

**Skrypty obsługi bazy danych. Przykład zastosowania
do tworzenia własnego systemu receptur lub
zdarzeniowego zapisu parametrów do bazy danych w
systemie *asix3*.**

Pomoc techniczna

*Dok. Nr PLP0003
Wersja: 23-11-2005*

ASKOM[®] to zastrzeżony znak firmy ASKOM Sp. z o. o., Gliwice. Inne występujące w tekście znaki firmowe bądź towarowe są zastrzeżonymi znakami ich właścicieli.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną lub inną powoduje naruszenie praw autorskich niniejszej publikacji.

ASKOM Sp. z o. o. nie bierze żadnej odpowiedzialności za jakiegokolwiek szkody wynikłe z wykorzystywania zawartych w publikacji treści.

Copyright © 2005, ASKOM Sp. z o. o., Gliwice



ASKOM Sp. z o. o., ul. Józefa Sowińskiego 13, 44-121 Gliwice,
tel. +48 (0) 32 3018100, fax +48 (0) 32 3018101,
<http://www.askom.com.pl>, e-mail: office@askom.com.pl

1. Wstęp

Niniejszy dokument opisuje możliwości wykorzystania modułu skryptów do obsługi baz danych w systemie **asix**. Z dokumentem związane są pliki składające się na przykładową aplikację, przykładowe skrypty w języku VBScript oraz program TableView służący do przeglądania i edycji baz danych (pliki można pobrać z witryny ASKOM [asix_Skrypty_Bazy_Danych.zip](#)). Dostęp do baz danych opiera się na technologii Microsoft® ActiveX® Data Objects (ADO).

Opisywane poniżej skrypty prezentują sposób budowy własnego systemu receptur lub zdarzeniowego zapisu parametrów do bazy danych - ze wskazaniem do zastosowania w systemie **asix3**. Najnowsza wersja pakietu - **asix4** posiada narzędzie znacznie ułatwiające projektowanie baz danych, ukierunkowanych na definiowanie i wybór receptur oraz archiwizację informacji procesowych przy pomocy zestawu predefiniowanych operacji i okien dialogowych. Jest to moduł **AsBase**, którego obsługa nie wymaga od projektanta przygotowania informatycznego.

2. Instalacja

Wszystkie pliki przykładu (wraz z programem TableView) należy przekopiować do wybranego katalogu. Program TableView korzysta z bibliotek zawartych w pakiecie **asix**. W związku z tym wymagane jest uprzednie zainstalowanie tego pakietu. Do uruchomienia przykładowej aplikacji wymagane jest oprogramowanie MDAC (Microsoft Data Access Components).

3. Program TableView

Do plików przykładu dołączony został program TableView¹, który jest prostym programem służącym do przeglądania i edycji baz danych. Program TableView umożliwia dostęp do jednej wybranej tablicy bazy danych. Uruchomienie programu powoduje otwarcie jednego lub kilku okien dialogowych pozwalających na wybór bazy danych oraz tabeli danych.

