procesowi. W ekranie startowym programu należy wybrać opcję *Utwórz witrynę*. Następnie należy wybrać typ witryny *Witryna zespołu* oraz szablon określający jej wygląd. Na koniec należy określić nazwę tworzonej strony.

Kolejnym krokiem jest utworzenie struktury list oraz plików. Na potrzeby procesu utworzono dwa foldery – *TempFiles* przechowujący arkusze przed zapisaniem ich danych na listach oraz *ArchivedFiles* przechowujący pliki po zakończeniu procesu. Ponadto utworzono trzy listy:

- *Lista_Uslug*,
- *Lista_Kwot*,
- *Lista_Indykacji*.

Ich struktura jest taka sama jak opisana w podsekcji 3.2.1. W celu dodania tych elementów należy wejść w utworzoną witrynę, następnie wybrać *Zawartość witryny* z bocznego paska nawigacji. Dalej, należy wcisnąć przycisk *Nowy* i wybrać typ elementu – *Lista* oraz *Biblioteka dokumentów*. Na końcu trzeba podać nazwę oraz opcjonalnie opis elementu. Po wykonaniu tych kroków, folder jest gotowy do użycia. Jednakże w przypadku list należy jeszcze zdefiniować kolumny.

W tym celu należy wejść w utworzoną listę i wybrać przycisk *Dodaj kolumnę* znajdujący się po prawej stronie domyślnie utworzonych kolumn. Dalej pojawi się okno z możliwością wyboru typu kolumny oraz jej nazwy. Po zatwierdzeniu, kolumna zostanie dodana do listy. Na koniec należy skonfigurować niektóre kolumny. Aby to zrobić należy wybrać ustawienia strony i nacisnąć pozycję *Ustawienia listy*. Tam wybieramy nazwę kolumny, którą chcemy skonfigurować.

Tabela 4.1 przedstawia konfigurację kolumn listy:

TABELA 4.1: Konfiguracja listy na platformie Sharepoint

Nazwa kolumny	Typ
Service.ID	Liczba
Service.Name	Pojedynczy wiersz tekstu
Service.Group	Pojedynczy wiersz tekstu
Service.Sub.Group	Pojedynczy wiersz tekstu
Service.Main.Group	Pojedynczy wiersz tekstu
Instruction.Link	Pojedynczy wiersz tekstu
Unit.Of.Measurement	Pojedynczy wiersz tekstu
Settlement.Type	Pojedynczy wiersz tekstu
Business.Service.Manager	Pojedynczy wiersz tekstu
Current.Year.Plan.EUR	Liczba
QTY.Current.Year	Liczba
Next.Year.Plan.EUR	Liczba
QTY.Next.Year	Liczba
Difference	Obliczeniowa
MPK	Pojedynczy wiersz tekstu
Year	Liczba
IndicationNo	Liczba
Comment.Date	Pojedynczy wiersz tekstu
Comment.Author	Wiele wierszy tekstu
Comment.PZ.to.WOB	Wiele wierszy tekstu
Comment.BSM	Wiele wierszy tekstu
Comment.K-DES	Wiele wierszy tekstu
Comment.Intern	Wiele wierszy tekstu
Decision	Liczba
Final.comment	Wiele wierszy tekstu

Kolumna *Difference* jest kolumną obliczeniową co oznacza, że jej wartość jest obliczana względem podanej formuły. W tym przypadku oblicza ona różnicę cen między rokiem następnym a bieżącym. Kolumna *MPK* pomimo wartości liczbowych, jest typu tekstowego gdyż znacznie upraszcza to filtrowanie oraz przepisywanie istniejących numerów do nowych danych.

Kolumna *Decision* jest kolumną liczbową ponieważ status decyzji konwertowany jest na kod liczbowy aby przyspieszyć proces filtrowania:

- *Accepted* \rightarrow 1,
- *Not Accepted* \rightarrow -1,
- *No Status* \rightarrow 0.

Początkowo planowano wykorzystać wbudowany w SharePoint mechanizm *lookup* do implementacji relacji między listami. Jednak ze względu na ograniczenie tego mechanizmu do relacji wyłącznie między dwiema listami, zdecydowano się na realizację powiązań na poziomie logiki aplikacji. Szczegóły tej implementacji zostały opisane w rozdziale dotyczącym realizacji systemu.

4.2 Ekran dodawania danych

Głównym wyzwaniem okazał się brak systematycznej organizacji danych w arkuszach programu Excel, co skutkowało niekompatybilnością z zaprojektowaną bazą danych. W celu rozwiązania tego problemu, opracowano dedykowany interfejs w aplikacji, który wspomaga użytkownika w procesie przetwarzania danych, minimalizując ryzyko wystąpienia błędów.

Rysunek 4.1 przedstawia finalny wygląd ekranu dodawania danych do systemu. Komponenty znajdujące się na ekranie zostały opisane poniżej.

The screenshot displays a web application interface for data entry. At the top, there's a navigation bar with a logo and a home icon. Below it, the interface is divided into several sections:

- Add attachments to process:** A section with a text input and a 'Dotłącz plik' button.
- Choose file to preview:** A section showing a selected file '1. Indikation June 2022_costs for 10.12.2024 17:05' with a close button.
- Click to open:** A button labeled 'Click to open: 1. Indikation June 2022_costs for'.
- Current incorrect column names:** A list of column names that need to be corrected.
- Choose correct name for each column:** A dropdown menu for selecting the correct name for each column.
- Column Matching Instructions:** A box with instructions on how to select the pre-uploaded Excel file and a 'Save' button.
- Update column names:** A button to update the column names.
- Choose year:** A dropdown menu for selecting the year (2024).
- Choose indication No.:** A dropdown menu for selecting the indication number (1).
- Upload data:** A button to upload the data.
- Assign/Edit MPK number:** A section with a filter by Name / ID / MPK and a 'Show filled fields' toggle.
- Table:** A table with columns 'Service Name', 'Service ID', and 'MPK'. The table contains 15 rows of data.
- Remaining MPK to fill: 59:** A box indicating the remaining MPK to fill and that data is saved automatically.

Service Name	Service ID	MPK
li	2	
C	1	
F	0	
F	3	
F	9	
F	4	
F	0	
F		
S	5	
F	5	
C	9	
C	3	
V	9	
C	1	
C		
li	2	
C	1	

RYSUNEK 4.1: Ekran dodawania danych do systemu

4.2.1 Zapis pliku w chmurze

Pierwszym etapem procesu jest tymczasowy zapis pliku Excel w chmurze, co umożliwia jego udostępnienie innym systemom. Do realizacji tego zadania wykorzystano kontrolkę¹ *Attachment*

¹Kontrolka – element służący do nawigacji, wyświetlania danych i obsługi aplikacji.

Control. Pozwala ona na zapisanie pliku w pamięci aplikacji. Odbywa się to przez naciśnięcie przycisku "Dołącz plik" lub przy użyciu mechaniki *przeciągnij i upuść* (ang. *Drag And Drop*).

Aby dalej przekazać plik oraz jego zawartość należy nacisnąć przycisk opisany jako *Save attachments* znajdujący się pod wcześniej omawianym elementem. Naciśnięcie go skutkuje wywołaniem szeregu funkcji opisanych we właściwości *OnSelect*. W pierwszej kolejności sprawdzane jest, czy plik został załadowany. Jeśli tak, to wywoływany jest przepływ *SaveFileAndRunScript*. Wynik przepływu jest zapisywany w zmiennej tablicowej, która w Power Apps określana jest jako *kolekcja*, o nazwie *FlowOutput*. Po wykonaniu się przepływu, pliki zapisane w folderze SharePoint, zostają usunięte z pamięci aplikacji.

Przepływ *SaveFileAndRunScript*

Rysunek 4.2, przedstawia edytor programu Power Automate. Widoczny w nim przepływ nazywany *SaveFileAndRunScript* jest odpowiedzialny za zapisanie pliku w chmurze oraz wstępne przetworzenie. W momencie wywołania przepływu, plik jest przekazany jako parametr wejściowy. Przepływ ten składa się z kilku kroków, które zostaną omówione w kolejności ich wykonywania.

1. Funkcja: Power Apps (V2)

Przepływ rozpoczyna się od funkcji wywoływanej bezpośrednio z aplikacji Power Apps. Jako parametry wejściowe przyjmuje:

- nazwę pliku (*File Name*),
- zawartość pliku (*File Content*) w formacie binarnym.

2. Zainicjalizowanie zmiennej

Element *Initialize variable* tworzy zmienną o nazwie *FileExists*, która przechowuje informację, czy plik o podanej nazwie znajduje się już na SharePoint.

3. Sprawdzenie istniejących plików

Blok *Get files* pobiera listę wszystkich plików z wybranego folderu SharePoint wraz z ich metadanymi, takimi jak nazwa, ścieżka czy data modyfikacji. Wynik zostaje zapisany w zmiennej *FileExists*, która przyjmuje wartość *true*, jeśli plik został znaleziony, lub *false*, jeśli plik nie istnieje.

