

*Wojciech Mueller, Piotr Rogacki, Jerzy Weres, Marcin Jarysz
Instytut Inżynierii Rolniczej
Akademia Rolnicza w Poznaniu*

INTRANETOWY, BAZODANOWY SYSTEM WSPOMAGAJĄCY ZARZĄDZANIE BADANIAM W RAMACH INŻYNIERII ROLNICZEJ

Streszczenie

Działalność ośrodków badawczych zresztą jak innych podmiotów instytucjonalnych i gospodarczych jest już praktycznie niemożliwa bez wspierających go systemów informatycznych. Specyfika działań ośrodków badawczych w obszarze inżynierii rolniczej w połączeniu z dużą mobilnością pracowników wymaga wykorzystywania do ich funkcjonowania specjalistycznych aplikacji. Brak tego typu narzędzi, spełniających postawione kryteria skłonił autorów do zaprojektowania i wytworzenia intranetowego systemu informatycznego, opartego o najnowszą technologię .NET, wspomagającego kompleksowe zarządzanie badaniami.

Słowa kluczowe: zarządzanie badaniami, system informatyczny, ASP.NET

Wprowadzenie

Proces zarządzania ośrodkami badawczymi z uwagi na swoją złożoność i różnorodność przetwarzanych informacji przebiega sprawniej, jeżeli jest wspierany przez specjalistyczny system informatyczny. Ów wspomniany proces obejmuje zróżnicowane obszary, a zatem system informatyczny wspomagający go winien mieć budowę modułową. Zaprojektowana i wytworzona przez autorów aplikacja intranetowa usprawniająca zarządzanie badaniami w ramach inżynierii rolniczej spełnia powyższy postulat. Istotnym podsystemem prezentowanej aplikacji jest moduł odpowiadający za kontrolowanie newralgicznego obszaru funkcjonowania instytucji, jakim są środki finansowe.

Do wytworzenia tak potrzebnego i specjalistycznego systemu informatycznego wykorzystano najnowsze technologie kryjące się za nazwą .NET jakimi są ASP.NET oraz ADO.NET.

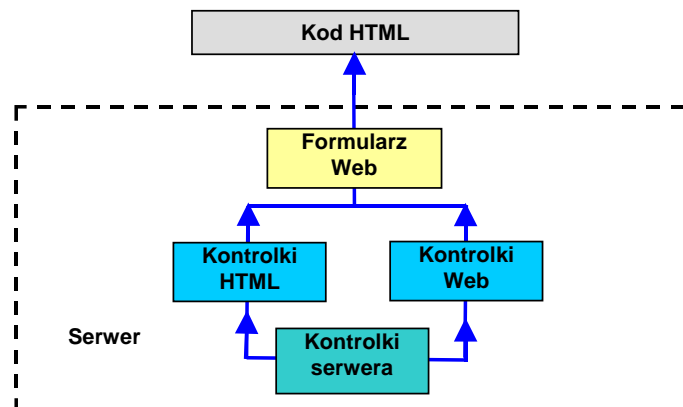
Platforma ASP.NET

ASP.NET to nowa platforma programistyczna, wkomponowana w środowisko .NET pozwalająca na tworzenie skomplikowanych, rozbudowanych internetowych systemów informatycznych. Stanowi ona kontynuację poprzedniczki ASP [Payne 2002], aczkolwiek poczynione zmiany wskazują, iż mamy do czynienia z technologią nowej generacji. Charakter i zakres dokonanych zmian sugeruje, zatem dalszy dynamiczny rozwój aplikacji internetowych. Obszarem o potencjalnie dużym zapotrzebowaniu na tego typu systemy informatyczne z uwagi na rozproszenie użytkowników jest sektor rolniczy.

Struktura aplikacji wytwarzanych z wykorzystaniem technologii ASP .NET ma charakter trójwarstwowy, co w przypadku internetowych systemów informatycznych staje się obecnie obowiązującym standardem. Pierwszą warstwę będącą interfejsem użytkownika stanowi dowolna przeglądarka internetowa, co oznacza iż proponowana technologia nie wprowadza w tym obszarze żadnych istotnych ograniczeń. Przetwarzanie danych zawartych w formularzach, jaki i uzyskanych z SZBD odbywa się po stronie serwera, a produktem końcowym owego procesu jest dokument HTML przesłany przeglądarce.

