

# Návrh databáze technologií využívajících obnovitelné zdroje energie pro drobné investory

Bakalářská práce

**Vypracovala:** Veronika Maurerová

**Vedoucí práce:** Ing. Luboš Nobilis

České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Softwarové technologie a management - Manažerská informatika

2014

České vysoké učení technické v Praze  
Fakulta elektrotechnická

Katedra ekonomiky, manažerství a humanitních věd

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Maurerová** Veronika

Studijní program: Softwarové technologie a management  
Obor: Manažerská informatika

*Název tématu:*

### **Návrh databáze technologií využívajících obnovitelné zdroje energie pro drobné investory**

*Pokyny pro vypracování:*

1. Rešerše databází s podobnou tematikou
2. Předpokládané využití databáze
3. Návrh databáze

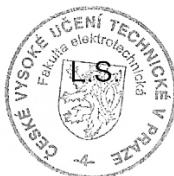
*Seznam odborné literatury:*

1. Conolly T., Begg C., Holowczak R.: Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází. Computer Press, 2009.
2. Hernandez M.J., Holowczak R.: Návrh databází. Grada, 2006.

Vedoucí bakalářské práce: Ing. Luboš Nobilis – ECO Trend s.r.o.

Platnost zadání: do konce letního semestru 2014/2015

*Doc. Ing. Jaroslav Knápek, CSc.*  
vedoucí katedry



*Prof. Ing. Pavel Ripka, CSc.*  
děkan

V Praze dne 10.2.2014

# Poděkování

Děkuji Ing. Luboši Nobilisovi za vedení této práce a dále Ing. Martinu Benešovi, Ph.D. za konzultace v rámci katedry. Další poděkování patří rodině, přátelům a všem těm, kteří mě ve snaze práci dokončit podpořili.

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto udělují nevýhradní oprávnění (licenci) k užití této práce, a to včetně všech příloh, jež jsou její součástí (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

Dne 22. května 2014 v Praze

.....

České vysoké učení technické v Praze  
Fakulta elektrotechnická

© 2014 Veronika Maurerová. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě elektrotechnické. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

## Odkaz na tuto práci

MAUREROVÁ, Veronika. *Návrh databáze technologií využívajících obnovitelné zdroje energie pro drobné investory*. Praha, 2014. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická. Vedoucí práce Ing. Luboš Nobilis.

# Abstrakt

Tato bakalářská práce řeší návrh databáze dle konkrétního zadání. Zadáním je databáze technologií využívajících obnovitelné zdroje energie pro drobné investory. Práce je reakcí na poptávku firmy ECO trend s.r.o. Cílem práce je prozkoumat dosavadní situaci v oblasti dostupných databází na podobné téma, uspořádat požadavky na danou funkční aplikaci, na základě těchto požadavků popsat metodiku návrhu relační databáze a navrhnout konkrétní databázi. V rámci požadavků na databázi jsem se zaměřila především na jejich formulaci. Součástí požadavků je i návrh rozhodovacího formuláře, který hodnotí potenciální investice do produktů uložených v databázi. V rámci návrhu formuláře zvažuji vhodné metody hodnocení investic. Při návrhu jsem se zaměřila na konceptuální, logický i fyzický návrh konkrétní relační databáze. Hlavním výsledkem této práce je dokumentace jak požadavků, tak návrhu databáze pro konkrétní aplikaci, a dále skript k vytvoření databáze na databázovém serveru.

**Klíčová slova:**

*případy užití, hodnocení investice, návrh relační databáze, SQL skript.*

# Abstract

The thesis describes the database design according to the specific assignment, which is to create a database of technologies using renewable energy sources for small investors. The assignment is a reaction on the demand of ECO trend s.r.o. company. The aim is to examine the current situation of available databases in a similar fields, list the requirements of the application, describe the methodology of relational database design and propose specific database according to these requirements. In the context of the requirements, I focused mainly on their formulation. One of the requirements is also a draft of a form, which evaluates potential investments in products listed in the database. In a part of this draft I consider suitable methods of evaluation of the investments. In the context of database design, I focused on the conceptual, logical and physical design of a specific relational database. The main result of this work is the documentation of the requirements, database design for the specific application, and the script that may be used to create the database on the database server.

**Keywords:**

*use cases, investment evaluation, relational database design, SQL script.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>9</b>
<b>2</b>	<b>Rešerše databází s podobnou tematikou</b>	<b>10</b>
2.1	Postup rešerše . . . . .	10
2.1.1	Formulace tématu . . . . .	10
2.1.2	Definice klíčových slov . . . . .	10
2.1.3	Vyhledávání . . . . .	11
2.2	Vyhodnocení nalezených databází . . . . .	11
2.2.1	Databáze environmentálních služeb a výrobků . . . . .	11
2.2.2	RES LEGAL Europe . . . . .	12
2.2.3	DAZE . . . . .	12
<b>3</b>	<b>Předpokládané využití databáze</b>	<b>14</b>
3.1	Definice systému . . . . .	14
3.2	Funkční požadavky na systém . . . . .	14
3.2.1	Pohledy . . . . .	15
3.2.1.1	Pohled běžného uživatele . . . . .	16
3.2.1.2	Pohled producenta . . . . .	17
3.2.1.3	Pohled správce . . . . .	18
3.3	Systémové požadavky na databázi . . . . .	21
3.3.1	Počáteční velikost databáze . . . . .	21
3.3.2	Růst velikosti databáze . . . . .	21
3.3.3	Typy a průměrný počet prohledávaných záznamů . . . . .	22
3.3.4	Výkon . . . . .	22
3.3.5	Uživatelské rozhraní . . . . .	22
3.3.6	Zabezpečení . . . . .	22
3.4	Specifikování funkčnosti formuláře . . . . .	22
3.4.1	Nalezení vhodného řešení . . . . .	24
3.4.2	Výpočet výhodnosti investice . . . . .	26
3.4.2.4	Výpočet ceny za spotřebovanou kilowatthodinu . . . . .	26
3.4.2.5	Výpočet efektivnosti investice . . . . .	27
<b>4</b>	<b>Návrh databáze</b>	<b>31</b>
4.1	Konceptuální návrh . . . . .	31
4.1.1	Identifikace entit . . . . .	31
4.1.2	Identifikování relací . . . . .	32
4.1.3	Určení atributů a primárních klíčů . . . . .	35
4.1.4	Kontrola redundance . . . . .	36

4.2	Logický návrh . . . . .	37
4.2.1	Struktura tabulek . . . . .	37
4.2.2	Kontrola tabulek pomocí normalizace . . . . .	39
4.2.3	Kontrola integritních omezení . . . . .	41
4.3	Fyzický návrh . . . . .	43
4.3.1	Návrh tabulek a integritních omezení . . . . .	43
4.3.2	Návrh odvozených sloupců . . . . .	48
4.3.3	Analýza transakcí . . . . .	50
4.3.4	Indexování . . . . .	51
4.3.5	Návrh pohledů . . . . .	52
4.3.6	Zohlednění denormalizace . . . . .	52
4.3.7	Návrh bezpečnostních opatření . . . . .	54
4.3.8	Testování SQL skriptu . . . . .	54
<b>5</b>	<b>Závěr</b>	<b>55</b>
<b>6</b>	<b>Seznam zdrojů</b>	<b>56</b>
<b>A</b>	<b>Seznam zkratk</b>	<b>58</b>
<b>B</b>	<b>Vize DTOZE</b>	<b>59</b>
<b>C</b>	<b>Finální ER model</b>	<b>59</b>
<b>D</b>	<b>Finální logický model</b>	<b>60</b>
<b>E</b>	<b>Finální databázový model</b>	<b>61</b>
<b>F</b>	<b>SQL skript DTOZE</b>	<b>63</b>
<b>G</b>	<b>Obsah příloženého CD</b>	<b>64</b>



## 1 Úvod

Tématem práce je Návrh databáze technologií využívajících obnovitelné zdroje energie pro drobné investory (dále jen DTOZE). Na zadání bakalářské práce se podílela firma ECO trend s. r. o., která specifikovala požadavky na databázi a dodala základní informace o datech, které mají být v databázi uloženy. Dále se podílela na specifikování využití databáze.

Databáze, která bude zpřístupněna přes webovou aplikaci, má poskytnout laické i odborné veřejnosti přehled o technologiích využívajících obnovitelné zdroje energie (dále jen OZE) a především rozsáhlé informace o konkrétních produktech a jejich producentech či dodavatelích. V databázi by měly být produkty, které jsou dostupné pro drobné investory (např. malé vodní a větrné elektrárny) a které lze umístit na pozemky či obydlí dostupné pro běžné uživatele elektrické energie či tepla. Tyto informace mají pomoci většímu rozšíření povědomí o využití OZE mezi běžné uživatele především v České republice. Součástí aplikace budou i další funkčnosti např. pokročilé vyhledávání či formulář který ze zadaných parametrů navrhne vhodné varianty využití OZE a zhodnotí výhodnost investice. Což má za cíl zlepšit orientaci v datech, ale také nabídnout konkrétní optimální varianty dle možností daného uživatele.

Cílem práce je prozkoumat současnou situaci v oblasti dostupných databází týkajících se OZE, uspořádat požadavky na danou funkční aplikaci a na základě těchto požadavků popsat metodiku návrhu databáze, jejíž výsledkem bude návrh konkrétní databáze. Cílem práce není vytvořit kompletní aplikaci, ale připravit databázi pro další implementaci. Chtěla jsem pracovat na projektu, který bude mít reálné využití, proto jsem zvolila po dohodě se zástupcem vedoucího katedry toto téma. Tvorba podobných systémů mě zajímá a zabývám se jí i přesto, že zaměření mého oboru je více ekonomické než softwarové.

Obsah práce je rozdělen na tři hlavní části. V první části se zabývám rešerší databází na podobné téma v odvětví OZE. Nalezené databáze krátce popisuji a zdůrazňuji jejich klady a zápory, popř. v čem mě inspirovaly. V další části se zabývám využitím databáze, kde zmiňuji požadavky na databázi, popisuji mechanismus vyhledávání a vysvětluji další funkčnosti aplikace, která má na základě návrhu databáze vzniknout. Tato část je nezbytná pro návrh databáze, který popisuji v poslední části. V poslední nejrozsáhlejší části se zaměřuji na metodiku návrhu databáze. Popisuji jednotlivé fáze návrhu od konceptuálního přes logický až po fyzický návrh a metodiku demonstruji na konkrétním návrhu.

## 2 Rešerše databází s podobnou tematikou

Rešerše se používá k hledání vhodných zdrojů informací před zahájením psaní odborné práce.<sup>1</sup> Využila jsem postup rešerše k vyhledání konkrétních databází, které jsou dostupné na Internetu. Cílem bylo zjistit, zda existují podobné databáze, jak jsou koncipované, jaké mají klady a zápory, zda má vůbec význam takovou databázi vytvářet. Hledala jsem databáze, které obsahují data podobná tématu týkající se OZE. U nalezených databází jsem hodnotila, jak se týkají konkrétních technologií využívajících OZE, na co se zaměřují, zda jsou přehledné, jak jsou aktuální apod.

### 2.1 Postup rešerše

V následujících podkapitolách popisuji postup rešerše a bližší seznámení s databázemi, které jsem pomocí rešerše našla. Postup rešerše jsem čerpala z interaktivního průvodce na stránkách Knihovny univerzitního kampusu Masarykovy univerzity v Brně<sup>2</sup>.

#### 2.1.1 Formulace tématu

Základem rešerše je formulovat správně téma, ke kterému hledám informace, v mém případě konkrétní webové stránky. Není vhodné se omezit pouze na český jazyk, ale je důležité formulovat téma i v jiných jazycích, minimálně alespoň v anglickém jazyce. Já zvolila téma následující:

Téma v českém jazyce: Databáze technologií využívajících obnovitelné zdroje energie pro drobné investory.

Téma v anglickém jazyce: Database of technologies using renewable energy sources for small investors.

#### 2.1.2 Definice klíčových slov

Důležité je si definovat i klíčová slova, jelikož téma je často složité a nelze jej vložit jako celek do vyhledávače. Samotná klíčová slova však nemusí stačit, a proto je vhodné si stanovit i nadřazená klíčová slova nebo synonyma. Opět vše stanovuji současně i v angličtině.

##### **Klíčová slova:**

databáze technologií využívajících obnovitelné zdroje energie, database of technologies using renewable energy sources;

technologie využívající zdroje energie, technologies using renewable energy sources.

##### **Nadřazená klíčová slova:**

databáze obnovitelných zdrojů energie, database of renewable energy sources;

databáze OZE, database of RES;

<sup>1</sup>Rešerše. *Ústřední knihovna ČVUT* [online]. [cit. 2013-12-10]. Dostupné z: <http://knihovna.cvut.cz/sluzby/reserse/co-je-reserse.html>.

<sup>2</sup>Rešerše: modelový příklad postupu. *Správa Univerzitního kampusu Masarykovy Univerzity* [online]. [cit. 2013-12-10]. Dostupné z: [http://ukb.muni.cz/kuk/animace/eiz/Reserse/reserse\\_praxe.html](http://ukb.muni.cz/kuk/animace/eiz/Reserse/reserse_praxe.html).

obnovitelné zdroje energie, renewable energy sources;  
zdroje energie, energy sources.

### **Synonyma:**

databáze technologií využívajících OZE, database of technologies using RES;  
databáze technologií využívajících obnovitelné přírodní zdroje energie, database of technologies using natural renewable energy sources;  
databáze technologií využívajících přírodní nefosilní zdroje energie, database of technologies using natural renewable non-fossil energy sources;  
katalog technologií využívajících OZE, catalog of technologies using RES.

### **2.1.3 Vyhledávání**

Až po přípravě vhodných slov a frází lze přejít k samotnému vyhledávání. K vyhledávání jsem použila tři vyhledávače: Google.com, Seznam.cz, Bing.com. Zadávala jsem nejdříve klíčová slova a synonyma, ale s výsledkem jsem nebyla spokojena. Musela jsem použít nadřazená slova. S nadřazenými slovy jsem již měla více výsledků. Ze všech výsledků jsem vyhodnotila jako vhodné následující databáze, týkající se obnovitelných zdrojů energie:

1. Databáze environmentálních služeb a výrobků, zobrazení subjektů na mapě, podrobnější informace o subjektech - <http://www.envisluzby.cz/>.
2. Mezinárodní evropská databáze politik v oblasti OZE - <http://www.res-legal.eu/>.
3. Databáze OZE - informace o jednotlivých druzích OZE, kontakty na subjekty v ČR zabývající se OZE, kontaktní osoby - <http://daze.vukoz.cz/daze/index.jsp>.

## **2.2 Vyhodnocení nalezených databází**

Rešerší jsem zjistila, že momentálně není na Internetu dostupná žádná online databáze technologií využívajících obnovitelné zdroje energie (natož pro drobné podnikatele či investory) v českém nebo anglickém jazyce. Nalezla jsem však několik databází, které jsou po obsahové stránce podobné – zabývají se tématem obnovitelných zdrojů energie. Některé z nich mě inspirovaly, jiné navedly k jinému řešení.

### **2.2.1 Databáze environmentálních služeb a výrobků**

Databázi environmentálních služeb a výrobků provozuje České ekologické manažerské centrum (CEMC). Databáze vznikla v rámci tříletého projektu „Důsledky liberalizace environmentálních služeb“, financovaného z prostředků Grantové agentury České republiky. Databáze vznikla v roce 2011 a nyní je v databázi zaregistrováno 1079 subjektů.

Obsahem databáze jsou subjekty, které poskytují služby, vyrábí technologie, výrobky, přípravky, vzdělávají nebo provádějí výzkum za účelem předejít vzniku škody na životním prostředí, zmírnit

či odstranit negativní dopad na životní prostředí, zdraví a majetek osob na území České republiky. Do databáze se subjekty dostávají přes registraci, která je zdarma. Uvedou základní údaje o firmě a výpis služeb či výrobků, které nabízejí.

Na webových stránkách lze zobrazit seznam všech subjektů, v kterém lze hledat podle typu služeb, dle zpracovatelů odpadů nebo dle názvu subjektu. Vedle seznamu jsou subjekty zaznamenány i v mapě se štítkem, který specifikuje zaměření firmy. Mapu lze přibližovat a oddalovat, popř. lze vybrat konkrétní kraj, okres či město. Lze zobrazit detail subjektu, kde jsou zobrazeny kontaktní údaje a na co se subjekt zaměřuje.<sup>3</sup>

### 2.2.2 RES LEGAL Europe

RES LEGAL Europe databáze byla vytvořena z iniciativy Evropské komise. Je to online volně dostupná databáze v anglickém jazyce, editovaná odborníky. Obsahem databáze jsou informace o právních předpisech, státních podporách a záležitostech týkající se obnovitelných zdrojů energie ve 27 zemích Evropské unie (dále EU), v zemích ESVO a některých nových členských zemích EU. Databáze zahrnuje tři energetické odvětví: elektřina, vytápění a chlazení, doprava.

Cílem databáze je poskytnout jednak rychlý přehled o různých právních předpisech dané země zaměřených na OZE a současně data zobrazit jasně, výstižně a přehledně. Databáze je aktualizována pravidelně. Datum vytvoření a poslední aktualizace, jméno odpovědné osoby a další informace se zobrazují u každého profilu dané země. Popisy jsou vytvořeny na základě příslušných zákonných zdrojů. Obsahují i odkazy na původní právní prameny v původním jazyce. Většinou je k dispozici i odkaz na překlad zákona. Na stránce je k dispozici také seznam kontaktů na vnitrostátní orgány a odborníky, které lze dále kontaktovat.

Každá země má na stránkách vlastní profil, na kterém lze dohledat veškeré informace týkající se dané země. Databáze umožňuje srovnávat jednotlivé země na základě státních podpor, záležitostí týkající se silových elektrických a komunikačních sítí a politik a předpisů. Dále lze na stránkách vyhledávat pomocí srovnávacího nástroje, který umožní vybrat země, typ energetického sektoru a další specializace daného sektoru a zobrazit požadované informace v tabulce. Lze si i prohlédnout archiv dokumentů seřazený dle zemí. Tato databáze je velmi přehledná i přes její rozsáhlost. Zaměřuje se však pouze na administrativní záležitosti ohledně OZE.<sup>4</sup>

### 2.2.3 DAZE

Databáze obnovitelných zdrojů energie DAZE shromažďuje informace o obnovitelných zdrojích energie. V první části obsahuje seznam firem, které se zabývají OZE, v další části se pak blíže zaměřuje na biomasu. Vznikla a vyvíjela se za finanční podpory výzkumného projektu „Zdokonalování stávajících technologií využívání obnovitelných zdrojů a úspor energie“ (MŽP ČR 320/3/99), projektu „Vyšší využití nepotravinářské zemědělské produkce v průmyslu“ (MZe ČR QF 4142) a projektu „Nepotravinářské využití biomasy v energetice“ (MŠMT 2B06131).

<sup>3</sup>Envisluzby.cz. CEMC. *Databáze environmentálních služeb a výrobků* [online]. 2011 [cit. 2013-12-10]. Dostupné z: <http://www.envisluzby.cz/>.

<sup>4</sup>Res-legal.eu. EUROPEAN COMMISSION. *Renewable energy policy database and support* [online]. 2012 [cit. 2013-12-10]. Dostupné z: <http://www.res-legal.eu/>.

Z databáze lze získat profily jednotlivých firem. Ty obsahují název firmy, adresu, webové stránky, popř. kontaktní osobu. Lze pak zobrazit i všechny kontaktní osoby obsažené v databázi a zobrazit si též jejich profil s kontaktními údaji. Seznamy jsou seřazené podle abecedy. Subjekty lze vyhledávat i dle zaměření na konkrétní typ OZE.

V druhé části lze získat velmi podrobné informace o plodinách, které jsou vhodné pro produkci biomasy. Ty jsou opět v seznamu seřazeném podle abecedy, lze však zobrazit i seznamy dle typu plodiny (např. bioplyn, brikety, vrby, topoly, atd.). Každá plodina má vlastní profil, kde jsou zobrazeny informace o typu plodiny, v jakých lokalitách ji lze pěstovat, jaké lze očekávat výnosy, jak se pěstuje, možnosti využití a mnoho dalšího. Informace jsou členěny do záložek. Ze stránek lze tisknout data dané plodiny, stáhnout soubor ve formátu PDF nebo zobrazit ve formátu XML.

Tato databáze se nejvíce podobá zadání DTOZE. Obsahuje strukturovaná data a profily jednotlivých subjektů a druhů OZE, i když specializovaných pouze na biomasu. Je určená pro odbornou i laickou veřejnost. DAZE je vytvářena a průběžně aktualizována pracovníky odd. fytoenergetiky VÚKOZ Průhonice ve spolupráci s dalšími organizacemi.<sup>5</sup>

---

<sup>5</sup>Daze.vukoz.cz. VÝZKUMNÝ ÚSTAV SILVA TAROUČY PRO KRAJINU A OKRASNÉ ZAHRADNICTVÍ, veřejná výzkumná instituce. *DAZE: Databáze obnovitelných zdrojů energie* [online]. 2005 [cit. 2013-12-10]. Dostupné z: <http://daze.vukoz.cz/daze/index.jsp>.

### 3 Předpokládané využití databáze

Před samotným návrhem databáze pro aplikační software (zkráceně aplikaci) je důležité si projít i předcházející fáze životního cyklu vývoje databáze, kterými jsou plánování databáze, definice systému a sběr a analýza požadavků.<sup>6</sup>

Informace z těchto fází se netýkají pouze samotné databáze (tedy ukládání a zpřístupnění dat), ale celého systému, který bude nad databází fungovat. Na sběru těchto informací se podílí zadavatel a vývojář (pokud se jedná o menší systém). V případě větších či složitějších systémů jsou do vstupní analýzy zahrnuti i budoucí uživatelé systému (např. zaměstnanci firmy či samotní zákazníci) a z řad vývojářů i několik analytiků, kteří používají různé metody zjišťování požadavků. Zjištěné informace je klíčové sepsat a potvrdit oběma stranami. Ujasní se tak veškeré požadavky a podmínky obou stran. Takový dokument se nazývá vize projektu. Vize projektu DTOZE vznikala souběžně s vypracováním tohoto textu a je přílohou B.

