Abstrakt

Práca je zameraná na návrh a implementáciu modulu informačného systému pre Poľnohospodárske družstvo vo Vrábľoch, ktoré pôsobí v oblasti rastlinnej a živočíšnej výroby. V prvej časti sú uvedené teoretické východiská, ktoré pojednávajú o riešenej problematike. Ďalej je analyzovaná aktuálna situácia podniku, nasledovaná kľúčovou časťou práce, samotným návrhom modulu informačného systému a rozanalyzovaním jeho konkrétnej implementácie, doplnené o ukážky užívateľského rozhrania.

Abstract

This thesis focuses on the design and implementation of module of information system for Agricultural Cooperative in Vrable, which operates in area of crop and livestock production. The first part of the thesis contains theoretical facts about main subject. In the next part current state of company is analyzed The key part of this thesis is design of particular module of information system and its implementation and design of user interface.

Kľúčové slová

Informačný systém, databáza, java, open source, poľnohospodárske družstvo, kataster

Key words

Information system, database, java, open source, agricultural cooperative, cadaster

Bibliografická citácia

SAKÁČ, M. *Návrh a implementácia modulu informačného systému pre Poľnohospodárske družstvo vo Vrábľoch.* Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2014. 17 s. Vedúci bakalárskej práce Ing. Jan Luhan, Ph.D.

Čestné prehlásenie

Prehlasujem, že predložená bakalárska práca je pôvodná a spracovaná samostatne. Prehlasujem, že citácia použitých prameňov je úplná, a že som vo svojej práci neporušil autorské práva (v zmysle Zákona č 121/2000 Sb., o práve autorskom a o právach súvisiacich s právom autorským).

V Brne, dňa 31. mája 2013

...............................................

Poďakovanie

Predovšetkým by som chcel poďakovať vedúcemu bakalárskej práce, pánovi Ing. Janovi Luhanovi Ph. D., za pomoc a cenné rady počas vypracovávania celej bakalárskej práce. Taktiež by som rád poďakoval Poľnohospodárskemu družstvu vo Vrábľoch za vynikajúcu spoluprácu a poskytnutie všetkých nevyhnutných materiálov.

1. Obsah

[Úvod 10](#_Toc382917444)

[Ciele práce 11](#_Toc382917445)

[1 Teoretické východiská 12](#_Toc382917446)

[1.1 Podnikový informačný systém 12](#_Toc382917447)

[1.1.1 Klasifikácia informačných systémov 12](#_Toc382917448)

[1.1.2 Spôsoby obstarania informačného systému 13](#_Toc382917449)

[1.2 Procesné riadenie 14](#_Toc382917450)

[1.3 UML 16](#_Toc382917451)

[1.3.1 Objekty a jazyk UML 16](#_Toc382917452)

[1.3.2 Význam a definícia požiadaviek 17](#_Toc382917453)

[1.3.3 Modelovanie prípadov užitia 17](#_Toc382917454)

[1.3.4 Návrhové triedy 18](#_Toc382917455)

[1.4 Návrh databáze 19](#_Toc382917456)

[1.4.1 Nájdenie faktov 19](#_Toc382917457)

[1.4.2 Entitne-relačné modelovanie 20](#_Toc382917458)

[1.4.3 Normalizácia 21](#_Toc382917459)

[1.4.4 Fáze návrhu databázy 22](#_Toc382917460)

[1.5 Použité technológie 23](#_Toc382917461)

[1.5.1 Java EE 23](#_Toc382917462)

[1.5.2 PostgreSQL 25](#_Toc382917463)

[1.5.3 HTML 25](#_Toc382917464)

[1.5.4 CSS 25](#_Toc382917465)

[1.5.5 JavaScript 26](#_Toc382917466)

[1.5.6 Použité návrhové vzory 26](#_Toc382917467)

[2 Analýza súčasného stavu 28](#_Toc382917468)

[3 Vlastné návrhy riešenia 29](#_Toc382917469)

[Záver 30](#_Toc382917470)

[Použitá literatúra 31](#_Toc382917471)

[Zoznam skratiek 32](#_Toc382917472)

[Zoznam obrázkov 32](#_Toc382917473)

[Zoznam tabuliek 33](#_Toc382917474)

[Zoznam príloh 33](#_Toc382917475)

# Úvod

# Ciele práce

Cieľom práce je návrh modulu informačného systému pre poľnohospodárske družstvo, jeho implementácia doplnená o nový prezentačný web podniku. Konkrétny modul bude evidovať majiteľov pôdy, od ktorých si družstvo pôdu prenajíma, katastrálne územia, ich parcely a listy vlastníctva, v ktorých daný vlastníci figurujú. Systém umožňuje užívateľom zhotovovať nájomné zmluvy, s potrebnými údajmi uloženými v databáze.

Výsledkom práce bude webová aplikácia, ktorá bude uložená na vybranom hostingu a dostupná z každého prehliadača pripojeného k sieti internet. Aplikácia bude autentizovať užívateľa a následne i autorizovať k náležitým úkonom v rámci informačného systému. Modul bude vyvíjaný tak, aby jeho funkcionalita vyhovovala aj iným poľnohospodárskym subjektom. Prínosom celej práce je uľahčenie manipulácie so štruktúrovanými dátami, ktorá doposiaľ prebiehala len pomocou textových procesorov, a tak bola náročná na personál a časovo neefektívna.

