

INSTYTUT AUTOMATYKI, ROBOTYKI I INŻYNIERII
INFORMATYCZNEJ
WYDZIAŁ ELEKTRYCZNY
POLITECHNIKA POZNAŃSKA

INŻYNIERSKA PRACA DYPLOMOWA

**APLIKACJA WSPOMAGAJĄCA WYBRANE
FUNKCJE SYSTEMU FINANSOWO-
KSIĘGOWEGO**

Hubert Krzepkowski

Promotor:
dr hab. Tadeusz Pankowski

Poznań, 2019

Karta tematu pracy

Streszczenie

Celem pracy było zaprojektowanie oraz implementacja aplikacji realizującej wybrane funkcje systemu księgowego. W pracy omówiono problematykę tematu oraz przedstawiono dostępne na rynku rozwiązania. Omówiono technologie wykorzystane przy tworzeniu pracy oraz zdefiniowano wymagania, jakie spełniać musi aplikacja. Przedstawiono projekt aplikacji oraz opisano proces implementacji i sposób korzystania z aplikacji. W pracy została także zawarta opinia użytkownika docelowego na temat aplikacji oraz podsumowanie zawierające opis wymagań, które zostały spełnione.

Abstract

The purpose of the work was to design and implement an application implementing selected functions of the accounting system. The work discusses the topic issues and presents solutions available on the market. The technologies used to create the application were discussed and the requirements that the application must meet were defined. Application design is presented. The process of implementation and the way of using the application are described. The work also included the target user's opinion on the application and a summary including a description of the requirements that have been met.

SPIS TREŚCI

1.	Wprowadzenie	9
1.1.	Problematyka tematu i analiza istniejących rozwiązań	9
1.2.	Cel i zakres pracy	9
1.3.	Struktura pracy	10
2.	Wykorzystywane technologie	11
2.1.	Technologie serwera i bazy danych	11
2.2.	Technologie aplikacji klienckiej	12
3.	Założenia i sformułowanie wymagań	13
3.1.	Założenia	13
3.2.	Wymagania funkcjonalne	14
3.3.	Wymagania pozafunkcjonalne	14
4.	Dokumentacja projektowa	15
4.1.	Architektura systemu	15
4.2.	Projekt bazy danych	15
4.3.	Projekt aplikacji klienckiej	22
5.	Dokumentacja programowa	23
5.1.	Baza danych	23
5.2.	Aplikacja kliencka	27
6.	Testy w środowisku użytkowym i opinia użytkownika	36
7.	Dokumentacja użytkowa	37
7.1.	Instrukcja obsługi aplikacji	37
8.	Uwagi końcowe	42
8.1.	Wnioski	42
8.2.	Perspektywy i planowany rozwój systemu	42
	Literatura	43

1. Wprowadzenie

1.1. Problematyka tematu i analiza istniejących rozwiązań

Księgowość jest to zapisywanie zdarzeń gospodarczych. Do zdarzeń gospodarczych należą między innymi takie zdarzenia jak: zakup towaru, sprzedaż towaru, otrzymanie zapłaty za towar, wypłata wynagrodzenia pracownikowi. Sposób zapisu tych zdarzeń określają standardy np. Międzynarodowe Standardy Rachunkowości, a większość aspektów związanych ze sposobem prowadzenia księgowości jest uregulowana prawnie w Ustawie o Rachunkowości. Najważniejsze cele prowadzenia księgowości to kontrola finansowego stanu firmy i uzyskanych wyników oraz sprawozdawczość finansowa, której celem jest dostarczenie do urzędu wymaganych informacji o stanie firmy. Księgowość można podzielić na księgowość uproszczoną oraz księgowość pełną, którą są zobowiązane prowadzić spółki prawa handlowego np. spółka z ograniczoną odpowiedzialnością. Niepoprawnie prowadzona księgowość może prowadzić do błędnego rozliczenia podatków, a więc przestępstwa skarbowego, które karane jest karą grzywny.

Na rynku istnieje wiele dostępnych aplikacji do obsługi pełnej księgowości np. Sage Symfonia 2.0 Finanse i Księgowość, Comarch ERP Optima. Aplikacje te oferują wszystkie potrzebne funkcje do obsługi pełnej księgowości oraz wiele funkcji dodatkowych służących do informowania o stanie firmy w postaci statystyk i wskaźników. Aplikacje te są z powodzeniem używane w niewielkich rozmiarów firmach, a także dużych korporacjach. Wadą tych programów jest ich rozbudowanie, które powoduje, że są one nieintuicyjne w obsłudze dla nowych użytkowników i wymagają nauki ich użytkowania. Dostęp do oprogramowania można zakupić jednorazowo, uzyskując licencję wieczystą lub na okres 12 miesięcy w formie subskrypcji. Cena tych licencji/subskrypcji jest na tyle wysoka, że opłacalne jest stworzenie własnego programu, który będzie posiadał tylko te funkcje, które są potrzebne w firmie, co pozwoli uzyskać dużą przejrzystość programu i łatwość w obsłudze.

1.2. Cel i zakres pracy

Celem pracy jest przedstawienie projektu i implementacji aplikacji realizującej podstawowe operacje księgowe wykonywane w wybranym przedsiębiorstwie. Opracowaną aplikację nazwano KSH (Księga handlowa). Zadaniem aplikacji KSH jest

realizacja wybranych funkcji związanych z prowadzeniem pełnej księgowości w taki sposób, aby proces ten ułatwić, usprawnić, a także uodpornić na błędy powstałe na skutek ludzkich pomyłek. Główną funkcją aplikacji KSH jest generowanie części składowych sprawozdania finansowego takich jak bilans oraz rachunek zysków i strat. Realizację pracy podzielono na następujące zadania:

- Analiza zasad prowadzenia pełnej księgowości w stopniu wystarczającym do realizacji pracy.
- Zaprojektowanie struktury bazy danych oraz utworzenie bazy na serwerze.
- Implementacja procedur wykonywanych po stronie bazy danych.
- Zaprojektowanie oraz implementacja aplikacji klienckiej z interfejsem graficznym.
- Przetestowanie aplikacji przez użytkownika docelowego.

1.3. Struktura pracy

Dalsza część pracy zorganizowana jest w następujący sposób. W rozdziale drugim zostały wymienione i opisane technologie oraz narzędzia wykorzystane do realizacji pracy. Wymagania, jakie powinna spełniać aplikacja KSH oraz założenia, jakie przyjęto na potrzeby realizacji pracy, zostały przedstawione w rozdziale 3. Projekt bazy danych i aplikacji klienckiej oraz sposób komunikacji pomiędzy częściami składowymi aplikacji znajduje się w rozdziale 4. Proces wdrożenia bazy danych oraz implementacji aplikacji klienckiej wraz z najistotniejszymi fragmentami systemu przedstawiono w rozdziale 5. W rozdziale 6 zawarto opinię użytkownika testującego aplikację. Sposób korzystania z systemu KSH oraz wygląd interfejsu graficznego zamieszczono w rozdziale 7. Podsumowanie efektów pracy oraz perspektywy rozwoju aplikacji KSH omówiono w rozdziale 8.

2. Wykorzystywane technologie

2.1. Technologie serwera i bazy danych

2.1.1. Microsoft SQL Server oraz Microsoft SQL Server Management Studio 17

Microsoft SQL Server - Platforma bazodanowa działająca w architekturze klient - serwer stworzona i rozwijana przez firmę Microsoft. Rozwiązanie to jest dostępne w różnych edycjach także w edycji darmowej (Express), które charakteryzuje się ograniczoną funkcjonalnością w stosunku do płatnych. Jeden serwer SQL może zawierać wiele niezależnych od siebie instancji, które umożliwiają tworzenie i obsługę baz danych. Podstawowe elementy serwera to silnik zapytań T-SQL (Transact-SQL), którego zadaniem jest przetwarzanie zapytań SQL. Na serwerze przechowywane są bazy danych oparte na modelu relacyjnym.

Microsoft SQL Server Management Studio 17 - Program służący do zarządzania, monitorowania i konfigurowania MS SQL Server wyposażony w interfejs graficzny. Umożliwia on pisanie zapytań, procedur i wyzwalaczy w języku T-SQL za pomocą edytora skryptów. Management Studio pozwala na podgląd i modyfikację istniejących w bazie danych obiektów i danych, a także posiada funkcje wykonywania kopii zapasowych i przywracania bazy danych z kopii.

2.1.2. SQL z rozszerzeniem Transact-SQL

SQL (*Structured Query Language*) - Język służący do tworzenia i modyfikowania relacyjnych baz danych oraz tworzenia zapytań do bazy danych, które mają na celu pobieranie, usunięcie lub modyfikację danych. Składnia tego języka jest bardzo zbliżona do składni języka angielskiego, co ma na celu ułatwienie nauki i późniejszego korzystania z tego języka.

T-SQL (*Transact-SQL*) - Transakcyjny SQL to rozszerzenie języka SQL przeznaczone do pracy z relacyjnymi bazami danych, które pozwala na tworzenie wyzwalaczy, a także procedur składowanych z parametrami. Dzięki temu rozszerzeniu możemy także tworzyć zmienne, pętle i instrukcje warunkowe. T-SQL został stworzony przez firmę Sybase, a następnie wykupiony przez Microsoft.

2.2. Technologie aplikacji klienckiej

2.2.1. Język C#

C# - Jeden z najbardziej popularnych wysokopoziomowych języków programowania stworzony i rozwijany przez firmę Microsoft. Pomimo że jest językiem ogólnego zastosowania, najczęściej jest używany do tworzenia aplikacji desktopowych lub aplikacji internetowych. C# jest językiem silnie typowanym i realizującym wiele paradygmatów programowania np. obiektowość, programowanie imperatywne, programowanie sterowane zdarzeniami, programowanie funkcyjne. Język ten posiada zautomatyzowane zarządzanie pamięcią metodą odśmiecania pamięci. Składnia języka C# jest bardzo zbliżona do składni innych wysokopoziomowych języków takich jak: Java, C oraz C++.

2.2.2. WPF

WPF (*Windows Presentation Foundation*) - Schemat tworzenia graficznych interfejsów, który oddziela logikę aplikacji od kodu związanego z wyglądem aplikacji. Do opisu elementów interfejsu graficznego WPF używa języka XAML, który oparty jest na języku XML. WPF jest także silnikiem graficznym renderującym grafikę wektorową oraz zbiorem narzędzi, które pozwalają budować interfejs graficzny z gotowych elementów.

2.2.3. Material Design

Material Design dla WPF - Najbardziej popularny pakiet narzędzi pozwalający nadać nowoczesny wygląd graficznemu interfejsowi stworzonemu przy pomocy WPF. Pakiet ten zawiera gotowe elementy interfejsu wystylizowane według nowoczesnych standardów, a jego kod źródłowy jest publicznie dostępny.