Pro návrh databáze je klíčové si se zadavatelem ujasnit požadavky a jednotlivé případy užití, jelikož z těchto dat se vychází při návrhu celého systému. Proto neuvádím celou vizi jakou součástí textu této práce, ale pouze její část. Definici systému, funkční a obecné požadavky a specifikování funkčnosti formuláře popisují následující kapitoly. Při analýze a popisu požadavků jsem vycházela z knihy *Mistrovství databáze*<sup>7</sup> a spolupracovala se zaměstnanci firmy ECO trend. Metodiku analýzy a sběru dat podrobně nerozepisuji, jelikož není předmětem této práce. Podrobněji se věnuji pouze vysvětlení funkčnosti formuláře (v kapitole č. 3.4), který je na základě požadavků nutné sestavit a rozhodnout se pro vhodné výpočty a správné metody hledání.

#### 3.1 Definice systému

Cílem DTOZE je zpřístupnit široké veřejnosti metodiku rozvoje drobných OZE bez veřejné podpory na internetu přes webovou aplikaci. Informace budou zpřístupněny přes webovou aplikaci především přes katalog produktů, které pro výrobu elektrické energie využívají OZE. Pomocí aplikace bude možné získat doporučení pro jaký produkt se rozhodnout či další informace o OZE či jednotlivých technologiích využívajících OZE.

#### 3.2 Funkční požadavky na systém

Funkční požadavky na systém lze rozdělit do tří základních modulů:

- katalog produktů:
  - zobrazení detailu produktu,
  - vícekritériální vyhledávání,
  - administrace katalogu,

<sup>6</sup>CONOLLY, Thomas, Carolyn 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. Str. 109.

<sup>7</sup>Tamtéž str. 109 - 153.

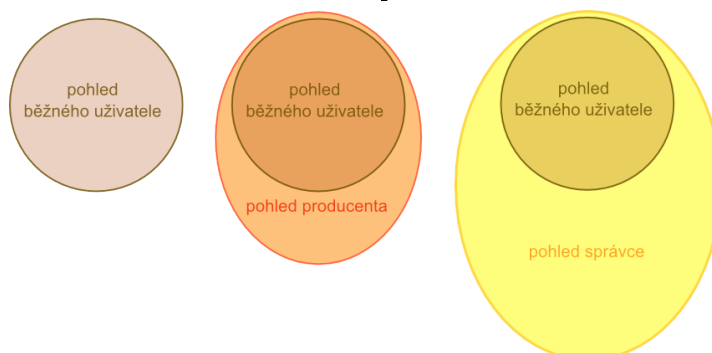
- interaktivní formulář:
  - vyhledání nejvhodnějších produktů dle zadání,
  - výpočet efektivnosti investice,
- obecné informace o OZE a technologiích využívajících OZE.

Podrobnější popis a konkrétní zpracování jsou uvedeny v jednotlivých pohledech. Pohled popisuje stručně požadavky, které byly sjednány. Dále jsou zobrazeny v diagramu jednotlivé případy užití, které byly vytvořeny na základě požadavků. Pro vizualizaci případů užití používám nástroj Enterprise Architect.

### 3.2.1 Pohledy

Systém bude rozlišovat několik pohledů - pohled běžného uživatele, producenta, administrátora. Pohled běžného uživatele sdílí producent i správce. Pohled producenta je ve většině případů odlišný od pohledu správce a proto nelze označit pohled správce za nadřazený pohledu producenta s přidáním funkčnosti. Prolínání pohledů znázorňuje obrázek č. 1.

Obrázek 1: Prolínání pohledů na DTOZE



Jednotlivé možnosti uživatelů v různých pohledech popisují následující podkapitoly. Pod každým pohledem je uveden diagram znázorňující jednotlivé případy užití a vazbu na pohled a dále je uvedena tabulka, která obsahuje případy užití na dle pohledů s popisem. V tabulce je také uvedena obtížnost implementování a prioritizace vypracování. Popisky k jednotlivým hodnotám jsou znázorněny v tabulce č. 1 a tabulce č. 2.

Tabulka 1: Popisek vysvětlující značené obtížnosti v tabulkách případů užití

Obtížnost	
1	lehké
2	středně obtížné
3	obtížné

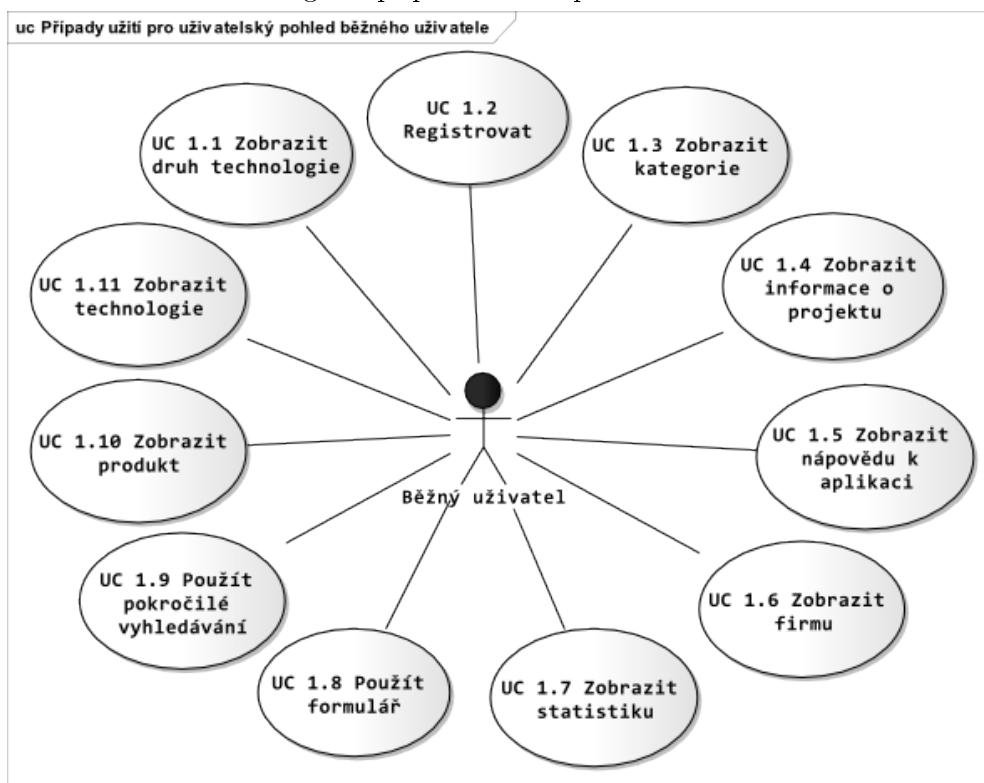
Tabulka 2: Popisky vysvětlující značení priority v tabulkách případů užití

Priorita	
1	malá (nemusí být implementováno v 1. fázi)
2	střední (mělo by být implementováno v 1. fázi)
3	velká (musí být implementováno v 1. fázi)

### 3.2.1.1 Pohled běžného uživatele

Běžný uživatel zobrazí technologie ve všech kategoriích, vyhledává a filtruje v těchto datech. Dále zobrazí obecné informace k technologiím či o výrobci, sleduje základní statistiky, zobrazí informace o projektu či zadavateli. Může využít filtrační formulář pro získání doporučení jaké technologie využívající OZE jsou vhodné. Z běžného uživatele se stane producent (registrovaný uživatel) po vyplnění registrace a schválení registrace správcem. Diagram a tabulka případů užití z pohledu běžného uživatele je znázorněn na obrázku č. 2 (diagram) a v tabulce č. 3.

Obrázek 2: Diagram případů užití z pohledu běžného uživatele





Tabulka 3: Tabulka případů užití z pohledu běžného uživatele

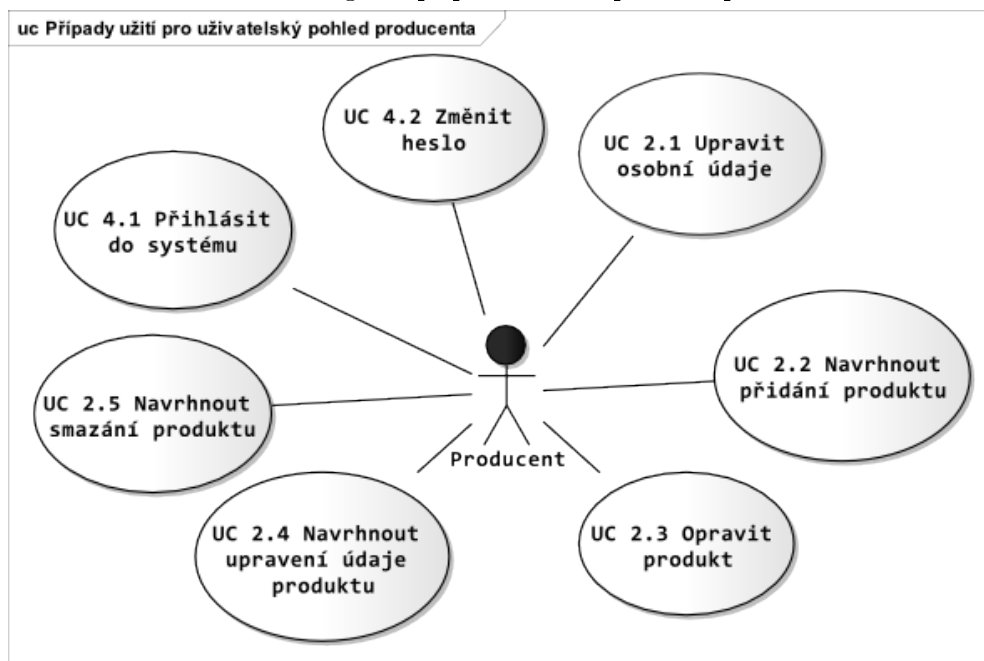
Případy užití z pohledu běžného uživatele			
Název	Popis	Obtížnost	Priorita
UC 1.1 Zobrazit druh technologie	Uživatel zobrazí obecné informace o druhu technologie. V rámci informací bude uveden i odkaz na kategorii, do které druh technologie spadá.	1	3
UC 1.2 Registrovat	Uživatel se registruje odesláním přihlášku do systému. Ta musí obsahovat celé jméno, kontaktní e-mail, název firmy, za kterou chce přidávat produkty, telefonní číslo na firmu, web. stránka firmy, adresu firmy (ulice, číslo popis., obec, stát). Každou firmu může zastupovat pouze jeden uživatel.	1	3
UC 1.3 Zobrazit kategorie	Uživatel zobrazí obecné informace o jednotlivých kategoriích obnovitelných zdrojů energie. V rámci informací budou uvedeny odkazy na jednotlivé druhy technologií.	1	3
UC 1.4 Zobrazit informace o projektu	Uživatel zobrazí obecné informace o projektu.	1	3
UC 1.5 Zobrazit nápovědu k aplikaci	Uživatel zobrazí nápovědu k aplikaci.	2	3
UC 1.6 Zobrazit firmu	Uživatel zobrazí detail firmy.	1	3
UC 1.7 Zobrazit statistiku	Uživatel zobrazí statistiky týkající se obsahu databáze - seznam nejčastěji zobrazených výrobků, seznam nejčastěji zobrazených firem, seznam naposledy přidaných výrobků, seznam naposledy upravených výrobků, počet produktů v databázi, počet firem v databázi, počet produktů využívající danou technologii.	2	2
UC 1.8 Použít formulář	Uživatel zadá: dostupné prostředky v českých korunách informace o stávající struktuře - kolik využívá elektrické energie ze sítě za rok; - jak je zásoben teplem - plynem, jakým uhlím z jaké oblasti, dřevem kusovým, dřevní štěpkou, obilím - zrna, slámou obilnou a kolik utratí za rok; - kolik mu kde a jaké proteče vody - pitná z veř. vodovodu, vlastní zdroj, užitková z veř. vodovodu, vlastní zdroj, splašková do veř. kan., jímky, dešťová do kanalizace, zasakování; dostupné kapacity - umístění - střecha, pozemek, fasáda, vodní hospodářství - druh umístění - šikmá či plochá střecha, pozemek svažitý či rovina, potok, vodní nádrž, studna (hloubka), hladina podz. vody, splaškové vody, dešťová voda - orientaci - J, JV, JZ, Z, V. Dle zadání a z dat dostupných v databázi systém nalezne vhodná řešení a vypočítá výhodnost investice.	3	3
UC 1.9 Použít pokročilé vyhledávání	Uživatel zobrazí formulář pro stanovení kritérií vyhledávání a aplikace dle zadaných požadavků zobrazí výsledky. Kriteria: hledání produktu - název, firma, datum přidání, datum poslední úpravy, seřazení dle datumu přidání vzestupně/sestupně, seřazení dle datumu poslední úpravy sestupně, kategorie, druh technologie, technologie, při výběru konkrétní technologie i konkrétní údaje jednotlivých technologií; hledání firmy - název, položky adresy (ulice, město, stát) s nejvíce přidanými produkty, nejhledanější, technologie nabízených produktů.	3	3
UC 1.10 Zobrazit produkt	Uživatel zobrazí detail produktu. Zobrazí se obecné informace a další typy údajů. Obecné údaje budou zobrazeny hned, jiné se zobrazí po rozkliknutí. Některé údaje mohou být zobrazeny i v případě, že nemají definovanou hodnotu. V rámci informací budou uvedeny pomocí odkazu kategorie, druh technologie a technologie daného produktu.	2	3
UC 1.11 Zobrazit technologii	Uživatel zobrazí obecné informace ke konkrétní technologii. V rámci informací bude uveden i odkaz na druh technologie, do které technologie spadá.	1	3

### 3.2.1.2 Pohled producenta

Producent může vše co běžný uživatel. Po úspěšné registraci a přihlášení se může podílet na obsahu databáze. Registrovat se může výrobce či dodavatel produktů a přidávat tak do databáze své výrobky, které vyrábí či dodává a ještě nejsou v databázi zaznamenané. Může uvést ve svém profilu bližší informace o své firmě či doplnit chybějící údaje. Na svém profilu může sledovat, kolik záznamů přidal

do databáze, které čekají na schválení a které byly zamítnuty. Jeho účet může být deaktivován. Diagram a tabulka případů užití z pohledu producenta je znázorněn na obrázku č. 3 (diagram) a v tabulce č. 4. S pohledem správce producent sdílí dva případy užití, které jsou znázorněny samostatně v tabulce č. 5.

Obrázek 3: Diagram případů užití z pohledu producenta



Tabulka 4: Tabulka případů užití z pohledu producenta

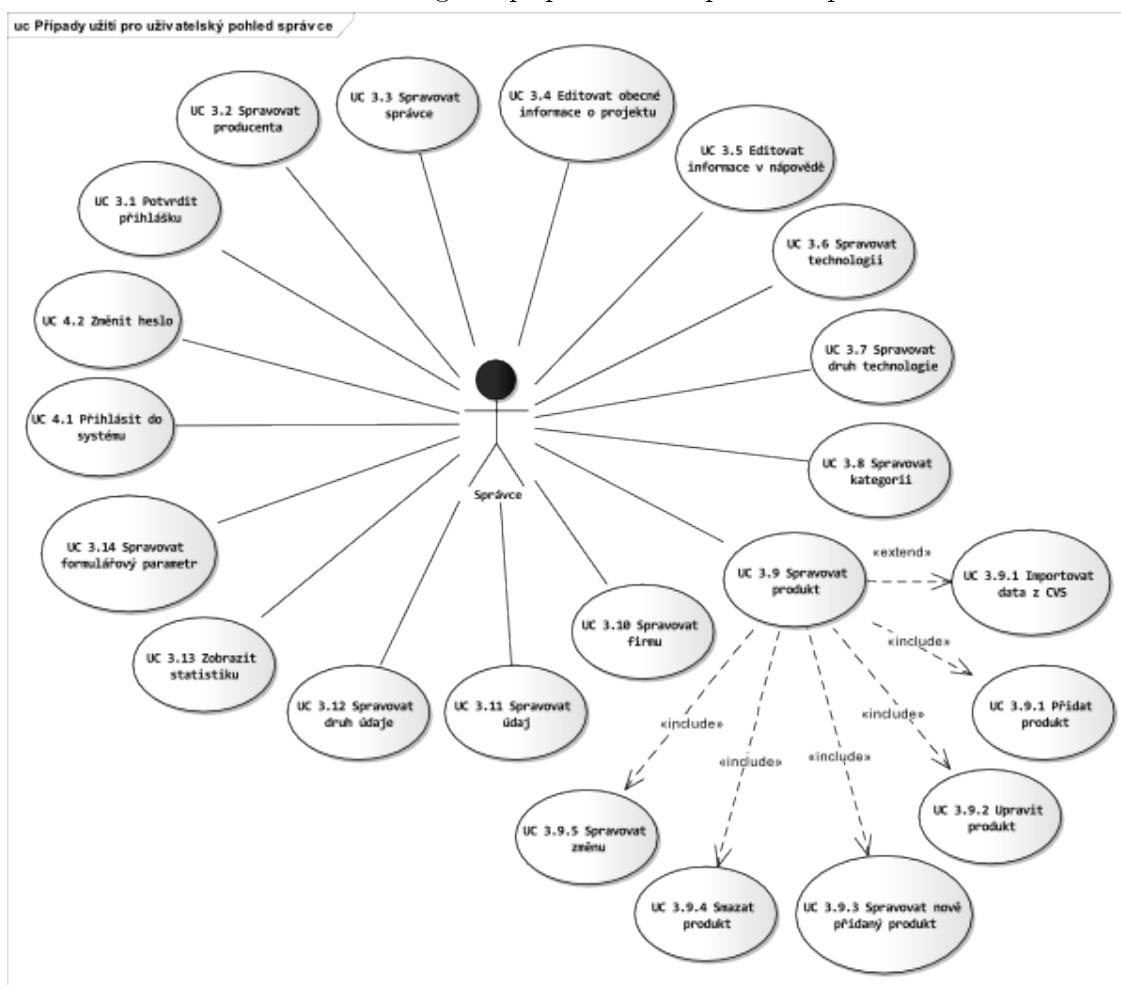
Případy užití z pohledu producenta			
Název	Popis	Obtížnost	Priorita
UC 2.1 Upravit osobní údaje	Uživatel nemůže přímo upravit údaje, které zadal při registraci. Vyplní žádost o upravení údajů, která se odešle správcům na e-mail.	1	2
UC 2.2 Navrhnout přidání produktu	Uživatel navrhne přidání nového produktu do databáze pod firmou, kterou zastupuje. Vybere technologii a vyplní požadované údaje (název povinně, url fotky nebo nahraje fotografii ze svého zařízení nepovinně, stručný popis produktu nepovinně). Na základě technologie se zobrazí další údaje, které uživatel vyplní. Odesláním se nastaví stav produktu na WAIT. V případě potvrzení je produkt zanesen do databáze. V případě zamítnutí může uživatel produkt ještě upravit nebo produkt smazat.	3	2
UC 2.3 Opravit produkt	Uživatel zobrazí produkt, který přidal a má stav KO. Takovému produktu opraví údaje, které dle zprávy od správce nevyhovují a nastaví stav produktu na WAIT. Uživatel sleduje stav produktu a popř. zprávu od správce.	2	2
UC 2.4 Navrhnout upravení údaje produktu	Uživatel upraví jakýkoli údaj produktu, který patří k firmě, kterou zastupuje. Změna je zobrazena správci. Ten ji schválí a nebo zamítne. Uživatel sleduje stav změny a popř. zprávu od správce.	3	2
UC 2.5 Navrhnout smazání produktu	Uživatel navrhne smazání produktu. Odesláním nastaví stav produktu na DEL. Uživatel sleduje stav návrhu a popř. zprávu od správce.	1	2

### 3.2.1.3 Pohled správce

Správce se stará o veškerý obsah webové aplikace. Přidává produkty do databáze, edituje je nebo maže. Produkty může přidávat manuálně či je importovat. Může editovat obecné informace o kategoriích a technologiích, obecné informace o projektu. Má přístup k veškerým statistikám. Dále potvrzuje

a ruší přihlášky uživatelů, kteří mají zájem přispívat do databáze. Validuje také nově přidané výrobky a žádosti o změnu od registrovaných uživatelů. Producentům, kteří nedodržují pravidla pro přidávání nových produktů může správce deaktivovat účet. Také administrátorům může deaktivovat účet. Má možnost vytvořit i nového správce. Dále také může editovat formulář, přidávat nebo mazat dotazníkové položky. Diagram a tabulka případů užití z pohledu správce je znázorněn na obrázku č. 4 (diagram) a v tabulce č. 6. S pohledem producenta správce sdílí dva případy užití, které jsou znázorněny samostatně v tabulce č. 5.