W warstwie drugiej, najważniejszym z punktu widzenia wytwarzania prezentowanego typu systemów informatycznych, a zarazem najbardziej rozbudowanym pakietem jest Web Forms. Tworzy on model programistyczny, za pomocą którego możemy budować programowalne strony WWW. Ta nowa technologia programowania po stronie serwera wraz z zlokalizowaną tam całą logiką aplikacji, pozwala na wydajne tworzenie szybkich internetowych systemów informatycznych. Wykorzystane w procesie budowy aplikacji kontrolki – obiekty Web Forms [Payne 2002] wysyłają podczas działania systemu znaczniki HTML (rys. 1), które są zrozumiałe dla przeglądarki. Jeżeli formularze Web Forms są tworzone w środowisku Visual Studio .NET, to kod programu (znajdujący się w pliku .vb) jest zawsze oddzielony od dokumentu .aspx, zawierającego interfejs użytkownika. Pliki te są łączone w procesie kompilacji, przebiegającej podczas uruchamiania aplikacji. Formularze Web zapewniają dwustronną komunikację z użytkownikiem, za pomocą dwóch zestawów kontrolki serwera: kontrolki HTML oraz kontrolki Web (oba wykorzystują HTML do wyświetlania ich przez przeglądarki internetowe).

Najczęściej wykorzystywane są kontrolki Web. Obejmują one tradycyjne kontrolki formularzy takie jak Label, TextBox, Button, RadioButton jak i inne o wyższym poziomie złożoności takie jak Calendar, DataGrid itp. Szczególnie przydatną kontrolką pozwalającą na prezentację informacji pochodzących z baz danych jest DataGrid. Warty podkreślenia jest fakt, iż kontrolki Web umożliwiają automatyczne dostosowanie interfejsu do możliwości przeglądarki klienta.



Rys. 1. Schemat generowania kontrolek serwera na stronie www
 Fig. 1. Server controls generation scheme (website)

W trzeciej warstwie struktury aplikacji najważniejszą rolę pełni ADO .NET – model programistyczny dostępu do danych [Worley 2003]. Aczkolwiek obiekty ADO .NET nie posiadają struktury hierarchicznej, to jednak etap pobierania danych ze źródła wymaga określonego użycia obiektów wraz z ich metodami. Operacji tej nie uda się zrealizować bez obiektu Connection, który pozwala na połączenie ze źródłem danych. W następnej kolejności wykorzystujemy obiekt Command lub DataAdapter. Rodzaj przeprowadzanych operacji na uzyskanych rekordach przesądza o zastosowaniu w dalszej kolejności obiektów DataReader lub DataSet.

Projektowanie

Dziedziną problemową prezentowanego system informatycznego jest zarządzanie badaniami w ramach inżynierii rolniczej. Ten sam obszar rzeczywistości obejmuje aplikacja wspomagająca dokumentowanie badań prowadzonych przez instytut, która była prezentowana przez autorów wcześniej. Zatem celowym stała się, na etapie projektowania jak i późniejszego wytwarzania, integracja obu systemów. Dotyczyła ona głównie trzech obszarów:

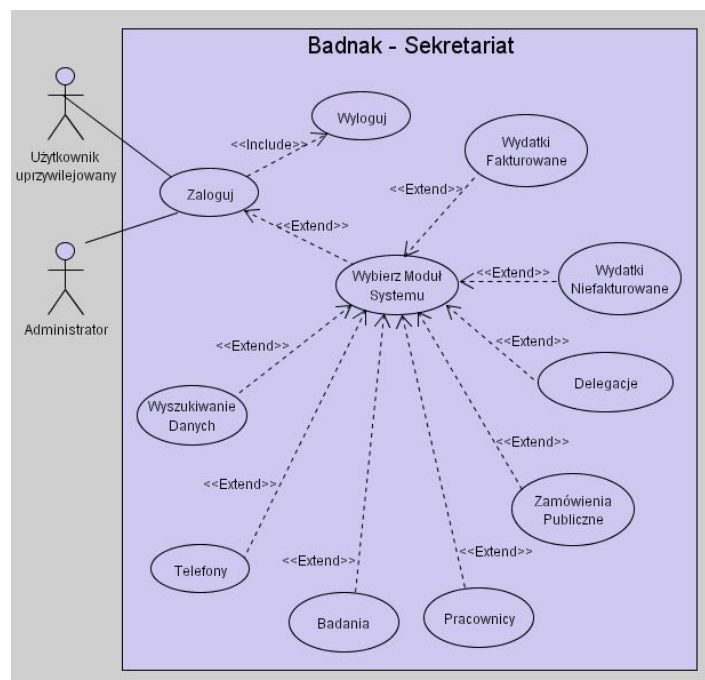
- poziomu programistycznego, obejmującego rodzaj aplikacji jak i użytych technologii,
- systemu bazodanowego,
- interfejsu użytkownika.