2.2.4. Microsoft Visual Studio

Microsoft Visual Studio - zintegrowane środowisko programistyczne stworzone i rozwijane przez firmę Microsoft. Visual Studio jest używane do tworzenia programów konsolowych, aplikacji desktopowych, stron internetowych, aplikacji internetowych,

web-serwisów oraz aplikacji mobilnych. Środowisko wspiera 36 języków programowania a dla większości najpopularniejszych języków takich jak np. C# wsparcie jest wbudowane. Visual Studio jest dostępne w wersji darmowej Community oraz płatnych wersjach Professional i Enterprise.

3. Założenia i sformułowanie wymagań

3.1. Założenia

Tworzona w pracy implementacja księgowości bilansowej dotyczy konkretnego przypadku firm będących tzw. Małym podatnikiem i jednocześnie mikro firmą. Dla tego typu podmiotów dopuszczono sprawozdania finansowe w nieco uproszczonej formie.

Docelowym środowiskiem pracy aplikacji KSH jest stacja robocza nieposiadająca połączenia z internetem, wobec tego zostało założone, że program nie będzie narażony na zagrożenia zewnętrzne związane z bezpieczeństwem. Z aplikacji będzie korzystać tylko jeden użytkownik.

Założona struktura tabel związanych z księgowaniem została przedstawiona na rysunku 1. Struktura ta jest odzwierciedleniem struktury kont księgowych - każde konto posiada swoją tabelę w bazie danych. Założono także że suma rekordów w bazie danych nie będzie przekraczać 40000.

ID Syntetyki	Opis Konta	Typ Konta	Aktywne/Pasywne/Mieszane	Nazwa Tabeli Konta
10	Srodki trwałe	bilansowe	A	tb010_SrodkiTrwale
50	Umorzenie środków trwałych	bilansowe	A	tb050_umorzenieSrodkiTrwale
100	Kasa	bilansowe	A	tb100_Kasa
130	Rachunek bieżący	bilansowe	A	tb130_RachunekBiezacy
201	Rozrachunki z odbiorcami	rozrachunkowe	A	tb201_RozrachunkiOdbiorcy
202	Rozrachunki z dostawcami	rozrachunkowe	P	tb202_RozrachunkiDostawcy
221	Rozrachunki Podatek Dochodowy	rozrachunkowe	AP	tb221_RozrachunkiPodatekDochodowy
223	rozrachunki z US z tyt VAT	rozrachunkowe	AP	tb223_RozrachunkiVATurzaSkarb
224	VAT należny	bilansowe	P	tb224_VATNalezny
225	VAT naliczony	bilansowe	A	tb225_VATNaliczony
242	pozostałe rozrachunki	rozrachunkowe	P	tb242_PozostaleRozrachunki
247	Pozyczki otrzymane	bilansowe	P	tb247_PozyczkiOtrzymane
250	Rozrachunki z udziałowcami	rozrachunkowe	P	tb250_RozrachunkiUdzialowcy
330	Towary	bilansowe	A	tb330_Towary
410	Zuzycie materialow i energia	wynikowe	A	tb410_KosztyMaterEnergia
420	Usługi obce	wynikowe	A	tb420_KosztyUslugiObce
460	Wynagrodzenia	wynikowe	A	tb460_KosztyWynagrodzenia
730	Przychody ze sprzedaży	wynikowe	P	tb730_PrzychodySprzedaz
735	Koszt własny sprzedaży	wynikowe	P	tb735_KosztWlasnySprzedazy
760	Pozostałe przychody operacyjne	wynikowe	P	tb760_PozostalePrzychodyOperacyjne
761	Pozostałe koszty operacyjne	wynikowe	A	tb761_PozostaleKosztyOperacyjne
801	Kapitał zakładowy	bilansowe	P	tb801_KapitalZakladowy
805	kapitał rezerwowy	bilansowe	P	tb805_KapitalRezerwowy
860	Wynik finansowy	bilansowe	P	tb860_WynikFinansowy
871	Podatek Dochodowy i inne obc.	wynikowe	P	tb871_PodatekDochodIInneObcWyniku

Rysunek 1. Plan kont księgowych.

3.2. Wymagania funkcjonalne

System KSH w celu realizacji swojego głównego przeznaczenia, którym jest generowanie części składowych sprawozdania finansowego, będzie posiadać następujące funkcje:

- Księgowanie dokumentów według ich rodzaju. W zależności od rodzaju dokumentu dane muszą zostać zapisane w odpowiednich tabelach. Przed zapisaniem danych do niektórych tabel należy najpierw wykonać na tych danych odpowiednie operacje matematyczne.
- Usuwanie zaksięgowanego dokumentu. Aby zachować spójność bazy danych dane muszą zostać usunięte ze wszystkich tabel, w których znajdują się rekordy powiązane z dokumentem.
- Generowanie i wyświetlanie księgi głównej zawierającej dane z wszystkich tabel przechowujących dane księgowe.
- Generowanie i wyświetlanie bilansu.
- Generowanie i wyświetlanie zestawienia obrotów i sald (ZOiS).
- Generowanie i wyświetlanie rachunku zysków i strat (RZiS).
- Wyświetlanie planu kont.
- Wyświetlanie tabeli z dokumentami.

3.3. Wymagania pozafunkcjonalne

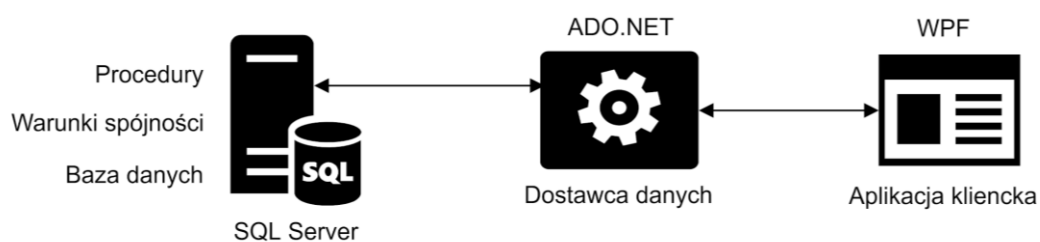
Aplikacja KSH musi także spełniać wymagania pozafunkcjonalne, które mają na celu zdefiniowanie cech jakościowych systemu. Do wymagań jakościowych należą:

- Poprawność działania aplikacji z punktu widzenia przepisów i standardów księgowość.
- Czas oczekiwania na pobranie danych z bazy danych nie dłuższy niż 10s.
- Interfejs aplikacji w języku polskim.

4. Dokumentacja projektowa

4.1. Architektura systemu

Aplikacja KSH składa się z aplikacji klienckiej pełniącej funkcje interfejsu użytkownika oraz bazy danych. Baza danych umieszczona będzie na lokalnym serwerze MS SQL Server w wersji Express. Z poziomu aplikacji klienckiej za pośrednictwem ADO.NET do bazy danych będą wysyłane zapytania SQL. Baza danych składa się z tabel spełniających warunki spójności oraz procedur składowanych. Komunikacja pomiędzy bazą danych a aplikacją kliencką jest realizowana przy pomocy protokołu pamięci współdzielonej.



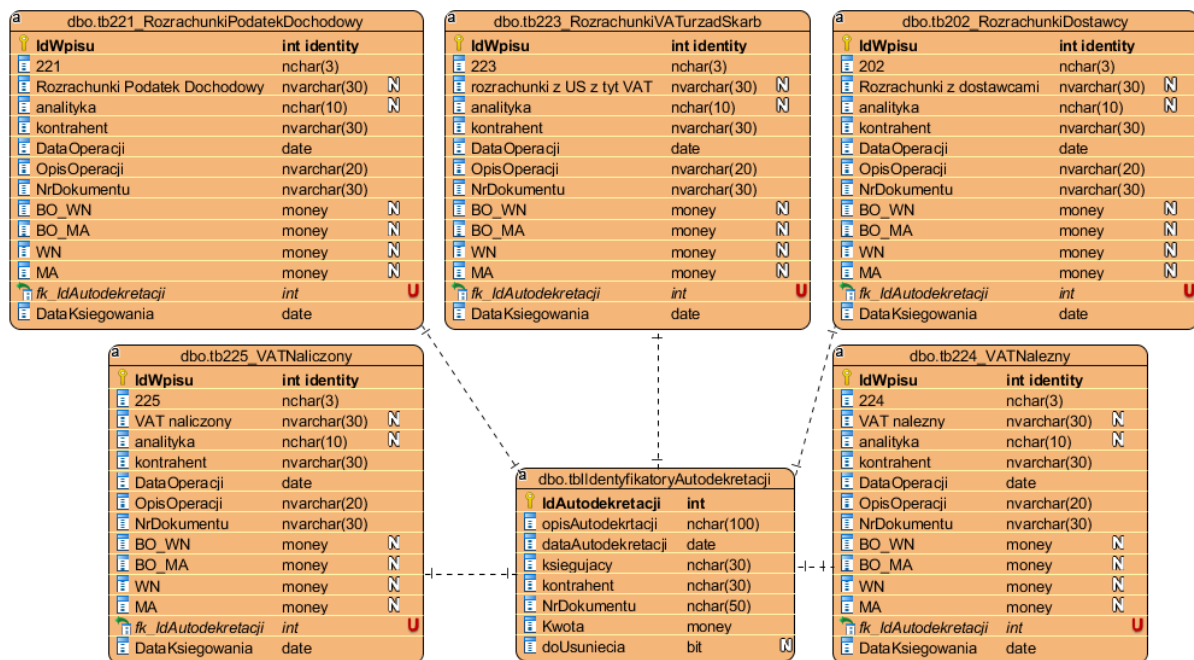
Rysunek 2. Diagram przedstawiający komunikację pomiędzy częściami składowymi systemu.