4. Instrukcja warunkowa *If*

Element *Condition* sprawdza wartość zmiennej *FileExists*. W zależności od wyniku:

- jeśli zmienna ma wartość *true* – przepływ kończy działanie,
- jeśli zmienna ma wartość *false* – przepływ kontynuuje proces zapisu.

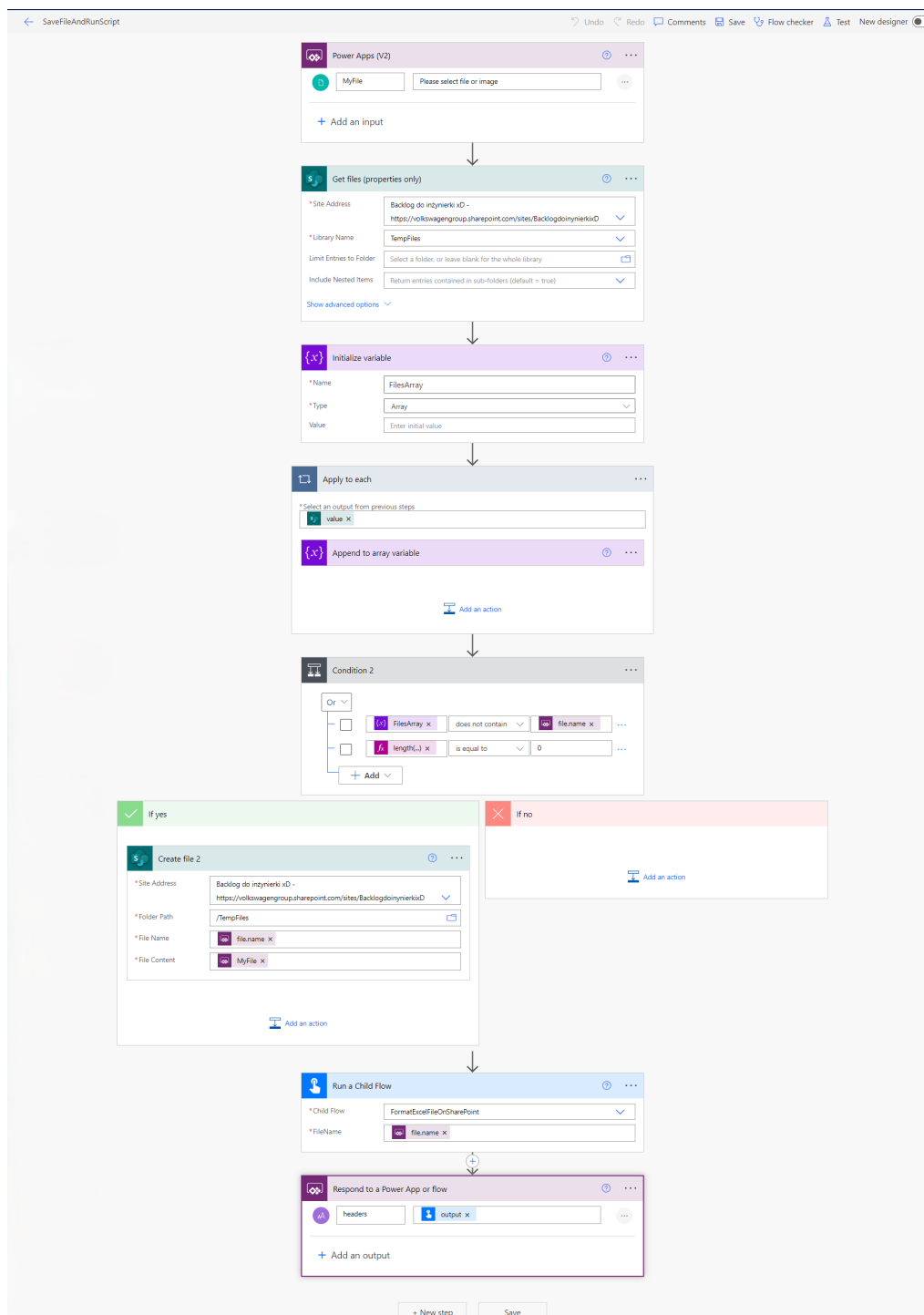
5. Utworzenie pliku

Blok *Create file* tworzy nowy plik w SharePoint, wykorzystując parametry:

- adres witryny SharePoint,
- ścieżkę do folderu docelowego,
- nazwę pliku,
- zawartość pliku.

6. Uruchomienie przepływu podrzędnego

Po pomyślnym zapisaniu pliku przepływ wywołuje tzw. *child flow*, który inicjuje działanie



RYSUNEK 4.2: Widok przepływu SaveFileAndRunScript

skryptu Office. Skrypt ten odpowiada za przetworzenie pliku w sposób zgodny z założeniami aplikacji. Jego wynik w formacie JSON jest zwracany do przepływu nadrzędnego.

7. Odpowiedź do aplikacji

Blok *Respond to Power Apps* kończy przepływ, zwracając do aplikacji dane w formacie JSON, przetworzone przez wspomniany skrypt.

4.2.2 Skrypt pakietu Office

Po utworzeniu pliku w SharePoint, w ramach przepływu następuje jego przetworzenie przez skrypt. Jest to niezbędne, jeśli chodzi o działanie procesu. Domyślnie otrzymane dane w pliku Excel są niewidoczne dla większości systemów, mogą one odczytać jedynie informacje zorganizowane w *tabele programu Excel*². Dlatego też powstał skrypt, który działa bezpośrednio w arkuszu. Jego zadaniem jest automatyczne utworzenie tabeli oraz dostosowanie jej do wymagań systemu. Poniżej przedstawiono kroki działania skryptu:

1. Wybór arkusza roboczego

Skrypt identyfikuje arkusz zawierający dane, analizuje zakres używanych komórek i usuwa ochronę hasłem, jeśli jest aktywna – krok ten jest wymagany, aby wprowadzanie zmian w arkuszu było możliwe.

2. Analiza danych

Skrypt rozpoczyna analizę od wyszukiwania początku tabeli w arkuszu. Następnie:

- usuwa puste kolumny, które nie zawierają żadnych danych,
- tworzy tabelę o dynamicznym rozmiarze, uwzględniając zakres danych znajdujących się w arkuszu,
- uzupełnia brakujące komórki w kluczowych kolumnach, korzystając z danych w poprzednich wierszach.

Takie podejście pozwala na uporządkowanie danych i przygotowanie ich do dalszego przetwarzania.

3. Dopasowanie nazw kolumn

Skrypt porównuje istniejące nazwy kolumn z listą standardowych nagłówków, korzystając z algorytmu *Jaro-Winkler*. Algorytm ten:

- analizuje podobieństwo tekstów, porównując wspólne znaki oraz ich kolejność,
- przyznaje dodatkowe punkty za zgodność początkowych znaków (prefiksu),
- zwraca wynik jako wartość z przedziału od 0 do 1, gdzie wartości bliższe 1 oznaczają większe podobieństwo.

Wynik tego procesu jest wykorzystywany w dalszych etapach aplikacji, m.in. do walidacji struktury danych. Jeśli podobieństwo jest mniejsze niż 90%, skrypt sugeruje ręczne dopasowanie nazwy kolumny.

4. Zwrócenie wyników

Skrypt generuje JSON zawierający mapowanie oryginalnych nazw kolumn z najlepszymi dopasowaniami z listy standardowych nagłówków.

Wprowadzenie przepływu podrzędnego było konieczne z uwagi na sposób, w jaki Power Automate obsługuje operacje na plikach w SharePoint. Gdy plik zostaje zapisany w folderze SharePoint, system przypisuje mu status wskazujący, czy jest gotowy do przetworzenia. W przypadku realizacji obu operacji (zapisu i przetwarzania pliku) w ramach jednego przepływu pojawiał się problem. Wynikał on z faktu, że przepływ pobierał dane z SharePoint już na etapie wstępnego sprawdzenia,

²Tabela w programie Excel wymaga osobnej deklaracji poprzez zaznaczenie zakresu komórek i wybór opcji *Narzędzia główne → Formatuj jako tabelę*

czy plik o określonej nazwie istnieje. Informacja ta była przechowywana w pamięci przepływu i nie była aktualizowana w trakcie jego dalszego wykonywania.

W efekcie, po utworzeniu nowego pliku, przepływ nie miał możliwości odświeżenia informacji o jego istnieniu i statusie. To powodowało błąd uniemożliwiający uruchomienie skryptu, gdyż system informował, że plik, dla którego miał być wykonany, nie istnieje.

Rozwiązaniem tego problemu było wyodrębnienie etapu przetwarzania pliku do osobnego przepływu podrzędnego. Przepływ podrzędny, uruchamiany po zakończeniu procesu zapisu pliku, działał niezależnie i pobierał aktualne dane z SharePoint w momencie swojego wywołania. Dzięki temu możliwe było wyeliminowanie problemu braku odświeżonych informacji o statusie pliku, co pozwoliło na poprawne uruchomienie skryptu Office Script.

