

Paweł DRZYMAŁA, Krzysztof SMÓŁKA, Henryk WELFLE, Sławomir WIAK
Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych

SYSTEM KART PRZEDMIOTÓW W RAMACH PROGRAMU ECTS NA POLITECHNICE ŁÓDZKIEJ

Streszczenie. Artykuł przedstawia założenia i wprowadzenie do systemu ECTS na Politechnice Łódzkiej. Przedstawiono projekt i realizację systemu Kart Przedmiotów w ramach ECTS. Praca opisuje poszczególne funkcjonalności systemu oraz ich rozwiązania. W ramach systemu opracowano abstrakcyjną warstwę bazodanową, która pozwala na współpracę systemu z różnymi bazami danych oraz buforowanie danych przeznaczonych do wypełniania formularzy (jako rozwiązania autorskie).

Słowa kluczowe: Systemy Zarządzania Bazami Danych, Oracle, PostgreSQL, PHP, ECTS.

SYSTEM CARD COURSES UNDER ECTS PROGRAMME AT TECHNICAL UNIVERSITY OF LODZ

Summary. The paper presents the assumptions and the introduction of the ECTS system at the Technical University of Lodz. Presents the design and implementation of the System Card Courses under ECTS. Work describes the different functionality of the system and their solutions. In the system developed an abstract database layer that allows the system to work with various databases and data caching for filling out forms (as a solution to the authors).

Keywords: Database Management System, Oracle, PostgreSQL, PHP, European Credit Transfer System (ECTS).

1. System ECTS na Politechnice Łódzkiej [1]

Politechnika Łódzka (PŁ) jest jedyną uczelnią techniczną w regionie i jedną z największych w kraju. W celu pełnej przejrzystości programów studiów oraz umożliwienia studen-

tom odbywania studiów w innych uczelniach w kraju i za granicą, Politechnika Łódzka wprowadziła system ECTS. Używany jest on jako system transferu i akumulacji punktów.

Punkty kredytowe przyznawane są wszystkim studentom, zarówno wyjeżdżającym, jak i przyjeżdżającym (studenci zagraniczni) oraz studentom odbywającym całość studiów na PŁ. Rok akademicki to typowo 60 punktów kredytowych. Podzielony jest on na dwa semestry, każdy typowo po 30 punktów ECTS. W szczególnych przypadkach mogą pojawić się drobne różnice. Punkty przydzielane są poszczególnym przedmiotom na podstawie nakładu pracy studenta, uwzględniają również efekty kształcenia. Nakład pracy studenta jest systematycznie monitorowany. System ECTS ułatwia uznanie okresu studiów za granicą dla studentów PŁ oraz studentów obcokrajowców przyjeżdżających na PŁ na krótkie okresy studiów.

2. Założenia systemu

W projekcie założono korzystanie ze sprawdzonej i stabilnej technologii „Open Source”, wyjątkiem komercyjnego systemu bazodanowego Oracle. Opracowana aplikacja WEB o nazwie „System Karta” może korzystać z SZBD w oparciu o silnik PostgreSQL, Oracle czy MySQL. Jako serwer aplikacji wybrany został Apache w wersji 2 z interpreterem PHP ver.5.x. Szata graficzna wykorzystuje kaskadowe arkusze stylów CSS.

Do programowania wykorzystano środowisko deweloperskie Eclipse PDT i DTP. „System Karta” opracowywany był przez wiele osób, dlatego wymagane było zastosowanie systemu kontroli wersji kodu.

Ze względów bezpieczeństwa założono, że system wykorzystywać będzie bezpieczne połączenie z użytkownikami, szyfrowane za pomocą Security Socket Layers (SSL). Dostęp możliwy jest po autoryzacji przez unikatowo generowany odcisk i wysyłany na konto w serwisie pocztowym w domenie Politechniki Łódzkiej.

