

**UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**IMPLEMENTACE PROPRIETÁRNÍHO INFORMAČNÍHO
SYSTÉMU PRO KONKRÉTNÍ FIRMU**

DIPLOMOVÁ PRÁCE

2009

Bc. Lukáš Jiránek

**UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**IMPLEMENTACE PROPRIETÁRNÍHO INFORMAČNÍHO
SYSTÉMU PRO KONKRÉTNÍ FIRMU**

DIPLOMOVÁ PRÁCE

**AUTOR PRÁCE: Bc. Lukáš Jiránek
VEDOUCÍ PRÁCE: Ing. Jana Holá, Ph. D**

2009

**UNIVERSITY OF PARDUBICE
FACULTY OF ELECTRICAL ENGINEERING
AND INFORMATICS**

**IMPLEMENTATION OF INFORMATION SYSTEM IN
A PARTICULAR COMPANY**

THESIS

**AUTHOR: Bc. Lukáš Jiránek
SUPERVISOR: Ing. Jana Holá, Ph. D**

2009

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Katedra softwarových technologií
Akademický rok: 2008/2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Lukáš JIRÁNEK**

Studijní program: **N2646 Informační technologie**

Studijní obor: **Informační technologie**

Název tématu: **Implementace proprietárního informačního systému pro konkrétní firmu**

Z á s a d y p r o v y p r a c o v á n í :

- Hlavním cílem diplomové práce je tvorba informačního systému, určeného pro koordinaci informačních toků v rámci jednotlivých oddělení, založeného na centrální databázi zákazníků a jejich objednávek. - Jádro systému bude vá automatické (dynamické) aktualizaci základních transakčních dat, systém musí být otevřený pro integraci s jinými firemními systémy za použití známých formátů (XML,SQL). - Základní požadavky na systém: modularita, komunikace klient-server, databázový server dostupný z Internetu. - Implementace pomocí technologií .NET, jazyka C# a SQL databáze.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **GÁLA, Libor, POUR, Jan, TOMAN, Prokop. Podniková informatika. 1. vyd. Praha : Grada, 2006. 482 s. ISBN 80-247-1278-4.**
2. **CHRISTIAN, Nagel, et al. C 2005 : Programujeme profesionálně. Jakub Mikuláščík, Petr Dokoupil. 1. vyd. Brno : Computer Press, a. s., 2006. 1400 s. ISBN 80-251-1181-4.**
3. **KEŘKOVSKÝ, Miloslav, DRDLA, Miloš. Strategické řízení firemních informací. 1. vyd. Praha : C.H. Beck, 2003. ISBN 80-7179-730-8.**

Vedoucí diplomové práce:


Ing. Jana Holá, Ph.D.
Katedra elektrotechniky

Datum zadání diplomové práce:

31. října 2008

Termín odevzdání diplomové práce:

22. května 2009



doc. Ing. Simeon Karamazov, Dr.

děkan



doc. Ing. Antonín Kavička, Ph.D.

vedoucí katedry

V Pardubicích dne 4. listopadu 2008

Prohlášení

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnut licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne

Bc. Lukáš Jiránek

PODĚKOVÁNÍ

Rád bych poděkoval vedoucí diplomové práce, paní Ing. Holé, Ph. D., za její vynaložený čas a spolupráci při tvorbě diplomové práce.

ANOTACE

Diplomová práce se zabývá návrhem a implementací systému podle zadaných požadavků, který bude po otestování nasazen do podnikové sítě. Výsledkem práce je funkční modulární systém, který umožňuje vyšší efektivitu při řešení firemních procesů zaměřených na správu objednávek a odběratelů.

KLÍČOVÁ SLOVA

Informační systém; vývoj systému; moduly; automatická synchronizace.

ANNOTATION

The thesis deals with design and implementation system in accordance with the requirements, which will be apply to the company network after test. The result of this work is modular system which allows greater efficiency in dealing with business processes to the management of orders and customers.

KEYWORDS

Information system; system development; modules; automatic synchronization.

Obsah

1 Úvod	2
2 Teoretická část	3
2.1 Informační systém (IS).....	3
2.2 ERP, plánování podnikových zdrojů.....	5
2.3 BI, Business Intelligence.....	8
2.4 CRM, řízení vztahů se zákazníkem.....	8
2.5 ECM, řízení podnikového obsahu.....	8
2.6 Technologie informačních systémů.....	8
2.6.1 Technologie .NET.....	9
2.6.2 Technologie Java.....	10
2.7 Relační databáze.....	12
2.7.1 Databázový systém MySQL.....	12
2.7.2 Databázový systém PostgreSQL.....	13
2.7.3 Databázový systém Oracle.....	14
3 Praktická část – Vývoj a implementace proprietárních IS	15
3.1 Popis firemní struktury zadavatele.....	15
3.2 Soupis požadavků zadavatele.....	16
3.2.1 Základní charakteristika navrhovaného systému.....	16
3.2.2 Soupis funkčních požadavků.....	16
3.2.3 Soupis nefunkčních požadavků.....	19
3.3 Analýza navrhovaného systému.....	20
3.3.1 Volba technologií.....	20
3.3.2 Model případů užití.....	20
3.3.3 Návrh analytických tříd.....	23
3.4 Návrh řešení systému.....	24
3.4.1 Databázový model.....	25
3.4.2 Model návrhových tříd.....	27
3.4.3 Popis jednotlivých částí navrženého systému.....	28
3.4.4 Realizace automatické synchronizace tabulek.....	34
3.4.5 Připojení modulu do systému.....	36
3.5 Administrace systému.....	37
3.5.1 Přihlášení uživatele.....	37
3.5.2 Hlavní program.....	38
3.5.3 Záložka Objednávky.....	39
3.5.4 Záložka Odběratelé.....	43
3.5.5 Záložka Uživatelé.....	48
3.6 Nasazení IS.....	49
4 Závěr	52
5 Literatura	53
6 Seznam ilustrací	55

1 Úvod

Firma pro zvýšení vlastní efektivity využívá dnes běžně informační a komunikační technologie, které zajišťují zejména informační podporu všech procesů. Informační systémy se stávají nepostradatelnými a zcela standardními nástroji, které pomáhají firmám ke zvýšení konkurenceschopnosti, protože dobře navrhnutý a fungující informační systém slouží ke snížení nákladů firmy, k urychlení firemních procesů a tím ke zvýšení jejich příjmů. V některých případech je pro firmy vhodné vyvinout a implementovat informační systém jako konkrétní řešení, které může lépe plnit požadavky firmy než obecná řešení.

Cílem této diplomové práce je návrh, tvorba a implementace informačního systému podle požadavků zadavatele, pro zajištění automatické správy a informační podpory procesu vyřízení zakázky (od přijetí objednávky po dodání výrobku). Vytvořený systém je aktuálně v beta verzi¹, která se odlaďuje.

¹ Beta verze je softwarový produkt, na kterém je již opravena většina chyb, nicméně je pořád nestabilní a na jeho chování se nedá spolehnout

2 Teoretická část

Obsahem této kapitoly je vymezení pojmu informační systém a přínosy nasazení informačního systému ve firmě. Dále kapitola obsahuje typy informačních systémů, kde se zaměřuje hlavně na popis informačního systému ERP (plánování podnikových zdrojů). Na závěr je v kapitole výčet používaných technologií a databázových systémů při tvorbě informačních systémů.

2.1 Informační systém (IS)

Informační systém (IS) lze chápat jako systém vzájemně propojených informací a procesů, které s těmito informacemi pracují. Přičemž pod pojmem procesy rozumíme funkce, které zpracovávají informace do systému vstupující a transformují je na informace ze systému vystupující. Zjednodušeně můžeme říci, že procesy jsou funkce zabezpečující sběr, přenos, uložení, zpracování a distribuci informací. Pod pojmem informace pak rozumíme data, která slouží zejména pro rozhodování a řízení v rozsáhlejších systémech.

Do celkové funkce IS se také promítá nezanedbatelná položka okolí. Okolí informačního systému tvoří veškeré objekty, které změnou svých vlastností ovlivňují samotný systém, a také objekty, které naopak mění své vlastnosti v závislosti na systému.

Jinak můžeme říci, že IS je softwarové vybavení firmy, které je schopné na základě zpracovávaných informací řídit procesy podniku nebo poskytovat tyto informace řídicím pracovníkům tak, aby byli schopni vykonávat řídicí funkce, mezi které patří zejména plánování, koordinace a kontrola veškerých procesů firmy [1].

Přínosy nasazení informačního systému ve firmě

Podnikové databáze mohou být před zavedením informačního systému spravovány na mnoha různých místech. Vzájemné sdílení dat mezi zaměstnanci nefunguje nebo vyžaduje velké úsilí. Dochází také k chybám, protože databáze neobsahují zcela aktuální údaje.

Informační systém ve firmě tuto situaci řeší. V ideálním případě jsou veškerá data uložena a spravována v rámci IS. Informace je do systému zapsána pouze jednou.

Když některý ze zaměstnanců zjistí, že u zákazníka nebo obchodního partnera došlo ke změnám, zanesse tyto nové údaje do podnikového IS. Aktualizace se okamžitě projeví ve všech odděleních, které mají k informačnímu systému (IS) přístup.

Bezpečnost citlivých údajů nelze bez IS spolehlivě zajistit. Může se sice zaznamenat chování uživatelů pomocí různých programů pro sledování, ale jen obtížně se zjistí například historie všech změn u zákazníka.

IS zaznamenává veškerou historii změn. Jednoduše se zjistí, kdo si údaje zobrazil, kdy je změnil, jaká byla původní a nově zadaná hodnota.

Seznam kontaktů, zákazníků a obchodních partnerů patří k hodnotným firemním datům. Jestliže se vedou informace o obchodních partnerech v tabulkovém editoru (například Microsoft Excel), lze takovou databázi odcházejícím zaměstnancem jednoduše zkopírovat a odnést.

Podnikový informační systém chrání firmu před ztrátami a problémy, které jsou spojeny s krádeží firemních databází. Zatímco se databáze v Excelu může jednoduše zkopírovat, každý export údajů z IS je zaznamenán a může být dokonce i zakázán.

Oběh a schvalování dokumentů nebo plateb odpovědnými pracovníky může být zdlouhavý. Dříve se dokumenty schvalovaly podpisem. To ovšem znamenalo přítomnost dotyčného pracovníka ve firmě.

Současné IS umožňují přehledně zobrazit všechny důležité údaje na jednom místě. Odpovědný pracovník schválí vše potřebné pohodlně kliknutím myši. Odpadá hledání založených nebo ztracených dokumentů. Všechno se významně urychluje.

Nasazení informačního systému (implementace IS) přináší výhodu také pro vedení firmy. Veškeré finanční, analytické a další výkazy je možné získat přímo z IS v **reálném čase**.

Firemní IS je navíc možné přizpůsobit oboru podnikání a zvyklostem daného odvětví. Dodavatelé IS proto nabízejí oborová řešení. Vycházejí přitom ze zkušeností a osvědčených požadavků ostatních firem. Nasazení takto upraveného řešení sice někdy

vyžaduje změnu vnitřních procesů, výhody a nové postupy ovšem umožní firmě dále růst.

Významnou výhodou je také možnost **vedení finančního výkaznictví** podle mezinárodních standardů, povinných pro firmy, působících na mezinárodních trzích.

Mezi další přínosy IS patří rovněž možnost propojení podniku s dodavateli a odběrateli na úrovni informačního systému. Internetovým obchodům umožňují výrobci přímo propojit systém s elektronickým obchodem. Pokud zavedení systému proběhne úspěšně, zvýší se efektivita výkonu zaměstnanců [2].

Informační systémy jsou rozděleny podle klíčových funkcí:

- ERP , plánování podnikových zdrojů,
- Business Intelligence,
- CRM, řízení vztahu k zákazníkům,
- ECM, řízení podnikového obsahu.

2.2 ERP, plánování podnikových zdrojů

Informační systém kategorie ERP (Enterprise Resource Planning) definujeme jako účinný nástroj, který je schopen pokrýt plánování a řízení všech klíčových interních podnikových procesů (zdrojů a jejich transformace na výstupy), a to na všech úrovních od strategické až po operativní. K těmto klíčovým procesům patří: výroba, logistika, personalistika a ekonomika.

ERP systémy se z hlediska nutnosti integrace dalších podnikových procesů rozvinuly do podoby, která se označuje jako ERP II, nebo "Extended" ERP. Podniková praxe si v rámci plánování podnikových zdrojů vyžádala těsnější propojení interních procesů s:

- externími procesy, u nichž není přesně definovaný vlastník a jejichž efektivní řízení nemá management podniku plně pod kontrolou (oblast řízení vztahů se zákazníky a řízení dodavatelského řetězce),

- procesy podporujícími (zejména) vrcholové rozhodování (OLAP reporting, datové sklady).

Mezi nejdůležitější vlastnosti ERP systému patří:

- zefektivnění a zrychlení ekonomických procesů,
- centralizaci dat a snížení chyb,
- dlouhodobé úspory v investicích do informačních systémů a hardware,
- zvýšení bezpečnosti,
- rychlejší výstupy pro vedení firmy,
- podpora pro vedení účetnictví podle mezinárodních standardů.

V konečném důsledku IS zvyšuje flexibilitu, takže i konkurenceschopnost[6].

K zásadním požadavkům, které od ERP systémů očekávají zákaznické organizace, patří:

- realizace měřitelných přínosů v oblasti snižování celé struktury nákladů vznikající neefektivním řízením firmy,
- realizace neměřitelných přínosů v oblasti řízení podnikových procesů a dostupnosti informací v reálném čase.

Moderní ERP systém je založen na vysoce sofistikovaných hardwarových a softwarových komplementárních produktech jako jsou databázové systémy, síťové operační systémy, víceprocesorové servery apod. K podpoře a plnění složitých podnikových cílů, resp. řízení podnikových procesů však nestačí pouze software a výkonný hardware. ERP systém by nemohl splnit svoje poslání bez aktivní účasti zaměstnanců a uživatelů systému. Všechny zmíněné aspekty musejí být navíc v souladu se strategií firmy, kterou sdílí jak management, tak pracovníci na všech úrovních řízení [7].

ERP systémy mají dva základní modely dodání:

On-premise model. Aplikace je nainstalována na serverech organizace vlastníci ERP systém. Organizace musí mít vnitřní zdroje na provoz a údržbu ERP systému.

Na upgradech, aktualizacích a úpravách systému se podílí sama organizace spolu s dodavatelskou firmou. Jedná se o nejběžnější model využívání ERP systémů.