LINK DO TEGO JARO_WINKLERA: <https://crucialbits.com/blog/a-comprehensive-list-of-similarity-search-algorithms/>

RYСУNEK 4.3: Formularz zapisu pliku w chmurze

Rysunek 4.3 ilustruje opisane wcześniej elementy ekranu, na którym użytkownik może dodawać załączniki do procesu, podpisane jako *"Add attachments to process"*. Obok znajduje się lista zapisanych plików, umożliwiającą wybór pliku do dalszego przetwarzania. Poniżej umieszczono przycisk *"Click to open:..."*, który pozwala na otwarcie wybranego pliku w nowym oknie przeglądarki, co ułatwia jego weryfikację i podgląd.

4.2.3 Walidacja nazw kolumn

Kolejnym etapem przed zapisaniem danych do bazy jest walidacja nazw kolumn. W tym celu zaimplementowano formularz, którego układ przedstawiono na rysunku 4.4. Formularz zawiera galerię – element umożliwiający wyświetlanie wielu rekordów danych o różnych typach. Pola wyświetlające dane w galerii mogą być dostosowywane w dowolny sposób w zależności od potrzeb użytkownika.

Galeria składa się z dwóch kolumn:

- Lewa kolumna prezentuje obecne nazwy kolumn, które są wyświetlane za pomocą kontrolki *Label*³.
- Prawa kolumna zawiera kontrolkę *ComboBox*⁴, która umożliwia wybór nazwy ze standardowej listy nagłówków. Wartości domyślne widoczne w kontrolkach *ComboBox* są generowane przez wcześniej opisany skrypt w taki sposób, aby do oryginalnej nazwy kolumny dopasowana została najbardziej podobna nazwa z predefiniowanej listy nagłówków. Ma to na celu minimalizację danych wprowadzonych przez użytkownika.

³*Label* – kontrolka tekstowa umożliwiająca wyświetlanie statycznych wartości.

⁴*ComboBox* – rozwijana lista z możliwością wprowadzania tekstu

Current incorrect column names:	Choose correct name for each column:
Service group	Service group
Service main group	Service main group
Service sub group	Service sub group
Business service	Business service
Service ID	Service ID
Business Service Manager	Business Service Manager
Unit of Measurement	Unit of Measurement
EUR w KVA	Select correct column name
QTY	Select correct column name
EUR w KVA1	Select correct column name
QTY1	Select correct column name
MPK	MPK
Department	Department

Column Matching Instructions

Select the pre-uploaded Excel file from the top list.

If you don't make changes just press Save button.

Update column names

Choose year: 2024 Choose indication No.: 1

Upload data

RYSUNEK 4.4: Formularz walidacji nazw kolumn

Po prawej stronie formularza znajduje się instrukcja użytkownika, zawierająca wskazówki dotyczące prawidłowego uzupełniania nazw kolumn. Poniżej instrukcji umieszczono przycisk *Update column names*, który umożliwia wprowadzenie zmian w strukturze danych.

Działanie tego mechanizmu opiera się na zastosowaniu skryptu pakietu Office, wywoływanego przy użyciu kolejnego przepływu. Skrypt jako parametr wejściowy przyjmuje zmienną tablicową w formacie JSON, zawierającą mapowanie oryginalnych nazw kolumn z poprawionymi wartościami wybranymi przez użytkownika. Następnie skrypt iteruje po wierszu zawierającym nagłówki kolumn i dokonuje ich zamiany zgodnie z mapowaniem. Po zakończeniu działania skrypt zwraca nową strukturę nazw kolumn.

Rysunek 4.4 prezentuje wszystkie elementy formularza, w tym kontrolki umożliwiające wybór roku i numeru indykacji, które zostały umieszczone pod przyciskiem *Update column names*. Kontrolki te, wraz z przyciskiem *Upload data*, są kluczowe dla kolejnego etapu przetwarzania danych, obejmującego ich integrację z listami SharePoint.

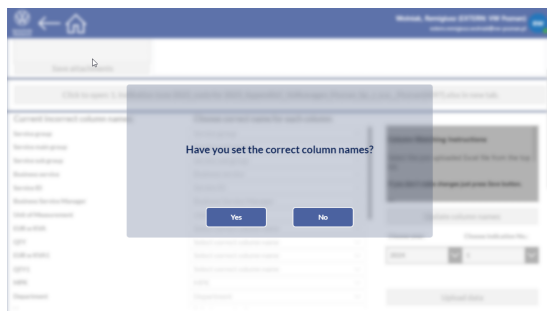
4.2.4 Integracja z listami SharePoint

Po zakończeniu walidacji nazw kolumn, kolejnym etapem jest zapisanie przetworzonych danych w utworzonej strukturze SharePoint. Proces ten rozpoczyna się od wyboru roku i numeru indykacji przy użyciu dedykowanych kontrolki *Dropdown*⁵. Wybrane wartości są następnie wykorzystywane podczas importu danych do odpowiednich list, co odbywa się za pomocą przycisku *Upload data*.

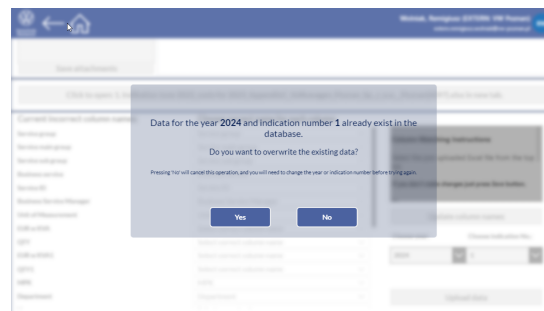
Skutki kliknięcia przycisku mogą się różnić w zależności od wybranych wartości i tego czy nazwy kolumny zostały zmienione. Rysunki 4.5 oraz 4.6 przedstawiają dwa możliwe scenariusze, które mogą wystąpić po naciśnięciu przycisku. Pierwszy z nich występuje kiedy uprzednio nie został wciśnięty przycisk *Update column names*. System upewnia się że użytkownik nie wgra przypadkowo danych z niepoprawnymi nagłówkami. Drugi natomiast pojawia się w przypadku, gdy wybrane przez użytkownika rok oraz numer indykacji, istnieją w bazie danych. System pyta czy użytkownik chce nadpisać dane, które już tam się znajdują czy anulować operację. W momencie kiedy nazwy kolumn nie zostaną zmienione oraz dane z wybranym rokiem i numerem indykacji istnieją w bazie danych, pojawiają się oba okna z informacjami.

Kiedy użytkownik upewni się, że wszystkie dane są poprawne i zatwierdzi operację, system przystępuje do importu danych. W tym celu wywołuje kolejny przepływ w programie Power Auto-

⁵ *Dropdown* – kontrolka umożliwiająca wybór jednej z dostępnych wartości z rozwijanej listy, bez możliwości edycji.



RYSUNEK 4.5: Zapytanie o poprawność nazw kolumn



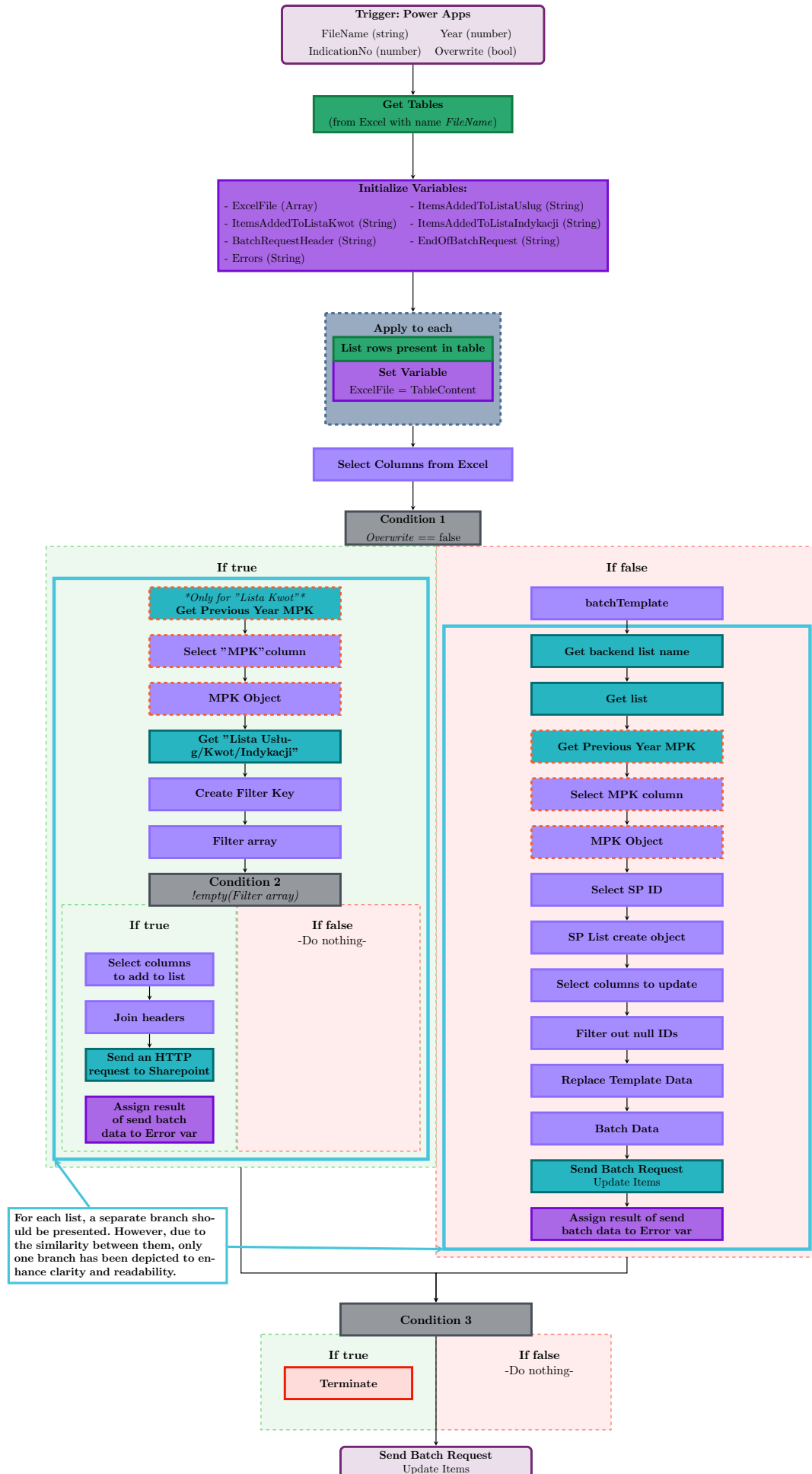
RYSUNEK 4.6: Zapytanie o nadpisanie danych

