



**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI**
POLITECHNIKI RZESZOWSKIEJ

Łukasz Miłoś

System zarządzania bazą BOM

**Projekt w ramach przedmiotu
"Usługi sieciowe w biznesie"**

Prowadzący zajęcia projektowe:
(dr inż.) Mariusz Borkowski (prof. PRz)

Rzeszów, 2022

Spis treści

1. Wstęp	5
2. Opis problemu	6
3. Rozwiązanie	10
4. Podsumowanie	19
Źródła	20

1. Wstęp

Projekt dotyczy przedmiotu „Usługi sieciowe w biznesie” i zgodnie z założeniami ma ściśle praktyczny charakter.

Sam przedmiot skupia się na zagadnieniach informatyzacji w przedsiębiorstwach, które wpływają na ich organizację i strukturę. Dzięki zastosowaniu różnej klasy systemów, działania ludzkie są wspierane, a często także zastępowane poprzez wykorzystanie nowych technologii, co ma swoje odzwierciedlenie w większej wydajności, produktywności, mniejszym ryzyku popełnienia błędu, a co za tym idzie także minimalizacji kosztów operacyjnych. Wiele zagadnień w przedsiębiorstwach można ułatwić poprzez zastosowanie odpowiedniego systemu zintegrowanego, stąd niezbędna wiedza o typach, funkcjach i przypadkach użycia poszczególnych technologii.

Spośród mnogości zagadnień wybrano temat dotyczący organizacji zasobów produkcyjnych, jako bardzo istotny element wielu przedsiębiorstw.

W ramach projektu utworzono system zarządzania bazą BOM (Bill of Materials). Zagadnienie to jest ściśle związane z przedmiotem i zostanie opisane w kolejnym rozdziale.

2. Opis problemu

BOM (Bill Of Materials) jest strukturalnym (hierarchicznym) zestawieniem materiałowym produktu końcowego zawierającym listę części składowych niezbędnych do jego wytworzenia wraz z podaniem cech określających dany zasób m.in. ceny i ilości. Takie zestawienia są wykorzystywane na różnych etapach produkcji, zarówno podczas projektowania, a także podczas wytwarzania czy nawet montażu.

Niekiedy BOM jest wzbogacany o opis kolejnych czynności, w których używane są poszczególne elementy składowe (przykładem jest dowolna instrukcja szafy przeznaczonej do samodzielnego montażu, która często zawiera elementy składowe wraz z wyszczególnieniem kolejnych czynności montażowych).

Poszczególne elementy posiadają kilka cech (atrybutów). W zależności od implementacji systemu parametry są różne. Oto kilka najpopularniejszych z nich:

- identyfikator
- nazwa
- ilość
- jednostka miary
- cena jednostkowa
- cena sumaryczna

Wyjściowym etapem w konstrukcji zestawienia materiałowego może być ogólne, koncepcyjne przedstawienie części składowych produktu finalnego w formie rysunku (2.1) lub grafu (2.2).

Należy zwrócić uwagę, że poszczególne elementy składowe, także składają się z mniejszych komponentów (hierarchia). Ważne jest zaprezentowanie odpowiedniego stopnia złożoności, który zależy od specyfiki danego przedsiębiorstwa. Przykładowo dla firmy produkującej rowery, schemat (2.1) może okazać się wystarczający, jeśli korzysta ona z gotowych produktów innych firm, a nie tworzy wszystkiego na własną rękę. Przykładem jest tutaj półprodukt hamulca. O ile firma zakupuje gotowe hamulce od innej marki i tylko montuje je w swoich rowerach to ten poziom wyszczególnienia jest wystarczający. Natomiast w przypadku produkcji hamulców wewnętrznie, należałoby



Rysunek 2.1: Przykład koncepcyjnego zestawienia BOM w formie rysunku

Źródło: <https://www.mecalux.pl/blog/zestawienie-materialowe-bom>

dodatkowo wyszczególnić części składowe hamulca (m.in. rączkę, mocowanie, śruby mocujące, linkę, gumki ścieralne, gumki ochronne, smar itd.).

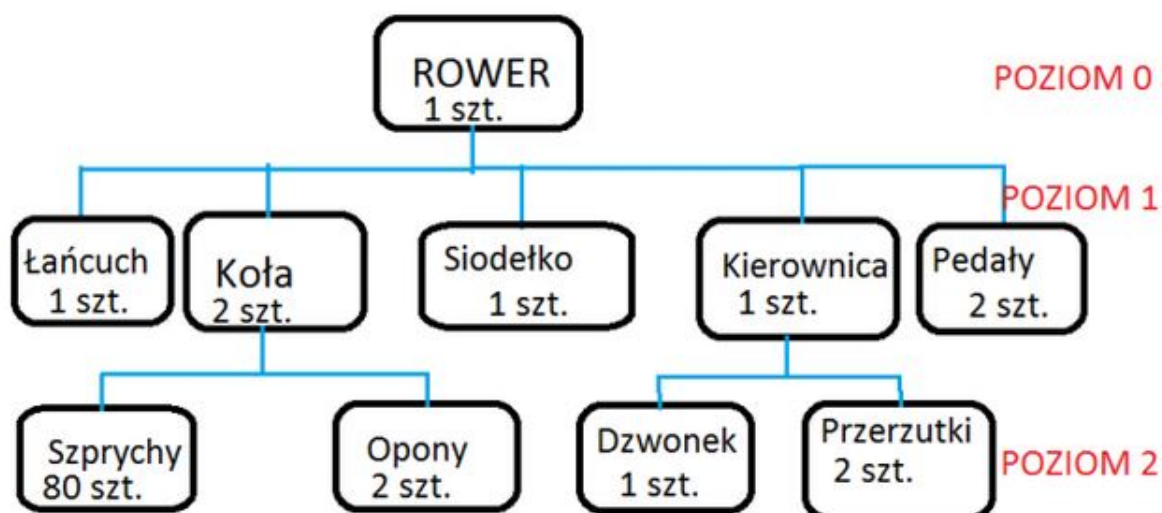
Innym sposobem koncepcyjnego zapisu zestawienia materiałowego dowolnego produktu może być graf (2.2). Na podstawie grafu można przedstawić hierarchię, ale nie jest możliwe przedstawienie informacji szczegółowych dotyczących poszczególnych wyrobów.

