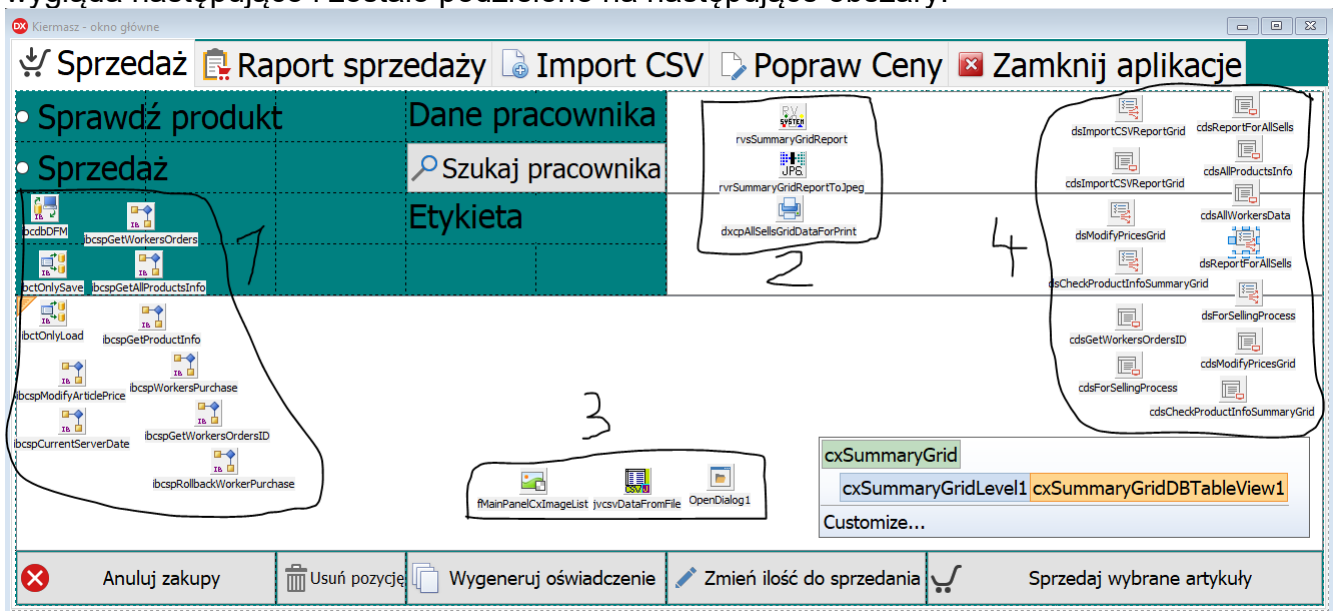


Instrukcja została podzielona na 6 głównych rozdziałów (każdy dla jednej z formatek znajdujących się w projekcie):

I. Formatka „uMainPanel”	str. 1
II. Formatka „uListOfWorkers”	str. 14
III. Formatka „uEditWorkerData”	str. 15
IV. Formatka „uInfoBox”	str. 16
V. Wątek „GetWorkersDataThread”	str. 17
VI. Wątek „ImportDataFromCSVToDBThread”	str. 18

I. Projekt „Kiermasz” - formatka „uMainPanel” Zakładka „Sprzedaż”

Po uruchomieniu projektu w programie Delphi 10.3, główne okno programu “Kiermasz” wygląda następująco i zostało podzielone na następujące obszary:



1. W tym obszarze znajdują się wszelkie elementy bazo-danowe: wszystkie procedury używane w programie, dwie transakcje, jeden obiekt do łączenia się z bazą: „ibcdbDFM”
Wszystkie użyte obiekty pochodzą z biblioteki IBDAC
nazwy procedur w programie są takie same jak ich odpowiedniki w bazie, np.:
ibcspGetWorkersOrders w programie = GET_WORKERS_ORDERS w bazie
Dwie transakcje, jedna do wczytywania, druga do zapisywania danych.
Jeden obiekt „ibcdbDFM” do łączenia się z bazą.