On-demand model. Tento model je znám také pod pojmy ASP (Application service provider) nebo SaaS (Software as a Service). Přestože mezi jednotlivými pojmy jsou rozdíly, hlavním společným rysem je, že ERP systém je dodáván vzdáleně přes internet. O aktualizace a upgrady systému se stará dodavatel, který ERP provozuje na svých serverech. U tohoto modelu bývají větší obavy o bezpečnost a spolehlivost služby, protože organizace nemá přímou kontrolu nad správou ERP systému.

Historie ERP

Počátek systémů ERP sahá do šedesátých let minulého století, kdy některé větší organizace vyvíjely a nasazovaly centralizované počítačové systémy, automatizující nejnáročnější úlohy spojené s chodem podniku – především kontrolu zásob a skladů. V sedmdesátých letech byly vyvinuty systémy řešící materiálové plánování výroby Material Requirements Planning (MRP). Od počátku osmdesátých let se pak v podnicích nasazovaly systémy sloužící pro řízení a optimalizaci dodávek materiálů a výroby (Manufacturing Resources Planning – MRP II). První systémy ERP (Enterprise Resource Planning) se začaly objevovat od konce osmdesátých let a na počátku let devadesátých. Ty už zpravidla zasahovaly širší oblast fungování podniku s výše uvedenými funkcionalitami a moduly. Přebíraly na svá bedra množství úloh dříve vykonávaných lidmi a vedly postupně k integraci většiny podnikových procesů. V protikladu ke starším systémům, tradičně vyvíjeným uvnitř firem speciálně pro danou organizaci, se jednalo o integrované, komerčně nabízené produkty, které lze přizpůsobit dané společnosti nebo odvětví. Tyto komplexní a drahé systémy však vyžadovaly implementaci náročnou na lidské zdroje a další náklady. Mnohdy bylo jejich zavádění spojeno také se změnou podnikových procesů – s jejich reengineeringem, tak aby odpovídaly jednotlivým modulům a nejlepším používaným postupům pro dané odvětví [8].

"Nový impulz přinesla všeobecná dostupnost a klesající cena informačních technologií. Trh s ERP rostl až o desítky procent ročně, protože podnikům přinášely konkurenční výhodu a mnohdy i radikální zlepšení fungování. Další kapitolou byl pak nástup internetu, který řešení, dříve omezená na hranice jediného podniku, propojuje

se systémy dalších organizací. Dodavatelé podnikových systémů reagují zabudováním nových funkcí a otevírají také architekturu systémů. Cílem je možnost komunikovat s dalšími aplikacemi i systémy mimo mateřský podnik," shrnuje Stanislav Fleissig, Senior Account Manager společnosti IXTENT s.r.o.

2.3 BI, Business Intelligence

BI (Business Intelligence) představuje komplex procesů, aplikací a technologií, které téměř výlučně podporují analytické a plánovací činnosti podniků a organizací a jsou postaveny na principu multidimenzionality, kterým zde rozumíme možnost nahlížet na realitu z několika možných úhlů pohledu.

BI je úzce spjata na ostatní systémy (hlavně na systémy ERP), čerpá z nich vstupní data a stále častěji i tato data do ostatních systémů vrací. To vede k dílčímu pracovnímu závěru, že kvalita řešení BI je úzce závislá na kvalitě ostatních (transakčních) systémech, zejména na kvalitě jejich produkčních dat nebo databází [3].

2.4 CRM, řízení vztahů se zákazníkem

CRM (Customer relationship management) je databázovou technologií podporovaný proces shromažďování, zpracování a využití informací o zákaznících firmy. Umožňuje tak poznat, pochopit a předvídat potřeby, přání a nákupní zvyklosti zákazníků a podporuje oboustrannou komunikaci mezi firmou a jejími zákazníky. Jako CRM v přeneseném smyslu se též označuje softwarové, hardwarové a personální vybavení firmy, které je výkonem těchto funkcí pověřeno[4].

2.5 ECM, řízení podnikového obsahu

ECM (Enterprise content Management) je strategií, metodikou a nástrojem pro zachycení, uchovávání, správu a doručení obsahu související s organizačními procesy. ECM nástroje a strategie umožňují spravovat nestrukturované informace a všechny informace které existují.[5].

2.6 Technologie informačních systémů

K tvorbě informačních systémů se v současné době zejména používají dva hlavní směry – technologie Java a technologie .NET.

2.6.1 Technologie .NET

.NET („dotnet“ podle anglického dot NET = tečka NET, NET pochází z anglického slova network - síť) je zastřešující název pro soubor technologií v softwarových produktech, které tvoří celou platformu, která je dostupná nejen pro Web, Windows i Pocket PC. Common Language Infrastructure je standardizovaná specifikace jádra .NET.

Základní komponentou je Microsoft .NET Framework, prostředí potřebné pro běh aplikací a nabízející jak spouštěcí rozhraní, tak potřebné knihovny. Pro vývoj .NET aplikací vydal Microsoft Visual Studio .NET.

Platforma .NET nepředepisuje použití žádného programovacího jazyka. Bez ohledu na to, v čem byla aplikace původně napsána, se vždy přeloží do mezijazyka Common Intermediate Language.

Nejpoužívanější programovací jazyky pro vývoj .NET aplikací jsou C#, Visual Basic .NET a Delphi. C# je programovací jazyk podobný jazykům C nebo Java. VB.NET je pokračovatelem jazyka Visual Basic [9].

K dispozici je nicméně řada dalších programovacích jazyků.

Výhody technologie .NET

Objektově orientované programování: Platforma .NET Framework je již od začátku kompletně založena na objektově orientovaných principech.

Dobry návrh: Knihovna základních tříd je od základu navržena vysoce intuitivním způsobem.

Jazyková nezávislost: V technologii .NET se všechny jazyky kompilují do společného *zprostředkujícího jazyka* (Intermediate Language). To znamená, že jazyky mohou spolupracovat způsobem, který dříve nebyl k dispozici.

Lepší podpora dynamických webových stránek: Technologie .NET nabízí integrovanou podporu webových stránek pomocí nové technologie ASP.NET. V rámci ASP.NET je kód stránek kompilován a lze jej psát v jazyku vysoké úrovně kompatibilním s technologií .NET.

Účinný přístup k datům: Sada komponent .NET souhrnně označována jako ADO.NET zajišťuje efektivní přístup k relačním databázím a různým zdrojům dat. Jsou také k dispozici komponenty, které poskytují přístup k systému souborů a k adresářům. Zejména je do technologie .NET integrována podpora XML, což umožňuje manipulaci s daty, která lze implementovat nebo exportovat na jiné platformy než Windows.

Sdílení kódu: Technologie .NET zavádí koncepci sestavení (assembly), která nahrazují tradiční knihovny DLL. Sestavení mají formální funkce pro správu verzí a je možné, aby vedle sebe existovaly různé verze sestavení.

Zabezpečení: Každé sestavení může také obsahovat integrované informace o zabezpečení, jež mohou přesně určovat, který uživatel nebo kategorie uživatelů či procesů smí volat určité metody definovaných tříd.

Instalace s nulovým dopadem: Existují dva typy sestavení: sdílená (shared) a soukromá (private). Sdílená sestavení jsou společné knihovny dostupné všem programátorům, zatímco soukromá sestavení jsou určena pouze k použití s konkrétním software. Soukromé sestavení je zcela soběstačné, takže se zjednodušuje proces instalace. Nevznikají žádné položky registru. Příslušné soubory stačí umístit do správné složky systému souborů.

Podpora webových služeb: Technologie .NET zahrnuje plně integrovanou podporu vývoje webových služeb, které lze vytvářet stejně snadno jako libovolný jiný typ aplikace.

Visual Studio: Pro technologii .NET se dodává vývojové prostředí Visual Studio, které umožňuje stejně snadno pracovat s jazyky C++, C#, J# a Visual Basic i s kódem ASP.NET.

C#: C# je nový objektově orientovaný jazyk, který je určen pro spolupráci s technologií .NET [10].

2.6.2 Technologie Java

Platforma Java je ideální volbou pro network computing. Lze jí využít na všech typech zařízení od serverů po mobilní telefony či smart karty. Její hlavní předností je

sjednocení jednotlivých podnikových infrastruktur a schopnost vytvoření kompaktní, plně zabezpečené síťové platformy pro jakýkoli typ podniku či uživatele.

Platforma Java těží z podpory velmi široké komunity vývojářů, kteří se aktivně podílejí na uvádění produktů založených na technologii Java do provozu a na trh. Díky této komunitě se platforma také neustále vyvíjí. Centrálním prvkem jejího rozvoje je program Java Community Process, což je otevřený program založený členy komunity proto, aby se bylo možno spolehnout na dodržování daných standardů [11].

Výhody jazyka Java:

Jednoduchý: jeho syntaxe je zjednodušenou (a drobně upravenou) verzí syntaxe jazyka C a C++. Odpadla většina konstrukcí, které způsobovaly programátorům problémy a na druhou stranu přibyla řada užitečných rozšíření.

Objektově orientovaný: s výjimkou osmi primitivních datových typů, jsou všechny ostatní datové typy objektové.

Distribuovaný: je navržen pro podporu aplikací v síti (podporuje různé úrovně síťového spojení, práce se vzdálenými soubory, umožňuje vytvářet distribuované klientské aplikace a servery).

Interpretovaný: místo skutečného strojového kódu se vytváří pouze tzv. mezikód. Tento formát je nezávislý na architektuře počítače nebo zařízení. Program pak může pracovat na libovolném počítači nebo zařízení, který má k dispozici interpret Javy, tzv. virtuální stroj Javy – Java Virtual Machine (JVM).

V pozdějších verzích Javy nebyl mezikód přímo interpretován, ale před prvním svým provedením dynamicky zkompilován do strojového kódu daného počítače (tzv. just in time compilation - JIT). Tato vlastnost zásadním způsobem zrychlila provádění programu v Javě, ale výrazně zpomalila start programu.

Robustní: je určen pro psaní vysoce spolehlivého softwaru – z tohoto důvodu neumožňuje některé programátorské konstrukce, které bývají častou příčinou chyb (např. správa paměti, příkaz goto, používání ukazatelů). Používá tzv. silnou typovou kontrolu – veškeré používané proměnné musí mít definovaný svůj datový typ.

Dynamický: Java byla navržena pro nasazení ve vyvíjejícím se prostředí. Knihovna může být dynamicky za chodu rozšiřována o nové třídy a funkce, a to jak z externích zdrojů, tak vlastním programem.

Elegantní: velice pěkně se v něm pracuje, je snadno čitelný (např. i pro publikaci algoritmu), přímo vyžaduje ošetření výjimek a typovou kontrolu [12].

2.7 Relaçní databáze

Relaçní databáze jsou databáze, vyhovující relačnímu modelu dat (RMD). RMD je založený na matematické teorii množin a predikátové logice a definuje způsob reprezentace dat, způsob jejich ochrany (integritní omezení) a možné operace nad daty.

RMD se jmenuje relační podle vztahů mezi daty. Základním kamenem RMD je databázová relace (tabulka) obsahující data.

Relace lze chápat jako tabulku dat uspořádanou do sloupců (atribut + doména) a řádků (n -tic – n rozměrný vektor). Databázová relace je tabulka, pohled, výsledek dotazu, což nám dává možnost pracovat s výsledky dotazů stejně jako s tabulkou [13].

Mezi nejrozšířenější databázové systémy patří:

- MySQL,
- PostgreSQL,
- Oracle,
- Microsoft SQL Server,
- DB2.

2.7.1 Databázový systém MySQL

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci GPL², tak pod komerční placenou licenci.

² Licence GPL poskytuje uživatelům počítačového programu práva svobodného softwaru a používá copyleft k zajištění, aby byly tyto svobody ochráněny, i když je dílo změněno nebo k něčemu přidáno.

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace MySQL, PHP a Apache jako základní software webového serveru.

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, trigger, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet [14].

2.7.2 Databázový systém PostgreSQL

PostgreSQL je relační databázový systém s otevřeným zdrojovým kódem. Má za sebou více než patnáct let vývoje a zakládá si na spolehlivosti a bezpečnosti. Je šířen pod licencí BSD³, které umožňuje volné spojování otevřeného kódu s uzavřeným. Často je srovnáván s další rozšířenou otevřenou databází MySQL.

Funguje pod rozšířenými operačními systémy včetně Windows a Linuxu. Splňuje podmínky ACID (Atomicita, Konzistence, Izolace a Trvalost), plně podporuje cizí klíče, operace JOIN, pohledy, spouště a uložené procedury. Obsahuje většinu SQL92 a SQL99 datových typů.

PostgreSQL umožňuje běh uložených procedur napsaných v několika programovacích jazycích, v Perlu, v Pythonu, v jazyku C nebo v speciálním PL/pgSQL – jazyku vycházejícím z PL/SQL firmy Oracle.

Předností systému PostgreSQL je rozšiřitelnost. Systém lze obohatit o nové datové typy, funkce, operátory, agregační funkce, procedurální jazyky.

³ BSD licence je licence pro svobodný software, mezi kterými je jednou z nejsvobodnějších. Umožňuje volné šíření licencovaného obsahu, přičemž vyžaduje pouze uvedení autora a informace o licenci, spolu s upozorněním na zřeknutí se odpovědnosti za dílo.

Nevýhodou, zvláště pro jednotlivce a malé firmy, je (hlavně v porovnání s MySQL) malá rozšířenost PostgreSQL na hostingových serverech a menší komunita, která by pomáhala s případnými problémy [15].

2.7.3 Databázový systém Oracle

Oracle je systém řízení báze dat (Oracle database management system – DBMS), moderní multiplatformní databázový systém s velice pokročilými možnostmi zpracování dat, vysokým výkonem a snadnou škálovatelností.

Přesněji Oracle Corporation je název firmy, oficiální název databázové platformy je Oracle Database.

Aktuální verzí je Oracle Database 11g. Tento systém podporuje nejen standardní relační dotazovací jazyk SQL podle normy SQL92, ale také proprietární firemní rozšíření Oracle (např. pro hierarchické dotazy), imperativní programovací jazyk PL/SQL rozšiřující možnosti vlastního SQL (v tomto jazyce je možné tvořit uložené procedury, uživatelské funkce, programové balíky a triggery), dále podporuje objektové databáze a databáze uložené v hierarchickém modelu dat (XML databáze, jazyk XSQL). Dále též obsahuje širokou paletu nástrojů pro podporu snadného nasazení na gridových sítích (písmeno g v označení verze je zkratkou "Growing to Grid"). Grid Computing podporovala i verze 10g (zde písmeno g značí pouze slovo Grid) [16].