Spodziewanymi efektami przyjętego rozwiązania miały być:

- przyspieszenie wytwarzania aplikacji,
- uproszczenie pielęgnacji systemu,
- ułatwienie obsługi.

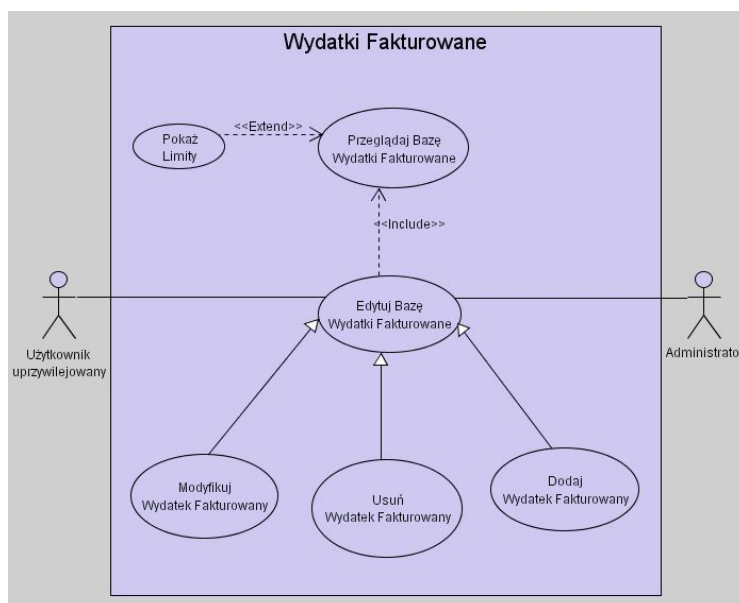
Prezentowany podsystem informatyczny wspomagający zarządzanie środkami finansowymi przeznaczonymi na badania - podobnie jak moduł prezentowany wcześniej – został zaprojektowany w notacji graficznej UML [Booch i in. 2000], zgodnie z zasadami inżynierii oprogramowania. Wysoki stopień złożoności dziedziny problemowej sprawił, iż proces wytwarzania aplikacji podporządkowano modelowi „realizacja przyrostowa”. Obranie takiego modelu cyklu życia oprogramowania miało na celu wyeliminowanie błędów wynikających z możliwości nieprecyzyjnego zidentyfikowania wymagań przy jednoczesnym udostępnieniu użytkownikowi poprawnych funkcji systemu.

Do wytworzenia modeli obiektowych na etapie projektowania wykorzystano narzędzie o stosunkowo dużych możliwościach jakim jest Visual Paradigm. Założony przez autorów wysoki stopień unifikacji obu modułów, zaowocował podobnymi diagramami przypadków użycia. Różnice występują na poziomie definicji scenariuszy tworzących wybrane przypadki użycia. Najwyższy w hierarchii, a zatem najogólniejszy diagram przedstawiony został na (rys. 2).



Rys. 2. Ogólny diagram przypadków użycia
Fig. 2. General diagram of case usage

Dalsze działania polegające na dekompozycji zidentyfikowanych ogólnych przypadków użycia doprowadziły do uzyskania diagramów niższego poziomu (rys. 3) o wystarczającym stopniu szczegółowości, celem realizacji późniejszego procesu implementacyjnego.

