Intranetowy system wspomagania prac administracyjno - biurowych w Policealnym Studium Informatyki

Pawel Prokop

9 czerwca 2000

1 Wprowadzenie

Prezentowany w pracy program powstał na określone zapotrzebowanie. Obecny system obsługi administracyjno-biurowej Policealnego Studium Szkoły Informatyki AE w Krakowie nadal w swojej filozofii oparty jest na tradycyjnej technologii procesów infotmacyjnych bazującej an dokumentach papierowych. W tej organizacji komputer stanowi element wspomagający wykonywanie określonych zadań (np. napisanie i wydrukowanie określonego zaświadczenia). Próba zmiany technologii procesów informacyjnych związanych z obsługą Studium wymaga zupełnie innej filozofii organizacji pracy, a w konsekwencji i innego od klasycznych aplikacji wspomagających prace administracyjno-biurowe. Zaprojektowane rozwiązanie i zrealizowane oprogramowanie jest próbą samodzielnie zbudowanej alternatywy oprogramowana w filozofii intranetu dla drogich programów komercyjnych wspomagających pracę grupową (np. opartych na Lotus Notes).

1.1 Podstawowy zakres projektu

Obcny zakres automatyzacji procesów informacyjnych związanych z obsługą administracyjno-biurową Policealnego Studium obejmuje zagadnienia prezentowane w następnuch, kolejnych podpunktach.

1.1.1 Zbieranie, gromadzenie oraz wyszukiwanie danuch dotyczących słuchaczy

Dane gromadzone będą w bazie danych. Dzięki temu nieporównywalnie zmaleje czas docierania do informacji dotyczących słuchacza. Wszystkie dane przechowywane będą w jednym miejscu co powinno zmniejszyć prawdopodobieństwo zagubienia danuch lub pomyłkowego ich umiejscowienia np. w niewłaściwej teczce.

1.1.2 Archiwizacja danych

Istnieje wyraźna potrzeba szybkiej i prostej archiwizacji bazy danych. Z biegiem czasu ilość napływających danych wzrośnie do takiego rozmiaru, że nie będzie możliwe dalsze ich składowanie. A czas dostępu do konkretnego elementu wzrośnie tak, że takie wyszukiwanie straci sens. Stąd postulat archiwizacji. Dodatkowo, archiwizacja będzie ważnym ogniwem zabezpieczenia systemu przed utratą danuch (np. wskutek awarii systemu lub ataku z zewnątrz).

1.1.3 Obsługa zdarzeń zachodzących w studium

Przykładowymi zdarzeniami w analizowanym systemie są takie zdarzenia jak: zaliczenie semestru przez słuchacza, dokonania przez niego wpłaty czesnego, wybranie tematu pracy dyplomowej itp. Wszystkie te zdarzenia rejestrowane będą w bazie danych.

1.1.4 Raporty

O użyteczności systemu najlepiej świadczy lista predefiniowanych raportów możliwa do uzyskania za pośrednictwem przysłowiowego „naciśnięcia” guzika.

Projektowane rozwiązanie powinno pozwalać łatwo uzyskiwać podstawowe raporty takie jak listy słuchaczy uczęszczających na zajęcia, listy zaliczeń oraz ich analizy statystyczne (np. średnie arytmetyczne dla ucznia, klasy, Studium) itp. są takie zdarzenia jak: zaliczenie semestru przez słuchacza, dokonania przez niego wpłaty czesnego, wybranie tematu pracy dyplomowej itp. Wszystkie te zdarzenia rejestrowane będą w bazie danych.

1.2 Rozwiązanie klient-serwer

Istnieje kilka sposobów rozwiązania powyższego problemu. Jednym z nich jest napisanie programu, który będzie korzystał z lokalnej bazy danych (zrealizowanej np, przy wykorzystaniu MS Access). W tym wariancie osoba siedząca przy komputerze, na którym zainstalowany będzie ten program będzie miała możliwość wprowadzania, jak i odczytywania danych. Jest to rozwiązanie dobre, jednakże co zrobić w momencie, gdy dwie osoby naraz muszą skorzystać utworzonej bazy danych ? Ponieważ w rozważanym kontekście taka potrzeba będzie powszechna (np. baza danych powinna być na bieżąco zasilana danymi przez nauczycieli) więc dużo lepszym pomysłem jest wielodostępna baza danych. Aby zapewnić łatwość posługiwania się bazą danych warto rozważyć zbudowanie interfejsu użytkownika w oparciu o przeglądarkę graficzną WWW, czyli skorzystać z filozofii komunikacji znanej powszechnie z internetu. Bazą danych można utworzyć na komputerze (serwerze), który będzie dostępny w sieci uczelnianej. Warunkiem dotarcia do bazy danych w tej sytuacji będzie możliwość korzystania z komputera upoważnionego do komunikacji z naszą bazą danych za pośrednictwem sieci. Jest rzeczą oczywistą, że w tym modelu, wiele użytkowników będzie mogło korzystać z tak posadowionej bazy danych równocześnie.
Kolejną zaletą wprowadzenia sieciowej organizacji jest istniejące już oprogramowanie. Klient korzystał będzie z przeglądarki internatowej podczas wprowadzania i wyprowadzania informacji z bazy. W obecnych czasach większość ludzi niejednokrotnie korzystało z przeglądarki graficznej : Netscape Communicator, lub Internet Explorer. Użycie wiec standardowego oprogramowania pociąga za sobą następujące korzyści :

1. potencjalny pracownik nie będzie potrzebował przechodzić szkolenia posługiwania się programem.
2. producent nie musi tworzyć złożonego systemu oprogramowania sieciowego, gdyż system taki już istnieje. Zaoszczędza w ten sposób wiele cennego czasu.

System, który prezentuję spełnia większość początkowych założeń, jednakże nie wszystkie zostały jednak zrealizowane, z powodu poważnych ograniczeń czasowych.

1.3 Wymagania

System nie ma wygórowanych wymagań sprzętowych, jednakże niezbędne jest zainstalowanie kilku programów po stronie serwera, oraz przeglądarki internetowej po stronie klienta.

1.3.1 Wymagania po stronie klienta

Wymagania po stronie klienta to tylko przeglądarka internetowa (ang. Browser), może to być jeden z programów : Netscape Navigator, Internet Explorer. Są to najpopularniejsze programy do przeglądania danych z internetu. Jest jeszcze jeden ważny wymóg - przeglądarka powinna akceptować rozszerzenia Java Script, więc w miarę ważne jest, żeby nie była to jedna z pierwszych wersji programu. System testowany był używając Netscape Navigatora w wersji 4.6, 4.7. Potrzebne jest również środowisko graficzne, Może to być zarówno Windows (95, 98), jak i UNIXowe graficzne środowisko X. (testowane na X-ach).
Wymagania sprzętowe : Potrzebne jest połączenie z siecią, do której przyłączony jest również komputer będący serwerem bazy. Czyli, co za tym idzie karta sieciowa, lub modem. Poza tym musi być to komputer klasy, która umożliwi zainstalowanie wyżej wymienionego oprogramowania.

1.3.2 Wymagania po stronie serwera

Wymagania programowe są następujące : Na komputerze powinien być zainstalowany system UNIXowy. W systemie powinna być obecna odpowiednio skonfigurowana usługa WWW, oraz system zarządzania bazą danych. O instalacji tego oprogramowania mówi rozdział 4. Co do sprzętu to musi być to komputer minimalnie Pentium 100 MHz, powinien mieć jak najwięcej pamięci RAM, minimalnie powinno to być 48 MB, żeby system działał w miarę szybko. Testowany był właśnie na takim sprzęcie i chodził bez zastrzeżeń. Oczywiście należy wziąć pod uwagę fakt iż był to mój domowy komputer, który w trakcie pracy programu nie był obciążony innymi pamięciożernymi, lub zajmującymi w dużym stopniu procesor procesami. Jeśli by tak było należy do tego celu użyć odpowiednio mocniejszej maszyny.

2 Frequently Asked Questions ? Instrukcja użytkownika w formie „Pytań i odpowiedzi” (FAQ)

2.0.3 Jak to włączyć ?

System Administrujący Danymi Studium Informatycznego działa w połączeniu z przeglądarką internetową. Należy więc będąc w trybie graficznym uruchomić program służący do przeglądania stron internatowych : np. Netscape Navigator, Internet Explorer lub Sylaba Communicator.

2.0.4 Uruchomilem przeglądarkę i co dalej ?

Należy połączyć się z serwerem na którym jest uruchomiony system. Aktualnie jest to SI : należy wiec wpisać w okno edycyjne adres : http://si.ae.krakow.pl/psi
Pojawi się wówczas okno z zapytaniem o użytkownika systemu, oraz zapytanie o hasło, które w odpowiednim miejscu należy podać.

2.0.5 Co jeśli nie znam hasła, lub zapomniałem hasło ?

Po prostu nie zostaniesz wpuszczony do systemu. Takich rzeczy się nie zapomina. Należy się wówczas skontaktować z administratorem bazy.

2.0.6 W jaki sposób wprowadzić mam nowy przedmiot do bazy danych ?

Należy kliknąć w przycisk po lewej stronie - 'Przedmioty' i wypełnić formularz, po czym kliknąć przycisk 'Dodaj'

2.0.7 W jaki sposób wprowadzić mam nowego wykładowcę ?

Należy kliknąć w przycisk 'Wykładowcy' po lewej stronie ekranu, wybrać opcje : 'Przyjęcie Wykładowcy' po czym wypełnić formularz i nacisnąć przycisk 'Dodaj'

2.0.8 W jaki sposób przyporządkować przedmiot wykładowcy ?

W tym celu należy kliknąć przycisk 'Wykładowcy', wybrać odpowiedniego wykładowcę z pola wyboru, potwierdzić wybór kliknięciem w 'Potwierdź'. Kolejno należy wybrać opcje 'Przedmioty' znajdującą się pod nazwiskiem, wybrać odpowiedni przedmiot z pola wyboru i potwierdzić wybór kliknięciem w przycisk 'Potwierdź'

2.0.9 A co gdyby w drugim semestrze zmienił się wykładowca danego przedmiotu?

Przyciskając przycisk 'Przedmioty' po lewej stronie a potem wybierając dany przedmiot w oknie 'Aktualne przedmioty' pojawi się na ekranie strona z ustawieniami dotyczącymi tego przedmiotu. W dolnej części znajduje się podział na semestry i wykładowców w danym semestrze. Zmiana następuję po wybraniu innego wykładowcy z okna wyboru znajdującego się po prawej stronie numeru semestru.

