
Przykładowy projekt

System zarządzania sprzedażą płyt audio w sklepie muzycznym

autorzy:

Tomasz Maracz

Radosław Kwiatkowski

Zielona Góra 2006

Spis treści

Spis treści.....	2
SPIN	3
Situation.....	3
Problem.....	3
Implied Need.....	4
Implication Question	4
Needs – Pay offs.....	4
USE-CASE diagram.....	5
USE-CASE scenario	6
Lista priorytetów MOSCOW	9
Must have – niezbędne składowe zawierające podstawowe funkcje/usługi.....	9
Should have – elementy, które powinny się znaleźć w systemie, ale nie są jego najistotniejszą składową	9
Could have – te elementy które mogą stanowić rozszerzenie funkcjonalne systemu, rozwiązania "przyszłościowe"	9
Won't have – elementy Systemu Informatycznego które nie będą przez nas realizowane	9
Specyfikacja wymagań pozafunkcjonalnych.....	10
Analiza i Design architektury systemu	11
model SAAM (Software Architecture Analysis Method)	11
Propozycje architektury	11
Ocena przydatności architektury pod względem wydajności i użyteczności	11
Ocena możliwości zastosowania środowisk Open Source do realizacji części programowej/systemowej projektu	12
Projekt interfejsu użytkownika	13
Plan testów.....	18
Zgodność z definicją wymagań.....	18
Testy komponentów	18
Testy integracyjne	18
Testy akceptacji.....	19
Estymacja kosztów SI	20
Koszt sprzętu + koszty licencji programowych	20
Koszt opracowania (projektu) systemu	20
Łączny koszt systemu	21

SPIN

Situation

Niniejszy projekt systemu informatycznego planuje się dla nowopowstałego sklepu muzycznego „Dobra muza”. Obecnie sklep nie dysponuje żadnym systemem informatycznym wspomagającym sprzedaż towaru. Klient samodzielnie lub z pomocą sprzedawcy przeszukuje fizyczne zbiory w celu odnalezienia konkretnej płyty. Paragony wystawiane są poprzez kasę fiskalną, a faktury wypisywane są ręcznie.

W sklepie znajduje się 5 tysięcy płyty CD podzielonych wg kategorii muzycznych:

- rock
- reggae
- metal
- blues
- jazz
- hip-hop
- classic
- inne

Każda kategoria podlega podziałowi alfabetycznemu wg wykonawcy.

Problem

Obecna sytuacja uniemożliwia szybkie wyszukiwanie określonej pozycji, zwłaszcza wtedy, gdy klient przychodząc do sklepu podaje tylko tytuł utworu, przy czym nie zna ani wykonawcy ani nie wie, w której kategorii można znaleźć daną płytę. Głównym problemem jest, więc bardzo długi czas odszukiwania danej płyty, a czasami nawet staje się to w ogóle niemożliwe. Nie ma też pewności, czy przykładowy utwór podany przez klienta znajduje się w aktualnej dyskografii.

Uciążliwe i czasochłonne jest również każdorazowe ręczne wypisywanie faktur, nawet przy ponownym zakupie przez tego samego klienta.

Implied Need

Sprawą priorytetową jest usprawnienie obsługi oraz sprzedaży płyt audio polegające przede wszystkim na:

- szerszych możliwościach wyszukiwania płyt,
- komputerowym fakturowaniu,
- dysponowaniu zawsze aktualną bazą danych.

Ponadto aplikacja powinna umożliwiać również sporządzanie i monitorowanie zamówień od klienta i dla dostawcy, zarządzanie magazynem (m.in. ewidencja ilościowa towaru, stan alarmowy towarów), zarządzanie płatnościami (dokumenty KW, KP, zaliczki, kompensaty, wezwanie do zapłaty, itp.), zarządzanie przelewami (bankowymi, do ZUS i US) oraz generowanie raportów (najlepiej, najgorzej rotujące płyty itp.).