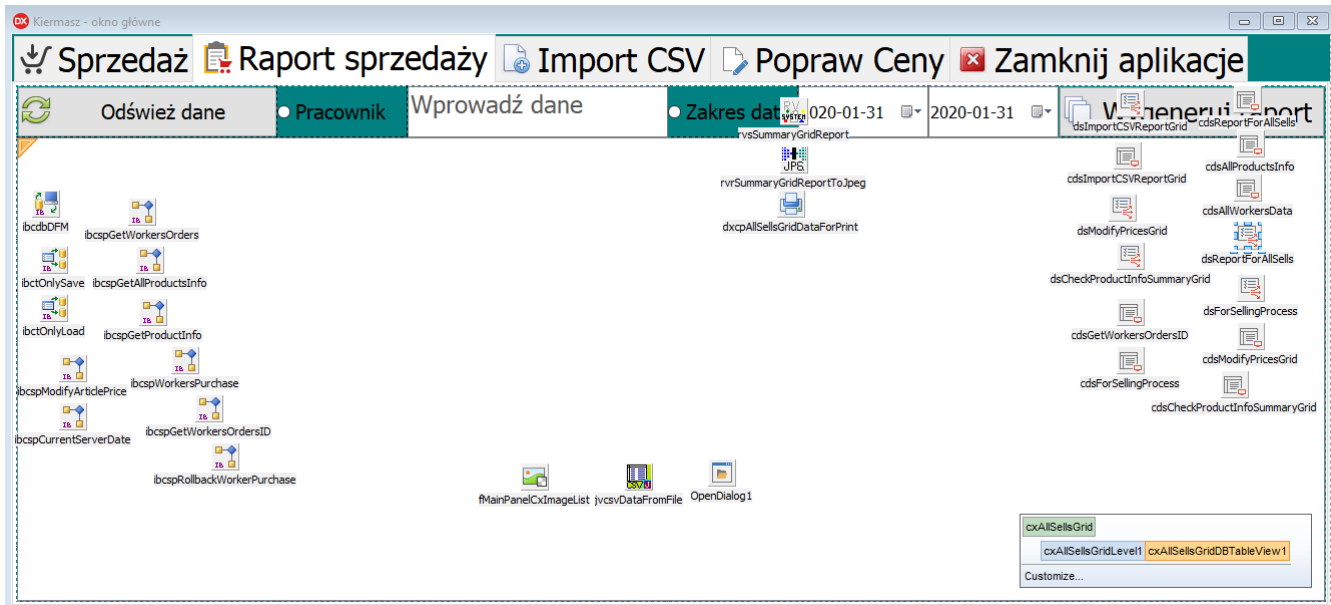
2. Obszar zawierający wszystkie obiekty użyte w ramach generowania widoku raportów oraz samych raportów, z bibliotek: ExpressPrinting System, Rave.

3. Obiekt: „fMainPanelCxImageList” - wszystkie ikonki przypisane do przycisków w tej formacie.

Dwa pozostałe: wczytywanie danych z pliku CSV do bazy, biblioteka: „Jv Data Access”.

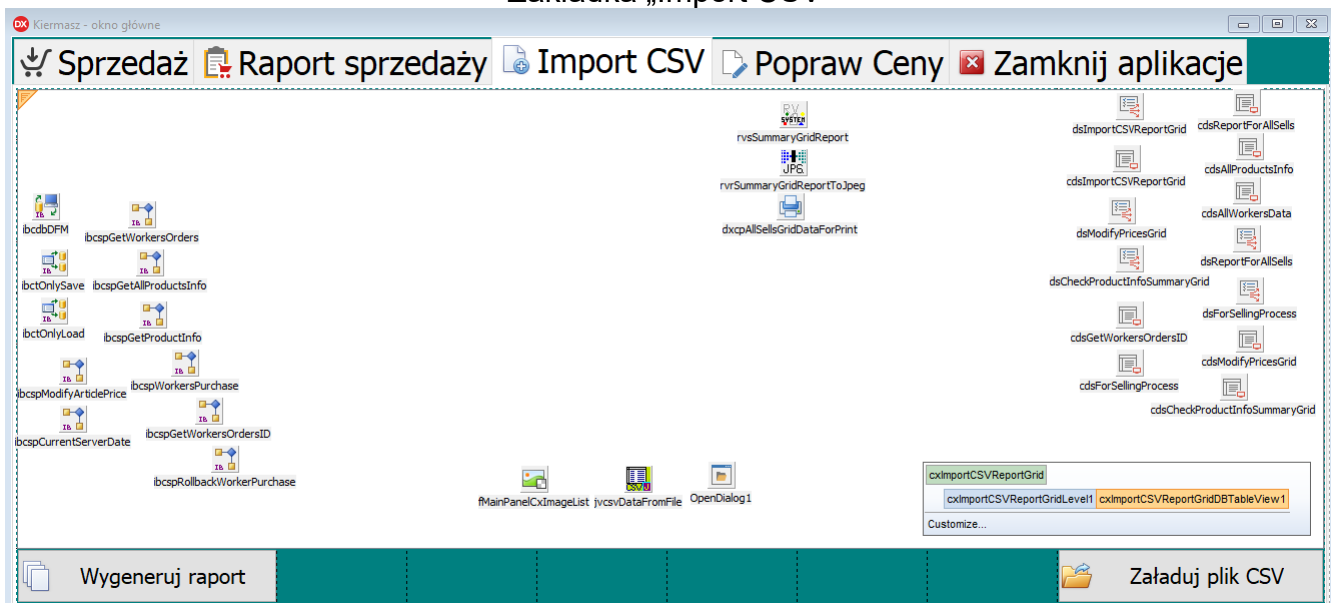
4. Obszar zawierający wszystkie obiekty będące wewnętrznymi tabelami przechowującymi dane z bazy na czas działania programu oraz obiekty podłączane do gridów w ramach wyświetlania danych w gridach. Biblioteka: Data Access.