W rzeczywistości zestawienia materiałowe przyjmują jednak formę tabeli (2.3) lub hierarchii (2.4), z wyszczególnieniem poszczególnych elementów składowych.

Niestety wykorzystanie tabeli nie przedstawia wprost hierarchii poszczególnych komponentów. Hierarchia jest tutaj (2.3) oznaczona za pomocą pierwszej kolumny (BoM level). Zestawienie w formie tabeli w tym przypadku zawiera identyfikator części, opis, ilość, jednostkę, koszt jednostkowy i koszt całościowy.

Najlepszym sposobem prezentacji zestawienia materiałowego (BOM) jest pokazanie hierarchii (2.4), np. za pomocą drzewa hierarchicznego (treeview).

Na załączonym zdjęciu (2.4) pokazano przykład docelowej hierarchii. Jest także możliwość rozwijania i zwijania poszczególnych elementów składowych, co pozwala



Rysunek 2.2: Przykład koncepcyjnego zestawienia BOM w formie grafu

Źródło: <https://logistykalogike.wordpress.com/2014/12/18/struktura-wyrobu/>

zwiększyć czytelność zestawienia.

Temat organizacji zasobów produkcyjnych jest bardzo istotny, ponieważ umożliwia zachowanie ciągłości produkcji, a także pozwala na utrzymanie płynności finansowej przedsiębiorstwa.

Dzięki posiadaniu precyzyjnych zestawień można planować zakup surowców (ustrzeżenie się braku, ale także nadmiaru zapasów), a także określać niezbędne do poniesienia koszty (planowanie budżetu). Nie można również zapomnieć o tym, że mając wiedzę na temat niezbędnych materiałów i ich ilości można zarządzać zapasami na przyszłość i nie dopuścić do braku materiałów w magazynie, co jest bardzo niebezpieczne. Istotny jest także atut minimalizacji błędów, zyskany dzięki korzystaniu z utworzonego zestawienia.

Projekt jest zatem próbą znalezienia rozwiązania poprzez utworzenie własnego systemu do zarządzania hierarchiczną bazą BOM. Naturalnie takie systemy istnieją na rynku, ale są one zazwyczaj częścią zintegrowanych systemów zarządzania przedsiębiorstwem i nie stanowią oddzielnych bytów.

Single pane Window 500mm x 500mm						
BoM level	Part #	Description	Qty	Units	Unit Cost	Cost
1	756	Window framing	1	4	\$3.00	\$12.00
1	95	Brackets	1	4	\$0.75	\$3.00
1	PRS045	Rubber seal	2	metre	\$0.50	\$1.00
2	342	Glass pane	1	1	\$9.50	\$9.50
2	LB8579	Safety label	1	1	\$0.10	\$0.10
3	GH098	Hinges	2	1	\$2.25	\$4.50
3	GS664	Screws	8	10	\$4.95	\$3.96
3	587	Latch	1	1	\$2.20	\$2.20
3	588	Latch hook	1	1	\$0.88	\$0.88
4	GS660	Screws for latch and hook	6	10	\$4.95	\$2.97
5	812	Protective wrap	1.5	metre	\$0.65	\$0.98
6	XYZ123	Cardboard box 600mm x 600mm	1	1	\$1.00	\$1.00
6	LB7487	Box label barcode	1	1	\$0.10	\$0.10
Total number parts			27.5	Total costs		\$42.19

Rysunek 2.3: Przykład zestawienia BOM w formie tabeli

Źródło: <https://www.unleashedsoftware.com/blog/everything-you-need-to-know-about-bill-of-materials>

Number	Ro... /	Position Nu...	Q...	Title (Item,CO)	Revi...
ADSK-ITEM-000006	-	-	-	Bogie Assy	2
ADSK-ITEM-000004	1	8	1	Frame	1
ADSK-ITEM-000000	2	18	8	SideWheel Bolts	1
ADSK-ITEM-000001	2.1	1	1	Unbrako UNC ...	1
ADSK-ITEM-000002	2.2	2	2	Plain Washer (L...	1
ADSK-ITEM-000003	2.3	3	1	Hex Thick Nut -...	1
Side Wheel Sub	3	19	1	Side Wheel Sub	1
Swingarm 2 - Lower Guide W...	3.1	1	1	Swingarm 2 - L...	1
Swingarm 1 - Lower Guide W...	3.2	2	1	Swingarm 1 - L...	1
Side Wheel	3.3	3	2	Side Wheel.ipt	1
Spring	3.4	4	1	Spring	1
ADSK-ITEM-000000	3.5	5	4	SideWheel Bolts	1