Obrázek 4: Diagram případů užití z pohledu správce



Tabulka 5: Tabulka případů užití společných pro pohled producenta a správce

Případy užití stejné pro pohled producenta i správce			
Název	Popis	Obtížnost	Priorita
UC 4.1 Přihlásit do systému	Uživatel se přihlásí do systému pomocí svého emailu a hesla.	2	3
UC 4.2 Změnit heslo	Uživatel změní heslo.	1	3

Tabulka 6: Tabulka případů užití z pohledu správce

Případy užití z pohledu správce			
Název	Popis	Obtížnost	Priorita
UC 3.1 Potvrdit přihlášku	Uživatel zobrazí seznam přihlášek do systému a dle vyplněných hodnot rozhodne, zda přihlášku přijme (systém vytvoří nového producenta a přiřadí k němu firmu a pošle informační e-mail) či odmítne (přihlášku smaže a pošle žadateli informační e-mail).	2	2
UC 3.2 Spravovat producenta	Uživatel může editovat údaje producentů. Dále může producentovi zneaktivnit účet. Produkty přidané producentem v databázi zůstanou, ale producent již dále nebude moci využívat možnosti producentů.	1	2
UC 3.3 Spravovat správce	Uživatel přidá nového správce. Vyplní jméno a e-mail. Systém odešle přihlašovací údaje na zadný e-mail.	2	3
UC 3.4 Editovat obecné informace o projektu	Uživatel doplní či smaže obecné informace o projektu.	1	3
UC 3.5 Editovat informace v nápovědě	Uživatel doplní či smaže informace v nápovědě.	1	3
UC 3.6 Spravovat technologii	Uživatel přidá novou technologii a přiřadí k ní údaje, které budou obsahovat veškeré produkty dané technologie. Dále technologii přiřadí do druhu technologie (povinně) a vyplní informace o dané technologii (nepovinně). Také edituje a maže technologie. Smazat technologii lze pouze pokud nemá vazby na žádné produkty, údaje a parametry dotazníku.	1	3
UC 3.7 Spravovat druh technologie	Uživatel přidá nový druh technologie. Zadá název (povinně) a informace o daném druhu technologie (nepovinně). Druh technologie zařadí do kategorie (povinně). Dále druh může editovat či smazat. Smazat lze pouze pokud neexistují vazby s technologiemi.	1	3
UC 3.8 Spravovat kategorii	Uživatel přidá kategorii, vyplní název (povinně) a informace o dané kategorii (nepovinně) nebo kategorii upraví či nenávratně smaže. Kategorii lze smazat pouze pokud neobsahuje žádné druhy technologií.	1	3
UC 3.9 Zobrazit produkty	Uživatel zobrazí produkty s možností správy.	2	3
UC 3.9.0 Importovat data z CSV	Uživatel importuje data z CSV do databáze. Importovat může produkty a firmy, pokud soubor splňuje požadavky (obsahuje sloupce a hodnoty pojmenované stejně jako sloupce v databázi).	3	3
UC 3.9.1 Přidat produkt	Uživatel přidá nový produkt do databáze. Vybere technologii a vyplní požadované údaje (název povinně, url fotky nebo nahraje fotografii ze svého zařízení nepovinně, stručný popis produktu nepovinně, firmu povinně). Na základě technologie se zobrazí další údaje, které uživatel vyplní.	2	3
UC 3.9.2 Upravit produkt	Uživatel může upravit jakékoliv údaje u produktů zanesených v databázi.	1	3
UC 3.9.3 Spravovat nově přidaný produkt	Uživatel zobrazí produkt, který má stav WAIT. Prostuduje produkt, zda obsahuje smyslplné údaje. Pokud je produkt v pořádku změní stav produktu na OK. V opačném případě nastaví stav na KO. Dále uvede důvod svého rozhodnutí.	2	2
UC 3.9.4 Smazat produkt	Uživatel smaže produkt z databáze.	1	1
UC 3.9.5 Spravovat změnu	Uživatel spravuje změnu navrženou producenty, která má stav WAIT. Změnu zanesou do databáze a změní stav změny na OK. V případě zamítnutí nastaví stav na KO. Dále vyplní informace o důvodech svého rozhodnutí, které se zobrazí producentovi, který změnu navrhl.	2	2
UC 3.10 Spravovat firmu	Uživatel přidá firmu. Vyplní název (povinně), webová stránka (nepovinně), telefonní číslo (povinně), ulice (povinně), číslo popisné (povinně), obec (povinně), PSČ (povinně) a stát (povinně), nebo firmu upraví či smaže. Smazat firmu lze pouze v případě, že neobsahuje žádné vazby na produkty či uživatele.	1	3
UC 3.11 Spravovat údaj	Uživatel přidá nový údaj. Vyplní název a přiřadí k údaji druh údaje, jednotku a technologii. Nebo údaj upraví či smaže. Údaj lze smazat pouze pokud neexistují žádné vazby na produkty.	1	3
UC 3.12 Spravovat druh údaje	Uživatel vytvoří druh údaje. Vyplní název. Nebo druh upraví či smaže. Druh lze smazat pouze pokud neexistují žádné vazby s údaji.	1	3
UC 3.13 Zobrazit statistiku	Uživatel zobrazí seznam naposledy přihlášených uživatelů; seznam uživatelů, kteří přidali nejvíce výrobků; správce, kteří vyřídili nejvíce změn či potvrdili nejvíce přihlášek nebo produktů.	2	2
UC 3.14 Spravovat formulářový parametr	Uživatel přidá dotazníkový parametr. Vyplní technologii, orientaci (nepovinně), druh umístění a jednotku (nepovinně).	1	3



### 3.3 Systémové požadavky na databázi

#### 3.3.1 Počáteční velikost databáze

Předpokládá se, že největší nárůst databáze bude ihned po spuštění celého systému. Pověření lidé od zadavatele již nashromáždili velké množství dat a další shromažďují. Předpokládanou velikost databáze při zahájení provozu znázorňuje tabulka č. 7.

Tabulka 7: Tabulka počáteční velikosti databáze

<b>Předpokládané položky databáze DTOZE</b>		
<b>Název</b>	<b>Popis</b>	<b>Počet</b>
Druh technologie	Specializace technologie (např. fotovoltaika, větrná elektrárna apod.). Spadá do určité kategorie.	10
Firma	Firma, která vyrábí či nabízí produkty.	50
Formulářový parametr	Parametr určený technologií, který je důležitý pro výpočty ve formuláři.	35
Hodnota údaje	Konkrétní hodnota údaje pro konkrétní produkt. Bude zachovávat datum přiřazení k údaji.	15 000
Kategorie	Kategorie technologií dle jednotlivých obnovitelných zdrojů energie (např. solární energie, větrná energie apod.).	5
Produkt	Produkt nabízí firma a obsahuje mnoho údajů. Spadá do konkrétní technologie.	1 000
Přihláška	Přihláška pro možnost podílet se jako producent na obsahu databáze.	0
Technologie	Technologie produktů, která určí údaje, které bude produkt obsahovat. (např. FTV panel, FTT panel, apod.)	11
Typ údaje	Specializace údaje, do jaké patří skupiny. Dle typu údaje budou jednotlivé údaje řazeny při prohlížení produktu (např. technické údaje, ekonomické údaje, apod.).	5
Údaj	Údaj k jednotlivým produktům. Bude definován jednotkou. (např. cena CZK)	300
Uživatel	Správci starající se o obsah databáze.	3

#### 3.3.2 Růst velikosti databáze

Databáze bude růst především v počtu produktů, které budou přidávány. Nepředpokládá se velký nárůst uživatelů ani v obecných položkách jako je např. technologie, kategorie, údaj apod. Předpokládaný růst za měsíc a za rok znázorňuje podrobněji tabulka č. 8.

Tabulka 8: Tabulka předpokládaného růstu za měsíc a za rok

<b>Předpokládaný růst databáze DTOZE</b>		
<b>Název</b>	<b>Nárůst za měsíc</b>	<b>Nárůst za rok</b>
Druh technologie	0	0
Firma	5	50
Formulářový parametr	0	2
Hodnota údaje	300	4000
Kategorie	0	0
Produkt	20	300
Přihláška	1	20
Technologie	0	0
Typ údaje	0	1
Údaj	0	10
Uživatel	0	5

### 3.3.3 Typy a průměrný počet prohledávaných záznamů

Předpokládá se, že databázový systém prostřednictvím webové aplikace navštíví ze začátku 5 lidí denně později až 20 lidí denně. Dle toho se odvíjí i typy a průměrný počet prohledávaných záznamů, které popisuje tabulka č. 9.

Tabulka 9: Tabulka typů vyhledávání a průměrný počet prohledávaných záznamů

Typy a přibližný počet vyhledávání záznamů v databázi DTOZE	
Typ vyhledávání	Počet
Vyhledání produktu	40 denně
Vyhledání údaje	600 denně
Vyhledání typu údaje	600 denně
Vyhledání firmy	5 denně
Vyhledání technologie	3 denně
Vyhledání druhu technologie	3 denně
Vyhledání kategorie	3 denně
Vyhledání formulářového parametru	30 denně

### 3.3.4 Výkon

Očekávaná odezva je méně než 1 sekundu při vyhledávání 1 záznamu. Očekávaná odezva je méně než 3 sekundy při vyhledávání více záznamů.

### 3.3.5 Uživatelské rozhraní

Uživatelské rozhraní bude mít podobu webové aplikace. Uživatelské rozhraní bude uživatelsky příjemné a bude obsahovat online dostupnou nápovědu pro všechny pohledy.

### 3.3.6 Zabezpečení

Aplikace bude zpřístupněna pomocí platného e-mailu a hesla. U producentů po potvrzení přihlášky, kde uvedou i údaje o firmě, kterou budou zastupovat. Správci mohou přidělit přístup i dalším osobám pověřeným správou databáze. Uživatelé budou mít práva upravovat obsah databáze dle jednotlivých pohledů (tedy dle pohledu producenta a pohledu správce).

## 3.4 Specifikování funkčnosti formuláře

Jednou z funkčností webové aplikace je interaktivní formulář. Tento formulář bude dostupný pro všechny uživatele aplikace. V popisu případů užití je zmíněno co přesně uživatel zadá za data, a co formulář vyhodnotí za výsledky, ale není konkrétně popsáno, jak bude fungovat. V následujících kapitolách podrobně popisují jaké výpočty bude aplikace provádět, a s jakými daty krom dat zadaných uživatelem bude tyto výpočty provádět. Jaká data získá od uživatele znázorňují jednotlivé obrázky - na obrázku č. 5 je formulář k získání informace o dostupných prostředcích, na obrázku č. 6 je formulář k získání informace o stávajících prostředcích a na obrázku č. 7 je poslední formulář k získání informace o dostupných kapacitách. Jedná se pouze o návrh, jednotlivé položky mohou být implementovány různě, dle toho, zda lze vybrat pouze jednu možnost či nikoliv, apod. Nyní je to řešeno tímto způsobem, aby bylo zřejmé, jaké vstupy lze získat.

Obrázek 5: Návrh formuláře DTOZE část financování

**Financování**

Dostupné prostředky:  CZK

**POKRAČOVAT >>>**

Obrázek 6: Návrh formuláře DTOZE část stávající prostředky

**Stávající prostředky**

**Elektrická energie**

Spotřeba ze sítě:  kWh/rok      cena za rok:  CZK

**Teplo**

Plyn:	<input type="text"/>	kWh/rok	cena za rok:	<input type="text"/>	CZK
Centrální zásobování:	<input type="text"/>	kWh/rok	cena za rok:	<input type="text"/>	CZK
Hnědé uhlí prachové Most:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Hnědé uhlí tříděné Most:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Hnědé uhlí prachové Sokolov:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Hnědé uhlí tříděné Sokolov:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Černé uhlí prachové Ostrava:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Černé uhlí energetické Ostrava:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Černé uhlí prachové Kladno:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Černé uhlí energetické Kladno:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Koks otopový:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Lignit:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Brikety:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
LTO:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Dřevo kusové:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Dřevní štěpka:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Obilí - zrno:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK
Sláma obilná:	<input type="text"/>	tun/rok	cena za rok:	<input type="text"/>	CZK

**POKRAČOVAT >>>**

Obrázek 7: Návrh formuláře DTOZE část dostupné kapacity

**Dostupné kapacity**

**Střecha**

šikmá:  m<sup>2</sup> orientace: V ▾

plochá:  m<sup>2</sup> orientace: V ▾

**Fasáda**

fasáda, balkon, zábradlí, apod.:  m<sup>2</sup> orientace: V ▾

**Pozemek**

svažitý:  m<sup>2</sup> orientace: V ▾

rovina:  m<sup>2</sup>

**Vodní hospodářství**

potok:  l/s

vodní nádrž:  m<sup>3</sup>

studna (hloubka):  m

hladina podzemní vody:  m

splašková voda:  m<sup>3</sup>/rok

dešťová voda:  m<sup>3</sup>/rok

POKRAČOVAT >>>

### 3.4.1 Nalezení vhodného řešení

Vstupní data jsou využita především pro určení produktu, který je pro uživatele dostupný ať už z finanční stránky, či z hlediska umístění. Každá technologie je svázána s umístěním. Např. FTV panel lze umístit na střechu rodinného domu či bytu nebo na fasádu či pozemek. Naopak studnu či vodní nádrž lze využít při výrobě elektrické energie tepelným čerpadlem. Dle umístění se odvíjí i parametr pro přepočet výkonu. Např. u fotovoltaiky je důležité, jak budou panely orientovány. Při orientaci na jih mají největší výkon, na západ či východ je např. výkon menší. Parametr pro přepočet výkonu je důležitý až ve výpočtech, na hledání vhodného řešení nemá vliv. Přehled předpokládaných technologií, které budou dostupné v databázi, jejich umístění a další specifikace jsou znázorněny v tabulce č. 10.



Tabulka 10: Umístění a další specifikace technologií v DTOZE

Formulář DTOZE - Specifikace technologií								
Technologie	Umístění	Druh umístění	Jednotka	Orientace	Parametr			
Biomasa	pozemek	rovina	m <sup>2</sup>	-	1			
FTT panel	fasáda	zábradlí, balkon, terasa	m <sup>2</sup>	J	0,95			
				JV/JZ	0,8			
				V/Z	0,7			
	střecha	šikmá	m <sup>2</sup>	J	1			
				JV/JZ	0,85			
		plochá	m <sup>2</sup>	-	1			
FTV panel	fasáda	zábradlí, balkon, terasa	m <sup>2</sup>	J	0,95			
				JV/JZ	0,8			
				Z/V	0,7			
	pozemek	svažitý	m <sup>2</sup>	J	1			
				JV/JZ	95			
		rovina	m <sup>2</sup>	-	1			
	střecha	šikmá	m <sup>2</sup>	J	1			
				JV/JZ	0,85			
		plochá	m <sup>2</sup>	-	1			
	Mikrohydretechnologie	podzemní voda	splaškové vody	m <sup>3</sup> /rok	-	-		
			dešťová voda	m <sup>3</sup> /rok	-	-		
			studna (hloubka)	m	-	-		
hladina podz. vody			m	-	-			
povrchová voda		potok	l/s	-	-			
		vodní nádrž	m <sup>3</sup>	-	-			
Tepelné čerpadlo	podzemní voda	studna (hloubka)	m	-	-			
		hladina podz. vody	m	-	-			
	povrchová voda	vodní nádrž	m <sup>3</sup>	-	-			
Tepelná čerpadlo - geotermální	pozemek	rovina	m <sup>2</sup>	-	1			
Větrná technologie	fasáda	zábradlí, balkon, terasa	m <sup>2</sup>	J	0,85			
				JV/JZ	0,85			
				V/Z	0,85			
	pozemek	rovina	m <sup>2</sup>	-	1			
				střecha	šikmá	m <sup>2</sup>	J	1
							JV/JZ	0,85
					plochá	m <sup>2</sup>	-	1

Od uživatele je získána informace o dostupných prostředcích v českých korunách (CZK) a informaci o dostupných kapacitách. Jedná-li se o plochu, je získána informace i o rozměru, u vodního hospodářství i o průtoku či výšce hladiny. Tyto informace stačí pro vyhledání vhodného řešení. Cílem je najít takové produkty, které vyhovují zadanému umístění a dostupným prostředkům. Dotaz do databáze v zjednodušeném jazyce vypadá následovně:

*Najdi všechny technologie, které lze umístit na zadané umístění. Dále najdi všechny produkty, které vyhovují nalezeným technologiím. Podle zadaných dostupných prostředků odstraň z množiny výsledků produkty u kterých je údaj cena vyšší, než je zadáno, a které nesplňují podmínku umístění (např. je-li uveden rozměr, zkontroluj zda se daný produkt vejde do rozměru zadaného uživatelem, je-li uveden minimální průtok, zkontroluj zda je splněn). Takové produkty seřaď podle ceny a zobraz jejich název, technologii, fotku pokud je dostupná a cenu pokud je dostupná.*

Uživatel smí zadat pouze jedno umístění. V případě změny se může vrátit pouze k formuláři dostupné kapacity a vyplnit umístění znovu bez vyplňování předchozích formulářů. V případě, že bude produkt splňovat technologii a cenu nebo doplňující údaje nebudou uvedeny, zařadí se na konec vyhledaných produktů s informací o této skutečnosti. Systém si dle umístění zjistí parametr *par* pro přepočet výkonu, který je použit v dalších výpočtech.

### 3.4.2 Výpočet výhodnosti investice

Stěžejním výpočtem je výpočet výhodnosti investice do konkrétního produktu. Hlavní pro tento výpočet je, aby měl produkt v databázi definované potřebné údaje. V případě, že tyto údaje budou chybět, nelze provést výpočet.

#### 3.4.2.4 Výpočet ceny za spotřebovanou kilowatthodinu

Než lze přejít k samotnému výpočtu výhodnosti konkrétní investice je nutné provést výpočet ceny za spotřebovanou kilowatthodinu (kWh). Výsledkem je cena elektrické energie, za kterou uživatel zaplatil/zaplatí (předpokládá se, že stáří údajů bude maximálně jeden rok.). Ta se vypočítá z informací získaných z formuláře v oddílu stávající infrastruktura. Konkrétně se jedná o odběr elektrické energie ze sítě, získávání tepla a cena za rok využívání. Výsledkem je cena za kWh. Výpočet se řídí dle vzorce (1), kde  $p$  je cena za jednotku (CZK/kWh),  $P_t$  je cena spotřebovaného paliva za rok (CZK) a  $C_t$  je spotřeba elektrické energie za rok (kWh):

$$p = P_t / C_t \quad (1)$$

V případě elektrické energie ze sítě je údaj rovnou v kWh, v případě tepla je nutné přepočíst počet tun na kWh. K dispozici jsou konstanty pro přepočet tun na kWh pro uvedené druhy paliv, které znázorňuje tabulka č. 11.

Tabulka 11: Výhřevnost paliv na 1 tunu materiálu (zdroj: interní materiály firmy ECO trend)

Druh paliva	Výhřevnost
Hnědé uhlí prach. Most	3255,56 kWh/t
Hnědé uhlí tříděné Most	4772,22 kWh/t
Hnědé uhlí prach. Sokolov	2913,89 kWh/t
Hnědé uhlí tříděné Sokolov	3936,11 kWh/t
Černé uhlí prach. Ostrava	6327,78 kWh/t
Černé uhlí energ. Ostrava	8113,89 kWh/t
Černé uhlí prach. Kladno	4325,00 kWh/t
Černé uhlí energ. Kladno	6280,56 kWh/t
Koks otopový	7636,11 kWh/t
Lignit	2441,67 kWh/t
Brikety	6402,78 kWh/t
LTO	11750,00 kWh/t
Dřevo kusové	4061,11 kWh/t
Dřevní štěpka	3966,67 kWh/t
Obilí - zrno	4027,78 kWh/t
Sláma obilná	4305,56 kWh/t

Cena za jednotku z údajů o teple je ale ještě ovlivněna v případě, že je vyplněno více druhů paliv. V tomto případě se vychází z průměrné ceny za jednotku ze všech paliv. Výpočet se tedy řídí dle vzorce (2), kde  $p$  je cena za jednotku (CZK/kWh),  $n$  je počet paliv,  $P_t$  je cena spotřebovaného paliva za rok (CZK),  $W_t$  je váha spotřebovaného paliva za rok (v tunách) a  $k$  je konstanta výhřevnosti (kWh/t):

$$p = \frac{\sum_{i=0}^n \frac{P_{it}}{W_{it} * k}}{n} \quad (2)$$

V případě vyplnění údajů o elektrické energii i tepla zároveň se cena ještě jednou zprůměruje.

### 3.4.2.5 Výpočet efektivnosti investice

Pro výpočet efektivnosti investice existuje několik postupů. Důležité ale je, aby dodržovaly určité zásady:<sup>8</sup>

- výpočet je založen na Cash Flow (peněžný tok, dále jen CF);
- ve výpočtu jsou zahrnuty správné položky (např. cena peněz v čase, diskont).
- výpočet počítá s v nominálních cenách, respektuje cenový vývoj jednotlivých položek příjmů a výdajů;
- ve výpočtu je vhodně zvolena doba hodnocení a zahrnuje případné důsledky projektu po skončení zvolené doby;
- počítá se s ohledem na důsledky financování.

Problémem může při počítání efektivnosti investice být, že se měří penězi. U ekologicko-energetických projektů hraje roli mnoho faktorů, které nelze vyčíslit penězi. Příkladem může být ochránění přírodního prostředí, zlepšení zdraví apod. Výpočtem efektivnosti investice lze získat informace o tom, kolik to bude stát a jaký bude mít investice ekonomický efekt, avšak do rozhodování je nutné zvážit i další neměřitelné faktory, které mají pro investora význam.<sup>9</sup>

Ze zásad uvedených výše je patrné, že je důležité stanovit dobu hodnocení a vypočítat CF za daný rok se všemi položkami, které je nutné zahrnout. Doba hodnocení bude stanovena dle životnosti daného produktu. CF je nutné složit z položek: přímá úspora  $S_t$  (která bude uspořena např. při neplacení elektrické energie ze sítě, či paliva za daný rok - výpočet z údajů, které zadal uživatel a z údajů v databázi) a provozní náklady  $C_t$  (nákup paliva, energie, instalace, údržba pozemku, apod., - výpočet z údajů, které zadal uživatel a z údajů v databázi). Vzorec (3) uvádí výpočet přímé úspory, která se vypočítá z roční výtěžnosti  $Y_t$  a ceny za jednotku, která je získána z předchozího výpočtu. U některých technologiích se stářím ubývá výkon, což je nutné zavést do výpočtu.  $Y_{bt}$  je pak výkon v předchozím roce a  $LP$  je zmiňovaný úbytek výkonu.<sup>10</sup> Některé technologie mají různý výkon dle orientace umístění,

<sup>8</sup>KNÁPEK, Jaroslav, Oldřich STARÝ a Jiří VAŠÍČEK. Zásady hodnocení ekonomické efektivnosti energetických projektů. *Metodika EFEKT* [online]. 2003 [cit. 2014-04-12]. Dostupné z: <http://efekt.xf.cz/metodikaEFEKT.pdf>. Str. 2.