mate, który przypisuje informacji do odpowiednich list w bazie danych upewniając się jednocześnie, że nie zostaną dodane duplikaty rekordów.

Przepływ ten jest bardzo rozbudowany, dlatego zamiast widoku edytora Power Automate, na rysunku 4.7 pokazany został schemat blokowy, który reprezentuje kolejność wykonywania poszczególnych kroków. Został on uproszczony, ponieważ bloki umieszczone w błękitnych ramkach, powinny być powielone do trzech równoległych gałęzi. Wynika to z faktu, że dla wszystkich list procedura jest identyczna, a różnicą są m.in. nazwy list użyte w nagłówkach, ciałach o raz stopkach rządów HTTP. Ponadto elementy, które mają przerywaną, pomarańczową ramkę to elementy, które znajdują się w głęzi dotyczącej listy kwot. Odpowiadają one za pobieranie i przepisywanie numeru MPK z roku wcześniej.

Poniżej wyjaśniono działanie poszczególnych bloków znajdujących się na schemacie. Elementy oznaczone symbolem (*) odnoszą się do bloków znajdujących się w pomarańczowej, przerywanej ramce.

1. Blok **Trigger: Power Apps** jest wyzwalaczem działania przepływu. Uruchamia się kiedy użytkownik wcisnie przycisk w aplikacji. Jako parametry wejściowe przyjmuje nazwę pliku, rok i numer indykacji jakie mają być przypisane do danych oraz informacje czy nadpisać istniejące rekordy czy też nie.
2. Blok **Get Tables** pobiera nazwy wszystkich tabel z pliku o nazwie przekazanej do wyzwalacza (każdy plik powinien zawierać jedną tabelę, ale w Power Automate trzeba pobrać wszystkie możliwe).
3. Funkcja **Initialize variable** tworzy następujące zmienne:
 - *ExcelFile* – zmienna tablicowa, przechowująca dane z arkusza,
 - *ItemsAddedToListUslug/Kwot/Indykacji* – te zmienne przechowują *ciało zapytania HTTP*, które jest konstruowane w trakcie działania przepływu.
 - *BatchRequestHeader* oraz *EndOfBatchRequest* – przechowują one stałe nagłówki i stopkę zapytania HTTP, które są wspólne dla wszystkich zapytań.
4. Pętla **Apply to each**, dodana automatycznie przez Power Automate, iteruje po nazwach tabel pobranych z pliku a następnie dla każdej z nich przepisuje dane do zmiennej *ExcelFile*.
5. Funkcja **Select Columns from Excel**, pozwala na kształtowanie danych. Jako wejście przyjmuje ona *ExcelFile* a następnie mapuje wartości tej zmiennej do wybranych kluczy. Dzięki temu można odwołać się do dowolnej kolumny danych.



RYSUNEK 4.7: Schemat blokowy procesu importu danych z arkusza kalkulacyjnego do bazy danych

6. Blok **Condition 1** sprawdza czy wartość *Overwrite* jest równa *false*.

Jeśli tak to wykonywane są instrukcje wewnątrz bloku *If true*. Gałąź ta odpowiada za dopisanie nowych danych do bazy. W tym celu wykonane są następujące instrukcje:

- * **Get Previous Year MPK** – Pobranie elementów z listy kwot dla roku niższej niż przekazany w parametrze wejściowym,
- * **Select "MPK"column** jako wejście przyjmuje odpowiedź z bloku wyżej, ale tutaj zamiast przypisywania wartości do klucza, zawiera następujące wyrażenie:

```
concat("",item()?['Service.ID'],'":',item())
```

item() odwołuje się do pojedynczego elementu danych wejściowych. Zatem to wyrażenie tworzy strukturę obiektów, gdzie nazwą obiektu jest *Service_ID*, natomiast jako właściwości obiektu przypisane są dane z arkusza odpowiadające tej usłudze.

- * **MPK Object** przekształca strukturę utworzoną w poprzednim kroku na listę obiektów JSON.
- **Get "Lista Usług/Kwot/Indykacji"** – pobiera pełną listę rekordów z odpowiedniej listy w bazie danych.
- **Create Filter Key** tworzy klucz filtrujący. Dla listy usług nie jest on wymagany. Dla listy kwot kluczem jest rok przekazany w parametrze wejściowym. Natomiast dla listy indykacji jest to rok oraz numer indykacji połączone w jeden ciąg znaków.
- **Filter array** blok ten wykorzystany jest do porównania elementów dla danego roku i indykacji na liście SharePoint z elementami w arkuszu. Ma on za zadanie zwrócić tablicę z elementami unikalnymi dla arkusza.
- **Condition 2** sprawdza czy tablica zwrócona przez *Filter array* nie jest pusta. Jeśli nie zawiera ona unikatowych elementów to przepływ kończy działanie. Jeśli natomiast tablica zawiera unikatowe elementy to przepływ przechodzi do kolejnego kroku w gałęzi *If true*.
- **Select columns to add to list** – mapuje informacje z arkusza do kluczy odpowiadających strukturze każdej z list w bazie danych.
- **Join headers** – konwertuje tablicę powstałą w poprzednim kroku na ciąg znaków, będący ciałem żądania HTTP. Instrukcja ta zmienia separator między wierszami tabeli ze średnika na nagłówek, który musi znajdować się między każdym wysłanym zestawem danych. Dla każdej z list jest on inny.
- **Send an HTTP request to SharePoint** – wysyła kompletne żądanie HTTP do odpowiedniej listy w bazie danych. Wysyłane żądanie zawiera:
 - Nagłówek otwierający żądanie – *BatchRequestHeader*,
 - Ciało żądania powstałe w kroku wcześniej – wynik *Join headers*,
 - Stopkę żądania – *EndOfBatchRequest*.
- **Assign result of send batch data to Error var** – jak nazwa bloku wskazuje, przypisuje odpowiedź serwera na wysłane żądanie w celu późniejszej analizy.

Kiedy jednak wartość zmiennej *Overwrite* wynosi *true*, oznacza to, że istniejące rekordy mają zostać zaaktualizowane. Przepływ przechodzi do gałęzi *If false* i wykonuje następujące kroki:

- **batchTemplate** – tworzy wspólny szablon żądania HTTP.

- **Get backend list name** pobiera ona wewnętrzną nazwę listy SharePoint. Jest to konieczne, ponieważ w rządaniu HTTP należy ostrożnie używać znaków specjalnych.
 - **Get list** odczytuje dane z każdej z list.
- * Funkcje **Get Previous Year MPK**, **Select "MPK"column** oraz **MPK Object** wykonują te same zadania co w wcześniej omawianym scenariuszu.
- Kroki **Select SP ID** i **SP List create object** działają podobnie jak mechanizm pobierania numerów MPK z roku poprzedniego z tym, że mapują one identyfikatory wewnętrzne elementów listy SharePoint. Jest to konieczne ponieważ aby zaktualizować rekord, należy odwołać się do niego względem tego właśnie identyfikatora a nie np. nazwy lub *Service.ID*.
 - **Select columns to update** – przydziela informacje z odpowiednich kolumn do odpowiednich list.
 - **Filter out null IDs** – odfiltrowuje elementy, które nie mają przypisanego identyfikatora SharePoint. Gdyby nie ten krok, próba aktualizacji rekordu bez identyfikatora zakończyłaby się błędem.
 - **Replace Template Data** – wstawia wybrane informacje do szablonu rządania HTTP.
 - **batchData** – w kroku tym, znaki specjalne są zakodowane procentowo⁶ (znane również jako *kodowanie URL*). Jest to wymagane aby uniknąć błędów.
 - **Send Batch Request** – wysyła rządanie aktualizacji danych do SharePoint.
 - **Assign result of send batch data to Error var** – przypisuje odpowiedź serwera na wysłane rządanie w celu późniejszej analizy.
7. **Condition 3** sprawdza czy zmienna *Errors* zawiera jakiekolwiek kody błędów. Jeśli tak to przepływ zostaje przerwany. W przeciwnym wypadku zwracana jest informacja do aplikacji, że zapis danych zakończył się powodzeniem.