3 Praktická část – Vývoj a implementace proprietárních IS

V této části se zabývám postupem při tvorbě aplikace. Nejprve popíši strukturu firmy a její požadavky na systém. Poté se začnu zabývat analýzou a návrhem řešení. Dále popíši části, které jsou do dané aplikace implementovány, a na závěr ukážu funkce a ovládání vytvořené aplikace.

Vývoj informačního systému probíhal v následujících krocích:

- určení charakteristiky systému podle požadavků,
- výběr technologie,
- analýza požadavků,
- vývoj jádra systému,
- implementace.

K tvorbě informačního systému byly použity nástroje:

- Case Studio verze 2.25: Nástroj pro tvorbu relačních databázových modelů.
- Microsoft Visual Studio 2008: Vývojové prostředí pro psaní programů technologií .NET.
- Enterprise Architect verze 7.0: Nástroj pro návrh a modelování systémů pomocí jazyka UML⁴.

3.1 Popis firemní struktury zadavatele

Zadávající firma se zabývá výrobou kancelářského a školního nábytku. Nyní je rozdělena do 3 oddělení – dřevovýroba, kovovýroba a obchod. Do budoucna se plánuje rozšíření firmy o další oddělení. Ne všechna oddělení se nacházejí v jedné lokalitě.

Pokud firma obdrží novou objednávku, dostane se nejprve do obchodního oddělení, kde se začne zpracovávat. Zde je zakázka rozdělena na jednotlivé části podle

⁴ UML (Unified Modeling Language) je jazyk s bohatou sémantikou a syntaxí, který usnadňuje návrh a vizualizaci různých typů systémů

výroby – na část určenou pro oddělení dřevovýroby, pro oddělení kovovýroby a nebo objednána od externího dodavatele. Výrobní oddělení obdrží seznam výrobků potřebných k vyrobení. Pokud jsou všechny části hotové, převezou se do oddělení dřevovýroby, kde se nábytek smontuje a odveze zákazníkovi.

Aktuální řešení ve firmě:

Firma používá 2 informační systémy. Jeden funguje jako databáze všech škol v České republice a druhý zobrazuje objednávky a jejich aktuální stav. Nevýhoda daného řešení je duplicita dat uchovávaných v systémech a absence rozšiřujících funkcí ulehčující práci se systémem.

3.2 Soupis požadavků zadavatele

V této kapitole jsou sepsány požadavky zadavatele na daný systém. Požadavky určují:

- základní charakteristika navrhovaného systému,
- funkční požadavky – popisují požadovanou službu systému,
- nefunkční požadavky – omezení současnou infrastrukturou.

3.2.1 Základní charakteristika navrhovaného systému

Základem navrhovaného systému je automatická správa a informační podpora procesu vyřízení zakázky (od přijetí objednávky po dodání výrobku). Systém je vytvářen podle síťové architektury klient-server, která odděluje klienta a server, kteří spolu komunikují přes počítačovou síť. Klientská část obsahuje většinu aplikační logiky. Serverová část obsahuje centrální databázi. Systém je zabezpečen autorizací a autentizací uživatele.

3.2.2 Soupis funkčních požadavků

Mezi funkční požadavky navrhovaného systému patří:

- nová aplikace bude spojovat funkcionalitu systémů aktuálně fungujících ve firmě, za účelem odstranění duplicitního vkládání dat do databází,
- import/export dat z/do jiných systémů ve firmě (automaticky či manuálně),

- data vstupují vyplněním formulářů uživatelem,
- aplikace bude obsahovat 4 základní moduly: odběratelé, objednávky, reporty, uživatelé.

Funkce v modulu odběratelé:

- zobrazovat všechny odběratele uložené v databázi rozdělené podle jejich typu (základní škola, střední škola...),
- přidání a editace odběratele,
- každé oddělení si bude moci nastavit připomínku k odběrateli a napsat text do poznámky (např. připomínka k zaslání katalogu),
- filtrovat odběratele.

Formulář odběratele má tyto náležitosti:

- typ odběratele,
- název,
- IČO, DIČ,
- seznam kontaktních osob se zadaným kontaktem (telefon, fax, email),
- fakturační adresa včetně kraje a okresu,
- poštovní adresa,
- poznámky s připomínkou.

Funkce v modulu objednávky:

- zobrazovat objednávky podle fáze, v jaké se nacházejí (nové, ve výrobě, po termínu, vyrobené, dodané),
- prohlížení objednávek a jejich příloh,
- přidání a editace objednávek,

- odebrání nepotvrzených objednávek,
- filtrovat objednávky,
- automaticky přesouvat nesplněné objednávky do sekce po termínu,
- automaticky upozorňovat na nově přidané objednávky a na objednávky přesunutá do další fáze.

Formulář objednávky má tyto náležitosti:

- číslo zakázky,
- oddělení na která se objednávka vztahuje,
- termín výroby a dodání,
- počet odpracovaných hodin,
- cena objednávky,
- dodací adresa (pokud se liší od fakturační),
- informaci zda se jedná o reklamaci,
- informaci zda obsahuje objednávku od subdodavatele (např. Nákup tabulí od externí firmy),
- prioritu objednávky,
- přílohy, které obsahují seznam výrobků a doplňující údaje,
- seznam výrobků.

Funkce v modulu reporty:

- zobrazovat statistiky pro všechna oddělení dohromady,
- zobrazovat statistiky pro jednotlivá oddělení,
- zobrazovat statistiky objednávek:

- počet objednávek vyrobených a dodaných před a po termínu za určité období,
- obrat za dané období.

Funkce v modulu uživatelé:

- měnit uživatelská hesla,
- nastavení uživatelům přístupových práv do částí systému:
 - vkládat objednávky a odběratele,
 - editovat všechny objednávky a odběratele,
 - editovat objednávky vytvořené aktuálním uživatelem,
 - stornovat objednávky,
 - mazat objednávky vytvořené aktuálním uživatelem,
 - přístup k reportům všech oddělení,
 - přístup k reportům svého oddělení,
 - měnit stav objednávky (nová, vyrobená, dodaná).

Práva uživatele budou kombinací výše sepsaných omezení přístupů do systému.

Administrátorský účet bude vytvářet uživatelské účty, nastavovat práva uživatele a importovat/exportovat data do/z databáze.

3.2.3 *Soupis nefunkčních požadavků*

Mezi nefunkční požadavky navrhovaného systému patří:

- uložení dat na firemním serveru,
- provoz systému na počítačích s operačním systémem Windows XP,
- komunikace s databází pomocí síťové architektury klient/server.

3.3 Analýza navrhovaného systému

V této kapitole se zabývám analýzou vytvářeného systému dle zadaných požadavků sepsaných v kapitole 3.2 *Soupis požadavků zadavatele*. Nejprve byly vytvořeny modely případů užití. Dále z případů užití vychází analytické třídy, které slouží pro návrh tříd v systému.

3.3.1 Volba technologií

K tvorbě dané aplikace je zapotřebí definovat jaké technologie se budou používat k vytvoření aplikace a jaký typ databázového systému zvolit.

Jelikož není v požadavcích zadána podmínka k multiplatformní aplikaci, ale pouze podmínka k funkčnosti pod operačním systémem Windows XP, systém bude psán v jazyce C# za použití technologie .NET. Tato technologie je využívána hlavně k tvorbě počítačových aplikací s výhodou jednoduššího programování a vyšší úrovně zabezpečení dané aplikace.

Volba databázového systému je vybrána podle velikosti společnosti a použitého operačního systému na databázovém serveru. S dodanou aplikací bude pracovat dohromady 10 – 20 uživatelů, proto není podmínkou vybírat databázové systémy pro velké společnosti (např. Oracle). Na trhu je mnoho databázových systémů zdarma a jejich funkce se vyrovnávají systémům u placených verzí. Ve firmě nyní funguje databázový server s operačním systémem Linux a proto je nejvýhodnější použít databázový systém PostgreSQL.

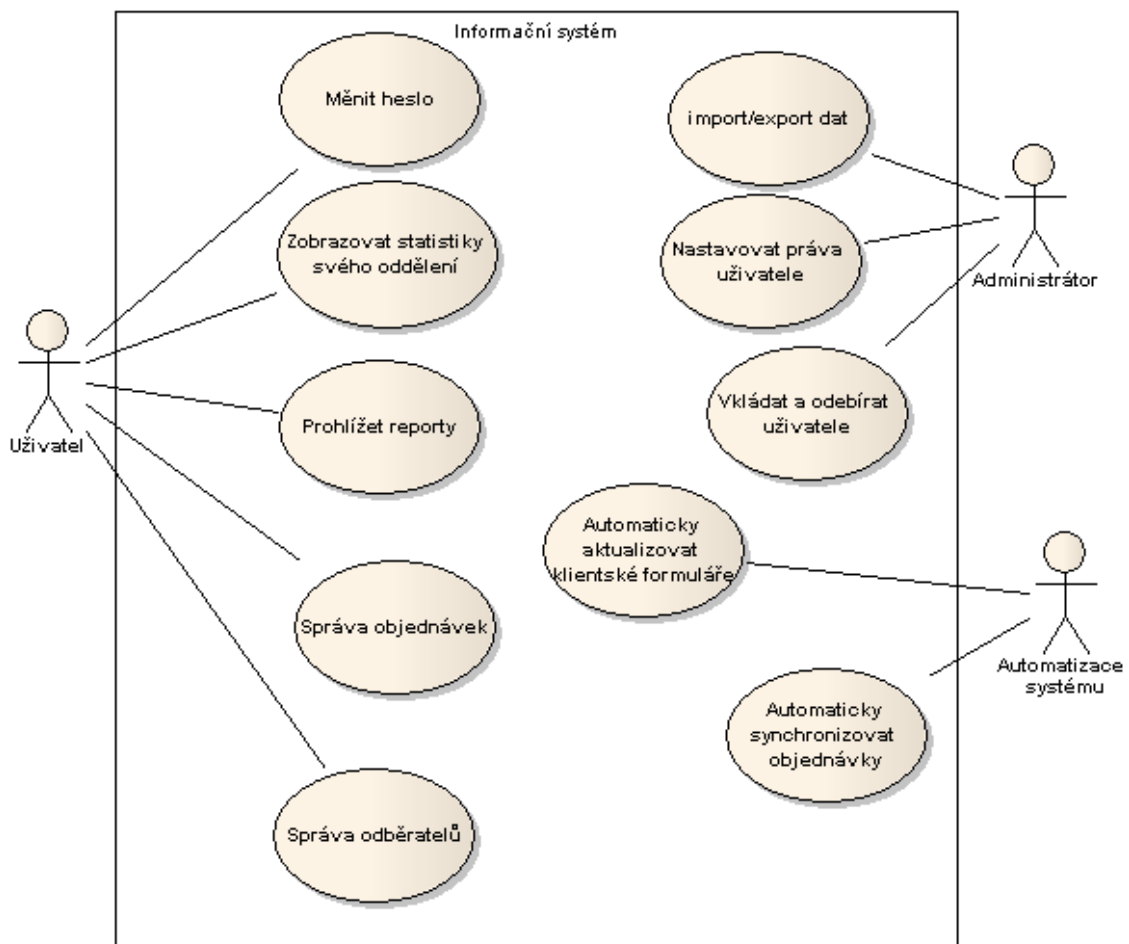
3.3.2 Model případů užití

Na obrázku číslo 1 je zobrazen model případů užití vytvořen z požadavků sepsaných v kapitole 3.2 *Soupis požadavků zadavatele*. Model obsahuje rámeček určující námi vytvářený systém pojmenovaný jako *Informační systém*. V něm jsou vypsány základní funkce, které má systém obsahovat. Dále obrázek obsahuje entity, které vstupují do systému a úsečky propojující entity s danými funkcemi.

Entita *Uživatel* znázorňuje uživatele pracujícího se systémem. Pro příliš jemné dělení přístupových práv jsou uživatelům přiřazeny všechny funkce, bez ohledu na to, jaká práva bude mít nastavena.

Entita *Administrátor* zobrazuje v modelu jeho nadřazené funkce v systému nad jakýmkoliv jiným uživatelem.

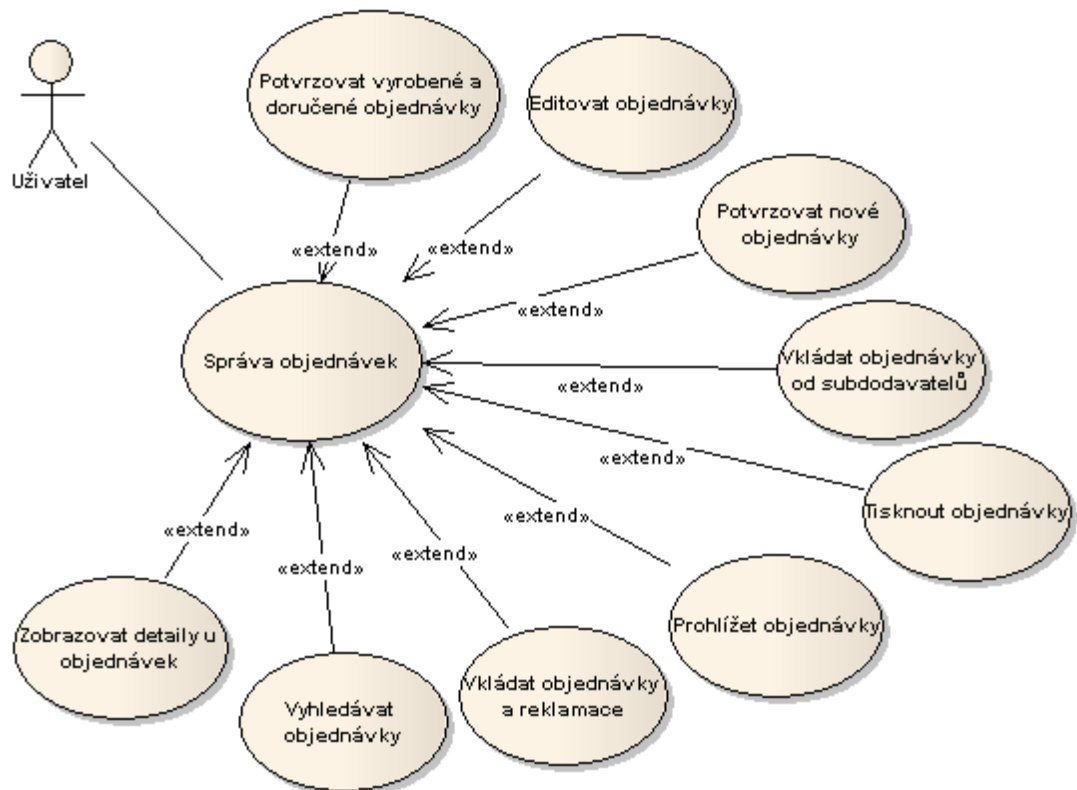
Entita *Automatizace systému* se liší od ostatních entit v tom, že není uživatelem, který systém používá, ale je to součást programu, která se spouští automaticky po jeho zapnutí.