Założenia dla „Systemu Karta”:

- zaawansowane przeszukiwanie bazy danych wzorców zatwierdzonych kart przedmiotów,
- przeszukiwanie bazy danych kart zgłaszanych przedmiotów dla różnych kryteriów wyboru,
- pełne i szybkie porównanie zgłaszanych wersji przedmiotów (różne osoby mogą zgłaszać ten sam przedmiot),
- wprowadzanie nowych kart przedmiotów w oparciu o istniejące wzorce lub zgłaszane i istniejące w bazie nie zatwierdzone przedmioty,
- edycja nowych (lecz nie zatwierdzonych) kart przedmiotów tylko przez kierowników,
- dynamiczne tworzenie dokumentów kart przedmiotów i wydruk dokumentacji papierowej celem przedłożenia do akceptacji dziekanatu,

- uwierzytelnianie użytkowników systemu na podstawie adresu e-mail i hasła, w tym moduł tworzenia nowego hasła wysyłanego za pomocą poczty elektronicznej.
 - W ramach „Systemu Karta” należało uwzględnić ograniczenia:
- wdrożenie na podstawie już istniejących baz danych, przeznaczonych do zarządzania przedmiotami i pracownikami (bez możliwości ingerencji w istniejącą strukturę i dane). Dozwolona była rozbudowa bazy o nowe tabele,
- narzucono ograniczenie współpracy systemu z różnymi SZBD (ORACLE, PostgreSQL, MySQL), tzn. opracowanie możliwie „uniwersalnej” abstrakcyjnej warstwy bazodanowej,
- duża wydajność aplikacji charakteryzującej się silnymi zabezpieczeniami, efektywnością przeszukiwania danych, ich porównywania (system kontroli wersji kart przedmiotów) i możliwość tworzenia wersji potomnych jako nowych propozycji. W tym celu wykorzystane zostały techniki buforowania, kompresji danych zarówno po stronie serwera, jak i klienta.
 - Podsumowując, aplikacja musiała uwzględniać wiele kompromisów podanych założeń i ograniczeń.

3. Konfiguracja systemu

Głównym plikiem konfiguracyjnym jest config.ini.php, który składa się z sekcji zawierające parametry konfiguracyjne opisane podpunktach 3.1. do 3.4.

3.1. Parametry konfiguracyjne: System

Parametry konfiguracyjne „System” określają dane systemu:

```
[system] ;Dane systemu: PostgreSQL, MySQL, Oracle
typ=PostgreSQL
http=http://localhost/karta/
email=admin@localhost.pl
awaria=nie
ograniczenia_ip=tak
```

- **typ=PostgreSQL** - określa aktywny współpracujący SZBD.
- **http=http://localhost/karta/** - określa przekierowanie strony głównej.
- **email=admin@localhost.pl** - określa konto poczty administratora.
- **awaria=nie** - {wartości [tak] [nie]} określa informację o dostępności systemu.
- **ograniczenia_ip=tak** - {wartości [tak] [nie]} parametr, który determinuje domenę zaufaną adresów IP, z których możliwe jest zalogowanie do systemu.

3.2. Parametry konfiguracyjne: Panel

Parametry konfiguracyjne „Panel” określają widok projektu i zgłaszane komunikaty:

```
[panel] ;Widok projektu, komunikatu: tak/nie
projekt=nie
komunikat=tak
```

- **projekt=nie** - {wartości [tak][nie]} parametr określający nieusuwalny dodatkowy panel pokazujący informacje o Systemie.
- **komunikat=tak** - {wartości [tak][nie]} parametr określający dodatkowy, zwijalny panel pokazujący aktualne komunikaty.

3.3. Parametry konfiguracyjne: Powiadomienia

Parametry konfiguracyjne „Powiadomienia” określają „logi” dla administratora:

```
[powiadomienia]
logowanie=tak
zmianahasla=tak
wyslaniehasla=tak
```

- **logowanie=tak** - {wartości [tak][nie]} parametr określający, czy w tabeli powiadomienia pojawi się informacja o tym, kto i kiedy się zalogował.
- **zmianahasla=tak** - {wartości [tak][nie]} parametr określający, czy w tabeli powiadomienia pojawi się informacja o tym kto i kiedy zmienił hasło.
- **wyslaniehasla=tak** - {wartości [tak][nie]} parametr określający, czy w tabeli powiadomienia pojawi się informacja o wysłaniu nowego hasła do użytkownika.