# Teoretické východiská

## Podnikový informačný systém

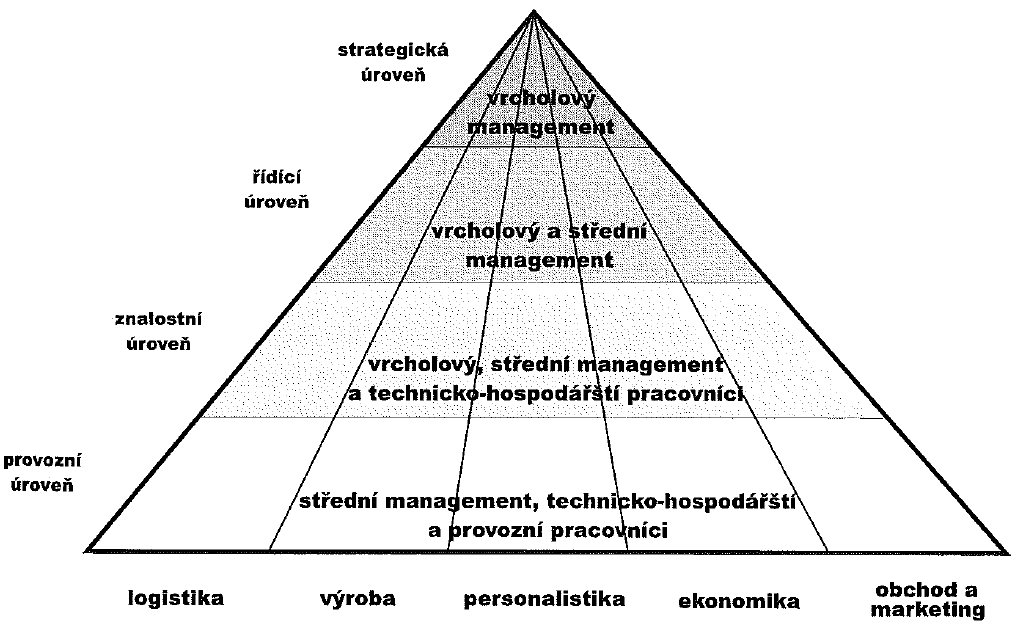
Pod týmto pojmom rozumieme systém, ktorý prostredníctvom stanovenej metodiky a dostupných technologických prostriedkov spracováva podnikové dáta a tvorí z nich znalostnú a informačnú bázu organizácie slúžiacu k riadeniu podnikových procesov, manažérskemu rozhodovaniu a správe podnikovej agendy (1).

### Klasifikácia informačných systémov

Jej hlavnou úlohou je charakterizovať hodnotu automatizovaného spracovania informácií pre pracovníkov na jednotlivých organizačných úrovniach. Jedná sa o nasledovné úrovne (1):

* **Prevádzková** – zaoberá sa spracovaním informácií týkajúcich sa rutín podnikovej agendy, ako napríklad realizácia výrobných zákaziek, nákupu a predaja, príjmu platieb apod. Informačné systémy pokrývajúce túto úroveň reagujú na plnenie každodennej činnosti a sledujú tok transakcií naprieč celou organizáciou. Odpovedajú na otázky typu: Je na sklade dostatok materiálu na montáž zákazky? Prebehla posledná finančná transakcia s našim hlavným dodávateľom (1)? Apod.
* **Znalostná** – zahrňuje klientske aplikácie podnikového informačného systému (ERP, CRM atď.), ale takisto prostriedky osobnej informatiky, ako sú napríklad kancelárske aplikácie, software určený pre tímovú kooperáciu (groupware) atď. Aplikácie tohto typu podporujú rast znalosti báze organizácie a riadia predovšetkým tok dokumentov. Umožňujú dohľadať odpovede na otázky typu: Aké sú výsledky z poslednej schôdzky s našimi dodávateľmi? Ako hospodári náš podnik (1)? Apod.
* **Riadiaca** – požaduje informácie nutné k plneniu administratívnych úloh a podpore rozhodovania. Informačný systém využívaný na tejto úrovni dáva odpoveď na dôležitú otázku: Fungujú veci tak ako majú? Odpovede poskytuje v podobe tzv. reportingu, generovanej výstupnej zostavy, obsahujúceho súhrn výsledkov z požadovanej oblasti (1).
* **Strategická** – je nápomocná vrcholovému managementu k identifikácií dlhodobých trendov vo vnútri, ale aj mimo organizácie. Ich primárnou úlohou je odhaliť očakávané zmeny a určiť, či je podnik schopný na zmenu zareagovať. K typickým otázkam, na ktoré informačné systémy poskytujú odpoveď, patrí napríklad: Aké sú dlhodobé trendy vo vývoji nákladov na produkciu v odvetví a ako s týmito nákladmi korešpondujú údaje z našej spoločnosti (1)?

Informácie pre riadiacu a strategickú úroveň pochádzajú nielen z prevádzkového systému podniku, ale takisto z externých zdrojov (1).