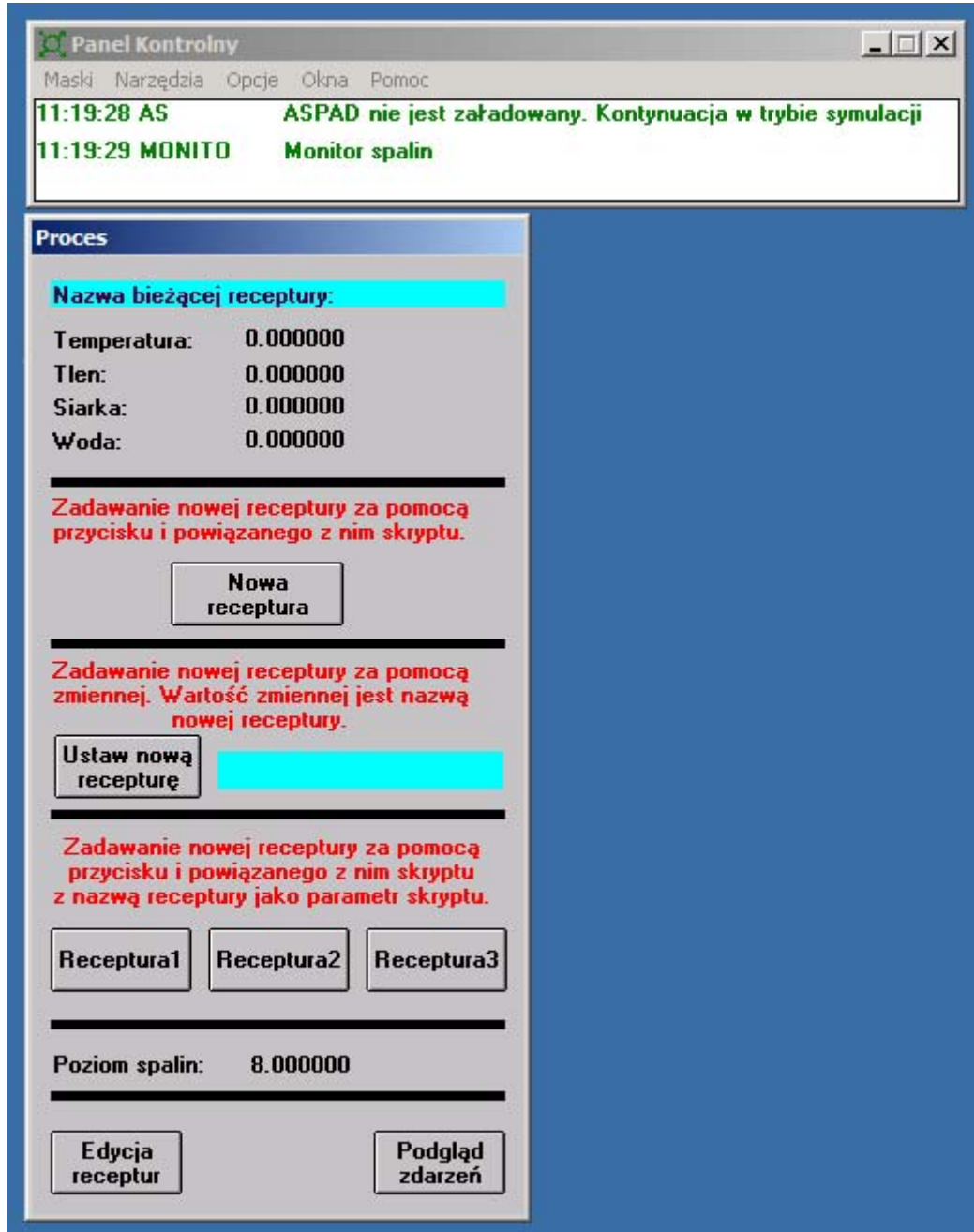
4. Receptury

Receptury wykorzystywane często w procesie szarżowym lub do automatycznego przestrajania układów regulacji można obsługiwać implementując proste skrypty napisane w VBS.

¹ Aktualnie narzędziem umożliwiającym tworzenie, edycję i przeglądanie bazy danych SQL jest AsBase – narzędzie dostępne w pakiecie **asix4**.

4.1. Przykładowa aplikacja

Z niniejszym dokumentem związana jest przykładowa aplikacja systemu **asix** o nazwie **Proces**.



Rysunek 1. Okno główne przykładowej aplikacji 'Proces'.

Aplikacja wyświetla wartości zmiennych procesowych o nazwach: Temperatura, Tlen, Siarka, Woda oraz Spaliny. Cztery pierwsze zmienne określają nastawy (receptura) hipotetycznego procesu produkcyjnego. Zmienna Spaliny to dodatkowy pomiar związany z procesem produkcyjnym. Pomiar Spaliny jest symulowany – określająca go zmienna ma przebieg piłokształtny o zakresie od 0 do 10. Dodatkowo zdefiniowane są zmienne AktRecept (aktualna receptura) oraz NowaRecept

(nowa receptura). Zmienne te są wykorzystywane do wyświetlania nazwy aktualnej receptury procesu technologicznego (AktRecept) oraz nazwy nowej receptury wprowadzanej przez operatora (NowaRecept).