Zakładka „Raport sprzedaży”



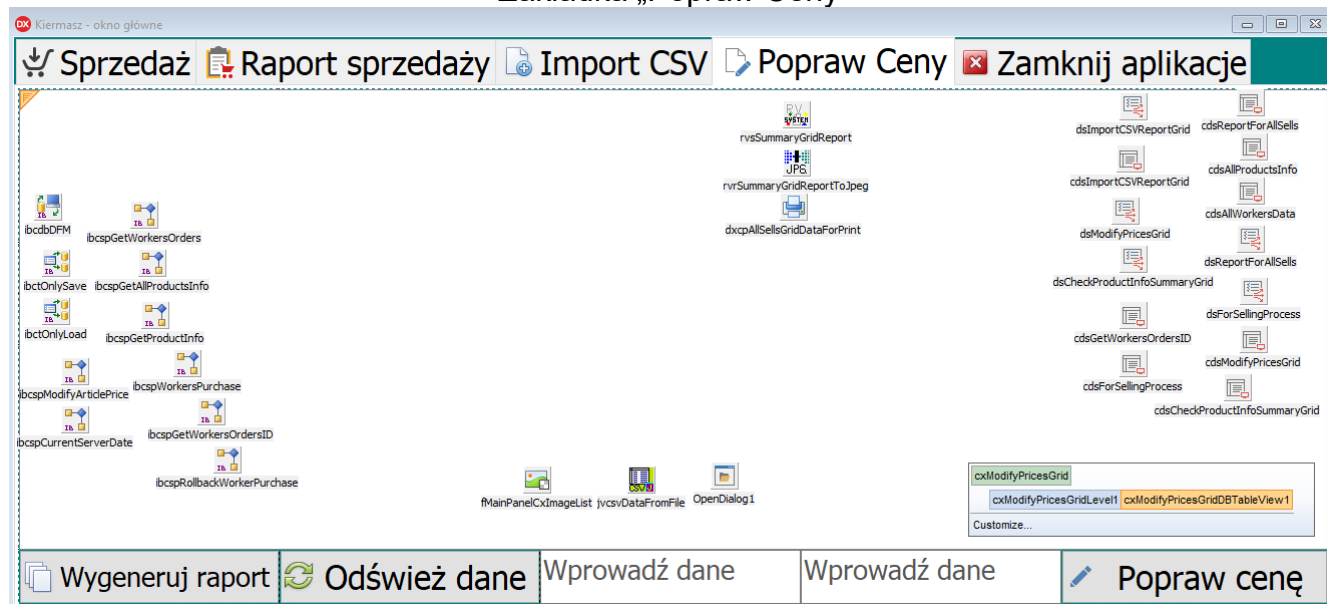
Jeden widok w gridzie, po wybraniu opcji „Odśwież dane”, „Pracownik” lub „Zakres dat (od – do)” dane zostaną pobrane do grida w zależności od wybranej opcji.

Zakładka „Import CSV”



Po wybraniu pliku, pojawiają się tutaj błędy, które wystąpiły w trakcie próby importu pliku CSV do bazy.

