

Bazy danych – opis projektu

Przedstawić projekt oraz implementację systemu bazodanowego, umożliwiającego realizację funkcjonalności wynikających z określonego zastosowania (np. obsługa biblioteki, księgarni, hotelu, biura podróży, magazynu, itp.). Przyjmuje się, że dostęp do bazy danych aplikacji jest obsługiwany w trybie klient/serwer przez odpowiedni system zarządzania bazą danych, będący serwerem relacyjnej bazy danych (np. MySQL, Sybase, Oracle, PostgreSQL, MS SQL Server). Baza danych może być udostępniana klientom bezpośrednio przez serwer lub pośrednio przez inny serwer, np. serwer WWW (Apache, IIS) lub serwer aplikacji (JBoss, Glassfish).

W ramach realizacji zadania przedstawić kolejne etapy rozwoju bazy danych systemu (typ bazy danych oraz system zarządzania bazą danych są do wyboru, np. bazy relacyjne, relacyjno-obiektowe) oraz aplikacji (interfejsu) użytkownika (np. interfejs może być w postaci kwerend (zapytań) SQL kierowanych do bazy danych z poziomu narzędzi SQL dostępnych w ramach wybranego systemu baz danych, aplikacji desktopowej, internetowej lub mobilnej opracowanej w wybranym języku programowania).

W szczególności w projekcie powinny wystąpić następujące etapy:

- *faza analizy i specyfikacji wymagań*: projektowanie wstępne, sformułowanie wymagań funkcjonalnych (funkcje systemu) i нефункциональных (narzędzia i technologie, parametry wydajnościowe systemu, i inne); definiowanie sposobu działania systemu (rodzaj aplikacji – sposób dostępu do danych, reguły przetwarzania danych, ograniczenia dziedzinowe, grupy użytkowników i dostępne dla nich funkcje systemu); wykorzystanie wybranych diagramów projektowych (np. przypadków użycia UML wraz ze scenariuszami użycia, diagramów DFD, i innych);
- *faza projektowania*: projekt bazy danych oraz aplikacji (interfejsu) użytkownika (klienta);
 - definiowanie kategorii bazodanowych (typów encji), reprezentujących wykorzystywane struktury danych; określenie powiązań między kategoriami w oparciu o diagram związków encji (ERD) i wybraną notację (np. Chena, Martina, UML); model konceptualny;
 - utworzenie modelu logicznego bazy danych (relacyjny, obiektowy), np. w postaci modelu CDM; definiowanie struktur danych (relacji), np. encji, klas, oraz powiązań między nimi (np. identyfikatory główne encji, atrybuty encji);
 - wybór systemu bazodanowego i utworzenie modelu fizycznego bazy danych (PDM), uwzględniającego tabele, dostępne typy danych, klucze obce, ograniczenia parametrów; normalizacja schematów relacji, redukcja związków wiele-do-wielu;
 - projekt mechanizmów przetwarzania i sterowania dostępem do danych (proste operacje CRUD, perspektywy, migawki, synonimy, procedury, wyzwalacze, pliki indeksowe);
 - projekt aplikacji użytkownika: typ aplikacji (desktopowa, internetowa, interfejs SQL), architektura (np. oparta o wzorzec projektowy MVC), interfejs dostępu do bazy danych (np. ADO, Hibernate), funkcje biznesowe – struktura menu, interfejs graficzny (np. strony ASP, JSP, i inne); opis powiązań między strukturami danych np. w oparciu o diagramy klas UML;
 - grupy i uprawnienia użytkowników (role użytkowników); wykorzystywane mechanizmy bezpieczeństwa (na poziomie bazy danych i aplikacji), np. schematy uwierzytelniania, szyfrowanie danych w bazie danych, szyfrowanie komunikacji z bazą danych;
- *faza implementacji*: realizacja elementów systemu (bazy danych, aplikacji/interfejsu użytkownika – klienta bazy danych) z wykorzystaniem wybranych technologii i narzędzi (np. Oracle i język Java; MS SQL Server i języki programowania w środowisku MS .NET, MySQL i język PHP, i inne);