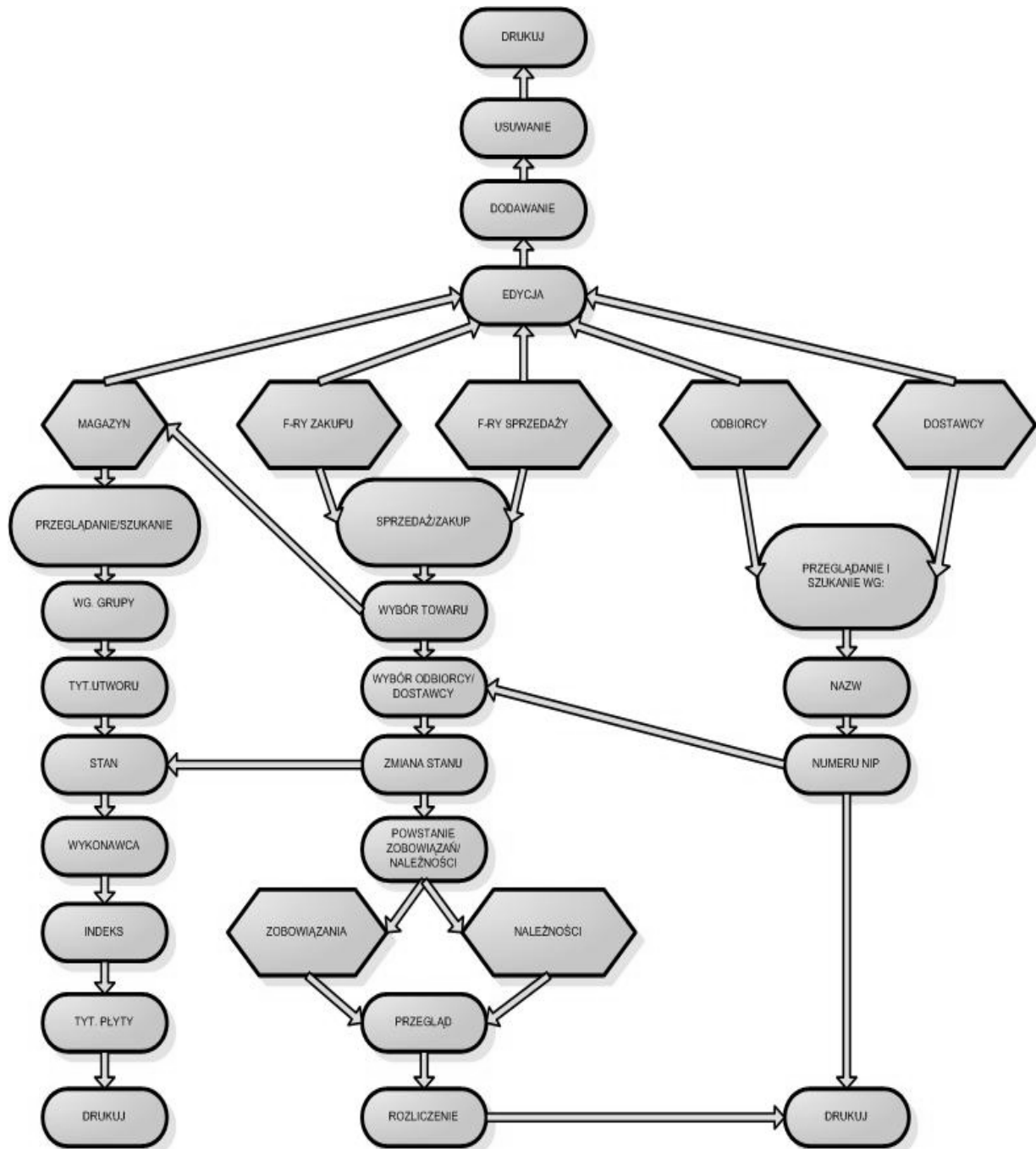
Implication Question

Na rynku jest wiele gotowych systemów wspomagania sprzedaży poczynając od komercyjnych programów, a skończywszy na dużych systemach, z których korzystają wielkie korporacje. Istnieje również szeroki wachlarz w programach open source. Niewiele jest jednak programów przeznaczonych specjalnie dla sklepów muzycznych.