4.2.5 Uzupełnianie numerów MPK

Ostatnią funkcją tego ekranu jest możliwość uzupełniania lub edycji numerów MPK. W tym celu ponownie wykorzystano galerię, która tym razem składa się z trzech kolumn. Dwie pierwsze kolumny zawierają pola tekstowe (*Label*), które prezentują nazwę usługi oraz odpowiadający jej identyfikator. Ostatnia kolumna zawiera pole danych wejściowych (*TextInput*), do którego użytkownik wprowadza odpowiedni numer MPK. Nad galerią znajduje się dodatkowe pole, w którym można wpisać kryteria filtrowania, takie jak nazwa, identyfikator bądź numer MPK. Obok znajduje się przełącznik (*Toggle*), który umożliwia wyświetlenie usług z przypisanym już numerem MPK. Istniejące numery MPK wyświetlają się jako domyślny tekst kontrolki *TextInput* i mogą być edytowane. Ostatnim elementem ekranu jest pole tekstowe informujące użytkownika o liczbie usług bez przypisanego miejsca powstawania kosztów oraz o tym, że dane są zapisywane automatycznie.

⁶*Kodowanie procentowe* – metoda reprezentowania znaków specjalnych w adresach URL w formie zgodnej z protokołem HTTP. Polega na zastępowaniu niebezpiecznych lub niedozwolonych znaków ich odpowiednikami w postaci procentowego kodu, który składa się z symbolu "%" i dwóch cyfr szesnastkowych reprezentujących kod ASCII danego znaku.

Assign/Edit MPK number: Filter by Name / ID / MPK ☐ Show filled fields

Service Name:	Service ID:	MPK:
IA	12	
Of	11	
Pl	10	
Pl	13	
Pl	19	
Fu	14	
pr	10	
Pl	1	
Sn	15	
Pl	15	
Gr	19	
Of	13	
W	19	
Cl	1	
Gr	1	
IA	12	
Of	11	

This step is required when some data does not have a 'Cost center number' assigned to it.

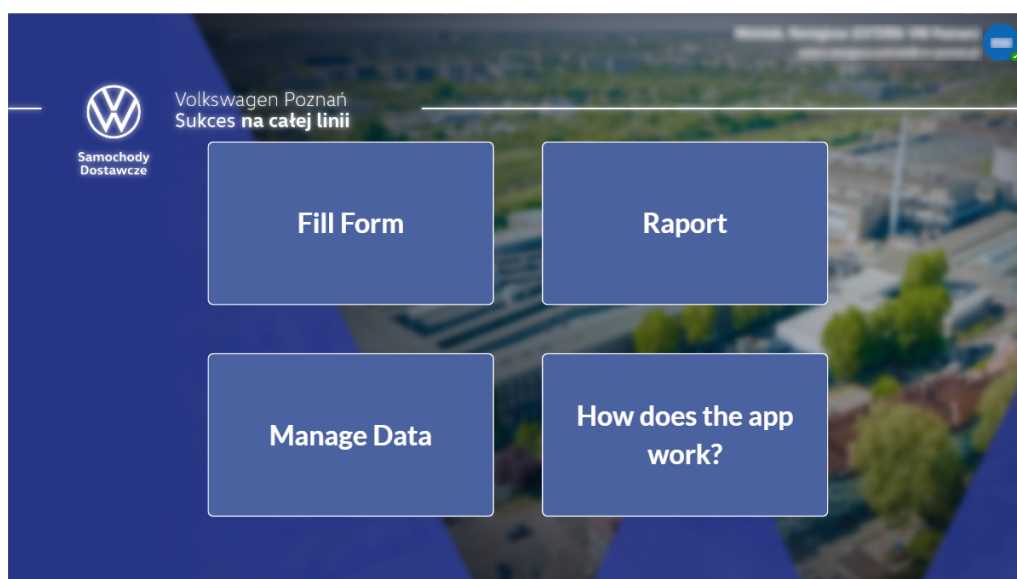
Remaining MPK to fill: **59**.

Data is saved automatically.

RYSUNEK 4.8: Formularz wypełniania/edycji numerów MPK

4.3 Ekran startowy aplikacji i przygotowanie danych

Kiedy dane zostały dostosowane do działania systemu, można przystąpić do implementacji pozostałych części rozwiązania. Kolejnym elementem jest ekran startowy aplikacji, widoczny na rysunku 4.9. Zawiera on przyciski, które przekierowują użytkownika do odpowiednich sekcji aplikacji.



RYSUNEK 4.9: Ekran startowy aplikacji

Dodatkowo, podczas uruchomienia aplikacji, pobierane są dane z list SharePointowych a następnie odpowiednio przetwarzane w celu płynnego wyświetlania ich w aplikacji. Kod wywołany podczas uruchamiania aplikacji został przedstawiony na listingu 4.1.

```

1
2 Set(varDownloadingData; true );;
3 ClearCollect(colYears,{Value: Text(Now();"yyyy")-2};
4 {Value: Text(Now();"yyyy")-1};
5 {Value: Text(Now();"yyyy")+0};

```

```

6  {Value: Text(Now();"yyyy")+1};
7  {Value: Text(Now();"yyyy")+2}
8  );;
9
10 ClearCollect(colNumbers; {Value: 1};{Value: 2};{Value: 3};{Value: 4};{
    Value: 5});;
11
12 Set(UserVar;UżytkownicyusługiOffice365.MyProfile());;
13
14 // W OnStart aplikacji lub OnVisible ekranu
15 ClearCollect(LocalServiceData; Lista_Uslug);;
16 ClearCollect(LocalCostData; Lista_Kwot);;
17 ClearCollect(LocalIndicationsData; Lista_Indykacji);;
18 ClearCollect(
19     MergedData;
20     AddColumns(
21         Lista_Uslug;
22         Kwoty; Lookup(
23             Lista_Kwot;
24             Service_ID = Lista_Uslug[@Service_ID] &&
25             Year = Max(Filter(Lista_Kwot; Service_ID = Lista_Uslug[
                @Service_ID]); Year)
26         );
27         Indykacje; Lookup(
28             Lista_Indykacji;
29             Service_ID = Lista_Uslug[@Service_ID] &&
30             Year = Max(Filter(Lista_Indykacji; Service_ID = Lista_Uslug[
                @Service_ID]); Year) &&
31             IndicationNo = Max(Filter(Lista_Indykacji; Service_ID =
                Lista_Uslug[@Service_ID] && Year = Max(Filter(
                Lista_Indykacji; Service_ID = Lista_Uslug[@Service_ID]);
                Year)); IndicationNo)
32         )
33     )
34 );;
35 Set(varDownloadingData; false);;

```

LISTING 4.1: Kod wywoływany podczas uruchamiania aplikacji

W pierwszej kolejności funkcja *Set*, przypisuje zmiennej *varDownloadingData* wartość *true*. Odpowiada ona za wyświetlenie indykatora ładowania oraz zablokowanie elementów ekranu co ma na celu zabezpieczenie przed wprowadzaniem zmian podczas ładowania się aplikacji. Następnie wywołana zostaje funkcja *ClearCollect*, która tworzy kolekcję o nazwie *colYears*, przekazanej jako pierwszy argument. Drugi argument zawiera tablicę, która ma być przypisana do kolekcji. Zawiera ona pięć elementów, które są rokami wykorzystywanymi w polach *Dropdown*. W podobny sposób powstaje kolekcja *colNumbers*, która zawiera numery indykacji. Funkcja *Set* w linii 12, pobiera informacje dotyczące obecnego użytkownika. Później powstają lokalne kopie trzech list: *LocalServiceData*, *LocalCostData* oraz *LocalIndicationsData*.

Przedostatnim krokiem jest utworzenie kolekcji *MergedData*, która zawiera połączone informacje z list *Lista_Uslug*, *Lista_Kwot* oraz *Lista_Indykacji*. Funkcja *AddColumns()* jako pierwszy argument przyjmuje nazwy kolumn, drugi, opcjonalny argument to źródło z którego pobierane są

dane natomiast trzecim jest warunkiem określającym jakie kolumny ze źródła wyodrębnić. Dodawane kolumny wyszukiwane są w innych listach z użyciem funkcji *LookUp()* oraz *Filter()*. Kolekcja ta łączy ze sobą dane dopasowując je względem *Service_ID*, *Year* oraz *IndicationNo* tworząc tabelę zawierającą informacje z najnowszej indykacji.