3.4. Parametry konfiguracyjne: Ustawienia bazy danych

Parametry konfiguracyjne „Ustawienia bazy danych” określają informacje o serwerze i połączeniu z SZBD:

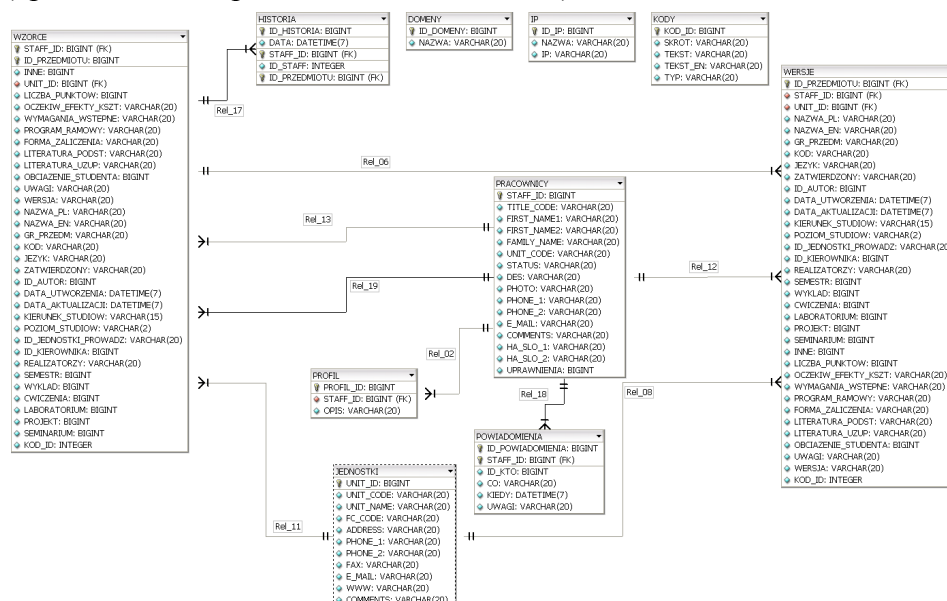
```
[PostgreSQL] ;Ustawienia dla PostgreSQL
serwer=localhost
port=1432
baza=karta
uzytkownik=user
haslo=user_passwd
```

- **serwer=localhost** – określa, gdzie znajduje się baza.
- **baza=karta** - określa nazwę bazy danych.
- **uzytkownik=user** - określa identyfikator użytkownika, który ma uprawnienia do bazy.
- **haslo=user_passwd** - określa hasło do bazy dla identyfikatora użytkownika jak wyżej.

Podsumowując, w sekcji konfiguracyjnej określamy informację dla abstrakcyjnej warstwy bazodanowej. Plik konfiguracji powinien być właściwie zabezpieczony przez szyfrowanie i odpowiednio dostępny z określonych domen.

4. Baza danych

Baza danych powinna odpowiadać parametrom ustawionym w parametrach konfiguracyjnych (np. ustawiona odpowiednia strona kodowa).



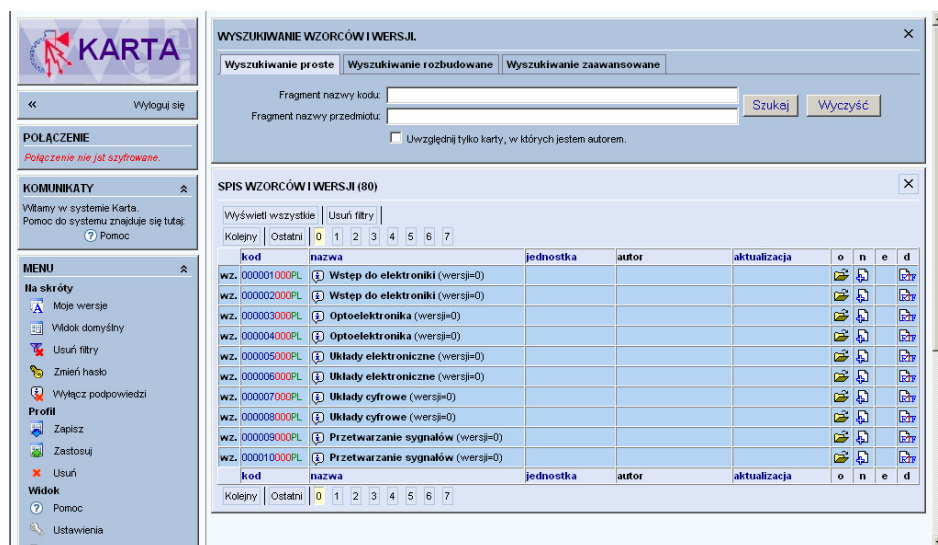
Rys.1. Struktura bazy danych
Fig.1. Database structure