Needs – Pay offs

Przy projektowaniu interfejsu oraz menu systemu SSA warto będzie wzorować się na najlepszych programach tego typu. Bazy danych natomiast zostaną opracowane w oparciu o program MySQL.

USE-CASE diagram



USE-CASE scenario

Nazwa	Przeglądanie, szukanie, drukowanie magazynu		
Opis	Po otwarciu programu i wybraniu z menu opcji <i>magazyn</i> użytkownik może przeglądać i szukać towary wg.: kategorii, tytułu utworu, tytułu płyty, wykonawcy, indeksu, stanu. Istnieje również możliwość wydruku wybranych informacji.		
Warunki wstępne	Poprawne zalogowanie się do systemu		
Wynik działania	Wyświetlenie na ekranie listy towarów, wysłanie do druku.		
Przepływ zdarzeń	Nr	Działania aktora	Odpowiedź systemu
	1	Wyświetl listę towarów wg kategorii	Wyświetlenie listy towarów wg kategorii
	2	Wyświetl listę towarów wg tytułu utworu	Wyświetlenie listy towarów wg tytułu utworu
	3	Wyświetl listę towarów wg tytułu płyty	Wyświetlenie listy towarów wg tytułu płyty
	4	Wyświetl listę towarów wg wykonawcy	Wyświetlenie listy towarów wg wykonawcy
	5	Wyświetl listę towarów wg indeksu	Wyświetlenie listy towarów wg indeksu
	6	Wyświetl listę towarów wg stanu	Wyświetlenie listy towarów wg stanu
	7	Drukowanie wybranych danych	Drukowanie wybranych danych
	8	Szukanie towaru wg kategorii, tyt. płyty, tyt. utworu, wykonawcy, stanu	Wyświetlenie danych spełniających kryteria

Nazwa	Edycja magazynu, kontrahentów, faktur		
Opis	Po otwarciu programu i wybraniu z menu opcji <i>edycja</i> użytkownik może dodawać, usuwać, modyfikować oraz drukować odpowiednie dane.		
Warunki wstępne	Poprawne zalogowanie się do sytemu		
Wynik działania	Usunięcie, dodanie bądź modyfikacja towaru, kontrahenta lub faktury.		
Przepływ zdarzeń	Nr	Działania aktora	Odpowiedź systemu
	1	Edycja towaru	Możliwość zmiany opisu towaru
	2	Dodawanie towaru	Dodanie indeksu do bazy
	3	Usuwanie towaru	Usunięcie towaru możliwe tylko, jeśli stan = zero
	4	Drukowanie listy towarów	Wysłanie do druku
	5	Edycja faktury	Odmowa dostępu
	6	Dodawanie faktury – wystawiania	System pozwala wybrać kontrahenta z bazy lub dodać nowego następnie możliwy jest wybór towarów z listy. Operacja ta może mieć wpływ na powstanie zobowiązania lub należności, oraz zmianę stanów towarowych.
	7	Usuwanie faktury	Możliwość usunięcia, jeśli f- ra została wystawiona w bieżącym miesiącu.
	8	Drukowanie faktury	Wysłanie do druku
	9	Edycja kontrahenta	Możliwość zmiany danych kontrahenta
	10	Dodawanie kontrahenta	Dodanie nowego kontrahenta do bazy
	11	Usuwanie kontrahenta	Usunięcie nowego kontrahenta z bazy jeśli nie występują zobowiązania i należności.
	12	Drukowanie kontrahenta	Wysłanie do druku