4.2. Projekt bazy danych

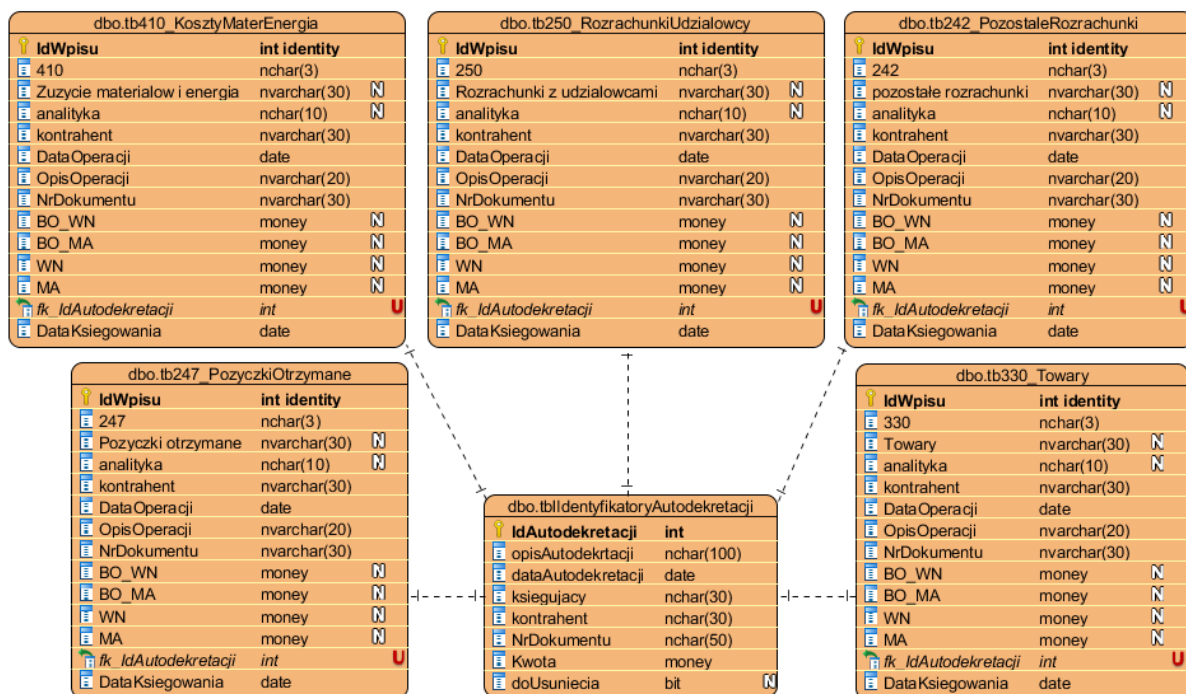
Po zapoznaniu się ze strukturą planu kont oraz wymaganiami funkcjonalnymi utworzono diagram związków encji ERD zawierający atrybuty obiektów, klucze główne poszczególnych tabel, klucze obce, relacje pomiędzy tabelami. Diagram został przedstawiony na rysunkach 3,4,5,6,7 i 8.



Rysunek 3. Diagram ERD bazy danych cz.1.



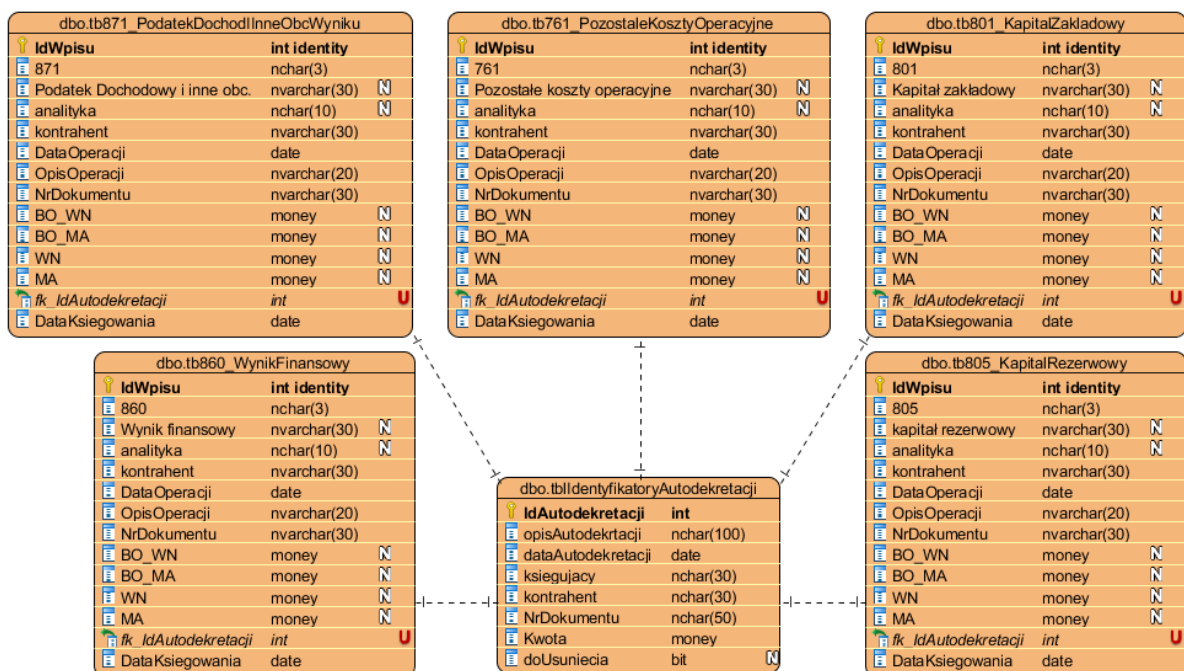
Rysunek 4. Diagram ERD bazy danych cz.2.



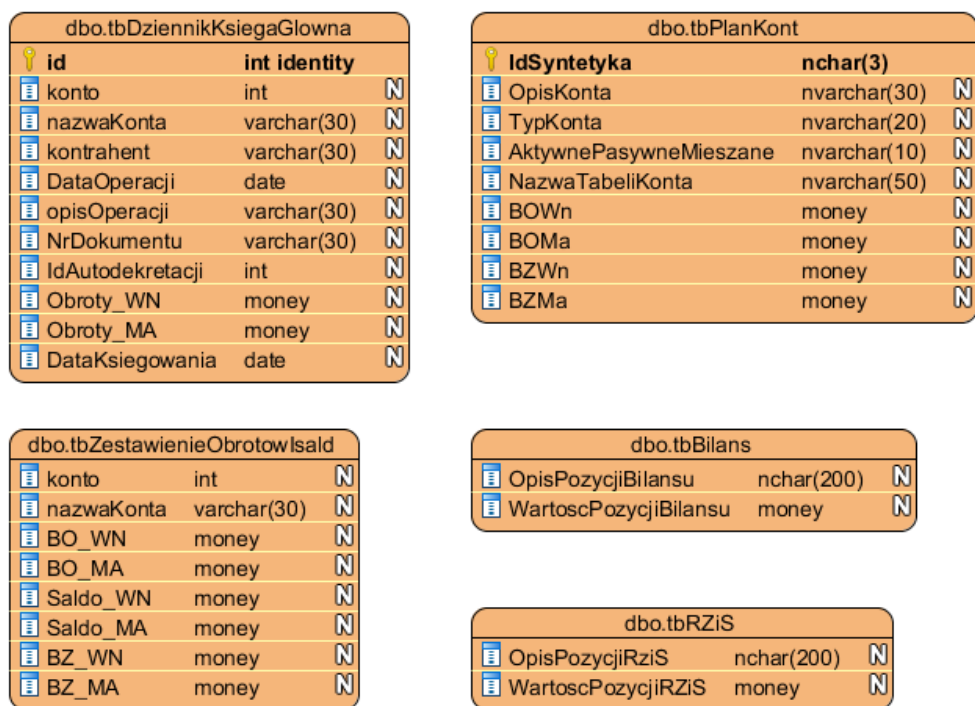
Rysunek 5. Diagram ERD bazy danych cz.3.



Rysunek 6. Diagram ERD bazy danych cz.4.



Rysunek 7. Diagram ERD bazy danych cz.5.



Rysunek 8. Diagram ERD bazy danych cz.6.

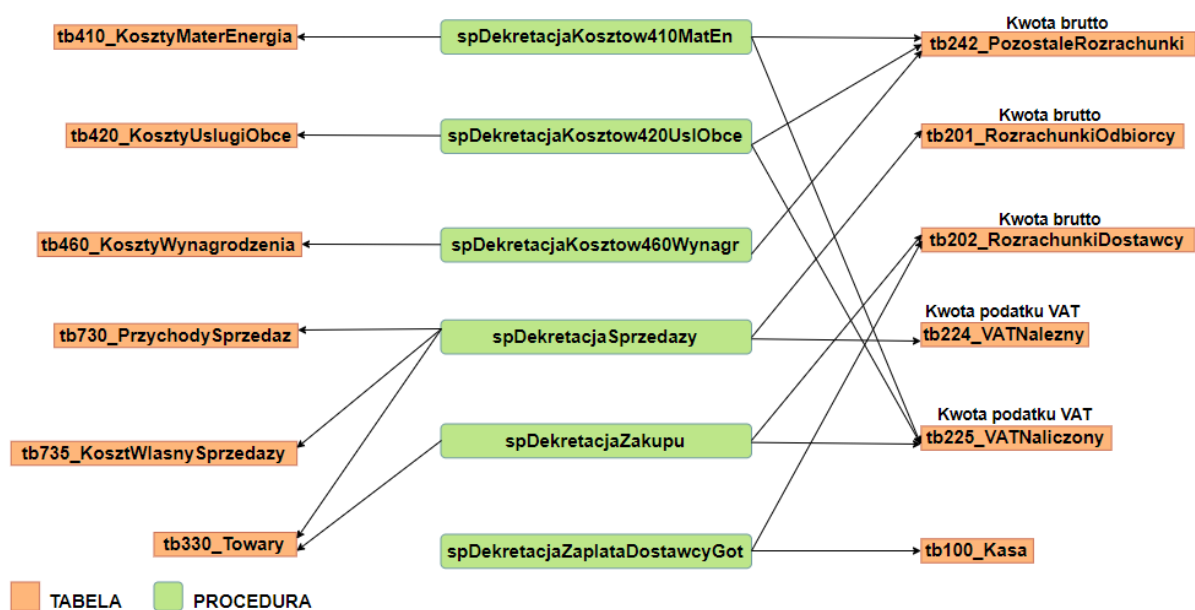
Wszystkie tabele przechowujące dane przedstawione na rysunkach 3,4,5,6, i 7 są w relacji typu jeden do jeden z tabelą **dbo.tblIdentyfikatoryAutodekretacji**, która przechowuje informacje o zaksięgowanych dokumentach. Usunięcie rekordu z tabeli **dbo.tblIdentyfikatoryAutodekretacji** powoduje usunięcie wszystkich rekordów powiązanych.

Zapewniając jednocześnie, że usunięcie rekordu nie spowoduje utraty spójności informacji w bazie danych. Spójność bazy danych zapewnia także unikalny klucz obcy `fk_IdAutodekretacji`, który gwarantuje, że w danej tabeli nie pojawi się więcej niż jeden rekord powiązany z dokumentem z tabeli `dbo.tblIdentyfikatoryAutodekretacji`. Na rysunku 8 zostały przedstawione tabele pomocnicze, które są używane przez procedury w celu drukowania wyników.