Na końcu do zmiennej *varDownloadingData* przypisywana jest wartość *false*, co oznacza, że dane zostały już pobrane i aplikacja jest gotowa do użytku.

Celem pobrania lokalnych kopii list, jest przyspieszenie działania mechanizmu filtrowania podczas wyboru usługi do edycji.

4.4 Edycja danych

Po zapisaniu najnowszych danych, następnym krokiem jest wprowadzenie niezbędnych aktualizacji dotyczących usług. W związku z tym utworzono dwa ekrany – pierwszy służy do wyboru usługi z listy, a drugi przedstawia niezbędne szczegóły oraz umożliwia edycję związanych z nią informacji.

4.4.1 Ekran wyboru usługi do edycji

Lista wyboru serwisu

Lista prezentuje dane pochodzące z dynamicznej kolekcji *MergedData*, która została utworzona w celu zintegrowania informacji z trzech różnych list SharePoint (*Lista_Uslug*, *Lista_Kwot*, *Lista_Indykacji*). Dzięki zastosowaniu wbudowanych funkcji *LookUp* (wyszukiwanie pojedynczego elementu w źródle danych na podstawie warunku) oraz *AddColumns* (dodawanie nowych kolumn do istniejącego źródła danych), dane są filtrowane tak, aby wyświetlać jedynie najnowsze rekordy, np. dla najświeższych decyzji i kwot. Taka struktura zapewnia użytkownikom dostęp do aktualnych informacji bez konieczności ręcznego przeszukiwania źródłowych list. Lista jest dynamiczna, co oznacza, że zmiany w danych źródłowych są automatycznie uwzględniane.

Każdy element na liście posiada dodatkową funkcję interakcji. Po najechaniu kursorem na wybraną usługę (funkcja *hover*), element wizualnie zmienia swój wygląd – zwęża się, zmienia kolor oraz staje się możliwy do kliknięcia. Kliknięcie przenosi użytkownika do dedykowanego ekranu edycji, który umożliwia szczegółowe zarządzanie wybraną usługą.

Pola wyszukiwania i filtry

Użytkownik ma możliwość zawężenia widocznych danych poprzez zastosowanie różnych kryteriów wyszukiwania. Pola obejmują:

- Wyszukiwanie po nazwie usługi (*Service Name*) – Obsługuje częściowe dopasowania dzięki wbudowanej funkcji *StartsWith* (sprawdza, czy ciąg tekstowy zaczyna się od określonej frazy).
- Filtrowanie po identyfikatorze usługi (*Service ID*) – Umożliwia precyzyjne wyszukiwanie konkretnego elementu.
- Filtrowanie według miejsca powstawania kosztów (*MPK*) – Pozwala na szybkie odnalezienie danych przypisanych do konkretnego obszaru finansowego.
- Filtrowanie według statusu decyzji (*Accepted, Not Accepted, No Status*) – Dzięki zastosowaniu funkcji *Switch* (zwraca wartość w zależności od spełnionego warunku), wartości statusów są mapowane na odpowiednie kody liczbowe.

Filtry te mogą być stosowane jednocześnie, co umożliwia precyzyjne dopasowanie wyświetlanych danych.

Wykres kołowy

Wykres kołowy ilustruje podział danych według statusów decyzji. Każdy segment odpowiada liczbie elementów z przypisanym statusem, co pozwala użytkownikowi szybko ocenić proporcje między kategoriami (*Accepted*, *Not Accepted*, *No Status*). Wykres jest dynamiczny – aktualizuje się w czasie rzeczywistym w oparciu o zastosowane filtry.

4.4.2 Ekran edycji elementu

Ekran edycji elementu prezentuje szczegółowe informacje dotyczące wybranej usługi, umożliwiając analizę danych historycznych oraz wprowadzanie nowych decyzji. Ekran składa się z kilku logicznie ułożonych sekcji, które są opisane poniżej.

Porównanie finalnych decyzji z poprzednich lat

W górnej części ekranu znajduje się tabela przedstawiająca porównanie danych finansowych oraz decyzji z trzech ostatnich lat. Dane te są automatycznie pobierane i zawierają:

- Rok (Year) – okres, którego dotyczy dana decyzja,
- Jednostka miary (Unit Of Measurement) – zazwyczaj ilość licencji. **w sumie to czasem było niecałkowite to idk**
- Zeszłoroczne i planowane na następny rok wartości finansowe – np. *Current Year Plan* oraz *Next Year Plan*,
- Różnice w finansach (Difference) – różnica między zeszłorocznymi i potencjalnymi przyszłymi kosztami,
- Status końcowej decyzji (Final Decision) – decyzje dotyczące planów z danego roku (*Accepted*, *Not Accepted*, *No Status*).

Sekcja ta pozwala użytkownikowi przeanalizować ostatnie decyzje oraz ocenić trendy finansowe dla danej usługi w kolejnych latach.

Link do instrukcji obsługi

Poniżej tabeli z porównaniem decyzji znajduje się link do dedykowanej instrukcji obsługi usługi. Za pomocą linku użytkownik posiada dostęp do szczegółowych informacji na temat zasad korzystania z danej usługi, co może być przydatne podczas edycji danych lub wprowadzania nowych decyzji.

Porównanie tegorocznych indykacji

Sekcja ta prezentuje szczegóły kolejnych indykacji w ramach bieżącego roku. Użytkownik może zobaczyć i analizować szczegóły poszczególnych indykacji, takich jak:

- Numer indykacji (Indication Number) – **??? wtf -¿ numer kolejny przypisany do konkretnej decyzji**,

- Komentarze – w tym *Internal Comment*, *Comment PZ to WOB*, *Comment K-DES*, które umożliwiają przekazanie informacji pomiędzy działami,
- Data i autor komentarza – **tutaj umó też zbędne bo chyba nie trzeba tłumaczyć że autor komentarza to autor komentarza a data komentarza to data komentarza nie? dane dotyczące daty wprowadzenia decyzji oraz osoby odpowiedzialnej**,
- Status decyzji (*Decision*) – użytkownik może zobaczyć, czy decyzja została zaakceptowana (*Accepted*), odrzucona (*Not Accepted*) lub jeszcze nie podjęta (*No Status*).

Formularz do uzupełnienia danych

Na samym dole strony znajduje się formularz umożliwiający wprowadzenie nowych danych lub aktualizację istniejących rekordów. Formularz zawiera pola takie jak:

- Rok (*Year*) – użytkownik może wybrać rok, którego dotyczy wpis,
- Numer indykacji (*Indication Number*) – kolejny numer przypisany do decyzji,
- Komentarze – pola do wprowadzenia uwag wewnętrznych, komentarzy między działami oraz końcowych komentarzy,
- Status decyzji (*Decision*) – lista rozwijana, umożliwiająca wybór odpowiedniego statusu (*Accepted*, *Not Accepted*, *No Status*),
- Data i autor – data oraz osoba odpowiedzialna za wprowadzenie wpisu.

Przycisk *Save* umożliwia zapisanie wprowadzonych zmian. Mechanizm ten:

- Sprawdza istnienie wcześniejszych indykacji, aby upewnić się, że zachowana jest poprawna kolejność numeracji,
- W przypadku istniejącego wpisu – aktualizuje dane (*Patch*),
- W przypadku nowego wpisu – tworzy nowy rekord (*Defaults*),
- Resetuje pola formularza oraz odświeża dane na ekranie, aby uwzględnić ostatnie zmiany,
- Informuje użytkownika o powodzeniu lub błędach operacji za pomocą komunikatów (*Notify*).

Podsumowanie

To to chyba do rozdziału 'Podsumowanie' będzie co? Ekran edycji elementu został zaprojektowany tak, aby umożliwić użytkownikom zarówno przeglądanie historycznych danych finansowych, jak i łatwe wprowadzanie nowych decyzji lub modyfikację istniejących. Dzięki logicznemu układowi sekcji oraz dynamicznej aktualizacji danych, interfejs wspiera efektywne zarządzanie informacjami dla każdej usługi.

4.5 Generowanie raportu

Docelowym działaniem aplikacji, będącym zwieńczeniem wszystkich jej funkcjonalności, jest możliwość wygenerowania finalnego raportu w formacie arkusza kalkulacyjnego na podstawie uzupełnionej bazy danych. W związku z tym w Power Apps przygotowano mechanizm łączący dane z trzech list sharepointowych, które następnie są przesyłane do Power Automate, gdzie podlegają dalszemu przetwarzaniu. Wynikiem tego jest kompletny arkusz Excel, który jest zapisywany na platformie SharePoint.

