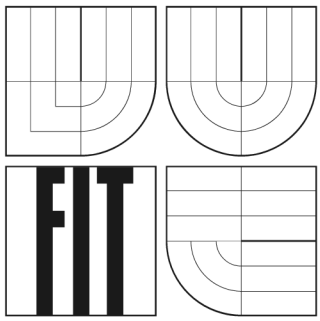
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Semestrálny projekt

Erik Ferencz

2007

Zadání:

1. Cílem práce je navrhnout informační systém organizace s vyšší přidanou hodnotou zejména pro oblast řízení skladu, marketingu a obchodu.
2. Seznamte se s C#, PostgreSQL, případně dalšími nástroji a technologiemi vhodnými pro informační systémy (dolování dat, problematika uživatelských rozhraní atd.)
3. Vytvořte detailní návrh systému.

Prehlásenie :

Prehlasujem, že som tento semestrálny projekt vypracoval samostatne, pod vedením Mgr. Zdeňka Martínka. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

Abstrakt:

Tento semestrálny projekt sa zaoberá analýzou , návrhom informačného systému na správu a vedenie obchodnej firmy .Systém je navrhnutý tak, že ho tvorí neobmedzené množstvo modulov ktoré spolu spolupracujú alebo na seba nadväzujú. Každý modul má vlastné tabuľky v databázy ,vlastné triedy ktoré tvoria strednú vrstvu aplikácie a vlastné grafické prostredie, avšak jednotlivé moduly sami o sebe nie sú schopné fungovať. Komunikácia s ostatnými modulmi systému bude prebiehať pomocou databázových serverov a SRBD (Systém Riadenia Báze Dát).Súčasťou bude i vybudovaná databáza, ktorá má štruktúru odpovedajúcu celému systému, ale je pripravená na doplnenie informácií od nových modulov systému. Na splnenie danej úlohy som sa musel oboznámiť s problematikou programovacích jazykov C# a so systémom riadenia báze dát PostgreSQL.

Kľúčové slová:

Structured Query Language (SQL), World Wide Web (WWW), webová aplikácia, window aplikácia , databáza, tabuľka, obchodné procesy

Abstrakt:

This term project is about analyses and design of information system for administration and managing business firm.System is designed as module system with unlimithed count of modules that coact or are connected with another modules. Each module has its own data tables in database, own Classes which make the middle layer of aplication and graphical interface, but modules are not independent (one module can not work as a system).Iner communication among modules is based on database servers.Part of the application is its own database. To accomplish this project I had to familiarize myself with a problemathic of programming in programming language C# and with database language PostgreSQL

Key words:

Structured Query Language (SQL), World Wide Web (WWW), web application, window form , database, table, business process

Obash

1 Úvod

S rozvojom a rozmachom internetu a webových aplikácií sa zvyšujú nároky na vedenie obchodnej politiky, rýchlosť, flexibilita predaja a hlavne oslovenie a informovanosť zákazníka, spojená s čo najnižšími nákladmi, komfortom pri nakupovaní a v neposlednom rade spokojnosťou nakupujúceho. Všetky tieto požiadavky však zväčšujú nároky na vnútorné vedenie firmy a následnú prehľadnosť v jednotlivých nákupných a predajných transakciách.

Postupne s rastom firmy a pokrokom ekonomických aplikácií sa mení aj politika informovanosti zákazníkov o danej firme a jej produktoch a poskytovaných službách. Prvotne sa ekonomické aplikácie objavovali len v podobe účtovníctva, neskôr boli vytvorené aplikácie pre vedenie skladov a vytváranie inventúr, avšak neexistovali technológie ktoré by umožnili spojenie všetkých obchodných procesov spojených s nákupom a predajom, ktoré by boli spojené do jednej aplikácie. V posledných piatich rokoch sa objavujú aplikácie ktoré umožňujú spojenie jednotlivých procesov, avšak ich cena a náročnosť na koncového užívateľa spôsobujú, že sa dané aplikácie špecializujú len na stredné a väčšie firmy. Takýto informačný systém je však dnes neodmysliteľnou súčasťou každej dynamicky sa rozvíjajúcej firmy ktorá sa snaží presadiť vo veľkej konkurencii.

Takýto informačný systém riadi chod celej firmy s administratívneho hľadiska a je dôležitý napríklad pre získanie certifikátov ISO.

Cieľom mojej semestrálnej práce je navrhnúť informačný systém firmy ktorý bude mať možnosť prepojenia jednotlivých obchodných procesov firmy a splnil požiadavky na zjednodušenie vedenia firmy a komunikácie so zákazníkom.

Aplikácia musí vedieť zabezpečiť dáta spojené s jednotlivými partnermi, naviazanie jednotlivých obchodných procesov a prehľadnosť dát spojených so skladom, predajom a nákupom tak aby zjednodušoval a urýchľoval prácu hlavne pracovníkom firmy. Zákazník by mal mať čo najmenej starostí s objednávaním a platbou vybraného tovaru. Zákazník bude informovaný o jednotlivých tovaroch hlavne pomocou špeciálnych cenníkov a ponúk. Aplikácia bude formovaná tak, aby bolo jednoduché dohľadať históriu platobných, skladových a iných obchodných dokladov jednotlivých zákazníkov z akéhokoľvek modulu aplikácie.

2 Analýza

Analýza je najdôležitejšou časťou pri tvorbe našej aplikácie.

V prípade nášho informačného systému existuje viacero skupín užívateľov.

Užívateľom sú prideľované práva na prístup a práva na vykonávanie jednotlivých funkcií.

Požiadavky koncových užívateľov:

1. Bude sa jednať o sieťovú, multiužívateľskú aplikáciu s paralelným prístupom k jednotlivým položkám dokladov.

2. Užívateľom budú prideľované jednotlivé práva na prehliadanie alebo editovanie jednotlivých položiek dokladov a práva na funkcie (napr. potvrdzovanie dokladov)

3. Transakcie užívateľov by mali byť bezpečné. Aby sa zabránilo zneužitiu konta daného užívateľa, bude do aplikácie zabudovaný základný mechanizmus, ktorý overí totožnosť užívateľa a overí jeho oprávnenia pri každom vstupe do modulu a pri snahe meniť nejaké položky modulu.

4. Jednotlivé moduly by mali byť medzi sebou prepojené. Napr. modul zákazníkov by mal mať prepojenie na modul faktúr, výdajok, dodacích listov atď.

5. Informačný systém musí mať jednoduché a intuitívne ovládanie .

6. Aplikácia musí umožňovať rýchle a jednoduché dohľadanie potrebných dát, filtráciu dát a ich celkové zobrazenie .

7. Manipulácia s dátami (editovanie, vytváranie nových dát) musí prebiehať v rámci transakcie a program nesmie umožňovať súčasné menenie jednej položky viacerými užívateľmi.

8. Užívateľ bude mať možnosť vyexportovať jednotlivé dáta pomocou tlačových zostáv, exportu do Excelu a do formátu XML

9. Aplikácia musí umožňovať riadenie viacerých skladov.

10. Na jednotlivých skladoch je potrebné kontrolovať umiestnenie tovaru, jednotlivé šarže tovaru a sériové čísla tovaru. Hlavnou funkciou skladov je sledovanie dispozície jednotlivých tovarov na sklade a sledovanie celkovej účtovej hodnoty skladu.

11. Na každom doklade musí byť uvedený užívateľ ktorý daný doklad vytvoril, kedy ho vytvoril, užívateľ ktorý daný doklad menil a kedy ho menil .

2.1 Jednotlivé moduly systému

2.1.1 Hlavné moduly

- Modul Užívatelia
- Modul Tovar
- Modul Cenník
- Modul Partneri
- Modul Kontaktné Osoby
- Modul Ponuky
- Modul Kúpne zmluvy
- Modul Objednávky Vydané
- Modul Faktúry Prijaté
- Modul Dodacie Listy Prijaté
- Modul Príjemky
- Modul Zákazky
- Modul Objednávky Prijaté
- Modul Faktúry Vydané
- Modul Dodacie Listy Vydané
- Modul Rezervačné Listy
- Modul Výdajky
- Modul Prevodky

- Modul Zapožičiavanie
- Firma

2.1.2 Podriadené moduly

- Mesto
- Pošta
- Štát
- Kraj
- Okres
- Oblasť
- Región
- Adresa
- Množstevné jednotky
- DPH
- Kurz
- Titul
- Značka
- Mena
- Umiestnenie
- Šarže
- Sériové čísla
- Spôsob dopravy
- Spôsob dodania
- Spôsob platby
- Forma objednávky
- Spôsob odberu
- Stredisko
- Obdobie
- Rada Nákupu
- Rada Predaja

- Funkcie
- Skupiny
- SkupinaT
- TypTovaru
- Poznámky

2.2 2.2 Analýza modulov systému

2.2.1 Požiadavky na Podriadené moduly

Moduly musia byť vytvorené tak, aby v nich bolo možné čo najjednoduchším spôsobom vyhľadať, vytvárať a editovať jednotlivé položky.

Dáta modulu by sa mali skladať z identifikačného čísla jednotlivej položky modulu, skratky a stručného popisu položky. V moduloch ktoré potrebujú viacej položiek, je dovolené rozšíriť ich o dané dáta oko napríklad kurz, či množstvo jednotiek pri mene. Jednotlivé moduly by malo byť možné spolu previazať. Jedná sa napríklad o moduly Kraj – Okres – Mesto, Mena – Kurz atď. Tieto moduly by mali slúžiť ako podriadené moduly pre hlavné moduly(Napr. Adresa v Partneroch).

2.2.2 Požiadavky na Modul Užívatelia

Modul musí byť vytvorený tak, aby v ňom bolo možné čo najjednoduchším spôsobom vyhľadať, vytvárať a editovať jednotlivé položky užívateľov a jednoducho naviazať podriadené dátové moduly ako sú Titul, Adresa, Funkcie, Poznámky .

Dáta modulu by sa mali skladať z identifikačného čísla jednotlivej položky modulu, prihlasovacieho mena, hesla , mena osoby, priezviska osoby, telefónu, emailovej adresy, príznaku o aktívnosti užívateľa, dátum poslednej zmeny, kto menil položky a stručného popisu položky. Prehľadné naviazanie jednotlivých práv k užívateľovi.

2.2.3 Požiadavky na Modul Tovar + Cenník

Moduly musí byť vytvorený tak, aby v nich bolo možné čo najjednoduchším spôsobom vyhľadať, vytvárať a editovať jednotlivé položky skladových kariet tovaru a jednoducho naviazať podriadené dátové moduly ako sú Značka, Množstevná jednotka, Mena, Kurz, Typ Tovaru, Štát pôvodu .