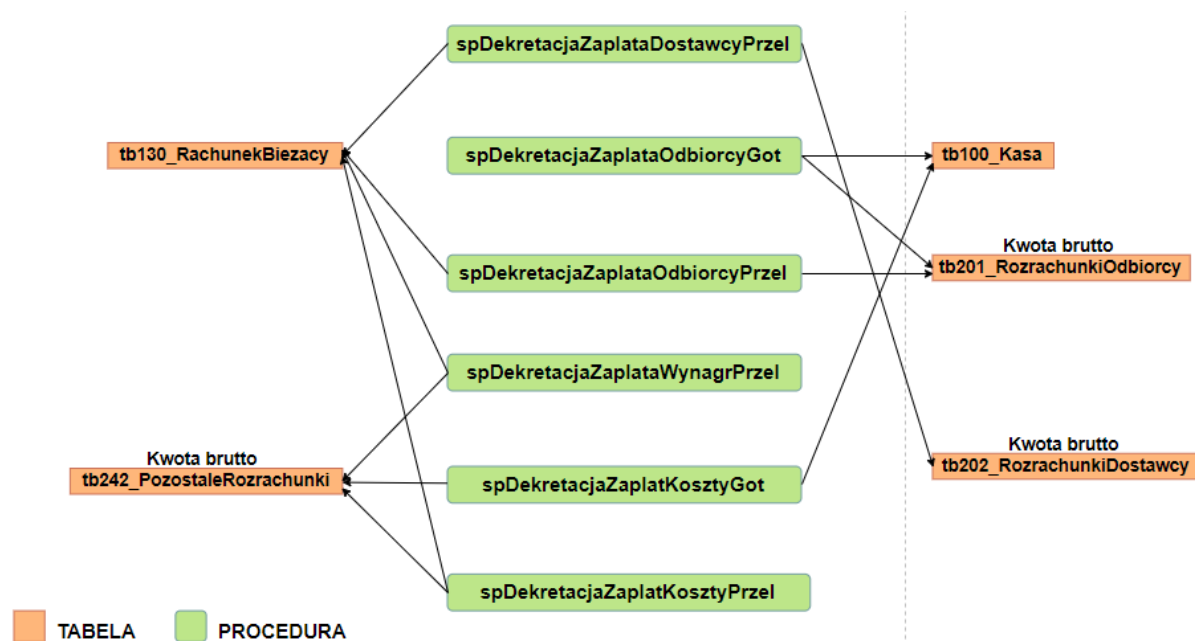
Procedury wykonywane przez bazę danych podzielono na procedury dekretacyjne, których celem jest wstawienie danych do odpowiednich tabel oraz procedury generujące zestawienia podsumowujące, których celem jest pobranie i przetworzenie danych z odpowiednich tabel w celu stworzenia bilansu, ZOiS, dziennik - główna oraz RZiS.

Dekretację wykonywane są jako pełne odzwierciedlenie zdefiniowanych operacji gospodarczych np. zakup towarów handlowych, sprzedaż, zapłata dostawcy przelewem, rozliczenia VAT itd. Zwalnia to osobę dokonującą księgowania z konieczności znajomości procedur księgowych dokonujących dekretacji na poszczególne konta, ponieważ może on posługiwać się językiem operacji gospodarczych, a nie księgowych co znacznie ułatwia i przyspiesza pracę oraz zabezpiecza przed popełnianiem błędów. Schemat dekretacji został przedstawiony na rysunku 9 i rysunku 10.

Zestawienia podsumowujące tworzone są według wzorców załączonych do ustawy o rachunkowości. Wzorce przedstawiono na rysunku 11 i rysunku 12.



Rysunek 9. Schemat dekretacji cz.1.



Rysunek 10. Schemat dekretacji cz.2.

(pieczęć jednostki)

BILANS sporządzony na dzień:

jednostka obliczeniowa: zł

Pozycja	Treść pozycji	Stan na	
AKTYWA			
A.	Aktywa trwałe		
I.	Wartości niematerialne i prawne		
II.	Rzeczowe aktywa trwałe		
III.	Należności długoterminowe		
IV.	Inwestycje długoterminowe		
V.	Długoterminowe rozliczenia międzyokresowe		
B.	Aktywa obrotowe		
I.	Zapasy		
II.	Należności krótkoterminowe		
III.	Inwestycje krótkoterminowe		
IV.	Krótkoterminowe rozliczenia międzyokresowe		
Aktywa razem			
PASYWA			
A.	Kapitał (fundusz) własny		
I.	Kapitał (fundusz) podstawowy		
II.	Należne wpłaty na kapitał podstawowy (wielkość ujemna)		
III.	Udziały (akcje) własne (wielkość ujemna)		
IV.	Kapitał (fundusz) zapasowy		
V.	Kapitał (fundusz) z aktualizacji wyceny		
VI.	Pozostałe kapitały (fundusze) rezerwowe		
VII.	Zysk (strata) z lat ubiegłych		
VIII.	Zysk (strata) netto		
IX.	Odpisy z zysku netto w ciągu roku obrotowego (wielkość ujemna)		
B.	Zobowiązania i rezerwy na zobowiązania		
I.	Rezerwy na zobowiązania		
II.	Zobowiązanie długoterminowe		
III.	Zobowiązania krótkoterminowe		
IV.	Rozliczenia międzyokresowe		
Pasywa razem			

Rysunek 11. Wzorzec arkusza bilansowego dla „Małego Podatnika”.

RACHUNEK ZYSKÓW I STRAT

sporządzony za okres

jednostka obliczeniowa:

Wiersz	Wyszczególnienie	Dane za rok	
A.	Przychody podstawowej działalności operacyjnej i zrównane z nimi, w tym:		
	- zmiana stanu produktów (zwiększenie - wartość dodatnia, zmniejszenie - wartość ujemna)		
B.	Koszty podstawowej działalności operacyjnej		
I.	Amortyzacja		
II.	Zużycie materiałów i energii		
III.	Wynagrodzenia, ubezpieczenia społeczne i inne świadczenia		
IV.	Pozostałe koszty		
C.	Pozostałe przychody i zyski, w tym:		
	- aktualizacja wartości aktywów		
D.	Pozostałe koszty i straty, w tym:		
	- aktualizacja wartości aktywów		
E.	Podatek dochodowy		
F.	Zysk/strata netto (A-B+C-D-E)		

Rysunek 12. Wzorec arkusza rachunku zysków i strat dla „Małego Podatnika”.

4.3. Projekt aplikacji klienckiej

Głównym zadaniem aplikacji klienckiej jest umożliwienie użytkownikowi interakcji z bazą danych. Interakcja ta może polegać na:

- Pobieraniu danych z bazy danych.
- Usunięciu danych z bazy danych.
- Dodaniu nowych danych do bazy danych.

Aplikacja kliencka musi wyświetlać wyniki tych interakcji. Wynik może przyjąć formę:

- Tabeli.
- Komunikatu potwierdzającego powodzenie usunięcia/zapisania danych.
- Komunikatu o ewentualnych błędach.

Szkielet interfejsu został przedstawiony na rysunku 9. Interfejs składa się z menu w postaci paska z przyciskami oraz z części zawierającej zawartość, która zmienia się z zależności od wybranego w menu przycisku. Zawartość może przyjąć formę tabeli, która zostanie pobrana z bazy danych za pomocą zapytania SQL lub formularza, który będzie sprawdzał poprawność wpisanych przez użytkownika danych. Jeżeli dane będą poprawne

to przy pomocy procedury składowanej w bazie danych, dane zostaną umieszczone w odpowiednich tabelach. Nie poprawne wpisanie danych do formularza będzie komunikowane poprzez zmianę koloru obramowania pola, które zawiera błędne dane na czerwony.



Rysunek 13. Szkielet interfejsu użytkownika.

5. Dokumentacja programowa

5.1. Baza danych

Baza danych została wdrożona na lokalnym serwerze MS SQL Server za pomocą skryptu SQL utworzonego na podstawie diagramu ERD. Skrypt ten utworzył wszystkie potrzebne tabele wraz z atrybutami danej tabeli, kluczami głównymi, kluczami obcymi oraz relacjami pomiędzy tabelami. Przykład skryptu tworzącego jedną z tabel kontowych przedstawiono na listingu 1.

```
CREATE TABLE [dbo].[tb202_RozrachunkiDostawcy](
    [IdWpisu] [int] IDENTITY(1,1) NOT NULL,
    [202] [nchar](3) NOT NULL,
    [Rozrachunki z dostawcami] [nvarchar](30) NULL,
    [analitka] [nchar](10) NULL,
```

```

[kontrahent] [nvarchar](30) NOT NULL,
[DataOperacji] [date] NOT NULL,
[OpisOperacji] [nvarchar](20) NOT NULL,
[NrDokumentu] [nvarchar](30) NOT NULL,
[BO_WN] [money] NULL,
[BO_MA] [money] NULL,
[WN] [money] NULL DEFAULT 0,
[MA] [money] NULL DEFAULT 0,
[fk_IdAutodekretacji] [int] UNIQUE NOT NULL REFERENCES
dbo.tblIdentyfikatoryAutodekretacji(IdAutodekretacji) on delete cascade,
[DataKsiegowania] [date] NOT NULL)

```

Listing 1. Skrypt tworzący tabele odwzorowującą konto księgowe.

Następnie utworzono tabele pomocnicze i dodano ograniczenie w tabeli tblIdentyfikatoryAutodekretacji zapobiegające dodaniu dwóch takich samych dokumentów. Skrypt dodający ograniczenie dla tabeli tblIdentyfikatoryAutodekretacji został przedstawiony na listingu 2.

```

ALTER TABLE tblIdentyfikatoryAutodekretacji
ADD CONSTRAINT uq_dokument UNIQUE(kontrahent, NrDokumentu);

```

Listing 2. Skrypt dodający ograniczenie dla tabeli tblIdentyfikatoryAutodekretacji .

Tabele pomocnicze:

- tbBilans
- tbDziennikKsiegaGlowna
- tblIdentyfikatoryAutodekretacji
- tbPlanKont
- tbRZiS
- tbSlownikOperacjiGospodarczych
- tbZestawienieObrotowIsald

Po wdrożeniu bazy danych zaimplementowano procedury. Procedury dekretacyjne posiadają podobną budowę i różnią się głównie zestawem tabel, do których wstawiają dane. Procedura dekretacyjna definiuje transakcję, która wstawia dane podane jako parametry do zbioru odpowiednich tabel. Zastosowanie transakcji jest konieczne, aby zabezpieczyć atomowość operacji oraz spójność danych w bazie danych. Jeżeli wykonanie transakcji zakończy się niepowodzeniem, stan bazy danych zostanie przywrócony do stanu sprzed transakcji.