Obrázok 1: Informačná pyramída podľa organizačných úrovni podniku (Prevzaté z 1)

### Spôsoby obstarania informačného systému

Podľa preferencií konkrétnej organizácii môže podnik pristúpiť k obstaraniu IS nasledujúcimi spôsobmi:

* Rozvojom existujúcich softwarových riešení, ktoré by síce na jednej strane maximálne využíval už vynaložené investície, ale na druhej strane nezaručoval celkový efekt i plnenie všetkých možných budúcich požiadaviek podniku (3).
* Vývojom nového vlastného informačného systému, ktorý by síce na mieru zodpovedal novým požiadavkám a potrebám podniku a využíval aktuálny potenciál ICT, ale predstavoval by časovo i finančne náročnejšie riešenie, naviac spojené s rizikom malej garancie výsledného produktu. Otázkou je zaistenie jeho dlhodobého rozvoja i udržanie špecialistov v podniku (3).
* Nákupom hotového softwarového produktu parametrizovatelného na podmienky podniku, ktorý cez svoju počiatočnú vyššiu investíciu predstavuje rýchlejšie zavedenie, má garantovanú funkčnosť a ďalší rozvoj. Na druhej strane prinesie nové vzťahy medzi podnikom a dodávateľom takéhoto riešenia, potrebu integrovať ho s existujúcimi aplikáciami v podniku a v neposlednej rade vytvoriť závislosť na externej organizácii, spojenú napríklad i s novými právnymi aspektami (3).

## Procesné riadenie

Kľúčovú rolu pri implementácii najlepších praktík, a teda aj pri zlepšovaní podnikových procesov hrajú moderné informačné systémy. V prvom rade musí podnik prejsť transformáciou z funkčne orientovanej organizácie na procesne riadený podnik. Tato zmena je predmetom zavedenia procesného managementu. V takom prípade sa dá na organizáciu pozerať ako na súbor podnikových procesov, ktoré prestupujú jednotlivými oddeleniami a dodávajú svoje výstupy interným či externým zákazníkom podniku. Procesná organizácia sa snaží riadiť prácu ako celistvý proces, ktorý je rozložený na jednotlivé, vzájomne logicky previazané procesy (1).

Podľa ISO 9000 je proces definovaný takto (2):

*„Proces je súbor vzájomne súvisiacich alebo vzájomne pôsobiacich činností, ktoré premieňajú vstupy na výstupy.“*

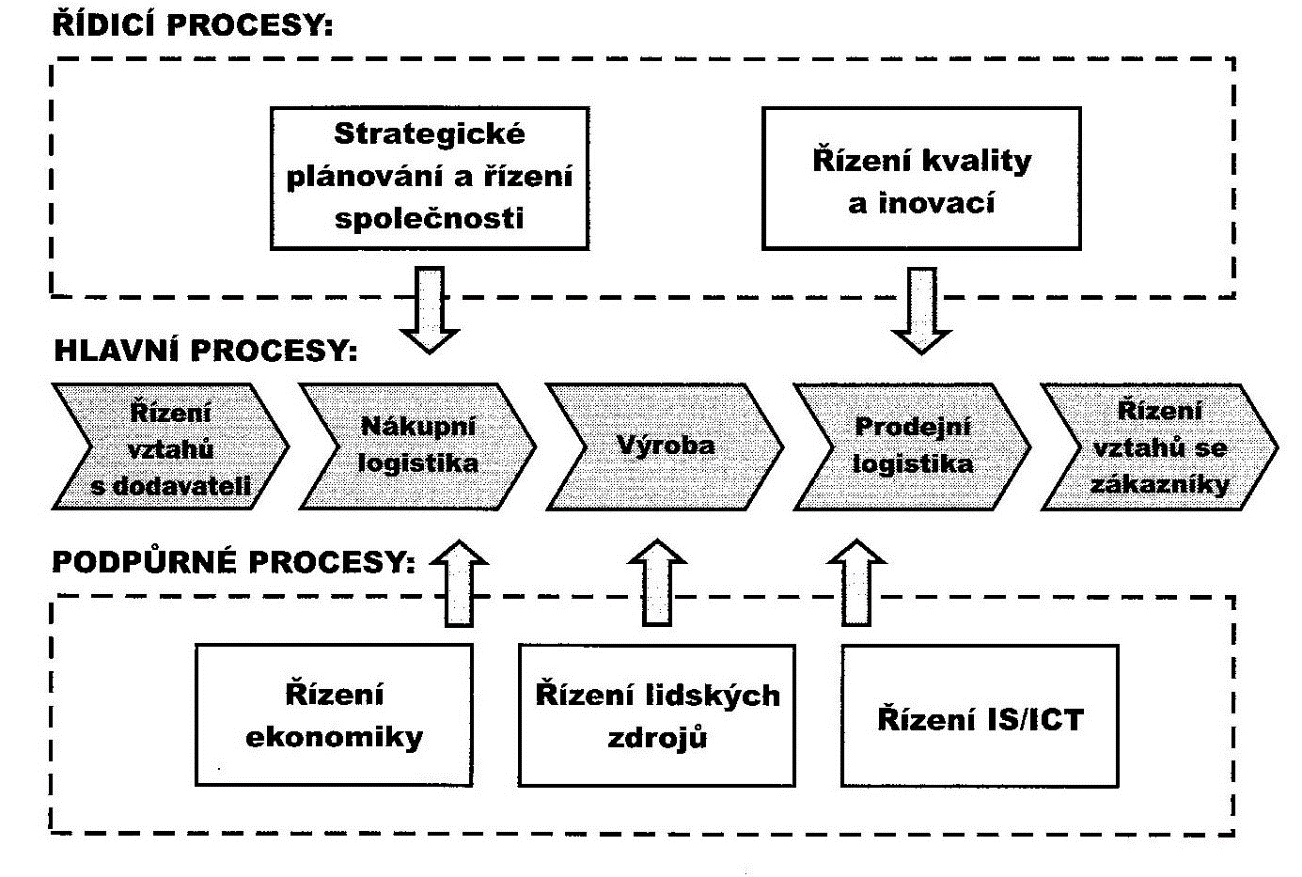
Musíme ale doplniť, že u zmienenej premeny vstupov na výstupy je podstatné vytváranie pridanej hodnoty pre zákazníka (1).