<sup>9</sup>Tamtéž str. 2-3.

<sup>10</sup>Tamtéž str. 4.

je tedy nutné výkon ještě vynásobit hodnotou parametru  $par$ , kterou si systém pamatuje od vyplnění formuláře o dostupných kapacitách.

$$S_t = p * ((Y_{bt} - LP) * par) \quad (3)$$

Dále je potřeba vyčíslit provozní náklady. Ty se stanoví dle údajů od uživatele (např. u pozemku zadá, kolik ho přibližně stojí údržba za rok). Dále každý produkt může mít ještě další individuální provozní náklady (např. cena instalace, údržba, apod.). Provozní náklady získané od uživatele jsou značeny  $C_{ut}$  a  $C_{pt}$  jsou provozní náklady konkrétního produktu. Celkové provozní náklady se získají prostým součtem, jak znázorňuje vzorec (4).

$$C_t = (C_{ut} + C_{pt}) \quad (4)$$

Nyní jsou k dispozici veškeré údaje k vypočtení CF za daný rok. Vzorec (5) uvádí tento výpočet, za předpokladu, že investor využívá produkt pro vlastní potřebu, veškerou energii spotřebuje (nepřeprodává) a nevyužije investiční úvěr.

$$CF_t = S_t - C_t \quad (5)$$

Správné kritérium by mělo být založeno na maximalizaci budoucích CF. Takovým výpočtem je čistá současná hodnota ( $NPV$  - Net Present Value), která převádí CF na sčitatelnou hodnotu s ohledem na cenu peněz v čase diskontováním od zvoleného okamžiku (např. od počátku provozu). Vzorec (6) počítá s  $CF$  v daném roce, vloženou investicí  $IN$  a s diskontní sazbou  $r$ , kde  $Tv$  je doba životnosti a  $t$  je hodnocené období.<sup>11</sup>

$$NPV = \sum_{t=1}^{Tv} CF_t * (1 + r)^{-t} - IN \quad (6)$$

Důležité je stanovit správný diskont, neboli procento o které je budoucí výnos každoročně snížen při přepočtu na současnou hodnotu. Tato hodnota by měla být srovnatelná s možným výnosem alternativní investice. Používá se obvykle diskont s tzv. rizikovou přírážkou, tedy úroková sazba státních dluhopisů s dobou splatnosti srovnatelnou s dobou návratnosti uvažované investice a přírážka (kolem 2%). V případě drobných investorů (např. domácnosti) lze počítat s průměrným diskontem 3 % (alternativní investicí může být např. dlouhodobý termínovaný vklad a i riziková přírážka může být menší).<sup>12</sup> Při možnosti využití pozemku se ještě připočítá rentabilita tohoto pozemku (např. má-li investor pole v úrodné oblasti, rentabilita bude vyšší než v neúrodné). Tuto hodnotu musí zadat uživatel, jinak s ní nebude počítáno. Vzorec pro NPV je vzorcem (7), kde  $r_{pu}$  je rentabilita pozemku zadaná uživatelem v procentech.

<sup>11</sup>KNÁPEK, Jaroslav, Oldřich STARÝ a Jiří VAŠÍČEK. Zásady hodnocení ekonomické efektivnosti energetických projektů. *Metodika EFEKT* [online]. 2003 [cit. 2014-04-12]. Dostupné z: <http://efekt.xf.cz/metodikaEFEKT.pdf>. Str. 5.

<sup>12</sup>BECHNÍK, Bronislav. Fotovoltaika jako náhrada biopaliv v dopravě. *tzbinfo.cz: stavebnictví, úspory energií, technická zařízení budov* [online]. 2. 9. 2013 [cit. 2014-04-12]. Dostupné z: <http://oze.tzb-info.cz/fotovoltaika/10291-fotovoltaika-jako-nahrada-biopaliv-v-doprave>.

$$NPV = \sum_{t=1}^{Tv} CF_t * (1 + (r + r_{pu}))^{-t} - IN \quad (7)$$

Hodnota, která vyjde zahrnuje tři případy. Pokud je  $NPV$  záporné, není vhodné do projektu investovat, v případě že je  $NPV$  kladné lze investici doporučit, neboť výnos z projektu je vyšší než vložené investice. Posledním případem je, že je  $NPV$  rovno nule, což značí, že výnos z projektu je stejný jako vložená investice. U investic do technologií využívajících OZE může být tento případ, nebo i mírně záporné  $NPV$  kladným výsledkem, jelikož je nutné zohlednit zmiňované nezpenžitelné důsledky na životní prostředí. V případě, že se posuzuje více investic zároveň, vybírá se taková, která má největší  $NPV$ .

Výhodou kritéria je, že splňuje výše specifikované zásady správného hodnocení investic. Nevýhodou naopak je, že se musí správně stanovit doba životnosti a diskontní sazba.

Dalším kritériem, které lze využít pro hodnocení investic je vnitřní výnosové procento ( $IRR$  - Internal Rate of Return). Tato hodnota uvádí, jaká diskontní sazba dává právě nulovou hodnotu diskontovaného CF v dané době životnosti. Zjednoduše lze uvést, že  $IRR$  je právě taková diskontní sazba  $r$ , kdy  $NPV = 0$ . Vzorec pro vypočtení  $IRR$  je vzorcem (8).<sup>13</sup>

$$\sum_{t=1}^{Tv} CF_t * (1 + IRR)^{-t} - IN = 0 \quad (8)$$

Pro vyhodnocení zda se vyplatí investovat je opět nutné znát diskontní sazbu. V případě, že je  $IRR$  větší nebo rovno než  $r$ , je vhodné investici realizovat, v opačném případě ne. Opět platí pravidlo, že je výhodnější investice s větším  $IRR$ .

Toto kritérium též splňuje zmiňované zásady, má však mnohem více nevýhod. Z matematického hlediska to není jednoznačný výpočet, může dojít k situaci, že reálná hodnota  $IRR$  neexistuje. Dále nelze použít ve výpočtu různý diskont v různých letech, což  $NPV$  dovoluje.<sup>14</sup>

Často používané kritérium při rozhodování je doba návratnosti. Prostá doba návratnosti ( $PP$  - Payback Period) udává v jakém časovém horizontu převyší součet CF výši investice, dle vzorce (9).

$$\sum_{t=1}^{PP} CF_t - IN \stackrel{!}{=} 0 \quad (9)$$

Kritériem je ziskat menší  $PP$  než je doba životnosti. Toto kritérium ale neuvažuje cenu peněz v čase a CF po ušlé době. Není proto vhodné se dle tohoto kritéria řídit, může být ale použito jako rychlý orientační údaj.<sup>15</sup>

Vylepšeným výpočtem je diskontovaná doba návratnosti ( $DPP$  - Discounted Payback Period). Vypočítá se pomocí již zmiňované diskontní sazby  $r$  podle CF v jednotlivých rocích hodnocení dle vzorce (10).

<sup>13</sup>KNÁPEK, Jaroslav, Oldřich STARÝ a Jiří VAŠÍČEK. Zásady hodnocení ekonomické efektivnosti energetických projektů. *Metodika EFEKT* [online]. 2003 [cit. 2014-04-12]. Dostupné z: <http://efekt.xf.cz/metodikaEFEKT.pdf>. Str. 5-6.

<sup>14</sup>Hodnocení investic: Vnitřní výnosové procento (IRR). *Business Víze.cz* [online]. 9. 11. 2010 [cit. 2014-04-12]. Dostupné z: <http://www.businessvize.cz/rizeni-a-optimalizace/hodnoceni-investic-vnitřni-vynosove-procento-irr>.

<sup>15</sup>KNÁPEK, Jaroslav, Oldřich STARÝ a Jiří VAŠÍČEK. Zásady hodnocení ekonomické efektivnosti energetických projektů. *Metodika EFEKT* [online]. 2003 [cit. 2014-04-12]. Dostupné z: <http://efekt.xf.cz/metodikaEFEKT.pdf>. Str. 6.

$$\sum_{t=1}^{DPP} CF_t * (1 - r)^{-t} - IN \stackrel{!}{=} 0 \quad (10)$$

Pro tento výpočet platí stejné kritérium jako pro prostou dobu návratnosti, tedy minimalizovat *DPP*. Tento výpočet již respektuje cenu peněz v čase, avšak stále zanedbává CF po uplynutí *DPP*. Proto není vhodné hodnotit investici pouze na základě tohoto kritéria.

V rámci aplikace využijí všechny zmiňované pohledy na hodnocení investice. Uživatel bude mít tedy možnost posoudit hodnoty všech zmiňovaných kritérií. V rámci aplikace bude u každé hodnoty krátká informace, co daný výsledek udává a zda investici doporučit či nikoli. Uživatel bude upozorněn, že tato doporučení jsou pouze orientační a nemusí plně odpovídat skutečnosti.

## 4 Návrh databáze

Návrh databáze je velmi důležitý pro budoucí implementaci. Vytváří můstek mezi požadavky a implementací. Snaží se zachytit vztahy, které mohou nastat, aby bylo zaručeno, že jsou všechny požadavky správně pochopeny a budou správně naimplementovány.<sup>16</sup>

Definice metodiky z knihy *Mistrovství – databáze* zní: „Proces vytvoření návrhu, který bude podporovat celkové poslání a dílčí cíle pro požadovaný databázový systém organizace.“<sup>17</sup>

Návrh databáze je jedna z fází životního cyklu vývoje databázového systému. Předchází jí fáze plánování databáze, definice systému, sběr a analýza požadavků, které jsem popsala v předchozích kapitolách, a po ní následuje implementace databáze, testování a provozní údržba.

Proces návrhu databáze se skládá z dalších tří fází – konceptuální návrh, logický návrh a fyzický návrh. Během konceptuálního návrhu se identifikují objekty, které je v databázi nutné uchovávat, a vztahy mezi nimi. V logickém návrhu se identifikované objekty i vztahy přetváří na množinu tabulek. V poslední fázi se rozhoduje, jak je nutné tabulky fyzicky implementovat.<sup>18</sup> Podrobněji o jednotlivých fázích pojednávají následující kapitoly. Při tvorbě návrhu se řídím dle metodologie popsané v knize *Mistrovství – databáze*.<sup>19</sup> Jednotlivé kroky doplňuji tvorbou samotné databáze, dle požadavků zadavatele.

Návrh databáze níže popsaný je určen pro návrh relační databáze. Pro vizualizaci návrhu používám nástroj Enterprise Architect a objektově orientovaný jazyk UML (Unified Modeling Language).

### 4.1 Konceptuální návrh

V této fázi návrhu se vytváří konceptuální model dat neboli ER (entity relation) model. Vychází z informací, které byly ujednány v rámci požadavků, z nichž identifikuje důležité entity a relace, které je nutné zachytit v požadované databázi. V rámci tvorby modelu se dbá mimo jiné i na minimalizování redundancí. V následujících kapitolách popisují postup při vytváření konceptuálního modelu.

#### 4.1.1 Identifikace entit

Nejdříve je nutné ujasnit, jaké entity bude v databázi potřeba ukládat. Projdou se nasbírané informace od zadavatele databáze a vytvoří se slovník dat, kde je uveden název entity, popis a předpokládaný počet výskytů. K těmto entitám je důležité zahrnout i další entity, které nebyly přímo zmíněny, ale bude je nutné na základě fungování systému uvážit. Všechny entity by měly mít velmi podrobný popis a informace o datech, které budou obsahovat. Již nyní je tedy důležité mít hrubou představu o tom, jak bude systém pracovat. Obsah slovníku není neměnný a v průběhu návrhu se může ještě změnit. Slovník pro DTOZE znázorňuje tabulka č. 12.

<sup>16</sup>CONOLLY, Thomas, Carolyn 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. Str. 207.

<sup>17</sup>Tamtéž str. 206.

<sup>18</sup>Tamtéž str. 207.

<sup>19</sup>CONOLLY, Thomas, Carolyn 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.

Tabulka 12: Slovník DTOZE z předpokládaným počtem entit za rok existence

Slovník DTOZE		
Entity	Popis	Počet za rok
Formulářová položka	Je určená povinně technologií, umístěním, druhem umístění, jednotkou, nepovinně orientací a parametrem.	35
Druh technologie	Specializace technologie. Je určen unikátním názvem a popisem. Spadá povinně do určité kategorie.	10
Druh umístění	Specializace umístění. Je určen unikátním názvem a popisem. Spadá povinně do určitého umístění.	5
Firma	Firma vyrábí nebo nabízí produkty. Je určena povinně názvem, unikátní webovou stránkou, adresou. Může být zastupována jedním producentem.	50
Hodnota údaje	Určuje konkrétní hodnotu pro daný údaj a produkt. Je určena povinně hodnotou. Eviduje datum přidání	20 000
Jednotka	Jednotka rozšiřující vlastnosti údaje. Je určena povinně unikátní zkratkou a nepovinně unikátním celým názvem.	50
Kategorie	Kategorie technologií dle jednotlivých obnovitelných zdrojů energie. Je určena unikátním názvem a popisem. Obsahuje několik druhů technologií.	5
Orientace	Rozšiřuje vlastnosti formulářové položky. Je určena povinně unikátní zkratkou a unikátním názvem.	5
Parametr	Rozšiřuje vlastnosti formulářové položky. Je určen povinně hodnotou v rozmezí 0 - 1,000.	10
Produkt	Produkt nabízí firma, spravuje jej buď producent nebo správce. Obsahuje mnoho údajů určené technologií. Spadá do konkrétní technologie. Je určen povinně unikátním názvem, nepovinně fotkou a popisem. Uchovává informace, kdo jej přidal, kdo jej naposledy upravil nebo schválil. Nachází se v několika stavech dle procesu schvalování.	1 000
Přihláška	Přihlášku se může běžný uživatel stát producentem. Musí uvést povinně informace o sobě (jméno a platný e-mail) a o firmě, kterou chce zastupovat (název, adresu). Dále je v přihlášce uchována informace o datu poslání či schválení.	10
Technologie	Technologií je určen každý produkt. Obsahuje povinně unikátní název a nepovinně popis. Spadá do druhu technologie.	11
Typ údaje	Specializuje údaj. Je určen povinně unikátním názvem.	5
Údaj	Z údajů se skládají informace o jednotlivých produktech. Je určen povinně unikátním názvem, dále může být viditelný či neviditelný. Je vázán na technologii a spadá do určitého typu údaje. Dále je určen nepovinně jednotkou.	4 000
Umístění	Umístění je určeno povinně unikátním názvem a nepovinně popisem.	5
Uživatel	Uživatel může být buď správce nebo producent. Přihlašuje se do systému pomocí unikátního e-mailu a hesla. Eviduje se datum posledního přihlášení.	10
Změna	Změna umožňuje producentovi editovat produkty (informace nebo údaje). Je určena povinně předmětem, popisem, navrženou hodnotou. Nachází se v různých stavech, dle toho v jakém procesu schválení se nachází. Může obsahovat zprávu od správce. Eviduje se datum poslání a datum schválení/zamítnutí.	100

#### 4.1.2 Identifikování relací

Dalším krokem je identifikace relací (vztahů) mezi jednotlivými entitami. Pro definování vztahu se vychází ze slovníku, či je nutné se opět vrátit k požadavkům na databázi. Typicky se relace označují slovesy.

Vztah je definován zpravidla mezi dvěma entitami. Vztahem může vzniknout i vazební entita, která obsahuje informace společné pro obě entity ve vztahu. Lze definovat vztah nad jednou entitou. Pak se takový vztah nazývá samoreferenční nebo také rekurzivní. Vzniknout může i n-ární vztah, tedy ve vztahu jsou více než dvě entity, ale tento typ vztahu je výjimečný.<sup>20</sup>

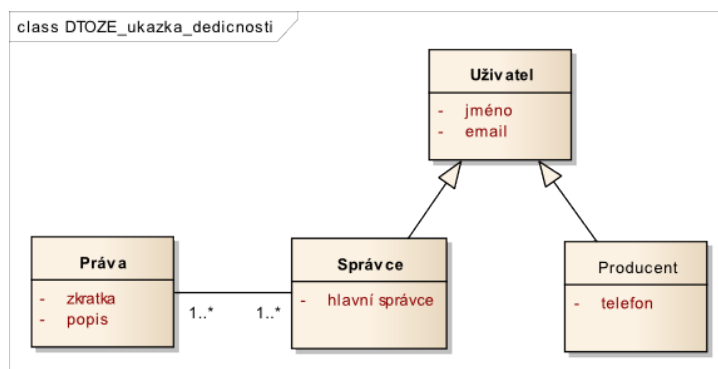
<sup>20</sup>HERNANDEZ, Michael J. a HOLOWCZAK. Návrh databází. 1. vydání. Praha: Grada, 2006, 408 s. Profesional. ISBN 80-247-0900-7. Str. 227-232.



Speciálním vztahem je dědičnost - tedy vztah nadtřída/podtřída. Nadtřída je entita, která obsahuje obecné atributy, společné pro více entit. Tyto entity se nazývají podtřídy a dědí atributy nadtříd. Nadtřída i podtřída je nutné v diagramu samostatně reprezentovat. Vztah mezi nadtřídou a kteroukoli její podtřídou je 1:1 a značí se plnou šipkou bez viditelných multiplicit. Každá podtřída je členem nadtříd, ale s konkrétnější rolí.<sup>21</sup>

Např. v návrhu DTOZE existuje entita uživatel, která má dvě role - producent a správce. Uživatelé mají mnoho společných vlastností (jméno, e-mail, heslo, atd.) a několik specifických pro danou roli (správce může upravovat všechny produkty, správce pouze ty, které přidal, atd.). V tomto případě se neliší v attributech, pouze ve vztazích a proto není nutné dědičnost zavádět. V případě, že by producent musel uvést telefon a správce by měl navíc vazby na práva k přístupu k úpravě dat, bylo by vhodné tuto skutečnost implementovat skrze dědičnost. Ukázka implementace dědičnosti je znázorněna na obrázku č. 8.

Obrázek 8: Ukázka vztahu dědičnosti



V případě zachycení pouze jednou entitou, by se do tabulky zanašely hodnoty NULL<sup>22</sup> ke sloupcům, které správce nebo producent nevyužije. To je jedním z hlavních důvodů, proč se dědičnost zavádí. Druhým důležitým důvodem je zpřehlednění diagramu, kdy je zřejmé, s čím je daná entita ve vztahu. V případě, že by vše bylo v jedné tabulce Uživatel, měla by tato entita vztah se vším a nebylo by jasné dané, která role je ve vztahu s čím.

Součástí relací je omezení multiplicity, která vymezují typy vztahů mezi relacemi. Těmito typy jsou vztahy 1:1 (každý záznam první entity je vázán právě s jedním záznamem druhé entity), 1:N (jeden záznam první entity má jeden nebo více záznamů druhé entity) a N:M (jeden záznam první entity může mít několik záznamů druhé entity a zároveň druhý záznam entity může mít několik záznamů první entity). Multiplicity pak mohou být specifikovány několika typy, dle toho zda je multiplicita nepovinná (0..1, 0..\*) či povinná (1..1, 1..\*<sup>23</sup>).

Relace se nejdříve zanesou do tabulky a následně je vhodné je znázornit v diagramu. Vztahy entit DTOZE jsou vypsány v tabulce č. 13 a na obrázku č. 9 jsou znázorněny v diagramu.

<sup>21</sup>CONOLLY, Thomas, Carolyn 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. Str. 178-179.

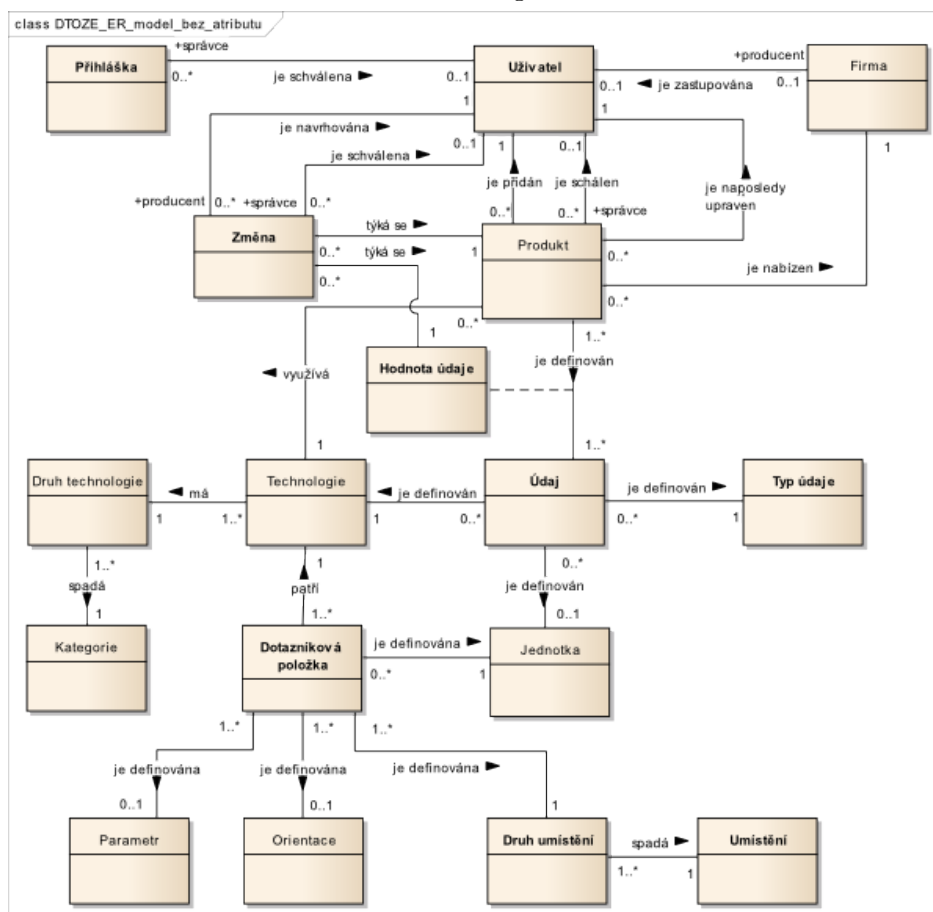
<sup>22</sup>NULL je nedefinovaná hodnota, která ale není číselná ani textová. Hodnota je to neznámá či neplatná pro daný záznam. CONOLLY, Thomas, Carolyn BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Vyd. 1. Brno: Computer Press, 2009, 584 s. ISBN 978-80-251-2328-7. Str. 68.