Zakładka „Popraw Ceny”



„Odśwież dane” do załadowania danych o wszystkich produktach z bazy, nie tych kupionych ale wszystkich, które „uczestniczą” w kiermaszu.

Możliwość poprawy ceny wybranego w TcxGrid artykułu - „Popraw cenę”.

Generowanie raportów - „Wygeneruj raport”.

Wyszukiwanie dynamiczne (po podaniu pełnej nazwy artykułu / jego odwo_id) w jednym z dwóch pól „Wprowadź dane”. Oba pola posiadają odpowiednie podpowiedzi, które wyświetlają się po najechaniu na jeden z nich.

Zakładka „Zamknij aplikację”

Zamyka program.

Uruchomienie programu

Na starcie program pyta się o pobranie danych pracowników oraz import pliku CSV z danymi artykułów do bazy:

Procedura TfMainPanel.FormShow(Sender: TObject):

dateFrom, dateTo – zmienne przechowujące zakresy dat podawane w zakładce „Raport sprzedaży”.

delayForClosingApp - zmienna pomocna przy decydowaniu, czy zamknąć aplikację.

IfFinalizeSellButtonClickInFMainPanel - zmienna używana przez formatkę "fEditWorkerData" do decydowania, jakie fragmentu kodu wykorzystać.

delayInFinalizeSellButtonClickInFMainPanel - zmienna pomocna w decydowaniu, czy podczas finalizacji sprzedaży należy sprawdzić, czy wszystkie dane adresowe zostały uzupełnione, lub czy już przejść do samego procesu finalizacji sprzedaży.

Wykorzystanie własnej funkcji do generowania okien dialogowych:

MyMessageDialog

Po wybraniu opcji importu pliku CSV, dodatkowo wykorzystanie procedury:

TfMainPanel.LoadCSVFileButtonClick(Sender: TObject);

Do pobrania danych wszystkich pracowników program wykorzystuje wątek:

GetWorkersDataThread

Funkcjonalność zakładki „Sprzedaż”:

Opcja „Sprawdź produkt”:

WorkerIDEditInSellTabPanel.Enabled – W tym trybie wyłączenie możliwości wprowadzania danych w polu „Dane pracownika”.

CancelCurrentSellingButton.Enabled - W tym trybie wyłączenie możliwości anulowania zakupów: przycisk „Anuluj zakupy”.

DeleteCurrentSummaryPositionButton.Enabled – W tym trybie wyłączenie przycisku „Usuń pozycję”.

FinalizeSellButton.Enabled – W tym trybie wyłączenie przycisku „Sprzedaj wybrane artykuły”.

GenerateReportForSellButton.Enabled - W tym trybie wyłączenie przycisku „Wygeneruj oświadczenie”.

ChangeAmountOfChosenProductToSellButton.Enabled - W tym trybie wyłączenie przycisku „Zmień ilość do sprzedania”.

cxSummaryGridDBTableView1.DataController.DataSource – Ustawienie źródła danych dla grida do odpowiedniego Data Source oraz późniejsze **CreateAllItems** w celu załadowania tych danych do grida.

Opcja „Sprzedaż”:

Procedura: **TfMainPanel.SellRadioButtonClick(Sender: TObject);**

Włączenie przycisków wymienionych w opcji wyżej.

Ustawienie odpowiedniego dla tej opcji Data Source dla grida, załadowanie danych.

Input „Etykieta” (ProductIDEdit):

Wprowadzanie nowych produktów do listy do sprzedania lub do sprawdzenia.

Procedura: **TfMainPanel.ProductIDEditKeyPress(Sender: TObject; var Key: Char);**

Szczegóły w kodzie źródłowym.

Input „Dane pracownika” (WorkerIDEditInSellTabPanel):

Określenie, dla którego konkretnie pracownika zostanie przeprowadzona sprzedaż.

Procedura: **TfMainPanel.WorkerIDEditInSellTabPanelKeyPress(Sender: TObject; var Key: Char);**

Po wpisaniu w pole numeru ewidencyjnego lub zeskanowaniu przepustki, dane są zapisywane jako zmienna globalna (prywatna). Używana przez inne procedury.

Kod procedury czeka na wciśnięcie klawisza ENTER, po czym nastąpi takie przypisanie wprowadzonego ciągu znaków do zmiennej globalnej (prywatnej).