2.0.10 W jaki sposób wprowadzić do bazy danych nowych słuchaczy ?

Należy kliknąć w przycisk 'Słuchacze' po lewej stronie ekranu, i wybrać opcje 'Przyjęcie słuchacza', po czym wypełnić formularz i potwierdzić dane przyciskiem 'Dodaj'

2.0.11 W jaki sposób podawać datę do systemu ?

Format wprowadzania daty jest środkowoeuropejski, a wiec : 'dd-mm-rrrr'

2.0.12 W jaki sposób zaznaczyć w bazie danych fakt wprowadzenia, lub pobrania dokumentów przez studenta ?

Po kliknięciu przycisku 'Słuchacze' po lewej stronie ekranu i wybraniu odpowiedniego słuchacza z okna wyboru należy kliknąć 'Dokumenty' - pod nazwiskiem słuchacza. Pojawi się wówczas strona z dokumentami. W celu wprowadzenia dokumentu przez słuchacza należy wcisnąć przycisk/zaznaczyc pole znajdujące się po prawej stronie od nazwy dokumentu. Po dokonaniu odpowiednich modyfikacji należy kliknąć 'Potwierdź'

2.0.13 W jaki sposób zaznaczyć w bazie fakt wprowadzenia opłaty przez słuchacza ?

Po kliknięciu przycisku 'Słuchacze' po lewej stronie ekranu i wybraniu odpowiedniego słuchacza z okna wyboru należy kliknąć 'Opłaty' - pod nazwiskiem słuchacza Pojawi się wówczas strona z opłatami dotyczącymi słuchacza. W górnej części ekranu znajduje się informacja dotycząca wszystkich opłat wniesionych przez słuchacza. Poniżej formularz który należy odpowiednio wypełnić i potwierdzić kliknięciem w przycisk 'Potwierdź' znajdujący się poniżej.

2.0.14 W jaki sposób wprowadzić zaliczenie semestru przez słuchacza ?

Po kliknięciu przycisku 'Słuchacze' po lewej stronie ekranu i wybraniu odpowiedniego słuchacza z okna wyboru należy kliknąć 'Zaliczenie' - pod nazwiskiem słuchacza Pojawi się wówczas strona dotycząca zaliczeń poszczególnych przedmiotów. W górnej części strony znajduje się informacja na temat wpisanych do tej pory zaliczeń poszczególnych przedmiotów oraz terminów w jakich zostały one zaliczone. Poniżej znajduje się formularz, który należy odpowiednio wypełnić, oraz wpisane informacje potwierdzić kliknięciem w przycisk 'Potwierdź'.

2.0.15 Co jeśli w trakcie trwania toku nauczania nastąpi zmiana długości trwania semestru ?

Po kliknięciu przycisku 'Przedmioty' po lewej stronie ekranu należy wybrać odpowiedni przedmiot, pojawi się wówczas strona zawierająca informacje o danym przedmiocie. Można wówczas modyfikować informacje dotyczące długości trwania przedmiotu. Należy jednak potem upewnić się, czy w dalszym ciągu poprawnie ustawione są informacje dotyczące wykładowcy.

2.0.16 W jaki sposób dokonać wpisu tematu pracy dyplomowej do bazy danych ?

Po kliknięciu przycisku 'Prace Dyplomowe' po lewej stronie ekranu pojawi się strona zawierająca informacje dotyczące wszystkich pracy dyplomowych. Należy wówczas poprawnie wypełnić formularz i potwierdzić kliknięciem w przycisk 'Dodaj'

2.0.17 W jaki sposób zmienić promotora danej pracy dyplomowej ?

Po kliknięciu przycisku 'Prace Dyplomowe' po lewej stronie ekranu pojawi się strona zawierająca informacje dotyczące wszystkich prac dyplomowych. Należy wtedy wybrać interesującą nas prace dyplomowa, pojawi się wówczas strona zawierająca formularze. W sekcji 'Zmiana promotora pracy dyplomowej' należy wprowadzić odpowiednie zmiany. Analogicznie dokonuje się zmian dotyczących słuchaczy przyporządkowanych danej pracy.

2.0.18 W jaki sposób przyporządkować słuchacza pracy dyplomowej ?

Analogicznie do pkt. 15

2.0.19 Co to są Raporty ?

Raporty są globalnymi zestawieniami informacji zapisanych w bazie. Są to tabele zawierające spisy wszystkich słuchaczy, wykładowców, prac dyplomowych... i inne. Do raportów można się dostać klikając przycisk 'Raporty' po lewej stronie ekranu.

3 Raporty

Celem raportów jest zebranie danych z bazy, oraz ich sumaryczne zestawienie. W naszym projekcie potrzebne będzie kilkanaście różnorodnych raportów. Ich wygląd powinien być odpowiednio dopasowany do monitora i drukarki, gdyż również celem raportu jest możliwość zapisania go na nośniku zwanym papierem. Powinien być stworzony na białym gładkim tle ze względu na oszczędność tonera w drukarce. I również powinna być użyta czarna czcionka rozmiaru ok 12pt. Oczywiście przeglądarki maja możliwość wymuszenia rozmiaru i koloru czcionek, oraz koloru tła na stronie aktualnie przeglądanej. Użytkownik będzie wiec mógł ustawić to wedle własnego upodobania.
Raport musi być stworzony w taki sposób, żeby identycznie wyglądał niezależnie od systemu operacyjnego. Chodzi tu o pozbycie się wszelkich obiektów, które różnią się miedzy sobą w UNIXie i Windowsie. Są to : textboxy, comboboxy, radiobuttony, checkboxy oraz zwykle buttony. Takie różnice niekorzystnie wpłyną na ogólny wygląd i przejrzystość raportu.
Za wydrukowanie raportu odpowiedzialne będzie środowisko przeglądarki, jaką użytkownik będzie używał. Najczęściej będzie to albo Netscape Navigator, albo Internet Explorer. Programy te umożliwiają drukowanie.
Raporty będą zrobione oddzielnie, chociaż będą wykorzystywać takie same fragmenty kodu programu. Zachcianki klienta odnośnie drukowania informacji mogą się zmieniać bardzo często
Z punktu widzenia programisty lepiej było by stworzyć funkcje wyświetlająca informacje i wywoływać ja dwa razy - raz podczas wprowadzania danych, drugi raz podczas drukowania raportu. Jeżeli wybredny klient zażyczyłby sobie drobna zmianę w wyglądzie raportu, owa zmiana była by również widoczna na formularzu co mogło by wpłynąć niekorzystnie na jego wygląd i przejrzystość. Dlatego lepiej jest wziąć pod uwagę również zachcianki klienta i nadłożyć sobie pracy w tym momencie jednocześnie oszczędzając czas w przyszłości.

4 PROJEKT BAZY DANYCH

Naszym zadaniem jest stworzenie bazy danych gromadzącej informacje wymagane przez Policealne Studium Informatyki.

4.1 Zidentyfikowanie obszaru problemu

Policealne Studium Informatyki jest szkołą dwu letnią. Przyjmowane są tu osoby, które ukończyły szkołę średnią Podczas zapisów do studium słuchacze wnoszą opłatę i wymagane dokumenty. W każdym miesiącu słuchacze zobowiązani są wnieść opłatę. Pod koniec trzeciego semestru lub na początku czwartego, każdy słuchacz powinien wybrać jeden z zaproponowanych przez wykładowców tematów pracy dyplomowej.
Do studium są przyjęci wykładowcy, każdy z nich wykłada określony przedmiot, w określonym semestrze. Albo jeden przedmiot wykładany jest przez kilku wykładowców w różnych semestrach. Wykładowca powinien pod koniec trzeciego semestru zaproponować kilka tematów prac dyplomowych. Prace dyplomową powinien napisać każdy słuchacz, jednakże dopuszcza się sytuacje, gdy jedna praca dyplomowa jest pisana przez kilku słuchaczy.
Plan zajęć na każdy z semestrów zostaje ogłoszony na początku każdego semestru. Jednakże w trakcie trwania semestru możliwe są niewielkie zmiany.
Jeżeli słuchacz sobie zażyczy studium powinno wydać odpowiednie zaświadczenie o nauce słuchacza w studium.

4.2 Zdefiniowanie obszaru problemu

Konieczne jest manipulowanie informacjami o następujących elementach :

1. słuchacze przyjęci do studium
2. zatrudnieni wykładowcy
3. przedmioty wykładane w studium
4. tematy prac dyplomowych
5. zaliczenia poszczególnych przedmiotów
6. dokumenty
7. wniesione opłaty
8. plan zajęć
9. raporty

4.2.1 Dekompozycja elementów

Obiekty występujące w bazie :

1. uczniowie : o każdym uczniu musi zostać zapisana duża ilość informacji: nazwisko, imię, drugie imię, dokładny adres (ulica, lokal, kod, miasto), numer pesel, numer telefonu, miejsce urodzenia, imiona rodziców, data przyjęcia i data ukończenia studium, płeć, informacja o tym, czy osoba posiada maturę, oraz jej stosunek do służby wojskowej, i rok na który aktualnie uczęszcza osoba.
2. wykładowcy : nazwisko, imię, adres zamieszkania i numery telefonów (służbowy i prywatny) wykładowcy, numer konta bankowego, tytuł

3. przedmioty, czyli nazwa przedmiotu, informacja o tym, czy aktualnie przedmiot jest wykładany, oraz o tym ile trwa i w którym semestrze się rozpoczyna

4. opłaty wniesione przez słuchaczy za dany miesiąc, wysokość wpłaty, oraz data wpłynięcia gotówki i rok kalendarzowy identyfikujący wpłatę

5. dokumenty, jakie wniósł słuchacz wybierane z listy dokumentów które przyjmuje szkoła

6. plan zajęć, dzień tygodnia i jaki przedmiot się o której godzinie rozpoczyna w którym semestrze i zależnie od grupy.

7. prace dyplomowe : temat pracy, informacje na temat promotora i recenzenta ich propozycje ocen, oraz informacje na temat studentów, którzy ta prace piszą.