4.5.1 Łączenie źródeł danych do formy docelowej

Pierwszym krokiem do wygenerowania gotowego raportu jest konsolidacja potrzebnych danych w odpowiedniej formie. Do tego celu zaimplementowano ekran przedstawiony na rysunku 4.10. Składa się on z dwóch formularzy – węższy po lewej stronie oraz szerszy po prawej. W pierwszym z nich znajdują się dwa pola wyboru typu *Dropdown*, w których należy wybrać numer indykacji oraz roku do pobrania danych z list, natomiast w drugiej widnieje podgląd zawartości danych do wygenerowania w formie tabeli, która może zostać przekazana do Power Automate.

RYSUNEK 4.10: Ekran generowania raportu

Zebrane dane przechowywane są w tymczasowej kolekcji *CombinedData*, która składa się z informacji z trzech źródeł, odpowiednio dopasowanych do wybranego przez użytkownika roku oraz indykacji. Przykładowy fragment kodu odpowiedzialnego za tworzenie tej kolekcji przedstawiono w listingu 4.2:

```

1      ClearCollect(
2      CombinedData;
3      AddColumns(
4          LocalServiceData;
5          MPK;
6          LookUp(
7              LocalCostData;
8              Service_ID = LocalServiceData[@Service_ID] && Year =
                Year_Dropdown_1.Selected.Value;
9          MPK
10         );
11         Comment_PZ_to_WOB;
12         LookUp(
13             LocalIndicationsData;
14             Service_ID = LocalServiceData[@Service_ID] && Year =
                Year_Dropdown_1.Selected.Value && IndicationNo =
                IndicationNo_Dropdown_1.Selected.Value;
15             Comment_PZ_to_WOB);
16     [...]
```

```
17 );;
```

LISTING 4.2: Fragment kodu tworzącego kolekcję *CombinedData*

W fragmencie kodu przedstawiono wykorzystanie funkcji *LookUp*, która pozwala na pobranie danych z różnych kolekcji na podstawie określonych kryteriów. Przykładowo, pierwsze wywołanie *LookUp* dopasowuje dane z tabeli *LocalCostData*

Nie no musze zobaczyc w apce jak te lookupy byly zrobione bo nie pamietam, ale to jutro.

Przecież masz w kodzie xD

4.5.2 Przekazywanie danych do Power Automate

W lewym dolnym rogu interfejsu znajduje się przycisk (*Generate file*), który po naciśnięciu uruchamia *flow* w Power Automate, odpowiedzialne za tworzenie pliku na podstawie danych wejściowych. Dane te są przekazywane do usługi Power Automate w postaci ciągu tekstowego, sformatowanego jako JSON⁷, wykorzystując połączone informacje z trzech źródeł zawarte w kolekcji *CombinedData*, dostosowane do wybranego przez użytkownika zestawu roku i indykacji. Kod wywołujący funkcję *GenerateRaport* przedstawiono w listingu 4.3.

```
1      GenerateRaport.Run(
2          Substitute(
3              "[ " &
4              Concat(
5                  CombinedData;
6                  "{ \"Service_ID\":\"\" & Service_ID & \"\", \" &
7                  \"Service_Name\":\"\" & Service_Name & \"\", \" &
8                  \"MPK\":\"\" & MPK & \"\", \" &
9                  \"Comment_PZ_to_WOB\":\"\" & Comment_PZ_to_WOB & \"\", \" &
10                 \"Decision\":\"\" & Decision & \"\" }\", \"
11             ) &
12             \" ] \",
13             \" }, ] \";
14             \" } ] \"
15         ),
16         IndicationNoCollect.Value,
17         YearNoCollect.Value
18     )
```

LISTING 4.3: Kod wywołujący funkcję *GenerateRaport*

⁷JSON - format wymiany danych, który jest oparty na strukturze tekstowej

Rozdział 5

Rozwinięcie

Rozdziały dokumentujące pracę własną studenta: opisujące ideę, sposób lub metodę rozwiązania postawionego problemu oraz rozdziały opisujące techniczną stronę rozwiązania — dokumentacja techniczna, przeprowadzone testy, badania i uzyskane wyniki.

Praca musi zawierać elementy pracy własnej autora adekwatne do jego wiedzy praktycznej uzyskanej w okresie studiów. Za pracę własną autora można uznać np.: stworzenie aplikacji informatycznej lub jej fragmentu, zaproponowanie algorytmu rozwiązania problemu szczegółowego, przedstawienie projektu np. systemu informatycznego lub sieci komputerowej, analizę i ocenę nowych technologii lub rozwiązań informatycznych wykorzystywanych w przedsiębiorstwach, itp.

Autor powinien zadbać o właściwą dokumentację pracy własnej obejmującą specyfikację założeń i sposób realizacji poszczególnych zadań wraz z ich oceną i opisem napotkanych problemów. W przypadku prac o charakterze projektowo-implementacyjnym, ta część pracy jest zastępowana dokumentacją techniczną i użytkową systemu.

W pracy **nie należy zamieszczać całego kodu źródłowego** opracowanych programów. Kod źródłowy napisanych programów, wszelkie oprogramowanie wytworzone i wykorzystane w pracy, wyniki przeprowadzonych eksperymentów powinny być przekazane promotorowi oraz wgrane wraz z pracą do systemu informatycznego uczelni.

Styl tekstu

Należy¹ [1] stosować formę bezosobową, tj. *w pracy rozważono*, *w ramach pracy zaprojektowano*, a nie: *w pracy rozważyłem*, *w ramach pracy zaprojektowałem*. Odwołania do wcześniejszych fragmentów tekstu powinny mieć następującą postać: „Jak wspomniano wcześniej,”, „Jak wykazano powyżej”. Należy unikać długich zdań.

Niedopuszczalne są zwroty używane w języku potocznym. W pracy należy używać terminologii technicznej, która ma sprecyzowaną treść i znaczenie.

Niedopuszczalne jest pisanie pracy metodą *copy-paste*, bo jest to plagiat i dowód intelektualnej indolencji autora. Dane zagadnienie należy opisać własnymi słowami. Zawsze trzeba powołać się na zewnętrzne źródła.

¹Uwagi o stylu pochodzą częściowo ze stron prof. Macieja Drozdowskiego.

Rozdział 6

Zakończenie

Zakończenie pracy zwane również Uwagami końcowymi lub Podsumowaniem powinno zawierać ustosunkowanie się autora do zadań wskazanych we wstępie do pracy, a w szczególności do celu i zakresu pracy oraz porównanie ich z faktycznymi wynikami pracy. Podejście takie umożliwia jasne określenie stopnia realizacji założonych celów oraz zwrócenie uwagi na wyniki osiągnięte przez autora w ramach jego samodzielnej pracy.

Integralną częścią pracy są również dodatki, aneksy i załączniki zawierające stworzone w ramach pracy programy, aplikacje i projekty.

Literatura

- [1] Maciej Drozdowski. Jak pisać prace dyplomowe – uwagi o formie. [on-line]
http://www.cs.put.poznan.pl/mdrozdowski/dyd/txt/jak_mgr.html, 2006.
- [2] Donald E. Knuth. *The T_EXbook*. Computers and Typesetting. Addison-Wesley, Reading, MA, USA, 1986.
- [3] Leslie Lamport. *L^AT_EX — A Document Preparation System — User’s Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985.

Dodatek A

Składanie dokumentu w systemie L^AT_EX

W tym rozdziale znajduje się garść informacji o tym, jak poprawnie składać tekst pracy w systemie L^AT_EX wraz z przykładami, które mają służyć do przeklejanania do własnych dokumentów.

A.1 Struktura dokumentu

Praca składa się z rozdziałów (`chapter`) i podrozdziałów (`section`). Ewentualnie można również rozdziały zagnieżdzać (`subsection`, `subsubsection`), jednak nie powinno się wykraczać poza drugi poziom hierarchii (czyli `subsubsection`).

A.2 Akapity i znaki specjalne

Akapity rozdziela się od siebie przynajmniej jedną pustą linią. Podstawowe instrukcje, które się przydają to *wyróżnienie pewnych słów*. Można również stosować **styl pogrubiony**, choć nie jest to generalnie zalecane.

Należy pamiętać o zasadach polskiej interpunkcji i ortografii. Po spójnikach jednoliterowych warto wstawić znak tyldy (`~`), który jest tak zwaną „twardą spacją” i powoduje, że wyrazy nią połączone nie będą rozdzielane na dwie linie tekstu.

Polskie znaki interpunkcyjne różnią się nieco od angielskich: to jest „polski”, a to jest “angielski”. W kodzie źródłowym tego tekstu będzie widać różnicę.

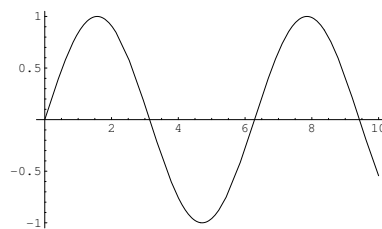
Proszę również zwrócić uwagę na znak myślnika, który może być pauzą „—” lub półpauzą: „-”. Należy stosować je konsekwentnie. Do łączenia wyrazów używamy zwykłego „-” (*północno-wschodni*), do myślników — pauzy lub półpauzy. Inne zasady interpunkcji i typografii można znaleźć w słownikach.