- *faza testowania poprawności działania systemu*: skonfigurowanie środowiska testowego (np. sieci lokalnej, maszyn wirtualnych) lub produkcyjnego; instalacja systemu bazodanowego; przeprowadzenie testów funkcjonalnych (akceptacyjnych), np. w oparciu o wybrane narzędzia (Selenium, WebInject);
- *faza oceny wydajności i optymalizacji systemu*: czasy przetwarzania zapytań, stopień wykorzystania zasobów systemu, wpływ obciążenia systemu na czasy odpowiedzi, i inne (zastosowanie specjalizowanych narzędzi, np. WCAT, WebInject).

Ustalić skład grup projektowych (grupy 1 – 2 osobowe) oraz podać tematy projektów (możliwy wybór z listy tematów przykładowych lub zaproponowanie własnego tematu); szkolenie BHP.

Przedstawić podczas zajęć i wysłać na maila osoby prowadzącej zajęcia opis wstępnych założeń projektowych (wzór na stronie WWW; termin: 4 zajęcia).

Realizować oraz prezentować podczas zajęć, na bieżąco, kolejne fazy rozwoju bazy danych i aplikacji klienckiej (np. demonstracja działania, prezentacje slajdów z przykładami).

Możliwa jest prezentacja elementów projektu również podczas konsultacji (zdalnych lub stacjonarnych).

Harmonogram

- **Zajęcia (1-4):** ustalanie składu grup projektowych; definiowanie tematów i wstępnych założeń projektowych.
- **Zajęcia (5-7):** wymagania funkcjonalne i нефункционалне aplikacji bazodanowej; diagramy wymagań (diagram i scenariusze przypadków użycia); konfigurowanie i testowanie wybranego systemu baz danych; testy operacji bazodanowych (CRUD) – przypomnienie.
- **Zajęcia (8-10):** projekt i implementacja bazy danych; modele bazy danych (konceptualny, logiczny, fizyczny); implementacja modelu fizycznego oraz operacji wykonywanych na bazie danych; wypełnienie bazy danymi; testy operacji bazodanowych przedstawionych na diagramie wymagań (use-case).
- **Zajęcia (11-12):** projekt i implementacja interfejsu użytkownika (aplikacji wykorzystującej bazę danych).
- **Zajęcia (13-14):** testy i ocena działania systemu bazodanowego; prezentacja spisu treści oraz wstępnej wersji sprawozdania z projektu.
- **Ostatnie zajęcia:** prezentacje końcowe.

Przedstawić podczas zajęć stronę tytułową oraz spis treści sprawozdania (wzór na stronie WWW), a następnie wysłać na adres e-mail osoby prowadzącej zajęcia końcowe sprawozdanie z projektu, które powinno zawierać podane niżej elementy (termin: *ostatnie zajęcia*).

1. Stronę tytułową.
2. Spis treści, spis rysunków, spis tabel, spis listingów.
3. Cel i zakres projektu.
4. Opis działania i schemat logiczny systemu (główne elementy składowe systemu i ich powiązanie). Grupy użytkowników. Wymagania funkcjonalne (funkcje systemu, np. diagram przypadków użycia UML, scenariusze użycia) i нефункционалне (technologie i narzędzia, parametry wydajnościowe systemu, bezpieczeństwo, i inne).
5. Założenia przyjęte podczas realizacji, np. rodzaj aplikacji użytkownika (interfejs SQL, desktopowa, internetowa), rodzaj bazy danych (relacyjna, obiektowa, relacyjno-obiektowa, i inne), mechanizmy sterowania przepływem i przetwarzania danych (zapytania, transakcje, wyzwalacze, procedury, migawki, perspektywy, synonimy).