8. oceny poszczególnych uczniów na końcu każdego semestru z każdego przedmiotu, termin w którym ta ocena została wstawiona, oraz rok kalendarzowy

4.3 Projekt tabel

4.3.1 Tabela dni_tygodnia

- id_dnia - jednoznaczny numer identyfikujący nazwę dnia; typ całkowity, któtki, nie może być wartością pustą

- nazwa_dnia - nazwa dnia („poniedziałek”, „wtorek”, etc); łańcuch długości 13 znaków, nie może być wartością pustą

Tablica 1: dni_tygodnia

nazwa	typ	spójność
id_dnia	SMALLINT	NOT NULL
nazwa_dnia	CHARACTER(13)	NOT NULL

4.3.2 Tabela dokumenty

W tabeli będą zapisane nazwy dokumentów jakie wymaga studium od słuchacza.

- id_dokumentu - jednoznaczny numer określający nazwę dokumentu; typ całkowity, krótki, wartość nie może być pusta.

- nazwa_dokumentu - nazwa dokumentu; łańcuch znaków długości 64 znaki, nie może być wartością pustą.

Tablica 2: dokumenty

nazwa	typ	spójność
id_dokumentu	SMALLINT	NOT NULL
nazwa_dokumentu	CHARACTER (64)	NOT NULL

4.3.3 Tabela email_naucz

W tabeli będzie zapisany adres e-mail nauczyciela, oraz jednoznaczny identyfikator nauczyciela, do którego należy ten adres.

- id_nauczyciela - numer jednoznacznie identyfikujący nauczyciela; typ całkowity krótki, nie może być wartością
- email - adres e-mail nauczyciela; łańcuch długości 64 znaków nie może być wartością pustą.

Tablica 3: email_nauczyciela

nazwa	typ	spójność
id_nauczyciela	SMALLINT	NOT NULL
email	CHARACTER (64)	NOT NULL

4.3.4 Tabela email_ucznia

W tabeli będzie zapisany adres e-mail słuchacza, oraz jednoznaczny identyfikator słuchacza, do którego należy ten adres.

- id_ucznia - numer jednoznacznie identyfikujący słuchacza; typ całkowity krótki, nie może być wartością
- e-mail - adres e-mail słuchacza; łańcuch długości 64 znaków nie może być wartością pustą.

Tablica 4: email_ucznia

nazwa	typ	spójność
id_ucznia	SMALLINT	NOT NULL
email	CHARACTER (64)	NOT NULL

4.3.5 Tabela godziny

W tabeli będą zapisane informacje na temat trwania zajęć, rozpiętości.

- id_lekcji - numer jednoznacznie identyfikujący każdą lekcję; typ całkowity krótki, nie może być wartością pustą
- godziny - łańcuch długości 15 bajtów zawierający godziny rozpoczęcia i zakończenia zajęć.

Tablica 5: godziny

nazwa	typ	spójność
id_godziny SMALLINT NOT NULL		
godziny	CHARACTER (15)	NOT NULL

4.3.6 Tabela miesiące

W tabeli będą zapisane nazwy miesięcy

- id_miesiąca - numer jednoznacznie identyfikujący miesiąc; liczba całkowita krótka, nie może być wartością pustą.
- nazwa_miesiaca - łańcuch długości 15 bajtów zawierający nazwę miesiąca („styczeń”, „luty” etc...)

Tablica 6: miesiące

nazwa	typ	spójność
id_miesiaca	SMALLINT	NOT NULL
nazwa_miesiaca	CHARACTER (15)	NOT NULL

4.3.7 Tabela naucz_praca

Tabela łącząca prace dyplomowe z nauczycielem (promotorem)

- id_pracy - numer jednoznacznie identyfikujący pracę dyplomową; liczba całkowita krótka, nie może być wartością pustą.
- id_nauczyciela - jednoznaczny numer wykładowcy będącego promotorem; liczba całkowita krótka, nie może być wartością pustą.

Tablica 7: naucz_praca

nazwa	typ	spójność
id_pracy	SMALLINT	NOT NULL
id_promotora	SMALLINT	NOT NULL

4.3.8 Tabela naucz_przed

Tabela łącząca przedmioty z wykładowcami.

- id_nauczyciela - jednoznaczny identyfikator nauczyciela; liczba całkowita krótka, nie może być wartością pustą.
- id_przedmiotu - jednoznaczny numer identyfikujący przedmiot; liczba całkowita krótka, nie może być wartością pustą.
- semestr - numer semestru w którym dany nauczyciel wykłada podany przedmiot; liczba całkowita krótka;

Tablica 8: naucz_przed

nazwa	typ	spójność
id_nauczyciela	SMALLINT	NOT NULL
id_przedmiotu	SMALLINT	NOT NULL
semestr	SMALLINT	

4.3.9 Tabela opłaty

W tabeli będą zapisane informacje na temat wymaganych opłat, przez studium, oraz dat ich wniesienia przez słuchacza.

- id_ucznia - jednoznaczny identyfikator ucznia; liczba całkowita krótka, nie może być wartością pustą.
- id_miesiąca - jednoznaczny identyfikator miesiąca; liczba całkowita, krótka, nie może być wartością pustą.
- rok_kal - rok kalendarzowy za którego miesiąc jest wpłata; liczba całkowita krótka, nie może być wartością pustą.
- data_wpłynięcia - data wpłynięcia opłaty; wartość typu datowego.
- kwota - kwota jaka wpłynęła; wartość typu walutowego.

Tablica 9: opłaty

nazwa	typ	spójność
id_ucznia	SMALLINT	NOT NULL
id_miesiaca	SMALLINT	NOT NULL
rok_kal	SMALLINT	NOT NULL
data_wplyniecia	DATE	
kwota	MONEY	

4.3.10 Tabela plan_zajęć

Tabela będzie zawierała plan zajęć

- id_lekcji - jednoznaczny identyfikator lekcji; liczba całkowita krótka, nie może być wartością pustą.
- id_dnia - jednoznaczny identyfikator dnia; liczba całkowita, krótka, nie może być wartością pustą.
- id_przedmiotu - jednoznaczny identyfikator przedmiotu; liczba całkowita krótka, nie może być wartością pustą.
- semestr - numer semestru; liczba całkowita krótka, nie może być wartością pustą.
- rok_nauki - rok nauki (pierwszy, albo drugi); liczba całkowita krótka.
- grupa - numer grupy studenckiej; liczba całkowita krótka.

- id_nauczyciela - numer jednoznacznie identyfikujący nauczyciela; liczba całkowita krótka.

Tablica 10: plan_zajęć

nazwa	typ	spójność
id_lekcji	SMALLINT	NOT NULL
id_dnia	SMALLINT	NOT NULL
id_przedmiotu	SMALLINT	NOT NULL
rok_nauki	SMALLINT	
grupa	SMALLINT	
id_nauczyciela	SMALLINT	

4.3.11 Tabela praca_dyplomowa

Tabela zawierająca informację na temat pracy dyplomowej.

- id_pracy - numer jednoznacznie identyfikujący pracę; liczba całkowita krótka; nie może być wartością pustą
- id_promotora - numer jednoznacznie identyfikujący promotora; liczba całkowita, krótka, nie może być wartością pustą.
- temat - temat pracy dyplomowej; typ łańcuchowy o długości 255 bajtów.
- data_obrony - data obrony pracy dyplomowej; pole typu datowego.

Tablica 11: praca_dyplomowa

nazwa	typ	spójność
id_pracy	SMALLINT	NOT NULL
id_promotora	SMALLINT	NOT NULL
temat	CHARACTER (255)	
data_obrony	DATE	

4.3.12 Tabela przedmioty

Tabela będzie zawierać informacje na temat przedmiotów wykładanych w studium.

- id_przedmiotu - jednoznaczny identyfikator przedmiotu; liczba całkowita, krótka, nie może zawierać wartości pustej.
- nazwa_przedmiotu - łańcuch długości 64 bajtów zawierający nazwę przedmiotu, nie może być wartością pustą.
- prowadzony - wartość logiczna - czy dany przedmiot jest aktualnie prowadzony w studium; nie może być wartością pusta.
- ilosc_sem - ilość semestrów trwania przedmiotu; liczba całkowita krótka.
- pocz_sem - semestr początkowy przedmiotu; liczba całkowita krótka.

Tablica 12: przedmioty

nazwa	typ	spójność
id_przedmiotu	SMALLINT	NOT NULL
nazwa_przedmiotu	CHARACTER (64)	NOT NULL
prowadzony	BOOLEAN	NOT NULL
ilosc_sem	SMALLINT	
pocz_sem	SMALLINT	

4.3.13 Tabela ucz_praca

- id_ucznia - jednoznaczny numer identyfikujący słuchacza; liczba całkowita krótka, nie może być wartością pustą.
- id_pracy - numer jednoznacznie identyfikujący pracę dyplomową; liczba całkowita krótka, nie może być wartością pustą.‘
- ocena_promotora - identyfikator oceny, wystawionej przez promotora; liczba całkowita, krótka;
- ocena_recenzenta - identyfikator oceny, wystawionej przez recenzenta; liczba całkowita krótka.
- obrona - identyfikator oceny, wystawionej podczas obrony pracy; liczba całkowita krótka.
- pytania - identyfikator oceny odpowiedzi na pytania, wystawionej podczas obrony pracy; liczba całkowita, krótka.
- dyplomowa - identyfikator oceny; jest to ocena, którą słuchacz ma na dyplomie.

Tablica 13: ucz_praca

nazwa	typ	spójność
id_ucznia	SMALLINT	NOT NULL
id_pracy	SMALLINT	NOT NULL
ocena_promotora	SMALLINT	
ocena_recenzenta	SMALLINT	
obrona	SMALLINT	
pytania	SMALLINT	
dyplomowa	SMALLINT	

4.3.14 Tabela wojsko

Tabela zawierająca oficjalne kategorie wojskowe.

- id_wojsko - numer jednoznacznie identyfikujący kategorię wojskową; liczba całkowita krótka, nie może być wartością pustą.
- nazwa_kat - nazwa kategorii; łańcuch długości 30 bajtów.

Tablica 14: wojsko

nazwa	typ	spójność
id_wojsko	SMALLINT	NOT NULL
nazwa_kat	CHARACTER (30)	NOT NULL

4.3.15 Tabela zaswiad_uczen

Tabela wiążąca wydanie zaświadczenia z uczniem.