Rysunek 2.4: Przykład zestawienia BOM w formie drzewa hierarchicznego

Źródło: <https://underthehood-autodesk.typepad.com/blog/items/page/2/>

3. Rozwiązanie

Sekcję rozwiązania podzielono na 4 części: zalecenia, założenia, opis, uruchomienie. Pierwsza z nich ogólnie przedstawia wyobrażenie idealnego systemu. Założenia są efektem podjęcia decyzji odnośnie wyboru technologii, która umożliwi realizację celu. Ostatnia z części krótko omawia sposób uruchomienia aplikacji.

Zalecenia

Analiza problemu doprowadziła do powstania pewnych zaleceń implementowanego systemu. Celem jest wykorzystanie zalet innych systemów, a także eliminacja ich wad. Najważniejsze zalecenia spisano poniżej:

- system ma być prosty i niezależny od innego oprogramowania (samodzielny)
- bardzo istotna jest prezentacja zestawienia materiałowego w postaci hierarchii, a także pokazanie szczegółów poszczególnych wyrobów.
- program powinien posiadać wygodny i intuicyjny interfejs użytkownika
- możliwość dodawania, usuwania i edycji poszczególnych pozycji zestawienia

Założenia

Na podstawie założeń zdecydowano się na implementację systemu przy użyciu języka programowania Python i wbudowanej biblioteki do obsługi interfejsu graficznego o nazwie tkinter. Python został wybrany jako przyszłościowy język wysokiego poziomu o wielorakich zastosowaniach. Biblioteka graficzna ułatwia proces tworzenia interfejsu graficznego i pozwala się skupić na faktycznej realizacji. Przy pomocy tych dwóch narzędzi można zorganizować niemal cały wymagany system.

Istotnym zagadnieniem jest jednak przechowywanie danych, ponieważ bez tego zestawienie istniałoby tylko podczas działania programu (przechowywane w pamięci). W celu uzyskania pełnej niezależności postawiono na przechowywanie zestawień w lokalnych plikach komputera w formacie JSON. Jest to tekstowy format zapisu danych, który już sam w sobie jest hierarchiczny.

Ważna jest także walidacja danych wejściowych (plików i interaktywnych pól tekstowych), a także obsługa błędów.

Opis

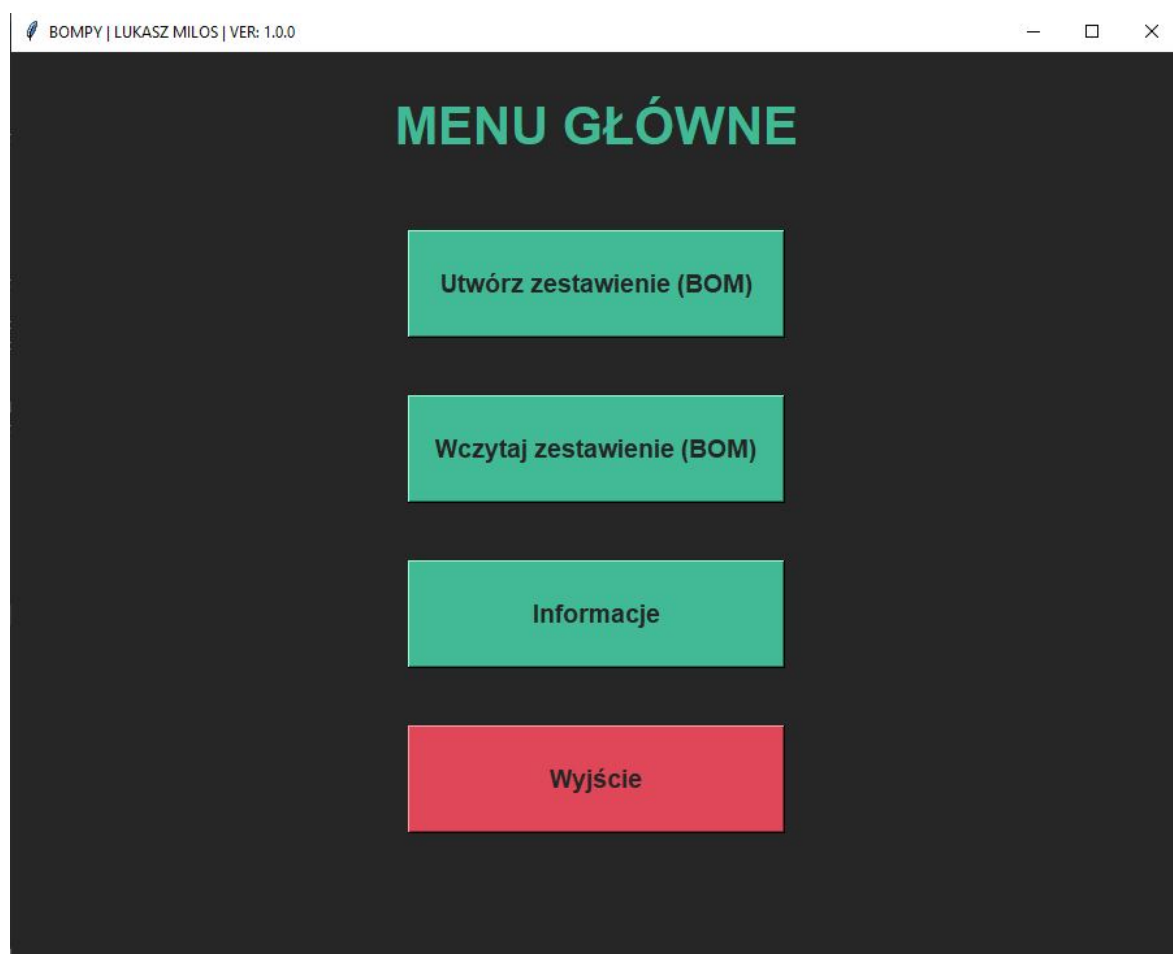
Zgodnie z założeniami zrealizowano system przy użyciu wybranych technologii. Warto zwrócić uwagę na samodzielność realizacji poszczególnych zagadnień, a także samą stylistykę kodu, wykorzystującą programowanie zorientowane obiektowo.