A.3 Wypunktowania

Wypunktowanie z cyframi:

1. to jest punkt,
2. i to jest punkt,
3. a to jest ostatni punkt.

Po wypunktowaniach czasem nie warto wstawiać wcięcia akapitowego. Wtedy przydatne jest polecenie `noindent`. Wypunktowanie z kropkami (tzw. *bullet list*) wygląda tak:



RYSUNEK A.1: Wykres.

- to jest punkt,
- i to jest punkt,
- a to jest ostatni punkt.

Wypunktowania opisowe właściwie niewiele się różnią:

elementA to jest opis,

elementB i to jest opis,

elementC a to jest ostatni opis.

A.4 Polecenia pakietu *ppcreefthesis*

Parę poleceń zostało zdefiniowanych aby uspójnić styl pracy. Są one przedstawione poniżej (oczywiście nie trzeba się do nich stosować).

Makra zdefiniowane dla języka angielskiego. Są nimi: `termdef` oraz `acronym`. Przykłady poniżej obrazują ich przewidywane użycie w tekście.

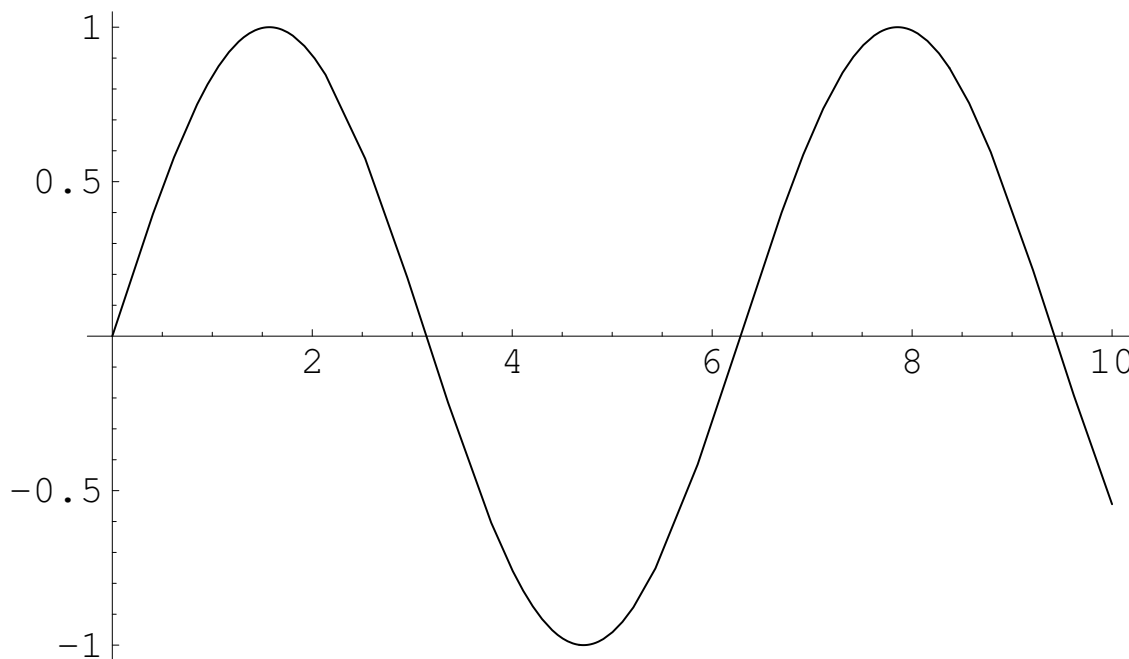
źródło	<code>we call this a \termdef{Database Management System} (\acronym{DBMS})</code>
docelowo	we call this a <i>Database Management System (DBMS)</i>

Makra zdefiniowane dla języka polskiego. Podobnie jak dla języka angielskiego zdefiniowano odpowiedniki polskie: `definicja`, `akronim` oraz `english` dla tłumaczeń angielskich terminów. Przykłady poniżej obrazują ich przewidywane użycie w tekście.

źródło	<code>nazywamy go \definicja{systemem zarządzania bazą danych} (\akronim{DBMS}, \english{Database Management System})</code>
docelowo	nazywamy go <i>systemem zarządzania bazą danych (DBMS, ang. Database Management System)</i>

A.5 Rysunki

Wszystkie rysunki (w tym również diagramy, szkice i inne) osadzamy w środowisku `figure` i umieszczamy podpis *pod* rysunkiem, w formie elementu `caption`. Rysunki powinny zostać umieszczone u góry strony (osadzone bezpośrednio w treści strony zwykle utrudniają czytanie tekstu). Rysunek A.1 zawiera przykład pełnego osadzenia rysunku na stronie.



RYSUNEK A.2: Ten sam wykres ale na szerokość tekstu.

A.5.1 Tablice

Tablice to piękna rzecz, choć akurat ich umiejętne tworzenie w \LaTeX u nie jest łatwe. Jeśli tablica jest skomplikowana, to można ją na przykład wykonać w programie OpenOffice, a następnie wyeksportować jako plik *PDF*. W każdym przypadku tablice wstawia się podobnie jak rysunki, tylko że w środowisko `table`. Tradycja typograficzna sugeruje umieszczenie opisu tablicy, a więc elementu `caption` ponad jej treścią (inaczej niż przy rysunkach).

Tablica A.1 pokazuje pełen przykład.

TABELA A.1: Przykładowa tabela. Styl opisu jest zgodny z rysunkami.

artykuł	cena [zł]
bułka	0,4
masło	2,5

A.5.2 Przydatne uwagi

- Znakiem myślnika jest w \LaTeX u dywiz pełen (`—`) albo półpauza (`-`), przykład: A niech to jasna cholera `—` wrzasnąłem.
- Połączenie między wyrazami to zwykły myślnik, przykład: północno-zachodni
- Sprawdź ostrzeżenia o `'overfull'` i `'underful'` boxes. Niektóre z nich można zignorować (spójrz na wynik formatowania), niektóre trzeba poprawić; czasem przeformułować zdanie.
- Przypisy stawia się wewnątrz zdań lub za kropką, przykład: Footnote is added after a comma.¹
- Nie używaj przypisów zbyt często. Zobacz, czy nie lepiej będzie zintegrować przypis z tekstem.

¹Here is a footnote.

- Tytuły tabel, rysunków powinny kończyć się kropką.
- Nie używaj modyfikatora [h] (here) do rysunków i tabel. Rysunki i tabele powinny być justowane do góry strony lub na stronie osobnej.
- Wyróżnienie w tekście to polecenie *wyraz*, należy unikać **czcionki pogrubionej** i podkreślenia (które wystają wizualnie z tekstu i rozpraszaają).
- Nazwy plików, katalogów, ścieżek, zmiennych środowiskowych, klas i metod formatujemy poleceniem `plik_o_pewnej_nazwie`.
- Po ostatniej zmianie do treści, sprawdź i przenieś wiszące spójniki wstawiając przed nie znak tyldy (twardej spacji), przykład: Ala i kotek nie lubią mleczka, a Stasiu lubi.
- Za i.e. (id est) i e.g. (exempli gratia) stawia się zwyczajowo przecinek w typografii amerykańskiej.
- Przed i za pełną pauza nie ma zwyczajowo spacji w typografii amerykańskiej, przykład: Darn, this looks good—said Mary.
- Zamykający cudzysłów oraz footnote wychodzą za ostatni znak interpunkcji w typografii amerykańskiej, przykłady: It can be called a “curiosity,” but it’s actually normal. Footnote is added after a comma.²
- Odwołania do tabel i rysunków zawsze z wielkiej litery, przykład: In Figure A.1 we illustrated XXX and in Table A.1 we show detailed data.

A.6 Literatura i materiały dodatkowe

Materiałów jest mnóstwo. Oto parę z nich:

- *The Not So Short Introduction...*, która posiada również tłumaczenie w języku polskim.
<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>
- Klasy stylu `memoir` posiadają bardzo wiele informacji o składzie tekstów anglosaskich oraz sposoby dostosowania \LaTeX a do własnych potrzeb.
<http://www.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf>
- Nasza grupa dyskusyjna i repozytorium Git są również dobrym miejscem aby zapytać (lub sprawdzić czy pytanie nie zostało już zadane).
<https://github.com/politechnika/put-latex>
- Dla łaknących więcej wiedzy o systemie \LaTeX podstawowym źródłem informacji jest książka Lamport’a [3]. Prawdziwy *hardcore* to oczywiście *The \TeX book* profesora Knutha [2].

²Here is a footnote.



© 2025 Remigiusz Wolniak, Michał Gajdzis

Instytut Robotyki i Inteligencji Maszynowej

Wydział Automatyki, Robotyki i Elektrotechniki Politechnika Poznańska

Skład przy użyciu systemu \LaTeX na platformie Overleaf.