Przycisk „Szukaj pracownika” (SearchWorkerButton):

Możliwość wpisania nazwiska szukanego pracownika i wyświetlenia listy ze znalezionymi pracownikami po wpisanym wcześniej nazwisku.

Procedura: **TfMainPanel.SearchWorkerButtonClick(Sender: TObject);**

Przycisk „Anuluj zakupy” (CancelCurrentSellingButton):

Anulowanie obecnie trwającej sprzedaży.

Procedura: **TfMainPanel.CancelCurrentSellingButtonClick(Sender: TObject);**

Odznaczenie obu możliwości „Sprawdź produkt”, „Sprzedaż”. Wyczyszczenie obu pól do podawania danych (Dane pracownika, Etykieta), wyczyszczenie obu data source przechowujących dane dla obu opcji („Sprawdź produkt”, „Sprzedaż”).

Przycisk „Usuń pozycję” (DeleteCurrentSummaryPositionButton):

Usunięcie z listy obecnie trwającej sprzedaży któregoś z artykułów (pierw zaznaczonego na gridzie).

Procedura: **TfMainPanel.DeleteCurrentSummaryPositionButtonClick(Sender: TObject);**

Przycisk „Wygeneruj oświadczenie” (GenerateReportForSellButton):

Generowanie oświadczenia dla pracownika z listą rzeczy wprowadzonych w czasie trwania procesu sprzedaży.

Procedura: **TfMainPanel.GenerateReportForSellButtonClick(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Obiekt „rvsSummaryGridReport”:

Generowanie raportów sprzedażowych z uwzględnieniem imienia i nazwiska pracownika.

Procedura: **TfMainPanel.rvsSummaryGridReportPrint(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Przycisk „Zmień ilość do sprzedania” (ChangeAmountOfChosenProductToSellButton):

Zmiana ilości wybranego artykułu z grida do sprzedania.

Procedura: **TfMainPanel.ChangeAmountOfChosenProductToSellButtonClick(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Przycisk „Sprzedaj wybrane artykuły” (FinalizeSellButton):

Przycisk do zakończenia obecnie trwającej sprzedaży.

Procedura: **TfMainPanel.FinalizeSellButtonClick(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Funkcjonalność zakładki „Raport sprzedaży”:

Przycisk „Odśwież dane” (RefreshDataButtonInReportForSellsMode):

Wczytywanie danych o wszystkich dokonanych sprzedażyach do grida.

Procedura: **TfMainPanel.RefreshDataButtonInReportForSellsModeClick(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Opcja do wyboru „Pracownik” (WorkerRadioButton):

Wczytywanie danych o dokonanych sprzedażyach do grida dla wybranego pracownika.

Procedura: **TfMainPanel.WorkerRadioButtonClick(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Input „Wprowadź dane” (WorkerIDEditInReportForSells):

Input przyjmujący dane wykorzystywane do zastosowania filtra na danych sprzedażowych w celu uzyskania sprzedaży tylko konkretnego pracownika.

Możliwość użycia numeru ewidencyjnego lub kodu z przepustki.

Procedura: **TfMainPanel.WorkerIDEditInReportForSellsChange(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Opcja do wyboru „Zakres dat” (DateRangeRadioButton):

Wczytywanie danych o dokonanych sprzedażyach w wybranym przedziale czasowym (daty,dni).

Procedura: **TfMainPanel.DateRangeRadioButtonClick(Sender: TObject);**

Umożliwienie korzystania z pickerów do wyboru dat:

DateFromTimePicker oraz DateToTimePicker

Reset wcześniej zastosowanego filtra danych w tabeli z danymi sprzedażowymi cdsReportForAllSells.

Opcja do wyboru daty od (DateFromTimePicker):

Wczytywanie danych o sprzedanych artykułach zaczynając od tej daty.

Procedura: **TfMainPanel.DateFromTimePickerChange(Sender: TObject);**

Zastosowanie odpowiedniego zasięgu filtrowania na danych sprzedażowych wykorzystującego datę (od).

Tabela ze wszystkimi danymi sprzedażowymi: cdsReportForAllSells

Opcja do wyboru daty do (DateTimePicker):

Wczytywanie danych o sprzedanych artykułach kończąc na tej dacie.

Procedura: **TfMainPanel.DateTimePickerChange(Sender: TObject);**

Zastosowanie odpowiedniego zasięgu filtrowania na danych sprzedażowych wykorzystującego datę (do).