Ponieważ dekretacja dokonuje jednocześnie zapisu na kilku kontach, konieczne jest zapisanie identyfikatora na każdym z kont jednoznacznie identyfikującego każdą wykonaną dekretację tak, aby można było wykonać w dowolnym momencie wycofanie całego zapisu. Mechanizmem wykorzystanym do tego celu jest integralność referencyjna z kaskadowym usuwaniem rekordów w tabeli podrzędnej. Usuwane są rekordy z tabeli `tblIdentyfikatoryAutodekretacji`, której klucz główny jest identyfikatorem zapisywanym w polu klucza obcego `fk_IdAutodekretacji` każdej tabeli kontowej, która bierze udział w zapisie dekretu. Przykład procedury dekretacyjnej został przedstawiony na listingu 3.

```
CREATE PROCEDURE [dbo].[spDekretacjaKosztow410MatEn]
@KosztNetto money,
@OpisOperacji nvarchar (20),
@DataOperacji date ,
@kontrahent varchar(50),
@ksiegujacy varchar(50) ,
@NrDokumentu varchar (50)

AS
    BEGIN TRY
    BEGIN TRAN

        declare @IdAutodkretacji int

        insert into tblIdentyfikatoryAutodekretacji (opisAutodekrtacji,
dataAutodekretacji, ksiegujacy, kontrahent, NrDokumentu, Kwota)

            values ('KosztyMatEn', getdate(), @ksiegujacy, @kontrahent, @NrDokumentu,
@KosztNetto)

        set @IdAutodkretacji = (select top(1) IdAutodekretacji from
tblIdentyfikatoryAutodekretacji order by IdAutodekretacji desc)
            insert into tb410_KosztyMaterEnergia([410], WN,
OpisOperacji, fk_IdAutodekretacji, DataOperacji, NrDokumentu, kontrahent,
DataKsiegowania)
                values ('410',@KosztNetto, @OpisOperacji, @IdAutodkretacji, @DataOperacji,
@NrDokumentu, @kontrahent, getdate())

            insert into tb225_VATNaliczony ([225], WN, OpisOperacji,
fk_IdAutodekretacji, DataOperacji, NrDokumentu, kontrahent, DataKsiegowania)
                values ('225',(@KosztNetto*0.23), @OpisOperacji, @IdAutodkretacji,
@DataOperacji, @NrDokumentu, @kontrahent, getdate())

            insert into tb242_PozostaleRozrachunki ([242], MA, OpisOperacji,
fk_IdAutodekretacji, DataOperacji, NrDokumentu, kontrahent, DataKsiegowania)
                values ('242', (@KosztNetto*1.23), @OpisOperacji, @IdAutodkretacji,
@DataOperacji, @NrDokumentu, @kontrahent, getdate())

    COMMIT TRAN
    END TRY
    BEGIN CATCH

        declare
```

```

@error int,
@message varchar(4000)

select @error = ERROR_NUMBER(), @message = ERROR_MESSAGE()
IF @error = 2627
Begin
    raiserror ('TAKI DOKUMENT ISTNIEJE JUŻ W BAZIE DANYCH: %d: %s', 16, 1,
@error, @message)
end
ELSE
    raiserror ('OPERACJA NIE POWIODŁA SIĘ: %d: %s', 16, 1, @error, @message)
ROLLBACK TRAN
END CATCH

```

Listing 3. Przykład procedury dekretującej koszty zdefiniowanej jako transakcja.

Działanie procedury dekretacyjnej składa się z następujących kroków:

- Deklaracja zmiennych.
- Wstawienie rekordu do tabeli tblIdentyfikatoryAutodekretacji w celu utworzenia nowego autonumeru w polu klucza głównego (IdAutodekretacji) tej tabeli.
- Rozpoczęcie transakcji.
- Pobranie najnowszego IdAutodekretacji z tabeli tblIdentyfikatoryAutodekretacji i przypisanie go do zmiennej.
- Wykonanie wymaganych operacji na danych wejściowych i wstawienie ich oraz IdAutodekretacji do odpowiednich tabel.
- Zakończenie transakcji lub przywrócenie stanu sprzed transakcji.

Procedury generujące zestawienia podsumowujące RZiS, bilans oraz ZOiS wykonują podstawowe działania matematyczne na wybranych danych z wybranych tabel, a następnie zwracają wygenerowane dane. Wybrane fragmenty procedury generującej rachunek zysków i strat zostały przedstawione na listingu 4.

```

CREATE PROCEDURE [dbo].[spRZiS]
AS
BEGIN
    delete from TbRZiS where 1=1
    declare
    @PrzychodyZeSprzedazy money

    set @PrzychodyZeSprzedazy=ISNULL((SELECT sum(MA) FROM
Tb730_PrzychodySprzedaz),0)

    INSERT INTO TbRZiS (OpisPozycjiRZiS, WartoscPozycjiRZiS) SELECT
'PrzychodyZeSprzedazy', @PrzychodyZeSprzedazy

    SELECT *

```

```

FROM TbRZiS
END

```

Listing 4. Fragmenty procedury generującej rachunek zysków i strat.

Działanie procedury generującej zestawienia podsumowujące składa się z następujących kroków:

- Czyszczenie danych z tabeli, do której procedura wstawia dane.
- Deklaracja zmiennych.
- Pobranie danych z odpowiednich tabel.
- Wykonanie operacji matematycznych na pobranych danych.
- Wstawienie obliczonych danych do odpowiedniej tabeli.
- Zwrócenie tabeli z obliczonymi danymi.

Zaimplementowano także dwie procedury wyszukujące dane z bazy danych. Procedurę [dbo].[spSzukajSzczegoly] która wyszukuje rekordy powiązane z danym dokumentem oraz procedurę [dbo].[spSzukajDekretacji] która pozwala na wyszukiwanie dokumentów o konkretnych parametrach. Kod procedury [dbo].[spSzukajDekretacji] został przedstawiony na listingu 5.

```

Select * from tblIdentyfikatoryAutodekretacji where
(IdAutodekretacji = @ID OR @ID IS NULL) AND
(opisAutodekrtacji = @Opis OR @Opis IS NULL) AND
(dataAutodekretacji BETWEEN @DataOd AND @DataDo OR @DataOd IS NULL AND @DataDo IS NULL)
AND (ksiegujacy = @Ksiegujacy OR @Ksiegujacy IS NULL) AND
(kontrahent = @Kontrahent OR @Kontrahent IS NULL) AND
(NrDokumentu = @NrDokumentu OR @NrDokumentu IS NULL) AND
(Kwota = @Kwota OR @Kwota IS NULL)
End

```

Listing 5. Kod procedury wyszukującej dokumenty z bazy danych.

5.2. Aplikacja kliencka

Graficzny interfejs użytkownika został stworzony przy użyciu WPF, w którym oddziela się kod odpowiedzialny za wygląd aplikacji od kodu odpowiedzialnego za funkcjonalność. Elementy wizualne aplikacji definiuje się w plikach XAML w postaci kontrolek WPF i ich atrybutów a funkcjonalności programuje się w języku C# w osobnych plikach zawierających klasy. WPF dostarcza zestaw gotowych

wystylizowanych kontroltek jednak, aby nadać kontrolką nowoczesny wygląd użyto pakietu Material Design, który modyfikuje wygląd kontroltek.

Głównym oknem aplikacji KSH jest MainWindow, które składa się z 3 elementów:

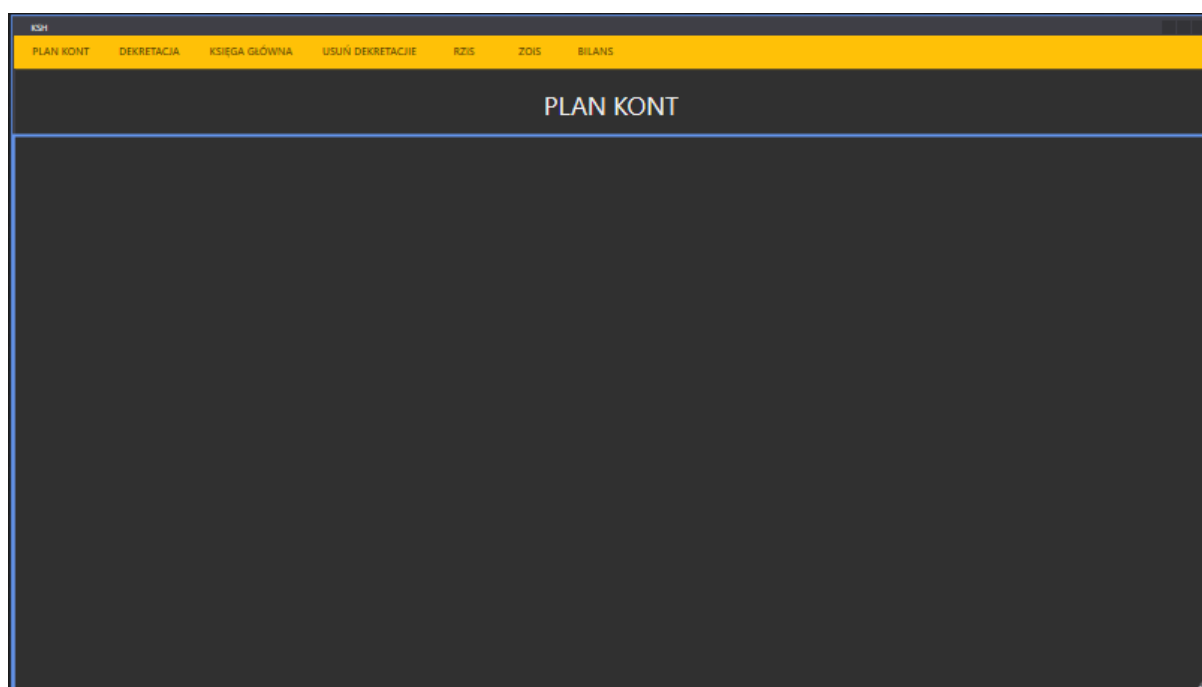
- Menu.
- Nagłówka strony.
- Kontrolki pozwalającej zmieniać wyświetlaną treść.

Fragment definicji elementów tworzących MainWindow w XAML został przedstawiony na listingu 6.

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto" />
        <RowDefinition Height="90*" />
        <RowDefinition Height="743*" />
    </Grid.RowDefinitions>
    <materialDesign:ColorZone Height="44" Mode="PrimaryMid">
        <StackPanel Margin="2" Orientation="Horizontal">
            <RadioButton
                Margin="4"
                Checked="PlanKont_Checked"
                Content="PLAN KONT"
                Style="{StaticResource MaterialDesignTabRadioButton}" />
            <RadioButton
                Margin="4"
                Checked="Dekretacja_Checked"
                Content="DEKRETACJA"
                Style="{StaticResource MaterialDesignTabRadioButton}" />
            <RadioButton
                Margin="4"
                Checked="Bilans_Checked"
                Content="BILANS"
                Style="{StaticResource MaterialDesignTabRadioButton}" />
        </StackPanel>
    </materialDesign:ColorZone>
    <TextBlock
        Name="Naglowek"
        Grid.Row="1"
        HorizontalAlignment="Center"
        Style="{StaticResource MaterialDesignDisplay1TextBlock}"
        Text="PLAN KONT" Margin="0,30,0,0" Width="Auto" />
    <ContentControl
        x:Name="ContentArea"
        Grid.Row="2"
        x:FieldModifier="public" />
</Grid>
```

Listing 6. Fragment definicji elementów tworzących MainWindow w XAML.

Menu składa się z kontrolki StackPanel, która zawiera w sobie przyciski. Nagłówek strony jest tekstem zmieniającym się w zależności od strony, na której jest użytkownik. Pozostały obszar okna jest przeznaczony na kontrolkę ContentControl, która pozwala zmieniać treść w zależności od strony, którą chce wyświetlić użytkownik. Treść ta jest definiowana przy pomocy UserControl kontrolki, którą może stworzyć programista. Kontrolka taka składa się z pliku o rozszerzeniu .xaml oraz pliku o rozszerzeniu .cs. Okno MainWindow zostało przedstawione na rysunku 14.



Rysunek 14. Główne okno aplikacji.

Kontrolki utworzone na potrzeby pracy:

- Bilans
- Dekretacja
- KsiegaGlowna
- PlanKont
- UsunDekretacje
- RZiS
- ZOiS

Kontrolki PlanKont, KsiegaGlowna, Bilans, RZiS, ZOiS posiadają bardzo podobną budowę a ich główną funkcją jest wyświetlanie danych w postaci tabeli. Kontrolki te

zbudowane są z jednej lub dwóch kontrolkek WPF DataGrid. Definicje kontrolki ZOiS w XAML przedstawiono na listingu 7.