Dáta modulu by sa mali skladať z identifikačného čísla jednotlivej položky modulu, artiklu(číslo skladovej karty), názov tovaru , skladovú jednotku, alternatívnu jednotku, množstvo alt. Jednotky vzhľadom ku skladovej jednotke, skladovú cenu, predajnú cenu, menu, kurz, štát pôvodu, príznak sledovania umiestnenia, príznak sledovania šarží , príznak

sledovania sériových čísel, minimálne množstvo na sklade, maximálne množstvo na sklade, údaje o poslednom nákupe a predaji, DPH, dátum vytvorenia, dátum poslednej zmeny, kto menil položky a stručného popisu položky. Zaradenie tovaru do viac ako jednej skupiny tovaru (Napríklad by mal modul umožňovať vytvorenie stromu skupín tovaru – Komodita – Kategória - Druh)

Zaradenie tovaru len do Cenníka, Len medzi skladové karty, poprípade do oboch.

2.2.4 Požiadavky na Modul Partneri

Modul musí byť vytvorený tak, aby v ňom bolo možné čo najjednoduchším spôsobom vyhľadať, vytvárať a editovať jednotlivé položky partnerov (dodávatelia, odberatelia) a jednoducho naviazať podriadené dátové moduly ako sú Adresa, Región, Oblasť, Banka, Spôsob Platby, Spôsob Dopravy .

Dáta modulu by sa mali skladať z identifikačného čísla jednotlivej položky modulu, skratky, skratky2, ICO, DIC, email, URL, adresa , región, oblasť, telefón, fax, banka, číslo bankového účtu, splatnosť faktúry, penále, spôsob platby, spôsob dopravy, dátum poslednej zmeny, kto menil položky a stručného popisu položky. Priradenie partnerov do jednotlivých skupín ako dodávatelia, odberatelia... atď.

2.2.5 Požiadavky na Modul Kontaktné Osoby

Modul musí byť vytvorený tak, aby v ňom bolo možné čo najjednoduchším spôsobom vyhľadať, vytvárať a editovať jednotlivé položky Kontaktných Osôb k Partnerom a jednoducho naviazať podriadené dátové moduly ako sú Titul, Funkcia, Partner.

Dáta modulu by sa mali skladať z identifikačného čísla jednotlivej položky modulu, mena osoby, priezviska osoby, telefónu domov a zamestnanie, emailovej adresy domov a zamestnanie, dátum poslednej zmeny, kto menil položky a stručného popisu položky. Priradenie partnerov do jednotlivých funkcií ako riaditeľ, sekretárka... atď.

2.2.6 Požiadavky na Modul Ponuky

Modul musí byť vytvorený tak, aby v ňom bolo možné čo najjednoduchším spôsobom vyhľadať, vytvárať a editovať jednotlivé položky Ponúk jednotlivým Partnerom a jednoducho naviazať podriadené dátové moduly ako sú Mena, Kurz, Partner.

Dáta modulu by sa mali skladať z identifikačného čísla jednotlivej položky modulu, užívateľa ktorý danú ponuku vytvoril, ceny ponuky, ceny s DPH, DPH , meny , osoby pre ktorú je ponuka vystavená, oslovenia osoby, funkcie, oslovného textu, dátum poslednej zmeny, kto menil položky a stručného popisu položky. Ponuka by sa mala skladať minimálne z jednej grupy ktorá obsahuje podriadené ponúkané položky tovaru ktoré sa prevezmú z dátového

modulu Cenník. Jednotlivé položky grúp a podriadené tovary je možné mazať, editovať a vytvárať nové.

2.2.7 Požiadavky na Modul Kúpne zmluvy

Modul musí byť vytvorený tak, aby v ňom bolo možné čo najjednoduchším spôsobom vyhľadať, vytvárať a editovať jednotlivé položky Kúpnych Zmlúv jednotlivým Partnerom a jednoducho naviazať podriadené dátové moduly ako sú Mena, Kurz, Partner , Ponuky.

Dáta modulu by sa mali skladať z identifikačného čísla jednotlivej položky modulu, užívateľa ktorý danú kúpnu zmluvu vytvoril, ceny , ceny s DPH, DPH , meny , osoby pre ktorú je kúpna zmluva, vystavená, oslovenia osoby, funkcie, oslovného textu, dátum poslednej zmeny, kto menil položky a stručného popisu položky. Kúpna zmluva by sa mala skladať minimálne z jednej grupy ktorá obsahuje podriadené ponúkané položky tovaru ktoré sa prevezmú z dátového modulu Cenník. Jednotlivé položky grúp a podriadené tovary je možné mazať, editovať a vytvárať nové. Jednotlivé grupy alebo podriadené tovary je možné preberať už z vytvorených Ponúk.

2.2.8 Požiadavky na Modul Objednávky Vydané

Modul musí byť vytvorený tak, aby v ňom bolo možné čo najjednoduchším spôsobom vyhľadať, vytvárať a editovať jednotlivé položky Objednávok jednotlivým Partnerom a jednoducho naviazať podriadené dátové moduly ako sú Mena, Kurz, Partner , Spôsob Dopravy, Spôsob Odberu , Spôsob Platby, Stredisko, Obdobie, Rada Nákupu.

Modul je nosným modulom pre obchodný proces nákupu, čo znamená , že by mal na seba viazať ostatné doklady nákupu ako je Faktúra prijatá, Dodací list prijatý a Príjemka.

Dáta modulu by sa mali skladať z identifikačného čísla jednotlivej položky modulu pre danú radu nákupu v danom období, užívateľa ktorý danú objednávku vytvoril, ceny, ceny s DPH, DPH , meny ,kurzu, spôsobu dopravy, spôsobu platby, spôsobu odberu, partnera pre ktorého je objednávka vystavená, dátum vystavenia, dátum dodania, dátum poslednej zmeny, kto menil položky, kto vystavil položku, stredisko a stručného popisu položky. Objednávka by sa mala skladať minimálne z jednej podriadenej položky tovaru ktoré sa prevezmú z dátového modulu Tovaru. Jednotlivé položky podriadeného tovaru je možné mazať, editovať a vytvárať nové.

Potvrdiť objednávku je možné až po tom, ako budú potvrdené všetky podriadené doklady nákupu.

2.2.9 Požiadavky na Modul Faktúry prijaté

Modul musí byť vytvorený tak, aby v ňom bolo možné čo najjednoduchším spôsobom vyhľadať, vytvárať a editovať jednotlivé položky Ponúk jednotlivým Partnerom a jednoducho naviazať podriadené dátové moduly ako sú Mena, Kurz, Partner , Spôsob Dopravy, Spôsob Odberu , Spôsob Platby, Stredisko, Obdobie, Rada Nákupu.

Dáta modulu by sa mali skladať z identifikačného čísla jednotlivej položky modulu pre danú radu nákupu v danom období, užívateľa ktorý danú objednávku vytvoril, ceny, ceny s DPH, DPH , meny ,kurzu, konštantný symbol, variabilný symbol, párovací symbol, spôsobu dopravy, spôsobu platby, forma objednávky, dátum vystavenia, dátum dodania, dátum poslednej zmeny, kto menil položky, kto vystavil položku, stredisko a stručného popisu položky. Faktúra by sa mala skladať minimálne z jednej podriadenej položky tovaru ktoré sa prevezmú z dátového modulu Tovaru alebo z nadriadenej Objednávky (podľa toho či je daný doklad viazaný alebo nie). Jednotlivé položky podriadeného tovaru je možné mazať, editovať a vytvárať nové.

Po potvrdení faktúry už nie je možné meniť nákupnú cenu.

2.2.10 Požiadavky na Modul Dodacie Listy prijaté a Príjemky

Rovnako ako Faktúry prijaté

Po potvrdení príjemky sa na skladní podriadený tovar na sklad v danom množstve ako je uvedené na príjemke.

2.2.11 Požiadavky na Moduly Predaja

Rovnaké požiadavky ako na nákup.

Po potvrdení výdajky sa vyskladni tovar v počte uvedenom na podriadenom tovare.

3 Výber vhodného softwaru

Na strane užívateľa musí byť internetový obchod založený na použití webového prehliadača alebo Window Form aplikácie. Pre uchovanie užívateľských transakcií tzn.

Užívateľom objednaných produktov, zoznamu ponúkaného tovaru a údajov zadaných užívateľom, bude aplikácia využívať databázu.

3.1 Databáza

K dispozícii sú dva typy uchovania dát:

1. Bežné dátové súbory (napríklad budú dáta uložené v textových súboroch)
2. Relačná databáza ako je napríklad Oracle, Sybase, MS SQL,PostgreSQL ,MySQL atd.

Bežné dátové súbory je možné vylúčiť, pretože na zostrojenie základných funkcií na zápis, úpravu a čítanie dát by sme museli vynaložiť veľké úsilie. Takéto funkcie sú v relačnej databáze k dispozícii pred implementované.

Ako koncovú databázu sme teda vybrali PostgreSQL, a to z týchto dôvodov:

1. Databáza PostgreSQL je ralačná databáza „open source“, a to je výhoda oproti komerčným relačným databázam.
2. Databáza PostgreSQL je ľahko dostupná a jej spravovanie nie je náročné – môže ju obsluhovať i človek ktorý nie je školeným správcom.
3. Databáza PostgreSQL podporuje klientske rozhranie pre programovanie API (Application on Programming Interface) pre široký okruh programovacích jazykov ako sú napr. Perl, C,C++,C#, PHP atd. Klientske programy, ktoré sú v týchto jazykoch napísané a ktoré pristupujú k dátam z databáze PostgreSQL môžu tieto rozhrania API používať. V týchto programovacích jazykoch je množstvo možností implementácie strednej vrstvy.

4 Prostriedky použité pre tvorbu aplikácie

4.1 C#

C# (v angličtine *si-sharp*) je objektovo-orientovaný programovací jazyk vyvinutý spoločnosťou Microsoft ako časť ich iniciatívy .NET. Microsoft si za základ pre nový jazyk

C# zobral C++ a jazyk Java. C# bolo navrhované s úmyslom vyvážiť silu jazyka C++ a tú spojiť s možnosťou rýchleho programovania "rapid application development", ktoré ponúkali jazyky ako napríklad Visual Basic, Delphi

4.2 PostgreSQL

PostgreSQL je voľne šíriteľný objektovo-relačný databázový systém (systém riadenia báz dát), uvoľnený pod flexibilnou licenciou BSD. Ponúka alternatívu k ostatným voľne šíriteľným databázovým systémom (ako sú MySQL, Firebird, MaxDB a iné), ako aj k prioprietárnym (akými sú napr. Oracle, DB2 od IBM či Microsoft SQL Server). Podľa mnohých databázových odborníkov je v súčasnosti PostgreSQL najvyspelejší a najsofistikovanejší voľne šíriteľný systém riadenia báz dát.