Tabela ze wszystkimi danymi sprzedażowymi: cdsReportForAllSells

Przycisk „Wygeneruj raport” (GenerateReportButton):

Drukowanie wybranego przez użytkownika (CSV lub PDF) raportu.

Procedura: **TfMainPanel.GenerateReportButtonClick(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Funkcjonalność zakładki „Import CSV”:

Przycisk „Wygeneruj raport” (GenerateReportImportCSVButton):

Drukowanie wybranego przez użytkownika (CSV lub PDF) raportu.

Procedura: **TfMainPanel.GenerateReportImportCSVButtonClick(Sender: TObject);**

Zasada działania taka sama jak w przypadku procedury:

TfMainPanel.GenerateReportButtonClick(Sender: TObject);

Przycisk „Załaduj plik CSV” (LoadCSVFileButton):

Wczytywanie do bazy danych informacji z pliku CSV.

Procedura: **TfMainPanel.LoadCSVFileButtonClick(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Funkcjonalność zakładki „Popraw Ceny”:

Przycisk „Odśwież dane” (RefreshDataButtonInModifyPricesMode):

Wczytanie danych o artykułach do grida.

Procedura: **TfMainPanel.RefreshDataButtonInModifyPricesMode(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Przycisk „Wygeneruj raport” (GenerateReportInModifyPriceModeButton):

Drukowanie wybranego przez użytkownika (CSV lub PDF) raportu.

Procedura: **TfMainPanel.GenerateReportInModifyPriceModeButtonClick(Sender: TObject);**

Zasada działania taka sama jak w przypadku procedury:

TfMainPanel.GenerateReportButtonClick(Sender: TObject);

Przycisk „Popraw cenę” (ModifyPriceButton):

Zmiana ceny zaznaczonego produktu w gridzie. Zmiana ceny jednocześnie na bazie z artykułami i w gridzie wraz z przeliczeniem nowej ceny z VAT.

Procedura: **TfMainPanel.ModifyPriceButtonClick(Sender: TObject);**

Szczegóły w kodzie źródłowym.

Input „Wprowadź dane” (OdwoIDSearchingModelInModifyPricesGrid):

Zastosowanie filtra na danych o artykułach z tabeli cdsModifyPricesGrid w celu znalezienia danego produktu po jego odwo_id.

Procedura: **TfMainPanel.OdwoIDSearchingModelInModifyPricesGridChange(Sender: TObject);**

Input „Wprowadź dane” (NameSearchingModelInModifyPricesGrid):

Zastosowanie filtra na danych o artykułach z tabeli cdsModifyPricesGrid w celu znalezienia danego produktu po jego nazwie.

Procedura: TfMainPanel.NameSearchingModelInModifyPricesGridChange(Sender: TObject);

Funkcjonalność zakładki „Zamknij aplikacje”:

Zamyka program, zamykając tym samym wszelkie połączenia z bazą.

Procedura: **TfMainPanel.cxCloseAppTabSheetShow(Sender: TObject);**

delayForClosingApp – żeby po wielokrotnym kliknięciu w zakładkę, zamykanie aplikacji (pytanie o zamknięcie itd.) działało poprawnie, tzn. delayForClosingApp > 1 = pytanie i zamknięcie, przechodzimy jednak do innej zakładki, więc delayForClosingApp = 1, oczekiwanie na ponowne kliknięcie w zakładkę „Zamknij aplikacje”.

Na tej samej zasadzie działa procedura uruchamiana w momencie kliknięcia „X” w oknie aplikacji (bez delayForClosingApp):

Procedura: **TfMainPanel.WMSysCommand(var message: TWMSysCommand);**

II. Projekt „Kiermasz” - formatka „uListOfWorkers”:

Formatka zawiera obiekt „fListOfWorkersCxlmaeList” przypisujący przyciskom ikonki graficzne, oraz zawiera także obiekty:

„cdsDataOfChoosedWorker” oraz „dsDataOfChoosedWorker” odpowiadające za przechowywanie danych wyszukanego pracownika z użyciem jego nazwiska.

Formatka wyświetla wszystkich pracowników według wpisanego wcześniej nazwiska.