- id_zaswiadczenia - jednoznaczny identyfikator zaświadczenia; liczba całkowita krótka, nie może zawierać wartości pustej.
- id_ucznia - jednoznaczny identyfikator słuchacza; liczba całkowita krótka, nie może być wartością pustą.
- data_poprania - data, kiedy studium wydało zaświadczenie słuchaczowi; pole typu datowego.

Tablica 15: zaswiad_uczen

nazwa	typ	spójność
id_zaswiadczenia	SMALLINT	NOT NULL
id_ucznia	SMALLINT	NOT NULL
data_pobrania	DATEr	

4.3.16 Tabela zaswiadczenia

Tabela będzie zawierać dane na temat zaświadczeń, które może wydać studium.

- id_zaswiadczenia - jednoznaczny numer identyfikujący zaświadczenie; liczba typu całkowitego, krótkiego, nie może być wartością pustą.
- nazwa_zas - nazwa zaświadczenia („do wojska”, „do pracy”, etc. . .); pole typu łańcuchowego o dłuigości 30 bajtów.

Tablica 16: zaswiadczenia

nazwa	typ	spójność
id_zaswiadczenia	SMALLINT	NOT NULL
nazwa_zas	CHARACTER (30)	

4.3.17 Tabela tytuly

Tabela zawierać będzie spis tytułów naukowych możliwych na uczelni.

- id_tytulu - jednoznaczny identyfikator tytułu; liczba całkowita krótka, nie może zawierać wartości pustej.
- tytul - tytuł naukowy; pole typu łańcuchowego o długości 20 bajtów.

Tablica 17: tytuly

nazwa	typ	spójność
id_tytulu	SMALLINT	NOT NULL
tytul	CHARACTER (20)	

4.3.18 Tabela wojsko_uczen

Tabela wiążąca słuchacza z kategorią wojskową.

- id_ucznia - jednoznaczny numer identyfikujący słuchacza; liczba całkowita krótka.
- id_wojsko - numer jednoznacznie identyfikujący kategorię wojskową; liczba całkowita, krótka.

Tablica 18: wojsko_uczen

nazwa	typ	spójność
id_ucznia	SMALLINT	NOT NULL
id_wojsko	SMALLINT	NOT NULL

4.3.19 Tabela uczniowie

Tabela będzie zawierać dane na temat słuchaczy przyjętych do studium.

- id_ucznia - numer jednoznacznie identyfikujący słuchacza; liczba całkowita krótka, nie może być wartością pustą.
- nazwisko - nazwisko słuchacza; pole typu łańcuchowego o długości 30 bajtów.
- imie1 - pierwsze imię słuchacza; pole typu łańcuchowego o długości 20 bajtów, nie może być wartością pustą.
- imie2 - drugie imię słuchacza, pole typu łańcuchowego o długości 20 bajtów.
- miejscowosc - miejscowość w jakiej zameldowany jest słuchacz; pole łańcuchowe o długości 30 bajtów.
- ulica - ulica na jakiej zameldowany jest słuchacz; pole łańcuchowe o długości 30 bajtów.
- lokal - dom / mieszkanie; pole typu łańcuchowego o długości 15 bajtów.
- kod - kod pocztowy słuchacza; pole typu łańcuchowego o długości 6 bajtów.
- pesel - numer PESEL słuchacza; pole typu łańcuchowego o długości 11 bajtów.
- data_urodzenia - data urodzenia słuchacza; pole typu datowego.

- miejsce_ur - miejsce urodzenia słuchacza (miasto); pole typu łańcuchowego o długości 30 bajtów.
- telefon - telefon domowy słuchacza; pole typu łańcuchowego o długości 15 bajtów.
- data_przyjęcia - data przyjęcia słuchacza do studium (złożenia przez niego odpowiednich dokumentów, oraz opłaty wpisowej); pole typu datowego.
- data_ukonczenia - data ukończenia studium przez słuchacza; pole typu datowego.
- imie_ojca - imię ojca słuchacza; pole łańcuchowe o długości 20 bajtów.
- imie_matki - imię matki słuchacza; pole łańcuchowe o długości 20 bajtów.
- ukonczona_szkoła - szkoła jaką ukończył przyjmowany słuchacz; pole typu łańcuchowego o długości 64 bajtów.
- matura - informacja o tym, czy słuchacz posiada maturę, czy nie; pole typu logicznego.
- płeć - informacja o płci słuchacza; pole typu logicznego (prawda - mężczyzna, fałsz - kobieta :)
- rok - rok na którym aktualnie jest słuchacz; liczba całkowita, krótka.

Tablica 19: uczniowie

nazwa	typ	spójność
id_ucznia	SMALLINT	NOT NULL
nazwisko	CHARACTER (30)	NOT NULL
imie1	CHARACTER (20)	NOT NULL
imie2	CHARACTER (20)	
miejscowosc	CHARACTER (30)	
ulica	CHARACTER (30)	
lokal	CHARACTER (20)	
kod	CHARACTER (6)	
pesel	CHARACTER (11)	
data_urodzenia	DATE	
miejsce_ur	CHARACTER (30)	
telefon	CHARACTER (15)	
data_przyjecia	DATE	
data_ukonczenia	DATE	
imie_ojca	CHARACTER (20)	
ukonczona_szkola	CHARACTER (64)	
matura	BOOL	
plec	BOOL	
rok	SMALLINT	
imie_matki	CHARACTER (20)	

4.3.20 Tabela nauczyciele

W tej tabeli zapisane będą informacje na temat wykładowców w studium.

- id_nauczyciela - numer jednoznacznie identyfikujący każdego wykładowcę; liczba całkowita krótka, nie może być wartością pustą.* nazwisko - nazwisko wykładowcy; pole typu łańcuchowego o długości 30 bajtów, nie może być wartością pustą.
- imie - pierwsze imię wykładowcy; pole typu łańcuchowego o długości 30 bajtów, nie może być wartością pustą
- miejscowosc - miejscowość w jakiej zameldowany jest wykładowca; pole typu łańcuchowego o długości 30 bajtów.
- ulica - ulica na jakiej zameldowany jest wykładowca; pole typu łańcuchowego o długości 20 bajtów.
- lokal - dom / mieszkanie w miejscu zameldowania wykładowcy; pole typu łańcuchowego o długości 15 bajtów.
- kod - kod pocztowy wykładowcy; poloe typu łańcuchowego o długości 6 bajtów.
- id_tytulu - numer identyfikujący tytuł naukowy wykładowcy; liczba całkowita krótka.
- tel_domowy - telefon domowy wykładowcy; pole typu łańcuchowego o długości 15 bajtów
- tel_sluzbowy - telefon służbowy wykładowcy; pole typu łańcuchowego o długości 15 bajtów.
- nr_konta - numer konta bankowego do rozliczeń ze szkołą; pole typu łańcuchowego o długości 64 bajtów.

Tablica 20: nauczyciele

nazwa	typ	spójność
id_nauczyciela	SMALLINT	NOT NULL
nazwisko	CHARACTER (30)	NOT NULL
imie	CHARACTER (20)	NOT NULL
miejscowosc	CHARACTER (20)	
ulica	CHARACTER (30)	
lokal	CHARACTER (15)	
kod	CHARACTER (6)	
id_tytulu	SMALLINT	
tel_domowy	CHARACTER (15)	
tel_sluzbowy	CHARACTER (15)	
nr_konta	CHARACTER (64)	

4.3.21 Tabela dokum_ucz

Tabela wiążąca dokumenty ze słuchaczem.

- id_dokumentu - jednoznaczny numer identyfikujący dokument; liczba całkowita krótka, nie może być wartością pustą.
- id_ucznia - jednoznaczny numer identyfikujący słuchacza; liczba całkowita krótka, nie może być wartością pustą.

Tablica 21: dokum_ucz

nazwa	typ	spójność
id_dokumentu	SMALLINT	NOT NULL
id_ucznia	SMALLINT	NOT NULL

4.3.22 Tabela oceny

W tabeli będą zapisane oceny jakie może zdobyć słuchacz (1.0, 1.5, 2.0... etc)

- id_oceny - jednoznaczny numer identyfikujący każdą ocenę; liczba całkowita krótka; nie może być wartością pustą.
- ocena - ocena; liczba zmiennoprzecinkowa; nie może być wartością pustą.

Tablica 22: oceny

nazwa	typ	spójność
id_oceny	SMALLINT	NOT NULL
ocena	FLOAT	NOT NULL

4.3.23 Tabela zaliczenie

W tabeli zapisane będą dane dotyczące zaliczenia każdego semestru przez słuchacza.

- id_ucznia - numer jednoznacznie identyfikujący słuchacza; liczba całkowita krótka, nie może być wartością pustą.
- id_przedmiotu - numer jednoznacznie identyfikujący przedmiot, liczba całkowita krótka, nie może być wartością pusta.
- id_oceny - identyfikator oceny; liczba całkowita krótka; nie może być wartością pustą.
- semestr - semestr na końcu którego słuchacz zdobył ocenę; liczba całkowita krótka, nie może być wartością pustą.
- termin - termin w którym słuchacz zdobył ocenę (1 - pierwszy, 2 - poprawkowy, 3 - egzamin); liczba całkowita, krótka, nie może być wartością pustą.
- rok_kal - rok kalendarzowy w którym słuchacz dostał ocenę.

Tablica 23: zaliczenia

nazwa	typ	spójność
id_ucznia	SMALLINT	NOT NULL
id_przedmiotu	SMALLINT	NOT NULL
id_oceny	SMALLINT	NOT NULL
semestr	SMALLINT	NOT NULL
termin	SMALLINT	NOT NULL
rok_kal	SMALLINT	

4.4 Klucze

4.4.1 Klucze podstawowe

W każdej tabeli będą występowały klucze podstawowe, jedno-, lub wielopolowe.