W aplikacji zadbano o minimalistyczny, modernistyczny, responsywny interfejs użytkownika, a także obsługę wszelkich błędów.

Aplikacja składa się z kilku stron i wielu okien modalnych, wyświetlanych w zależności od kontekstu, co czyni ją bardzo przejrzystą.

Menu główne

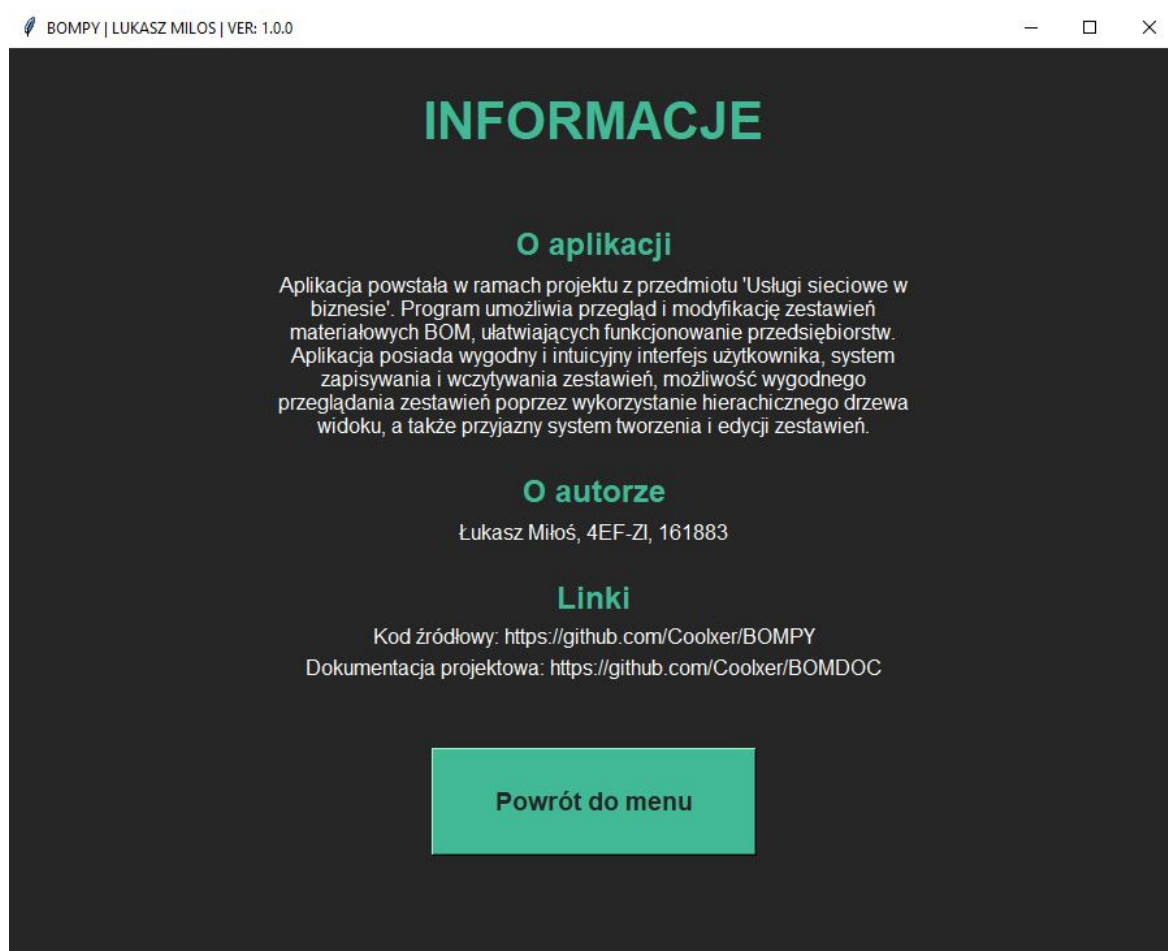
Na zdjęciu (3.5) widoczne jest menu główne aplikacji. Z poziomu menu głównego możliwe jest zamknięcie aplikacji, wyświetlenie strony informacyjnej, wczytanie zestawienia BOM z pliku lub utworzenie nowego zestawienia BOM.