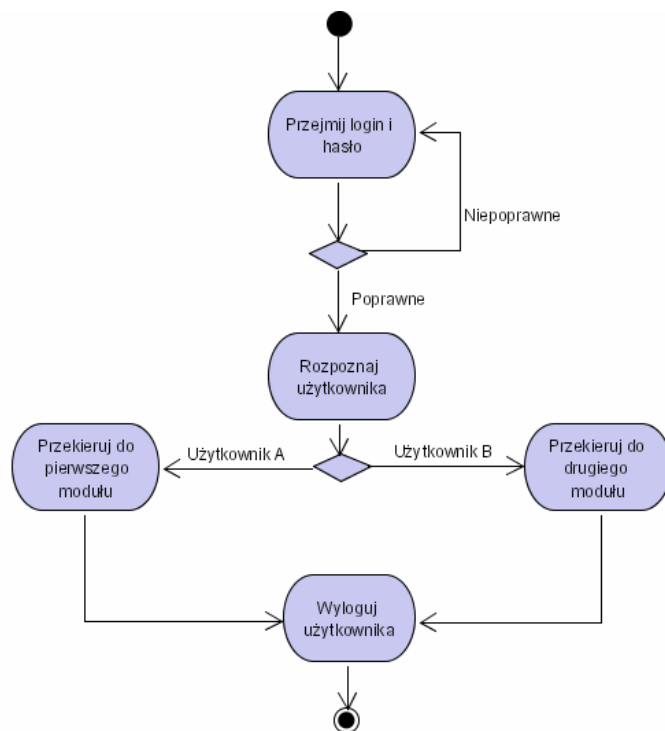


Rys. 3. Przykład diagramu przypadków użycia dla segmentu „Wydatki Fakturowane”

Fig. 3. A diagram of case usage for the segment of “Invoiced Expenses”

Użytkownik uzyskuje dostęp do aplikacji na podstawie nazwy użytkownika i hasła, co określa w sposób jednoznaczny zadania, jakie pełni on w systemie. Rozwiązanie tego problemu programistycznego zaproponowane przez autorów w postaci diagramów czynności przedstawia (rys. 4).

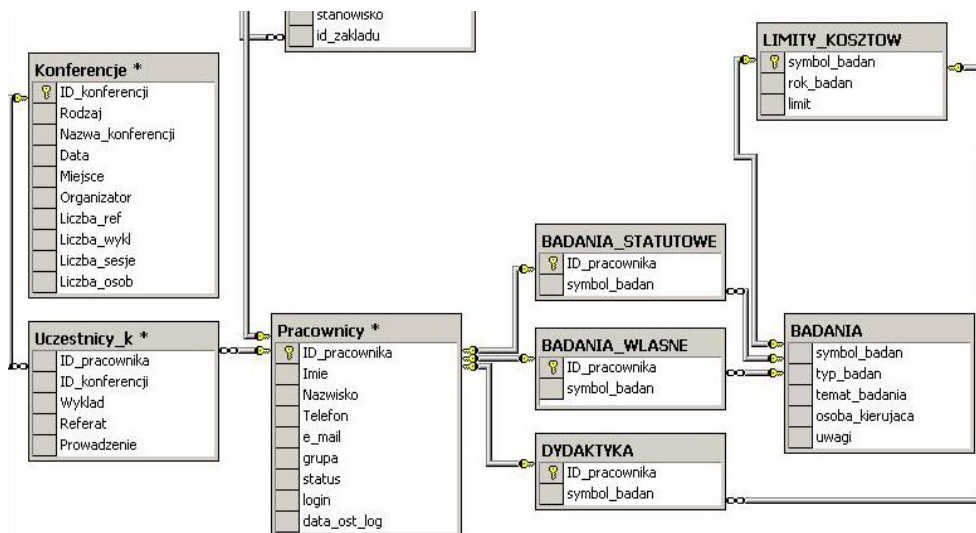
Przyjęta dziedzina problemowa tworzonej aplikacji oraz względy bezpieczeństwa narzucały wykorzystanie profesjonalnego SZBD. Rodzaj gromadzonych informacji niezbędnych dla funkcjonowania systemu uzyskany na etapie modelowania obiektowego przesądził o wyborze relacyjnej bazy danych. Względę ekonomiczne zadecydowały, iż przyjęto jako SZBD SQL SEWER 2000.



Rys. 4. Diagram czynności – rozpoznanie i przekierowanie użytkownika
Fig. 4. Activity diagram – user identification & transfer

Kolejne etapy projektowania baz danych, występujące po modelowaniu koncepcyjnym będące fazami opracowywania całej aplikacji przebiegały przy użyciu narzędzi programistycznych wybudowanych w SQL Serwer [Griffin 2002]. Wcześniejsza faza wyboru konkretnego SZBD spowodowała, iż kolejne etapy projektowania bazy; modelowanie logiczne i fizyczne przebiegały jednocześnie [Waymire, Sawtell 2002]. Uzyskane w ten sposób struktury relacyjne przedstawia (rys. 5).