Proces má tieto základné charakteristiky:

* Je opakovateľný, pokiaľ je štandardizovaný
* Jeho výstupom je produkt alebo služba s pridanou hodnotou
* Je merateľný parametrami, akými sú kvalita, náklady, priebežná doba apod.
* Má svojho vlastníka – osobu alebo pracovný tím, ktorý má nad jeho fungovaním kontrolu a ktorý je zodpovedný za jeho prevádzku a zlepšovanie
* Má svojho zákazníka – či už interného alebo externého
* Je jednoznačne vymedzený jeho začiatok a koniec a nadväznosť na ďalšie procesy
* Využíva podnikové zdroje (finančné, hmotné, ľudské) (1)

Procesy možno rozdeliť na:

* **Riadiace** (strategické plánovanie, riadenie kvality a inovácií) – zabezpečujú rozvoj a riadenie výkonu spoločnosti a vytvárajú podmienky pre fungovanie ostatných procesov
* **Hlavné** (výroba, logistika, riadenie vzťahov so zákazníkmi) – vytvárajú hodnotu v podobe výrobku alebo služby pre externého zákazníka. Sú súčasťou hodnototvorného reťazca podniku.
* **Podporné** (ekonomika, riadenie ľudských zdrojov) – zaisťujú podmienky pre fungovanie ostatných procesov tým, že im dodávajú hmotné i nehmotné výstupy. Nie sú súčasťou hodnototvorného reťazca organizácie (1).



Obrázok 2: Hodnototvorný reťazec, riadiace a podporné procesy výrobného podniku (Prevzaté z 3)

## UML

Jazyk UML (Unified Modelling Language, unifikovaný modelovací jazyk) je univerzálny jazyk pre vizuálne modelovanie systémov. Najčastejšie je spojovaný s modelovaním objektovo orientovaných softwarových systémov, ale napriek tomu má oveľa širšie využitie, čo vyplýva z jeho zabudovaných rozširovacích mechanizmov. Jazyk UML poskytuje vizuálnu syntax, ktorú môžeme využiť pri zostavovaní svojich modelov (4).

### Objekty a jazyk UML

Základným predpokladom jazyka UML je skutočnosť, že umožňuje modelovanie softwaru, rovnako ako ďalších systémov ako kolekciu spolupracujúcich objektov. Tento prístup funguje rovnako spoľahlivo v obchodných a podnikateľských procesoch a aj ďalších aplikáciách (4).

### Význam a definícia požiadaviek

Nedostatočne špecifikované požiadavky a nedostatočne zapojenie užívateľov sú dve hlavné príčiny konečného neúspechu celého projektu. Obe príčiny sú zlyhaním v procese inžinierskych požiadaviek. Konečný softwarový systém je založený na množine požiadaviek, je ich efektívne inžinierstvo kľúčovým faktorom celého vývoja softwarového projektu (4).

Požiadavku možno definovať ako špecifikáciu toho, čo by malo byť implementované. Rozlišujeme dva typy požiadaviek:

* Funkčné – určujú, aké chovanie bude systém ponúkať.
* Nefunkčné – špecifikujú vlastnosti alebo obmedzujú podmienky daného systému (4).

Požiadavky sú základom všetkých systémov. Sú v podstate vyjadrením toho, čo by mal systém robiť, ale nie toho ako by to mal robiť (4).

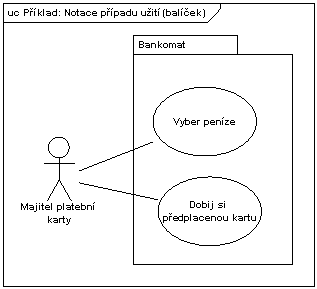
### Modelovanie prípadov užitia

Je jednou z foriem inžinierskych požiadaviek a zároveň doplnkovým spôsobom získavania požiadaviek. Skladá sa z nasledujúcich aktivít:

* Nájdenie hraníc systému
* Vyhľadanie aktérov
* Nájdenie prípadov užitia
  + špecifikácia prípadu užitia
  + určenie alternatívnych scenárov
* Tento postup je potrebné opakovať, pokiaľ nedôjde k ustáleniu prípadov užitia, aktérov a hraníc systému (4).

Výstupom uvedených aktivít je model prípadov užitia, ktorý poskytuje hlavný zdroj objektov a tried Tento model obsahuje štyri komponenty:

* Hranice systému – ohraničenie zobrazené okolo prípadu užitia.
* Aktéri – sú role, pridelené osobám alebo predmetom používajúcich daný systém.
* Prípady užitia – činnosti, ktoré môžu aktéri so systémom vykonávať.
* Relácie – zmysluplné vzťahy medzi aktérmi a prípadmi užitia (4).