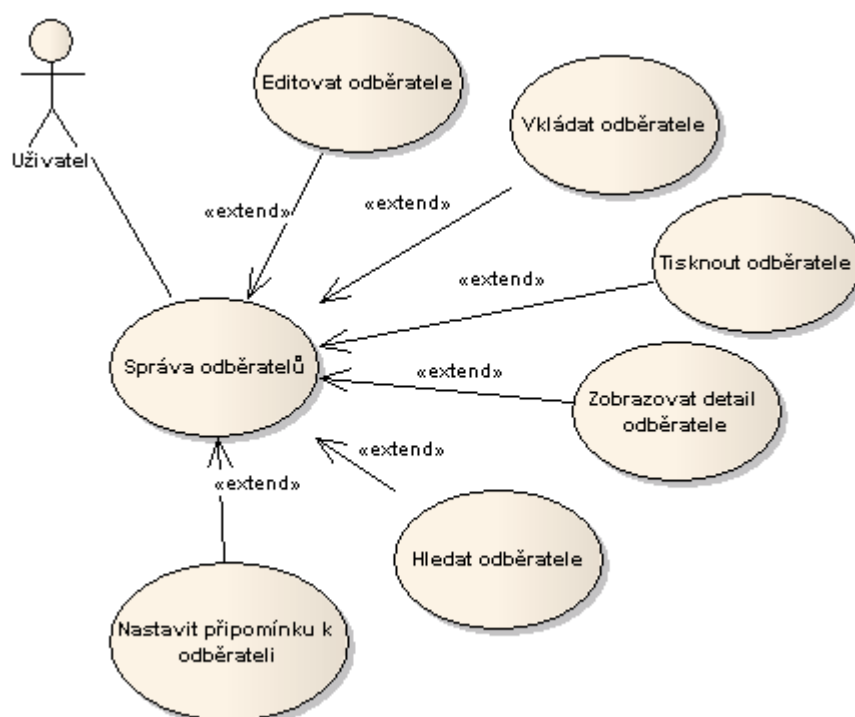
Obrázek 1: Případy užití – role v systému

Pro lepší přehlednost jsou detailnější funkce pro práci s objednávkami a odběrateli shrnuty do jedné množiny. Výpis doplňujících funkcí je zobrazen na obrázku 2 pro správu objednávek a na obrázku 3 pro správu odběratelů.

Případy užití jsou vytvořeny v aplikaci Enterprise Architect.



Obrázek 2: Případy užití pro správu objednávek



Obrázek 3: Případy užití pro správu odběratelů

3.3.3 Návrh analytických tříd

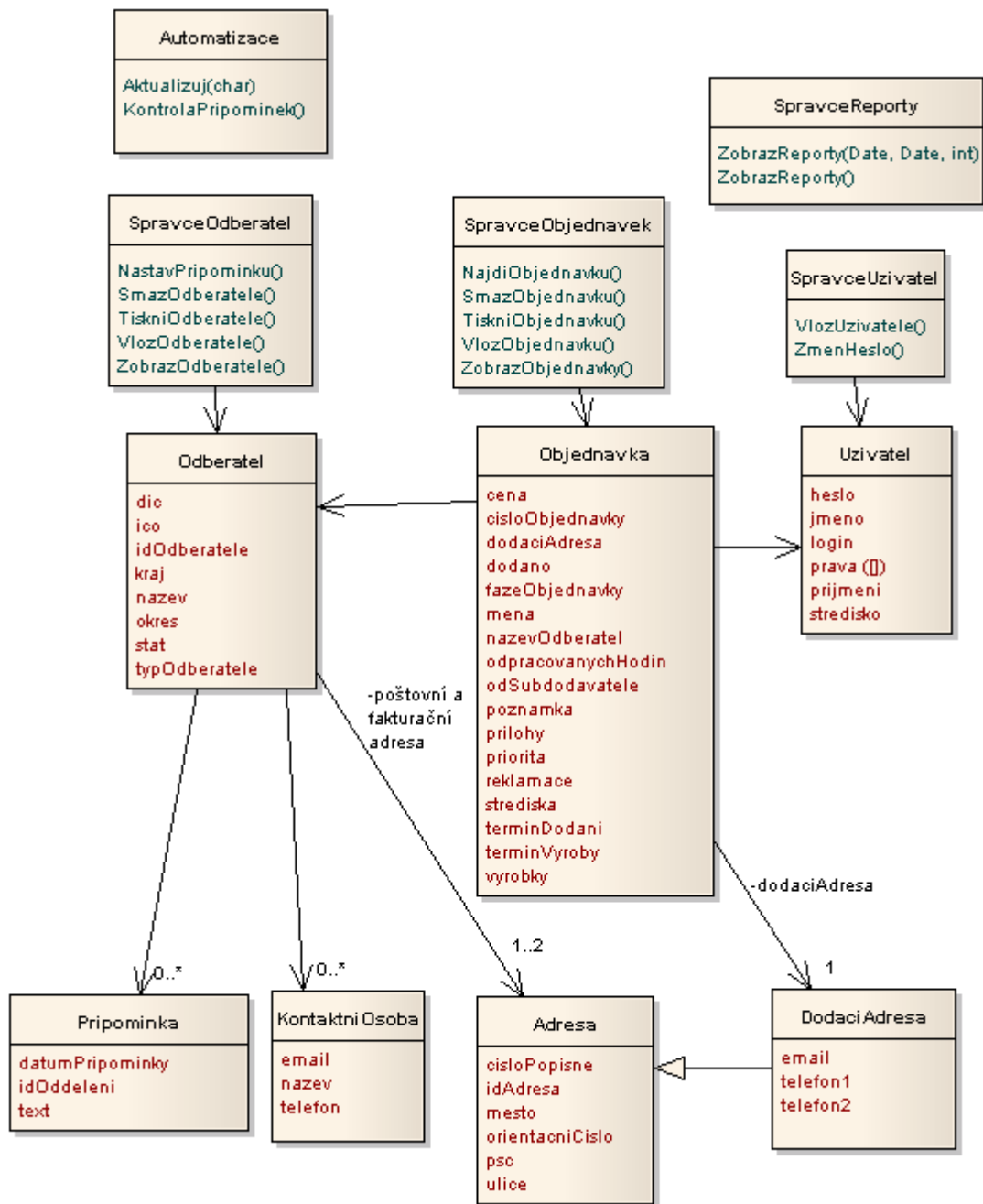
Návrh analytických tříd ukazuje základní logiku systému. To znamená, jaké základní třídy budou ve výsledném systému obsaženy a jaké atributy nebo metody budou obsahovat. Při tvorbě analytického modelu se model omezuje pouze na třídy, jež jsou součástí slovníku problémové domény. Modelem se snažíme o maximální stručnost a jednoduchost formulace struktury a chování systému v analytickém modelu. Tento model je dále využíván v další fázi tvorby systému [17].

Návrh analytických tříd je vytvořen v aplikaci Enterprise Architect.

Na obrázku číslo 4 je zobrazen návrh analytických tříd, které byly vytvořeny podle předchozí kapitoly 3.3.2 *Model případů užití*. Základní třídy v modelu jsou *Odběratel*, *Uživatel*, *Objednavka* a jejich podřazené třídy, které nám ukazují, jaké atributy se v nich budou uchovávat.

Třídy, které začínají slovem *Spravce*, zastřešují podřazené třídy a obsahují ovládací metody.

Poslední třída *Automatizace* bude mít funkce, automaticky synchronizovat načtené údaje s údaji v databázi.



Obrázek 4: Analytický model

3.4 Návrh řešení systému

Tato kapitola se zabývá návrhem řešení. Nejprve se věnuje návrhu databázového modelu, včetně jeho popisu. Poté se zabývá návrhem modelu tříd a na závěr popisuje hlavní části programu.

3.4.1 Databázový model

Databázový model je nástroj pro prezentaci struktury a funkcionality databáze. Umožňuje definovat schéma databáze, určující organizaci dat, způsoby jejich ochrany a zajištění správnosti (integritní omezení) a přípustné operace s daty.

Model pro dané zadání je zobrazen na obrázku 5. Centrem v daném modelu jsou tabulky *Odberatel*, *Objednavka* a *Uzivatel*.

Tabulka *Uživatel*

Tabulka *Uzivatel* obsahuje všechny uživatele, kteří mohou přistupovat do systému. Kromě základní identifikace uživatele obsahuje tabulka následující vlastnosti:

- Oddělení, ve kterém je daný uživatel zaměstnán.
- Časy kdy naposledy načel informace z tabulek *Odberatel* a *Objednavka*. Tyto atributy existují z důvodu automatické synchronizace dat, která má uživatel aktuálně načtena v aplikaci a dat, která se nacházejí v databázi.
- Propojení s tabulkou *prava_uzivatele*, která obsahuje seznam přístupových práv nastavená pro daného uživatele.

Tabulka *Odběratel*

Tabulka *Odberatel*, uchovává odběratele, přidané do systému. Mezi vlastnosti odběratele patří:

- Atributy *ico* a *dic* jsou nastaveny jako nepovinné, protože odběrateli mohou být i soukromé osoby.
- Cizí klíč *id_uzivatel* uchovává uživatele, který daného odběratele vytvořil. To je z důvodu vyhledávání a zamezení pozměnění údajů neoprávněným uživatelem.
- Zadání fakturační adresy je povinné a pokud se liší poštovní adresa od fakturační, uchovává se rovněž.
- Možnost uložení odběratele pod více typů, ale minimálně do jednoho (například mateřská a základní škola)

- Možnost nastavení pouze jedné poznámky pro každé oddělení, včetně data, kdy se má poznámka zobrazit.
- Obsahuje kontaktní osoby, které patří k danému odběrateli.

Tabulka *Objednávka*

Tabulka *Objednávka* obsahuje objednávky vložené do systému. Vlastnosti objednávek jsou následující:

- Atribut *od_subdodavatele* nastavuje, zda objednávka obsahuje zboží objednané od externího dodavatele
- Počet hodin potřebných ke zhotovení dané objednávky.
- Datумы *termin_dodani* a *termin_vyroby* uchovávají předběžné termíny pro výrobu a dodání objednávky. Termíny, kdy byla opravdu objednávka vyrobena a dodána uchovávají atributy *dodano* a *vyrobena*.
- Pokud se liší fakturační adresa od adresy dodání, uloží se do tabulky *dodaci_adresa*, včetně základních kontaktů na osobu.
- Obsahuje stav *faze_objednavky*, který nám ukazuje, v jaké fázi se objednávka nachází (například vyrobená nebo dodaná)
- Funkcí tabulek *sloupce_faze_objednavky* a *sloupec_objednavka* je uchovávat názvy sloupců, které mají být vykresleny pro každou fázi objednávky.

Návrhové třídy obsahují kompletní specifikaci:

- Kompletní sadu atributů včetně jejich detailní specifikace včetně názvu, typu, viditelnosti a nepovinné implicitní hodnoty.
- Převod operací specifikovaných v analytické třídě na úplnou sadu metod.

Tvorba modelu návrhových tříd je proces, ke kterému se často vracíme z fáze implementace a model dále doladujeme. Proto dále budu popisovat pouze poslední verzi návrhového modelu tříd. Pro jednodušší znázornění, modely neobsahují metody nezbytně nutné k popsání funkčnosti aplikace.

3.4.3 Popis jednotlivých částí navrženého systému

Vytvářená aplikace je rozdělena na dvě části – jádro systému a připojené moduly. Všechny části systému jsou navrženy v aplikaci Enterprise Architect a naprogramovány v aplikaci Microsoft Visual Studio 2008.

Jádro systému

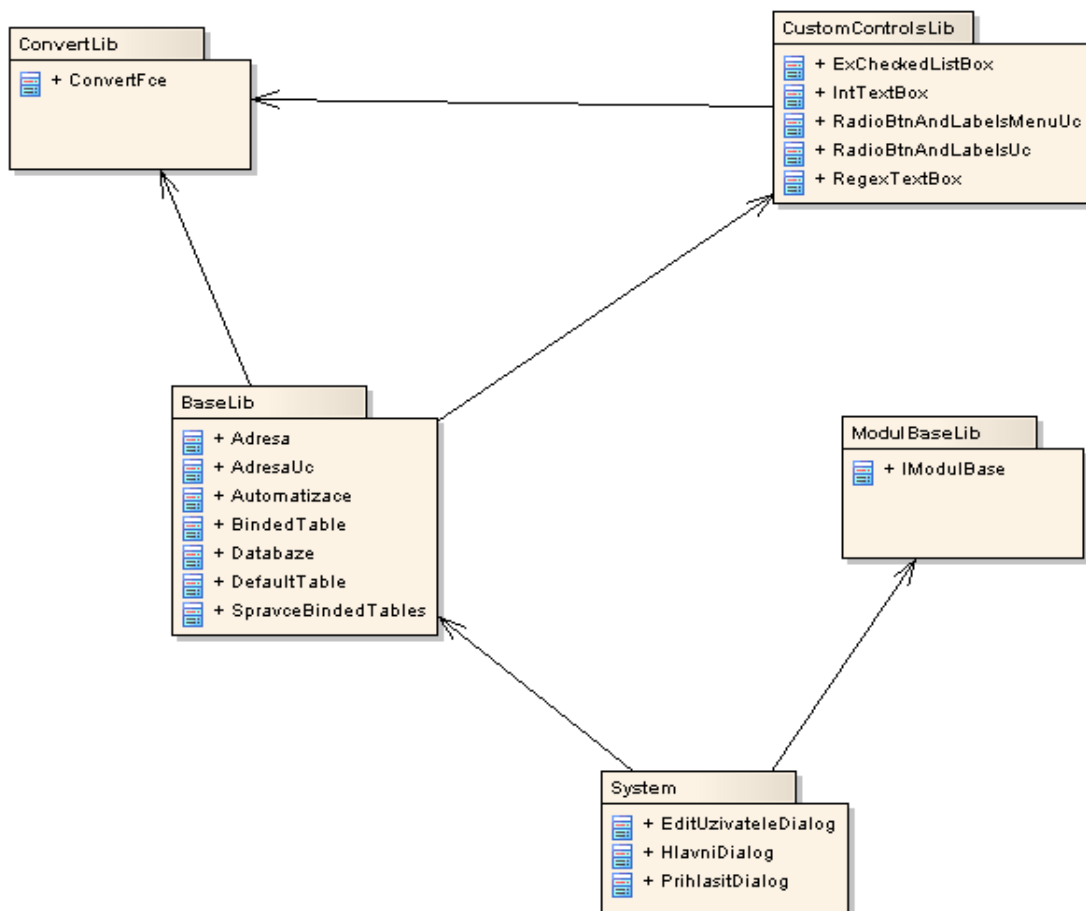
Jádrem vytvářeného systému je hlavní projekt a 4 pomocné knihovny, které lze vidět na obrázku 6. Hlavní projekt *System* je nejdůležitější částí projektu, který obsahuje základní formulář, kam se zobrazují načtené moduly.