Rysunek 3.5: Menu główne aplikacji

Strona informacyjna

Na zdjęciu (3.6) przedstawiono stronę informacyjną zawierającą krótką charakterystykę programu, autora i linki do repozytoriów.



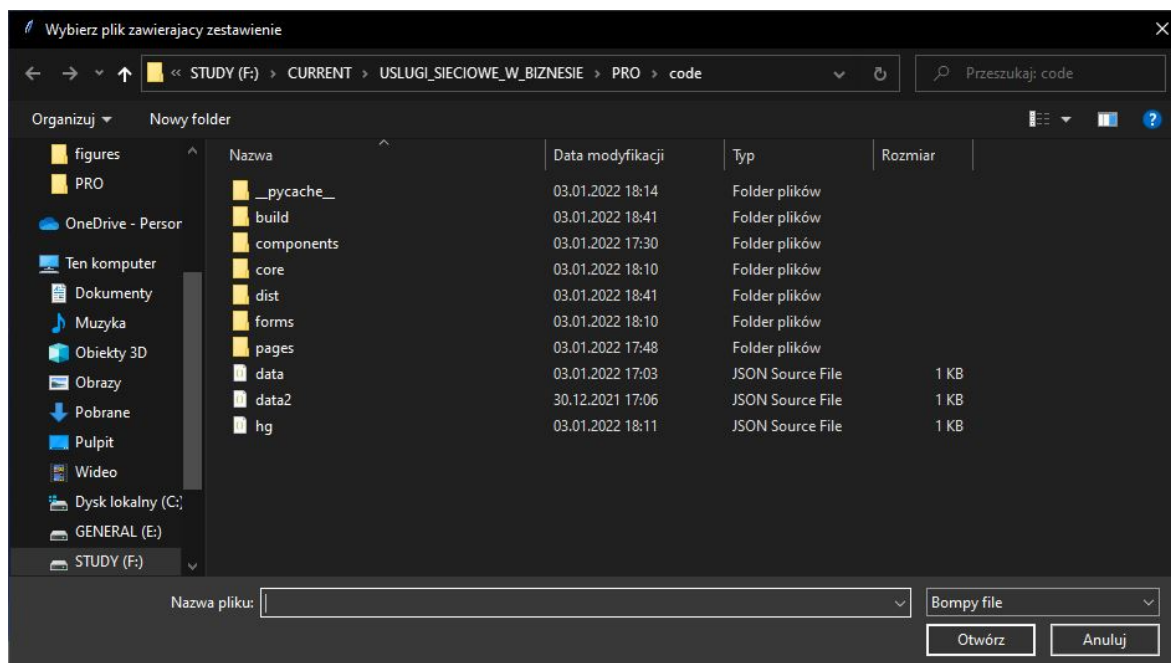
Rysunek 3.6: Strona informacyjna aplikacji

Wczytanie zestawienia BOM

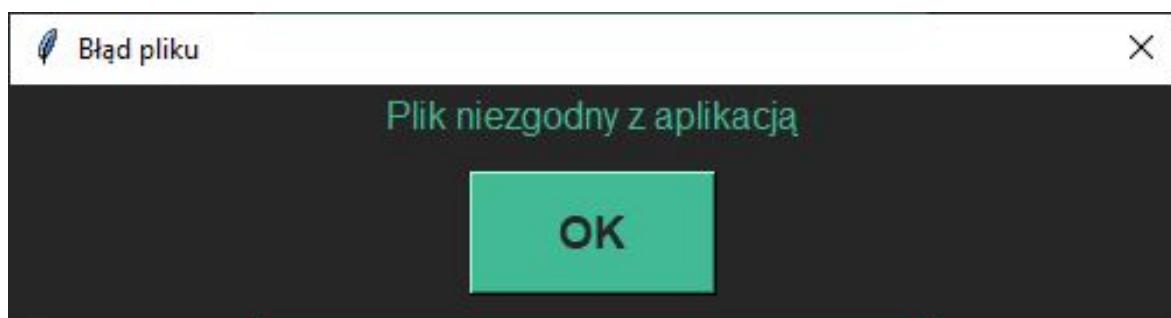
W kwestii wczytywania zestawienia z pliku o formacie *.json warto zwrócić uwagę na walidację poprawności pliku. Plik programowy musi być odpowiedni, tzn. przeznaczony dla tej aplikacji (odpowiedni klucz i wartość).

W celu wczytywania pliku wykorzystano dostępne okno dialogowe (3.7), które odpowiednio skonfigurowano, tak aby możliwe do wyboru były jedynie pliki o rozszerzeniu *.json.

Jeśli stwierdzona zostanie niepoprawność pliku źródłowego, zostanie wyświetlony odpowiedni komunikat (3.8).