1. Tabela „dni_tygodnia" klucz jednopolowy w polu : „id_dnia".
2. Tabela „dokumenty" klucz jednopolowy w polu : „id_dokumentu".
3. Tabela „email_naucz" klucz jednopolowy w polu : „email"
4. Tabela „email_ucznia" klucz jednopolowy w polu : „emial"
5. Tabela „godziny" klucz jednopolowy w polu : „id_lekcji"
6. Tabela „miesiace" klucz jednopolowy w polu : „id_miesiaca"
7. Tabela „naucz_praca" klucz jednopolowy w polu : „id_pracy"
8. Tabela „naucz_przed" klucz wielopolowy w polach : „id_przedmiotu" i „semestr"
9. Tabela „oplaty" klucz wielopolowy w polach : „id_ucznia", „id_miesiaca", „rok_kal"
10. Tabela „plan_zajec" klucz wielopolowy w polach : „id_lekcji", „id_dnia", „id_przedmiotu", „semestr"
11. Tabela „przedmioty" klucz jednopolowy w polu „id_pracy"
12. Tabela „praca_dyplomowa" klucz jednopolowy w polu : „id_pracy"
13. Tabela „przedmioty" klucz jednopolowy w polu „id_przedmiotu"
14. Tabela „ucz_praca" klucz jednopolowy w polu „id_ucznia"
15. Tabela „wojsko" klucz jednopolowy w polu „id_wojsko"
16. Tabela „zaswiadczenia" klucz jednopolowy w polu „id_zaswiadczenia"
17. Tabela „tytuly" klucz jednopolowy w polu „id_tytulu"
18. Tabela „wojsko_uczen" klucz jednopolowy w polu „id_ucznia"
19. Tabela „uczniowie" klucz jednopolowy w polu „id_ucznia"

20. Tabela „nauczyciele” klucz jednopolowy w polu „id_nauczyciela”

21. Tabela „dokum_ucz” klucz wielopolowy w polach „id_dokum” i „id_ucznia”

22. Tabela „oceny” klucz jednopolowy w polu „id_oceny”

23. Tabela „zaliczenie” klucz wielopolowy w polach : „id_ucznia”, „id_przedmiotu”, „semestr”, „termin”

4.4.2 Klucze obce

W bazie danych występuje bardzo wiele relacji, więc niezbędne są oprócz kluczy podstawowych również klucze obce.

1. W tabeli „email_naucz” istnieje klucz obcy w polu „id_nauczyciela” w połączeniu z tabelą „nauczyciele”.

2. W tabeli „email_ucznia” istnieje klucz obcy w polu „id_ucznia” w połączeniu z tabelą „uczniowie”.

3. W tabeli „naucz_praca” istnieją dwa klucze obce w polach : „id_pracy” w połączeniu z tabelą „prace_dyplomowe” oraz „id_nauczyciela” w połączeniu z tabelą „nauczyciele”.

4. W tabeli „naucz_przed” istnieją dwa klucze obce w polach : „id_nauczyciela” w połączeniu z tabelą „nauczyciele” oraz „id_przedmiotu” w połączeniu z tabelą „przedmioty”.

5. W tabeli „opłaty” istnieją dwa klucze obce w polach : „id_ucznia” w połączeniu z tabelą „uczniowie”, oraz „id_miesiąca” w połączeniu z tabelą „miesiące”.

6. W tabeli „plan_zajec” istnieją cztery klucze obce w polach : „id_lekcji” w połączeniu z tabelą „godziny”, w polu „id_dnia” w połączeniu z tabelą „dni_tygodnia”, w polu „id_przedmiotu” w połączeniu z tabelą „przedmioty”, oraz w polu „id_nauczyciela” w połączeniu z tabelą nauczyciele.

7. W tabeli „praca_dyplomowa” istnieje klucz obcy w polu „id_promotora” w połączeniu z polem „id_nauczyciela” z tabeli „nauczyciele”

8. W tabeli „ucz_praca” istnieją dwa klucze obce w polach : „id_ucznia” w połączeniu z tabelą „uczniowie”, w polu „id_pracy” w połączeniu z tabelą „id_pracy”, oraz pięć kluczy w polach „ocena_promotora”, „ocena_recenzenta”, „obrona”, „pytania” i „dyplomowa” w połączeniu z polem „id_oceny” w tabeli „oceny”.

9. W tabeli „zaswiad_uczen” istnieją dwa klucze obce w polach : „id_ucznia” w połączeniu z tabelą „uczniowie”, oraz „id_zaswiadczenia” w połączeniu z tabelą „zaswiadczenia”.

10. W tabeli „wojsko_uczen” istnieją dwa klucze obce w polach : „id_ucznia” w połączeniu z tabelą „uczniowie”, „id_wojsko” w połączeniu z tabelą „wojsko”

11. W tabeli „dokum_ucz” istnieją dwa klucze obce w polach : „id_dokumentu” w połączeniu z tabelą „dokumenty”, „id_ucznia” w połączeniu z tabelą „uczniowie”.

12. W tabeli „zaliczenie” istnieją trzy klucze obce w polach : „id_ucznia” w połączeniu z tabelą „uczniowie”, „id_przedmiotu” w połączeniu z tabelą „przedmioty” oraz „id_oceny” w połączeniu z tabelą oceny.

4.5 Prawa spójności dziedziny

Aby zabezpieczyć bazę przed nonsensownymi i paradoksalnymi danymi wprowadzamy pewne prawa, które należy przestrzegać wprowadzając dane do bazy. Pewne z nich możliwe są do zaimplementowania na poziomie samego systemu zarządzania bazą danych (PostgreSQL), jednak o większość z nich musi się zatroszczyć osoba tworząca aplikację współpracującą z bazą danych. W naszym przypadku jest to program napisany w PHP3.
Ograniczenia występujące w każdym polu to ograniczenia dotyczące maksymalnego rozmiaru danych w polu jak i jego nie-pustej wartości - podane podczas definicji tabel w rozdziale 1.3. Ponadto aplikacja pracująca na bazie powinna także wprowadzać dane odpowiednich typów, oraz sprawdzać unikalność indeksów. Wartości wpisywane do pól będących kluczami obcymi powinny istnieć. Należy również pamiętać o kaskadowym usuwaniu danych z bazy.
W wypadku wystąpienia błędu najpierw powinna zareagować aplikacja, wyświetlając odpowiedni komunikat, nie dopuszczając do sytuacji w której sama baza danych wyświetli niezrozumiały najprawdopodobniej dla użytkownika komunikat. Komunikaty PostgreSQL'a o ewentualnych błędach przydadzą się administratorowi bazy podczas implementacji bazy jak i późniejszemu administrowaniu.
Oprócz ograniczeń zdefiniowanych podczas opisów tabel i pól istnieją ponadto indywidualne ograniczenia niektórych pól zabezpieczające bazę.

1. W tabelach : „oplaty”, „zaliczenie” pole „rok_kal” musi być realne, czyli powinno się mieścić w przedziale ¡1980, 2080¿; domyślnie powinien być podawany rok bieżący.

2. W tabelach : „naucz_przed”, „zaliczenie” pole „semestr” powinno się mieścić w przedziale ¡1,4¿

3. W tabeli „oplaty” pole „data_wplyniecia” nie powinno być większe niż aktualna data; domyślnie powinna być podawana aktualna data.

4. W tabeli „oplaty” pole „kwota” nie może być mniejsze od 0.

5. W tabeli „plan_zajęć” pole „rok_nauki” powinien być równy 1, lub 2.

6. W tabeli „praca_dyplomowa” pole „data_obrony” podczas wpisywania przez użytkownika nie powinna być datą która już minęła.

7. W tabeli „przedmioty” pole : „ilosc_sem” powinno być wartością z przedziału ¡1,4¿, przy czym warunek, że : „ilosc_sem+pocz_sem(5” zawsze bezwzględnie powinien być spełniony.

8. W tabeli „przedmioty” pole : „pocz_sem” powinno być wartością z przedziału ¡1,4¿, przy czym warunek, że : „ilosc_sem+pocz_sem(5” zawsze bezwzględnie powinien być spełniony.
9. W tabeli „zaswiad_uczen” pole „data_pobrania” nie powinno zawierać wartości większej niż aktualna data; domyślnie powinna być podana data aktualna.
10. W tabeli „uczniowie” pole „data_przyjecia” nie powinno zawierać wartości większej niż data aktualna; domyślna wartość dla tego pola to data aktualna.
11. W tabeli „uczniowie” pole „data_urodzenia” nie powinna zawierać wartości większej niż data aktualna.
12. W tabeli „uczniowie” pole „data_ukończenia” nie powinna być większa niż data aktualna.
13. W tabeli „uczniowie” pole „rok” powinno zawierać wartość : 1, lub 2.
14. W tabeli „zaliczenie” pole „termin” powinno zawierać wartość : 1, 2, lub 3.

5 System zarzĄdzania bazĄ danych : PostgreSQL

5.1 Wstęp do PostgreSQLa

Postgres jest obiektowo-relacyjną bazą danych zaprojektowaną w roku 1986 na kalifornijskim Uniwersytecie Berkeley. Autorem i wykonawcą projektu był profesor Michel Stonebraker aż do roku 1993 w którym zakończono projekt Postgres wersją „Postgres release 4.2”. Był to na tyle prosty i „silny” system zarządzania bazami danych, że znalazł wielu użytkowników i zwolenników.
W roku 1994 panowie Andrew Ju i Jolly Chen dodali do Postgresa strukturalny język SQL oraz wprowadzili wiele korzystnych zmian. Nowsza wersja nazywała się Postgres95, była szybsza o 20-30% i bardziej rozbudowana od swojego poprzednika. W 1996 roku powstał PostgreSQL, który posiadał więcej wbudowanych typów i działał szybciej od poprzedników, nowsze wersje numerowano od 6.0. Obecnie pojawiła się wersja 7.0 i jest dostępna na www.postgresql.org jako testowa wersja beta.