Tworzenie tabel i powiązań między nimi realizowano równolegle dla całej aplikacji, która jak sygnalizowano powyżej tworzą dwa podsystemy, bazujące częściowo na tych samych strukturach relacyjnych.



Rys. 5. Fragment struktury bazy danych
 Fig. 5. A part of database structure

Narzędzia programistyczne

Proces tworzenia oprogramowania realizowano w środowisku Visual Studio .Net, o czym zdecydowały cechy tego narzędzia takie jak:

- wizualne projektowanie,
- duże podobieństwo budowania programowalnych stron WWW do tworzenia aplikacji okienkowych,
- udostępnianie widoku kodu wyjściowego HTML,
- rozbudowany system pomocy.

To kompletne, wizualne środowisko tworzenia aplikacji w połączeniu z platformą FrameWork pozwala na wykorzystanie wielu języków programowania: C++, C#, oraz Visual Basic .NET. Kontrolki używane w tym środowisku są wspólne dla wszystkich języków programowania, co umożliwia wytworzenie zarówno aplikacji opartych o formularze Windows jak i aplikacji internetowych opartych o interfejs przeglądarki.

Szybkość oraz skuteczność tworzenia aplikacji wynika z tego, że VS .NET dysponuje bardzo dużą liczbą narzędzi wspomagających ten proces. Należy jednak wspomnieć, że do tworzenia aplikacji .NET nie jest niezbędnym posiadanie wyżej wspomnianego oprogramowania.

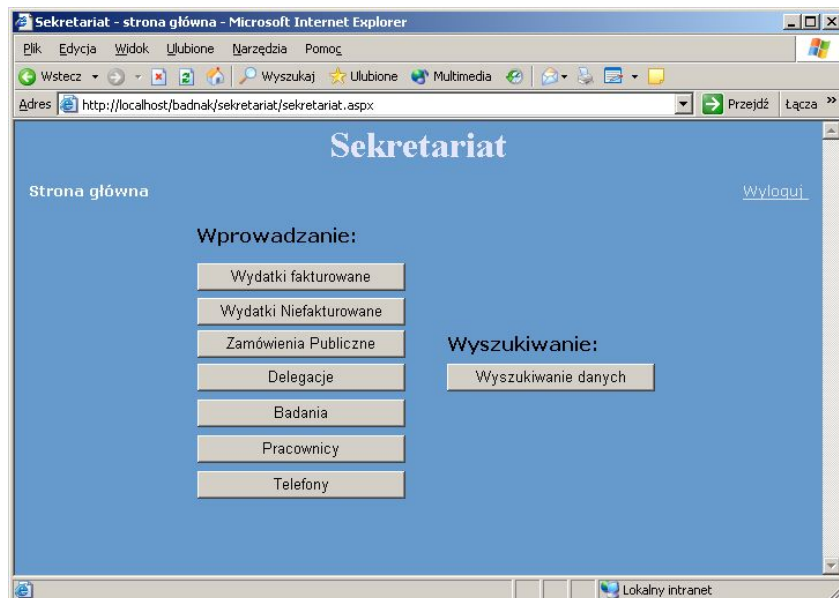
Niezbędne minimum stanowi zainstalowane w systemie środowisko .NET Framework Software Development Kit oraz najprostszy edytor tekstu. .NET SDK zawiera .NET Framework, które jest niezbędne do wykonywania aplikacji .NET oraz wszystko to, co jest potrzebne do testowania i wdrażania systemu. Ponadto dla mniej zaawansowanych użytkowników firma Microsoft zaproponowała narzędzia serii Express. Są to łatwe do opanowania i przyjazne w użytkowaniu środowiska wizualne, pozbawione jednak wielu funkcji VS .NET. Godną uwagi propozycją jest również kolejne dostępne narzędzie - Web Matrix. Aplikacja ta jest darmowa i zapewnia równie szybkie i wydajne tworzenie aplikacji. Pozbawiona została jednak bardzo istotnej cechy programowania wizualnego w .NET – oddzielenia kodu interfejsu użytkownika od kodu aplikacji. Zupełnie odrębnym projektem jest platforma programistyczna o otwartym dostępie do kodu źródłowego – Mono. Tworzona jest przez programistów z całego świata, a głównym sponsorem projektu jest firma Novell. W przyszłości ma zachować całkowitą kompatybilność z .NET. W chwili obecnej już można tworzyć aplikacje w języku C#. Środowisko to bardzo szybko się rozwija i prawdopodobnie już w niedalekiej przyszłości będzie możliwe tworzenie najbardziej skomplikowanych systemów.