Wartości zmiennych składających się na recepturę są pobierane z bazy danych na żądanie operatora, a następnie są przekazywane do elementów wykonawczych procesu (np. do sterownika). Każdy rekord bazy danych receptur składa się z nazwy receptury oraz wartości nastaw dla temperatury, tlenu, wody i siarki. Dodatkowo rekord zawiera pole o nazwie Uwagi, w którym można umieścić opis poszczególnych receptur.

Oprócz wizualizacji stanu procesu oraz zadawania nowych receptur, przykładowa aplikacja pokazuje, w jaki sposób można przy użyciu skryptów zapisywać w bazie danych informacje o ważnych zdarzeniach. Jeśli wartość spalin przekroczy liczbę 9, to do bazy danych zdarzeń zapisywany jest rekord zawierający wartość spalin, aktualne wartości nastaw dla tlenu, siarki, wody i temperatury, nazwę bieżącej receptury. Dodatkowo wartości spalin są uśredniane i co jedną godzinę zapisywany jest rekord do bazy zdarzeń zawierający informacje o średniej wartości spalin w ciągu ostatniej godziny. Każdy rekord bazy zawiera pole o nazwie Uwagi z tekstem określającym zdarzenie, którego dotyczy (tj. przekroczenie limitu i obliczenie średniej).

4.2. Bazy danych

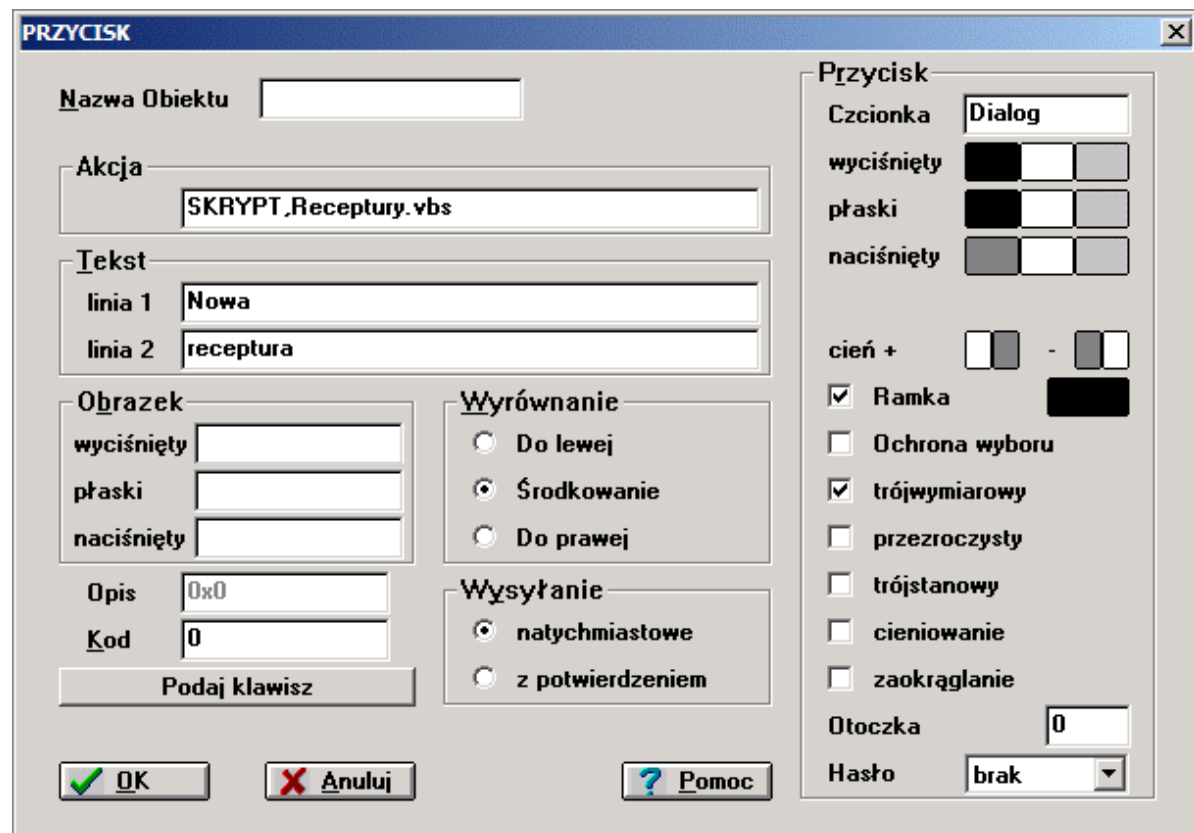
Aplikacja operuje na bazie danych Receptury zawierającej tabelę o nazwie Receptury oraz na bazie o nazwie Zdarzenia zawierającej tabelę o nazwie Zdarzenia (obie tabele można również umieścić w jednej bazie danych). Skrypty zawierają definicję „łańcucha dostępu” do bazy danych, który jednoznacznie określa położenie i nazwę bazy danych. „Łańcuch dostępu” należy zmodyfikować stosownie do konfiguracji systemu. Skrypty przykładowej aplikacji operują na bazie Microsoft Access. Poprzez odpowiednie zmodyfikowanie „łańcuchów dostępu” mogą również korzystać z innych baz danych. W szczególności, skrypty zawierają definicje dostępu do bazy danych SQL. Są one jednak umieszczone w komentarzach i do ich aktywacji należy usunąć znak komentarza. Do korzystania z baz danych SQL wymagane jest odpowiednie oprogramowanie (np. Microsoft Data Engine (MSDE), Microsoft SQL server). Oprogramowanie do baz danych Microsoft Access jest zwykle zainstalowane (nawet jeśli nie jest zainstalowany Microsoft Access). Jest ono zawarte w oprogramowaniu MDAC (Microsoft Access Components), które zawiera również technologię ADO wymaganą do uruchomienia przykładowej aplikacji.

