



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
	Implementacja relacji	12
3.2.2	Dodawanie informacji do bazy danych	12
3.2.3	Interfejs procesu decyzyjnego	12
	Ekran listy serwisów	12
	Ekran szczegółowy serwisu	13
3.2.4	Generowanie raportu	13
4	Implementacja	14
4.1	Ekran dodawania danych	14
4.1.1	Zapis pliku w chmurze	14
	Przepływ SaveFileAndRunScript	14
4.1.2	Skrypt pakietu Office	16
4.1.3	Walidacja nazw kolumn	18

4.1.4	Integracja z listami SharePoint	19
5	Rozwinięcie	20
6	Zakończenie	21
	Literatura	22
A	Składanie dokumentu w systemie L^AT_EX	23
A.1	Struktura dokumentu	23
A.2	Akapity i znaki specjalne	23
A.3	Wypunktowania	23
A.4	Polecenia pakietu <code>ppcreefthesis</code>	24
A.5	Rysunki	24
A.5.1	Tablice	25
A.5.2	Przydatne uwagi	25
A.6	Literatura i materiały dodatkowe	26

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 IT*¹ 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 Bręczyszczykiewicz 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 z tym 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 skrószycem, 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

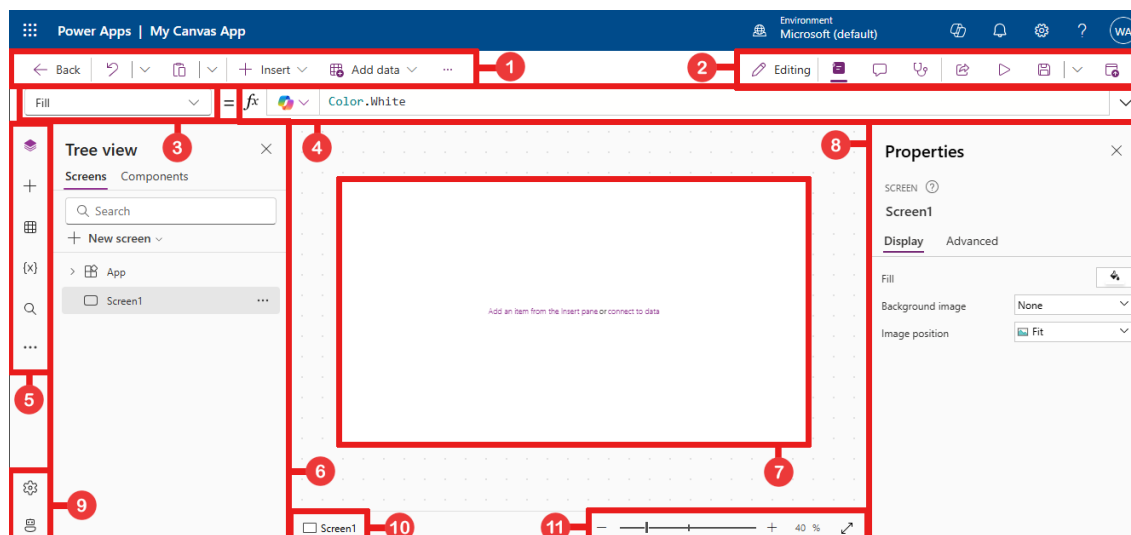
Power Apps to kolejne środowisko Low-Code, wchodzące w skład pakietu Microsoft 365, które jest dedykowanym rozwiązaniem do tworzenia aplikacji biznesowych. Dzięki intuicyjnemu interfejsowi graficznemu daje możliwość prostej implementacji mechanizmu działania nawet przez osoby bez zaawansowanej wiedzy 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 edycję *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 kluczowe 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 posiadały ustandaryzowanej struktury, co znacząco utrudniało ich czytelność i wymagało od użytkowników dodatkowego czasu na analizę zawartych informacji.

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

TABELA 3.1:

	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 uniemożliwiał również stworzenie spójnej bazy, co ograniczało 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

Dedykowane narzędzie z prostym i intuicyjnym interfejsem znacząco ułatwia nawigację po bazie danych, eliminując problemy związane z tradycyjnymi rozwiązaniami, takimi jak arkusze kalkulacyjne. Przyjazny dla użytkownika interfejs w tym 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.