5 Návrh

5.1 Definícia základných pojmov

5.1.1 Zmena, Počiatočné stavy a Holder

Základnou funkciou jednotlivých modulov je zobrazovanie, editovanie a pridávanie položiek. Keďže naša aplikácia bude sieťová, musíme zabezpečiť, aby mohli užívatelia súčasne čítať jednotlivé položky modulu, súčasne vytvárať nové položky modulu, ale zmena už existujúcich položiek musí byť výlučný.
Pomocou jednoduchých SQL príkazov sme schopný zabezpečiť súčasné čítanie dát z databáze avšak súčasné vytváranie nových položiek je ťažšie z dôvodu existencie jedinečných identifikátorov položiek v moduloch. Preto musí existovať modul Počiatočné stavy, ktorý bude prideľovať jednotlivým transakciám, ktoré obsahujú vkladanie nového prvku do modulu, nové identifikačné číslo tak, aby dve súčasne spustené transakcie obsahovali rozdielny identifikátor.
Vyriešenie exkluzívneho menenia existujúcej položky je ešte zložitejšie. Aby sme daný problém mohli vyriešiť, bude v každom module existovať funkcia Zmena ktorá sa pozrie do tabuľky Holder kde podľa čísla modulu a identifikačného čísla položky modulu zistí, či už niekto danú položku mení. Ak áno, vypíše meno užívateľa, ak nie, vloží do tabuľky Holder údaje o menenej položke a dovolí užívateľovi meniť dát položky.

5.1.2 Filter a Zložky

V prípade, že moduly budú obsahovať veľké množstvo dát (napr. nad 1000 položiek), môže sa zmenšiť prehľadnosť dát, a tým sa sťaží obsluha programu.

Preto bude aplikácia obsahovať možnosť jednoduchého filtrovania dát modulu a to pomocou jednotlivých polí položiek modulu. Napr. pomocou identifikačného čísla položky, skratky položky, názvu ..... atď. Modul musí byť ľahko prevedený z normálneho zobrazenia položiek do filtrovania a naspäť a vyfiltrované položky musia byť natrvalo uložené v pamäti až do ďalšej zmeny podmienky filtra.
Druhým spôsobom zlepšenia orientácie dát je vytváranie zložiek. Zložky budú dvojakého druhu a to
1. Programovo vytvárané zložky : napr. rozdelenie tovaru do stromu skupín(napr. Komodita, kategória, druh) , kde každá zložka bude reprezentovať dáta vyfiltrované podľa skupiny ku ktorej bol tovar priradený.

2.Užívateľom vytvárané zložky: Užívateľ si môže sám vytvoriť zložku tak, že zadá presný SQL príkaz pre filtráciu položiek. Takto vytvorené užívateľské zložky je možné prednastaviť ako zložky ktoré sa budú načítavať pri zobrazení modulu. Čo znamená, že jednotlivým užívateľom môžeme zakázať prístup k položkám tak, že vytvoríme zložku ktorá neobsahuje dáta, ktoré nechceme aby daný užívateľ videl a danú zložku mu nastavíme ako zložku ktorá sa má pre daného užívateľa zobrazovať pri spustení modulu.

Rozdiel medzi Filtrom a Zložkou:
Filter – položky modulu sa vyfiltrujú podľa podmienky len raz a uložia sa do pamäte. To znamená, že ak do modulu po vytvorení filtra vložíme nejaké položky ktoré spĺňajú podmienku filtra, tieto položky sa vo filtri nezobrazia.
Zložka – položky modulu sa vyfiltrujú vždy keď príde požiadavka na zobrazenie zložky. To znamená, že ak do modulu po vytvorení zložky pridám nejaké položky ktoré spĺňajú podmienku zložky, tieto položky sa v zložke zobrazia.

5.1.3 Vyhľadávanie a radenie položiek

Modul musí podporovať vyhľadávanie a radenie položiek podľa jednotlivých polí v položkách modulu. (napr. radenie a vyhľadávanie položiek pomocou mesta v module Adresa).
Aplikácia bude schopná dohľadať prvú položku ktorá bude obsahovať v poli, podľa ktorého sa vyhľadáva daný reťazec a nastaví kurzor na danú položku.

5.1.4 Obdobie a Rada

Doklady spojené s nákupom a tovarom sú identifikované nielen podľa unikátneho čísla v module ale aj pomocou obdobia do ktorého spadajú (napr. Rok 2006,2007,2008....) a rady obchodného tovaru. Napríklad Faktúry majú radu TU = tuzemsko, ZA = zahraničie. Identifikátor dokladu má potom tvar napr. 123/TU/2006 Pre každú radu v každom období, môže doklad začínať od čísla 1.

5.1.5 Potvrdzovanie dokladov

Niektoré doklady v obchodných procesoch je možné vytvoriť len raz a potom ich údaje nemôžu byť spätne menené aby nevznikli nezhody napríklad na sklade alebo v účtovníctve. Na zabránenie takýchto manipulácií bude slúžiť potvrdzovanie dokladov. Potvrdzovať doklady bude môcť len užívateľ ktorý ma na to právo a v niektorých prípadoch potvrdenie dokladu vedie aj k ďalším zmenám, ako je zmena počtu tovaru na sklade (príjemka, výdajka), uzavretie obchodného prípadu (zákazka, objednávka vydaná).

5.1.6 Dispozícia, Zadané, Rezervované , Objednané

Dispozícia – je to množstvo tovaru v skladových jednotkách ktoré je fyzicky na sklade.
Zadané – reprezentuje, aké množstvo tovaru v skladových jednotkách môžem vyskladniť na novej výdajke .To znamená, že ak mám dispozíciu 100 Ks ale 30 ks je požičaných tak môžem v skutočnosti vyskladniť len 70 Ks, číže zadané obsahuje 30 Ks.

Rezervované - reprezentuje, aké množstvo tovaru v skladových jednotkách potrebujem doobjednať.

Objednané - je to množstvo tovaru v skladových jednotkách ktoré je objednané, ale ešte nie je dodané.

5.2 Návrh databáze pre informačný systém

Databáza je jadrom informačného systému firmy. Základnou štruktúrou databáze sú tabuľky :

1. Užívateľ
2. Na Sklade
3. Tovar
4. Partneri
5. Kontaktné Osoby
6. Ponuky
7. Kúpne Zmluvy

8. Nákup
9. Predaj
10. Holder
11. ObjednávkyVy
12. FaktúruPrj
13. DodListPr
14. Prijemky
15. Zákazky
16. ObjednávkyPrj
17. FaktúryVyd
18. DodListVyd
19. RezList
20. Výdajky

V tabuľke ***uzivatel*** sú uložené jednotlivé údaje užívateľa, osobné údaje, ale ja tie, ktoré sú potrebné pre správne fungovanie systému.

```
CREATE TABLE Uzivatel (
 idUzivatel INTEGER UNIQUE NOT NULL,
 Funkcia INTEGER  NOT NULL REFERENCES Funkcia(idFunkcia),
 Logname VARCHAR(20)  UNIQUE NOT NULL,
 PSWD VARCHAR(45) NOT NULL,
 Meno VARCHAR(15) NOT NULL,
 Priezvisko VARCHAR(25) NOT NULL,
 Aktivny SMALLINT  NULL,  -- 1 aktivny , 0 neaktivny
 Popis VARCHAR(66) NULL,
 Tel VARCHAR(25) NULL,
 Email VARCHAR(40) NULL,
 DPZ DATE NOT NULL,
 DPZ_Kdo INTEGER NOT NULL REFERENCES Uzivatel (idUzivatel),
 Prihlaseny INTEGER NOT NULL,
```

```
  PRIMARY KEY(idUzivatel)
)With Oids;
```

Funkcia – reprezentuje funkciu užívateľa jak vo firme, tak v našom systéme

Logname – musí byť unikátne

PSWD – je zakódované v MD5. Pozná ho len daný užívateľ

Aktívny – reprezentuje či je daný užívateľ aktívny v systéme alebo nie. To znamená, že neaktívny užívateľ, nemôže v systéme nič robiť aj keď má na to práva. (Napríklad užívateľ ktorý pre danú firmu už nepracuje). Takto sa zachovajú všetky náveznosti bez potreby vymazať užívateľa.

DPZ – dátum poslednej zmeny

DPZ_Kdo - kdo poslední menil položku (ktorý užívateľ).

Prihlásený – koľkokrát je užívateľ prihlásený v systéme.

V tabuľke ***nasklade*** sú uložené jednotlivé údaje tovarov, ktoré sú aktuálne na sklade.

```
CREATE TABLE NaSklade (
  CiTov INTEGER NOT NULL REFERENCES Tovar (idTovar),
  Sarze INTEGER NOT NULL REFERENCES Sarze (idSarze),
  Umiestnenie INTEGER NOT NULL REFERENCES Umiestnenie (idUmiestnenie),
  Sklad INTEGER NOT NULL REFERENCES Sklady (idSklady),
  Dispozicia FLOAT NOT NULL,
  Zadane FLOAT NOT NULL,
  CenaSkl FLOAT NOT NULL
)With Oids;
```

Tabuľku ***tovar*** tvoria ucelené informácie o jednotlivých tovaroch.

```
CREATE TABLE Tovar (
```

```
idTovar INTEGER UNIQUE NOT NULL,
SklJed INTEGER  NOT NULL REFERENCES mnjednotka (idmnjednotka),
CisSklKart VARCHAR(20) UNIQUE NOT NULL,
Nazov VARCHAR(66) NOT NULL,
Znacka INTEGER NOT NULL REFERENCES Znacka (idZnacka),
AltJed INTEGER  NULL,  -- moze byt aj null
MnAlt FLOAT NULL,
SklCena FLOAT NULL,--
PredCena FLOAT NULL,--
MinMno FLOAT NULL,
MaxMno FLOAT NULL,
Mena INTEGER NOT NULL REFERENCES Mena (idMena),
Kurz FLOAT  NULL,
Mena_zap INTEGER NOT NULL, -- 0 nenastavujem menu, 1 - nastavujem menu
Objednanych FLOAT NULL,
PoslPredaj DATE NULL,
DV DATE NOT NULL,
DPZ DATE NOT NULL ,
DPZ_Kdo INTEGER NOT NULL REFERENCES Uzivatel (idUzivatel),
PoslNakup DATE NULL,
TypTov INTEGER NOT NULL REFERENCES typtovaru (idTypTovaru),--
StatPov INTEGER NOT NULL REFERENCES Stat (idStat),
Skratka2 VARCHAR(20)  NULL,
Popis varchar(66) NULL,
PopisnyText TEXT NULL,
PosCena FLOAT NULL,-- pos cena nakupu
Rezervovanych FLOAT NULL,
SklKarta_Cennik INTEGER NOT NULL, -- 1 sklkarta, 2 cennik, 3 oboje
dph INTEGER NOT NULL REFERENCES DPH (idDPH),
SlSerCis INTEGER NOT NULL, -- 0 nesledujem , 1 sledujem
SlSarz INTEGER NOT NULL, -- 0 nesledujem , 1 sledujem
```

```
  SlUmiest INTEGER NOT NULL, -- 0 nesledujem, 1 sledujem
  PRIMARY KEY(idTovar)
)With Oids;
```