Rysunek 3.7: Wybór pliku zestawienia

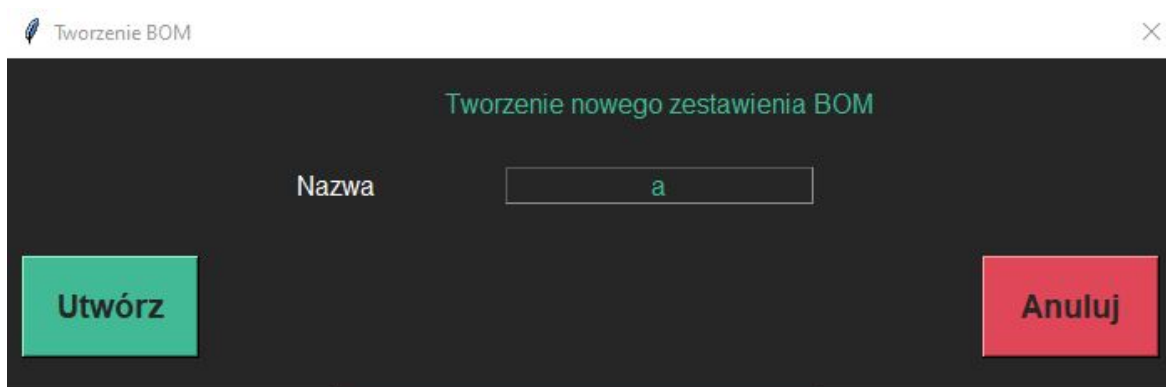


Rysunek 3.8: Błędny plik zestawienia

Tworzenie nowego zestawienia BOM

Po wybraniu opcji «Utwórz zestawienie (BOM)» z menu głównego, pojawia się okno dialogowe (3.9), w którym należy podać nazwę zestawienia.

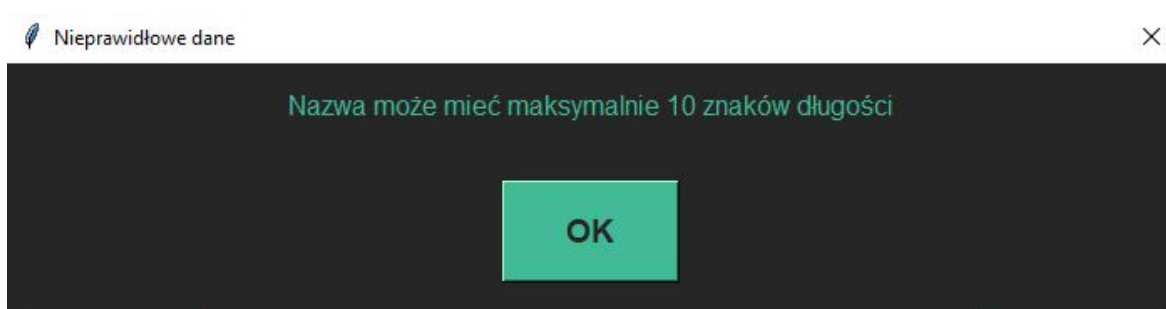
Nazwa zestawienia musi spełniać kryterium minimum 3 znaków i maksymalnie 10 znaków. W przeciwnym razie zostanie wyświetlony stosowny komunikat (3.10) lub (3.11)