Obrázok 3: Ukážka diagramu prípadu užitia (Prevzaté z 5)

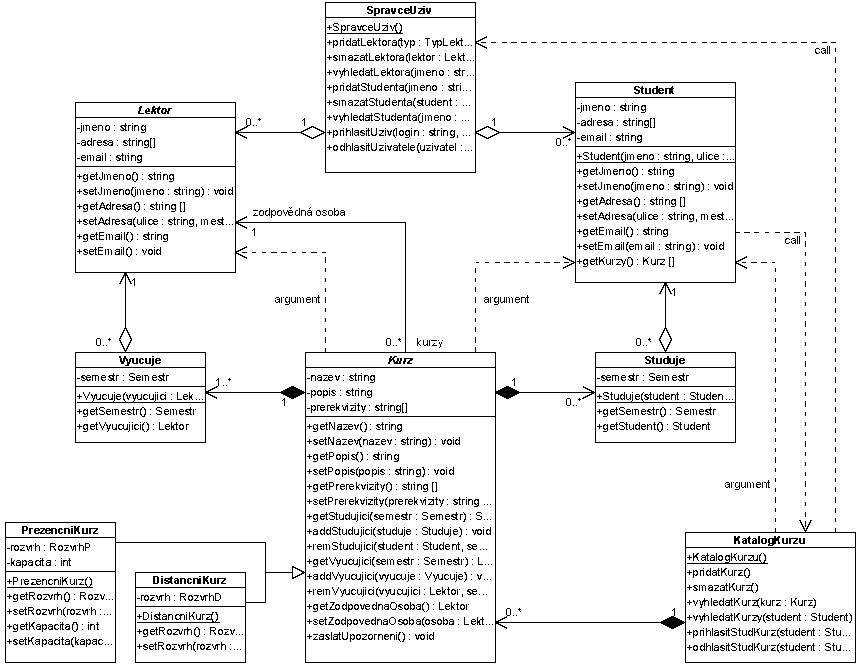
### Návrhové triedy

Sú triedy, ktorých špecifikácia je na takom stupni, že sú pripravené pre implementáciu.

Počas analýzy je zdrojom tried problémová doména. Je to množina požiadaviek, ktorá popisuje problém, ktorý sa snažíme vyriešiť. Zdrojom analytických tried môžu byť prípady užitia, špecifikácie sprievodných požiadaviek, slovníky pojmov a akékoľvek ďalšie súvisiace informácie (4).

Návrhové triedy sa dajú získať z dvoch zdrojov:

* Z problémovej domény prostredníctvom upresňovania analytických tried. Súčasťou upresnenia je aj doplňovanie implementačných detailov. V priebehu tejto činnosti sa častokrát stane, že je potrebné koncepčnú analytickú triedu rozbiť na niekoľko podrobných návrhových tried (4).
* Z domény riešenia. Táto doména poskytuje technické nástroje, ktoré umožňujú implementáciu systému (4).



Obrázok 4: Ukážka návrhového diagramu tried (Prevzaté z 6)

## Návrh databáze

### Nájdenie faktov

Táto etapa je kľúčová behom počiatočných fáz životného cyklu vývoja, vrátane fáz plánovania databáze, definície systému a zberu a analýza požiadaviek. V tejto fáze zisťujeme terminológiu, problémy, príležitosti, obmedzenia, požiadavky a priority organizácie a užívateľov systému. Nájdenie faktov sa používa nielen pri návrhu databáze, ale aj v neskorších častiach životného cyklu vývoja (7).

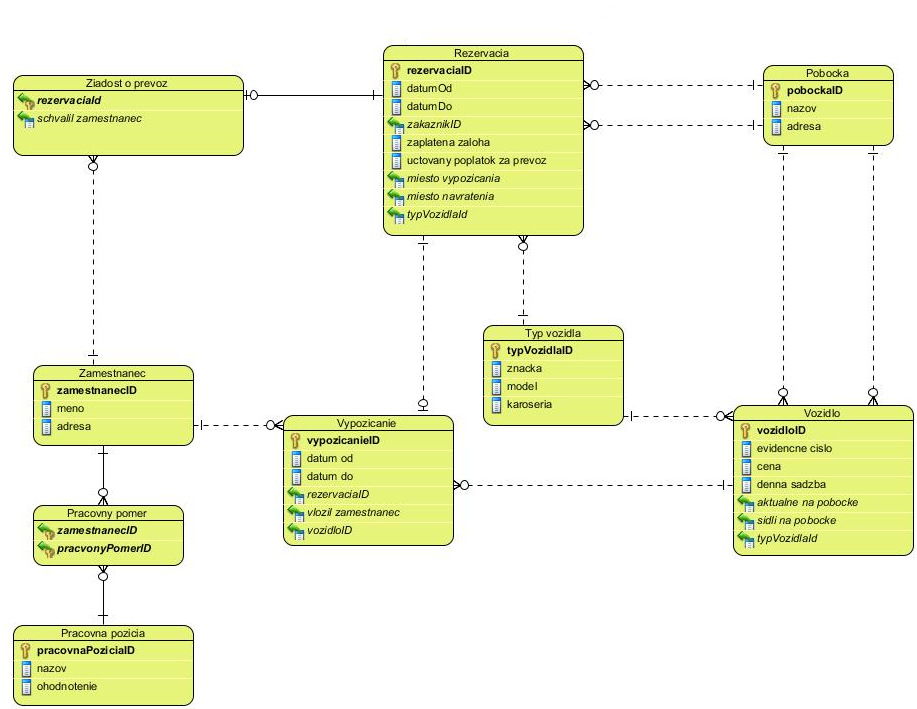
Zvyčajne sa počas jedného databázového projektu ako techniky zisťovanie faktov používajú viaceré prístupy, ako napríklad:

* Preskúmanie dokumentácie
* Rozhovor
* Pozorovanie organizácie v prevádzke
* Sekundárny výskum
* Dotazníky (7)

### Entitne-relačné modelovanie

Jedným z najnáročnejších aspektov návrhu databáze je skutočnosť, že návrhári, programátori a koncoví užívatelia vidia dáta rozdielnym spôsobom. Ak neexistuje spoločný pohľad na chod organizácie, vytvorený návrh nemôže splniť požiadavky užívateľov (7).