Knihovna *ConvertLib* obsahuje třídu s metodami pro jednodušší práci s přetypováním proměnných.

Knihovna *CustomControlsLib* obsahuje rozšířené komponenty, použité v projektu.

Knihovnu *ModulBaseLib* tvoří jedno vytvořené rozhraní, které musí implementovat všechny načítané moduly.

Knihovna *BaseLib* obsahuje třídy pro uchovávání načtených dat z databáze, automatickou aktualizaci tabulek a třídu pro práci s adresou.



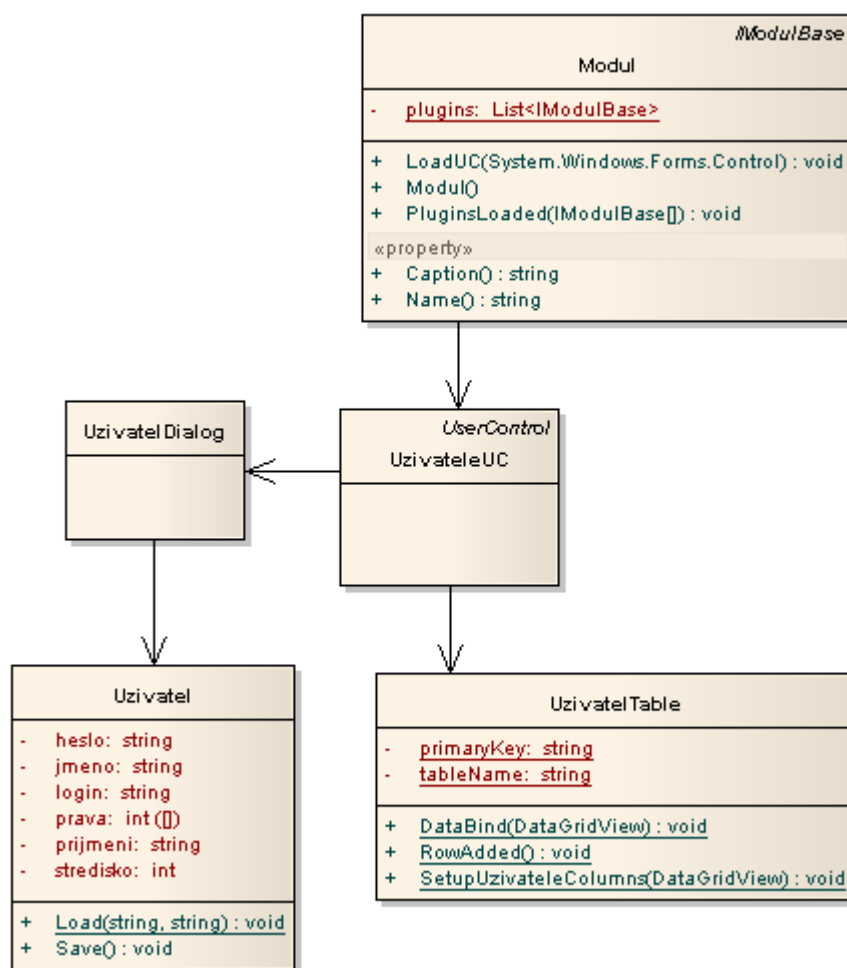
Obrázek 6: Základní rozdělení systému – návrh vytvořen v EA

Modul *Uživatelé*

Tento modul zastřešuje práci s uživatelskými účty v systému. Vypisuje tabulku uživatelských účtů a umožňuje přidání, nebo jeho změnu.

Třídy a vazby mezi sebou je možno vidět na obrázku 7. Kromě přístupové třídy *Modul* a třídy s definovaným vzhledem daného modulu *UzivateleUC* se vyskytuje třída zastřešující práci s daty o uživatelských účtech.

Modul *Uživatelé* dále obsahuje formulář pro přidání nového uživatele a třídu *Uzivatel*, která načte/uloží uživatele z/do databáze.



Obrázek 7: Modul uživatelé

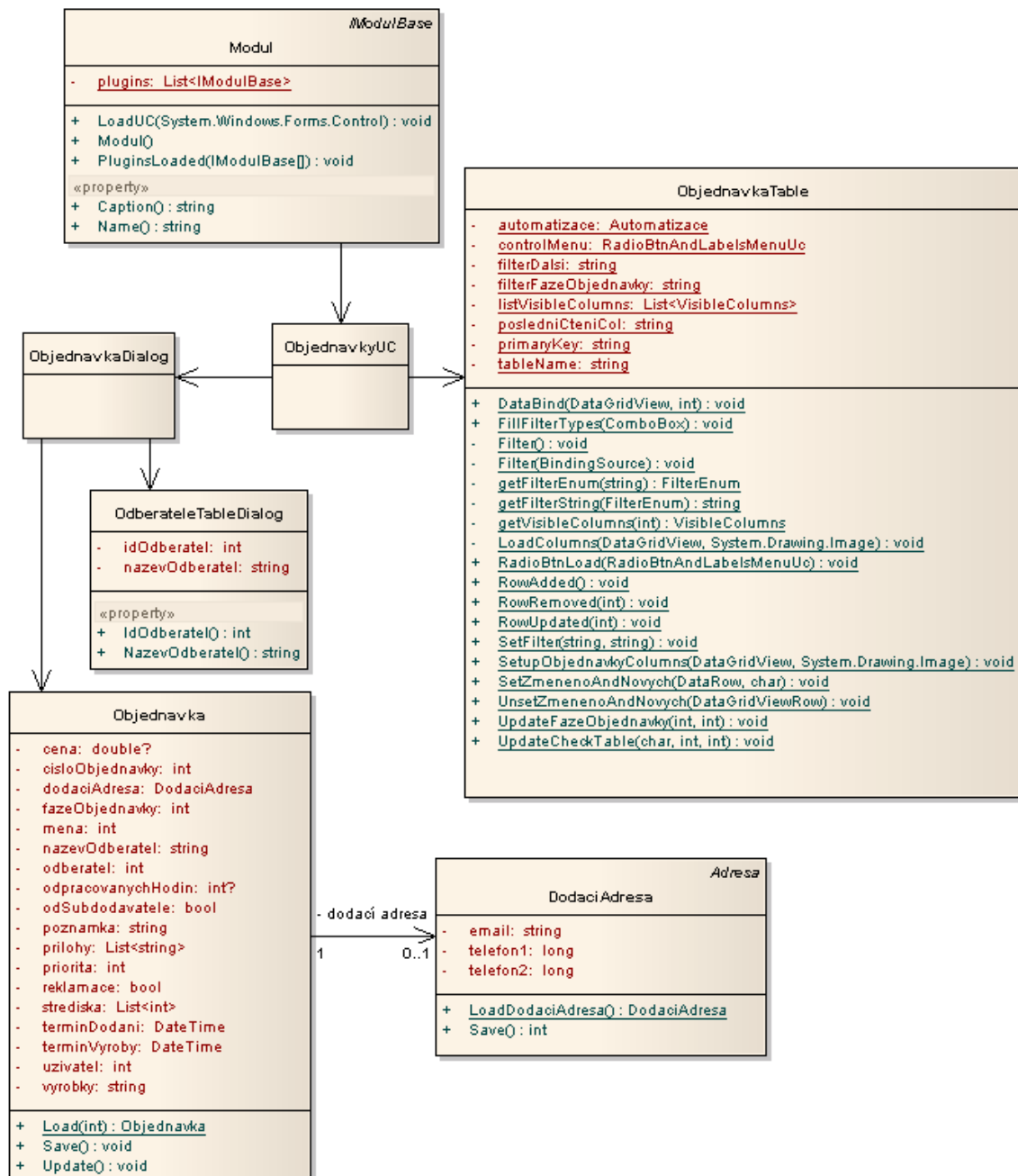
Modul Objednávky

Modul *Objednávky* rozšiřuje systém o správu objednávek. Umožňuje přidávat, editovat objednávky, filtrovat je a třídit dle zadaných kritérií.

Diagram modulu *Objednávky* je zobrazen na obrázku 8. Kromě přístupové třídy *Modul*, diagram obsahuje třídu s definovaným vzhledem modulu *ObjednávkyUC*. V této třídě se veškerá práce nad zobrazenou tabulkou zpracovává třídou *ObjednávkaTable*, která zastřešuje práci s načtenými daty o uživatelích.

Pro zobrazení detailu o objednávce, nebo pro zadání nové je vytvořen formulář *ObjednavkaDialog*. Ve formuláři je výběr odběratele realizován pomocí dalšího formuláře *OdberateleTableDialog*, který zobrazí seznam všech odběratelů v systému a vrací vybrané identifikační číslo a název odběratele.

Třída *Objednavka*, kromě výčtu všech potřebných atributů, obsahuje metody pro načtení, uložení a uložení změn u existující objednávky z/do databáze. Součástí třídy je třída *DodaciAdresa*, která rozšiřuje základní třídu *Adresa* v knihovně *BaseLib*. Tato třída je vytvořena, pokud je objednávka zaslána na jinou adresu, než fakturační.



Obrázek 8: Modul objednávky

Modul Odběratelé

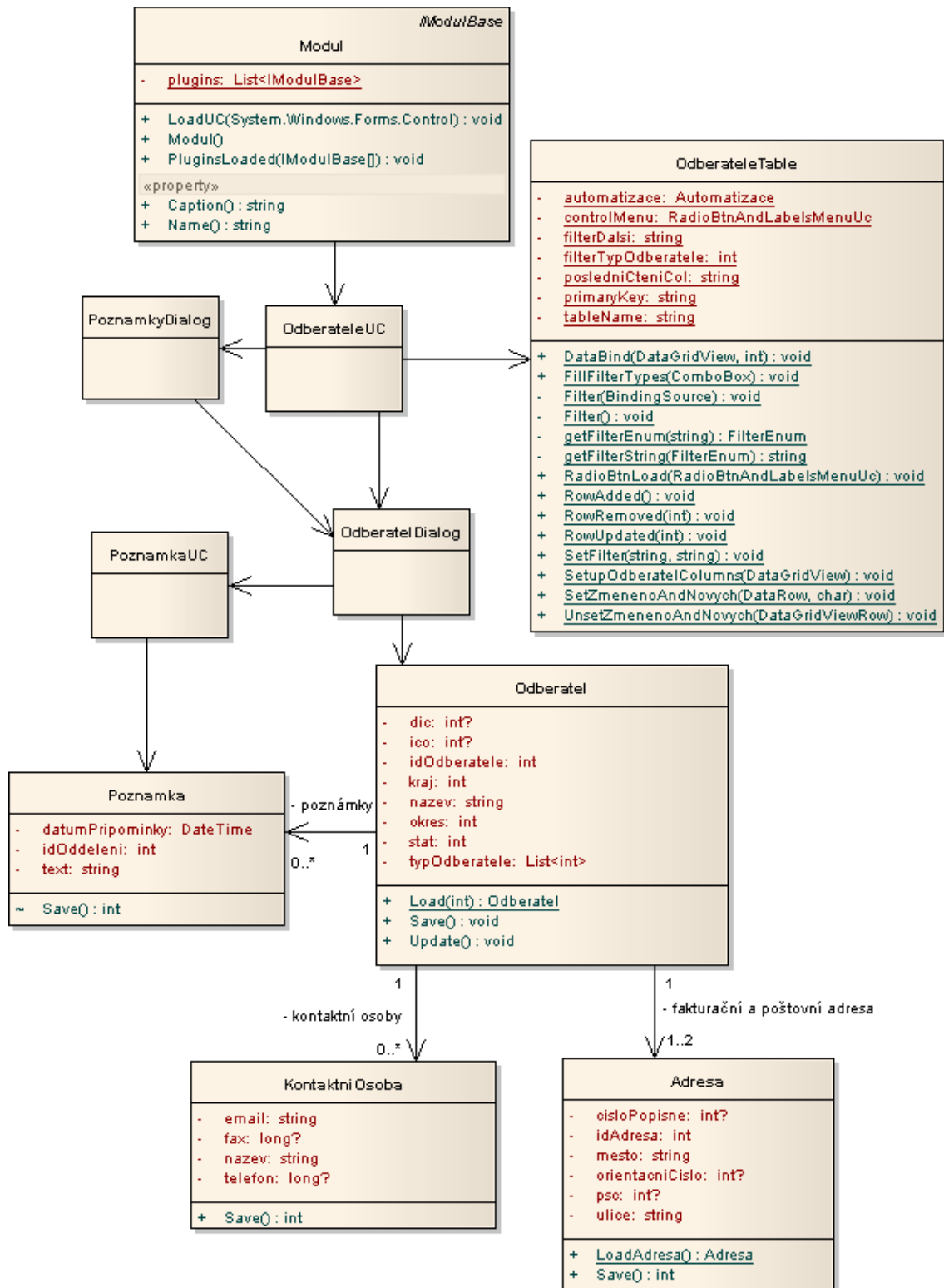
Modul *Odběratelé* rozšiřuje systém o správu odběratelů. Do funkcí pro práci s odběrateli patří přidávání/editace odběratelů pomocí zadávacího formuláře, filtrace a třídění odběratelů. Dále umožňuje nastavení připomínky u odběratele pro každé oddělení.

Diagram modulu *Odběratelé* znázorňuje obrázek 9. Jako všechny předchozí moduly obsahuje přístupovou třídu *Modul* a definici vzhledu ve třídě *OdberateleUC*. Veškerá práce s tabulkou odběratelů se provádí pomocí třídy *OdberateleTable*.

Diagram opět obsahuje formulář pro zobrazení, editaci a vytvoření nového uživatele, pojmenovaný jako *OdberatelDialog*. Pro jednodušší zobrazení připomínek pro všechna oddělení je vytvořena pomocná třída *PoznamkaUC*, která definuje vzhled pro poznámku v jednom oddělení.