Funkcjonalność:

Przycisk „Zamknij” (CloseButton):

Zamyka formatkę „uListOfWorkers”

Procedura: **TfListOfWorkers.CloseButtonClick(Sender: TObject);**

Przycisk „Edytuj dane pracownika” (EditDataOfChoosenWorkerButton):

Wyświetlenie formatki „uEditWorkerData” umożliwiającej edycję danych adresowych wybranego w gridzie (w formatce: uListOfWorkers) pracownika.

Procedura: **TfListOfWorkers.EditDataOfChoosenWorkerButtonClick(Sender: TObject);**

Przycisk „Kontynuuj zakupy” (ChooseWorkerButton):

Zapisanie numeru ewidencyjnego wybranego pracownika do pola obok „Dane pracownika” w formatce „uMainPanel” (zakładka „Sprzedaż”). Pole jest dostosowane do skanerów (dane zakończone ENTERem), więc zasymulowanie naciśnięcia klawisza ENTER (Keybd_Event) w ramach wprowadzenia numeru ewidencyjnego do programu.

Procedura: **TfListOfWorkers.ChooseWorkerButtonClick(Sender: TObject);**

III. Projekt „Kiermasz” - formatka „uEditWorkerData”:

Formatka umożliwiająca zmianę danych adresowych wybranego wcześniej w gridzie pracownika w formatce „uListOfWorkers”.

Formatka zawiera tylko jeden obiekt: „fEditWorkerDataCxlImageList” przypisujący przyciskom w tej formatce odpowiednie ikonki graficzne.

Przycisk „Anuluj” (CancelButton):

Zamyka formatkę, nadaje zmiennym odpowiednie wartości, kontynuuje zakupy poprzez nadanie zmiennej fMainPanel.delayInFinalizeSellButtonClickInFMainPanel wartość true.

Dwie procedury posiadają taką samą funkcjonalność:

TfEditWorkerData.WMSysCommand(var message: TWMSysCommand);

TfEditWorkerData.CancelButtonClick(Sender: TObject);

Przycisk „Zapisz zmiany” (SaveChangesButton):

Zapisanie w tabeli fMainPanel.cdsAllWorkersData przechowującej dane pracowników zmienione wartości dla adresu. Procedura wykonuje się w zależności, skąd formatka „uEditWorkerData” została wywołana.

Procedura: **TfEditWorkerData.SaveChangesButtonClick(Sender: TObject);**

Szczegóły w kodzie źródłowym.

IV. Projekt „Kiermasz” - formatka „ulInfoBox”:

Formatka wyświetlana jedynie przy starcie programu w ramach poinformowania użytkownika, że program wykonuje się i pobiera dane, więc trzeba poczekać.

Formatka użyta w procedurach:

procedure TfMainPanel.FormShow(Sender: TObject);

procedure TfMainPanel.LoadCSVFileButtonClick(Sender: TObject);

V. Projekt „Kiermasz” - wątek „GetWorkersDataThread”

Wątek wykorzystywany jest do pobrania danych wszystkich pracowników podczas uruchomienia aplikacji.

Procedura:

tGetWorkersDataThread.Execute

Główna procedura wątku tworząca obiekty niezbędne do pobrania danych wszystkich pracowników.

Zmienna „tSuccess” - określa, czy główne zadanie wątku zostało wykonane (tym samym cały wątek).

Po zakończeniu pracy następuje synchronizacja z drugą procedurą

Synchronize(SYNC_OnFinished);

Procedura:

tGetWorkersDataThread.SYNC_OnFinished

Druga procedura uruchamiana po zakończeniu się głównej procedury.

Następuje zapisanie pobranych danych do Client Data Source znajdującego się w głównej formacie „uMainPanel”

Dane zapisane w tabeli: „cdsAllWorkersData” wykorzystywane są potem podczas pracy z programem.

Na sam koniec obiekty stworzone przez wątek na czas jego pracy są zamykane.

VI. Projekt „Kiermasz” - wątek „ImportDataFromCSVToDBThread”

Wątek importujący dane z pliku CSV do bazy danych (artykuły), na podstawie których ma odbywać się kiermasz.

Wątek użyty w procedurze: **TfMainPanel.LoadCSVFileButtonClick(Sender: TObject);**

Procedura:

tlImportDataFromCSVToDBThread.Execute

Przygotowanie wszystkich obiektów bazo-danowych niezbędnych do dalszego zaimportowania danych z pliku CSV do bazy danych.

Procedura:

tlImportDataFromCSVToDBThread.SYNC_OnFinished

Szczegóły w kodzie źródłowym.