System informatyczny

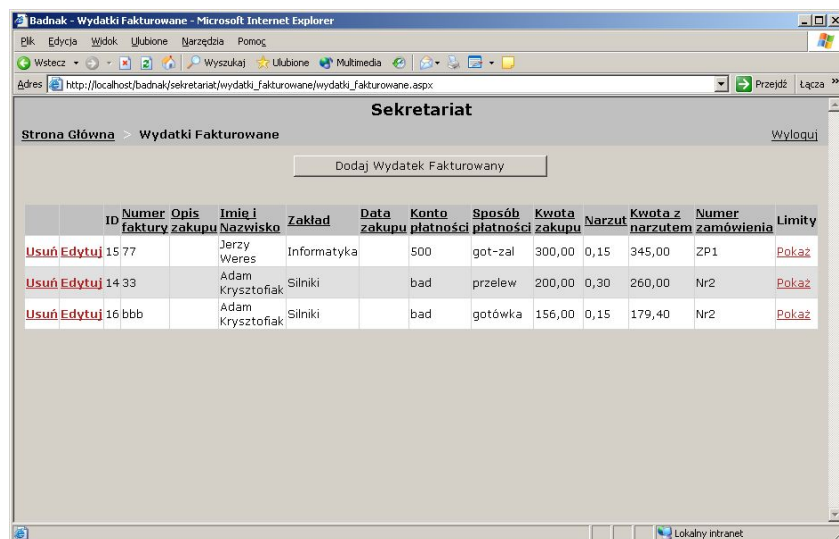
Omawiany podsystem tworzą hierarchicznie uporządkowane formularze, dedykowane określonym grupom użytkowników. Stają się one dostępne dopiero po właściwym rozpoznaniu użytkownika, co ma miejsce na stronie głównej wytworzonej aplikacji. Obszar finansowy jako jeden z newralgicznych punktów funkcjonowania zarówno przedsiębiorstw jak i instytucji wymaga bardzo precyzyjnego zdefiniowania użytkowników, grup i przyznawanych im uprawnień.

Pozytywne przejście etapu identyfikacji udostępnia użytkownikowi panel sterowania (rys. 6) w postaci pogrupowanych przycisków, za którymi kryją się kolejne formularze, zawierające potrzebne dane pobrane z SZBD lub pozwalające na wprowadzenie nowej porcji informacji. Udostępniane dane są prezentowane w postaci tabelarycznych formularzy zintegrowanych z przyciskami akcji (rys. 7), których rodzaj zależy od posiadanych przez użytkownika uprawnień. Zlokalizowanie przycisków akcji w osobnych kolumnach tabeli eliminuje proces selekcji rekordu podlegającego usuwaniu bądź modyfikacji.