MnAlt – množstvo alternatívnej jednotky (pomer skladovej a alternatívnej jednotky napr. 1Ks = 5kg)

DV – dátum vytvorenia

Typ Tovaru – Tovarový alebo Služba

StatPov – štát pôvodu

SklKarta_Cennik – kde sa bude položka zobrazovať, len cenník, len skladové karty alebo oboje

SlSerCis – sledovať sériové čísla
SlSarz - sledovať šarže

SlUmiest – sledovanie umiestnenia na sklade

Tabuľku ***partneri*** tvoria informácie o dodávateľoch a odberateľoch.

```
CREATE TABLE Partneri (
  IdPart INTEGER UNIQUE NOT NULL,
  Skratka VARCHAR(20)  UNIQUE NOT NULL,
  Skratka2 VARCHAR(20) NULL,
  ICO VARCHAR(20) NULL,
  DIC VARCHAR(20) NULL,
  Ulica VARCHAR(60) NOT NULL,
  Email VARCHAR(40) NULL,
  Splatnost INTEGER NOT NULL,
  ZpPlat INTEGER NOT NULL REFERENCES zpplatby (idZpPlatby),
  Banka INTEGER NOT NULL REFERENCES Banka (idBanka),
  DPZ DATE NOT NULL ,
  DV DATE NOT NULL,
  DPZ_Kdo INTEGER NOT NULL REFERENCES Uzivatel (idUzivatel),
  Region INTEGER NOT NULL REFERENCES Region (idRegion),
```

```
  Oblast INTEGER NOT NULL REFERENCES Oblast (idOblast),
  URL VARCHAR(50) NULL,
  Telefon VARCHAR(25) NULL,
  Fax VARCHAR(25) NULL,
  NadrPart INTEGER  NULL,
  Popis VARCHAR(66) NULL,
  ICDPH VARCHAR(20) NULL,
  Mobil VARCHAR(25) NULL,
  ZpDopr INTEGER NOT NULL REFERENCES zpdopravy (idZpDopravy),
  BCUct VARCHAR(34) NULL,
  Stav SMALLINT NULL,
  Registracia VARCHAR(200),
  Penale FLOAT NULL,
  PRIMARY KEY(IdPart)
)With Oids;
```

ZpPlat - spôsob platby

ZpDopr – spôsob dopravy

Registrácia – registrácia v obchodnom registri

BCUct – číslo bankového účtu

Tabuľku ***nákup*** tvoria informácie o prepojených položkách tovaru v dokladoch nákupu.

```
CREATE TABLE Nakup (
  CiPol INTEGER NOT NULL,
  ID INTEGER NOT NULL ,
  Rada INTEGER NOT NULL ,
  RadaFaktPrj INTEGER NOT NULL ,
  idFaktPrj INTEGER NOT NULL ,
  RadaDodListPr INTEGER NOT NULL ,
```

```
  idDodListPr INTEGER NOT NULL ,
  RadaPrijemky INTEGER NOT NULL ,
  idPrijemky INTEGER NOT NULL ,
  DatPot DATE NULL,
  CiTov INTEGER NOT NULL REFERENCES Tovar (idTovar),
  Sarze INTEGER NOT NULL REFERENCES Sarze (idSarze),
  Umiestnenie INTEGER NOT NULL REFERENCES Umiestnenie (idUmiestnenie),
  Mnozstvo FLOAT NULL,
  AltJed INTEGER NULL,
  Sklad INTEGER NOT NULL REFERENCES Sklady (idSklady),
  CeJednPredpo FLOAT NULL,
  CeJedn FLOAT NULL,
  CeSkl FLOAT NULL,
  DPH INTEGER NOT NULL REFERENCES DPH (idDPH),
  PRIMARY KEY(CiPol,ID, Rada)
)With Oids;
```

Tabuľku ***predaj*** tvoria informácie o prepojených položkách tovaru v dokladoch predaja.

```
CREATE TABLE Predaj (
  CiPol INTEGER  NOT NULL,
  ID INTEGER NOT NULL ,
  Rada INTEGER NOT NULL ,
  RadaRezList INTEGER NOT NULL ,
  idRezList INTEGER NOT NULL ,
  RadaVydajky INTEGER NOT NULL ,
  idVydajky INTEGER NOT NULL ,
  RadaFaktVyd INTEGER NOT NULL ,
  idFaktVyd INTEGER NOT NULL ,
  RadaObjednavkyP INTEGER NOT NULL ,
```

```
  idObjednavkyP INTEGER NOT NULL ,
  RadaDodListVy INTEGER NOT NULL ,
  idDodListVy INTEGER NOT NULL ,
  DatPot DATE NULL,
  CiTov  INTEGER NOT NULL REFERENCES Tovar (idTovar),
  Mnozstvo FLOAT NULL,
  Sarze INTEGER NOT NULL REFERENCES Sarze (idSarze),
  Umiestnenie INTEGER NOT NULL REFERENCES Umiestnenie (idUmiestnenie),
  AltJed INTEGER NULL,
  Sklad INTEGER NOT NULL REFERENCES Sklady (idSklady),
  CeJednPredpo FLOAT NULL,
  CeJedn FLOAT NULL,
  CeSklPredpo FLOAT NULL,
  CeSkl FLOAT NULL,
  DPH INTEGER NOT NULL REFERENCES DPH (idDPH),
  PRIMARY KEY(ID, Rada)
)With Oids;
```

Tabuľku ***holder*** tvoria informácie o položkách jednotlivých modulov,ktoré sú aktuálne menené nejakým užívateľom.

```
CREATE TABLE Holder(
  Uzivatel INTEGER NOT NULL REFERENCES Uzivatel(idUzivatel),
  Formular INTEGER NOT NULL,
  ID INTEGER NOT NULL,
  SKR Varchar(10) NULL,
  Prefix INTEGER NOT NULL
)With Oids;
```

Po prepojení daných tabuliek príslušnými väzbami vzniká návrh celkového systému vedenia obchodnej firmy

Obrázok 1.2 vid. Príloha

5.3 Návrh užívateľského prostredia pre systém informačného systému firmy

Užívateľské prostredie systému sa bude opierať o dobre navrhnutú databázu a bude maximálne využívať možnosti jazykov C# a PostgreSQL.

Užívateľské rozhranie – koncový užívateľ

Dôležitá je prehľadnosť, rýchlosť a schopnosť čo najlepšie užívateľa informovať o tom čo ho zaujíma.

Úvodná stránka musí byť upútava ale prehľadná, musí oboznamovať užívateľa s rozdelením do modulov systému tak ako mu je to známe z obchodných procesov vo firme

Prehľadnosť

Je jednou z najdôležitejších vecí, systém by mal byť navrhnutý tak, že vedie užívateľa krok za krokom od prihlásenia až k vystaveniu obchodných dokladov. Ak je postup čo i len trocha zložitejší, treba vytvoriť malý návod s ukážkovým príkladom v ktorom vysvetlíme celý postup.

Menu nášho informačného systému by malo byť prehľadné a výstižné. Odkazy v menu by nemali byť zanorené príliš hlboko. Treba si uvedomiť, že užívateľ nemusí používať niektoré obchodné procesy každý deň, a preto je pre neho jednoduchšie ak majú nadpisy v menu jednoduchý, výstižný ale presný výraz, ktorý privedie užívateľa presne tam kam si praje.

Informovanosť

Užívateľ musí byť informovaný o všetkých procesoch ktoré sa okolo neho dejú a musí byť informovaný o tom kam všade sa nemôže v našom aplikácii dostať.

Najhlavnejšie sú však informácie priamo spojené s obchodným procesom, ktorý sa snaží užívateľ vytvoriť v systéme.

Jednoduchosť

Treba predpokladať, že sa užívateľ nepotrebuje vydieť všetky detajly dokladov ktoré ho aktuálne nezaujímajú a preto treba dohľadanie dokladu čo najviac spríjemniť a zjednodušiť.

Prílišné množstvo nadbytočných informácií zvyšuje neprehľadnosť a náročnosť aplikácie. Väčšie množstvo informácií je podané užívateľovi až vtedy keď si ich praje vidieť.

Rýchlosť aplikácie

Informačný systém je čisto sieťovou aplikáciou, a preto sa pri ňom zvyšujú nároky na rýchlosť spracovania údajov a hlavne, na rýchlosť ich zobrazenia.