System wykorzystuje następujące tabele:

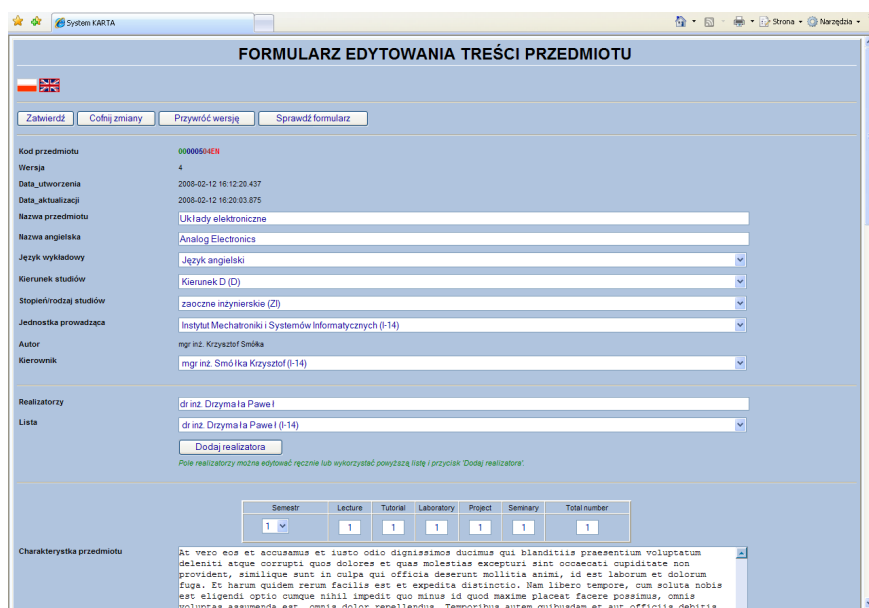
- domeny – zawiera listę dostępnych domen wykorzystywanych podczas logowania,
- ip – zawiera listę możliwych adresów IP wykorzystywanych do pracy z „Systemem Kartą”,
- jednostki – zawiera listę i opis jednostek uczelnianych,
- kody – zawiera listę kodów jednostek uczelnianych,
- powiadomienia – zawiera „logi” dotyczące konfiguracji systemu,
- pracownicy – zawiera listę i dane personalne pracowników,
- profil – zawiera ustawienia własne profilu użytkownika,
- wersje – zawiera wersje kart przedmiotów,
- wzorce – zawiera wzorce kart przedmiotów,
- historia – zawiera zapis historii edycji wersji przedmiotów.

Struktura bazy danych przedstawiona została na rys.1.

5. Układ graficzny



Rys.2. Układ graficzny aplikacji
Fig.2. Application layout



Rys.3. Przykładowy formularz edycji
Fig.3. Sample form editing

Istnieją trzy podstawowe układy graficzne systemu. Na rys. 2 przedstawiono układ graficzny aplikacji po zalogowaniu. Rys. 3 prezentuje przykładowy formularz edycji. Przykładowy formularz wyszukiwania rozbudowanego zilustrowano na rys. 4. System posiada układ graficzny, który można skonfigurować w szerokim zakresie podczas pracy. Możliwe jest związanie poszczególnych pól w części menu oraz dodawanie i usuwanie poszczególnych paneli w głównej części układu graficznego. Możliwe jest również zastosowanie własnego układu graficznego użytkownika systemu.

WYSZUKIWANIE WZORCÓW I WERSJI

Wyszukiwanie proste | Wyszukiwanie rozbudowane | Wyszukiwanie zaawansowane

Uwzględnij wyrażenie: 0

Uwzględnij w: Grupa przedm.

Pomiń wyrażenie: Basic

Pomiń w: Nazwa angielska

Uwzględnij tylko karty, w których jestem autorem.

