Baza danych dla ewidencji magazynowej

1. Cel zadania

Celem zadania jest realizacja systemu bazodanowego, pozwalającego na kompleksową obsługę magazynów. System powinien składać się z aplikacji klienckiej oraz bazy danych MySQL.

2. Słowny opis projektu

a. Aplikacja kliencka

Aplikacja kliencka powinna umożliwiać pracownikom dodawanie, edycje usuwanie oraz realizacje zamówień. Ważne jest także aby umożliwiała dostęp do historii zamówień, ewidencji klientów czy dostawców, a także łatwe operowanie na zgromadzonych w bazie danych. Możliwe musi również być załączanie faktury czy tworzenie korekt.

b. Serwer oraz baza danych MySQL

Kontener danych zrealizowany zostanie za pomocą darmowego MySQL umieszczonego na darmowym serwerze, prawdopodobnie Apache. Musi umożliwiać realizację wymagań postawionych przez użytkownika i funkcji udostępnianych przez aplikację kliencką.

3. Wymagania funkcjonalne

Id	Nazwa	Opis	Priorytet
001	Obsługa zamówień zewnętrznych	System pozwala na przyjmowanie oraz obsługę zamówień od klientów	Wysoki
002	Obsługa zamówień wewnętrznych	System pozwala na przyjmowanie oraz obsługę zamówień od innych magazynów	Wysoki
003	Ewidencja towarów	System pozwala na ewidencje towarów	Wysoki
004	Podział na kategorie	System pozwala na podział towarów na kategorie	Średni

005	Przechowywanie informacji o pracownikach, klientach i dostawcach	System pozwala na przechowywanie informacji o pracownikach, klientach i dostawcach	Średni
006	Przeglądanie historii zamówień	System pozwala na przeglądanie historii zamówień	Średni
007	Ewidencja faktur	System pozwala na ewidencje faktur	Wysoki

4. Wymagania niefunkcjonalne

- System współpracuje z serwerem Apache oraz bazą danych MySQL
- System jest kompatybilny z systemami operacyjnymi Windows XP, Vista, 7 w wersjach 32 bitowych
- System powinien obsługiwać do 500 użytkowników jednocześnie.

5. Diagram ER bazy danych oraz propozycje indeksów

model.pdf

model_indeksy.pdf

Ponieważ model z indeksami znacznie traci na czytelności, załączamy także wersję bez nich.

6. Skrypt DDL

db_create.sql

7. Hierarchia menu

Menu aplikacji składać się będzie z następujących podmenu:

- Plik:
  - Zapisz parametry połączenia
  - Wczytaj parametry połączenia
  - Zapisz widok do pliku HTML

- Zakończ

Podstawową funkcjonalnością menu plik jest możliwość wyeksportowania parametrów połączenia do pliku oraz zapis aktualnie widocznej tabeli jako HTML.

- Baza danych:
  - Ustaw dane połączenia
  - Połącz
  - Rozłącz

Aby umożliwić użytkownikowi zmianę parametrów połączenia z bazą danych w prosty sposób.

- Zarządzaj:
  - Zamówienia wejściowe
  - Zamówienia wyjściowe
  - Kategorie
  - Części
  - Lokalizacje
  - Osoby:
    - Pracownicy
    - *Kontrahenci*
    - Dostawcy

Jest to kompleksowe menu służące do realizacji podstawowych funkcjonalności programu. Każda z opcji wyświetla na ekranie odpowiedni ekran, charakterystyczne dla siebie GUI, umożliwiające dodanie, usunięcie oraz modyfikację poszczególnych elementów.

- Raporty:
  - Zamówienia zrealizowane
  - Zamówienia oczekujące na realizację
  - Faktura zamówienia
  - Wykaz dostawców
  - Obroty dla lokalizacji

Podmenu raportowe zawiera podstawowe i najczęściej wykorzystywane raporty dzienne i miesięczne. W zależności od wymagań użytkownika istnieje możliwość prostego dodawania nowych raportów do menu.

8. Podział realizacji funkcjonalności serwera i klienta

Klient

- wyświetlanie żądanych danych z bazy danych
- obsługa edycji, dodawania i usuwania rekordów
- udostępnianie przyjaznego interfejsu użytkownik
- możliwie dokładna wstępna walidacja wprowadzonych danych

Serwer

- końcowa walidacja danych, sprawdzenie integralności
- logowanie zdarzeń i akcji
- obsługa sesji
- pobieranie, edycja oraz usuwanie rekordów
- umożliwienie nawiązania wielu połączeń jednocześnie
- realizacja powyższych za pomocą procedur wbudowanych i triggerów

9. Integralność danych

Integralność danych na etapie dodawania rekordów zapewniona jest poprzez odpowiednio zaplanowane związki między tabelami. Nie jest na przykład możliwa sytuacja, w której dwie faktury przypisane są do jednego zamówienia.

W procesie usuwania, z bazy danych muszą zniknąć wszystkie rekordy powiązane z tym usuwanym. Np. jeśli rozwiązujemy kontakt z danym dostawcą, musimy zadbać także o anulowanie wszystkich zamówień z nim powiązanych.

10. Szkielet aplikacji – podział na moduły

moduly.png

Moduły przedstawione w pliku komunikują się także między sobą. *JDBC* – odpowiedzialne za połączenie modułów z bazą skomunikowane jest z każdym pozostałym blokiem. *Moduł Zamówień* musi mieć dostęp do *Modułu Części*, w celu sporządzenia poprawnego zamówienia, składającego się z istniejących towarów; a także do *Modułu Ewidencyjnego*, aby przypisać do zamówienia odpowiedniego pracownika lub kontrahenta.