```
<UserControl x:Class="KSHWP.ZOiS"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-
namespace:KSHWP"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="1*" />
            <RowDefinition Height="3*" />
        </Grid.RowDefinitions>

        <DataGrid
            Grid.Row="0"
            Name="PodsumowanieDataGrid"
            Margin="10"
            CanUserAddRows="False"
            CanUserSortColumns="True"
            ColumnWidth="Auto"
            IsReadOnly="True"
            SelectionMode="Extended"
            ClipboardCopyMode="IncludeHeader"/>

        <DataGrid
            Grid.Row="1"
            Name="ZOiSDataGrid"
            Margin="10"
            CanUserAddRows="False"
            CanUserSortColumns="True"
            ColumnWidth="Auto"
            IsReadOnly="True"
            SelectionMode="Extended"
            ClipboardCopyMode="IncludeHeader"/>

    </Grid>
</UserControl>
```

Listing 7. Definicja kontrolki ZOiS w XAML.

Kontrolka Dekretacja służy do wprowadzania danych, które mają zostać zapisane w bazie danych. Składa się ona z listy dekretacji oraz formularza do wprowadzania danych. Lista dekretacji jest kontrolką ListBox a poszczególne elementy listy to kontrolki TextBlock. Kliknięcie na element listy powoduje zmianę procedury, która zostanie użyta do zapisania danych. Formularz składa się z kontrolkek:

- TextBlock.

- TextBox.
- DatePicker.
- ComboBox.
- Button.

Po pomyślnym zapisaniu danych na dole ekranu pojawia się kontrolka Snackbar informująca o zapisaniu danych. Wybrane fragmenty definicji kontrolki Dekretacja w XAML przedstawiono na listingu 8.

```
<Grid
    Width="Auto"
    Margin="-25,0,0,0"
    HorizontalAlignment="Left">
    <ListBox
        Margin="27,10,654,0"
        IsEnabled="{Binding IsChecked, ElementName=EnableListBox}"
        SelectionChanged="ListBox_SelectionChanged">
        <TextBlock>
            DekretacjaKosztow410MatEn
        </TextBlock>
    </ListBox>
</Grid>

<Grid
    Width="530"
    Margin="0,10,68,15"
    HorizontalAlignment="Center"
    DataContext="{Binding RelativeSource={RelativeSource AncestorType={x:Type
UserController}}}">

</Grid.Resources>
<Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
</Grid.RowDefinitions>
<TextBox
    Name="opis"
    Grid.Row="7"
    Grid.Column="0"
    Margin="8,17,7,-9"
    materialDesign:HintAssist.Hint="Opis Operacji"
    FontSize="20"
    RenderTransformOrigin="0.548,-0.581"
    Text="" />
<DatePicker
    Name="data"
    Grid.Row="3"
    Grid.Column="0"
    Margin="8,5,7,7"
    materialDesign:HintAssist.Hint="Wybierz Datę"
    FontSize="20"
    Style="{StaticResource MaterialDesignFloatingHintDatePicker}" />
```

```

        <Button
            x:Name="Zapisz"
            Grid.Row="9"
            Width="120"
            Margin="49,88,49,-69"
            Click="Zapisz_Click"
            Content="Zapisz" />
    </Grid>
</Grid>
</UserControl>

```

Listing 8. Wybrane fragmenty definicji kontrolki Dekretacja w XAML.

Kontrolka UsunDekretacje umożliwia użytkownikowi:

- Wyszukiwanie dokumentów w bazie danych.
- Podgląd wszystkich rekordów powiązanych z danym dokumentem.
- Usuwanie dokumentów z bazy danych.

Wyszukiwanie elementów składa się z kontrolek:

- TextBox
- DatePicker
- Button

Pozostałe funkcje są realizowane przy pomocy kontrolki DataGrid, która posiada zagnieżdżoną kontrolkę DataGrid wyświetlającą powiązane z dokumentem rekordy po kliknięciu na wiersz zawierający dany dokument. Przyciski usuwania dokumentu są generowane przez funkcję generującą zawartość DataGrid. Definicja Kontrolki DataGrid w XAML została przedstawiona na listingu 9.

```

<DataGrid
    Name="IdDekretacjiDG"
    Grid.Row="1"
    Margin="10,0,10,10"
    CanUserAddRows="False"
    CanUserSortColumns="True"
    ColumnWidth="Auto"
    IsReadOnly="True"
    SelectionMode="Single">
    <DataGrid.RowDetailsTemplate>
        <DataTemplate>
            <Grid Background="#ffc107">
                <Button
                    Name="ZwinSzczegoly"
                    Margin="20,0,60,0"
                    HorizontalAlignment="Left"
                    VerticalAlignment="Center"

                    Click="zwinSzczegoly_Click"
                    RenderTransformOrigin="5.062,0.491"
                    Style="{DynamicResource MaterialDesignToolForegroundButton}"

```



```

        ToolTip="Zwiń">
        <materialDesign:PackIcon
            Foreground="Black"
            Width="24"
            Height="24"
            Kind="ArrowCompressUp" />
    </Button>
    <DataGrid
        Name="szczegoly"
        Margin="73,20,20,20"
        CanUserAddRows="False"
        CanUserSortColumns="True"
        ColumnWidth="Auto"
        IsHitTestVisible="False"
        IsReadOnly="True"
        Loaded="DataGrid_Loaded"
        AutoGeneratingColumn="Szczegoly_AutoGeneratingColumn"/>
    </Grid>
</DataTemplate>
</DataGrid.RowDetailsTemplate>
</DataGrid>
</Grid>

```

Listing 9. Przykładowa definicja kontrolki DataGrid w XAML.

Funkcjonalności kontrolek zostały zaprogramowane w metodach klas kontrolek. Funkcjonalności te są powiązane z kontrolkami poprzez ich atrybuty np. `Click="Zapisz_Click"`. Atrybut Click jest powiązany z metodą Zapisz_Click. Jeżeli kontrolka zawierająca atrybut Click zostanie kliknięta, to zostanie wykonana metoda przypisana do tego atrybutu. Rodzaje oraz przykłady metod zaprogramowanych na potrzeby pracy:

- Metoda zmieniająca treść kontrolki ContentControl - przykład został przedstawiony na listingu 10.

```

private void UsunDekretacje_Checked(object sender, RoutedEventArgs e)
{
    this.Cursor = Cursors.Wait;
    ContentArea.Content = new UsunDekretacje();
    Naglowek.Text = "USUŃ DEKRETACJE";
    this.Cursor = Cursors.Arrow;
}

```

Listing 10. Metoda tworząca nową kontrolkę i ustawiająca ją jako aktualny widok.

Metoda UsunDekretacje_Checked tworzy kontrolkę UsunDekretacje, a następnie podstawia ją jako treść kontrolki ContentControl, a także zmienia treść nagłówka i wygląd kursora na czas zmiany widoku.

- Metoda wykonująca zapytania SQL - przykład został przedstawiony na listingu 11.

```
private void pobierzPlanKont() {
    try{
        using (SqlConnection _con = new SqlConnection(connectionString))
        {
            string queryStatement = "SELECT * FROM tbPlanKont";
            using (SqlCommand _cmd = new SqlCommand(queryStatement, _con)) {
                DataTable wynik = new DataTable();
                SqlDataAdapter _dap = new SqlDataAdapter(_cmd);
                _con.Open();
                _dap.Fill(wynik);
                _con.Close();
                PlanKontDataGrid.ItemsSource = wynik.AsDataView();
            }
        }
    }
    catch (Exception error)
    {
        Error komunikat = new Error(error);
    }
}
```

Listing 11. Metoda pobierająca tabele z bazy danych.

Działanie metody pobierzPlanKont polega na otwarciu połączenia do bazy danych, wykonaniu zapytania SQL i jeżeli zapytanie zwraca dane to przekazanie otrzymanych danych do odpowiedniej kontrolki DataGrid. Jeśli podczas wykonywania metody wystąpią błędy, zostanie utworzone okno z komunikatem o błędach.

- Metoda wykonująca procedurę bazy danych - przykład został przedstawiony na listingu 12.

```
private void Szukaj_Click(object sender, RoutedEventArgs e)
{
    DataTable wyszukane = new DataTable();
    this.Cursor = Cursors.Wait;
    try
    {
        this.Cursor = Cursors.Wait;
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            using (SqlCommand cmd = new SqlCommand("spSzukajDekretacji", con))
            {
                cmd.CommandType = CommandType.StoredProcedure;
                if (Id.Text != "") { cmd.Parameters.Add("@ID",
                    SqlDbType.VarChar).Value = Id.Text; } else { cmd.Parameters.Add("@ID",
                    SqlDbType.VarChar).Value = DBNull.Value; }
                if (Kwota.Text != "") { cmd.Parameters.Add("@Kwota",
                    SqlDbType.Money).Value = Kwota.Text; } else { cmd.Parameters.Add("@Kwota",
                    SqlDbType.Money).Value = DBNull.Value; }

                SqlDataAdapter adp = new SqlDataAdapter(cmd);
            }
        }
    }
}
```

```

        con.Open();
        cmd.ExecuteNonQuery();
        adp.Fill(wyszukane);
        con.Close();
        var grid = sender as DataGrid;
        IdDekretacjiDG.ItemsSource = wyszukiwane.AsDataView();
        this.Cursor = Cursors.Arrow;
    }
}
}
catch (Exception error)
{
    Error komunikat = new Error(error);
}
}

```

Listing 12. Metoda wykonująca procedurę z bazy danych.

Działanie metody Szukaj_Click polega na nawiązaniu połączenia z bazą danych, utworzeniu komendy SQL typu procedura, pobraniu zawartości pól tekstowych i dodaniu ich do komendy SQL jako parametrów procedury, wykonanie procedury. Jeżeli dana procedura zwraca dane, to zostaną one przekazane do odpowiedniej kontrolki DataGrid. Jeśli podczas wykonywania metody wystąpią błędy zostanie utworzone okno z komunikatem o błędach.

- Metoda generująca DataGrid - przykład został przedstawiony na listingu 13.