<sup>23</sup>HERNANDEZ, Michael J. a HOLOWCZAK. *Návrh databází*. 1. vydání. Praha: Grada, 2006, 408 s. Profesional. ISBN 80-247-0900-7. Str. 227-232.

Tabulka 13: Vztahy mezi entitami DTOZE

Vztahy mezi entitami s multiplicitami				
Druh technologie	1..*	spadá	1	Kategorie
Druh umístění	1..*	spadá	1	Umístění
Firma	0..1	je zastupována	0..1	Uživatel
Produkt	0..*	je přidán	1	Uživatel
Produkt	0..*	je naposledy upraven	1	Uživatel
Produkt	0..*	je schválen	0..1	Uživatel
Produkt	1..*	je definován	1..*	Údaj
Produkt	0..*	je nabízen	1	Firma
Produkt	0..*	využívá	1	Technologie
Přihláška	0..*	je schválena	0..1	Uživatel
Technologie	1..*	má	1	Druh technologie
Údaj	0..*	je definován	1	Technologie
Údaj	0..*	je definován	1	Typ údaje
Údaj	0..*	je definován	0..1	Jednotka
Dotazníková položka	1..*	patří	1	Technologie
Dotazníková položka	1..*	je definována	0..1	Parametr
Dotazníková položka	1..*	je definována	0..1	Orientace
Dotazníková položka	1..*	je definována	1	Druh umístění
Dotazníková položka	0..*	je definována	1	Jednotka
Změna	0..*	je navrhována	1	Uživatel
Změna	0..*	je schválena	0..1	Uživatel
Změna	0..*	týká se	0..1	Produkt
Změna	0..*	týká se	0..1	Produkt, Údaj

Obrázek 9: ER diagram DTOZE



### 4.1.3 Určení atributů a primárních klíčů

Žádná entita nemá význam bez dalších informací. Tyto informace se shromažďují pomocí atributů. K určení atributů je nutné opět prozkoumat slovník dat. Atributy jsou např. vlastnosti, jakost, charakteristika apod. Dělí se na jednoduché (odkazuje pouze na jednu hodnotu), složené (odkazuje na několik hodnot), odvozené (jejich hodnota je určena pomocí jiných atributů) a atributy relací (váže se k relaci, nikoli pouze k entitě).<sup>24</sup>

Mezi zvolenými atributy se vybírají kandidáti na primární klíč, který jednoznačně identifikuje každý záznam dané entity. Může to být jeden atribut, ale i složený. Takový kandidát nesmí nabývat NULL hodnoty. Z kandidátů se pak vybírá jeden primární klíč na základě následujících kritérií: kandidát s minimální množinou atributů; kandidát, u něhož je nejméně pravděpodobná změna hodnot; kandidát s nejmenším počtem znaků; kandidát s nejmenší maximální hodnotou. Pokud neexistuje mezi existujícími atributy vhodný kandidát na primární klíč, je nutné vytvořit umělý primární klíč, např. číselný atribut ID (od slova IDentification).<sup>25</sup>

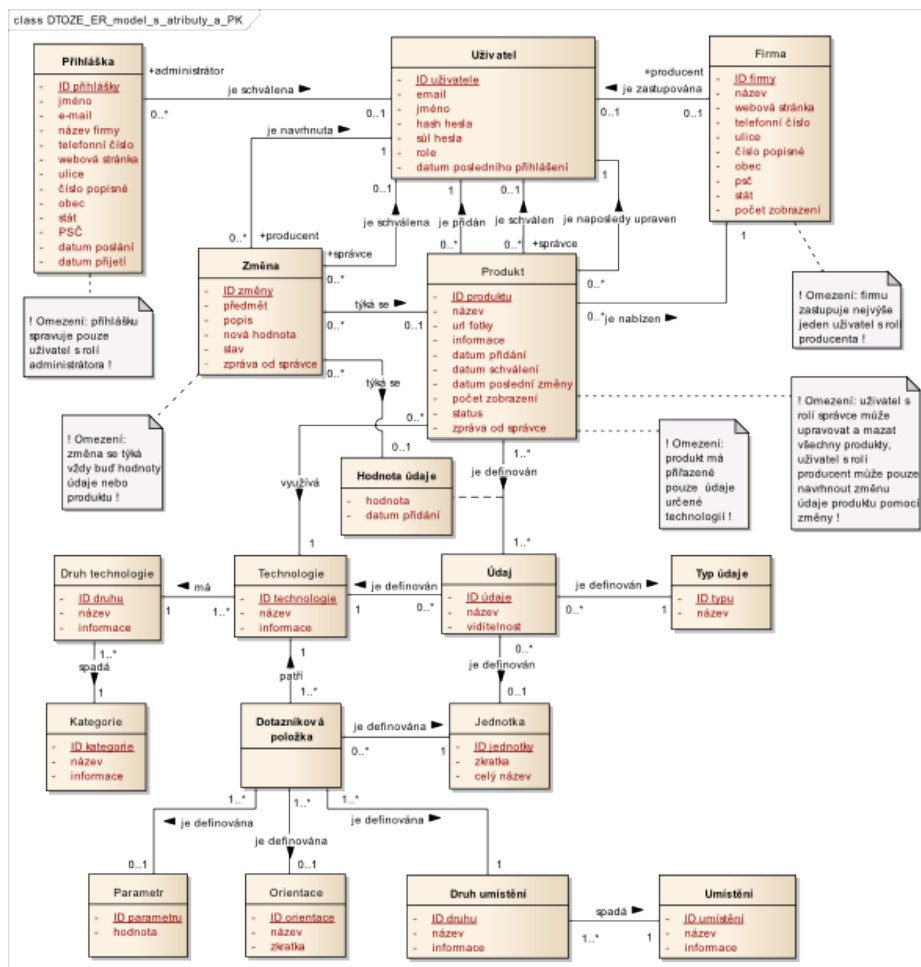
Na obrázku č. 10 je zobrazen ER diagram již s atributy a určenými primárními klíči. Při určování primárních klíčů se mi nepodařilo najít vhodného kandidáta, a proto je u všech entit určen umělý klíč. Uvažovaným kandidátem byl u některých entit název, ale u něho nelze zaručit, že bude jednoduchý a neproměnlivý. Dále byl kandidátem na primární klíč např. atribut hodnota entity Parametr. Ten je sice unikátní a krátký, jelikož se jedná o číslo, ale je vyjádřen číslem s desetinnou čárkou a též nelze zaručit, že nebude proměnlivý.

V diagramu jsou navíc zaneseny poznámky o omezení, které vyplynuly z požadavků, avšak v rámci vztahů je nelze zanést do modelu. Tyto poznámky jsou důležité pro dále pro logický a fyzický návrh či pro samotnou implementaci.

<sup>24</sup> CONOLLY, Thomas, Carolyn 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. Str. 214–216.

<sup>25</sup> Tamtéž str. 219.

Obrázek 10: ER diagram s atributy a vyznačenými primárními klíči (podtržené)



#### 4.1.4 Kontrola redundance

Splněním předchozích kroků je model téměř hotový. Jelikož je model postupně rozšiřován, je možné, že do něj byly zanesly redundance - nadbytečně zaznamenané informace. Ty mohou být zaneseny buď v entitách, nebo v relacích. K odstranění případných redundancí v entitách je nutné zpětně prozkoumat vztahy 1:1 a atributy jednotlivých entit. Tam by se mohla omylem vyskytnout entita stejná jako jiná, pouze pojmenovaná synonymem. Relace je redundantní pokud vyjadřuje informaci, kterou lze získat prostřednictvím jiné relace. V modelu se vyhledávají entity, které mají několik relací a znovu se posuzuje význam a účel.<sup>26</sup>

U mého návrhu se nevyskytují žádné vztahy 1:1 a problém jsem našla pouze u entity Přihláška, kde je mnoho dat zaneseno redundantně zároveň v entitách Uživatel a Firma. Tato redundance je ale záměr. Je nutné ukládat poslané přihlášky odděleně od zveřejněných dat, jelikož je pravděpodobné, že všechny přihlášky nebudou přijaty a zaneseny rovnou do databáze, jelikož nebudou splňovat podmínky pro přijetí.

<sup>26</sup> CONOLLY, Thomas, Carolyn 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. Str. 220–222.

## 4.2 Logický návrh

Jestliže je hotový ER model (nebo také ER diagram), lze začít pracovat na logickém návrhu. Cílem je vytvořit detailní popis tabulek na základě entit z konceptuálního modelu. Dále se tabulky normalizují, aby se opět zamezilo redundanci, tentokrát na úrovni sloupců tabulek. U logického návrhu se držím následujícího postupu.

### 4.2.1 Struktura tabulek

V této fázi se připraví struktura tabulek, která již bude použita ve fyzické databázi. Každá entita ER modelu nyní vyjadřuje tabulku a její atributy vyjadřují sloupce tabulky. K popsání struktury se používá jazyk DBDL (Database Definition Language), díky kterému lze strukturu tabulky vyjádřit textem. Pravidla jsou následující: nejdříve se uvede jméno tabulky a poté v závorce názvy sloupců. Primární klíče jsou opět zobrazeny podtrženě. V případě DTOZE je struktura vyznačena na obrázku č. 11.

Obrázek 11: Struktura DTOZE pomocí DBDL

Druh technologie	(	<u>ID druhu</u>	,	název	,	informace	)								
Druh umístění	(	<u>ID umístění</u>	,	název	,	informace	)								
Firma	(	<u>ID firmy</u>	,	název	,	webová stránka	,	telefonní číslo	,	ulice	,	číslo popisné	,	obec	,
		PSČ	,	stát	,	počet zobrazení	)								
Jednotka	(	<u>ID jednotky</u>	,	zkratka	,	název	)								
Kategorie	(	<u>ID kategorie</u>	,	název	,	informace	)								
Orientace	(	<u>ID orientace</u>	,	zkratka	,	název	)								
Parametr	(	<u>ID parametru</u>	,	hodnota	)										
Produkt	(	<u>ID produktu</u>	,	název	,	uri fotky	,	informace	,	datum přidání	,	datum schválení	,	datum posl. změny	,
		počet zobrazení	,	status	,	zpráva od správce	)								
Přihláška	(	<u>ID přihlášky</u>	,	jméno	,	e-mail	,	název firmy	,	tel. na firmu	,	web. stránka	,	ulice	,
		číslo popis.	,	obec	,	stát	,	PSČ	,	datum posláni	,	datum přijetí	)		
Technologie	(	<u>ID technologie</u>	,	název	,	informace	)								
Typ údaje	(	<u>ID typu údaje</u>	,	název	)										
Údaj	(	<u>ID údaje</u>	,	název	,	viditelnost	)								
Umístění	(	<u>ID umístění</u>	,	název	,	informace	)								
Uživatel	(	<u>ID uživatele</u>	,	email	,	jméno	,	hash hesla	,	súl hesla	,	role	,	datum posl. přihláš.	)
Změna	(	<u>ID změny</u>	,	předmět	,	popis	,	nová hodnota	,	stav	,	zpráva od správce	,	datum posláni	,
		datum schválení	)												

Při zaznamenávání relací přibudou k tabulkám ještě další sloupce, a to cizí klíče. Ty se identifikují na základě vztahu tabulek. Tabulky se podle vztahu rozlišují na rodičovské a dceřiné. Primární klíč rodičovské tabulky se přenáší jako cizí klíč (nový sloupec) do dceřiné tabulky. Je opět několik pravidel pro tvorbu cizích klíčů, kompletní shrnutí znázorňuje tabulka č. 14<sup>27</sup>. Aplikování na strukturu DTOZE v jazyce DBDL pak znázorňuje obrázek č. 12. Cizí klíče jsou zobrazeny tučně.

<sup>27</sup> CONOLLY, Thomas, Carolyn 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. Str. 242.

Tabulka 14: Shrnutí reprezentace entit, relací a atributů s více hodnotami pomocí tabulek

Entita/relace	Mapování
<b>Silná nebo slabá entita</b>	Vytvoření tabulky, která obsahuje všechny jednoduché atributy.
<b>Binární relace 1:*</b>	Umístění kopie primárního klíče entity na straně "jedné" do tabulky reprezentující entitu na straně "více". Všechny atributy relace se také umístí na stranu "více".
<b>Rekurzivní relace 1:*</b>	Protože entity na straně "jedné" i "více" jsou stejné, tabulka reprezentující entitu získá kopii primárního klíče, který je přejmenován a také atributy relace.
<b>Binární relace 1:1:</b>	
Povinná participace na obou stranách	Spojení entit do jedné tabulky.
Povinná participace na jedné straně	Umístění kopie primárního klíče entity s nepovinnou participací tabulky reprezentující entitu s povinnou participací. Všechny atributy relace se umístí také do tabulky reprezentující entitu s povinnou participací.
Nepovinné participace na obou stranách	Pokud nemáme další informace, umístíme kopii primárního klíče některé entity do tabulky druhé entity. Pokud máme více informací, entitu, která se více blíží povinné participaci, stanovíme jako dceřinou.
<b>Binární relace **:*</b>	
Komplexní relace	Vytvoření tabulky pro reprezentující relace a začlenění všech atributů spojených s relací. Umístění kopie primárního klíče z každé rodičovské entity do nové tabulky, kde bude fungovat jako cizí klíč.
Atribut s více hodnotami	Vytvoření tabulky při reprezentaci atributu a umístění kopie primárního klíče rodičovské entity do nové tabulky, kde bude fungovat jako cizí klíč.

Obrázek 12: Struktura DTOZE pomocí DBDL po zavedení cizích klíčů

Druh technologie	( <u>ID druhu</u> , název , informace , <u>ID technologie</u> )
Druh umístění	( <u>ID umístění</u> , název , informace , <u>ID umístění</u> )
Firma	( <u>ID firmy</u> , název , webová stránka , telefonní číslo , ulice , číslo popisné , obec , PSČ , stát , počet zobrazení , <u>ID uživatele</u> )
Jednotka	( <u>ID jednotky</u> , zkratka , název )
Kategorie	( <u>ID kategorie</u> , název , informace )
Orientace	( <u>ID orientace</u> , zkratka , název )
Parametr	( <u>ID parametru</u> , hodnota )
Produkt	( <u>ID produktu</u> , název , uri fotky , informace , datum přidání , datum schválení , datum posl. změny , počet zobrazení , status , zpráva od správce , <u>ID uživatele</u> , <u>ID uživatele</u> , <u>ID firmy</u> , <u>ID technologie</u> )
Příhláška	( <u>ID příhlášky</u> , jméno , e-mail , název firmy , tel. na firmu , web. stránka , ulice , číslo popis. , obec , stát , PSČ , datum poslání , datum přijetí , <u>ID uživatele</u> )
Technologie	( <u>ID technologie</u> , název , informace , <u>ID druhu technol.</u> )
Typ údaje	( <u>ID typu údaje</u> , název )
Údaj	( <u>ID údaje</u> , název , viditelnost , <u>ID technologie</u> , <u>ID typu údaje</u> , <u>ID jednotky</u> )
Umístění	( <u>ID umístění</u> , název , informace )
Uživatel	( <u>ID uživatele</u> , email , jméno , hash hesla , súl hesla , role , datum posl. přihláš. )
Změna	( <u>ID změny</u> , předmět , popis , nová hodnota , stav , zpráva od správce , datum poslání , datum schválení , <u>ID uživatele</u> , <u>ID uživatele</u> , <u>ID produktu</u> , <u>ID údaje</u> )

Dále vznikla vazební tabulka Hodnota údaje kvůli vztahu \*\*:\*. Primární klíč je složený a zároveň jsou atributy ID produktu a ID údaje. také cizí klíče.  
Hodnota údaje ( ID produktu , ID údaje , hodnota , datum přidání )

Další netradiční vazební tabulkou je Dotazníková položka, která sdružuje kombinace několika výčtových entit dohromady. Primární klíč je též složený.  
Dotazníková položka ( ID technologie , ID parametru , ID orientace , ID druhu umístění , ID jednotky )

#### 4.2.2 Kontrola tabulek pomocí normalizace

Vytvořené tabulky a jejich relace je nyní nutné prověřit, zda neporušují některé z normálních forem. Normalizace je proces, při kterém se relace upravují za účelem jednodušší práce a lepší manipulace s daty, zabránění redundance a lepší konzistence dat. Definovány jsou tři základní normální formy: 1NF, 2NF, 3NF.<sup>28</sup> Čím vyšší normu logický návrh splňuje, tím se zpravidla lépe pracuje s daty v databázi.

Tabulka je v první normální formě (1NF), pokud neexistuje pole tabulky, které by obsahovalo více než jednu hodnotu. Např. pokud je v tabulce uchována hodnota adresy v jednom poli (tedy pole obsahuje ulici, číslo popisné, poštovní a směrovací číslo a město), pak taková tabulka není v 1NF. Řešením je vytvoření nové tabulky a vytvoření relace s původní tabulkou.

Aby byla tabulka v druhé normální formě (2NF), musí všechna data záviset na celém klíči. V případě, že bude nutné upravit pole, které není závislé na celém klíči, bude se muset změna zanášet na několik míst. To je ale nežádoucí. Řešením je rozložení původní tabulky na více tabulek, kde se závislost na ne celém klíči vymění za cizí klíč.

Třetí normální forma (3NF) udává, že všechny neklíčové atributy musí být navzájem nezávislé. Pokud by například byly v tabulce atributy primární klíč ID osoby, jméno osoby, pobočka firmy, ve které pracuje a adresa této pobočky, pak není tabulka ve 3NF, jelikož atribut adresa pobočky je závislá na atributu pobočky. Řešením je opět rozdělení na více tabulek.<sup>29</sup>

Existují další normy: Boyce-Coddova normální forma (BCNF), čtvrtá normální forma 4NF a pátá normální forma (5NF).<sup>30</sup> Tyto formy identifikují a řeší málo frekventované problémy, a proto je zde neuvádím.

Dle postupů zmíněných výše je velice pravděpodobné, že se struktura tabulek po normalizaci může ještě změnit. Změny se zavedou do struktur tabulek a dále se s nimi pracuje. Při vytváření struktur tabulek jsem již na normalizaci intuitivně myslela, a proto po přezkoumání tabulek DTOZE nedošlo ke změnám. Našla jsem však několik atributů, které by mohly potencionálně porušovat některou z forem. Na obrázku č. 13 je zobrazeno, v jakých normách jsou tabulky DTOZE, a u kterých je podezření na porušení některých z forem. Dále vysvětlují důvody, proč ke změnám tabulek nedošlo.

<sup>28</sup>KULHAN, Jakub. Normalizace relačních databází. *Programujte.com* [online]. 2008 [cit. 2013-12-10]. Dostupné z: <http://programujte.com/clanek/2008071900-normalizace-relacnich-databazi/>.

<sup>29</sup>SKŘIVAN, Jaromír. Databáze a jazyk SQL. *Interval.cz* [online]. 2000 [cit. 2013-12-10]. Dostupné z: <http://interval.cz/clanky/databaze-a-jazyk-sql/>.

<sup>30</sup>CONOLLY, Thomas, Carolyn 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. Str. 199.

Obrázek 13: Normalizace tabulek DTOZE

										1 NF	2 NF	3 NF																				
Druh technologie	(	ID druhu	,	název	,	informace	,	ID technologie	)	x	x	x																				
Druh umístění	(	ID umístění	,	název	,	informace	,	ID umístění	)	x	x	x																				
Firma	(	ID firmy	,	název	,	webová stránka	,	telefonní číslo	,	ulice	,	číslo popisné	,	obec	,	?	x	!														
		PSČ	,	stát	,	počet zobrazení	,	ID uživatele	)																							
Jednotka	(	ID jednotky	,	zkratka	,	název	)			x	x	x																				
Kategorie	(	ID kategorie	,	název	,	informace	)			x	x	x																				
Orientace	(	ID orientace	,	zkratka	,	název	)			x	x	x																				
Parametr	(	ID parametru	,	hodnota	)					x	x	x																				
Produkt	(	ID produktu	,	název	,	uri fotky	,	informace	,	datum přidání	,	datum schválení	,	datum posl. změny	,	počet zobrazení	,	status	,	zpráva od správce	,	ID uživatele	,	ID uživatele	,	ID firmy	,	ID technologie	)	x	x	x
Příhláška	(	ID příhlášky	,	jméno	,	e-mail	,	název firmy	,	tel. na firmu	,	web. stránka	,	ulice	,	?	!	!														
		číslo popis.	,	obec	,	stát	,	PSČ	,	datum posláni	,	datum přijetí	,	ID uživatele	)																	
Technologie	(	ID technologie	,	název	,	informace	,	ID druhu technol.	)	x	x	x																				
Typ údaje	(	ID typu údaje	,	název	)					x	x	x																				
Údaj	(	ID údaje	,	název	,	viditelnost	,	ID technologie	,	ID typu údaje	,	ID jednotky	)	x	x	x																
Umístění	(	ID umístění	,	název	,	informace	)			x	x	x																				
Uživatel	(	ID uživatele	,	email	,	jméno	,	hash hesla	,	sůl hesla	,	role	,	datum posl. přihláš.	)	?	x	x														
Změna	(	ID změny	,	předmět	,	popis	,	nová hodnota	,	stav	,	zpráva od správce	,	datum posláni	)	x	x	x														
		datum schválení	,	ID uživatele	,	ID uživatele	,	ID produktu	,	ID údaje	)																					
Hodnota údaje	(	ID produktu	,	ID údaje	,	hodnota	,	datum přidání	)	x	x	x																				
Dotazníková položka	(	ID technologie	,	ID parametru	,	ID orientace	,	ID druhu umístění	,	ID jednotky	)	x	x	x																		

Dnes je již zcela běžné, že má člověk více telefonů či e-mailů. Mohlo by tedy dojít k situaci, že uživatel při registraci nebude vědět, které údaje zadat. Řešením by bylo vytvořit další tabulky, které by umožňovaly konkrétnímu záznamu mít více zmíněných atributů. Uvažila jsem toto rozdělení za zbytečné, jelikož tyto kontakty mají sloužit uživatelům k rychlému zorientování, kam se obrátit v případě zájmu o produkt. V případě, že by bylo uvedeno více telefonních čísel či e-mailových adres, musel by uživatel řešit dilema na jaké se obrátit. V databázi bude tedy povoleno mít pouze jeden kontaktní e-mail a jedno telefonní číslo u entit, kterých se to týká.