Nazwa	Przeglądanie/szukanie kontrahentów		
Opis	Po otwarciu programu i wybraniu z menu opcji <i>dostawcy/odbiorcy</i> użytkownik może przeglądać oraz szukać kontrahentów według nazwy i numeru nip oraz drukować wybrane dane.		
Warunki wstępne	Poprawne zalogowanie do systemu		
Wynik działania	Wyświetlenie wspólnej listy kontrahentów lub osobno dostawców i odbiorców		
Przebieg zdarzeń	Nr	Działania aktora	Odpowiedź systemu
	1	Przeglądanie odbiorców lub dostawców wg nazwy	Wyświetlenie odbiorców lub dostawców wg nazwy
	2	Przeglądanie odbiorców lub dostawców wg numeru nip	Wyświetlenie odbiorców lub dostawców wg numeru nip
	3	Szukanie odbiorców lub dostawców wg nazwy lub numeru nip	Wyświetlenie danych spełniających kryteria
	4	Drukowanie listy kontrahentów	Wysłanie wybranych danych do druku

Nazwa	Przeglądanie, drukowanie i rozliczanie zobowiązań i należności		
Opis	Po otwarciu programu i wybraniu z menu opcji <i>zobowiązania/należności</i> użytkownik może przeglądać, rozliczać, drukować.		
Warunki wstępne	Poprawne zalogowanie do systemu		
Wynik działania	Wyświetlenie listy zobowiązań lub należności		
Przebieg zdarzeń	Nr	Działania aktora	Odpowiedź systemu
	1	Przeglądanie listy zobowiązań lub należności	Wyświetlenie listy zobowiązań lub należności
	2	Wybranie dokumentu do rozliczenia	System pyta o podanie kwoty i po wprowadzeniu i zatwierdzeniu następuje rozliczenie dokumentu i wysyłanie do druku
	3	Drukowanie listy należności lub zobowiązań	Wysłanie wybranych danych do druku

Lista priorytetów MOSCOW

Must have – niezbędne składowe zawierające podstawowe funkcje/usługi

1. Przeglądanie, szukanie, drukowanie magazynu.
2. Edycja magazynu, kontrahentów, faktur.
3. Przeglądanie/szukanie kontrahentów.
4. Przeglądanie, drukowanie i rozliczanie zobowiązań i należności.

Should have – elementy, które powinny się znaleźć w systemie, ale nie są jego najistotniejszą składową

1. Sporządzanie i monitorowanie zamówień od klienta i dla dostawcy.
2. Stan alarmowy towarów.
3. Wezwanie do zapłaty.
4. Zarządzanie przelewami.
5. Generowanie raportów (najlepiej, najgorzej rotujące płyty itp.).

Could have – te elementy które mogą stanowić rozszerzenie funkcjonalne systemu, rozwiązania "przyszłościowe"

1. Współpraca z internetową muzyczną bazą danych.
2. Współpraca z systemem audio – możliwość odsłuchania poszczególnych utworów.
3. Możliwość podłączenia do czytnika kodów.

Won't have – elementy Systemu Informatycznego które nie będą przez nas realizowane

1. Wbudowana funkcja odtwarzania muzyki – są dostępne osobne aplikacje.
2. Połączenie z systemem sprzedaży internetowej – za wysoki koszt realizacji (odmowa klienta).

Specyfikacja wymagań pozafunkcyjnych

1. Funkcja wystawiania paragonów i drukowania poprzez kasę fiskalną.
2. Możliwość obsługi programu wyłącznie za pomocą klawiatury.
3. Wymagania sprzętowe:
 - minimalna pamięć RAM: 128MB,
 - wolny obszar na dysku: do 10MB dla programu + miejsce dla bazy danych zależne od użytkownika.

Analiza i Design architektury systemu model SAAM (Software Architecture Analysis Method)

Propozycje architektury

System będzie pracował na platformie Windows 98/NT/2000/XP. Zostanie on zaimplementowany w postaci bazy danych a w przyszłości dodatkowo z modułem wyszukiwania przez Internet co wymusi na użytkownika połączenie z siecią Internet.

Do przeglądania bazy danych przez Internet system będzie wymagał oprogramowania Internet Explorer 5.0 lub Netscape Communicator 6.0.