5.2 Instalacja pakietu postgresql_6.5.3

Istnieje kilka sposobów instalowania pakietu postgresql. Można użyć RPM’ów, które są bardzo proste i szybkie w instalacji, lub samemu skompilować źródła. Pierwszy sposób posiadający zaletę prostoty i szybkości nie pozwala ingerować użytkownikowi w strukturę pakietu (katalogi, moduły...), oprócz tego któraś wersja posiadała błędy i poważnie zraziła mnie do instalacji tego typu.
Najnowsze źródła postgresa można ściągnąć ze strony : www.postgresql.org bezpłatnie.
Należy rozpakować źródła, dajmy na to do katalogu : /usr/src

```
cp postgresql-6.5.3.tar.gz /usr/src/
cd /usr/src
gzip -d postgresql-6.5.3.tar.gz
tar -xvf postgresql-6.5.3.tar
```

uruchomić program konfigurujący,

```
cd postgresql-6.5.3/src
./configure
```

można dodać następujące opcje :

```
--with-odbc    // zainstaluje sterowniki ODBC
--with-perl // zainstaluje bibliotekę perla
--prefix=/usr/local/pgsql  // postgresql zostanie zainstalowany
   // do katalogu /usr/local/pgsql
```

właściwa kompilacja programu :

```
gmake all
gmake install
```

program zostanie zainstalowany do odpowiednich katalogów
Użytkownikiem, który będzie miał prawa dostępu do bazy będzie „postgres”, może być to każdy inny użytkownik wyłączając jednak roota. Należy wiec zalogować się jako ten właśnie użytkownik jeżeli jeszcze on nie istnieje należy stworzyć

```
useradd -d /usr/local/pgsql postgres   // tworzy użytkownika
su postgres // loguje się jako ten użytkownik
```

Zmiany jakie należy wprowadzić w pliku .bash_profile w domowym katalogu postgresa :

```
export LD_LIBRARY_PATH=/usr/local/pgsql/lib
export PATH=$PATH:/usr/local/pgsql/bin
export MANPATH=$MANPATH:/usr/local/pgsql/man
export PGLIB=/usr/local/pgsql/lib
export PGDATA=/usr/local/pgsql/data

export USERNAME BASH_ENV PATH MANPATH PGLIB PGDATA
```

Następnym krokiem będzie uaktualnienie systemu z wprowadzonymi zmianami :

```
source .bash_profile
```

Kolejno należy stworzyć bazę danych, która będzie wyjściową bazą danych dla kolejnych. Instrukcja :

```
initdb
```

…stworzy pustą bazę danych „template1” w katalogu wskazanym przez zmienna PGDATA. Jeżeli użytkownik nie zdefiniuje powyższych zmiennych należy użyć parametrów programu „initdb”
Uruchomienie demona, który będzie komunikował się z baza wygląda następująco :

```
postmaster -i
```

Parametr -i włącza komunikacje poprzez protokoły TCP/IP. Postmaster domyślnie działa na porcie 5432, jeżeli chcemy uruchomić go na innym, to do tego służy parametr -p ¡port¿. Należy jednak pamiętać na którym porcie działa demon, gdyż komunikując się z nim musimy użyć tego właśnie portu. Jeżeli chcemy, aby działanie postmastera zapisywało się do pliku z logami wówczas uruchomić należy go w poniższy sposób

```
postmaster -i >pg_log 2>&1&
```

postmaster zostanie uruchomiony w tle a wszystkie jego działania zostaną zapisane w pliku pg_log. Jeżeli chcemy, żeby serwer bazy danych uruchamiał się podczas startu systemu i zamykany zostawał razem z systemem należy dodać poniższy skrypt do katalogu /etc/rc.d/init.d/ i utworzyć odpowiednie linki.

```
--- cut here ---

#!/bin/sh
#
# Ten skrypt odpala serwer PostgreSQL - postmaster.
#

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# See how we were called.
case "$1" in
 start)
  # Start daemons.
  if [ ! -f /var/lock/subsys/postgres ]
   then
    echo -n "Starting postgres daemon:"
    su postgres -c "/usr/local/pgsql/bin/postmaster \
-D /usr/local/pgsql/data -p 5432 -i \
>/usr/local/pgsql/pg_log 2>&1&"
    echo -n " postmaster"
    echo
    touch /var/lock/subsys/postgres
  else
    echo -n " (already running)"
  fi
   ;;
  stop)
   # Stop daemons.
```

```
   echo -n "Shutting down postgres daemon: "
   killall -TERM postmaster 2>/dev/null
   killall -TERM postgres 2>/dev/null
   echo
  rm -f /var/lock/subsys/postgres
;;
  *)
   echo "Usage: postgres {start|stop}"
   exit 1
esac
exit 0
--- cut here ---
```

5.3 Pozostałe programy pakietu PostgreSQL

- createdb - tworzy nową bazę danych na podstawie bazy 'template' osoba, która tworzy tę bazę staje się automatycznie jej administratorem i posiada prawa zapisu i odczytu bazy.
- createuser - tworzy nowego użytkownika baz danych, który będzie miał prawo odczytu i zapisu do baz. Prawa w bazie ustala się używając komendy GRANT.
- destroydb - usuwa istniejącą bazę danych
- destroyuser - usuwa istniejącego użytkownika
- initdb - tworzy nowy system zarządzania baza danych
- initlocation - tworzy miejsce w którym przechowywane są dane z bazy
- pg_dump - konwertuje zawartość bazy danych do pliku w formie skryptu zawierającego zapytania SQL
- pg_dumpall - konwertuje wszystkie bazy danych do skryptowego pliku
- postgres - uruchamia serwer bazy danych
- postmaster - program, który zarządza wieloma serwerami postgres
- psql - jest to klient postgresa.
- vacuumdb - czyści zawartość bazy danych

5.4 Sposoby dostępu do bazy danych:

Istnieje kilka sposobów komunikowania się użytkownika z bazą danych:

1. klinet - psql
2. JDBC
3. Zagniezdzony SQL
4. pglib/pglib++

5. php3/php/phtml
6. heitml

1. PSQL - Komunikacja z bazą danych odbywa się w trybie tekstowym po uruchomieniu programu „psql". PSQL łączy się z odpowiednim portem (-p) serwera (-h), i otwiera zadana bazę danych (-d). Można tez wprowadzić dane z pliku (-f). Jest to wygodne, gdyż można sobie przygotować skrypt z danymi do wprowadzenia i jednym poleceniem dokonać update'u bazy.
2. JDBC - Java Data Base Connectivity - jest to sterownik umożliwiający połączenie klienta z bazą danych serwera przy użyciu języka Java. Zapytania kierowane do bazy, oraz interpretacja wyników występuje po stronie klienta, co jest wygodnym rozwiązaniem gdy serwer jest obciążony.
3. Zagnieżdżony SQL - Za pomocą zagnieżdżonego SQLa tworzy się aplikacje CGI(Common Gateway Interface) używając preprocesora (ecpg), który dostępny jest w pakiecie postgresa.
4. pglib/pglib++ - w wyniku tego tworzy się aplikację CGI, używając bibliotek dostępnych razem z pakietem postgresa.
5. php/php3/phtml - PHP jest to skryptowt język interpretowany przez moduł serwera www. Stosując ten sposób komunikacji z bazą danych powstał opisywany w pracy system.

6 Serwer WWW (World Wide Web)

Serwer www powinien być zainstalowany na serwerze UNIXowym, gdyż przy jego użyciu będziemy komunikować się z bazą danych. Najbardziej popularnym serwerem jest apache, którego darmowo można ściągnąć ze strony www.apache.org. Najwygodniej jest jednak posłużyć się jakimś bliskim mirrorem. Istnieją w Polsce dwa mirrory, z których ściągnięcie apache'a nie trwało nawet pół minuty.

6.1 Wybór odpowiedniej wersji apache'a oraz sposobu instalacji.

W poniższym rozdziale pobieżnie omówię sposoby instalacji serwera Apache 1.3.* zarówno ze źródeł jak i gotowych pakietów rpm.

6.1.1 Apache 1.3.6 pakiet rpm z dystrybucji RedHata 5.2

Do instalacji z pakietu rpm służy polecenie :

```
rpm -Uvh apache-1.3.6.i386.rpm
```

Moduły są włączane jako DSO (Dynamic Shared Object).
W dystrybucji obecne są moduły :

```
libperl.so
mod\_php.so
libphp3.so
```

. . . których nie ma standardowo w apache'u. Jednakże pewne zastrzeżenia budzi „libphp3.so", który potrzebny jest nam do komunikacji z baza danych PostgreSQL. Otóż twórcy pakietu nie pomyśleli o PostgreSQL i funkcja połączenia pg_connect jest wyłączona PHP3 trzeba sobie samemu doinstalować - o tym dalej.

6.1.2 Apache-1.3.6 pakiet rpm z dystrybucji Mandrake 6.0

Do instalacji służy polecenie :

```
rpm -Uvh apache-1.3.6.rpm
```

RPM Mandrake'a zawiera w sobie libphp3.so, w wersji 3.0.7 z pełnym supportem dla PostgreSQL'a. Żeby zainstalować tego RPMa trzeba było złamać zależności (ang. dependencies), gdyż wymagał PostgreSQL-6.4, a mamy już zainstalowanego 6.5.3. Oprócz tego wymagane były jeszcze biblioteki z : freetype i readline. rpm'y tych pakietów znajdują się na płycie Mandrake'a.
Moduł libphp3.so potrzebuje libpq.so.2.0, żeby mógł się załadować libpq.so.2.0 znajduje się w katalogu postgresa tutaj w : /usr/local/pgsql/lib ten katalog jest nie włączony do przeszukiwania standardowo. Dobrze jest utworzyć linka o takiej samej nazwie gdzieś na ścieżce bibliotek.

```
ln -s /usr/local/pgsql/lib /usr/lib/libpq.so.2.0
```

Teraz demon httpd będzie mógł spokojnie wystartować podczas startu systemu.

