

RAPORT KOŃCOWY PROJEKTU

Temat: „Wieloplatformowy program do obsługi faktur”



Adresat: dr inż. Jacek Kołodziej

**Wykonawcy: Daniel Krysiak
Przemysław Szpunar
Grzegorz Śmierzchalski**

Spis Treści

1. Charakterystyka projektu.....	2
2. Przebieg realizacji projektu.....	3
3. Metodyki i narzędzia.....	4
4. Testowanie.....	5

Charakterystyka projektu

Celem projektu było zrealizowanie prostego programu do wystawiania faktur, obsługiwanego przez jednego użytkownika na lokalnej bazie danych (SQLite 3). Międzyplatformowego i „Wolnego”, dlatego użyto przenośnych bibliotek i „Wolnych” narzędzi: SQLite3, QT4.5 SDK, G++. Program był testowany w środowiskach Linux 32 i 64 bit, oraz Windows 32 bit, powinien jednak działać także na wszystkich innych platformach Unix, MAC, oraz prawdopodobnie ARM (w tym Windows CE, Android, etc.)

Program uwzględnia cykl życia faktury, tj.: WZ, Faktura, Korekta. Pilnuje spójności danych na wystawionych fakturach.

Funkcjonalności zaimplementowane do tej pory: dodawania/aktualizacja/usuwanie klientów i produktów.

Przebieg realizacji projektu

W trakcie realizacji nie doszło do znacznych odchyśleń względem wstępnie przyjętego harmonogramu, głównie dzięki uniezależnieniu modułów programu.

Pojawiły się problemy z kompatybilnością kodowania znaków – zapomniano tego ustalić w fazie projektowania, oraz nie przewidziano problemów z UTF-8 w QtCreatorze. By temu zapobiec, trzeba było zweryfikować kompatybilność środowisk w trakcie projektowania.

Pojawiły się opóźnienia ze względu na archaiczny interfejs SQLite3 dla C++. By temu zapobiec, trzeba było w fazie analizy przyjrzeć się temu interfejsowi i ew. wybrać jakieś inne API (nakładkę) na SQLite3, najlepiej zrealizowaną obiektowo, dedykowaną dla C++, a nie czystego C.

Metodyki i narzędzia

Do realizacji projektu wykorzystano platformę Google Code (faktury.googlecode.com), wraz z repozytorium SVN, oraz bugtrackiem. Ponadto zespół wykorzystywał forum dyskusyjne PhpBB 3 na serwerach Rootnode (dext.rootnode.net/faktury), wspólną skrzynkę pocztową, gdzie wysyłane były automatyczne informacje o zachodzących w projekcie zmianach, oraz komunikator Jabber, dzięki któremu możliwe było rozwiązywanie problemów „na gorąco”.

Projekt był realizowany w duchu FLOSS, dlatego został wydany na licencji GPLv3.

Program został zrealizowany zgodnie z wzorcem MVC. Ze względu na to, iż zespół projektowy używał różnych środowisk i narzędzi (QTCreator, NetBeans, Linux, Windows), oraz na wyodrębnienie obszarów odpowiedzialności, projekt został podzielony na 3 niemal całkowicie niezależne moduły: Modelu (Daniel Krysiak), Widoku (Grzegorz Śmierzchalski), oraz Kontrolerów (Przemysław Szpunar). Widoki można oglądać niezależnie od reszty projektu wszędzie tam gdzie jest dostępny QTDesigner, Model można skompilować i przetestować samodzielnie. Dopiero warstwa Kontrolerów, która łączy moduły w całość wymaga obecności całego kodu.

Kod został udokumentowany w plikach źródłowych, oraz w dużej części pokryty testami automatycznymi.

Testowanie

Program przechodzi pomyślnie ręczne testy zaimplementowanej funkcjonalności.

Kod modelu został w 100% pokryty testami automatycznymi. Kod Kontrolerów został pobieżnie pokryty testami. QT jest zaprojektowane tak że testy automatyczne są bardzo ciężkie do wykonania albo i niemożliwe (Przy tej ilości czasu). Przetestowana funkcjonalność to dodawanie klientów – Wykorzystywane jest do tego też wyszukiwanie i usuwanie ale nie są one pokryte testami bezpośrednio.