U tabulky Příhlášky a Firmy se může zdát, že porušuje 2NF, jelikož obsahuje informace o firmě, které jsou závislé na názvu firmy. Důležitý je zde fakt, že tabulka přihlášek slouží k uchování všech dat zadaných uživatelem, který projevil zájem stát se producentem. Tato data by se již neměla po zapsání měnit, pouze se z nich bude čerpat při přidávání záznamů do tabulek Producent, Firma. Tato tabulka není v 2NF (a tedy ani ve 3NF), ale struktura naší databáze to nevdává. Dále ani entita Firma není v 2NF ani 3NF ze stejného důvodu. Firma obsahuje položky adresy. Ale na jedné adrese může být více firem, bude tedy docházet k redundancím. Navíc ulice, město a stát je určen PSČ, to by mělo být také v jiné tabulce. V tomto okamžiku je důležité, jak bude k těmto datům přistupováno a kolik firem bude v databázi zastoupeno. Předpokládá se, že firem v databázi mnoho nebude, ale přistupovat k detailům o firmách se bude relativně často. Proto je výhodnější opakovat některá data než je dohledávat v jiné tabulce.

Při normalizaci se dbá nejvíce na to, aby se data neopakovala a byla snadno upravitelná. V některých případech ale může být výhodnější jiné řešení. Tím se ale zabývám podrobněji až při fyzickém návrhu v kapitole č. 4.3.6 o denormalizaci. Při logickém návrhu není ještě důležité fungování databázového systému i když je už vhodné o efektivním uložení dat přemýšlet během celého navrhování databáze.



### 4.2.3 Kontrola integritních omezení

Integritní omezení se zavádějí pro zaručení úplnosti, přesnosti a konzistence dat v databázi. Jedná se o:

- Požadovaná data – některé sloupce nesmí obsahovat hodnotu NULL.
- Omezení množin přípustných hodnot – stanovení množiny hodnot, kterých může sloupec nabývat.
- Multiplicita – přesně definovaná kardinalita (mohutnost) u jednotlivých relací.
- Referenční integrita – souvisí s cizími klíči, pokud cizí klíč obsahuje hodnotu, pak tato hodnota musí odkazovat na existující záznam v rodičovské tabulce.
- Jiná integritní omezení – stanoví se na základě požadavků na databázi, platí pro konkrétní model.<sup>31</sup>

Konkrétní nastavení těchto omezení už ale závisí na DBMS (database management system), který se použije pro implementaci. Je vhodné ale zvážit již předem, jak data omezit. Některá z omezení, která lze nyní vyhodnotit jsou zaznamenána v logickém návrhu v tabulce č. 15. Jedná se o omezení NULL hodnot, unikátnosti sloupců, předpokládané množiny hodnot a velikosti.

Dále byla vyhodnocena následující omezení, která se týkají více sloupců či tabulek nebo je nutné je vymezit speciálně. Na tato omezení je nutné brát ohled při fyzickém návrhu popř. při implementaci aplikace.

- Přihlášku může spravovat pouze jeden nebo žádný Uživatel s rolí správce.
- Firmu může zastupovat pouze jeden nebo žádný Uživatel s rolí producenta.
- Uživatel s rolí správce může upravovat a mazat všechny Produkty, Uživatel s rolí producent může pouze navrhnout editaci Údaje produktu pomocí Změny.
- Změna se týká vždy buď Hodnoty údaje nebo Produktu.
- Produkt má přiřazené pouze údaje určené technologií.
- Datum přidání (poslání) musí být vždy stejné nebo starší než datum schválení. Datum schválení musí být vždy stejné nebo starší než datum poslední úpravy.
- E-mail musí být v platném formátu.
- URL adresy musí být v platném formátu.

<sup>31</sup>CONOLLY, Thomas, Carolyn 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. Str. 247-250.

Tabulka 15: Logický návrh tabulek společně s integritními omezeními.

Logický model DTOZE											
	klíč	unikátní	NULL	množina hodnot	max. délka		klíč	unikátní	NULL	množina hodnot	max. délka
<b>Firma</b>						<b>Parametr</b>					
ID firmy	PK	ANO	NE	{1..n}	-	ID parametru	PK	ANO	NE	{1..n}	-
název	-	NE	NE	text	50	hodnota	-	ANO	NE	{0,000..1,000}	3
webová stránka	-	ANO	ANO	NULL nebo text	neomezeno	<b>Přihláška</b>					
telefonní číslo	-	NE	ANO	NULL nebo text	20	ID přihlášky	PK	ANO	NE	{1..n}	-
ulice	-	NE	NE	krátký text	20	jméno	-	NE	NE	krátký text	50
číslo popisné	-	NE	NE	krátký text	10	e-mail	-	ANO	NE	krátký text	50
obec	-	NE	NE	krátký text	50	název firmy	-	NE	NE	krátký text	50
PSČ	-	NE	NE	krátký text	10	tel. na firmu	-	NE	NE	krátký text	20
stát	-	NE	NE	krátký text	50	web. stránka	-	ANO	ANO	NULL nebo text	neomezeno
počet zobrazení	-	NE	NE	{1..n}	-	ulice	-	NE	NE	krátký text	20
ID uživatele (producent)	FK	ANO	ANO	NULL nebo {1..n}	-	číslo popis.	-	NE	NE	krátký text	10
<b>Orientace</b>						obec	-	NE	NE	krátký text	50
ID orientace	PK	ANO	NE	{1..n}	-	PSČ	-	NE	NE	krátký text	10
zkratka	-	ANO	NE	krátký text	10	stát	-	NE	NE	krátký text	50
název	-	ANO	NE	krátký text	30	datum posláni	-	NE	NE	datum	-
<b>Dotazníková položka</b>						datum přijetí	-	NE	ANO	NULL nebo datum	-
ID technologie	PK, FK	NE	NE	{1..n}	-	ID uživatele (správce)	FK	NE	ANO	NULL nebo {1..n}	-
ID parametru	PK, FK	NE	NE	0 nebo {1..n}	-	<b>Hodnota údaje</b>					
ID orientace	PK, FK	NE	NE	1 nebo {1..n}	-	ID produktu	PK, FK	NE	NE	{1..n}	-
ID druhu umístění	PK, FK	NE	NE	{1..n}	-	ID údaje	PK, FK	NE	NE	{1..n}	s
ID jednotky	PK, FK	NE	NE	{1..n}	-	hodnota	-	NE	NE	krátký text	30
<b>Uživatel</b>						datum přidání	-	NE	NE	datum	-
ID uživatele	PK	ANO	NE	{1..n}	-	<b>Údaj</b>					
email	-	ANO	NE	krátký text	50	ID údaje	PK	ANO	NE	{1..n}	-
jméno	-	NE	NE	krátký text	50	název	-	NE	NE	krátký text	30
hash hesla	-	NE	NE	krátký text	50	viditelnost	-	NE	NE	{FALSE, TRUE}	5
sůl hesla	-	NE	NE	krátký text	50	ID technologie	FK	NE	NE	{1..n}	-
role	-	NE	NE	{admin, producent}	10	ID typu údaje	FK	NE	NE	{1..n}	-
datum posl. přihláš.	-	NE	ANO	datum	-	ID jednotky	FK	NE	ANO	NULL nebo {1..n}	-
<b>Umístění</b>						<b>Druh technologie</b>					
ID umístění	PK	ANO	NE	{1..n}	-	ID druhu	PK	ANO	NE	{1..n}	-
název	-	ANO	NE	krátký text	30	název	-	ANO	NE	krátký text	30
informace	-	NE	ANO	NULL nebo text	neomezeno	informace	-	NE	ANO	NULL nebo text	neomezeno
<b>Technologie</b>						ID kategorie	FK	NE	NE	{1..n}	-
ID technologie	PK	ANO	NE	{1..n}	-	<b>Kategorie</b>					
název	-	ANO	NE	krátký text	30	ID kategorie	PK	ANO	NE	{1..n}	-
informace	-	NE	ANO	NULL nebo text	neomezeno	název	-	ANO	NE	krátký text	30
ID druhu technol.	FK	NE	NE	{1..n}	-	informace	-	NE	ANO	NULL nebo text	neomezeno
<b>Jednotka</b>						<b>Typ údaje</b>					
ID jednotky	PK	ANO	NE	{1..n}	-	ID typu údaje	PK	ANO	NE	{1..n}	-
zkratka	-	ANO	NE	krátký text	10	název	-	ANO	NE	text	30
název	-	ANO	ANO	NULL nebo text	30	<b>Změna</b>					
<b>Produkt</b>						ID změny	PK	ANO	NE	{1..n}	-
ID produktu	PK	ANO	NE	{1..n}	-	předmět	-	NE	NE	krátký text	50
název	-	ANO	NE	krátký text	30	popis	-	NE	NE	text	neomezeno
url fotky	-	NE	ANO	NULL nebo text	neomezeno	nová hodnota	-	NE	NE	krátký text	30
informace	-	NE	ANO	NULL nebo text	neomezeno	stav	-	NE	NE	{OK, KO, WAIT}	4
datum přidání	-	NE	NE	datum	-	zpráva od správce	-	NE	NE	text	neomezeno
datum schválení	-	NE	ANO	NULL nebo datum	-	ID uživatele (producent)	FK	NE	NE	{1..n}	-
datum posl. změny	-	NE	NE	datum	-	ID uživatele (správce)	FK	NE	ANO	{1..n}	-
počet zobrazení	-	NE	NE	0 nebo {1..n}	-	ID produktu	FK	NE	NE	{1..n}	-
stav	-	NE	NE	{OK, WAIT, KO}	4	ID údaje	FK	NE	ANO	NULL nebo {1..n}	-
zpráva od správce	-	NE	NE	text	neomezeno	datum přidání	-	NE	NE	datum	-
ID uživatele (přidal)	FK	NE	NE	{1..n}	-	datum schválení	-	NE	ANO	NULL nebo datum	-
ID uživatele (schválil)	FK	NE	NE	{1..n}	-	<b>Druh umístění</b>					
ID uživatele (upravil)	FK	NE	NE	{1..n}	-	ID umístění	PK	ANO	NE	{1..n}	-
ID firmy	FK	NE	NE	{1..n}	-	název	-	ANO	NE	krátký text	30
ID technologie	FK	NE	NE	{1..n}	-	informace	-	NE	ANO	NULL nebo text	neomezeno
						ID umístění	FK	NE	NE	{1..n}	-

### 4.3 Fyzický návrh

Zatímco při logickém návrhu byla upřena pozornost na to, co bude obsahem databáze, při vytváření fyzického návrhu je hlavní otázka, jak to bude zprovozněno. Pro vytvoření fyzického návrhu je již nutné znát funkčnost cílového DBMS.

DBMS je systém, který umožňuje komunikovat uživateli či aplikaci s databází. Existuje několik druhů DBMS - mezi nejznámější, založené na relačním ukládání, dat patří např. MySQL, Microsoft SQL Server, Oracle či PostgreSQL.<sup>32</sup>

Komunikace je umožněna pomocí dotazovacího jazyka. U zmiňovaných relačních DBMS je to např. SQL (Structured Query Language). Jedná se o neprocedurální jazyk (nevyžaduje určení přístupové metody k datům), je relativně snadný pro naučení a pomocí něho lze přistupovat do databáze ať už kvůli konfiguraci databáze, či kvůli správě dat.<sup>33</sup>

Každý databázový systém prošel různým vývojem a i když vychází ze stejného dotazovacího jazyka, může se od ostatních v některých ohledech lišit, např. v implementaci datových typů, programovacích strukturách či v některé syntaxi. Proto je pro fyzický návrh již důležité znát DBMS, v rámci kterého bude databáze implementována. Pro implementaci DTOZE jsem vybrala databázový systém PostgreSQL, jelikož je dostupný na zařízení, na kterém bude implementována celá aplikace.

V rámci fyzického návrhu se vychází z logického návrhu. Zahrnuje návrh podkladových tabulek a doplnění integritních omezení dle cílového DBMS, návrh a implementaci odvozených sloupců, analýzu transakcí, které mají být podporovány, zavedení indexů, implementování pohledů, zvážení denormalizace, návrh bezpečnostních opatření<sup>34</sup> a otestování vzniklé databáze.

#### 4.3.1 Návrh tabulek a integritních omezení

V tomto kroku se vychází z dat nashromážděných v rámci logického návrhu. Dochází zjednodušeně k překladu tabulek z logického návrhu do jazyka SQL a syntaxe cílového DBMS a implementaci integritních omezení, popř. doplnění dalších.

Nejen pro vizualizaci lze opět využít některý z modelovacích nástrojů a vytvořit v něm databázový model. Databázový model nese mnohem více informací o jednotlivých tabulkách než ER model či logický model. Proto jsem model DTOZE rozdělila na tři logické celky, které budou čitelnější, než jeden veliký.

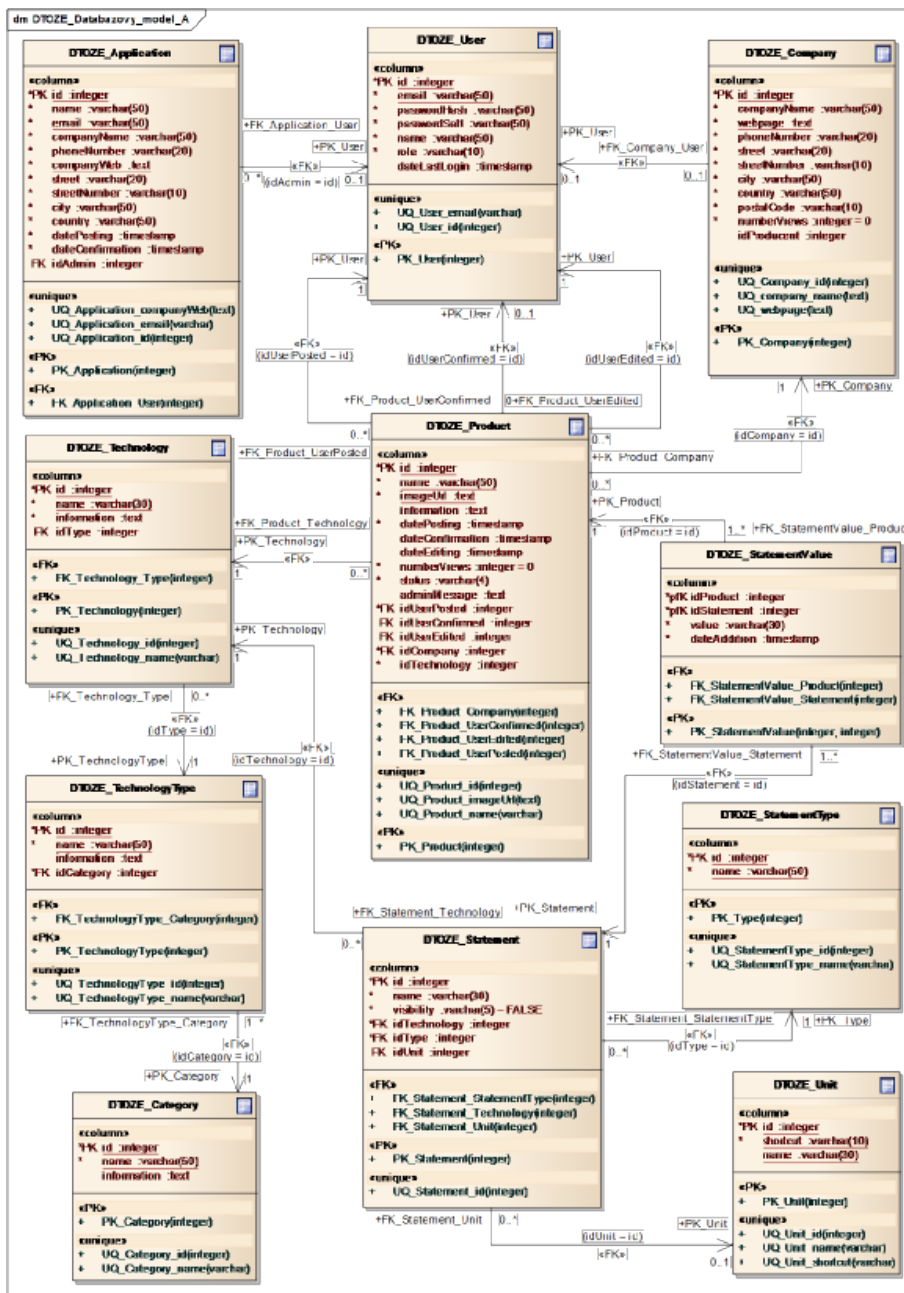
Databázový model DTOZE A, který je zobrazen na obrázku č. 14 znázorňuje hlavní strukturu databáze. Databázový model DTOZE B na obrázku č. 15 rozšiřuje entity Uživatel (DTOZE\_User), Produkt (DTOZE\_Product), Hodnota údaje (DTOZE\_StatementValue) a Údaj (DTOZE\_Statement) o entitu Změna (DTOZE\_Change) a zachycuje v jakém jsou vztahu. Databázový model DTOZE C na obrázku č. 16 zobrazuje několikanásobný vztah Dotazníková položka (DTOZE\_SurveyItem) s ostatními entitami, kterých se týká.

<sup>32</sup>CONOLLY, Thomas, Carolyn 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. Str. 39.

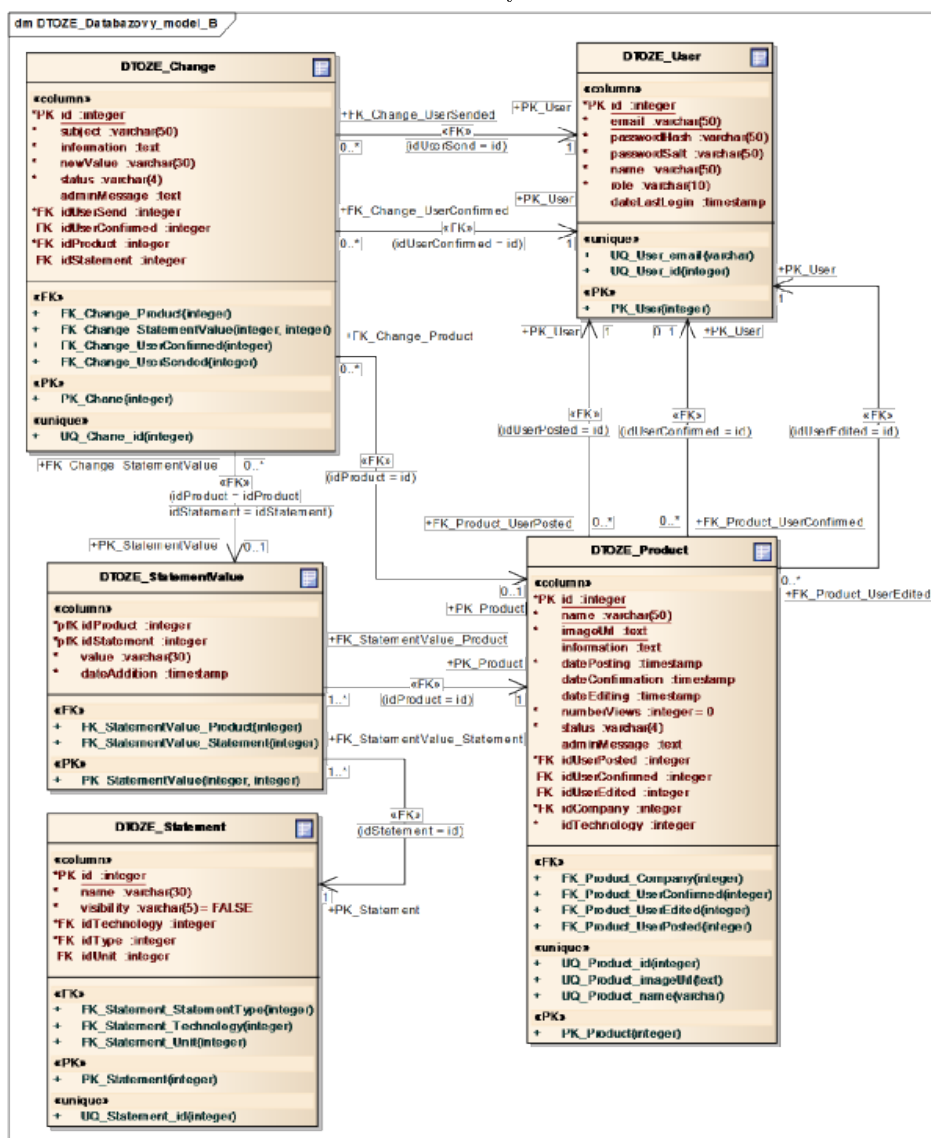
<sup>33</sup>Tamtéž str. 76.

<sup>34</sup>Tamtéž str. 259.