KARTY: SPIS WZORCÓW I WERSJI (66)

Wyświetl wszystkie | Wyświetl po 20 | Wyświetl po 50 | Pokaż aktualny filtr (Wyszukiwanie rozbudowane) | Usuń filtry

1	2	3	4	5	6	7	Następny	Ostatni
kod	nazwa	jednostka	autor	aktualizacja	o	n	e	d
wz. 02 00 0001 00 PL	Wstęp do elektroniki (wersji=1)							
wer. 02 00 0001 01 GE	Wstęp do elektroniki	I-14	Smółka Krzysztof	2008-04-21				
wz. 02 00 0002 00 PL	Wstęp do elektroniki (wersji=0)							
wz. 02 00 0003 00 PL	Optoelektronika (wersji=0)							
wz. 02 00 0004 00 PL	Optoelektronika (wersji=1)							
wer. 02 00 0004 01 PL	Optoelektronika	I-14	Smółka Krzysztof	2008-04-21				
wz. 02 00 0005 00 PL	Układy elektroniczne (wersji=1)							
wer. 02 00 0005 01 PL	Układy elektroniczne	I-14	Smółka Krzysztof	2008-04-20				
wer. 02 00 0005 01 PL	Układy elektroniczne	I-14	Smółka Krzysztof	2008-04-21				
wer. 02 00 0005 01 PL	Układy elektroniczne	I-14	Smółka Krzysztof	2008-04-21				

Historia (ostatnie 5 zmian):

1	2	3	4	5	6	7	Następny
kod	nazwa	1	2	3	4	5	6
2008-04-21 06:18:29	mgr inż. Krzysztof Smółka	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier
2008-04-21 06:20:56	mgr inż. Krzysztof Smółka	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier
2008-04-21 06:21:27	mgr inż. Krzysztof Smółka	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier
2008-04-21 06:20:56	mgr inż. Krzysztof Smółka	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier
2008-04-21 06:20:52	mgr inż. Krzysztof Smółka	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier	Inżynier

Rys.4. Formularz wyszukiwania rozbudowanego
Fig.4. Extended search form

6. Techniki buforowania i dostępu do danych

Rozdział zawiera istotne techniki buforowania i dostępu do danych z wykorzystaniem abstrakcyjnej warstwy bazodanowej dla „Systemu Karta”.

6.1. Buforowanie

W aplikacjach z dostępem WWW wymaga się odpowiedniego podejścia do buforowania informacji prezentowanej klientowi. Buforowanie danych wpływa na wydajność serwisu WEB. W technikach programistycznych zagadnienie to opiera się na wiedzy, w jaki sposób wykorzystywane są usługi sieciowe.

W literaturze [6] opisane są różne podejścia do zagadnienia buforowania:

- **brak buforowania** - po odświeżeniu strony w przeglądarce, wyświetlana karta za każdym razem będzie generowana łącznie z pobraniem informacji ze źródeł zewnętrznych (np. bazy danych).
- **buforowanie danych** - dane zostaną tylko raz pobrane ze środowiska zewnętrznego i będą wykorzystane wielokrotnie w celu budowania strony.
- **buforowanie elementów** - dane zostaną pobrane tylko raz ze środowiska zewnętrznego. Na ich podstawie zostanie zbudowany element strony (np. pole rozwijalne), który będzie wykorzystywany wielokrotnie.

- **buforowanie strony** – cała strona zostanie w pełni wygenerowana po stronie serwera i zbuforowana w celu wielokrotnego jej wysłania do klienta.

W realizacji projektu „System Karta” zastosowano podejście wykorzystujące (w zależności od potrzeb) wszystkie wymienione techniki buforowania.

W „Systemie Karta” pierwsze rozwiązanie (brak buforowania) było wykorzystane, gdy strona zawierała niewiele elementów. Buforowanie jedynie zmniejszyłoby wydajność i przejrzystość procesu tworzenia strony.

Drugie rozwiązanie (**buforowanie danych**) w projekcie zastosowano wszędzie tam, gdzie ilość danych pobieranych z bazy była duża i miała cechy powtarzalności. Buforowanie danych zmniejszyło liczbę odwołań i obciążenie bazy danych. Serwer aplikacji na podstawie pobranych danych musiał generować kod wynikowy dla klienta.

Trzecie podejście (**buforowanie elementów**) w „Systemie Karta” rozwiązało problem generowania przez serwer aplikacji fragmentów strony dla klienta. Serwer tworzył kod wynikowy dla przeglądarki z fragmentów strony.