6. Wykorzystywane środowiska, serwery baz danych, metody, narzędzia, technologie projektowania oraz implementacji systemu (np. język specyfikacji UML, wzorce projektowe, stosowane narzędzia wspomagające projektowanie).
7. Projekt bazy danych. Schematy relacji (struktura encji); model conceptualny (diagram ERD), logiczny (CDM) i fizyczny bazy danych (PDM). Normalizację schematów relacji.
8. Mechanizmy komunikacji, przetwarzania i sterowania przepływem danych (zapytania do bazy danych, perspektywy, migawki, procedury, wyzwalacze, transakcje, indeksowanie, pliki parametrów, procesy bazodanowe), a także wykorzystywane protokoły komunikacyjne.
9. Projekt prostej aplikacji (interfejsu) użytkownika. Architektura systemu i schemat komunikacji z bazą danych. Mechanizmy przetwarzania danych, dostępu do bazy danych (przetwarzanie transakcyjne, perspektywy, migawki, zapytania SQL, funkcje API, mapowanie relacyjno-objektowe, warstwa pośrednicząca, serwery aplikacji, serwery WWW). Wykorzystane mechanizmy bezpieczeństwa (np. sposób uwierzytelniania, uprawnienia grup użytkowników - role, szyfrowanie danych w bazie danych, szyfrowanie połączeń sieciowych). Struktura menu aplikacji, postać interfejsu graficznego.
10. Implementacja bazy danych. Implementacja wybranych elementów aplikacji użytkownika (interfejsu).
11. Sposób instalowania, uruchamiania i testowania aplikacji (testy funkcjonalne w oparciu o wybrane narzędzia, np. Selenium, WebInject, i inne). Konfigurowanie środowiska testowego (np. maszyny wirtualne, sieć komputerowa) i/lub produkcyjnego.
12. Ocena wydajności i optymalizację działania systemu (czasy przetwarzania zapytań, stopień obciążenia systemu, mechanizmy zwiększające wydajność i dostępność systemu, np. WCAT, Webinject).
13. Podsumowanie – przedstawienie głównych osiągnięć pracy, właściwości opracowanych rozwiązań i zastosowanych technologii, wnioski końcowe.
14. Literatura: wykorzystane pozycje bibliograficzne, źródła internetowe.

Przykładowe tematy

- Bazodanowy system obsługi hotelu, schroniska, biura podróży, salonu samochodowego.
- System monitorowania stanu magazynów, hurtowni, urzędów, zintegrowany z bazą danych.
- System obsługi wypożyczalni samochodów, książek, kaset DVD.
- System zakupów internetowych zintegrowany z bazą danych.
- System obsługi biblioteki, księgarni, urzędu, linii lotniczej.
- System obsługi bankomatów, kas biletowych.
- Metody dostępu do baz danych z poziomu stron WWW.
- Testowanie przetwarzania transakcyjnego.
- System przetwarzania danych oparty o aplikację mobilną.
- Projekt bezpiecznej bazy danych agencji ubezpieczeniowej.

Literatura

- [1] Górski J., Inżynieria oprogramowania w projekcie informatycznym, Mikom, Warszawa, 2000.
- [2] Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika, WNT, Warszawa, 2001.
- [3] Date C. J., Wprowadzenie do systemów baz danych, WNT, Warszawa, 2000.
- [4] Beynon-Davies P., Systemy baz danych, WNT, Warszawa, 2000.
- [5] Garcia-Molina H., Ullman J.D., Widom J., Systemy baz danych. Kompletny podręcznik. Wydanie II, Prentice Hall, New Jersey, 2011.
- [6] Poźniak-Koszałka I., Relacyjne Bazy danych w środowisku Sybase, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2004.
- [7] Mendrala D., Szeliga M., Praktyczny kurs SQL Wydanie II, Helion, Gliwice, 2011.
- [8] Celko J., SQL. Zaawansowane techniki programowania, PWN, Warszawa, 2008.
- [9] Kim W., Wprowadzenie do obiektowych baz danych, WNT, Warszawa, 1996.
- [10] Mazur H., Mazur Z., Projektowanie baz danych, PTI, Wrocław 2002.