Obrázek 14: Databázový model DTOZE A



Obrázek 15: Databázový model DTOZE B

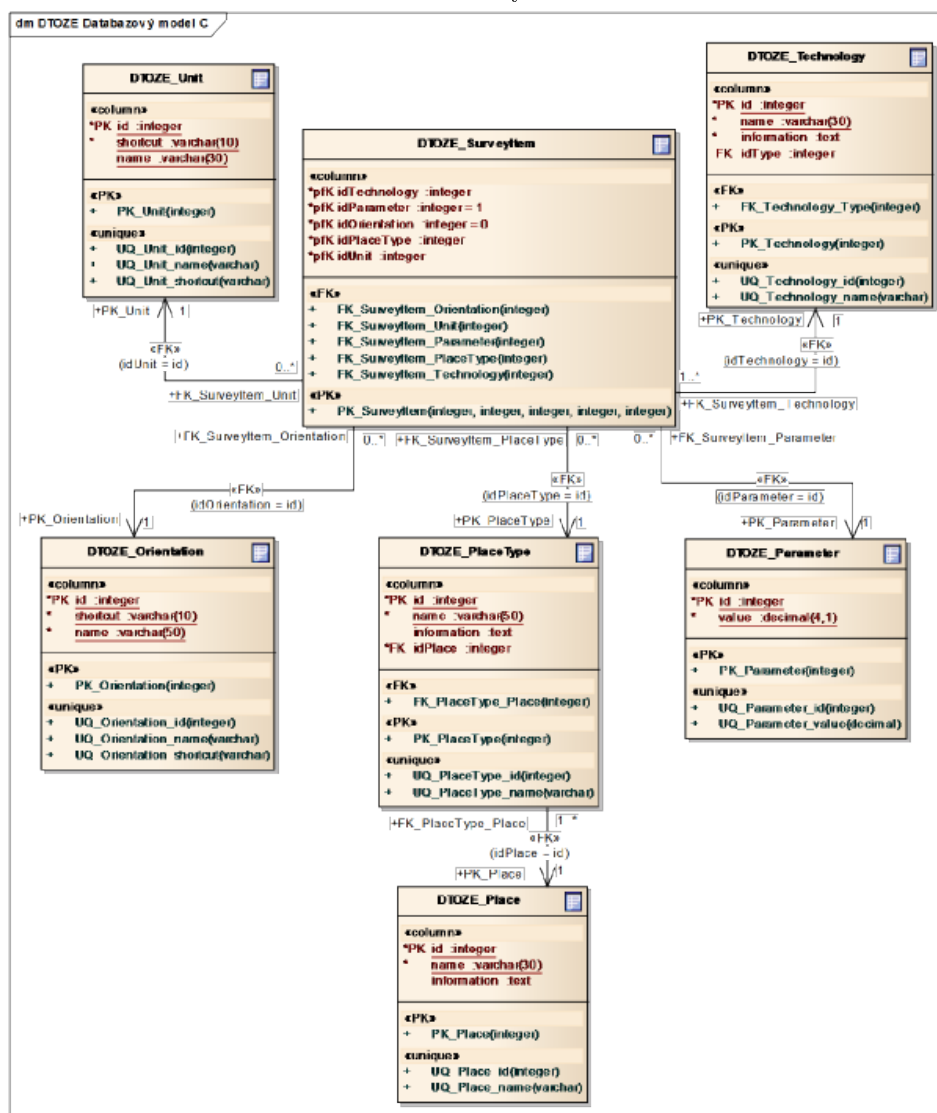


Zatímco ER model by měl být srozumitelný jak pro zadavatele tak pro analytiku, databázový model musí přehledný především pro vývojáře, který nad ním implementuje aplikaci. Protože je dobrým zvykem psát kód v angličtině a z modelu bude generován SQL skript<sup>35</sup>, přeložila jsem veškeré názvy do angličtiny. Dále je před každým názvem uvedena prefix DTOZE, který jsem zavedla z bezpečnostního důvodu.<sup>36</sup> Vztahy jsou vyjádřeny pomocí spojení přes cizí klíč. V závorkách jsou zobrazeny i sloupce, přes které je spojení realizováno.

<sup>35</sup>Vysvětleno v následujícím odstavci.

<sup>36</sup>Vysvětleno v kapitole č. 4.3.7.

Obrázek 16: Databázový model DTOZE C



Výhodou je, že modelovací nástroje většinou umožňují generovat SQL skript pro vytvoření tabulek včetně navolených datových typů a integritních omezení dle vybraného DBMS. Není nutné tedy ovládat SQL jazyk pro příkazy na vytvoření tabulek a integritních omezení, nicméně je vhodné je znát, jelikož se na vygenerovaný skript se nelze vždy spolehnout a je nutné jej ještě upravovat. Např. Enterprise Architect z databázového modelu DTOZE nevygeneroval dva cizí klíče a bylo je nutné doplnit. Dále je vhodné vygenerovaný skript upravit do logických celků, aby byl přehlednější (např. příkazy pro vytvoření primárních klíčů je vhodné mít přímo u tabulky, které se nastavení týká, apod.) V rámci práce se nebudu vysvětlení příkazů SQL věnovat podrobně, zmíním pouze základní příkazy pro vytvoření kompletní databáze, které jsou součástí SQL skriptu.

Pro vytváření tabulek slouží v SQL příkaz CREATE TABLE. V rámci tohoto příkazu lze nastavit veškeré vlastnosti tabulky - sloupce a omezení sloupců včetně primárních a cizích klíčů. Omezení lze pojmenovat, a je to doporučeno z důvodu pozdější změny či vymazání omezení. Pro vytvoření unikátních hodnot, primárních a cizích klíčů proto je použit příkaz ALTER TABLE ve spojení s příkazem

ADD CONSTRAINT. Ukázka vytvoření jedné z tabulek znázorňuje následující úryvek kódu v jazyce SQL pro databázový systém PostgreSQL.

```

1 CREATE TABLE DTOZE_Statement (
2     id integer NOT NULL,
3     name varchar(50) NOT NULL,
4     idTechnology integer NOT NULL,
5     idStatementType integer NOT NULL,
6     idUnit integer
7 );
8
9 CREATE SEQUENCE statement_id_seq;
10
11 ALTER TABLE DTOZE_Statement ALTER id SET DEFAULT NEXTVAL('statement_id_seq');
12
13 ALTER TABLE DTOZE_Statement
14     ADD CONSTRAINT UQ_Statement_id UNIQUE (id);
15
16 ALTER TABLE DTOZE_Statement
17     ADD CONSTRAINT UQ_Statement_name UNIQUE (name);
18
19 ALTER TABLE DTOZE_Statement ADD CONSTRAINT PK_Statement
20     PRIMARY KEY (id);

```

V rámci sloupců a tabulek lze definovat i omezení CHECK. Toto omezení před vložením dat do databáze zkontroluje podmínku pro vložení. Této podmínky lze využít v rámci jednoho, ale i více sloupců tabulky. Použití opět demonstřuji na části SQL skriptu DTOZE. Omezení valide\_email kontroluje pomocí regulárního výrazu správný formát e-mailu. Omezení non\_empty\_message kontroluje zda není zpráva prázdná.

```

1 ALTER TABLE DTOZE_Application ADD CONSTRAINT valide_email
2     CHECK (email ~ '^[A-Za-z0-9._%~]+@[A-Za-z0-9.-]+[.][A-Za-z]{2,6}$');
3
4 ALTER TABLE DTOZE_Change
5     ADD CONSTRAINT non_empty_message CHECK (adminMessage <> '');

```

V rámci definování vztahů a atributů jsem narazila i na několik omezení, která nelze definovat pouze v rámci sloupců či tabulek. Tato omezení lze také implementovat v rámci podmínky CHECK, jelikož lze v rámci omezení CHECK použít i uživatelsky definované funkce, kde lze pracovat s daty pomocí jazyka SQL. Ty se vytváří pomocí příkazu CREATE FUNCTION, v případě použití v CHECK musí vracet typ BOOLEAN<sup>37</sup>. Definice a použití funkce znázorňuji v následujícím úryvku kódu.

<sup>37</sup>Hodnotu TRUE nebo FALSE dle vyhodnocení těla funkce.

```

1 CREATE OR REPLACE FUNCTION check_user_role(INTEGER) RETURNS BOOLEAN AS '
2 DECLARE
3     tmpRole VARCHAR(10);
4 BEGIN
5 SELECT role INTO tmpRole FROM DTOZE_user WHERE id = $1;
6     IF FOUND THEN
7         RETURN (tmpRole LIKE 'producent');
8     ELSE
9         RETURN false;
10    END IF;
11 END;
12 ' LANGUAGE plpgsql;
13
14 ALTER TABLE DTOZE_Company ADD CONSTRAINT valide_user
15 CHECK (check_user_role(idProducent));

```

Otázkou stále zůstává, zda je kontrola na databázové vrstvě ve vícevrstevných aplikacích užitečná. V případě, že v se v rámci aplikace někdo pokusí o vložení nějakých dat a nad daty vyjde v podmínce CHECK FALSE, databázový systém vyhodí výjimku, která ale v aplikaci nebude znázorněna (pokud to nebude implementováno) a uživatel se nedozví, že se jeho záznam neuložil. Proto je nutné data kontrolovat už na vyšších vrstvách, aby bylo zaručeno, že budou např. ve správném formátu a pokud ne, dostane uživatel upozornění ještě před tím, než se k databázi přistoupí. Domnívám se, že omezení CHECK je pro použití v DTOZE zbytečné. V SQL skriptu proto budou uvedeny pouze ukázkové příklady.

### 4.3.2 Návrh odvozených sloupců

Při konceptuálním návrhu dat bylo možné definovat tzv. odvozené atributy. Jedná se o atributy, jejichž hodnota je odvozená (nejčastěji vypočítaná) z jiných atributů či tabulek. Nejčastěji se jedná o počty v rámci vztahů mezi entitami. Tyto atributy lze realizovat pomocí odvozených sloupců. Tyto sloupce je v tabulce nutné udržovat, aby stále obsahovaly aktuální informace, za cenu jednoduchého dotazování.

K tomu slouží v DBMS tzv. trigger<sup>38</sup> neboli spouštěče. Ty provedou požadovanou akci nad daným sloupcem či více sloupci nebo nad jednou či více tabulkách a to buď při vkládání, upravování či mazání. Trigger se skládá ze tří částí: událost nebo události, které spustí trigger (např. INSERT, UPDATE, DELETE nebo i CREATE, ALTER, DROP a další); podmínka, která určuje, za jakých okolností má být trigger spuštěn; akce, která se vykoná po splnění podmínky. Trigger lze definovat na úrovni řádku, ale i na úrovni pouze příkazu. Zatímco trigger definovaný na úrovni řádku se spustí pro všechny řádky, kterých se týká spouštěcí událost, trigger na úrovni příkazu se provede pouze jednou. Trigger se vytvoří příkazem CREATE TRIGGER a v rámci vytváření triggeru lze používat již dříve definované funkce.<sup>39</sup>

V původním návrhu DTOZE se nevyskytují žádné odvozené sloupce, jelikož jsem počítala s tím, že je nepoužiji. Nyní jsem ale situaci přehodnotila. Některé odvozené sloupce by bylo vhodné doimple-

<sup>38</sup>Trigger slouží i k operacím, které je nutné provést v rámci jedné transakce, viz další kapitola.

<sup>39</sup>CONOLLY, Thomas, Carolyn 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. Str. 508-509.



mentovat, místo neustálého přepočítávání. Např. je to vhodné pro veškeré statistiky. Zavedené změny je důležité zavést do všech diagramů a vytvořit příslušné triggery pomocí SQL. Změny v návrhu DTOZE jsou k zhlédnutí v přílohách, kde jsou uvedeny finální modely ze všech fází návrhu - ER model je přílohou C, logický model přílohou D a databázový model přílohou E. Implementaci triggeru použitého v DTOZE znázorňuje následující kód v jazyce SQL pro databázový systém PostgreSQL. Trigger po přidání, smazání nebo upravování nového produktu do databáze upraví odvozené sloupce v tabulkách DTOZE\_User, DTOZE\_Company a DTOZE\_Technology. Je však nutné brát ohled na to, že se produkt může nacházet v několika různých stavech (OK, KO, WAIT). V případě, že není produkt ve stavu OK, nepočítá se s ním v uložených hodnotách, jelikož častěji bude potřeba znát počet produktů ve stavu OK, než celkový počet v databázi.

```

1 CREATE OR REPLACE FUNCTION update_productCounters() RETURNS TRIGGER AS '
2 BEGIN
3     IF TG_OP = 'DELETE' THEN
4         IF OLD.status LIKE 'OK' THEN
5             UPDATE DTOZE_User SET countProductPosted = countProductPosted - 1 WHERE id = OLD.idUserPosted;
6             UPDATE DTOZE_Company SET countProduct = countProduct - 1 WHERE id = OLD.idCompany;
7             UPDATE DTOZE_Technology SET countProduct = countProduct - 1 where id = OLD.idTechnology;
8         END IF;
9         RETURN OLD;
10    END IF;
11    IF TG_OP = 'INSERT' THEN
12        IF NEW.status LIKE 'OK' THEN
13            UPDATE DTOZE_User SET countProductPosted = countProductPosted + 1 WHERE id = NEW.idUserPosted;
14            UPDATE DTOZE_Company SET countProduct = countProduct + 1 WHERE id = NEW.idCompany;
15            UPDATE DTOZE_Technology SET countProduct = countProduct + 1 where id = NEW.idTechnology;
16        END IF;
17        RETURN NEW;
18    END IF;
19    IF TG_OP = 'UPDATE' THEN
20        IF (OLD.status LIKE 'OK' AND NEW.status LIKE 'OK') THEN
21            IF (OLD.idUserPosted != NEW.idUserPosted) THEN
22                UPDATE DTOZE_User SET countProductPosted = countProductPosted + 1 WHERE id = NEW.idUserPosted;
23                UPDATE DTOZE_User SET countProductPosted = countProductPosted - 1 WHERE id = OLD.idUserPosted;
24            END IF;
25            IF OLD.idCompany != NEW.idCompany THEN
26                UPDATE DTOZE_Company SET countProduct = countProduct + 1 WHERE id = NEW.idCompany;
27                UPDATE DTOZE_Company SET countProduct = countProduct - 1 WHERE id = OLD.idCompany;
28            END IF;
29            IF OLD.idTechnology != NEW.idTechnology THEN
30                UPDATE DTOZE_Technology SET countProduct = countProduct + 1 where id = NEW.idTechnology;
31                UPDATE DTOZE_Technology SET countProduct = countProduct - 1 where id = OLD.idTechnology;
32            END IF;
33        ELSEIF (OLD.status NOT LIKE 'OK') AND (NEW.status LIKE 'OK') THEN
34            UPDATE DTOZE_User SET countProductPosted = countProductPosted + 1 WHERE id = NEW.idUserPosted;
35            UPDATE DTOZE_Company SET countProduct = countProduct + 1 WHERE id = NEW.idCompany;
36            UPDATE DTOZE_Technology SET countProduct = countProduct + 1 where id = NEW.idTechnology;
37        ELSEIF (OLD.status LIKE 'OK') AND (NEW.status NOT LIKE 'OK') THEN
38            UPDATE DTOZE_User SET countProductPosted = countProductPosted - 1 WHERE id = OLD.idUserPosted;
39            UPDATE DTOZE_Technology SET countProduct = countProduct - 1 where id = OLD.idTechnology;
40            UPDATE DTOZE_Company SET countProduct = countProduct - 1 WHERE id = OLD.idCompany;
41        END IF;
42        RETURN NEW;
43    END IF;
44 END;
45 ' LANGUAGE 'plpgsql';
46
47 CREATE TRIGGER TGR_update_productCounters
48 AFTER INSERT OR DELETE OR UPDATE ON DTOZE_Product
49 FOR EACH ROW EXECUTE PROCEDURE update_productCounters();

```

### 4.3.3 Analýza transakcí

Pro bezpečné ukládání a čtení dat je velmi důležité, aby veškeré operace byly atomické, konzistentní, izolované a stálé. Tyto problémy řeší DBMS pomocí transakcí - jednoho či několika SQL příkazů, které se vykonávají najednou. Atomicitou je myšleno, že transakce musí proběhnout celá a nebo neproběhnout vůbec. V rámci konzistence nesmí dojít k situaci, že během provádění jedné transakce ovlivňuje data ke zpracování jiná transakce. S tím souvisí i izolovanost, neboli změny provedené před potvrzením transakce musí být izolovány před ostatními transakcemi. Poslední vlastností je stálost, tedy po potvrzení transakce musí být databáze opět v konzistentním stavu a to neustále.<sup>40 41</sup>

O to, aby transakce probíhali správně, se z velké části stará DBMS, některé případy je však nutné ošetřit dodatečně. Pro jeden příkaz do databáze DBMS obvykle vytvoří samostatnou transakci. U některých skupin příkazů je to ale nežádoucí. Např. při převodu peněz z jednoho účtu na druhý je nutné zkontrolovat, zda má odesílatel dostatek peněz, poté odesílateli strhnout peníze a zároveň příjemci účet navýšit o poslanou částku. V případě, že by se v oddělené transakci peníze odečítali, v jiné přičítali a mezitím spadl systém, z jednoho účtu se peníze přičtou a na druhém neodečtou. Takových případů může nastat několik.

Proto je nutné takové případy v databázi najít a zdokumentovat a popřípadě ošetřit pomocí nástrojů DBMS (např. pomocí zmínovaných triggerů či úložných procedur), nebo jinými způsoby dle implementace. Dále je vhodné analyzovat i nejpoužívanější transakce a odhadnout průměrný a maximální počet provedení za hodinu/den, popř. rozmezí hodin, kdy je vytížení největší. Tyto informace dále poslouží při zefektivňování databázového návrhu.

U návrhu DTOZE jsem nenalezla žádné nebezpečné operace, které by bylo nutné provádět v rámci jedné transakce. Tabulka č. 16 znázorňuje analýzu transakcí provedenou nad DTOZE.

Tabulka 16: Analýza transakcí nad DTOZE

<b>Analýza transakcí DTOZE</b>		
<b>Typ vyhledávání</b>	<b>Průměrný počet</b>	<b>Maximální počet</b>
Vyhledání produktu	40 denně	100 denně
Vyhledání údaje	600 denně	1 500 denně
Vyhledání typu údaje	600 denně	1 500 denně
Vyhledání hodnoty údaje	1000 denně	2000 denně
Vyhledání firmy	5 denně	10 denně
Vyhledání technologie	3 denně	10 denně
Vyhledání druhu technologie	3 denně	10 denně
Vyhledání kategorie	3 denně	10 denně
Vyhledání formulářového parametru	30 denně	100 denně
Vyhledání změny	všechny 1 denně	všechny 3x denně
Vyhledání uživatelských údajů	1 denně	3 denně

<sup>40</sup>Tyto vlastnosti nesou název ACID, dle prvních písmen těchto vlastností v anglickém jazyce - atomicity, consistency, isolation, durability.

<sup>41</sup>Teorie databází: Transakce. *www.manualy.net* [online]. 22. 12. 2005 [cit. 2014-05-09]. Dostupné z: <http://www.manualy.net/article.php?articleID=8>.

#### 4.3.4 Indexování

Při návrhu databáze se při konceptuálním a logickém návrhu klade důraz na to, aby byla data správně uložena - nedocházelo k redundanci a byla snadno editovatelná. Ve fyzickém návrhu je důležité se zaměřit i na to, aby se k datům přistupovalo rychle a efektivně.

V databázovém systému lze kromě vytvoření tabulek a integritních omezení nastavit i tzv. indexování. Data se do databáze ukládají do tabulek většinou postupně za sebe. V případě, že je nutné najít nějaká data, která splňují zadané kritérium, musí se projít celá tabulka a u všech záznamů kontrolovat, zda nesplňují toto kritérium. K tomu, aby se nemusely všechny záznamy procházet, slouží v DBMS indexy. „Index je datová struktura, která umožňuje DBMS rychleji lokalizovat konkrétní záznamy v souboru, a tak zrychlit odezvu na uživatelské dotazy.“<sup>42</sup> Index je seřazený a každá položka indexu odkazuje na požadovanou položku a jeden nebo více údajů o její lokalizaci.<sup>43</sup>

Indexy lze vytvářet nad jedním nebo několika sloupci tabulky a každá tabulka může mít několik indexů. Index definovaný nad sloupcem tabulky pak umožní rychlý přístup k datům podle hodnot sloupce. Existuje několik typů indexů. Speciální index má automaticky např. sloupec s primárním klíčem či sloupec s omezením UNIQUE<sup>44</sup>. Dále lze vytvořit tzv. sekundární index. Takový index umožňuje např. rychlé seřazení tabulky podle sloupců, nad kterými je index definován.<sup>45</sup> Např. u tabulky DTOZE\_Product se díky primárnímu klíči bude přistupovat především pomocí indexu nad primárním klíčem. Ale jelikož se v rámci aplikace budou produkty nejčastěji vypisovat podle data schválení či data poslední změny, je vhodné nad těmito sloupci vytvořit sekundární index.

Indexy jsou u tabulek s méně než stovkami záznamů téměř zanedbatelné. Naopak databázový systém je zatěžován udržováním indexů. U větších tabulek však mohou výrazně zefektivnit přístup k datům.<sup>46</sup> Při vytváření indexů je vhodné vycházet z analýzy transakcí, kde jsou popsány často prováděné transakce, či transakce kriticky důležité pro chod aplikace. Dále je vhodné definovat index pro cizí klíče, přes které se často spojují tabulky.<sup>47</sup>

Pro DTOZE jsem se rozhodla vytvořit sekundární index u zmiňované tabulky DTOZE\_Product (nad sloupci dateConfirmation, dateEditing, numberViews) a u tabulky DTOZE\_Company (nad sloupcem numberViews). Dále jsem vytvořila index nad cizími klíči tabulek, které budou obsahovat nejvíce záznamů a budou často spojovány přes cizí klíč - týká se tabulky DTOZE\_Product a DTOZE\_Statement. K ostatním tabulkám se bude přistupovat nejčastěji přes primární klíč. Implementaci jednoho ze zmiňovaných indexů znázorňuje následující úryvek kódu v jazyce SQL pro databázový systém PostgreSQL.

```
1 CREATE INDEX IX_Product_idCompany
2 ON DTOZE_Product (idCompany);
```

<sup>42</sup> CONOLLY, Thomas, Carolyn BEGG a Richard HOLOWCZAK. *Mistrouství - databáze: profesionální průvodce tvorbou efektivních databází*. Vyd. 1. Brno: Computer Press, 2009, 584 s. ISBN 978-80-251-2328-7. Str. 511.