Niestety użyty pakiet, nie jest dostępny w najbardziej rozbudowanym wariantcie. Wprowadza to pewne ograniczenia, ponieważ brakuje w nim 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. Głównym problemem było ograniczenie związane z brakiem możliwości tworzenia relacji między kilkoma listami, co znacząco utrudniało zarządzanie danymi o złożonej strukturze.

¹Platforma Power (ang. *Power Platform*) – Składowa pakietu Microsoft 365. Zawiera ona takie programy jak Power Apps, Power Automate czy Power BI.

3.1.5 Optymalizacja

Priorytetem implementowanego rozwiązania jest usprawnienie całego procesu decyzyjnego. Głównym elementem jest redukcja czasu wymaganego na realizację poszczególnych zadań, co osiągnięto dzięki wprowadzeniu mechanizmów automatyzacji biurowej. Ważnym aspektem jest również poprawa efektywności analizy oraz przetwarzania danych przez użytkowników, umożliwiając podejmowanie bardziej trafnych decyzji w krótszym czasie.

Zmniejszenie liczby osób zaangażowanych w realizację procesu umożliwia optymalizację wykorzystania zasobów ludzkich. Dzięki temu możliwe jest ograniczenie kosztów operacyjnych i bardziej efektywne zarządzanie personelem, co przyczynia się do zwiększenia ogólnej wydajności przedsiębiorstwa.

3.2 Koncepcja rozwiązania

W niniejszym rozdziale 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. Mimo 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 indykacji. 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.

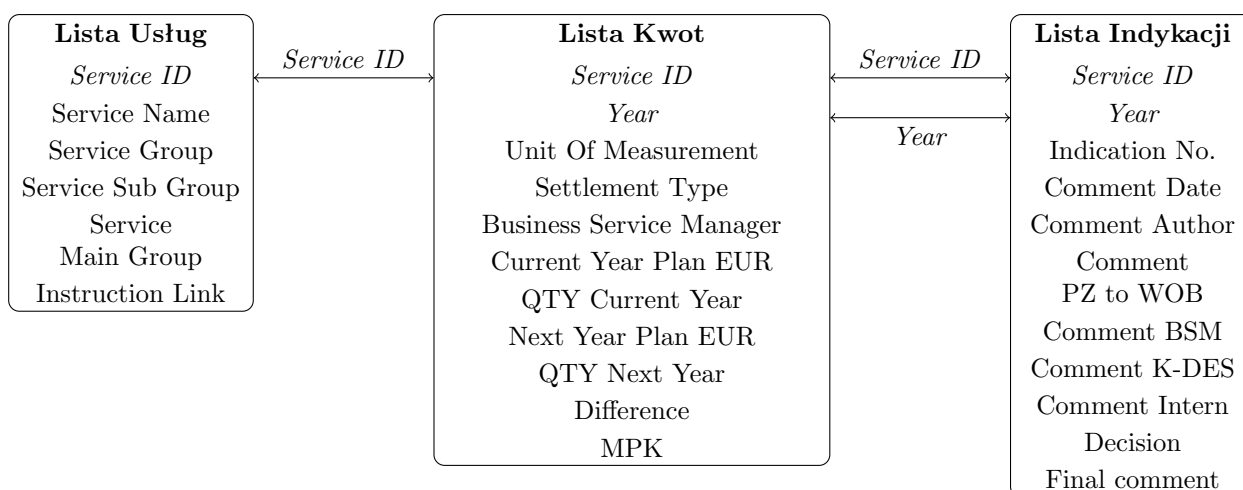
Atrybuty danych

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.

Implementacja relacji

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.

3.2.2 Dodawanie informacji do bazy danych

Po ustaleniu struktury danych wykorzystywanych przez system, kolejnym etapem było określenie sposobu importu informacji z arkuszy kalkulacyjnych do bazy 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. Proces rozpoczyna się od tymczasowego umieszczenia pliku Excel w folderze współdzielonym na platformie SharePoint. Podczas implementacji napotkano problem z widocznością danych – większość systemów może odczytać jedynie informacje zorganizowane w *tabele programu Excel*².

Rozwiązaniem okazało się zastosowanie skryptu pakietu Office, działającego bezpośrednio w arkuszu. Skrypt ten automatycznie tworzy tabelę o dynamicznym rozmiarze oraz usuwa puste kolumny. Zaimplementowany algorytm skutecznie identyfikuje początek tabeli oraz określa jej wymiary na podstawie liczby wierszy i kolumn, pomijając nieistotne dane.