Třída *Odběratel* umožňuje načtení, uložení a uložení změn u existujícího odběratele z/do databáze. K uložení složitějších datových typů, obsahuje 3 pomocné třídy - *Poznamka*, *KontaktniOsoba* a *Adresa*.

Poslední třída *PoznamkyDialog* zobrazuje seznam poznámek, zadaných daným oddělením na aktuální den.



Obrázek 9: Modul Odběratelé

3.4.4 Realizace automatické synchronizace tabulek

Podmínkou pro funkční synchronizaci tabulek je nutno si v databázi vytvořit trigger na tabulce, kterou chceme synchronizovat. Daný trigger nám při změně dat musí zavolat funkci *table_changed*, která přidá záznam o změně do tabulky *zmena_tabulek_log*.

Další podmínkou je vytvořit sloupec v tabulce *Uzivatel*, kde se bude ukládat čas posledního načtení dat z příslušné tabulky.

Tabulka, kterou chceme synchronizovat musí obsahovat sloupec *uzivatel_zmeneno*, kam se musí ukládat id uživatele, který provedl změnu nad danou tabulkou.

Aktivace automatické synchronizace

Metoda pro automatickou synchronizaci se nachází ve třídě *Automatizace* obsažená v knihovně *BaseLib*. Nové vlákno, které bude kontrolovat změny v databázi, se vytvoří při zavolání konstruktoru třídy s parametry:

- *tableName* – název tabulky, která se má kontrolovat,
- *idUzivatel* – id přihlášeného uživatele,
- *funkce* – ukazatel na funkci, která zajistí označení změněného záznamu, s parametry změněný řádek a typ změny jaká nastala.
- *PosledniCteniCol* – název sloupce v tabulce *Uzivatel*, který uchovává čas posledního čtení dané tabulky z databáze.

Algoritmus synchronizace

Po zavolání konstruktoru se aktivuje nové vlákno, které se bude pouštět každou minutu a kontrolovat zda nastala změna v tabulce a v databázi.

Algoritmus je popsán slovně a také zobrazen na obrázku 12 jako sekvenční diagram.

Automatická synchronizace je prováděna spouštěním obslužné metody *changedRows* v pravidelných intervalech. Tato metoda načte záznamy z tabulky

zmena_tabulek_log uložených v databázi, kde je čas vytvoření záznamu menší, než čas ve sloupci *PosledniCteniCol* a název tabulky je roven hodnotě v proměnné *tableName*.

Načtené záznamy se uloží do seznamu změn, kam se uloží id primárního klíče tabulky a typ změny, který pro daný záznam nastal.

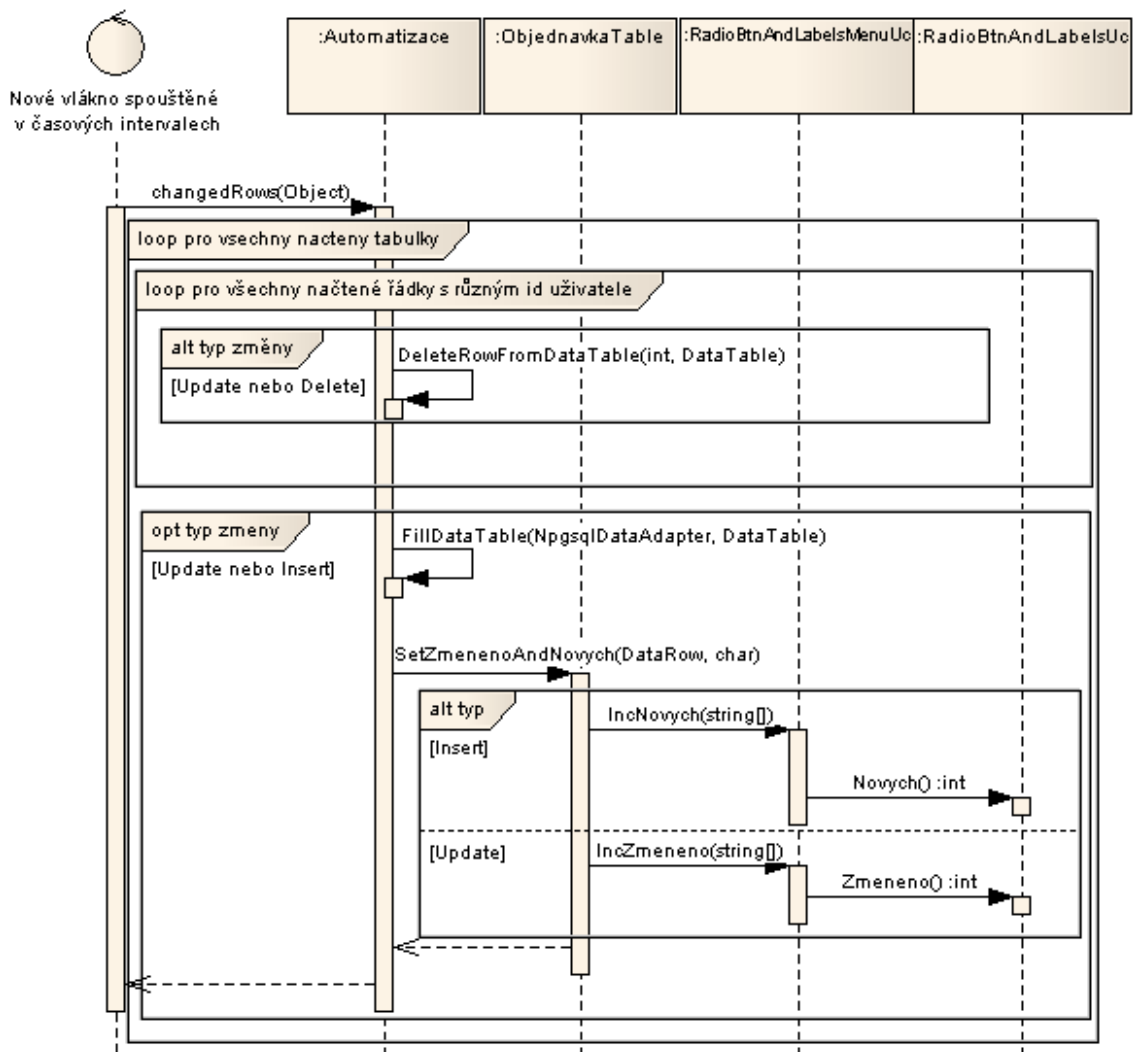
Uložený seznam se prochází a kontroluje metodou *zmenenoJinymUzivitelem*. Pokud byla změna provedena jiným uživatelem, pokračuje se zpracováním položky v seznamu.

Dále se vyhledají všechny řádky, které byly odebrány, nebo změněny a odeberou se z tabulky, uložené v aplikaci zavoláním metody *DeleteRowFromDataTable*. Pokud byly přidány nové řádky, nebo řádky změněny, tak se nové hodnoty načtou z databáze pomocí metody *FillDataTable*.

V další části algoritmus zobrazí kolik bylo načteno nových a kolik změněných záznamů zavoláním ukazatele na funkci *funkce* s parametry: vloženým, nebo změněným řádkem a jeho typem (vloženo nebo změněno). V metodě se nalezne typ odběratele pro tabulku odběratelů a fáze objednávky pro tabulku objednávky.

Tyto hodnoty se dále předávají třídě *RadioBtnAndLabelsMenuUC* zavoláním metody *IncNovych* (pro typ vloženo) nebo *IncZmeneno* (pro typ změněno).

V této třídě se nalezne odpovídající komponenta typu *RadioBtnAndLabelsUc* a předá se jí počet nových resp. změněných řádků.



Obrázek 10: Sekvenční diagram pro automatickou aktualizaci tabulek

3.4.5 Připojení modulu do systému

Načítání modulů do systému je realizováno vždy po spuštění systému a není možné je připojovat a odpojovat za běhu.

Pokud chceme připojit modul k vytvořenému systému, musí splňovat následující podmínky:

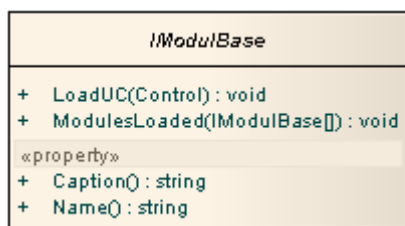
- modul se musí nacházet ve složce *Modules* vytvořené v adresáři programu včetně knihoven, které využívá ke své práci,
- napojovaný modul musí obsahovat třídu pojmenovanou *Modul*, která musí implementovat rozhraní *IModulBase* z knihovny *MudulBaseLib*.

Popis rozhraní *IModulBase*

Rozhraní *IModulBase* v knihovně *ModulBaseLib* udává jaké vlastnosti a metody musí dané třídy implementovat (obrázek 11).

Rozhraní vymezuje následující podmínky:

- metodu *LoadUC*, která má jako parametr rodičovský prvek, na který se napojí vizuální část daného modulu,
- metodu *ModulesLoaded*, která parametrem předává seznam všech načtených modulů,
- vlastnost *Name*, která vrací název modulu,
- vlastnost *Caption*, která vrací titulek daného modulu.



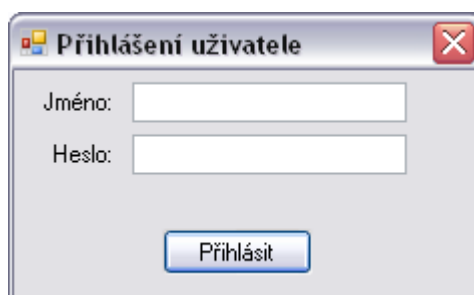
Obrázek 11: Rozhraní *IModulBase*

3.5 Administrace systému

Systém obsahuje 2 typy uživatelů – uživatel a administrátor. Uživatel má omezení přístupu do části uživatelé, která slouží pro administraci uživatelských účtů. Práce se systémem je jednoduchá a intuitivní, popsána v následujících kapitolách.

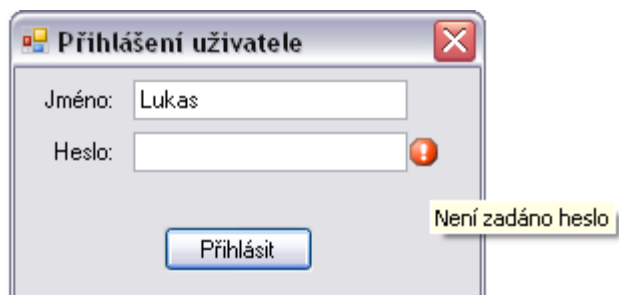
3.5.1 Přihlášení uživatele

Po spuštění aplikace jsou vyžádány přihlašovací údaje (obrázek 12). Kde uživatel zadá své přihlašovací jméno a heslo a potvrdí stiskem tlačítka *Přihlásit*.



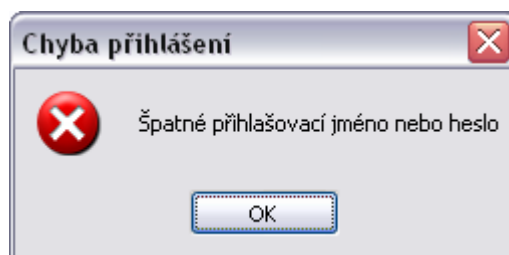
Obrázek 12: Okno pro přihlášení uživatele

Obě přihlašovací pole jsou povinná z důvodu autorizace (heslo) a autentizace (přihlašovací jméno). Pokud uživatel nezadá žádné jméno nebo heslo, je za polem zobrazen červený vykřičník, který informuje, že je pole nevyplněno. Příklad špatného vyplnění formuláře je na obrázku 13.



Obrázek 13: Okno pro potvrzení přihlášení - chybí heslo

Pokud uživatel neexistuje a nebo je zadáno špatné heslo, zobrazí se chybové okno, jaké je vidět na obrázku 14. Po stisku tlačítka *OK* se uživatel vrátí na přihlašovací formulář, kde musí zadat své přihlašovací údaje.



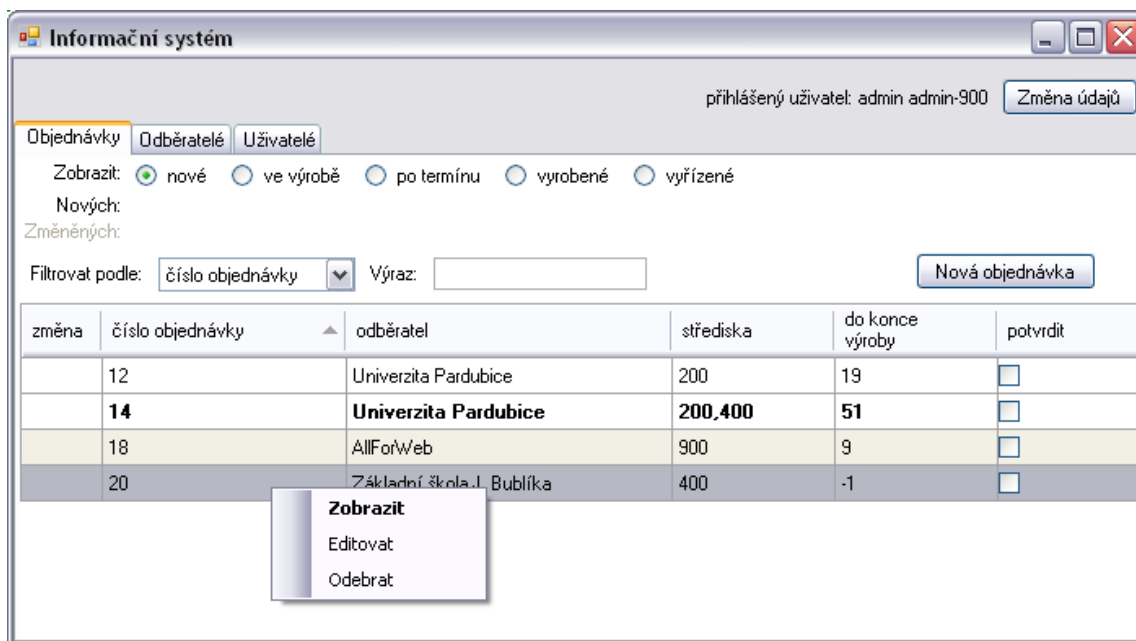
Obrázek 14: Okno pro potvrzení chyby při přihlášení

Po úspěšném přihlášení se uživatel dostane do hlavní části programu.

3.5.2 Hlavní program

Vzhled okna hlavního programu je zobrazen na obrázku 16. Skládá se z horního panelu, kde je zobrazen název aktuálně přihlášeného uživatele a tlačítka, které zobrazí formulář pro změnu údajů o uživateli.