Z powyższych wymagań systemowych wynikają wymagania sprzętowe:

- komputer zgodny ze standardem IBM PC (zalecany procesor min. Pentium II)
- pamięć RAM 128 MB
- karta graficzna zgodna ze standardem SVGA 16 MB
- karta muzyczna
- dysk twardy z ok. 1GB wolnej przestrzeni dyskowej (zależne od wielkości bazy danych użytkownika)
- drukarka
- głośniki

Ocena przydatności architektury pod względem wydajności i użyteczności

Powody wyboru takiej architektury:

- możliwość wykorzystania licznych narzędzi CASE dostępnych pod tą platformą,
- zalety interfejsu programowego Windows API, zwłaszcza jego możliwości w programowaniu graficznego interfejsu użytkownika,
- konieczność równoległej pracy systemu z innymi systemami przeznaczonymi dla Windows zwłaszcza z systemem audio.

Ocena możliwości zastosowania środowisk Open Source do realizacji części programowej/systemowej projektu

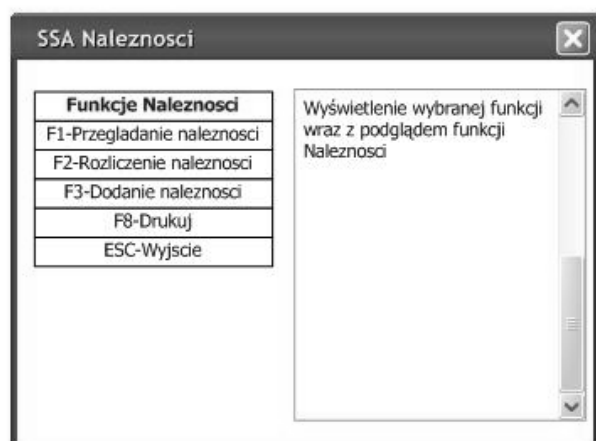
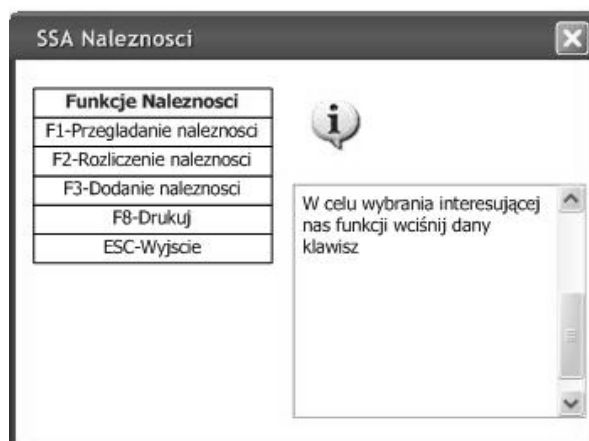
Cały program zostanie zbudowany w oparciu o relacyjne bazy danych, które są dużo wydajniejsze niż pliki tekstowe, a ich obsługa za pomocą języka zapytań SQL jest prosta i przyjemna.

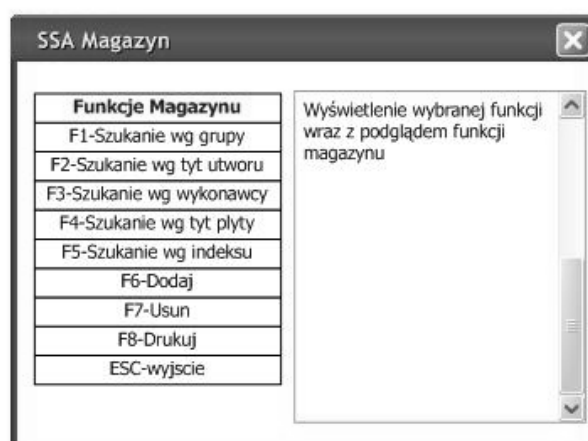
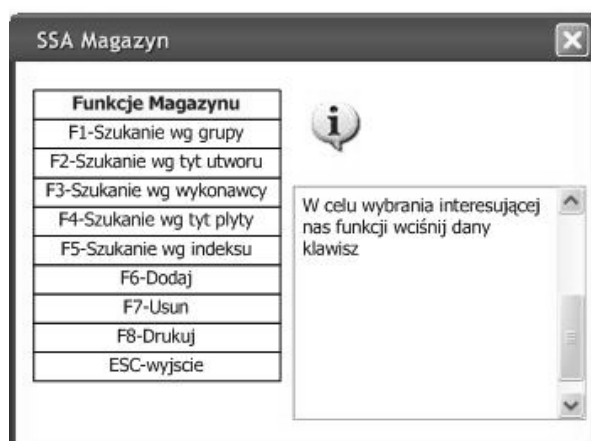
MySQL jest szybkim, wielowątkowym serwerem baz danych obsługującym język zapytań SQL i dostępnym wg Powszechnej Licencji Publicznej GNU (*GNU General Public License*, GPL).