Czwarte podejście (**buforowanie strony**) w „Systemie Karta” wykorzystywane było wszędzie tam, gdzie możliwe było powtarzanie całych szablonów stron dla wszystkich klientów.

Zastosowanie buforowania wymusza nowatorskie podejście do zagadnienia aktualności danych. Jeżeli możliwe jest umieszczenie w buforze danych, które dla pewnego interwału czasowego (np. co 1 - 2 godziny) można uznać za statyczne (np. nazwiska pracowników, identyfikatory jednostek, adresy e-mail, kierunki studiów itp.), to zmiana nie wpływa na pracę w systemie. Aktualizacja danych możliwa jest na dwa sposoby:

- przez aktualizację danych w buforach (problem synchronizacji danych źródłowych z umieszczonymi w buforze),
- przez aktualizację danych w bazie na podstawie interwału czasowego (interwał czasu ustawia się w systemie).

Podsumowując, należy stwierdzić, że zastosowane techniki buforowania i selekcji danych w „Systemie Karta” dla określonych fragmentów serwisu pozwoliły na znaczne zwiększenie wydajności przy zmniejszonej ilości odwołań do SZBD.

6.2. Abstrakcyjna warstwa bazodanowa

Od współczesnych aplikacji wymaga się nie tylko efektywności działania, lecz także elastycznego podejścia do jej współpracy z systemem bazodanowym. Dobrze zaprojektowana aplikacja powinna zostać odseparowana od wewnętrznej notacji językowej środowiska SZBD (technik łączenia z systemem, metod wyzwalania procedur i przekazywania parametrów itp.). Opracowując „System Karta” zastosowano abstrakcyjną warstwę bazodanową do realizacji

wyżej postawionego celu. Jednocześnie istotnym elementem jest rozdzielenie warstwy prezentacyjnej, logiki biznesowej i bazy danych.

Abstrakcyjna warstwa bazodanowa wykorzystuje wzorzec projektowy o nazwie „Active Record”, który zakłada, że dany obiekt odpowiada jednemu wierszowi w bazie [7]. Alternatywą dla powyższego wzorca jest pakiet PEAR::DB_DataObject [8].

Podana technika programistyczna, separująca system bazy danych, umożliwiła korzystanie z systemów takich, jak: MySQL, PostgreSQL i Oracle.

W systemie warstwa pośrednicząca zawierała funkcje związane z „tłumaczeniem” danego zapytania dla uniwersalnego wzorca SZBD.

Zdefiniowane w „Systemie Karta” funkcje tłumaczące to:

- tran_sql_connect()
- tran_sql_select_db(\$value)
- tran_sql_query(\$query)
- tran_sql_fetch_assoc(\$date)
- tran_sql_fetch_object(\$date,\$row)
- tran_sql_num_rows(\$date)
- tran_sql_insert_id(\$value)
- tran_sql_escape_string(\$value)
- tran_sql_free_result(\$rez)
- tran_sql_close(\$link)
- tran_sql_close_force()

Przykładowe funkcje tłumaczące (plik trans_PostgreSQL.php) odpowiadające za komunikację z bazą danych PostgreSQL mają następującą definicję:

```
<?php
// Translate - PostgreSQL
function tran_sql_connect()
{
    if($_SESSION['connect']) return $_SESSION['connect'];
    $con=true;
    $serwer="host=".get_ini_config('PostgreSQL','serwer');
    $port=get_ini_config('PostgreSQL','port');
    if($port!='') $port=" port=$port";
    $baza="dbname=".get_ini_config('PostgreSQL','baza');
    $uzytkownik="user=".get_ini_config('PostgreSQL','uzytkownik');
    $haslo="password=".get_ini_config('PostgreSQL','haslo');
    @ $link=pg_connect("$serwer $port $baza $uzytkownik $haslo") or $con=false;
    if($con)
    {
        if(isset($_SESSION['connect'])) $_SESSION['connect']=$link;
        return $link;
    }
    else
        return false;
}
function tran_sql_select_db($value)
{
    return pg_select_db($value);
}
```

```
function tran_sql_query($query)
{
if(ereg('LIMIT[[:space:]]+[[:digit:]]+[[:space:]]*', [[:space:]]*[[:digit:]]+[[:space:]]*',$query))
{
    $str_exp=explode('LIMIT',$query);
    $str_exp[1]=ereg_replace(' ',' OFFSET ', $str_exp[1]);
    $query=implode('LIMIT',$str_exp);
}
return pg_query($query);
}
function tran_sql_fetch_assoc($date)
{
return pg_fetch_assoc($date);
}
function tran_sql_fetch_object($date,$row)
{
return pg_fetch_object($date,$row);
}
function tran_sql_num_rows($date)
{
return pg_num_rows($date);
}
function tran_sql_insert_id($value)
{
return 0;
}
function tran_sql_escape_string($value)
{
return pg_escape_string($value);
}
function tran_sql_free_result($rez)
{
return pg_free_result($rez);
}
function tran_sql_close($link)
{
if(!$_SESSION['connect']) return pg_close($link);
}
function tran_sql_close_force()
{
if(isset($_SESSION['connect']))
{
    $link=$_SESSION['connect'];
    $_SESSION['connect']=false;
    return pg_close($link);
}
}
?>
```