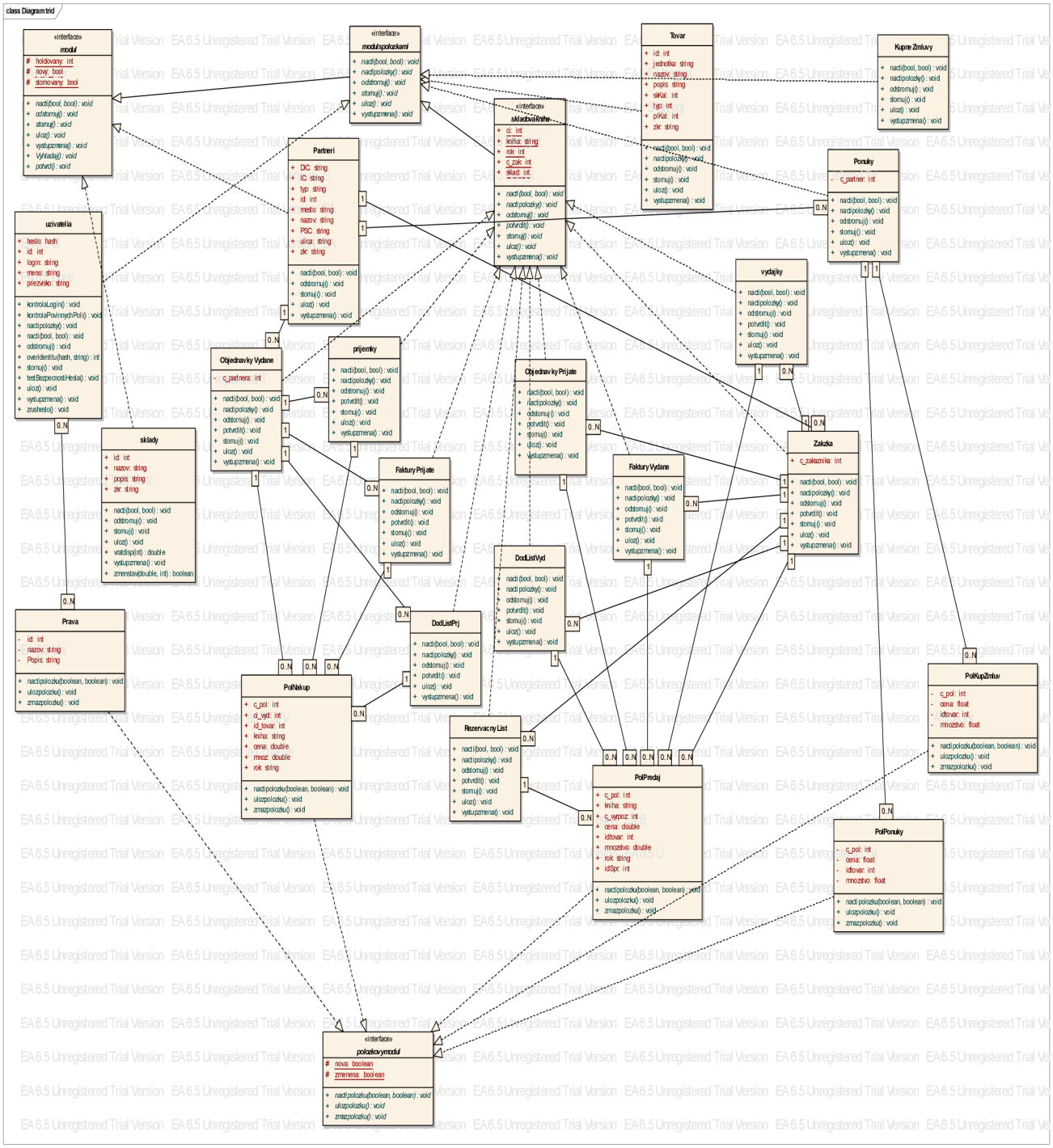
Aplikácia by mala zobrazovať údaje v čo najkratšom dosiahnuteľnom čase aj pri zobrazení veľkého množstva údajov. To sa môže dosiahnuť napríklad pomocou cacheovania dát.

Obrázok 1.3 vid. Príloha

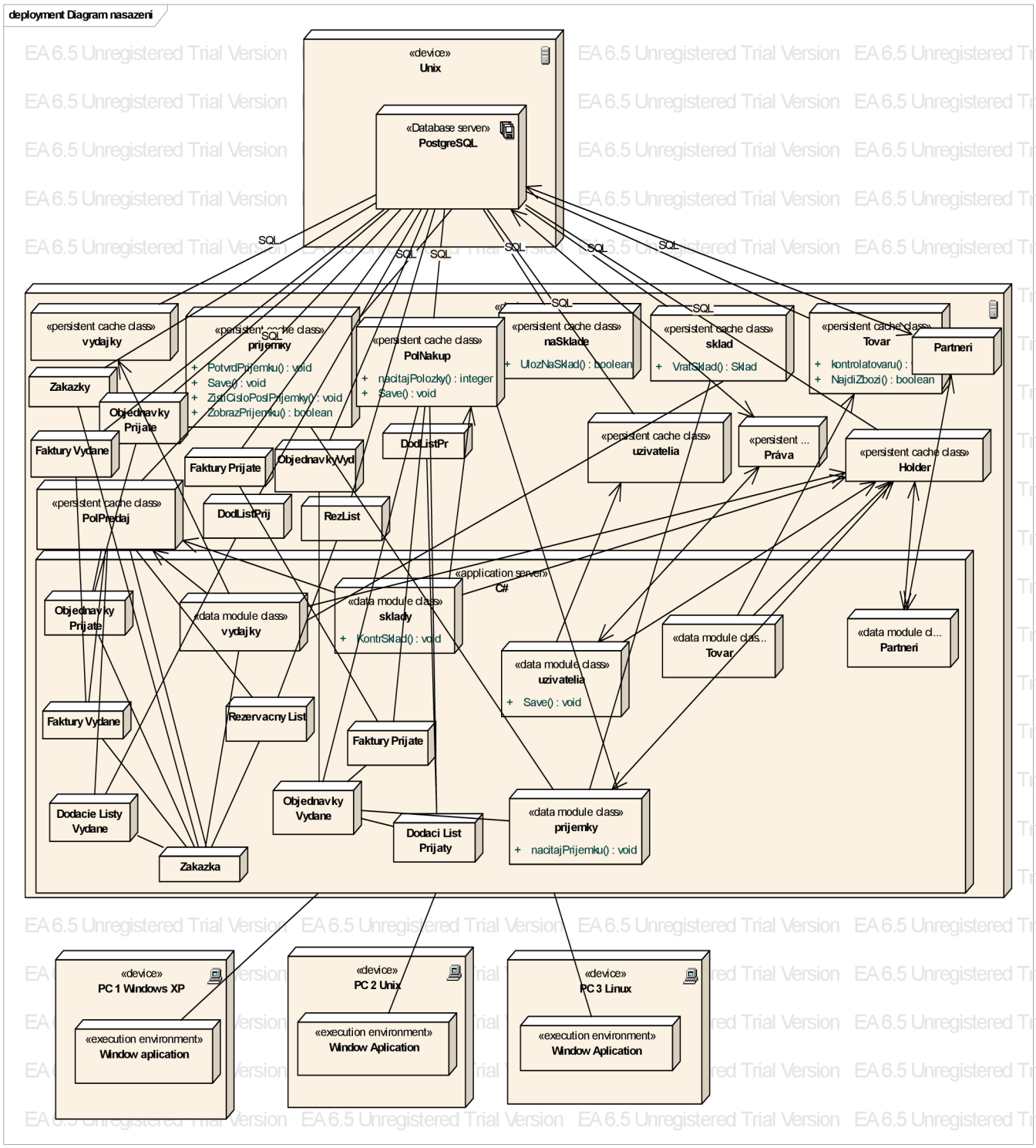
6 Diagram Tried

6.1 Návrh tried

Návrh tried jednotlivých modulov a ich náväznosť.



7 Architektúra aplikácie



8 Záver

Zadaním mojej semestrálnej práce bolo zoznámenie sa s technológiou C# , PostgreSQL a vytvorenie návrhu informačného systému firmy. Všetky body zadania boli splnené s tým, že táto práca bude rozvinutá diplomovou prácu ktorá vyústi do implementácie daného systému.

Literatúra :