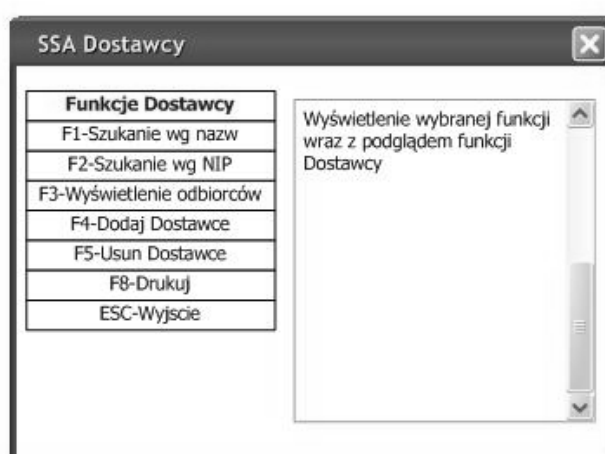
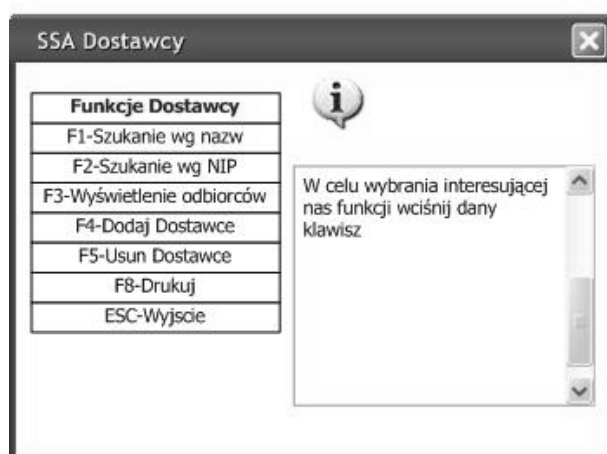
Pracuje z wieloma użytkownikami i doskonale nadają się do wykorzystania razem z PHP jako darmowa platforma aplikacji internetowych, co może być w przyszłości potrzebne przy współpracy z internetową bazą danych.

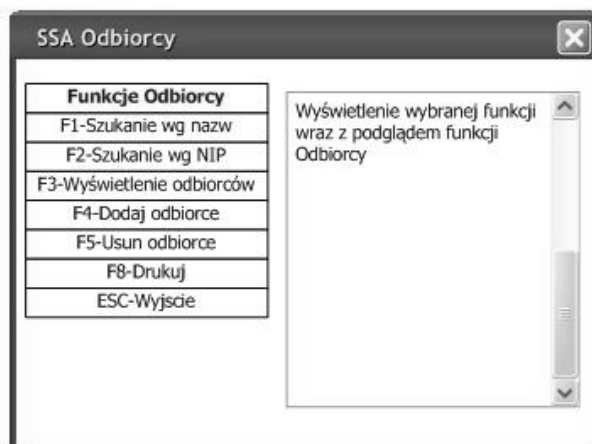
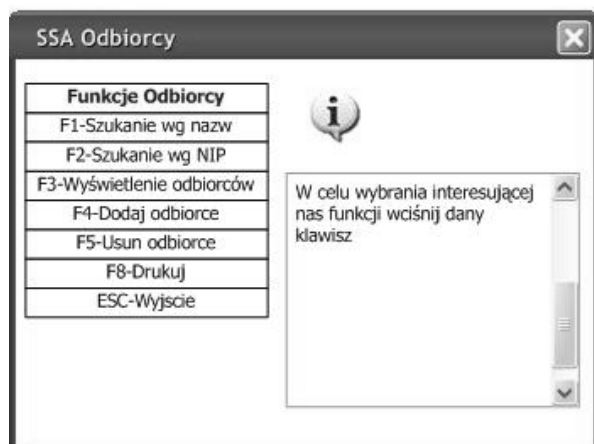
W MySQL-u można między innymi tworzyć nowe bazy danych, a w nich tabele, dodawać nowe rekordy, edytować je lub usuwać. Istnieje możliwość zarządzania użytkownikami, w tym również dokładnego określania praw dostępu do określonego pola w bazie danych. W danym momencie z bazy danych może korzystać nieograniczona liczba osób; zależy to jedynie od zasobów fizycznych serwera.