W celu dostosowania danych do struktury bazy, zaimplementowano formularz walidacyjny dla nazw kolumn. System pobiera nazwy istniejących kolumn z arkusza i umożliwia ich mapowanie na predefiniowaną listę nagłówków z list SharePoint. Proces ten wykorzystuje wspomniany wcześniej skrypt do pobrania aktualnych nazw.

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

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

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 listy serwisów

Pierwszy ekran pełni funkcję panelu nawigacyjnego, prezentując podstawowe informacje o serwisach:

- Service Name – nazwa serwisu,

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

- Service ID – unikalny identyfikator,
- MPK – numer księgowy obszaru odpowiedzialnego,
- Decision – aktualny status decyzji.

Zaimplementowany system filtrowania umożliwia:

- wyszukiwanie serwisów po ID lub nazwie,
- filtrowanie według przypisanych numerów MPK,
- segregację według statusu decyzji (*Accepted*, *Not Accepted*, *No Status*).

Ekran szczegółowy serwisu

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

Sekcja historyczna Zawiera dwie uzupełniające się tabele:

- tabela przeglądowa – prezentująca historię decyzji z poprzednich lat,
- tabela szczegółowa – wyświetlająca pełne informacje dla wybranego roku (domyślnie rok poprzedni).

Formularz decyzyjny Zestaw pól do wprowadzenia informacji o bieżącej indykacji:

- 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).

Panel podglądu W celu zwiększenia czytelności, długie komentarze (powyżej 30 znaków) są początkowo prezentowane w formie skróconej. Pełna treść komentarza jest dostępna po jego wybraniu i wyświetlana w dedykowanym panelu podglądu.

System automatycznego uzupełniania wartości domyślnych przyspiesza proces decyzyjny, jednocześnie zachowując możliwość modyfikacji każdego z parametrów w razie potrzeby.

3.2.4 Generowanie raportu

Rozdział 4

Implementacja

W niniejszym rozdziale przedstawiono szczegóły techniczne zaimplementowanego rozwiązania. Omówione zostaną kluczowe 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, przedstawiona zostanie integracja z platformą SharePoint oraz implementacja skryptów usprawniających pracę z pakietem Microsoft Office. Rozdział stanowi techniczne rozwinięcie przyjętych założeń projektowych, prezentując metodykę realizacji poszczególnych komponentów systemu.

W ramach analizy technicznej zostaną szczegółowo omówione poszczególne komponenty systemu oraz sposób ich integracji. Szczególna uwaga zostanie poświęcona mechanizmom przepływu danych, automatyzacji procesów oraz implementacji logiki biznesowej w środowisku low-code.

4.1 Ekran dodawania danych

Zdecydowano, że pierwszym ekranem aplikacji będzie ekran zapisu danych. Decyzja ta wynika z faktu, że bez przetworzonych danych, utworzenie innych ekranów byłoby zdecydowanie trudniejsze. Ekran ten składa się z elementów, które zostaną omówione poniżej.