Podsumowując, abstrakcyjna warstwa bazodanowa w „Systemie Karta” daje możliwość stosunkowo elastycznego dopasowywania się do często zmienianych dostawców baz danych.

7. Podsumowanie

Praca prezentuje techniki i technologie programowania „Systemu Karta” w ramach ECTS na Politechnice Łódzkiej. Artykuł opisuje poszczególne funkcjonalności systemu oraz ich rozwiązania: zaawansowane przeszukiwanie bazy danych wzorców zatwierdzonych kart przedmiotów, przeszukiwanie bazy danych kart zgłaszanych przedmiotów według różnych kryteriów wyboru, kompleksowe i szybkie porównanie zgłaszanych wersji przedmiotów,

wprowadzanie nowych kart przedmiotów w oparciu o istniejące wzorce, edycję nowych kart, dynamiczne tworzenie dokumentów w postaci formatów DOC i PDF. Zaprezentowano projekt, jego realizację w oparciu o technologię „Open Source”. Opisano techniki buforowania w celu zapewnienia dużej wydajności. Istotnym elementem systemu jest wprowadzenie abstrakcyjnej warstwy bazodanowej do komunikacji aplikacji z SZBD.

BIBLIOGRAFIA

1. ECTS - Programy Studiów: <http://ectslabel.p.lodz.pl/>
2. Dokumentacja PHP: <http://php.net/manual>
3. Dokumentacja PostgreSQL: <http://www.postgresql.org/docs/>
4. Dokumentacja Oracle: <http://www.oracle.com/technology/documentation/>
5. Peterson J.: Server-Side Caching Options, ASP101.com: http://www.asp101.com/articles/john/server_side_caching/index.asp, dostępne 10.01.2010
6. Sacha J.: Metoda aktywnego rekordu, PHP Solutions Nr 4/2006.
7. PEAR, dokumentacja systemu dystrybucji rozszerzeń do języka PHP: http://pear.php.net/package/DB_DataObject/
- 1.

Recenzenci: Prof. dr hab. inż. Bolesław Pochopień
Dr inż. Paweł Kasproski

Wpłynęło do Redakcji 20 stycznia 2010 r.

Abstract

The paper presents programming techniques and technologies "System Card" in the ECTS at the Technical University of Lodz. Paper describes the various functionalities of the system and their solutions: advanced search a database of approved papers patterns of items, search the database of items reported papers according to different selection criteria, comprehensive and quick comparison of reported versions of items, introducing new items based on existing models and creation of documents in the DOC and PDF formats. The paper presents the project, its implementation based on the "Open Source" technology. It described the tech-

niques of caching to ensure high performance. An important element of the system is to introduce an abstraction layer for database applications to communicate with the RDBMS.

Adresy

Paweł DRZYMAŁA: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22, 90-924 Łódź, Polska, pdrzymal@p.lodz.pl.

Krzysztof SMÓŁKA: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22, 90-924 Łódź, Polska, ksmolka@p.lodz.pl.

Henryk WELFLE: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22, 90-924 Łódź, Polska, welfle@p.lodz.pl.

Sławomir WIAK: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22, 90-924 Łódź, Polska, wiakslaw@p.lodz.pl.