Entitne-relačné modelovanie návrhu databáze odpovedá prístupu k návrhu metódou zhora dole. ER modelovanie začína určením dôležitých dát (takisto nazývaných entity) a reláciami medzi dátami, ktoré je potrebné v modeli reprezentovať. Ďalej sa postupuje k podrobnostiam ako napríklad informáciám, ktoré je potreba o entitách a vzťahoch uchovávať (nazývané atribúty) a obmedzeniam platným pre entity, vzťahy a atribúty (7).



Obrázok 5: Ukážka ER diagramu (Vlastná tvorba)

### Normalizácia

Použitie normalizácie ako prístupu zdola nahor znamená analýzu súvislosti medzi atribútmi a potom zoskupenie atribútov na základe tejto analýzy do tabuliek, ktoré predstavujú entity a relácie. Tento prístup je však náročné použiť, keď je veľký počet atribútov, pretože potom je ťažké zaviesť všetky vzťahy medzi atribúty (7).

Oproti tomu existuje ešte metóda prístupu zhora nadol k návrhu databáze. Pri tomto prístupe používame ER modelovanie pre reprezentáciu hlavných entít, relácií a atribútov dát. ER model potom prevedieme na tabuľky (7).

#### Prvá normálna forma

Táto forma poníma o tom, že tabuľka, v ktorej každý priesečník stĺpca a záznamu obsahuje len jedinú hodnotu (7).

Inými slovami, relácia je v prvej normálnej forme, pokiaľ každý jej atribút obsahuje len atomické hodnoty. Teda hodnoty z pohľadu databáze sú už ďalej nedeliteľné. Napríklad v relácii obsahujúcej dáta o nejakej osobe budeme chcieť mať viac telefónnych čísel. Aby tabuľka bola v prvej normálne forme, musíme buď rozdeliť atribút telefón do viacerých atribútov (iba za predpokladu, že sa počet telefónnych čísel nezvýši), alebo oddeliť telefónne čísla do samostatnej tabuľky, čo je podstatne flexibilnejšie riešenie (8).

#### Druhá normálna forma

Druhá forma hovorí o tom, že tabuľka musí byť v prvej normálnej forme a zároveň sú hodnoty každého stĺpca, ktorý nie je súčasťou primárneho kľúča, determinovaný všetkými hodnotami stĺpcov, ktoré tvoria primárny kľúč (7).

Inak povedané, relácia sa nachádza v druhej normálnej forme, ak je v prvej normálnej forme a každý nekľúčový atribút je plne závislý na primárnom kľúči, a to na celom kľúči a nie len na jeho podmnožine. Z čoho vyplýva, že túto normálnu formu musíme riešiť iba v prípade, že máme viachodnotový primárny kľúč (8).

#### Tretia normálna forma

Hovorí, že tabuľka, ktorá je v 1. NF a 2.NF a v ktorej všetky hodnoty v stĺpcoch, ktoré nepatria k primárnemu kľúču, sú determinované iba stĺpcom primárneho kľúča a nie sú determinované žiadnymi inými stĺpcami (7).

Inak formulované, v tejto normálnej forme sa nachádza tabuľka, ak spĺňa predchádzajúce dve formy a žiadny z jej atribútov nie je tranzitívne závislý na primárnom kľúči. Čiže všetky jej nekľúčové atribúty sú navzájom nezávislé (8).

### Fáze návrhu databázy

Metodológia návrhu sa skladá zo štádií a krokov, ktoré vedú návrhára technikami zodpovedajúcimi každému štádiu projektu.

#### Konceptuálny návrh

Je proces vytvorenia modelu dát používaných v organizácii bez akýchkoľvek úvah o fyzickej implementácii. Vytvára sa konceptuálny model dát na základe dát používaných v organizácii bez rozpracovania podrobností, ako napríklad podkladový model alebo iných úvah o fyzickej implementácii (7).

Konceptuálny model dát identifikuje dôležité entity a relácie, ktoré je potrebné reprezentovať v databáze a je súčasne zdrojom informácií pre logickú fázu návrhu (7).

#### Logický návrh

Je proces vytvorenia modelu dát používaných organizáciou, ktorý je založený na špecifickom modeli dát, ale nezávislý na konkrétnom DBMS a iných úvahách o fyzickej implementácii (7).

V tejto fáze vytvárame logickú reprezentáciu databázy. Ako základ logického modelu je možné použiť relačný model dát, a preto reprezentujeme dôležité entity a relácie ako množinu relačných tabuliek. Logický návrh databázy je zdrojom informácii pre fázu fyzického návrhu (7).

#### Fyzický návrh

Jedná sa o proces vytvorenia popisu implementácie databázy vo vonkajšej pamäti; popisuje podkladové tabuľky. Organizáciu súborov, indexy používané pre dosiahnutie efektívneho prístupu k dátam, všetky súvisiace integritné obmedzenia (7).