4.1.1 Zapis pliku w chmurze

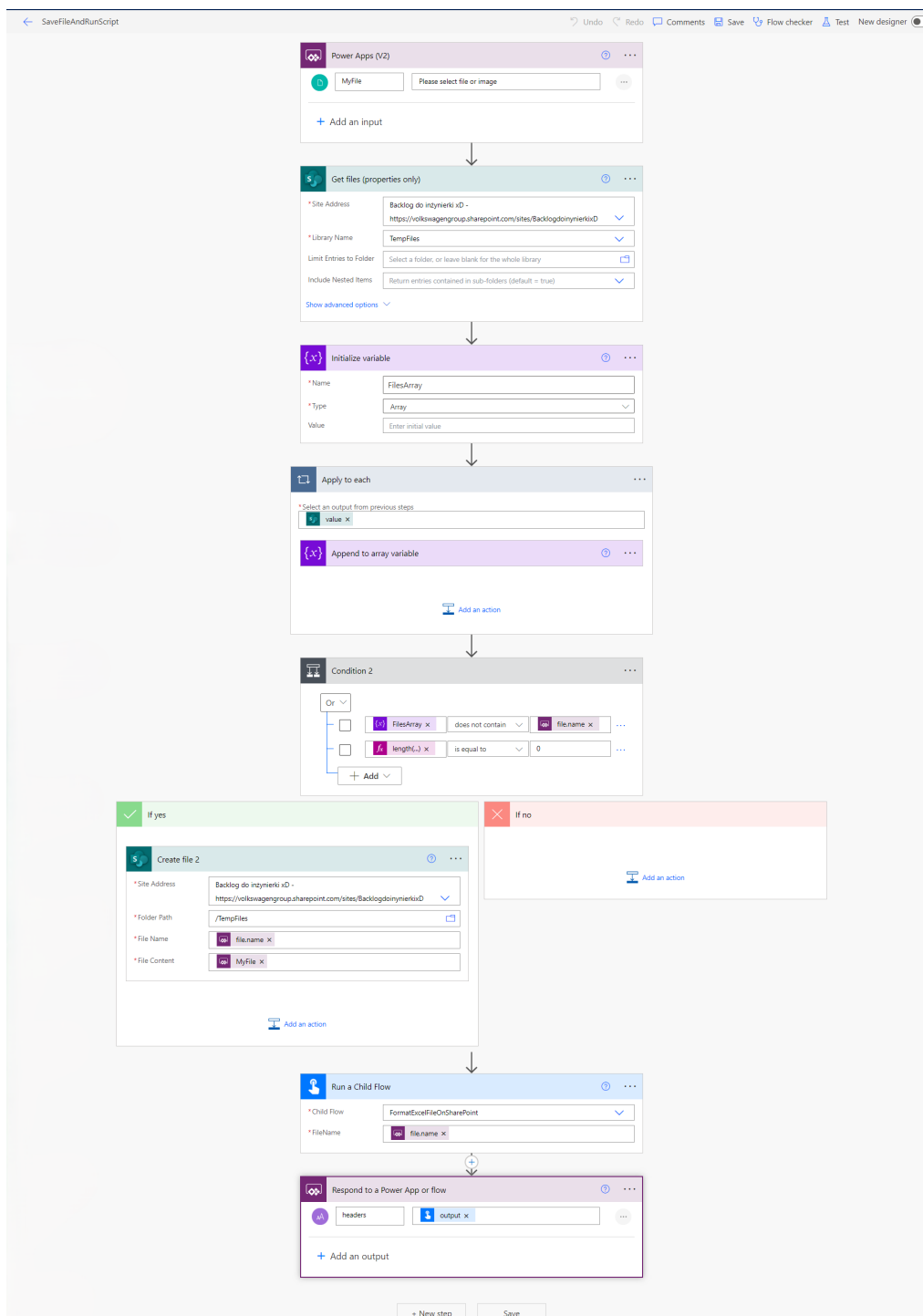
Pierwszym etapem procesu jest zapis pliku w chmurze, co umożliwia jego udostępnienie innym systemom. Do realizacji tego zadania wykorzystano kontrolkę¹ *Attachment Control*. Pozwala ona na zapisanie pliku w pamięci aplikacji. Odbywa się to przez naciśnięcie przycisku "Dołącz plik" lub przy użyciu mechaniki *przeciągnij i upuść* (ang. *Drag And Drop*).

Aby 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 wywołany 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 pamięci aplikacji zostają usunięte.

Przepływ *SaveFileAndRunScript*

Rysunek 4.1, przedstawia edytor programu Power Automate. Widoczny w nim przepływ nazywany *SaveFileAndRunScript* jest odpowiedzialny za zapisanie pliku w chmurze oraz wstępne

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



RYSUNEK 4.1: Widok przepływu SaveFileAndRunScript

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 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.

4.1.2 Skrypt pakietu Office

Po utworzeniu pliku w SharePoint, w ramach przepływu następuje jego przetworzenie przez skrypt. Jego zadaniem jest dostosowanie pliku 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, 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/>

RYSUNEK 4.2: Formularz zapisu pliku w chmurze

Rysunek 4.2 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.

4.1.3 Walidacja nazw kolumn

Kolejnym etapem przed zapisaniem danych do bazy jest walidacja nazw kolumn. W tym celu zaimplementowano formularz, którego układ przedstawiono na rysunku 4.3. 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*².
- Prawa kolumna zawiera kontrolkę *ComboBox*³, umożliwiającą wybór nazwy z listy standardowych nagłówków. Lista wartości w kontrolce *ComboBox* jest generowana przez skrypt opisany w poprzednich sekcjach.

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. 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 4.3 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.