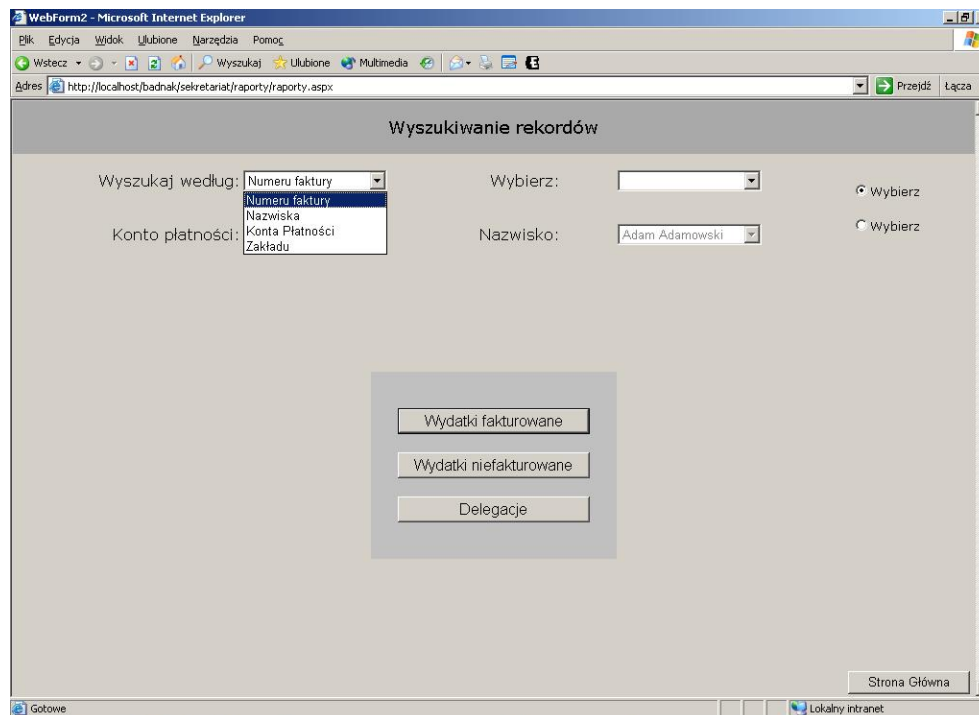
Proces wyszukiwania danych osiągany jest poprzez generowanie odpowiednich zapytań SQL skierowanych do SZBD. Budowane są one na podstawie wybranych przez użytkownika kryteriów, zawartych w polach listy zaprojektowanego formularza (rys. 8) Uzyskane tym sposobem poszukiwane dane przedstawione w postaci tabelarycznej, mogą być przeglądane, modyfikowane, a w dalszej kolejności prezentowane w formie dynamicznie tworzonych raportów.



Rys. 6. Strona główna drugiego modułu „Sekretariat”
Fig. 6. Main page of the “Secretary office” module



Rys. 7. Formularz „Wydatki fakturowane”
Fig. 7. “Invoiced Expenses” form



Rys. 8. Formularz „Wyszukiwanie rekordów”
Fig. 8. “Record search” form

Na poziomie tego podsystemu zarządzamy informacjami o pracownikach, z które są dostępne w obu modułach aplikacji.

Podsumowanie

Realizacja poszczególnych sekwencji działań składających się na zaprojektowanie, wytworzenie i przetestowanie intranetowego systemu informatycznego usprawniającego zarządzanie badaniami w obszarze zasobów finansowych, pozwoliła autorom na sformułowanie następujących uwag i wniosków:

- Środowisko programistycznego Visual Studio .NET umożliwia wygodne, wizualne i szybkie tworzenie internetowych oraz intranetowych systemów informatycznych opartych na technologii ASP .NET.
- Zastosowanie jako SZBD SQL Servera 2000 pozwoliło na wykorzystanie z modelu programistycznego ADO.NET obiektów dedykowanych temu serwerowi, co przyspiesza funkcjonowanie aplikacji.

- Segmentowa budowa systemu ułatwia jego pielęgnację i czyni go bardziej podatnym na modyfikacje.
- Wytworzona intranetowa aplikacja, usprawnia przepływ informacji finansowych i wspomaga proces zarządzania tymi środkami, a jej funkcjonalności pozwalają na wykorzystanie systemu w innych ośrodkach i placówkach naukowych.

Bibliografia

Booch G., Rumbaugh J., Jacobson I. 2000. UML przewodnik użytkownika. WNT, Warszawa.

Griffin J. 2002. XML i SQL Server 2000. Helion, Gliwice.

Payne Ch. 2002. ASP.NET dla każdego. Helion, Gliwice.

Waymire R., Sawtell R. 2002. MS SQL Server 2000 dla każdego, Helion, Gliwice.

Worley S. 2003. ASP.NET Vademecum profesjonalisty. Helion, Gliwice.

AN INTRANET DATABASE SYSTEM ENHANCING RESEARCH MANAGEMENT WITHIN AGRICULTURAL ENGINEERING

It is practically impossible for research centers or other administration units to operate without supporting them information science systems. Specific operation of agricultural engineering research centers combined with high mobility of their workers requires special applications to be implemented. Lack of tools that would fulfill the criteria above made the authors of this paper design & create an intranet information system based on the latest NET technology, a computer aid for a complex research management.

Key words: research management, information science system. ASP, NET