Zbytek okna zabírají 3 záložky – *Objednávky*, *Odběratelé* a *Uživatelé*.



Obrázek 15: Hlavní okno programu

3.5.3 Záložka Objednávky

Na této záložce se zobrazují všechny objednávky, uložené v systému a rozdělené do 5 kategorií podle toho, v jakém stavu se daná objednávka nachází.

Aktuálně jsou v systému pro objednávky nastaveny následující stavy:

- *nové* – objednávky, které jsou vloženy do systému, ale ještě nejsou kompletně zadány a nebo čekají na potvrzení, že se daná objednávka začne vyrábět,
- *ve výrobě* – sem spadají objednávky, které se aktuálně vyrábějí a není překročen termín výroby,
- *po termínu* – obsahuje objednávky, které se vyrábějí, ale už se překročil termín výroby,
- *vyrobené* – zde jsou vyrobené objednávky, které čekají na sestavení, odeslání nebo distribuci k zákazníkovi,
- *vyřízené* – objednávky dodané k zákazníkovi, objednávka je považována za vyřízenou.

Pro přesouvání objednávek do další fáze se používají zaškrtačací prvky v tabulce. Například na obrázku 15 se po zaškrtnutí prvku v sloupci *potvrdit*, přesune

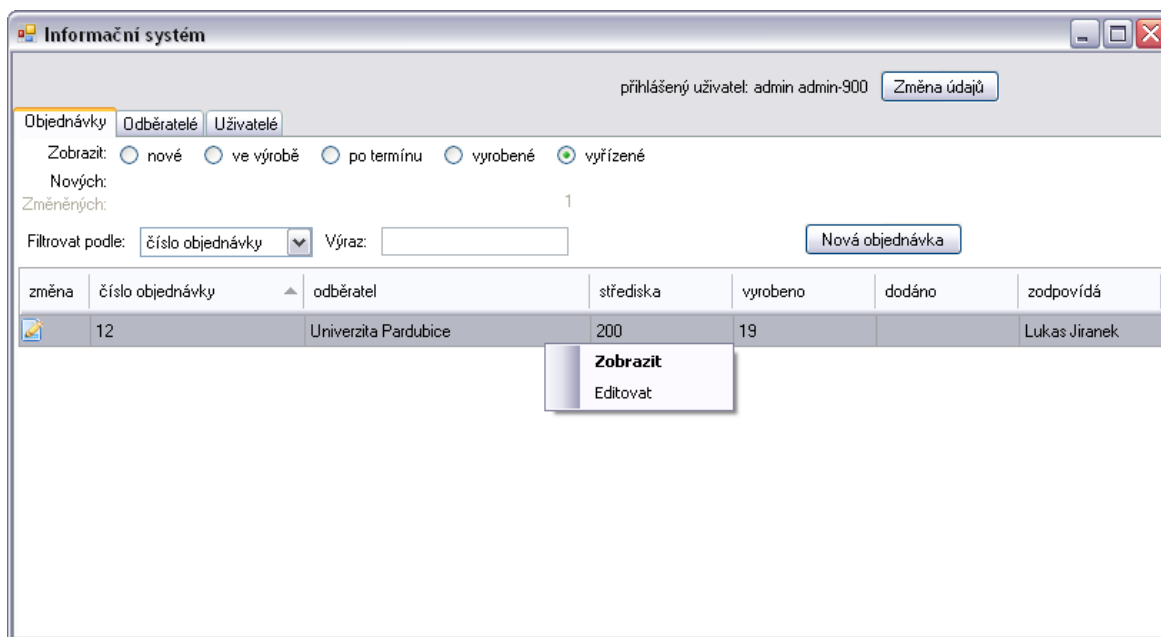
objednávka do fáze *ve výrobě*, resp. později do *po termínu*, když je překročena výrobní doba. Pro přesun do dalších fází jsou vytvořeny sloupce:

- *vyrobena*, který se nachází v tabulkách *ve výrobě* a *po termínu*,
- *dodáno* obsahuje tabulka *vyrobena*.

Popis sloupců v tabulce:

- *změna* – zobrazuje, zda byl daný řádek přidán, nebo změněn jiným uživatelem (po zobrazení detailu objednávky se pole vyprázdní),
- *číslo zakázky* – číslo zakázky vyplněné při vytváření objednávky,
- *odběratel* – název odběratele spojeného s danou objednávkou,
- *střediska* – výpis středisek, na která se vztahuje daná objednávka,
- *do konce výroby* – vypisuje počet dní zbývajících k dodržení termínu výroby,
- *vyrobena* – zobrazuje zaškrťovací pole, potvrzující vyrobenou zakázku, nebo termín vyrobení zakázky daným oddělením, nebo zobrazuje počet kladných dnů před termínem resp. záporných dnů výroby po termínu,
- *dodáno* – zobrazuje zaškrťovací pole potvrzující dodanou zakázku. Pokud je objednávka již dodána, zobrazuje počet kladných dnů dodání před termínem, resp. záporných dnů dodání po termínu,
- *zodpovídá* – název uživatele, který danou objednávku vložil do systému,
- *potvrdit* – odsouhlasení nové objednávky, že se může začít výroba..

Rozdílnost sloupců pro různý stav objednávky je vidět na obrázcích 15 a 16.



Obrázek 16: Okno zobrazené objednávky

Zobrazení nových nebo změněných objednávek

Přidané nové resp. změněné objednávky neukazuje pouze ikona ve sloupci *změna*, ale také hodnoty vypsané pod výběrem zobrazení stav objednávky. Pokud v dané tabulce nenastala žádná změna nezobrazí se nic. Pokud se přidal nový řádek resp. změnil, zobrazí se počet nových resp. změněných řádků pod daným stavem objednávky. Příklad změn v tabulce objednávky je vidět na obrázku 16, kde je změněná objednávka v tabulce vyřízených objednávek.

Další funkce

Zobrazení detailních informací o objednávce lze realizovat pomocí dvojitého poklepáním myši na námi vybraný řádek, kde se otevře formulář pouze pro čtení s detailním popisem objednávky.

Tabulka také obsahuje vyskakovací menu, po stisknutí pravého tlačítka na libovolném řádku. Menu obsahuje následující 3 volby:

- *Zobrazit* – Vyvolá formulář s detailními informacemi jako při dvojitém poklepání myši.
- *Editovat* – Otevře formulář, ve kterém se mohou měnit data v objednávce oprávněným uživatelem.

- *Odebrat* – Tato funkce je přístupná pouze pokud se objednávka nachází ve stavu *nová*. V dalších fázích je funkce odebrání objednávky zakázána.

Pro snadnější práci s tabulkou objednávek je vytvořená filtrace řádků. Uživatel si může vybrat z možnosti filtrace podle čísla objednávky, nebo názvu odběratele a vyplní hledaný řetězec o min. délce 3 znaků. Další možností je seřadit sloupce a to jak sestupně, tak vzestupně.

Tabulka objednávek upozorňuje na objednávky s vyšší prioritou (text na řádku je tučně) a na reklamace (řádek je vybarven světle šedou barvou). Zvýraznění objednávek ukazuje obrázek 15.

Poslední nepopsaná část v záložce objednávek je tlačítko *Nová objednávka*, která otevře okno pro přidání nové objednávky.

Formulář pro přidání/změnu objednávky

Na obrázku 17 je náhled, jak vypadá vyplněný formulář objednávky. Povinné údaje pro úspěšné vložení do databáze jsou: číslo objednávky, odběratel a vybrané minimálně jedno středisko.

Pro výběr odběratele je vytvořeno tlačítko *Vyber odběratele*, které zobrazí seznam všech odběratelů v systému a uživatel si jen vybere. Název odběratele se zobrazí v poli Odběratel.

Pokud uživatel chce přidat jednu nebo více příloh stiskne tlačítko *Vybrat*, které zobrazí dialog pro procházení adresářů v počítači, z nichž si uživatel vybere přílohy, které chce připojit k objednávce. Po potvrzení přidání příloh, se v tabulce zobrazí seznam příloh vybraných uživatelem. Pro odebrání přílohy lze stisknout tlačítko křížku, které odebere danou přílohu z objednávky.

Ve spodní části formuláře jsou vytvořena tlačítka *Uložit objednávku* a *Zrušit*. Jak je z názvu patrné, první tlačítko uloží novou nebo změny u existující objednávky. Druhé tlačítko zavře formulář a vrátí se na hlavní okno programu aniž by data uložil.

Objednávka

Číslo zakázky: 23 4765

Odběratel: základní a mateřská škola

Středisko:

- 200 - Šumperk
- 400 - Oskava
- 900 - Obchod

Přiloha: C:\234765.pdf

Odpracovaných hodin: 130

- reklamace
- od subdodavatele

Priorita: normální

Termín výroby: 1. června 2009

Termín dodání: 12. června 2009

Cena: 25 000 Kč

Výrobky:

- 20x PC stůl
- 20x kovové židle typu E34

Dodací adresa:

Kontaktní osoba:

Telefon1:

Telefon2:

Email:

Ulice:

Popisné číslo:

Orientační číslo:

Město:

PSČ:

Poznámka:

Obrázek 17: Formulář pro zadání nové objednávky

3.5.4 Záložka Odběratelé

Na záložce odběratelé se zobrazují všichni odběratelé nacházející se aktuálně v systému. Náhled odběratelů je vidět na obrázku 18. Vzhled karty je vytvořen, aby vypadal stejně jako karta Objednávky, pro jednodušší a přehlednější orientaci v systému.

Odběratelé jsou aktuálně rozděleni do následujících kategorií:

- *mateřské školy,*
- *základní školy,*
- *střední školy,*
- *vysoké školy,*

- *ostatní*,
- *vše* – zobrazí všechny odběratele v systému.

Odběratelé mohou být zařazeny ve více kategoriích.

Popis sloupců:

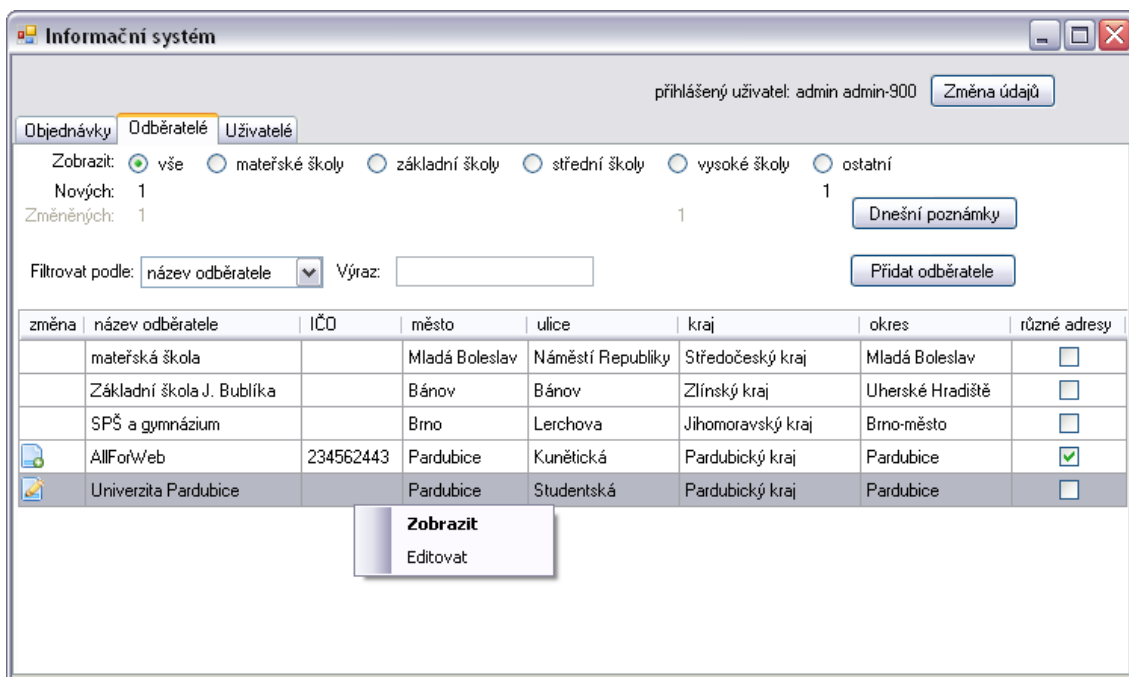
- *Změna* – zobrazuje, zda byl daný řádek přidán, nebo změněn jiným uživatelem. Po zobrazení detailu odběratele se pole vyprázdní.
- *Název* odběratele – vypisuje název odběratele, který byl vyplněn při jeho vytváření.
- *Město, ulice, kraj a okres* – ulice, kraj a okres odběratele.
- *Poštovní adresa* – pokud se liší od fakturační adresy.

Sloupce jsou všechny stejné, na rozdíl od tabulky objednávky.

Funkce na záložce odběratelé

Funkce na záložce odběratelé jsou totožné s funkcemi na kartě objednávek. Rozdíl je pouze ve filtrování řádků. Řádky lze filtrovat podle názvu odběratele, IČO, města.

Po stisknutí tlačítka *Přidat odběratele* se otevře okno pro přidání nového odběratele.



Obrázek 18: Okno výpisu odběratelů

Okno pro zobrazení připomínek

Při každém spuštění programu je zkontrolováno, zda nejsou nastaveny připomínky k odběratelům pro dané oddělení. Pokud je nastavena, pro aktuální den, alespoň jedna připomínka, zobrazí se okno s tabulkou připomínek (obrázek 19). Okno lze také vyvolat po stisknutí tlačítka *Dnešní poznámky*. Tabulka obsahuje údaje:

- název odběratele kterému byla připomínka nastavena,
- vlastní text poznámky,
- nastavený datum připomínky,
- tlačítko pro zobrazení detailu objednávky.



Obrázek 19: Okno s připomínkami

Formulář pro přidání/změnu odběratele

Na obrázku 20 je zobrazen příklad vyplněného formuláře. Mezi povinné údaje pro úspěšné uložení odběratele patří:

- typ odběratele – zde musí být vybrán minimálně jeden typ,
- název odběratele,
- fakturační adresa – povinné je zadat okres, kraj, město a číslo popisné.

Postup pro přidání nového kontaktu

Pro přidání nové kontaktní osoby se v rámečku *Kontaktní osoba* vyplní údaje a stiskne tlačítko *Přidat kontakt*. Při přidání nové kontaktní osoby je povinné zadat název a minimálně jeden kontakt (telefon, fax nebo email) a vyplnit správně formát telefonního čísla včetně státní předvolby. Po správném vyplnění údajů se kontaktní osoba přidá do tabulky vpravo od kontaktní osoby. Pokud chce uživatel kontaktní osobu odebrat, stiskne červený křížek vedle daného odběratele a potvrdí dotazovací dialog, zda si opravdu přeje kontakt odebrat.