Przykładowa aplikacja zawiera skrypty *NowaBaza.vbs* oraz *NowaBazaSQL.vbs*, które pokazują, w jaki sposób można utworzyć bazę za pomocą skryptu. Pierwszy skrypt tworzy bazę danych Microsoft Access, a drugi bazę SQL. Oba skrypty są bardzo podobne. Różnice dotyczą łańcuchów dostępu oraz wstępnej fazy tworzenia bazy. Oddzielny skrypt dla bazy SQL został stworzony, ponieważ oprogramowanie SQL nie posiada zaimplementowanych niektórych funkcji tworzenia baz danych (baza danych SQL tworzona jest za pomocą komendy języka DDL). Oba skrypty uruchamia się poza systemem **asix**.

4.3. Wprowadzanie nowych receptur

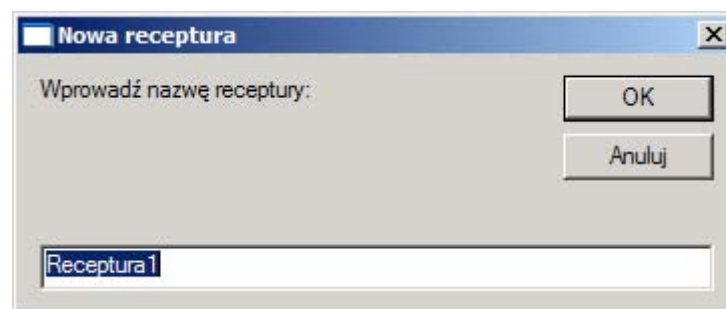
W przykładowej aplikacji pokazano dwie metody wprowadzania nazwy receptury.

Pierwsza metoda polega na wykorzystaniu akcji operatorskiej o nazwie SCRIPT. Przyciśnięcie odpowiedniego przycisku na masce technologicznej powoduje uruchomienie akcji operatorskiej polegającej na uruchomieniu skryptu.



Rysunek 2. Parametryzacja obiektu PRZYCISK uruchamiającego skrypt Receptury.vbs.

Program skryptu otwiera okno dialogowe pozwalające na wprowadzenie przez operatora nazwy receptury. Nazwa ta jest następnie wykorzystana do odczytu odpowiedniego rekordu z bazy receptur. Nastawy z tego rekordu są następnie przez skrypt przekazywane do elementów sterujących procesu technologicznego.



Rysunek 3. Okno służące do deklarowania nazwy receptury.