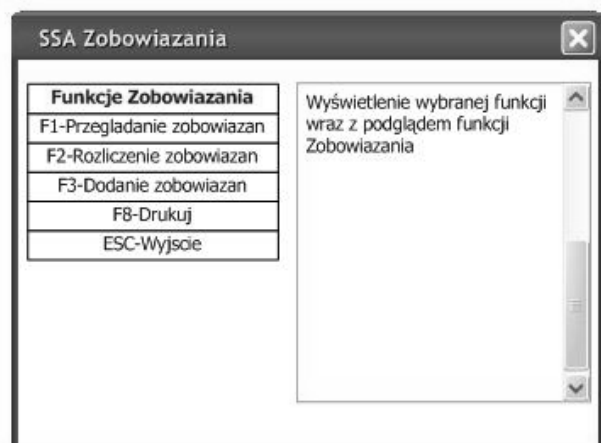
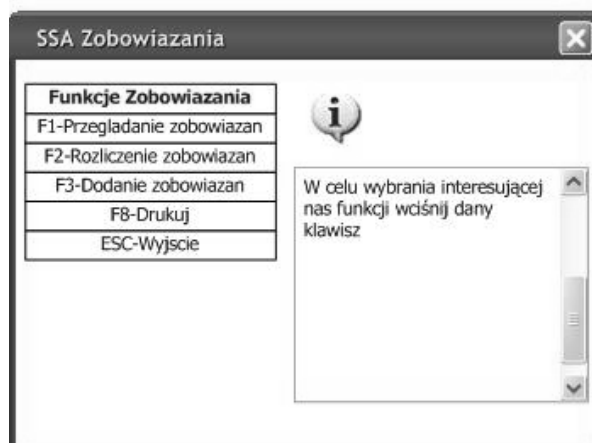
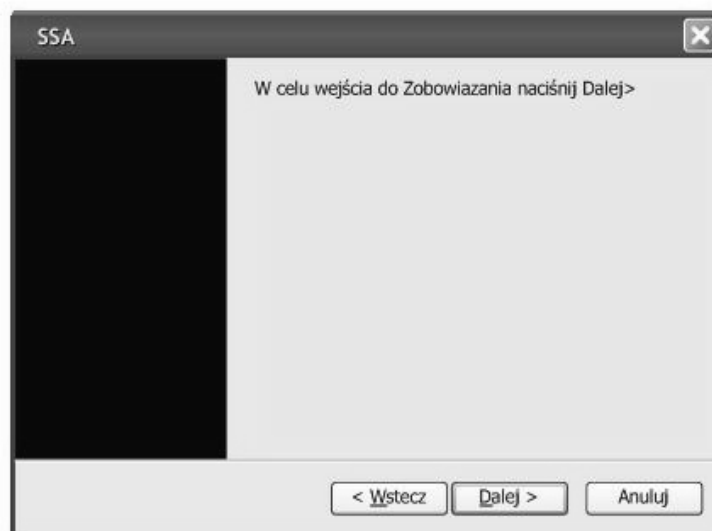
Projekt interfejsu użytkownika











Plan testów

Zgodność z definicją wymagań

Sprawdzenie całego programu pod kątem zawartych w nim funkcji a wymaganiami klienta postawionymi w fazie określania wymagań. Ustalenie zgodności z listą priorytetów Moscow.