## Použité technológie

### Java EE

Java platforma, Enterprise Edition (Java EE) je štandardom v komunitne riadenom podnikovom software. Java EE je vyvinutá s použitím Java Community Process, s príspevkami od odborníkov z oboru, komerčných a open source organizácií, skupín používateľov Javy a nespočetných jedincov. Každé vydanie integruje funkcie, ktoré sú zladené s potrebami priemyslu, zlepšuje prenositeľnosť aplikácií a zvyšuje produktivitu vývojárov. Dnes Java EE ponúka bohatú platformu pre vývoj podnikového softwaru (9).

#### Hibernate

Hibernate ORM (objektovo-relačné mapovanie) je knižnica pre jazyk Java, ktorá poskytuje rámec pre mapovanie objektových domén na entity relačnej databázy. Hibernate je bezplatný software, ktorý je voľne šírený pod licenciou GNU Lesser General Public Licence (10).

Jeho primárnou funkciou je mapovanie tried z programov napísaných v jazyku Java do databázových tabuliek, čo znamená premenu z Java dátových typov na typy jazyka SQL. Taktiež dokáže generovať SQL dotazy a tak uľahčuje vývojárom prácu so samotnou databázou. Jeho použitie takisto zabezpečuje vývojárom úplnú nezávislosť na konkrétnom databázovom systéme (10).

#### Spring

Je open source aplikačný rámec a takisto správca inverznej kontroly pre platformu Java. Základné rysy rámca môžu byť použité v akejkoľvek Java aplikácii, ale takisto existujú napr. rozšírenia pre vytváranie webových aplikácii. Hoci rámec nepredpisuje žiadny konkrétny programovací model, je veľmi populárny v Java komunite ako alternatíva k modelu EJB (Enterprise JavaBean) (11).

Spring pozostáva z veľkého počtu modulov, ktoré zabezpečujú širokú škálu funkcionality. Napr. Inversion of control container (konfiguruje aplikačné komponenty a manažuje životný cyklus Java objektov), Aspect-oriented programming, Transaction management a mnoho ďalších (11).

#### Stripes

Je open source webový aplikačný rámec založený na návrhovom vzore MVC(model-view-controller). Jeho cieľom je byť jednoduchší a prístupnejší ako konkurencia pomocou Java technológií ako sú napr. generiká a anotácie. Riadi sa heslom „Konvencia nad konfiguráciou“. V praxi to znamená, že Stripes aplikácie sotva potrebujú nejaké konfiguračné súbory, čo znižuje čas vývoja a neskôr prácu, ktorá je nevyhnutná na údržbu aplikácie (12).

### PostgreSQL

PostgreSQL je výkonný open source objektovo-relačný databázový systém. Je vydávaný pod tzv. PostgreSQL licenciou, ktorá je podobná MIT licencii, a tak je jeho šírenie voľné a bezplatné. Má za sebou viac ako 15 rokov aktívneho vývoja a osvedčenú architektúru, ktorá si zaslúži silnú povesť pre svoju spoľahlivosť, integritu dát a korektnosť. Dokáže bežať na všetkých hlavných operačných systémoch, vrátane Linuxu, UNIXových systémov (AIX, BSD, Mac OS X apod.) a Windowse. Má plnú podporu cudzích kľúčov, spojení tabuliek, pohľadov, triggerov a uložených procedúr a funkcií. Takisto umožňuje ukladanie veľkých binárnych objektov, vrátane obrázkov, zvukového záznamu, alebo videa. Taktiež má natívne programovacie rozhranie pre C/C++, Javu, .Net, Perl, Python, Ruby atď. (15)

### HTML

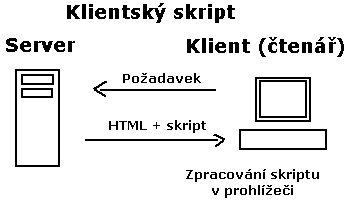
Hypertext Markup Language je sada značkovacích symbolov a pvkov vložených do súborov určených pre zobrazenie vo WWW (World Wide Web) prehliadači. Značky hovoria prehliadaču, ako má zobraziť obsah jednotlivých textových súborov. Každý značkovací prvok sa nazýva element, alebo tag. Niektoré prvky sa vyskytujú len v pároch, tým pádom prvý z dvojice slúži na označenie začiatku sekcie a druhý na označenie jej konca (16).

### CSS

Je kolekcia metód pre grafickú úpravu webových stránok. Skratka znamená Cascading Style Sheets, čo môžeme voľne preložiť ako “kaskádové štýly“. Kaskádové, pretože sa na seba môžu vrstviť definície štýlu, ale platí len tá posledná (17).

### JavaScript

Je programovací jazyk, v ktorom sa píšu klientske skripty. To znamená, že sa program resp. zdrojový kód odosiela so stránkou na stranu klienta (do prehliadača) a až tam je vykonaný. Protikladom klientskych skriptov sú skripty serverové, ktoré sú vykonávajú na serveri a u klienta sa zobrazuje len ich výsledok (18).



Obrázok 6: Klientsky skript (Prevzaté z 18)

### Použité návrhové vzory

Návrhový vzor predstavuje všeobecné riešenie problému, ktorý sa opakovane objavuje pri návrhu software. Návrhový vzor nie je knižnicou alebo časťou zdrojového kódu, ktorá by sa dala priamo vložiť do nášho programu. Jedná sa o popis či šablónu, ako riešiť problém spôsobom, ktorý môže byť použitý v rôznych situáciách (13).