```

grid.AutoGeneratingColumn += (s, e) =>
{
    if (e.PropertyType == typeof(System.DateTime)) {
        (e.Column as DataGridTextColumn).Binding.StringFormat = "dd/MM/yyyy"; }
    if (e.PropertyType == typeof(System.Decimal)) {
        (e.Column as DataGridTextColumn).Binding.StringFormat = "{0:N}" + " Zł"; }
    if (e.PropertyName == "IdWpisu") { e.Column.Header =
        "ID Wpisu"; }
    if (e.PropertyName == "NrDokumentu") {
        e.Column.Header = "Nr Dokumentu"; }
    };
}

```

Listing 13. Metoda wykonywana podczas generowania zawartości kontrolki DataGrid.

Metoda AutoGeneratingColumn pozwala nam na zmianę formatu danych, niewyświetlanie pewnych kolumn oraz zmianę nagłówków kolumn w kontrolce DataGrid podczas generowania zawartości kontrolki.

- Metoda sprawdzająca poprawność danych wpisanych do formularza została przedstawiona na listingu 14.

```

private bool Validacja()
{

```

```

Regex regex = new Regex("^[1-9][0-9]*$");
Match matchKwota = regex.Match(Kwota.Text);
Match matchKWS = regex.Match(KWS.Text);
Color kolor = (Color)ColorConverter.ConvertFromString("#89FFFFFF");
Brush brush = new SolidColorBrush(kolor);
Kwota.BorderBrush = brush;
opis.BorderBrush = brush;
KWS.BorderBrush = brush;
bool wynik = true;
if (!matchKwota.Success) { Kwota.BorderBrush = Brushes.Red; wynik = false; }
if (opis.Text == "") { opis.BorderBrush = Brushes.Red; wynik = false; }
if(KWS.Visibility == Visibility.Visible)
{
    if (!matchKWS.Success) { KWS.BorderBrush = Brushes.Red; wynik = false; }
}