Rysunek 3.9: Tworzenie nowego zestawienia

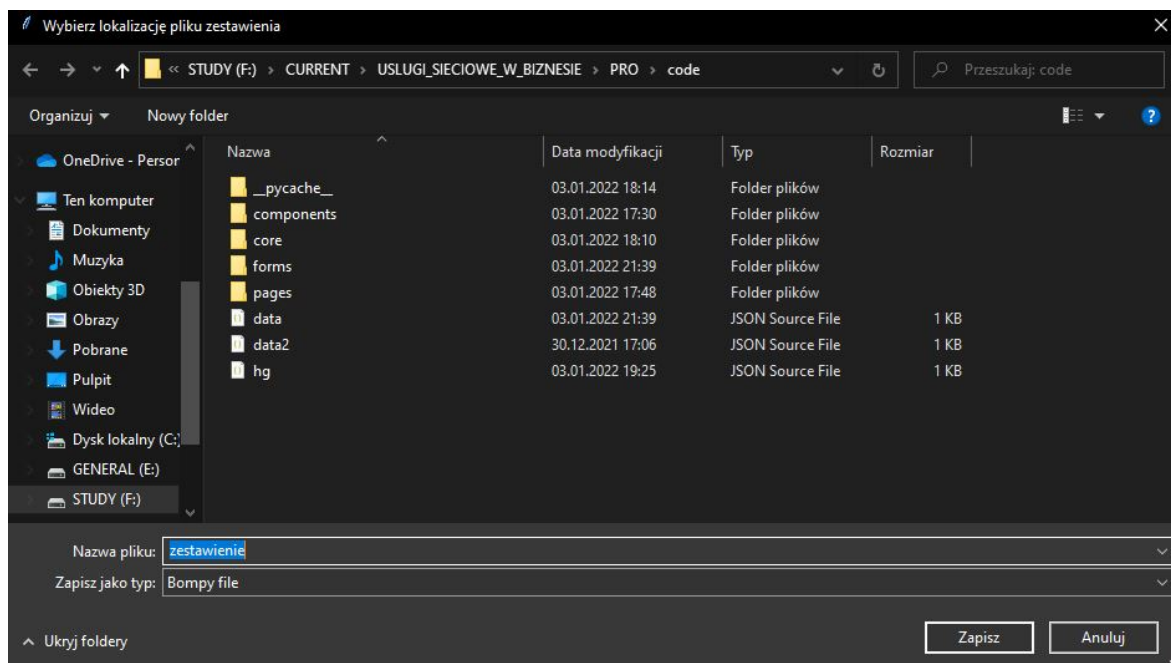


Rysunek 3.10: Błąd zbyt krótkiej nazwy zestawienia



Rysunek 3.11: Błąd zbyt długiej nazwy zestawienia

Jeśli podana nazwa jest prawidłowa użytkownik przy użyciu stosownego okna dialogowego (3.12) wybiera nazwę i lokalizację pliku, w którym jego zestawienie będzie na bieżąco zapisywane.



Rysunek 3.12: Wybór nazwy i lokalizacji pliku nowego zestawienia

Przegląd i modyfikacja zestawienia

Po pomyślnym wczytaniu zestawienia lub utworzeniu nowego zestawienia użytkownik zostaje przekierowany na stronę prezentującą zestawienie (3.13).

Zestawienie prezentowane jest w specjalnie przygotowanym widoku, umożliwiającym również jego modyfikację.

Na samej górze prezentowana jest nazwa zestawienia, zazwyczaj oznaczającą nazwę produktu końcowego.

Prezentowany widok drzewa składa się z 6 kolumn. Pierwsza kolumna stanowi identyfikator półproduktu, a następne oznaczają kolejno nazwę, ilość, jednostkę (kg., szt., m), cenę jednostkową i sumę (ilość * cena jednostkowa). Użytkownik nie wpisuje bezpośrednio sumy, ponieważ jest ona obliczana automatycznie na podstawie ilości i ceny jednostkowej.

Widok drzewa jest strukturą hierarchiczną i posiada możliwość zwijania i rozwijania poszczególnych wierszy, co sprawia, że przeglądanie zestawienia jest bardzo wygodne i praktyczne. Na samej górze listy znajduje się pusty wiersz, służący do dodawania komponentów składowych dla najwyższego poziomu hierarchii.

Po prawej stronie widoczny jest natomiast pasek przewijania listy. Użytkownik może także przewijać listę przy pomocy scroll'a myszki komputerowej.

BOMPY | LUKASZ MIŁOS | VER: 1.0.0

ROWER

	Nazwa	Ilość	Jednostka	Cena jedn.	Suma
▼ 000001	rama	1	szt.	210.45	210.45
000011	łańcuch	1	szt.	32.5	32.5
000012	pedał	2	szt.	15	30.0
000013	zębatka	5	szt.	19.99	99.95
000014	tarcza	2	szt.	24	48.0
▼ 000002	koło	2	szt.	250	500.0
000021	szprycha	80	szt.	1.99	159.2
000022	opona	2	szt.	24.2	48.4
000023	piasta	2	szt.	50	100.0

Dodaj

Edytuj

Usuń

Menu

Rysunek 3.13: Prezentacja przykładowego zestawienia

Na dole znajduje się panel sterowania, przy pomocy którego można dodać, edytować lub usunąć komponent, a także powrócić do menu głównego. Poszczególne przyciski modyfikacji są aktywne tylko wtedy, gdy wybrany został dowolny wiersz. Zaznaczony wiersz jest wyróżniony kolorem i powiększoną czcionką. Zaznaczenie wiersza jest konieczne nie tylko w przypadku chęci usunięcia czy edycji, ale także do dodawania nowego komponentu, ponieważ zaznaczony wiersz determinuje miejsce wstawienia nowego elementu.

Dodawanie obiektu

Na zdjęciu (3.14) przedstawiono przykład dodawania obiektu przy użyciu specjalnie zaprojektowanego formularza. Użytkownik wypełnia poszczególne pola edycyjne. W przypadku wyboru jednostki komponentu zastosowano listę wyboru. Dla każdego interaktywnego pola tekstowego prowadzona jest walidacja poprawności (minimalna/maksymalna długość, typ tekstowy/numeryczny), a dla identyfikatora także unikatowość. W przypadku niespełnienia wymogów wyświetlany jest stosowny komunikat.

Dodawanie obiektu

Identyfikator: 000131

Nazwa: łożysko

Ilość: 5

Jednostka: szt.

Cena jedn.: 8.50

Dodaj **Anuluj**

Rysunek 3.14: Dodawanie obiektu

Edytowanie obiektu

Na zdjęciu (3.15) pokazano edytowanie obiektu za pomocą formularza. Wszelkie parametry są ładowane według bieżących danych, a użytkownik może modyfikować wybrane pola. Tutaj oczywiście również jest walidacja poprawności wpisywanych danych, a także sprawdzenie integralności identyfikatora, przy czym identyfikator może pozostać niezmieniony.

Edytowanie obiektu

Identyfikator: 000131

Nazwa: łożysko

Ilość: 5

Jednostka: szt.