11. Opis transakcji dla modułu zamówień

Transakcja oznacza zablokowanie bazy danych przed reagowaniem na kolejne zapytania. Te są buforowane, a następnie wykonane zbiorowo. W przypadku niepowodzenia w wykonywaniu choć jednego z zapytań, przywracany jest stan sprzed rozpoczęcia transakcji.

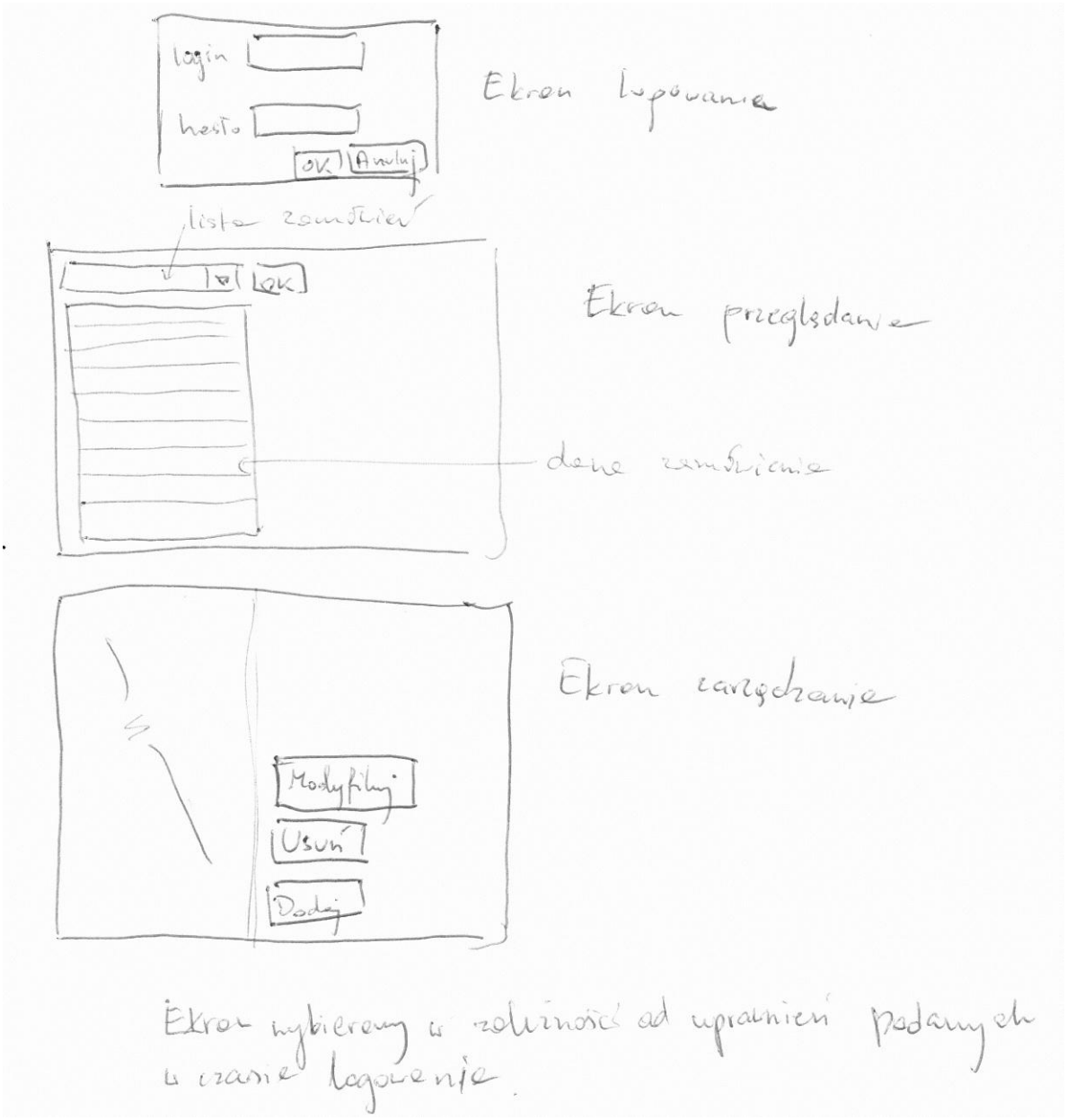
W naszym module konieczna będzie transakcyjność w następujących przypadkach:

- *Złożenie zamówienia* – nie powinno być możliwe złożenie nowego zamówienia, jeśli z jakiegoś powodu nie uda się wygenerować i przypisać do niego nowej faktury lub zebrać wszystkich pozycji.
- *Usunięcie zamówienia* – należy zadbać o to, aby przy usuwaniu zamówienia (oczywiście niezrealizowanego) zniknęły także jego pozycje, a także wygenerowana faktura (pro-forma!). Nie powinno być w systemie faktur, dla których nie istnieją szczegóły zamówienia.

Silnik bazy danych MySQL obsługuje możliwość blokowania pojedynczych wierszy, co zostanie zastosowane w naszym rozwiązaniu, aby nie blokować wszystkich tabel i wierszy.

12. Projekt GUI dla modułu zamówień

(poniżej)


login
hasło
OK
Anuluj
Ekran logowania
lista zamówień
OK
Ekran przeglądania
dane zamówienia
Modyfikuj
Usuń
Dodaj
Ekran zarządzania
Ekran wybierany w zależności od uprawnień podanych w czasie logowania.