Poprzez odpowiednie sparometryzowanie akcji operatorskiej SCRIPT można również spowodować załadowanie odpowiedniej receptury bez wyświetlania okna dialogowego. Można np. umieścić na masce kilka przycisków, z których każdy powoduje załadowanie określonej receptury. Nazwa receptury może być umieszczona w parametrach akcji operatorskiej SCRIPT.

PRZYCISK

Nazwa Obiektu

Akcja

Tekst

linia 1

linia 2

Obrazek

wyciśnięty

płaski

naciśnięty

Opis

Kod

Podaj klawisz

Wyrównanie

Do lewej

Środkowanie

Do prawej

Wysyłanie

natychmiastowe

z potwierdzeniem

Przycisk

Czcionka

wyciśnięty

płaski

naciśnięty

cień + -

Ramka

Ochrona wyboru

trójwymiarowy

przezroczysty

trójstanowy

cieniowanie

zaokrąglanie

Otoczka

Hasło

Rysunek 4. Przykładowa parametryzacja przycisku powodującego załadowanie określonej receptury.

Druga metoda uruchamiania nowej receptury polega na wykorzystaniu zmiennej tekstowej zdefiniowanej w kanale NONE (zmienna NowaRecept). Operator wprowadza nazwę nowej receptury w taki sam sposób, w jaki wysyła się sterowania do procesu technologicznego. Odpowiedni skrypt monitoruje zmienną NowaRecept i po wykryciu jej zmiany, uruchamia nową recepturę.

The screenshot shows the 'PRZYCISK' dialog box. It has a title bar with a close button. The main area is divided into several sections:

- Nazwa Obiektu:** An empty text field.
- Akcja:** A text field containing 'WYŚLIJ_STEROWANIA'.
- Tekst:** Two text fields: 'linia 1' with 'Ustaw nową' and 'linia 2' with 'recepturę'.
- Obrazek:** Three color selection boxes for 'wyciśnięty', 'płaski', and 'naciśnięty'.
- Wyrównanie:** Radio buttons for 'Do lewej', 'Środkowanie' (selected), and 'Do prawej'.
- Wysyłanie:** Radio buttons for 'natychmiastowe' (selected) and 'z potwierdzeniem'.
- Przycisk:** A section with 'Czcionka' set to 'Dialog', and three color selection boxes for 'wyciśnięty', 'płaski', and 'naciśnięty'. Below these are two more color selection boxes for 'cień +' and 'cień -'. There are several checkboxes: 'Ramka' (checked), 'Ochrona wyboru' (unchecked), 'trójwymiarowy' (checked), 'przezroczysty' (unchecked), 'trójstanowy' (unchecked), 'cieniowanie' (unchecked), and 'zaokrąglenie' (unchecked). At the bottom of this section are 'Otoczka' (set to 0) and 'Hasło' (set to 'brak').

At the bottom of the dialog are three buttons: 'OK' (with a green checkmark), 'Anuluj' (with a red X), and 'Pomoc' (with a blue question mark).

Rysunek 5. Parametryzacja obiektu PRZYCISK z akcją WYŚLIJ_STEROWANIA.

The screenshot shows the 'NAPIS' dialog box. It has a title bar with a close button. The main area is divided into several sections:

- Nazwa Obiektu:** An empty text field.
- Parametry:** Two text fields for 'Czcionka m' and 'Czcionka s', both set to 'Dialog'. Below are four color selection boxes for characters '=', '?', '^', and '->'. There is a checkbox for 'Tło koloru Okna' which is unchecked.
- Edycja:** Checkboxes for 'Autom. edycja' (checked), 'Inicjacja poprzednią' (checked), 'Inicjacja bieżącą' (unchecked), and 'Autom. wysłanie' (unchecked).
- Wyrównanie:** Radio buttons for 'Do lewej', 'Środkowanie' (selected), and 'Do prawej'.
- Dana monitorowana:** A text field containing 'NowaRecept'. There is a checkbox for 'bez monitorowania' which is unchecked.
- Dana sterowana:** A text field containing 'NowaRecept'. There is a checkbox for 'ze sterowaniem' which is checked.
- Hasło:** A dropdown menu set to 'brak'.
- błędy komunikacji:** A checkbox which is checked.
- Znak kontrolny:** A text field containing '?'.
- Długość:** A text field containing '20'.