#### DAO

Data Access Object (DAO) abstrahuje a zapuzdruje všetok prístup k zdrojom dát. DAO spravuje spojenie so zdrojom dát, aby mohol získavať a ukladať dáta. Inými slovami, DAO jednoducho zavádza rozhranie medzi zdroj dát a konzumentmi dát, čím izoluje logiku správy dát do jednej vrstvy a segreguje vrstvu perzistencie dát od aplikačnej logiky (14).

Použitie návrhového vzoru DAO prináša okrem iného aj tieto výhody:

* Zavádza transparentnosť
* Prináša jednoduchšiu migráciu a prenositeľnosť
* Redukuje komplexitu kódu v biznis objektoch
* Centralizuje všetok prístup k dátam do jednej vrstvy
* Pridáva vrstvu do hierarchie (14)

# Analýza súčasného stavu

# Vlastné návrhy riešenia

# Záver

# Použitá literatúra

1. SODOMKA, Petr a Hana KLČOVÁ. *Informační systémy v podnikové praxi. 2. aktualiz. a rozš. vyd.* Brno: Computer Press, 2010, 501 s.

ISBN 978-80-251-2878-7.

1. PEACH, R. W. *The ISO 9000 Handbook (Fourth Edition).* New York: McGraw-Hill, QSU Publishing Company, 2002. ISBN 1-932191-00-3.
2. BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: podnik v informační společnosti.* 3., aktualiz. a dopl. vyd. Praha: Grada, 2012, 323 s. ISBN 9788024743073.
3. ARLOW, J. a I. NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 1. vyd. Brno: Computer Press, 2007. 567 s. ISBN 978-80-251-1503-9.
4. RYDVAL, R. *Případy užití (Use Cases)* [online]. [cit. 2014-03-04]. Dostupné z: <http://ocup.ocup.cz/search/label/p%C5%99%C3%ADpady%20u%C5%BEit%C3%AD>
5. BUHNOVÁ, B. *Ilustrační příklad v jazyce UML* [online]. [cit. 2014-03-04]. Dostupné z: <http://www.fi.muni.cz/~buhnova/PV167/priklad.html>
6. CONOLLY, Thomas, Carolyn E BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází.* Vyd. 1. Brno: Computer Press, 2009, 584 s. ISBN 978-80-251-2328-7.
7. *Teorie relačních databází: Normalizace* [online]. 2007 [cit. 2014-03-07]. Dostupné z: <http://www.manualy.net/article.php?articleID=13>
8. *Java EE at a Glance* [online]. 2014 [cit. 2014-03-12]. Dostupné z: <http://www.oracle.com/technetwork/java/javaee/overview/index.html>
9. *Hibernate ORM* [online]. 2014 [cit. 2014-03-12]. Dostupné z: <http://hibernate.org/orm/>
10. *Spring Framework Overview* [online]. 2014 [cit. 2014-03-13]. Dostupné z: <http://www.tutorialspoint.com/spring/spring_overview.htm>
11. *Stripes Overview* [online]. 2013 [cit. 2014-03-13]. Dostupné z: <http://www.stripesframework.org/display/stripes/An+Overview+-+Guiding+Principles>
12. SHVETS, Alexander. *Design Patterns* [online]. 2014 [cit. 2014-03-16]. Dostupné z: <http://sourcemaking.com/design_patterns>
13. *Data Access Object* [online]. 2013 [cit 2014-03-16]. Dostupné z: <https://kore.fi.muni.cz/wiki/index.php/Data_Access_Object>
14. *About* [online]. 2014 [cit 2014-03-16]. Dostupné z: <http://www.postgresql.org/about/>
15. ROUSE, Margaret. *HTML (Hypertext Markup Language)* [online]. 2005 [cit. 2014-03-18]. Dostupné z: <http://searchsoa.techtarget.com/definition/HTML>
16. JANOVSKÝ, Dušan. *CSS styly – úvod* [online]. 2013 [cit. 2014-03-18] Dostupné z: <http://www.jakpsatweb.cz/css/css-uvod.html>
17. *Úvod do JavaScriptu* [online]. 2013 [cit. 2014-03-18]. Dostupné z: <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>

# Zoznam skratiek

ORM

GNU

SQL

DAO

# Zoznam obrázkov

[Obrázok 1: Informačná pyramída podľa organizačných úrovni podniku (Prevzaté z 1) 13](file:///C:\Users\Martin\Dropbox\BAKALARKA\DOKUMENT\Bakalarka\BP_Martin_Sakáč.docx#_Toc382917435)

[Obrázok 2: Hodnototvorný reťazec, riadiace a podporné procesy výrobného podniku (Prevzaté z 3) 16](#_Toc382917436)

[Obrázok 3: Ukážka diagramu prípadu užitia (Prevzaté z 5) 18](#_Toc382917437)

[Obrázok 4: Ukážka návrhového diagramu tried (Prevzaté z 6) 19](#_Toc382917438)

[Obrázok 5: Ukážka ER diagramu (Vlastná tvorba) 21](#_Toc382917439)

[Obrázok 6: Klientsky skript (Prevzaté z 18) 26](#_Toc382917440)

# Zoznam tabuliek

# Zoznam príloh