6.1.3 Apache-1.3.6 (źródło)

Najlepszą metodą jest zainstalowanie apache'a ze źródeł - jak wszystkiego zresztą. . .
Skonfigurowałem swojego apache'a z następującymi opcjami :

```
./configure --prefix=/usr/local/apache \
--sysconfdir=/etc/httpd --datadir=/home/httpd \
--libexecdir=/usr/lib/apache \
--enable-module=rewrite --enable-shared=rewrite \
--enable-module=so \
--enable-module=env --enable-shared=env \
--enable-module=setenvif --enable-shared=setenvif \
--enable-module=mime --enable-shared=mime \
--enable-module=negotiation --enable-shared=negotiation \
--enable-module=alias --enable-shared=alias \
--enable-module=userdir --enable-shared=userdir \
--enable-module=dir --enable-shared=dir \
--enable-module=autoindex --enable-shared=autoindex \
--enable-module=access --enable-shared=access \
--enable-module=auth --enable-shared=auth \
--enable-module=asis --enable-shared=asis \
--enable-module=include --enable-shared=include \
--enable-module=cgi --enable-shared=cgi \
--enable-module=actions --enable-shared=actions \
--enable-module=status --enable-shared=status \
```

```
--enable-module=log_config --enable-shared=log_config \
--enable-module=imap --enable-shared=imap
```

6.1.4 Jakie moduły zainstalować. (błyskawiczny przegląd)

- mod_access - umożliwia sterowanie dostępem do zasobów. Można zabronić dostępu z jakiegoś adresu IP, lub hosta (konfiguracja w pliku access.conf):

  ```
  deny from zly.host.ae.krakow.pl
  deny from 149.156.2*
  ```

- mod_actions - umożliwia uruchamianie skryptów CGI plików pewnych typów
- mod_alias - włącza działanie aliasów i przekierowań np.

  ```
  Alias /icons/ "/home/httpd/icons/"
  ```

- mod_asis - pozwala przesylac pliki (send-as-is), czyli apache nie doda do pliku nagłówka HTTP.
- mod_auth - umożliwia dostęp do zasobów tylko zarejestrowanym użytkownikom
- mod_auth_anon - umożliwia dostęp użytkownikom anonimowym po podaniu adresu e-mail zamiast hasła - podobnie jak podczas dostępu do serwisu ftp.
- mod_auth_db - sprawdzanie użytkownika odbywa się używając plików Berkley DB.
- mod_auth_dbm - podobnie jak wyżej, z tą różnicą, ze to pliki DBM
- mod_autoindex - indeks katalogu może być wygenerowany automatycznie przez serwer (jeśli nie istnieje plik indeksowy)
- mod_browser - pozwala przeglądarce na modyfikowanie zmiennych.
- mod_cern_meta - emuluje nagłówki CERN HTTP. działanie podobne do mod_asis
- mod_cgi - umożliwia uruchamianie programów CGI
- mod_cookies - włącza Netscape cookies.
- mod_digest - rozpoznanie użytkownika poprzez MD5
- mod_dir - supportuje DirectoryIndex
- mod_dld - moduł, który został zastąpiony przez mod_so (ale jest wymagany podczas kompilacji mod_perl'a)
- mod_dll - umożliwia ładowanie modułów z Widowzowskicg dll'ow. Moduł również zastąpiony przez mod_so
- mod_env - ustawia zmienne środowiskowe dla CGI i SSI.

- mod_example - demonstruje działanie API
- mod_expires - generuje nagłówek ęxpire”według kryteriów podanych przez użytkownika
- mod_headers - generuje nagłówki HTTP
- mod_imap - włącza działanie mapy bitowej jako pliku .map
- mod_include - umożliwia działanie SSI (Server Side Includes)
- mod_info - umożliwia przegląd ustawień serwera oraz załadowanych modułów
- mod_isapi - supportuje rozszerzenia ISAPI w Windowzie
- mod_log_* - określa sposób logowania na serwerze.
- mod_mime - określa typy MIME i zachowanie.
- mod_mime_magic - umożliwia rozpoznanie typu pliku po odczytaniu kilku jego bajtów
- mod_mmap_static - wczytuje do pamięci często używane, ale nie zmieniane pliki
- mod_negotiation - umożliwia bardziej niezawodne komunikowanie się przeglądarki z serwerem. (suport, rożnych typów, języków, i rodzaju kodowania znaków)
- mod_proxy - włącza cashowanie serwera proxy.
- mod_rewrite - dynamicznie nadpisuje podany URL
- mod_setenvif - ustawia zmienne zależnie od przeglądarki
- mod_so - wlacza dynamiczne sterowanie ładowaniem modułów
- mod_spelling - poprawia ewentualne błędy w adresie URL, które użytkownik mógł popełnić
- mod_status - przydatne dla administratora - wyświetla bieżące informacje na temat działania Apache'a
- mod_userdir - definiuje katalog użytkownika, w którym będzie znajdowała się jego strona www.
- mod_unique_id - przydziela unikalne id każdemu zadaniu przychodzącymi od klienta
- mod_usertrack - loguje klientów, którzy używają cookies

6.1.5 Kompilacja źródła:

Kompilacja rozpoczyna się poleceniem :

```
make
```

gdy komputer skończy wyświetlać informacje o aktualnie kompilowanym pliku, należy jeszcze zainstalować wszystko do odpowiednich katalogów. Do tego służy polecenie :

```
make install
```

Jeżeli wystąpił jakiś błąd to należy wykasować powstałe moduły :

```
make clean
```

i rozpocząć kompilację od nowa, należy też zwrócić uwagę, co działo się podczas konfigurowania. Być może nie mamy jakiegoś kompilatora, który jest wymagany, albo nie mamy zainstalowanej jakiejś konkretnej biblioteki. Jeżeli błąd będzie występował to trzeba już szukać przyczyn w dokumentacji pakietu. Co zrobić, żeby serwer httpd uruchamiał się podczas startu systemu ? Należy napisać skrypt i umieścić do w katalogu : /etc/rc.d/init.d
Oto taki przykładowy skrypt :

```
--- cut here ---

#!/bin/bash
#
#  /etc/rc.d/init.d/httpd
#
# Starts the httpd daemon
#
# chkconfig: 345 40 60
# processname: httpd

# Source function library.
. /etc/rc.d/init.d/functions

test -x /usr/local/apache/bin/httpd || exit 0


case "$1" in
  start)
   # Check if atd is already running
   if [ ! -f /var/lock/subsys/httpd ]; then
       echo -n 'Starting http daemon: '
       daemon /usr/local/apache/bin/httpd
       echo
       touch /var/lock/subsys/httpd
   fi
   ;;
  stop)
```

```
   echo -n 'Stopping http daemon: '
   killproc /usr/local/apache/bin/httpd
   echo
   rm -f /var/lock/subsys/httpd
   ;;
  reload|restart)
   $0 stop
   $0 start
   ;;
  status)
   status /usr/local/apache/bin/httpd
   ;;
  *)
   echo "Usage: /etc/rc.d/init.d/httpd {start|stop|restart|reload|status}"
   exit 1
esac

exit 0

--- cut here ---
```

Potem trzeba jeszcze utworzyć odpowiednie symlinki do tego skryptu w katalogach odpowiadających odpowiednim runlevelom. rc0.d - rc6.d (0-halt), (3-run), (6-reboot). Należy użyć polecenia „ln". Po tych zmianach należy zrestartować komputer.
Może się zdarzyć błąd podczas ładowania modułu... np wtedy gdy próbujemy włączyć moduł, który był skompilowany ze wcześniejszą wersja apache'a . Tak się zdarza, gdy doinstalowujemy moduł z pakietu rpm/ (np. libperl.so działający na apachu 1.3.3 nie działa na wersji 1.3.6)

6.1.6 Konfiguracja serwera httpd.

(W tej pracy podam tylko zmiany jakie zrobiłem w stosunku do defaultowej konfiguracji proponowanej przez producenta.
Konfiguracja serwera httpd składa się z trzech etapów:

1. konfigurowanie działania samego serwera - plik : httpd.conf
2. konfigurowanie zasobów - srm.conf
3. konfigurowanie praw dostępu do poszczególnych katalogów - plik access.conf Żeby nie porobić jakiś głupot zaleca się wszystko umieścić w pliku httpd.conf i pozostawienie plików srm.conf i access.conf w miarę możliwości pustych. Ja jednak z przyzwyczajenia rozdzielam to na trzy pliki.

Plik : httpd.conf miejsce, gdzie httpd będzie trzymał swoje pliki konfiguracyjne i logi.

```
ServerRoot="/etc/httpd"
```

należy pokazać gdzie znajdują się pozostałe pliki konfiguracyjne. Jeśli tego nie zrobimy to domyślnie przyjmie bieżący katalog.

```
ResourceConfig srm.conf
AccessConfig access.conf
```

żeby zadziałały polecenia php3 należy włączyć moduł w sekcji DSO (DYNAMIC SHARED OBJECT), wcześniej oczywiście trzeba moduł skompilować

```
LoadModule php3_module  /usr/lib/apache/libphp3.so
AddModule mod_php3.c
```

żeby serwer się jakoś nazywał należy mu nadać nazwę :

```
ServerName pablo.com.pl
```

aby w logach w miarę możliwości pozostawały nazwy klientów, a nie tylko ich adresy IP ustawiamy

```
HostnameLookups On
```

żeby logi zapisywały się w katalogu z ustawieniami serwera :

```
CustomLog /etc/httpd/logs/access_log common
```

tworzenie wirtualnego hosta : (numer IP powinien być odpowiednio inny)

```
NameVirtualHost 192.168.1.10

<VirtualHost 192.168.1.10>
ServerAdmin root@pablo.com.pl
DocumentRoot /home/maiden
ServerName maiden.com.pl
ErrorLog logs/maiden_error
CustomLog logs/maiden common
</VirtualHost>
```

Ustawienie wirtualnego hosta spowoduje, że będziemy mieli na naszym serwerze niby drugi serwer www. Oczywiście czynności te powinny być poprzedzone odpowiednimi zmianami w konfiguracji samego linux'a. Należy dodać odpowiedni wpis do pliku : /etc/hosts w tym przypadku plik powinien wyglądać podobnie :

```
127.0.0.1   pablo.com.pl   pablo
192.168.1.10   maiden.com.pl  maiden
```

Można „ręcznie" wpisać numery, używając edytora (np. vi), można użyć programu :

```
linuxconf
```

Resztę ustawień pozostawiłem tak jak sobie były

Plik : srm.conf Katalog, który zostanie wyświetlony po wywołaniu przez przeglądarkę:
"http://pablo.com.pl"to będzie główny (root) katalog serwera.

```
DocumentRoot "/home/httpd/html"
```

jeżeli damy użytkownikom prawo tworzenia własnych stron www, to należy serwer poinformować w jakim miejscu w katalogu użytkownika będą one dostępne :

```
UserDir public_html
```

należy podać nazwę pliku, który będzie używany jako domyślny index. Domyślnie podane jest : index.html, proponuje dodać również index.shtml, który będzie zawierał polecenia SSI w celu wzbogacenia strony www o efektywne ładowanie skryptów cgi oraz o same polecenia SSI (o czym dalej) Dodałem jeszcze index.php3 gdyż będę używał języka PHP3 do programowania w www.

```
DirectoryIndex index.html index.shtml index.php3
```

kolejno dodamy sobie suport polskiego języka :

```
AddLanguage pl .po
```

Żeby chodziło na serwerze php3 nie wystarczy tylko włączyć odpowiedni moduł. Musimy jeszcze poinformować serwer o rozszerzeniu plików php3, to znaczy jakie pliki ma przeszukiwać w poszukiwaniu poleceń php3.

```
AddType application/x-httpd-php3 .php3
AddType application/x-httpd-php3-source .phps
```

analogicznie postępujemy z plikami zawierającymi polecenia SSI

```
AddType text/x-server-parsed-html .shtml
AddHandler text/x-server-parsed-html .shtml
```