Testy komponentów

Na tym etapie będą sprawdzane wszystkie moduły programowe:

- magazyn,
- faktury,
- dostawcy,
- odbiorcy,
- zobowiązania,
- należności.

Dane testowe będą podzielona równomiernie na:

- nietypowe poprawne,
- typowe poprawne i błędne
- nietypowe błędne

Wszystkie testy będą wykonywane już w fazie implementacji bezpośrednio po zakończeniu realizacji poszczególnych modułów przez programistów.

Testy integracyjne

W tej fazie będą integrowane poszczególne moduły. Program będzie sprawdzany pod kątem współpracy z podsystemami oraz równoległej pracy z innymi systemami przeznaczonymi dla Windows zwłaszcza z systemem audio.

Pod uwagę będzie brany każdy moduł osobno oraz wzajemne oddziaływanie poszczególnych modułów na inne.

W trakcie testowania przyjętych będzie kilka strategii tj. m.in.:

- wątkowa THREAD – np. działanie kilku procesów jednocześnie,
- obciążeniowa STRESS – np. zachowanie bazy danych pod dużym obciążeniem,
- wydajnościowa PERFORMANCE – np. sortowanie wielu danych

Sprawdzeniu będą podlegać również wymogi programowe i sprzętowe zaproponowane przy okazji analizy architektury systemu (model SAAM).

Testy te przeprowadzą specjaliści po stronie wykonawcy.

Czas trwania: ok. 3 tygodnie

Testy akceptacji

Gotowy system będzie przekazany do przetestowania przyszłemu użytkownikowi. Będzie to tak zwany test alfa.

Estymacja kosztów SI

Koszt sprzętu + koszty licencji programowych

Zakup 1 komputera klasy PC wraz z:

- monitorem
- drukarką
- głośnikami

4.000 zł.

Z uwagi na zastosowanie serwera MySql dostępnego wg Powszechnej Licencji Publicznej GNU (*GNU General Public License*, GPL) dodatkowe koszty licencji programowych nie występują.

Koszt opracowania (projektu) systemu

$Koszt = KosztPodstawowy \times niezawodność \times czas \times zasoby \times narzędzia \times ekspertyza \times pensja$;
gdzie

$KosztPodstawowy = ZłożonośćSystemu \times KDSI \times Multiplikator$
w osobomiesiącach

(ZłożonośćSystemu- od 2.4 dla s.jednowarstwowych do 3.6 dla wielowarstwowych, KDSI- Kilo Delivered Source Instructions -tysiące dostarczonych linii kodu, Multiplikator- złożoność danych od 0.7 dla alfanumerycznych do 1.7 dla obiektowych -medialnych)

niezawodność -współczynnik niezawodności i bezpieczeństwa systemu od 0.5 dla nieistotnego poziomu do 2.0 dla w pełni zabezpieczonego SI

czas -wydłużenie lub skrócenie czasu realizacji od 0.5 dla długiego cyklu do 2.0 dla wykonania przed terminem

zasoby- wielkość zasobów danych/sieci od 0.5 dla MB do 2.0 dla TB itd...

narzędzia- współczynnik dostępności narzędzi, od 0.5 dla w pełni dostępnych CASE do 2.0 dla unikatowych rozwiązań bez wspomagania

ekspertyza- współczynnik zależny od zewnętrznych audytów i ekspertyzy, od 0.7 dla braku eksp. do 2.0 dla wymagających certyfikatów

pensja- pensja instytucjonalna programisty, czyli koszty firmy zatrudniającej programistę (średnia unijna 10000 Euro miesięcznie)

koszt podstawowy = $2.6 \times 5 \times 1 = 13$

koszt = $13 \times 1 \times 1 \times 1 \times 1 \times 10\,000 = 13\,000$ zł.

Łączny koszt systemu

Koszt opracowania projektu + koszt sprzętu = $13\,000 + 4\,000 = 17\,000$ zł.