Postup pro vyplnění poštovní adresy

Pro vyplnění poštovní adresy je vytvořen blok pro vložení poštovní adresy. Po vyplnění řádku v poštovní adresa, se automaticky odškrtně prvek *Poštovní adresa je stejná jako fakturační*. Stejně funguje i naopak. Pokud jsou všechna pole prázdná, auto-

matically se zaškrtně. Při ukládání odběratele, program automaticky přehlíží vyplněné údaje v poštovní adrese, pokud je zaškrtnuto *Poštovní adresa je stejná jako fakturační*.

Vyplnění poznámky

Pod záložkou s názvem oddělení jsou obsaženy vložené poznámky jednotlivých oddělení. Každý uživatel smí měnit údaje pouze v poznámce pro své oddělení a ostatní poznámky může pouze prohlížet. Poznámka obsahuje pole pro vložení textu poznámky, pole pro aktivaci připomenutí a kalendář, kde se zvolí datum připomenutí.

Odběratel

Typ organizace: základní školy
 střední školy
 vysoké školy
 mateřské školy
 ostatní

Název organizace: 6. základní škola
IČO: 5678345
DIČ: 4345456

Kontaktní osoba
Název:
Telefon:
Fax:
Email:
Přidat kontakt

název	telefon	fax	email	
ředitel	+420 765 456 897		reditel@zs_albre...	✖
zástupce			zastupce@zs_al...	✖

Fakturační adresa
Stát: Česká republika
Okres: Karviná
Kraj: Moravskoslezský kraj
Ulice: Školní
Popisné číslo: 120
Orientační číslo:
Město: Albrechtice
PSČ: 735 43

Poštovní adresa je stejná jako fakturační

Poštovní adresa
Ulice:
Popisné číslo:
Orientační číslo:
Město:
PSČ:

200 - Šumperk 400 - Oskava 900 - Obchod

Poznámka:
20. 8. 2009 Poslat nový katalog

Připomenout:
20. srpna 2009

Uložit odběratele Zrušit

Obrázek 20: Formulář pro přidání/změnu odběratele

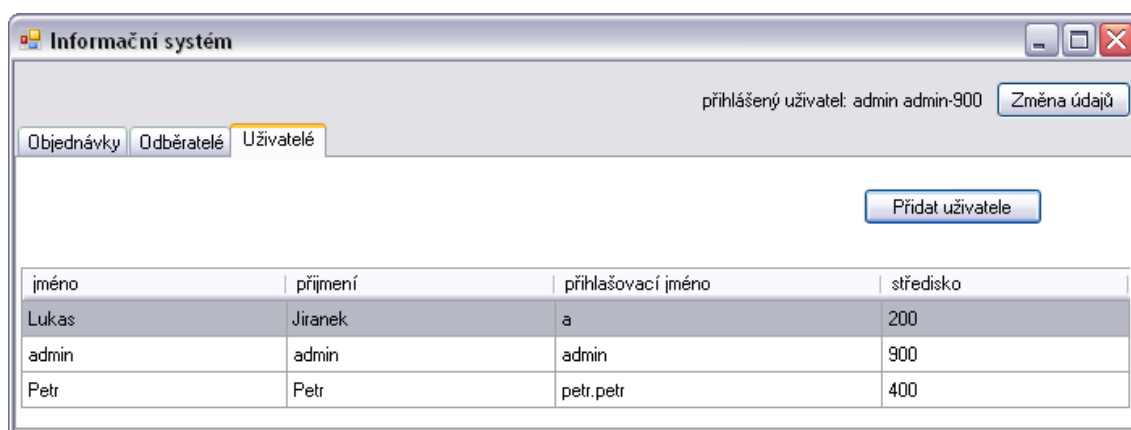
3.5.5 Záložka Uživatelé

Poslední záložkou v hlavním okně je záložka uživatelé. Tato část je přístupná pouze pro účet superuživatele, který může spravovat všechny uživatelské účty v systému.

Na záložce jsou vypsány všechny uživatelské účty, které mohou přistupovat do systému. Náhled výpisu uživatelských účtů je na obrázku 21.

V seznamu všech uživatelů je vypsáno jejich jméno, příjmení, jméno pod kterým se přihlašují do aplikace a středisko ve kterém pracují.

Jedinou funkcí na této kartě je přidání nového uživatelského účtu. To je realizováno po stisknutí tlačítka *Přidat uživatele*.



Obrázek 21: Okno výpisu uživatelů

Formulář pro přidání uživatele

Pro přidání nového uživatele slouží formulář zobrazený na obrázku 22. Zde se vyplní údaje o novém uživateli, nastaví se heslo, které si může později změnit a práva pro přístup k funkcím v systému.

Uživatel může mít následující přístupová práva:

- vkládat objednávky a odběratele,
- editovat všechny objednávky a odběratele,
- editovat své objednávky,

- stornovat objednávky,
- stornovat své objednávky,
- přístup k reportům všech oddělení,
- přístup k reportům svého oddělení,
- měnit stav objednávky (nová, vyrobená, dodaná),
- vkládat objednávky od externích dodavatelů.

The screenshot shows a window titled "Uživatel" with the following fields and options:

- Jméno: Lukáš
- Příjmení: Jiránek
- Středisko: 400 - Oskava
- Přihlašovací jméno: lukas
- Heslo: masked
- Heslo znovu: masked
- Práva:
 - vkládat objednávky a odběratele
 - editovat všechny objednávky a odběratele
 - editovat své objednávky
 - stornovat objednávky
 - stornovat své objednávky
 - přístup k reportům všech oddělení
 - přístup k reportům svého oddělení
 - měnit stav objednávky (nová, vyrobená, d...)

Buttons at the bottom: Přidat uživatele, Zrušit

Obrázek 22: Formulář pro přidání nového uživatele

3.6 Nasazení IS

V této kapitole je popsáno, jak nasadit dodaný informační systém do firemního informačního prostředí. Proces implementace (nasazení) systému probíhá v jednotlivých krocích:

- nastavení databáze,
- instalace,

- spuštění ve zkušebním provozu,
- migrace databázemi
- školení uživatelům
- ostrý provoz.

Nastavení databáze

Pro nasazení daného systému do provozu je zapotřebí mít na serveru nainstalovaný databázový systém PostgreSQL. Zde se musí ručně vytvořit databáze a uživatelský účet, který bude moci spravovat vytvořenou databázi.

Pomocí vzdáleného připojení do konzole databáze, nebo přes webovou aplikaci *phppgadmin* se na databázovém systému spustí dodaný skript *IS_Postgres.sql*, který vytvoří potřebné tabulky v databázi, triggery, funkce, indexy, sekvence a administrátorský účet pro přístup do systému.

Instalace informačního systému

Instalace samotného systému na uživatelův počítač je realizována pomocí dodaného instalačního balíku *ISSetup.exe*, který spustí instalaci na uživatelském počítači. Dodaný systém ke spuštění vyžaduje mít nainstalovaný balíček .NET Framework verze minimálně 2.0. Pokud operační systém tento balíček neobsahuje, je stažen z Internetu, nainstalován a pak se teprve spustí samotná instalace dodaného systému. Po instalaci se na plochu v počítači přidá zástupce programu, kterým se daný systém spouští.

Zkušební provoz

Nainstalovaný a nastavený systém poběží ve zkušebním provozu, kde se bude testovat jeho správná a bezchybná funkčnost. Bude vybrán určitý okruh zaměstnanců, kteří budou s dodaným systémem pracovat a testovat jeho bezchybný provoz. Systém se bude testovat nejprve bez jakýchkoliv nahraných dat. Po otestování základní funkčnosti se do systému napojí firemní data.

Migrace dat

Před ostrým nasazením systému do provozu se databáze musí naplnit daty, která firma získala z předchozího informačního systému. Jelikož se struktura nové a staré databáze liší musí se data zanalyzovat. Poté se připraví skript, který vyexportovaná data (ve formátu SQL) přenesou do nové databáze ve správném formátu.

Zaškolení uživatelů

Před spuštěním systému do ostrého provozu, je nutno zaškolit všechny uživatele, kteří budou systémem používat. Uživatelé se musí seznámit s jeho funkcí, aby ho mohli využívat v jeho maximální míře a tím jim napomáhal ke zrychlení firemních procesů.

Ostrý provoz

Ostrý provoz je poslední fází nasazování systému do firemního prostředí. Systém je úspěšně dokončen a může se začít používat v běžném provozu. Dochází k plnému zatížení systému.

4 Závěr

V rámci diplomové práce byl vyvinut informační systém pro automatickou správu a informační podporu procesu vyřízení zakázky (od přijetí objednávky po dodání výrobku). Informační systém by měl firmě zvýšit efektivitu v procesu řízení zakázek konkrétními funkcemi a nahradit v současné době nevyhovující řešení ve firmě zadavatele.

Při vývoji systému byl kladen důraz na jeho budoucí rozšiřitelnost, prostřednictvím možnosti přidávat další moduly pro další požadované funkce. Aktuálně systém obsahuje moduly pro správu objednávek, odběratelů a uživatelů.

V současné době je informační systém funkční podle požadavků zadavatele a probíhá fáze testování pro zajištění bezchybného budoucího chodu systému. Systém v této chvíli neobsahuje omezení uživatelských účtů v přístupu do jednotlivých modulů, role uživatelů jsou pouze zatím dvě (administrátor a uživatel), ale do budoucna mohou být přidávány další. Plánovaná je také integrace s dalšími informačními systémy v podniku.

5 Literatura

- [1] ŠMÍD, Vladimír. Pojem informačního systému. Management informačního systému [online]. 2002 [cit. 2009-05-20]. Dostupný z WWW: <<http://www.fi.-muni.cz/~smid/mis-infsys.htm>>.
- [2] HASALÍK, Radim. Proč si vybrat podnikový informační systém [online]. 2008 [cit. 2009-05-20]. Dostupný z WWW: <<http://informacni-system-erp.cz/proc-podnikovy-informacni-system.html>>.
- [3] GÁLA, Libor, POUR, Jan, TOMAN, Prokop. Podniková informatika. Mušálek Petr. 1. vyd. Praha : Grada Publishing a.s., 2006. 484 s. Management v informační společnosti. ISBN 80-247-1278-4.
- [4] Customer relationship management [online]. 2007 , 27. 4. 2009 [cit. 2009-05-20]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/CRM>>.
- [5] What is ECM [online]. 2008 [cit. 2009-05-20]. Dostupný z WWW: <<http://www.aiim.org/What-is-ECM-Enterprise-Content-Management.aspx>>.
- [6] Výhody ERP systému (Enterprise Resource Planning) [online]. 2008 [cit. 2009-05-20]. Dostupný z WWW: <<http://erpsystem.cz/erp-system/erp-system-enterprise-resource-planning/vyhody-erp-systemu.html>>.
- [7] SODOMKA, Petr. Analýza českého ERP trhu (1. část) [online]. 03.08.2004 [cit. 2009-05-20]. Dostupný z WWW: <<http://www.cvis.cz/hlavni.php?stranka=novinky/clanek.php&id=139>>.
- [8] Stručná historie systémů ERP. Hospodářské noviny [online]. 26.4.2006 [cit. 2009-05-20]. Dostupný z WWW: <http://hn.ihned.cz/c3-18324610-500000_d-strucna-historie-systemu-erp>
- [9] .NET [online]. 2007 , 21. 3. 2009 [cit. 2009-05-20]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/.NET>>.

- [10]NAGEL, Christian, et al. C# 2005 : Programujeme profesionálně. Odpovědný redaktor: Václav Kadlec; překlad: Jakub Mikulaščík, Petr Dokoupil. 1. vyd. Brno : Computer Press, 2006. 1400 s. ISBN 80-251-1181-4.
- [11]Technologie Java [online]. 1994-2009 [cit. 2009-05-20]. Dostupný z WWW: <<http://cz.sun.com/products/software/learnabout/java/>>.
- [12]Java (programovací jazyk) [online]. 2007 , 15. 5. 2009 [cit. 2009-05-20]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/Java_\(programovací_jazyk\)](http://cs.wikipedia.org/wiki/Java_(programovací_jazyk))>.
- [13]Teorie relačních databází: Relační model dat. Teorie DB [online]. 2006 [cit. 2009-05-20]. Dostupný z WWW: <<http://www.manualy.net/article.php?articleID=9>>.
- [14]MySQL [online]. 2007 , 15. 5. 2009 [cit. 2009-05-20]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Mysql>>.
- [15]PostgreSQL [online]. 2007 [cit. 2009-05-20]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/PostgreSQL>>.
- [16]Oracle [online]. 2007 , 9. 2. 2009 [cit. 2009-05-20]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Oracle>>.
- [17]ARLOW, Jim, NEUSTADT, Ila. UML2 : a unifikovaný proces vývoje aplikací. Brno : Computer Press, a.s., 2007. 567 s. ISBN 978-80-251-1503-9.

6 Seznam ilustrací

Obrázek 1: Případy užití – role v systému.....	21
Obrázek 2: Případy užití pro správu objednávek.....	22
Obrázek 3: Případy užití pro správu odběratelů.....	22
Obrázek 4: Analytický model.....	24
Obrázek 5: Databázový model – vytvořen v Case Studio.....	27
Obrázek 6: Základní rozdělení systému – návrh vytvořen v EA.....	29
Obrázek 7: Modul uživatelé.....	30
Obrázek 8: Modul objednávky.....	31
Obrázek 9: Modul Odběratelé.....	33
Obrázek 10: Sekvenční diagram pro automatickou aktualizaci tabulek.....	36
Obrázek 11: Rozhraní IModulBase.....	37
Obrázek 12: Okno pro přihlášení uživatele.....	37
Obrázek 13: Okno pro potvrzení přihlášení - chybí heslo.....	38
Obrázek 14: Okno pro potvrzení chyby při přihlášení.....	38
Obrázek 15: Hlavní okno programu.....	39
Obrázek 16: Okno zobrazené objednávky.....	41
Obrázek 17: Formulář pro zadání nové objednávky.....	43
Obrázek 18: Okno výpisu odběratelů.....	45
Obrázek 19: Okno s připomínkami.....	46
Obrázek 20: Formulář pro přidání/změnu odběratele.....	47
Obrázek 21: Okno výpisu uživatelů.....	48
Obrázek 22: Formulář pro přidání nového uživatele.....	49