Plik : access.conf W tym pliku konfigurujemy prawa dostępu serwera www do poszczególnych katalogów.
sekcja dla katalogu żootśerwera. Ustawiam :

```
Options :
 Includes // żeby działały polecenia SSI
 FollowSymLinks // żeby Symlinki normalnie działały
 Indexes
 AllowOverride :
 AuthConf   // żeby dana strona była dostępna wówczas, gdy
  // użytkownik poda odpowiednie hasło Dostęp
// będzie możliwy po sprawdzeniu autentyczności
// użytkownika - ustawienia w pliku .htaccess

<Directory "/home/httpd/html">
Options Indexes FollowSymLinks Includes
AllowOverride AuthConfig
Order allow,deny
Allow from all
</Directory>
```

w sekcji, w której ustawiamy prawa dostępu do katalogu zawierającego skrypty cgi, czyli katalogu /home/httpd/cgi-bin należy włączyć opcje umożliwiającą uruchomienie programu cgi, oraz SSI.

```
<Directory "/home/httpd/cgi-bin">
AllowOverride None
Options ExecCGI Includes
Order allow,deny
Allow from all
</Directory>
```

Oczywiście włączenie tych opcji zwiększa możliwość ataków na nasz serwer, trzeba się wiec dobrze zastanowić komu przydzielimy prawa do uruchamiania programów CGI.

Konfiguracja praw dostępu w pliku .htaccess Konfiguracja w pliku .htaccess - indywidualne ustawienia do każdego katalogu. Jeśli tylko jest włączona opcja AuthConfig w AllowOverride to podczas dostępu do katalogu przeczytany zostanie najpierw plik .htaccess np :

```
AuthUserFile /etc/httpd/htaccess_pass
AuthType Basic
AuthName "Policealne Studium Informatyki"
require user test
```

w którym : AuthUserFile to ścieżka do pliku z hasłami (plik ten nie powinien znajdować się w katalogu bieżącym, lub jakimś katalogu do którego dostęp może mieć klient www.). Do stworzenia pliku z hasłami służy program htpasswd, który znajduje się w katalogu z binariami apache. Żeby stworzyć plik używamy parametru -c jeżeli plik istnieje parametr ten spowoduje, że zostaną usunięte poprzednie dane. Polecenie :

```
/usr/local/apache/bin/htpasswd -c /etc/httpd/ktaccess_pass test
```

stworzy plik i zapyta się o nowe hasło dla użytkownika 'test'

- AuthType - sposób w jaki użytkownik zostanie zidentyfikowany
- AuthName - to jest nazwa która zostanie wyświetlona podczas pytania browsera o nazwę użytkownika i jego hasło
- require user test - mówi o tym, ze do otworzenia strony potrzebny jest użytkownik 'test' i oczywiście hasło które używa użytkownik.

Można jeszcze ustawić, żeby dostęp był możliwy dla kilku użytkowników Należy wówczas zdefiniować grupę użytkowników i odpowiednio użyć opcji AuthGroupFile i require group xxxx.

7 PHP

Źródła PHP znajdują się na stronie : www.php.net i są dostępne darmowo.

7.1 Sposób instalacji PHP3

Można wkompilowac go w apache'a, ale tego nie polecam, gdyż plik wynikowy : httpd zajmie wtedy o jakieś 1.4 MB więcej... to dużo, gdyż używając DSO (Dynamic Shared Objects) httpd zajmuje tylko 230 kB. Do przyłączenia tego modułu użyjemy DSO, czyli skompilujemy go do postaci:

```
libphp3.so
```

Tak jest lepiej, bo gdy nie będziemy chcieli go używać, łatwo można wyłączyć ten moduł - w pliku httpd.conf wystarczy wziąć w komentarz linie w której go ładujemy, nie trzeba więc będzie jeszcze raz kompilować apache'a w momencie gdy php3 już nie będzie nam potrzebne.

7.2 Instalacja

Najpierw należy rozpakować źródło do odpowiedniego katalogu :

```
cp php-3.0.12.tar.gz
gzip -d php-3.0.12.tar.gz
tar -xvf php-3.0.12
```

7.2.1 Konfiguracja:

```
cd php-3.0.12
./configure --with-pgsql --with-apxs=/usr/local/apache/bin/apxs
```

Metoda z opcją –with-shared-apache=DIR , która teoretycznie powinna dawać takie same rezultaty niestety wywracała się podczas kompilacji. Dokładnie podczas likowania jakiejś tam biblioteki. (ld) Program apxs służy do tworzenia modułów dla apache'a i będzie się znajdował w wyżej podanym katalogu. W zasadzie do tej metody nie potrzeba źródeł apache'a których wymaga się podczas skompilowania modułu w plik wykonywalny ELF. Jako rozszerzenie opcji –with-pgsql można podać katalog gdzie znajduje się PostgreSQL - standardowo jest to /usr/local/pgsql

7.2.2 Kompilacja

Do kompilowania służy polecenie:

```
make
make install
```

Jeżeli pojawi się jakiś błąd należy wyczyścić powstałe moduły (make clean) i zacząć od nowa. W przypadku, gdy błąd będzie się powtarzał należy zobaczyć co na ten temat piszą w dokumentacji.
Aby moduł php zaczął działać należy jeszcze włączyć odpowiednie opcje w plikach : httpd.conf i srm.conf (patrz poprzedni rozdział). Potem konieczne jest tylko ponowne uruchomienie serwera www:

```
httpd restart
```

8 Problemy jakie niesie za soba zastosowanie PHP3 I PostgreSQLa

8.1 Security

- czyli problem odpowiedniego zabezpieczenia danych i dostępu do bazy. Ogólnie wiadome jest, ze nie ma stuprocentowej możliwości zabezpieczenia serwera przed włamaniem. W przypadku naszej bazy danych jest podobnie, pomimo postawienia zabezpieczenia istnieje możliwość włamania Zastanówmy się więc w jaki sposób się zabezpieczyć, żeby zminimalizować prawdopodobieństwo niepożądanego zniszczenia bazy przez hackera.

8.1.1 Zabezpieczenie od strony apache'a

Z poziomu apache'a blokujemy dostęp do systemu zarządzającego baza danych, używając pliku .htaccess (opis dokładnego działania w rozdziale poświęconym apache'owi) Dostep do bazy danych ma użytkownik : nobody, gdyż on jest użytkownikiem serwera www na komputerze. Każdy użytkownik internetu ma możliwość połączenia się z baza danych i modyfikowania jej poprzez www. Możliwość taka istnieje tylko wówczas, gdy potencjalny hacker zna strukturę tabel występujących w bazie. Aby uniemożliwić użytkownikowi serwera zdobycie takiej wiedzy należy zmienić prawa dostępu do katalogu zawierającego system. Do niego dostęp będzie miał tylko „nobody".

8.1.2 Zabezpieczenie z poziomu PostgreSQLa

Istnieje kolejny sposób na poznanie struktury bazy : jeżeli użytkownik serwera ma równocześnie możliwość korzystania z systemu PostgreSQL, może on wówczas połączyć się z baza za pomocą klienta psql i podglądnąć nazwy tabel komenda : Należałoby jeszcze uniemożliwić dostęp do bazy innym użytkownikom od strony klienta psql.

9 Zakończenie

Eksperymentalna realizacja projektu automatyzacji procesów informacyjnych związanych z obsługą administracyjno-biurową Policealnego Studium w pełni potwierdziła zarówno słuszność przyjętej koncepcji jak i wartość darmowych narzędzi informatycznych powszechnie dostępnych w internecie (Linux, PostgreSQL, Appache) do jej realizacji. Już w aktualnej, niepełnej implementacji, system zawiera atrakcyjne dla użytkownika funkcje omówione w rozdziale 3 w formie FAQ. Ograniczony czas realizacji tego projektu nie pozwolił na pełną jego implementacje. W szczególności mowa o takich planowanych funkcjach jak:

1. Wyświetlanie zaświadczeń dla odpowiednich instytucji, że dany uczeń bierze udział w wykładach w Policealnum Studium Informatyki. Zaświadczenia takie wymagane są przez WKU.

2. „Ustawienia" - brakuje formularza, który pozwoliłby z poziomu przeglądarki modyfikować np. rodzaje dokumentów, kategorii wojskowej, nazwę serwera na którym zainstalowana jest baza danych.

3. System pomocy (FAQ) w formacie HTML'a dostępny bezpośrednio z przeglądarki na stronie systemu.

W przyszłości warto również pomysleć o rozbudowie systemu. Zdaniem autora, lista postulowanych rozszerzeń powinna zawierać takie pozycje jak:

1. Możliwość układania i zmieniania planu zajęć z równoczesnym systemem wykrywania konfliktów i zapobiegania ich powstawaniu. Jest to na tyle ważne, że powinno zostać potraktowane priorytetowo w kolejnych wersjach systemu. W opisie systemu zostawiam informacje na temat tabel do którego należy dorobić odpowiednie oprogramowanie w PHP.

2. Kolejnym pominiętym w realizacji tematem jest możliwość definiowania raportów z poziomu przeglądarki. To znaczy sumaryczne zestawienie do wydruku będzie tworzył użytkownik po stronie klienta. Jest to również ważne, gdyż obecnie system zawiera kilka raportów co staje się poważnym ograniczeniem. Tworzenie kolejnego wymagało by niewielkiej modyfikacji systemu. Ewentualnie alternatywnym do tego rozwiązaniem, aczkolwiek nie najlepszym jest stworzenie jeszcze kilku potrzebnych formularzy.

3. Kolejne co pozostaje do zrobienia w przyszłości to zachowanie w bazie danych na temat miejsca praktyk studentów

10 Literatura i źródła internetowe wykorzystane w pracy:

[Butz97]- Fred Butzen, Dorothy Forbes - Linux - Bazy Danych
[Kim96]- Eugene Eric Kim - CGI, Programowanie CGI - Przewodnik
Ponadto dokumentacje następujących programów dostępne w internecie na niżej wymienionych stronach:
Apache - http://www.apache.org
PostgreSQL - http://www.postgresql.org
PHP - http://www.php.net

Spis treści

Spis tablic