[1] www.microsoft.com

[2] www.interval.cz

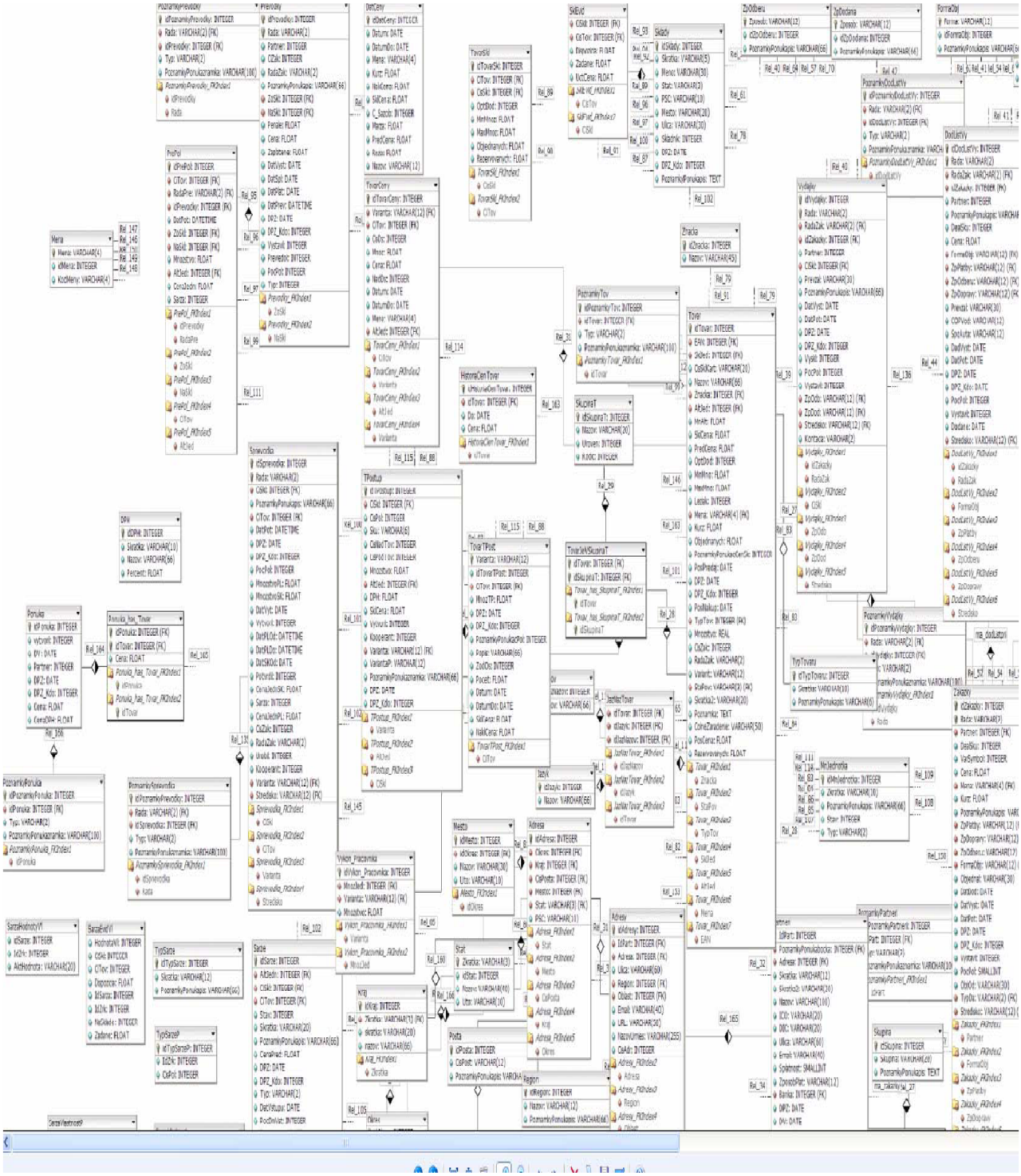
[3] www.developer.sk

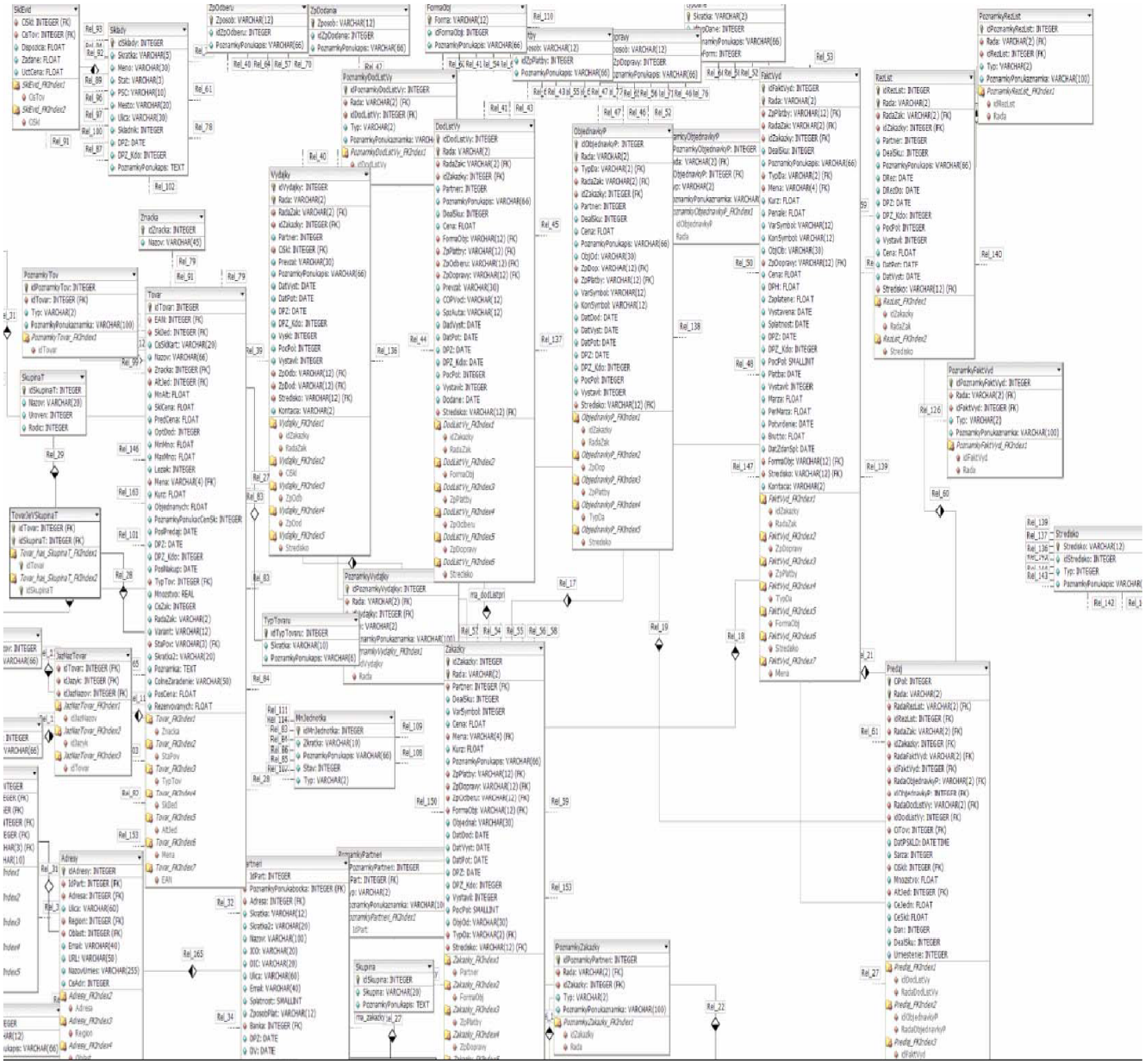
[4] Christian Negel, C# 2005 Computer Press 2005

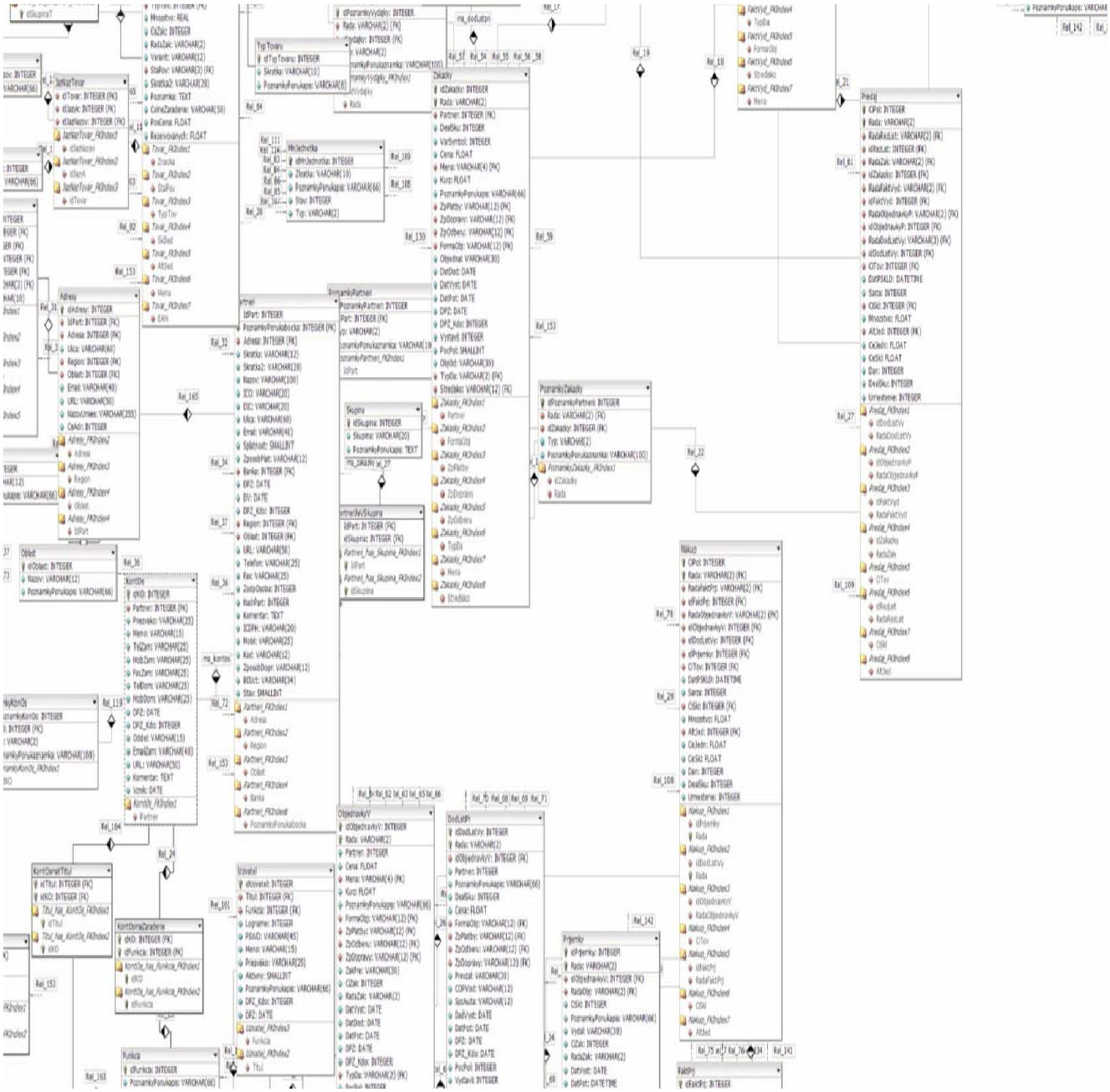
9 Príloha:

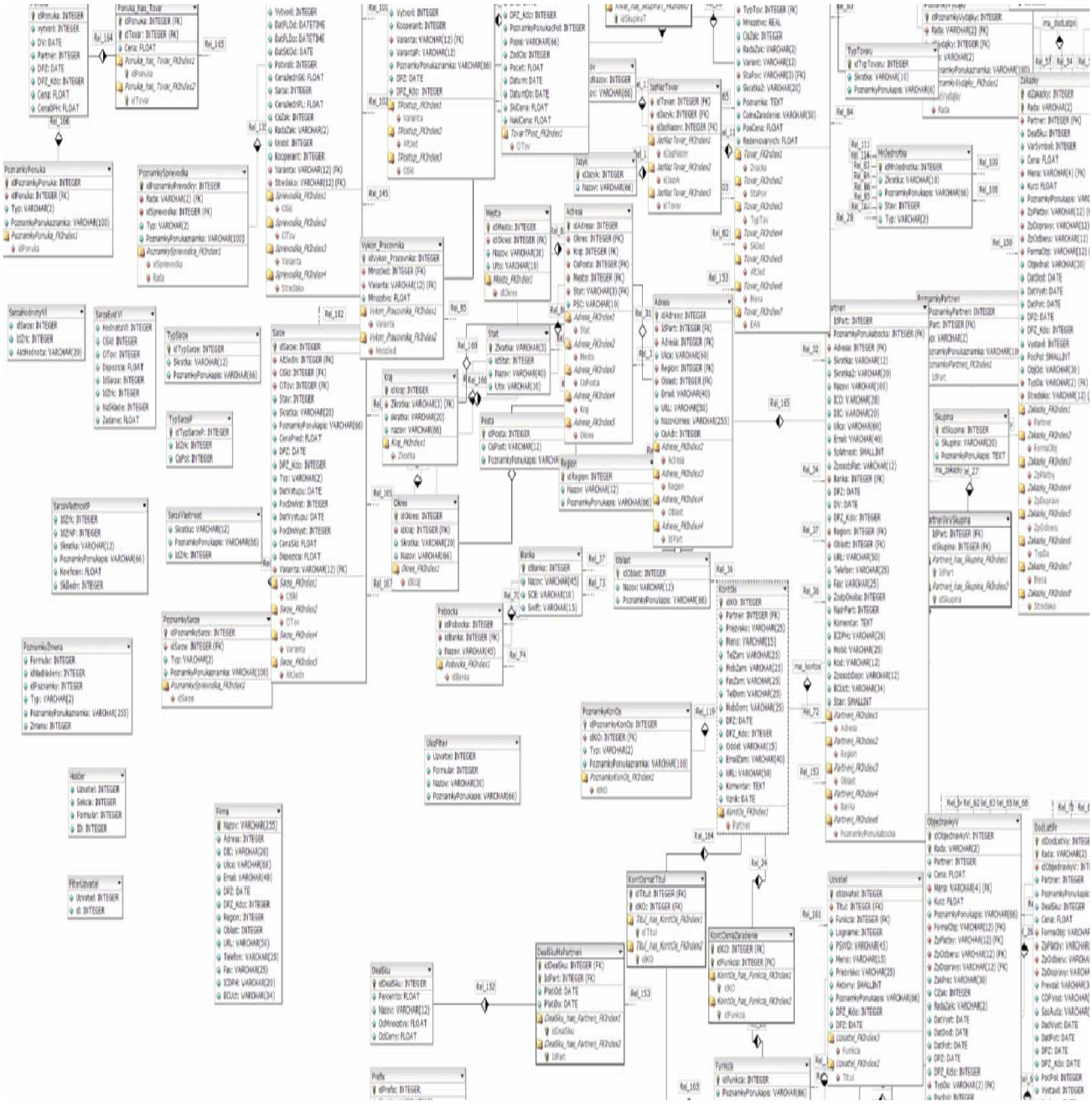
Obrazok 1.2

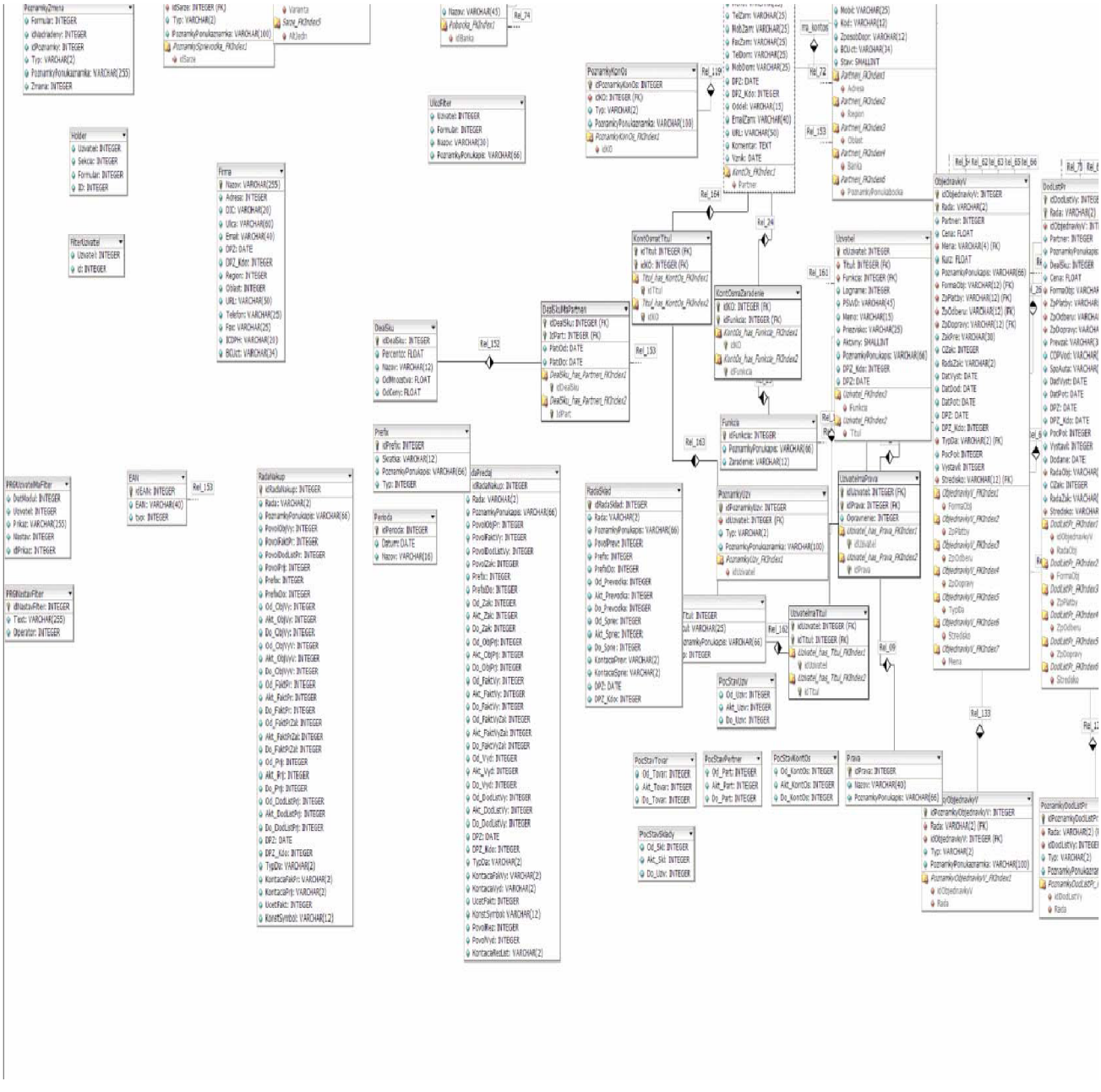
Návrh databáze.