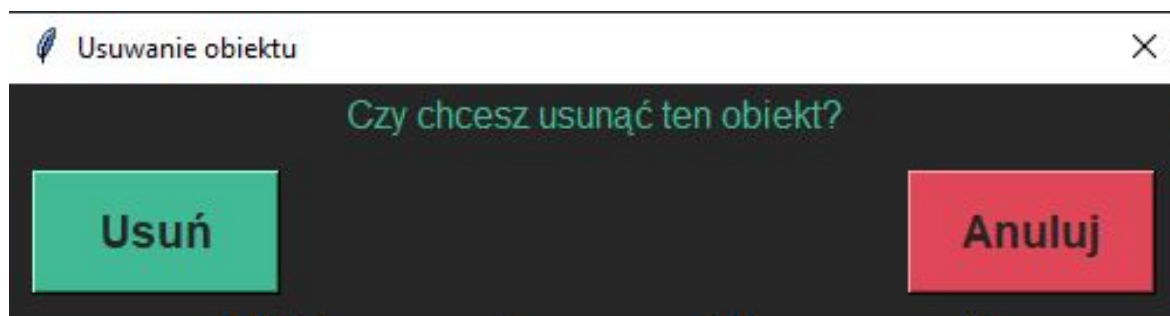
Cena jedn.: 8.5

Zapisz **Anuluj**

Rysunek 3.15: Edytowanie obiektu

Usuwanie obiektu

Usuwanie wybranego obiektu przedstawiono na zdjęciu (3.16). Użytkownik musi potwierdzić czy żąda usunięcia danego obiektu, ponieważ zmiany są nieodwracalne.



Rysunek 3.16: Usuwanie obiektu

Wszelkie modyfikacje (dodawanie, edytowanie, usuwanie obiektów) powodują zaktualizowanie wyświetlanej listy i automatyczny zapis nowego zestawienia do pliku.

Przy użyciu wspomnianych narzędzi użytkownik może w pełni kontrolować zestawienie materiałowe.

Uruchomienie

W celu uruchomienia aplikacji należy ją pobrać ze zdalnego repozytorium <https://github.com/Coolxer/BOMPY> i rozpakować w dowolnym miejscu.

Do uruchomienia aplikacji niezbędne jest posiadanie interpretera Python i wbudowanej biblioteki tkinter (instalowana automatycznie wraz z interpreterem o ile nie odznaczono tej opcji).

W katalogu głównym aplikacji znajduje się plik **main.py**, który należy uruchomić przy pomocy interpretera.

Jeśli wszystko wykonano prawidłowo na ekranie powinno się wyświetlić okno graficzne aplikacji, w przeciwnym wypadku należy rozwiązać wyświetlone błędy.

4. Podsumowanie

W ramach projektu stworzono pełnoprawną aplikację desktop’ową do zarządzania bazą BOM. Zagadnienie BOM (Bill Of Materials) jest istotne z perspektywy wielu przedsiębiorstw, stąd zainteresowanie tą tematyką.

Realizację tematu rozpoczęto od analizy zagadnienia i sporządzenia zaleceń programu. Następnie podjęto pewne założenia dotyczące wyboru technologii i najważniejszych kwestii programowych.

Wybór Python’a jako języka wiodącego projektu był bardzo trafnym rozwiązaniem, ponieważ jest on bardzo intuicyjny, a wiedza i doświadczenie zdobyte podczas realizacji tego projektu z pewnością zaowocuje w przyszłości, zwłaszcza, że python rośnie w popularności i mnogości zastosowań.

Wybrana biblioteka graficzna tkinter okazała się być dość trudną w zastosowaniu, ponieważ wiele elementów należało wykonać samodzielnie. Jednak pozostałe alternatywy nie są wbudowane w interpreter i/lub wymagają płatnej licencji.

Wybór formatu *.json był oczywistą kwestią ze względu na jego hierarchiczność, a także łatwość obsługi i popularność w środowisku programistycznym.

Podczas projektowania aplikacji zadbano o modernistyczny i responsywny interfejs użytkownika, a także przejrzystość prezentowania wszelkich informacji. Dużą uwagę skupiono na estetyce kodu, wykorzystując paradygmat programowania obiektowego, a także wzbogacając kod źródłowy komentarzami.

Zdobyte w przeszłości doświadczenia spowodowały zaimplementowanie wielu walidacji danych wejściowych, a także obsługę znacznej ilości błędów.

Aplikacja umożliwia pełny przepływ informacji, rozpoczynający się od wczytania/utworzenia pliku, modyfikacji zawartości i zapisu pliku.

Esencją projektu jest wykorzystanie drzewa widoku do prezentacji zestawienia, możliwość zwijania i rozwijania poszczególnych poziomów zagnieżdżenia. Zaimplementowano także funkcje dodawania, edycji i usuwania komponentów. Do wspomnianych celów wykorzystano specjalnie zaprojektowane formularze wzbogacone o walidację danych.

W realizacji projektu wykorzystano system kontroli wersji GIT, a kod programu umieszczono na zdalnym repozytorium Github pod linkiem <https://github.com/Coolxer/BOMPY>.

Literatura

[1] <https://docs.python.org/3/>

[2] <https://docs.python.org/3/library/tk.html>

[3] <https://www.mecalux.pl/blog/zestawienie-materialowe-bom>

[4] <https://www.unleashedsoftware.com/blog/everything-you-need-to-know-about-bill-of-materials>