return wynik;
}

```

Listing 14. Metoda sprawdzająca poprawność danych wpisanych do formularza.

Działanie metody Validacja polega na sprawdzeniu, czy wszystkie dane wpisane przez użytkownika do formularza spełniają określone wymagania. Jeżeli dane pole nie spełnia wymagań, kolor jego obramowania zostaje zmieniony na czerwony, aby zasygnalizować użytkownikowi niepoprawność wpisanych przez niego danych.

6. Testy w środowisku użytkowym i opinia użytkownika

Aplikacja KSH została zaprezentowana użytkownikowi docelowemu w celu przetestowania działania funkcji oraz uzyskania opinii. Opinia użytkownika po przetestowaniu aplikacji:

„Program realizuje część podstawowych funkcji potrzebnych w firmie i aby aplikacja mogła znaleźć praktycznie zastosowanie w firmie, należy utworzyć brakujące funkcje takie jak np. polecenie księgowania na dowolne konto. Generowane przez program zestawienia są poprawne. Interfejs aplikacji jest prosty i nie zawiera tysięcy niepotrzebnych funkcji, więc łatwo jest się nią posługiwać, ale posiada elementy, które są mało czytelne i można by je poprawić np. lista dokumentów, które można księgować na stronie z dekreacją. Szybkość działania jest bardzo dobra i jest największym atutem aplikacji. Dużym plusem jest także możliwość łatwego kopiowania danych za pomocą

skrótów klawiszowych. Po dodaniu kilku dodatkowych funkcji Aplikacja jak najbardziej będzie nadawała się do użytkowania ”

- Jerzy Krzepkowski.

Na Podstawie opinii można wywnioskować, że aplikacja KSH nie jest jeszcze gotowa do użytkowania, ale ma szansę taką gotowość osiągnąć. W tym celu należałoby poprawić czytelność interfejsu oraz dodać brakujące funkcjonalności, których realizacja nie była przewidziana w ramach pracy. Można także zauważyć, że zaimplementowane funkcjonalności spełniają wymagania funkcjonalne i нефункционалне. System KSH wymaga więc dalszej pracy i kolejnych testów.

7. Dokumentacja użytkowa

7.1. Instrukcja obsługi aplikacji

Po uruchomieniu aplikacji użytkownik znajduje się na zakładce plan kont, który wyświetla plan kont przechowywany w bazie danych. Aby zmienić zakładkę należy kliknąć na jeden z przycisków znajdujących się na żółtym górnym pasku aplikacji. Zakładki zostały podzielone na zakładki jedynie prezentujące dane oraz zakładki umożliwiające użytkownikowi wykonywanie operacji na bazie danych. Zakładki służące do prezentacji danych zostały przedstawione na rysunkach 15, 16, 17, 18 i 19.

KSH

PLAN KONTDEKRETACJAKSIĘGA GŁÓWNAUSUŃ DEKRETACJIERZISRZISZOISBILANS

PLAN KONT

ID Syntetyki	Opis Konta	Typ Konta	Aktywne/Pasywne/Mieszane	Nazwa Tabeli Konta	BO WN	BO MA	BZ WN	BZ MA
010	Srodki trwałe	bilansowe	A	tb010_SrodkiTrwale	10.00 Zł			
050	Umorzenie środków trwałych	bilansowe	A	tb050_umorzenieSrodkiTrwale				
100	Kasa	bilansowe	A	tb100_Kasa	100.00 Zł			
130	Rachunek bieżący	bilansowe	A	tb130_RachunekBiezacy	1,300.00 Zł			
201	Rozrachunki z odbiorcami	rozrachunkowe	A	tb201_RozrachunkiOdbiorcy	2,010.00 Zł			
202	Rozrachunki z dostawcami	rozrachunkowe	P	tb202_RozrachunkiDostawcy	0.00 Zł	2,020.00 Zł		
221	Rozrachunki Podatek Dochodowy	rozrachunkowe	AP	tb221_RozrachunkiPodatekDochodowy				
223	rozrachunki z US z tyt VAT	rozrachunkowe	AP	tb223_RozrachunkiVATurzaSkarb				
224	VAT należny	bilansowe	P	tb224_VATNalezny				
225	VAT naliczony	bilansowe	A	tb225_VATNaliczony				
242	pozostałe rozrachunki	rozrachunkowe	P	tb242_PozostaleRozrachunki				
247	Pozyczki otrzymane	bilansowe	P	tb247_PozyczkiOtrzymane				
250	Rozrachunki z udziałowcami	rozrachunkowe	P	tb250_RozrachunkiUdzialowcy				
330	Towary	bilansowe	A	tb330_Towary	30,000.00 Zł			
410	Zuzycie materialow i energia	wynikowe	A	tb410_KosztyMaterEnergia				
420	Uslugi obce	wynikowe	A	tb420_KosztyUslugiObce	9,999.00 Zł	0.00 Zł		

Rysunek 15. Zrzut ekranu przedstawiający zakładkę plan kont.

KSIĘGA GŁÓWNA								
Data Operacji	Kontrahent	Opis Operacji	ID Autodekretacji	Nr Dokumentu	Konto	Obroty WN	Obroty MA	Data Księgowania
08/02/2019	Kontrahent7	Dekretacja Zaplata Dos	32063	123	130	0.00 Zł	1,000.00 Zł	08/04/2019
08/02/2019	Kontrahent8	Dekretacja Zaplata Dos	32064	123	130	0.00 Zł	1,000.00 Zł	08/04/2019
08/02/2019	Kontrahent5	Dekretacja Zakupu	32061	123	202	0.00 Zł	1,230.00 Zł	08/04/2019
08/02/2019	Kontrahent6	Dekretacja Zakupu	32062	123	202	0.00 Zł	1,230.00 Zł	08/04/2019
08/02/2019	Kontrahent7	Dekretacja Zaplata Dos	32063	123	202	1,000.00 Zł	0.00 Zł	08/04/2019
08/02/2019	Kontrahent8	Dekretacja Zaplata Dos	32064	123	202	1,000.00 Zł	0.00 Zł	08/04/2019
08/02/2019	Kontrahent1	Dekretacja Kosztow 410	32056	123	225	230.00 Zł	0.00 Zł	08/04/2019
08/02/2019	Kontrahent2	Dekretacja Kosztow 410	32058	123	225	230.00 Zł	0.00 Zł	08/04/2019
08/02/2019	Kontrahent4	Dekretacja Kosztow 410	32059	123	225	230.00 Zł	0.00 Zł	08/04/2019
08/02/2019	Kontrahent5	Dekretacja Zakupu	32061	123	225	230.00 Zł	0.00 Zł	08/04/2019
08/02/2019	Kontrahent6	Dekretacja Zakupu	32062	123	225	230.00 Zł	0.00 Zł	08/04/2019
08/02/2019	Kontrahent1	Dekretacja Kosztow 410	32056	123	242	0.00 Zł	1,230.00 Zł	08/04/2019
08/02/2019	Kontrahent2	Dekretacja Kosztow 410	32058	123	242	0.00 Zł	1,230.00 Zł	08/04/2019
08/02/2019	Kontrahent4	Dekretacja Kosztow 410	32059	123	242	0.00 Zł	1,230.00 Zł	08/04/2019
08/02/2019	Kontrahent5	Dekretacja Zakupu	32061	123	330	1,000.00 Zł	0.00 Zł	08/04/2019
08/02/2019	Kontrahent6	Dekretacja Zakupu	32062	123	330	1,000.00 Zł	0.00 Zł	08/04/2019

Rysunek 16. Zrzut ekranu przedstawiający zakładkę księga główna.

KSH		PLAN KONT	DEKRETACJA	KSIĘGA GŁÓWNA	USUŃ DEKRETACJIE	RZIS	ZOIS	BILANS
RZiS								
Opis		Wartość						
PrzychodyZPodstawowejDzialOperac		12,000.00 Zł						
PrzychodyZeSprzedazy		12,000.00 Zł						
PozostalePrzychodyOperacyjne		0.00 Zł						
KosztyDzialalnosciOperacyjnej		10,200.00 Zł						
KosztWlasnySprzedazy		7,200.00 Zł						
KosztyMaterEnergia		3,000.00 Zł						
KosztyUslugiObce		0.00 Zł						
KosztyWynagrodzenia		0.00 Zł						
PozostaleKosztyOperacyjne		0.00 Zł						
ZyskStrataZeSprzedazy		1,800.00 Zł						
ZyskStrataZDzialalnosciOperacyjnej		1,800.00 Zł						
PodatekDochodowy		342.00 Zł						
ZyskStrataNetto		1,458.00 Zł						

Rysunek 17. Zrzut ekranu przedstawiający zakładkę rachunek zysków i strat.

KSH		PLAN KONT	DEKRETACJA	KSIĘGA GŁÓWNA	USUŃ DEKRETACJIE	RZIS	ZOIS	BILANS
ZOiS								
BO WN		BO MA	Obroty WN	Obroty MA	BZ WN	BZ MA		
0.00 Zł		0.00 Zł	30,110.00 Zł	30,110.00 Zł	0.00 Zł	0.00 Zł		
Konto		BO WN	BO MA	Saldo WN	Saldo MA	BZ WN	BZ MA	
130		0.00 Zł	0.00 Zł	0.00 Zł	2,000.00 Zł	0.00 Zł	0.00 Zł	
201		0.00 Zł	0.00 Zł	14,760.00 Zł	0.00 Zł	0.00 Zł	0.00 Zł	
202		0.00 Zł	0.00 Zł	2,000.00 Zł	2,460.00 Zł	0.00 Zł	0.00 Zł	
224		0.00 Zł	0.00 Zł	0.00 Zł	2,760.00 Zł	0.00 Zł	0.00 Zł	
225		0.00 Zł	0.00 Zł	1,150.00 Zł	0.00 Zł	0.00 Zł	0.00 Zł	
242		0.00 Zł	0.00 Zł	0.00 Zł	3,690.00 Zł	0.00 Zł	0.00 Zł	
330		0.00 Zł	0.00 Zł	2,000.00 Zł	7,200.00 Zł	0.00 Zł	0.00 Zł	
410		0.00 Zł	0.00 Zł	3,000.00 Zł	0.00 Zł	0.00 Zł	0.00 Zł	
730		0.00 Zł	0.00 Zł	0.00 Zł	12,000.00 Zł	0.00 Zł	0.00 Zł	
735		0.00 Zł	0.00 Zł	7,200.00 Zł	0.00 Zł	0.00 Zł	0.00 Zł	

Rysunek 18. Zrzut ekranu przedstawiający zakładkę zestawienie obrotów i sald.

BILANS	
Opis	Wartość
Aktywa	23,560.00 Zł.
Wartosci niemater i prawne	1,000.00 Zł.
Rzeczowe Aktywa Trwale	15,000.00 Zł.
Należności Długoterminowe	0.00 Zł.
Należności Krotkoterminowe	14,760.00 Zł.
Zapasy	-5,200.00 Zł.
Srodki na rachunkach i w kasie	-2,000.00 Zł.
Pasywa	23,560.00 Zł.
Kapitał Fundusz Podstawowy	5,000.00 Zł.
Kapitał Fundusz Zapasowy	3,000.00 Zł.
Zobowiązania Krotkoterminowe	460.00 Zł.
Kapitał Fundusz Rezerwowy	13,642.00 Zł.
Zysk Strata Z Lat Ubiegłych	0.00 Zł.
Zysk Strata Netto	1,458.00 Zł.

Rysunek 19. Zrzut ekranu przedstawiający zakładkę bilans.

Zakładki umożliwiające wykonywanie operacji na bazie danych to dekreteacja oraz usuń dekreteację.

Po przejściu do zakładki dekreteacja należy wybrać z listy znajdującej się po lewej stronie okna rodzaj dokumentu, który ma być księgowany następnie wypełnić formularz odpowiednimi danymi. Aby zapisać dane, należy nacisnąć przycisk Zapisz. Jeżeli któreś pole formularza zostało błędnie wypełnione, kolor obramowania tego pola zmieni się na kolor czerwony. Zakładka dekreteacja została przedstawiona na rysunku 20.

Zakładka usuń dekreteację umożliwia wyszukiwanie dokumentów zapisanych w bazie danych, podgląd dokumentów zapisanych w bazie danych wraz z wszystkimi rekordami powiązanymi, usunięcie dokumentu z bazy danych. Aby wyszukać dokument należy wypełnić wybrane pola formularza znajdującego się w górnej części okna. Po wpisaniu danych należy nacisnąć przycisk z ikoną lupy. Wyszukane dane pojawią się w tabeli poniżej. Aby uzyskać podgląd rekordów powiązanych z danym dokumentem należy, kliknąć na wierszu zawierającym dane o dokumencie. Aby zwinąć tabelę przedstawiającą rekordy powiązane należy kliknąć na ikonę strzałki znajdującą się po lewej stronie tej tabeli. Aby usunąć dokument z bazy danych należy kliknąć na przycisku usuń w wierszu w którym znajdują się informacje o danym dokumencie. Zakładka usuń dekreteację została przedstawiona na rysunku 21.

KSH

PLAN KONT

DEKRETACJA

KSIĘGA GŁÓWNA

USUŃ DEKRETACJE

RZIS

ZOIS

BILANS

DEKRETACJA

DekretacjaKosztow410MatEn

DekretacjaKosztow420UslObce

DekretacjaKosztow460Wynagr

DekretacjaSprzedazy

DekretacjaZakupu

DekretacjaZaplataDostawcyGot

DekretacjaZaplataDostawcyPrzel

DekretacjaZaplataOdbiorcyGot

DekretacjaZaplataOdbiorcyPrzel

DekretacjaZaplataWynagrPrzel

DekretacjaZaplataKosztyGot

DekretacjaZaplataKosztyPrzel

DEKRETACJA

DekretacjaKosztow410MatEn

Koszt Netto

1000

Wybierz Date

08.02.2019

Kontrahent

Kontrahent1

Księgujący

hk

Numer Dokumentu

1

Opis Operacji

DekretacjaKosztow410MatEn

Zapisz

Rysunek 20. Zrzut ekranu przedstawiający zakładkę dekreteacja.

KSH

PLAN KONT

DEKRETACJA

KSIĘGA GŁÓWNA

USUŃ DEKRETACJE

RZIS

ZOIS

BILANS

USUŃ DEKRETACJE

ID

Opis

Od / Dnia

Do

Księgujący

Kontrahent

Nr Dokumentu

Kwota

ID Autodekretacji	Opis	Data Dekretacji	Księgujący	Kontrahent	Nr Dokumentu	Kwota	
32056	KosztyMatEn	08/04/2019	hk	Kontrahent1	123	1,000.00 ZŁ	Usuń
32058	KosztyMatEn	08/04/2019	hk	Kontrahent2	123	1,000.00 ZŁ	Usuń

IdWpisu

410

Zuzycie materialow i energia

Analityka

Kontrahent

Data Operacji

Opis Operacji

Nr Dokumentu

BO WN

BO MA

WN

MA

ID Autod

2

225

Kontrahent2

08/02/2019

DekretacjaKosztow410

123

230.00 ZŁ

0.00 ZŁ

32058

2

242

Kontrahent2

08/02/2019

DekretacjaKosztow410

123

0.00 ZŁ

1,230.00 ZŁ

32058

2

410

Kontrahent2

08/02/2019

DekretacjaKosztow410

123

1,000.00 ZŁ

0.00 ZŁ

32058

32059	KosztyMatEn	08/04/2019	hk	Kontrahent4	123	1,000.00 ZŁ	Usuń
32061	DekretacjaZakupu	08/04/2019	hk	Kontrahent5	123	1,000.00 ZŁ	Usuń
32062	DekretacjaZakupu	08/04/2019	hk	Kontrahent6	123	1,000.00 ZŁ	Usuń

Rysunek 21. Zrzut ekranu przedstawiający zakładkę usuń dekreteacje.

41

8. Uwagi końcowe

8.1. Wnioski

Celem pracy było stworzenie aplikacji realizującej podstawowe operacje księgowe wykonywane w konkretnej firmie. W pracy zdefiniowano jakie narzędzia będą wykorzystywane przy tworzeniu aplikacji, zdefiniowano założenia oraz wymagania. Przedstawiono projekt aplikacji oraz proces jej tworzenia. Utworzona aplikacja została przedstawiona użytkownikowi w celu jej przetestowania oraz uzyskania opinii. Opinia użytkownika wskazuje na to, że aplikacja musi być dalej rozwijana. Aby mogła wejść do użytku w firmie muszą zostać zaimplementowane dodatkowe funkcjonalności. Z opinii użytkownika wynika także że funkcjonalności, które zostały zaimplementowane, spełniają zdefiniowane w pracy wymagania, a niektóre elementy aplikacji wymagają drobnych poprawek.

8.2. Perspektywy i planowany rozwój systemu

Księgowość jest nierozłącznym elementem prowadzenia działalności gospodarczej, można więc założyć, że aplikacje służące do obsługi księgowości będą tak długo potrzebne, jak długo będą istnieć firmy.

Aplikacje KSH można rozwijać poprzez dodawanie nowych funkcji i rozbudowanie systemu do rozmiarów oferowanych przez komercyjne rozwiązania dla księgowości. Aby program mógł stać się alternatywą dla obecnie używanych rozwiązań, aplikacja musiałaby zmienić strukturę na dynamiczną, umożliwiając użytkownikowi definiowanie planu kont oraz schematów dekretacji, ponieważ każda firma może posiadać inną strukturę księgową. Kolejnym istotnym aspektem rozwoju systemu KSH może być możliwość tworzenia kont użytkowników i nadawanie im uprawnień.

KSH można by także rozwijać w kierunku aplikacji webowej. Takie rozwiązania stają się coraz bardziej popularne, ale wiąże się to z koniecznością zabezpieczenia aplikacji przed niebezpieczeństwami związanymi z przesyłaniem danych poufnych oraz nieuprawnionym dostępem do aplikacji.

Rozwój systemu w najbliższym czasie będzie obejmował funkcje potrzebne w firmie i dopiero kiedy aplikacja będzie w pełni gotowa do użytkowania zostanie wybrany kierunek dalszego rozwoju.

Literatura:

1. https://www.findict.pl/academy/rachunkowosc/podstawy_rachunkowosci/co_to_jest_rachunkowosc.php, [dostęp Luty 2019]
2. <https://www.sqlshack.com/sql-server-network-configuration/>, [dostęp Luty 2019]
3. <http://www.druki.gofin.pl/sprawozdania-finansowe-dla-jednostek-mikro,wzory,125.html>, [dostęp Luty 2019]
4. <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>, [dostęp Luty 2019]
5. <http://www.ii.uni.wroc.pl/~wzychla/cs.html>, [dostęp Luty 2019]
6. <https://docs.microsoft.com/pl-pl/dotnet/framework/wpf/getting-started/>, [dostęp Luty 2019]
7. <http://materialdesigninxaml.net/>, [dostęp Luty 2019]
8. https://en.wikipedia.org/wiki/Microsoft_Visual_Studio, [dostęp Luty 2019]
9. <https://docs.microsoft.com/pl-pl/dotnet/framework/wpf/advanced/xaml-overview-wpf>, [dostęp Luty 2019]
10. https://en.wikipedia.org/wiki/Microsoft_SQL_Server, [dostęp Luty 2019]

Dodatki:

Płyta CD z następującą zawartością:

- Praca inżynierska - w wersji elektronicznej.
- Oprogramowanie systemu KSH.