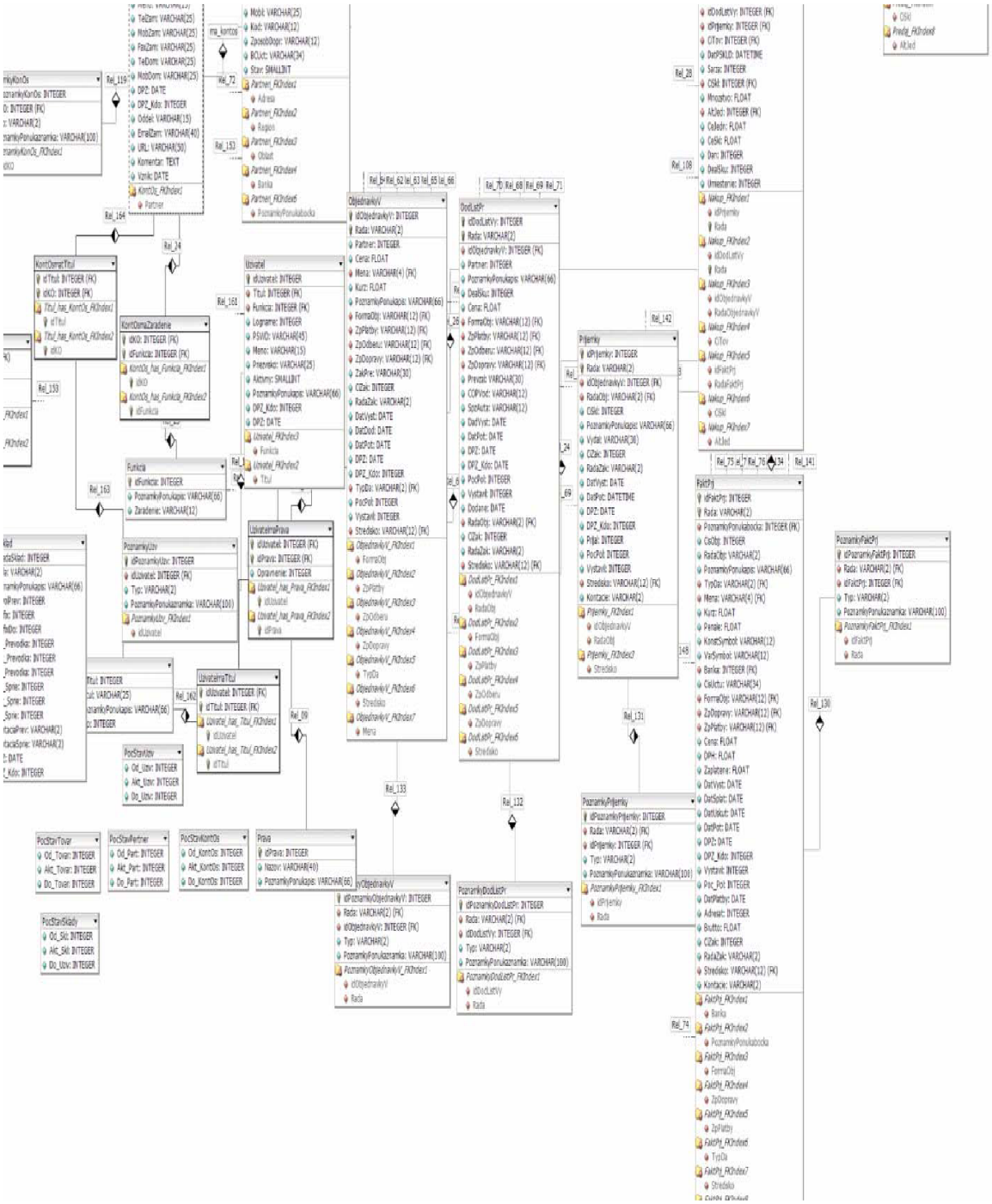






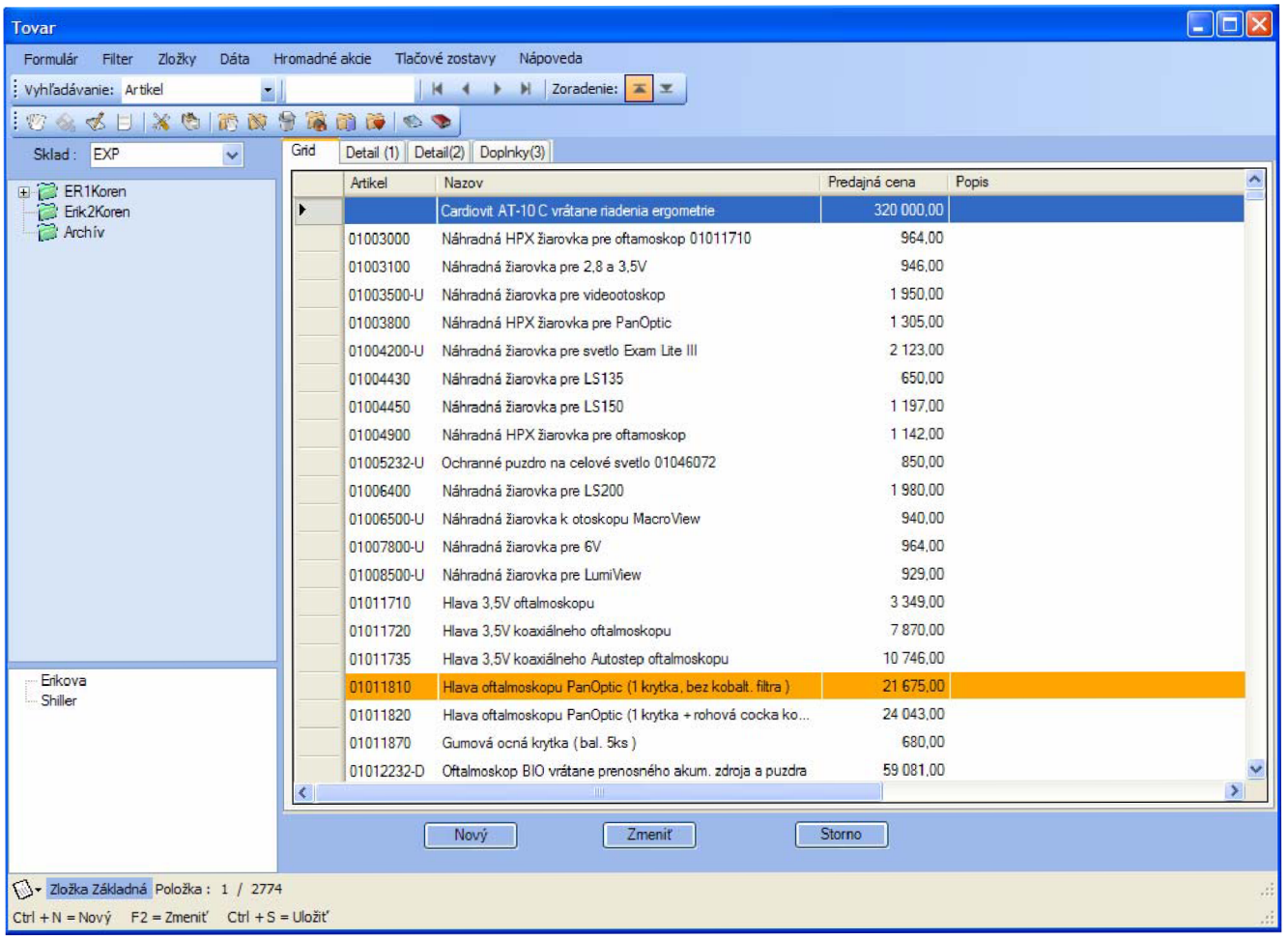


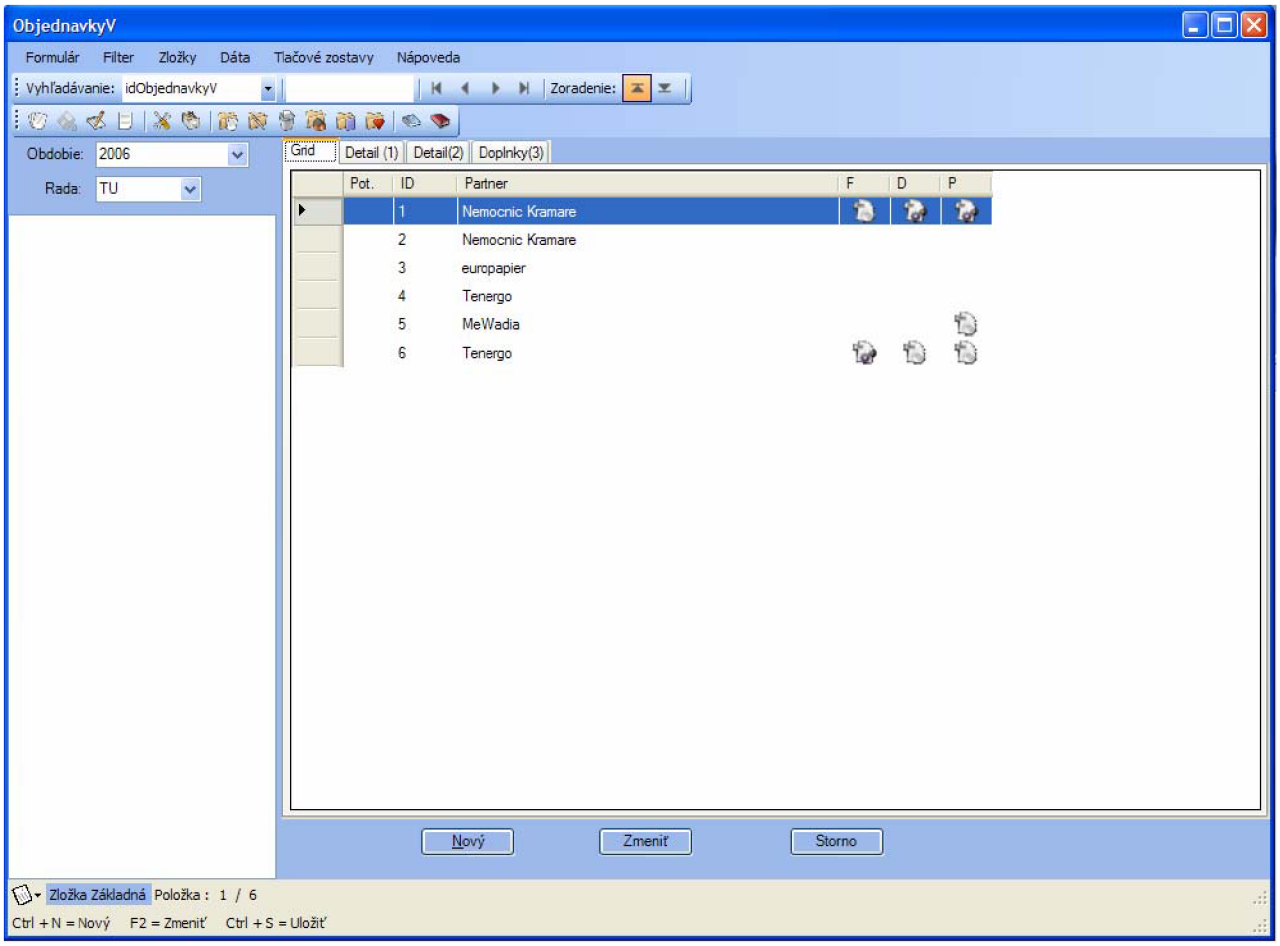


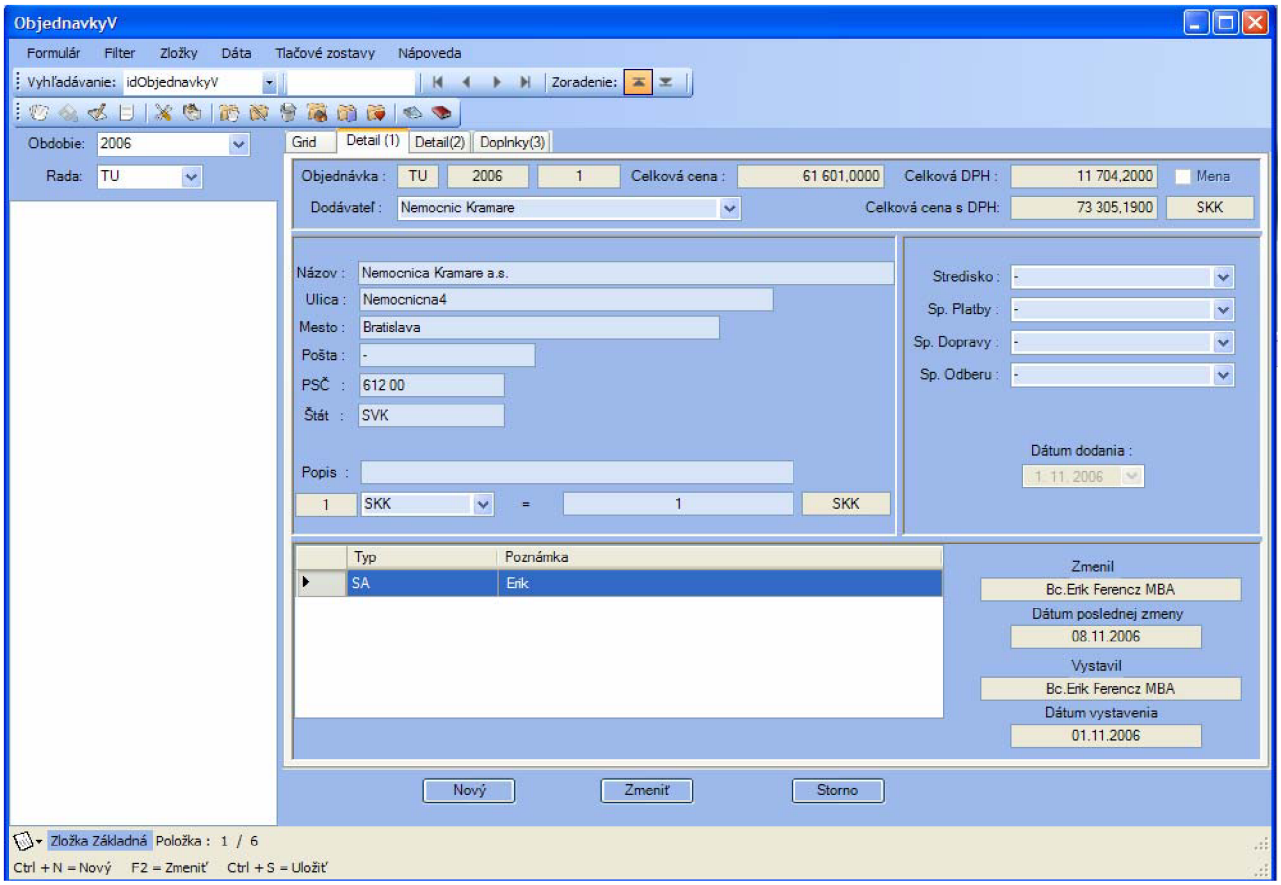


Obrazok 1.3

Návrh prostredia.




ObjednavkyV
Formulár Filter Zložky Dáta Tlačové zostavy Nápoveda
Vyhľadávanie: idObjednavkyV
Zoradenie:
Obdobie: 2006
Rada: TU
Grid Detail (1) Detail(2) Doplnky(3)
Pot. ID Partner F D P
1 Nemocnic Kramare
2 Nemocnic Kramare
3 europapier
4 Tenergo
5 MeWadia
6 Tenergo
Nový Zmeniť Storno
Zložka Základná Položka : 1 / 6
Ctrl + N = Nový F2 = Zmeniť Ctrl + S = Uložiť


ObjednavkyV
Formulár Filter Zložky Dáta Tlačové zostavy Nápoveda
Vyhľadávanie: idObjednavkyV
Zoradenie:
Obdobie: 2006
Rada: TU
Grid Detail (1) Detail(2) Doplnky(3)
Objednávka : TU 2006 1 Celková cena : 61 601,0000 Celková DPH : 11 704,2000 Mena
Dodávateľ : Nemocnic Kramare Celková cena s DPH: 73 305,1900 SKK
Názov : Nemocnica Kramare a.s.
Ulica : Nemocnicna4
Mesto : Bratislava
Pošta : -
PSČ : 612 00
Štát : SVK
Popis :
1 SKK = 1 SKK
Stredisko : -
Sp. Platby : -
Sp. Dopravy : -
Sp. Odberu : -
Dátum dodania : 1. 11. 2006
Typ Poznámka
SA Erik
Zmenil
Bc.Erik Ferencz MBA
Dátum poslednej zmeny
08.11.2006
Vystavil
Bc.Erik Ferencz MBA
Dátum vystavenia
01.11.2006
Nový Zmeniť Storno
Zložka Základná Položka : 1 / 6
Ctrl + N = Nový F2 = Zmeniť Ctrl + S = Uložiť