At the bottom of the dialog are three buttons: 'OK' (with a green checkmark), 'Anuluj' (with a red X), and 'Łącze' (with a lightning bolt icon). There is also a 'Pomoc' button (with a blue question mark).

Rysunek 6. Parametryzacja obiektu NAPIS.

4.4. Lista plików przykładowej aplikacji

Lista plików przykładowej aplikacji:

Masks\Receptury.vbs – pozwala wybrać recepturę za pomocą okna dialogowego;

Skrypt RECEPTURY.VBS pozwala nadać zmiennym wartości pobrane z wybranego rekordu bazy receptur. RECEPTURY.VBS można wywołać bez parametrów, co spowoduje wyświetlenie okienka dialogowego do wprowadzania nazwy żądanej receptury, lub z podaniem jako parametr nazwy receptury - co spowoduje jej natychmiastowe załadowanie.

Receptury2.vbs – wybór receptury za pośrednictwem zmiennej **asixa**;
Monitor.vbs – skrypt monitoruje poziom spalin i po przekroczeniu dopuszczalnej wartości zapisuje informacje do bazy zdarzeń;
NowaBaza.vbs – tworzenie bazy Microsoft Access;
NowaBazaSQL.vbs – tworzenie bazy SQL;
A dovbs.inc – skrypt zawierający definicje symboli wykorzystywanych w innych skryptach;
Proces.ini – plik inicjalizacyjny aplikacji;
Masks\Proces.msk – maska procesowa;
Proces.Asmen – definicje zmiennych;
Receptury.mdb – baza danych receptur;
Zdarzenia.mdb – baza danych zdarzeń;
TableView.exe – program do przeglądu i edycji baz danych;
TableView.dat – przykładowy plik konfiguracyjny programu TableView.

Zdarzeniowy zapis do bazy umożliwia skrypt MONITOR.VBS, który śledzi wartości zmiennej *Spaliny* i w przypadku spełnienia założonych warunków (tu: przekroczenie zadanej wartości progowej) zapisuje do tabeli *Zdarzenia* wartości zmiennych pobrane z *ASMENA*.

Skrypty dołączone do przykładu nie są skomplikowane i podstawowa znajomość języka Visual Basic Script pozwala na ich dostosowanie do własnych potrzeb.

4.5. Przeglądanie i modyfikowanie bazy receptur oraz logu

Zapisane w bazie rekordy (*Receptury* i *Zdarzenia*) można przeglądać przy pomocy programu TableView. Jest oczywistym, że w przypadku przeglądania tabeli *Zdarzenia* zablokowana musi być możliwość edycji zawartości pól w rekordach.

Podczas przeglądania tabeli **Receptury** możliwe jest wprowadzanie zmian do zawartości bazy, jak również wprowadzanie nowych receptur.

5. Spis rysunków

<i>Rysunek 1. Okno główne przykładowej aplikacji 'Proces'.</i>	4
<i>Rysunek 2. Parametryzacja obiektu PRZYCISK uruchamiającego skrypt Receptury.vbs.</i>	6
<i>Rysunek 3. Okno służące do deklarowania nazwy receptury.</i>	6
<i>Rysunek 4. Przykładowa parametryzacja przycisku powodującego załadowanie określonej receptury.</i>	7
<i>Rysunek 5. Parametryzacja obiektu PRZYCISK z akcją WYŚLIJ_STEROWANIA.</i>	8
<i>Rysunek 6. Parametryzacja obiektu NAPIS.</i>	8

Spis treści

1. WSTĘP.....	3
2. INSTALACJA	3
3. PROGRAM TABLEVIEW	3
4. RECEPTURY	3
4.1. PRZYKŁADOWA APLIKACJA	4
4.2. BAZY DANYCH.....	5
4.3. WPROWADZANIE NOWYCH RECEPTUR	5
4.4. LISTA PLIKÓW PRZYKŁADOWEJ APLIKACJI.....	8
4.5. PRZEGLĄDANIE I MODYFIKOWANIE BAZY RECEPTUR ORAZ LOGU	9
5. SPIS RYSUNKÓW.....	11