² *Label* – kontrolka tekstowa umożliwiająca wyświetlanie statycznych wartości.

³ *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: Choose indication No.:

Upload data

RYSUNEK 4.3: Formularz walidacji nazw kolumn

4.1.4 Integracja z listami SharePoint

Po zakończeniu walidacji nazw kolumn, kolejnym etapem jest integracja przetworzonych danych z listami SharePoint. Proces ten rozpoczyna się od wyboru roku i numeru indykacji przy użyciu dedykowanych kontrolerek *Dropdown*⁴. Wybrane wartości są następnie wykorzystywane podczas importu danych do odpowiednich list, co odbywa się za pomocą przycisku *Upload data*.

⁴*Dropdown* – kontrolka umożliwiająca wybór jednej z dostępnych wartości z rozwijanej listy, bez możliwości edycji.

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 przeklejania 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.

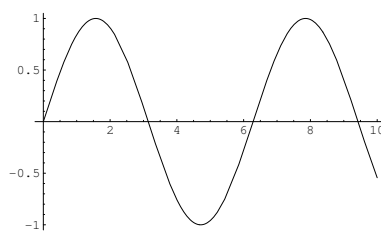
Proszę również zwrócić uwagę na znak myślnika, który może być pauzą „—” lub półpauzą: „-”. Należy stosować je konsekwentnie. Do łączenia wyrazów używamy zwykłego „-” (*północno-wschodni*), do myślników — pauzy lub półpauzy. Inne zasady interpunkcji i typografii można znaleźć w słownikach.

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 uspoźnić 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.

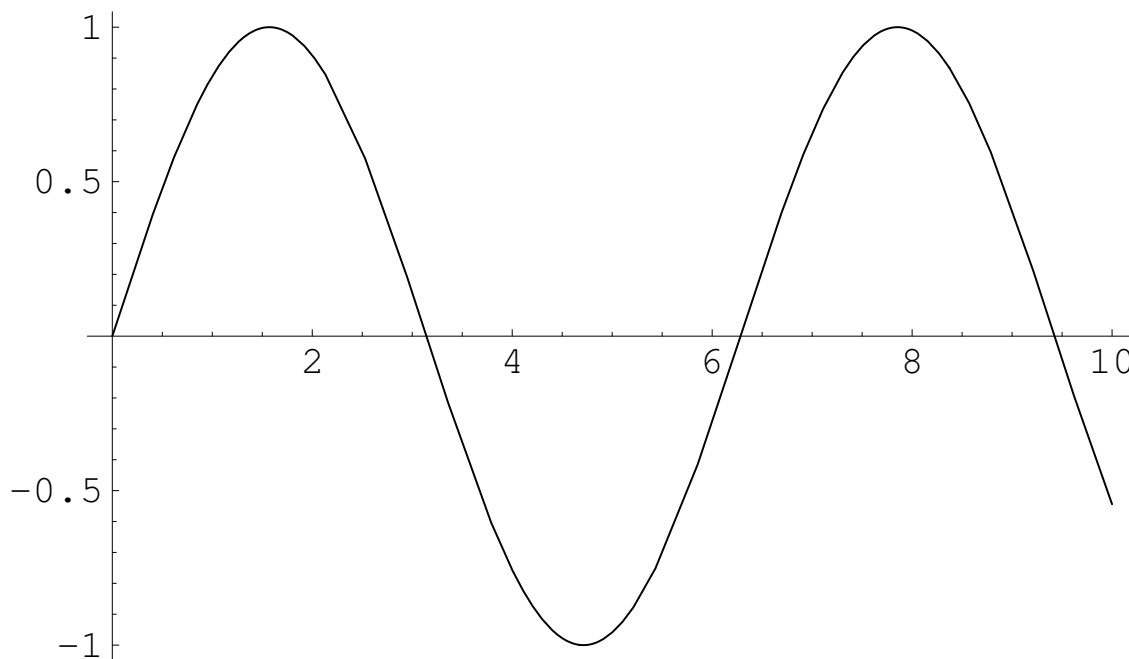
źródło	<code>we call this a \termdef{Database Management System} (\acronym{DBMS})</code>
docelowo	we call this a <i>Database Management System (DBMS)</i>

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

¹Here is a footnote.

- Nie używaj przypisów zbyt często. Zobacz, czy nie lepiej będzie zintegrować przypis z tekstem.
- 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.