<sup>43</sup> Tamtéž str. 511.

<sup>44</sup> Jedná se o omezení zaručující unikátnost sloupce.

<sup>45</sup> VRÁNA, Jakub. Využití databázových indexů. *Root.cz* [online]. 22. 7. 2003 [cit. 2014-04-09]. Dostupné z: <http://www.root.cz/clanky/vyuziti-databazovych-indexu/>.

<sup>46</sup> Tamtéž.

<sup>47</sup> VONDRÁ, Tomáš. Obvyklé problémy s SQL: indexy. *fuzzy.cz: Because the world is not a crisp...* [online]. 4. 8. 2009 [cit. 2014-04-09]. Dostupné z: <http://www.fuzzy.cz/cs/clanky/obvykle-problemy-s-sql-indexy/>.

### 4.3.5 Návrh pohledů

Jazyk SQL umožňuje krom vytváření tabulek definovat struktury tabulkám podobné, ale ve své podstatě naprosto odlišné - pohledy. Pohled je struktura, ke které lze přistupovat jako k tabulkám (tedy lze z pohledu data číst, vkládat, mazat), avšak plnohodnotnými tabulkami nejsou. Jedná se totiž pouze o pojmenovaný SQL dotaz, který vybírá konkrétní sloupce pro další čtení či upravování. Nezabírá tedy žádné místo. Pohledy lze vytvořit buď z jedné tabulky (např. zredukováním sloupců), nebo spojením více tabulek (např. spojení tabulek, které je často používáno).

Pohledy lze použít především pro zpřehlednění dotazů, nebo pro dotazy, které by se jinak často opakovaly. Zásadní význam ale mohou mít i při zabezpečování databáze. Pokud do databáze přistupuje fyzicky více uživatelů s různými pravomocemi, lze pomocí pohledů omezit přístup k některým sloupcům či k celým tabulkám.

Aplikace nad DTOZE bude implementována pomocí objektově relačního mapování (ORM). K tabulkám se bude přistupovat pomocí objektů, které budou namapovány na tabulky. K datům se nebude přistupovat pomocí SQL (v některých případech to bude nutné), ale jinými mechanismy. Nemyslím si tedy, že jsou pohledy konkrétně pro DTOZE užitečné. Avšak je vhodné tuto funkčnost znát a v případě nutnosti implementovat. Na ukázkou uvádím pohled vytvořený nad tabulkami DTOZE\_Statement a DTOZE\_Unit, který spojuje tyto tabulky, a v případě nepoužití ORM by měl velký význam, jelikož hodnoty údaje a jednotky budou vždy zobrazovány společně. Pohledy též nejsou součástí SQL skriptu DTOZE. Implementaci pohledu znázorňuje následující úryvek kódu v jazyce SQL pro databázový systém PostgreSQL.

```

1 CREATE VIEW full_statement AS
2 SELECT DTOZE_Statement.id, DTOZE_Statement.name, DTOZE_Statement.visibility,
   DTOZE_Unit.shortcut
3 FROM DTOZE_Statement AS s, DTOZE_Unit AS u
4 WHERE s.id = u.id;
```

### 4.3.6 Zohlednění denormalizace

V rámci logického návrhu jsem se zabývala normalizací tabulek. Týkala se sloupců jednotlivých tabulek. Cílem bylo zamezit redundanci a zajistit snadné editování dat. Tabulky jsem rozdělila a svázala pomocí cizích klíčů. Takové rozdělení tabulek může být z hlediska redundance vhodné, ale z hlediska efektivnímu přístupu k datům občas nevhodné.

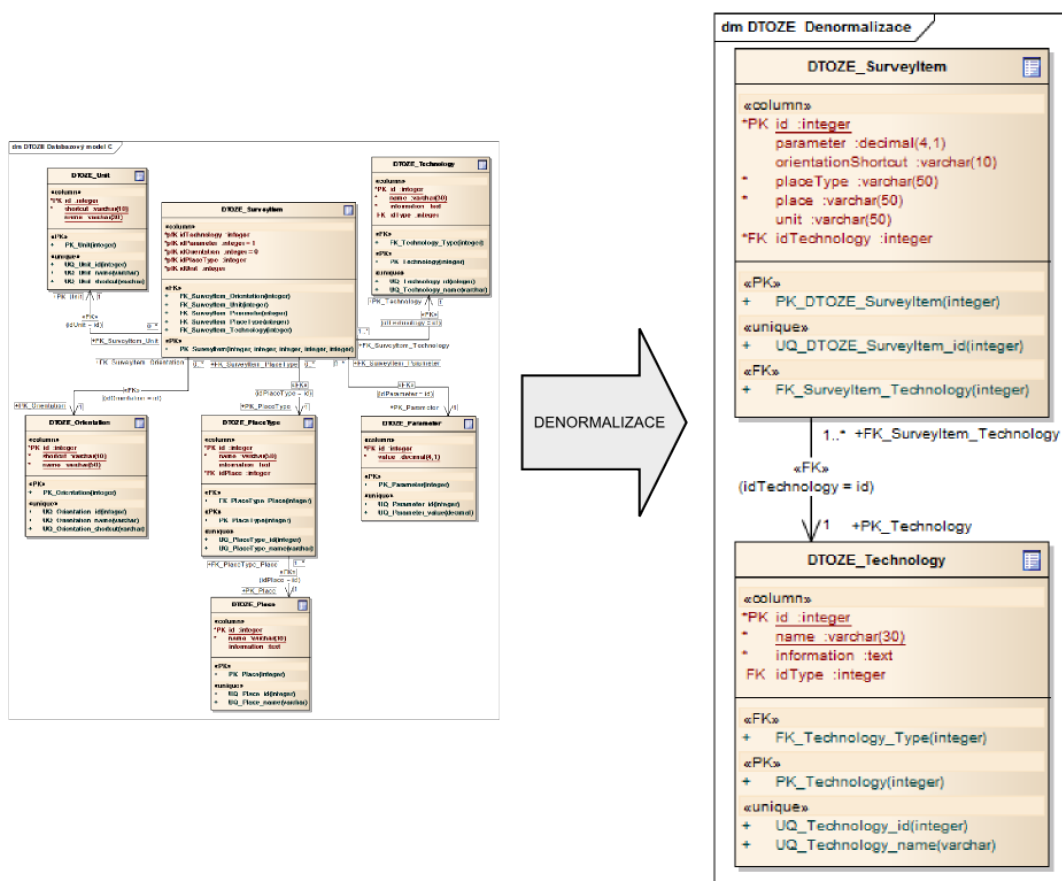
Proto v rámci fyzického návrhu může dojít ještě k přezkoumání tabulek a následné denormalizaci dle potřeb. Denormalizace by se především měla týkat databáze, která svým návrhem nesplňuje požadavky na výkonnost. Tedy týká se to především aplikací, které pracují s velkým množstvím<sup>48</sup> dat. Hlavními výhodami denormalizace je minimalizace potřeby spojení tabulek, snížení počtu cizích klíčů v tabulkách, snížení počtu indexů a počtu tabulek. Nevýhodami je naopak zpomalení aktualizace dat a může zvětšit velikost tabulek. Další nevýhodou je, že se vztahuje ke konkrétní aplikaci, je na to tedy

<sup>48</sup>Odhadem v řádech miliónů a více záznamů.

nutné brát zřetel. Dále může v některých případech zhoršit implementaci a flexibilita.<sup>49</sup> V případě upravení tabulek je nutné se vrátit k předchozímu kódu a dle změn upravit omezení či indexování.

V rámci normalizace DTOZE jsem již porušila některé normy z důvodu efektivnějšího přístupu k datům a denormalizaci již praktikovala (u tabulky DTOZE\_Company). Denormalizaci bych například mohla uvážit i u několikanásobného vztahu mezi DTOZE\_Technology, DTOZE\_Orientation, DTOZE\_PlaceType, DTOZE\_Unit a DTOZE\_Parameter, které jsou spojené pomocí vztahové tabulky DTOZE\_SurveyItem. Pro získání celistvé informace je nutné spojit pět tabulek. Jelikož je ale dat v jednotlivých tabulkách málo (max. 10 záznamů), lze tyto tabulky spojit rovnou v jednu, za cenu redundance dat. Jak znázorňuje obrázek č. 17. (Čitelnější původní diagram před denormalizací je samostatně na obrázku č. 16.)

Obrázek 17: Ukázka denormalizace na DTOZE



Jelikož předpokládaný počet záznamů v DTOZE nepřesáhne celkově za rok fungování databáze více jak 100 000 záznamů a předpokládaný růst po roce užívání je velmi malý, uvažuji, že požadavky na výkon budou bez problémů splněny, není tedy nutné se denormalizací v rámci DTOZE zabývat.

<sup>49</sup>CONOLLY, Thomas, Carolyn 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. Str. 278.

### 4.3.7 Návrh bezpečnostních opatření

Návrh bezpečnostních opatření se týká především vytvoření pohledů pro jednotlivé uživatele databáze. Jelikož bude přístup do databáze všem uživatelům umožněn pouze přes uživatelské rozhraní, není potřeba uživatelské pohledy implementovat. Přímý přístup k tabulkám DTOZE bude mít pouze aplikace a správce serveru přes uživatelské jméno a heslo.

Nad rámec běžného zabezpečení jsem uvažovala i různé útoky na zabezpečení aplikace. Např. není vhodné pojmenovávat tabulky snadno odhadnutelnými názvy, jako je např. Users. Vhodné je doplnit názvy např. prefixem, který mají všechny tabulky společné (např. DTOZE\_User). V případě proniknutí přes zabezpečení aplikace lze poté snadno odhalit, kde se nacházejí citlivé informace, které lze takto snadno získat. Avšak toto nezajistí ukradení dat, pouze jej může zbrzdit. Přednostně je nutné v rámci zabezpečení celé aplikace zajistit, aby se k datům v databázi nebylo možné dostat jinak, než je žádoucí. Prefix je ale vhodný i z důvodu rozšiřování aplikace. V případě, že by nějaká aplikace přistupovala k více databázím, je jednoznačně určeno, které tabulky patří do jaké databáze.

### 4.3.8 Testování SQL skriptu

V rámci fyzického návrhu byl SQL skript několikrát modifikován. Byla doplněna nová omezení či dotvořené funkce či triggeru. Během vytváření nových funkcí bylo průběžně testováno, zda je na zavedené operace očekávaná odpověď a zda jsou všechny příkazy syntakticky správně.

Funkčnost je sice zkontrolována, ale je vhodné ji ještě ověřit i v rámci hodnot v tabulkách. Proto je potřeba do vytvořených tabulek vložit testovací data a ověřit správnou provázanost systému na konkrétních hodnotách. Pro vkládání dat do databáze je příkaz INSERT INTO, pro upravování UPDATE a pro mazání DELETE. Již v rámci vkládání dat lze najít chyby např. v definovaných omezeních. Upravováním a mazáním dat lze zkontrolovat fungování triggerů např. na odvozených sloupcích, zda mění hodnoty správně. Výsledný skript je také vhodné komentovat pro další uživatele, kteří s ním budou pracovat. Finální SQL skript pro vytvoření DTOZE je přílohou F.

## 5 Závěr

V úvodu práce byla provedena rešerše aplikací na Internetu zabývajících se podobnou tematikou. Dále byly popsány požadavky na databázi, konkrétně databázi technologií využívajících obnovitelné zdroje energie pro drobné investory, a navrhnuty výpočty pro hodnocení investice do zařízení využívajících OZE. Celá databáze byla navrhnutá dle metodiky, která byla detailně vysvětlena a demonstrována na konkrétním zadání.

Hlavním cílem práce bylo navrhnout databázi dle požadavků. Cíl práce se podařil naplnit, zároveň byly zohledněny požadavky na webovou aplikaci, která bude vytvořena nad databází. Výsledkem je dokumentace návrhu databáze a funkční SQL skript pro fyzické vytvoření databáze na serveru. Cílem práce bylo také popsat srozumitelně metodiku návrhu relační databáze. V rámci práce vznikl stručný návod, který shrnuje postup při návrhu databáze pro různé aplikace. Snahou bylo popsat veškeré případy, na které lze při návrhu databáze narazit, což se výběrem nepříliš snadného zadání podařilo naplnit. Tento manuál dále plánuji zpřístupnit online.

Při vytváření návrhu byl největší problém se správným definováním požadavků. Nejdříve jsem definovala požadavky na databázi velmi stručně, nevytvořila jsem detailní popis a začala tvořit návrh. Při zpětném kontrolování, zda model uchovává všechny informace, včetně těch, které nejsou popsány, ale budou důležité při implementaci, jsem narazila na mnoho nesrovnalostí. Musela jsem se vrátit k požadavkům a definovat je konkrétněji. Při detailním promyšlení případů užití jsem již myslela i na implementaci a propojila vše dohromady. Celý návrh jsem pak ale musela předělat s ohledem na změny, které nebyly malé. Při příštím návrhu tedy tuto část nepodcením a více se na ní zaměřím.

Dalším krokem navazujícím na práci je rozšíření dokumentace projektu a následně implementace webové aplikace nad vzniklou databází. Možností jak rozšířit v budoucnu vzniklou aplikaci je několik. Jedním z požadavků, které nebyly zahrnuty do návrhu, je rozšíření verze z českého jazyka i do jazyka anglického. A to z prostého důvodu - většina výrobců je ze zahraničí. Databáze by se tak mohla více rozrůstat. Dále je to možnost zveřejňování článků s tematikou drobných technologií využívající OZE, které by pomohly rozšiřovat povědomí o problematice OZE a upozorňovat na zajímavosti. Dalším nezahrnutým požadavkem je propojení s metadaty geoinformačních vrstev České republiky (mapou iradiace, větrnou mapou apod.) v rámci projektu RESTEP (Regional Sustainable Energy Policy) a upravení návrhu vhodného řešení s ohledem na data z těchto map. Na tyto plány bude brán zřetel při navázání na práci.

## 6 Seznam zdrojů

1. BECHNÍK, Bronislav. Fotovoltaika jako náhrada biopaliv v dopravě. *tzbinfo.cz: stavebnictví, úspory energií, technická zařízení budov* [online]. 2. 9. 2013 [cit. 2014-04-12]. Dostupné z: <http://oze.tzb-info.cz/fotovoltaika/10291-fotovoltaika-jako-nahrada-biopaliv-v-doprave>.
2. CONOLLY, Thomas, Carolyn 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.
3. Daze.vukoz.cz. VÝZKUMNÝ ÚSTAV SILVA TAROUČY PRO KRAJINU A OKRASNÉ ZAHRADNICTVÍ, veřejná výzkumná instituce. *DAZE: Databáze obnovitelných zdrojů energie* [online]. 2005 [cit. 2013-12-10]. Dostupné z: <http://daze.vukoz.cz/daze/index.jsp>.
4. Envisluzby.cz. CEMC. *Databáze environmentálních služeb a výrobků* [online]. 2011 [cit. 2013-12-10]. Dostupné z: <http://www.envisluzby.cz/>.
5. HERNANDEZ, Michael J. a HOLOWCZAK. *Návrh databází*. 1. vydání. Praha: Grada, 2006, 408 s. Profesional. ISBN 80-247-0900-7.
6. Hodnocení investic: Vnitřní výnosové procento (IRR). *BusinessVize.cz* [online]. 9. 11. 2010 [cit. 2014-04-12]. Dostupné z: <http://www.businessvize.cz/rizeni-a-optimalizace/hodnoceni-investic-vnitрни-vynosove-procento-irr>.
7. KNÁPEK, Jaroslav, Oldřich STARÝ a Jiří VAŠÍČEK. Zásady hodnocení ekonomické efektivity energetických projektů. *Metodika EFEKT* [online]. 2003 [cit. 2014-04-12]. Dostupné z: <http://efekt.xf.cz/metodikaEFEKT.pdf>.
8. KULHAN, Jakub. Normalizace relačních databází. *Programujte.com* [online]. 2008 [cit. 2013-12-10]. Dostupné z: <http://programujte.com/clanek/2008071900-normalizace-relacnich-dat-abazi/>.
9. Res-legal.eu. EUROPEAN COMMISSION. *Renewable energy policy database and support* [online]. 2012 [cit. 2013-12-10]. Dostupné z: <http://www.res-legal.eu/>.
10. Rešerše. *Ústřední knihovna ČVUT* [online]. [cit. 2013-12-10]. Dostupné z: <http://knihovna.cvut.cz/sluzby/reserse/co-je-reserse.html>.
11. Rešerše: modelový příklad postupu. *Správa Univerzitního kampusu Masarykovy univerzity* [online]. [cit. 2013-12-10]. Dostupné z: [http://ukb.muni.cz/kuk/animace/eiz/Reserse/reserse\\_praxe.html](http://ukb.muni.cz/kuk/animace/eiz/Reserse/reserse_praxe.html).
12. SKŘIVAN, Jaromír. Databáze a jazyk SQL. *Interval.cz* [online]. 2000 [cit. 2013-12-10]. Dostupné z: <http://interval.cz/clanky/databaze-a-jazyk-sql/>.
13. Teorie databází: Transakce. *www.manualy.net* [online]. 22. 12. 2005 [cit. 2014-04-09]. Dostupné z: <http://www.manualy.net/article.php?articleID=8>.



14. VONDRA, Tomáš. Obvyklé problémy s SQL: indexy. *fuzzy.cz: Because the world is not a crisp...* [online]. 4. 8. 2009 [cit. 2014-04-09]. Dostupné z: <http://www.fuzzy.cz/cs/clanky/obvykle-pr oblemy-s-sql-indexy/>.
15. VRÁNA, Jakub. Využití databázových indexů. *Root.cz* [online]. 22. 7. 2003 [cit. 2014-04-09]. Dostupné z: <http://www.root.cz/clanky/vyuziti-databazovych-indexu/>.

## A Seznam zkratek

BCNF - Boyce-Coddova normální forma

CF - Cash Flow

CEMC - České ekologické manažerské centrum

DBDL – Database Definition Language

DBMS – Database Management System

DTOZE – databáze technologií využívajících obnovitelné zdroje energie

DPP - Discounted Payback Period

ER – Entity Relation

ID - IDentification

IRR - Internal Rate of Return

NPV - Net Present Value

ORM - objektově relačního mapování

OZE – obnovitelné zdroje energie

PDF - Portable Document Format

PP - Payback Period

RES – Renewable Energy Sources

SQL - Structured Query Language

UML - Unified Modeling Language

VÚKOZ - Výzkumný ústav Silva Taroucy pro krajinu a okrasné zahradnictví

XML - Extensible Markup Language

1NF - první normální forma

2NF - druhá normální forma

3NF - třetí normální forma

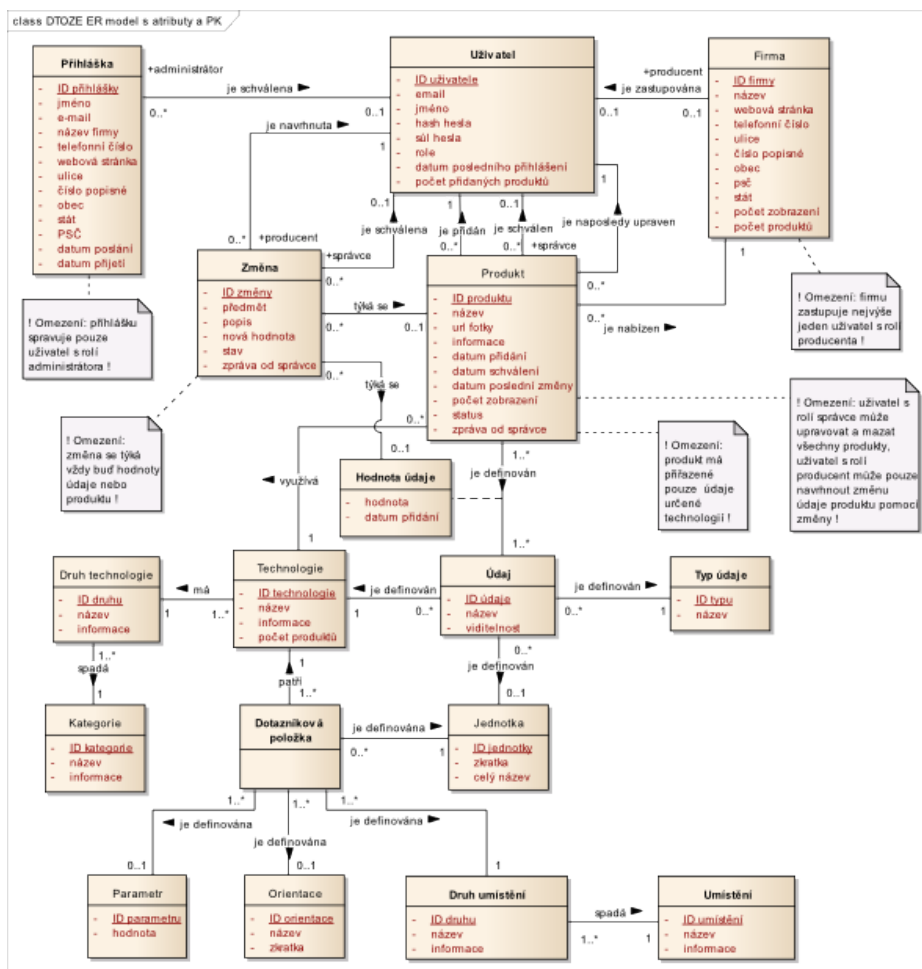
4NF - čtvrtá normální forma

5NF - pátá normální forma

## B Vize DTOZE

Vize je dostupná v elektronické podobě na nosiči společně s elektronickou verzí práce pod názvem DTOZE\_vision\_v1.pdf.

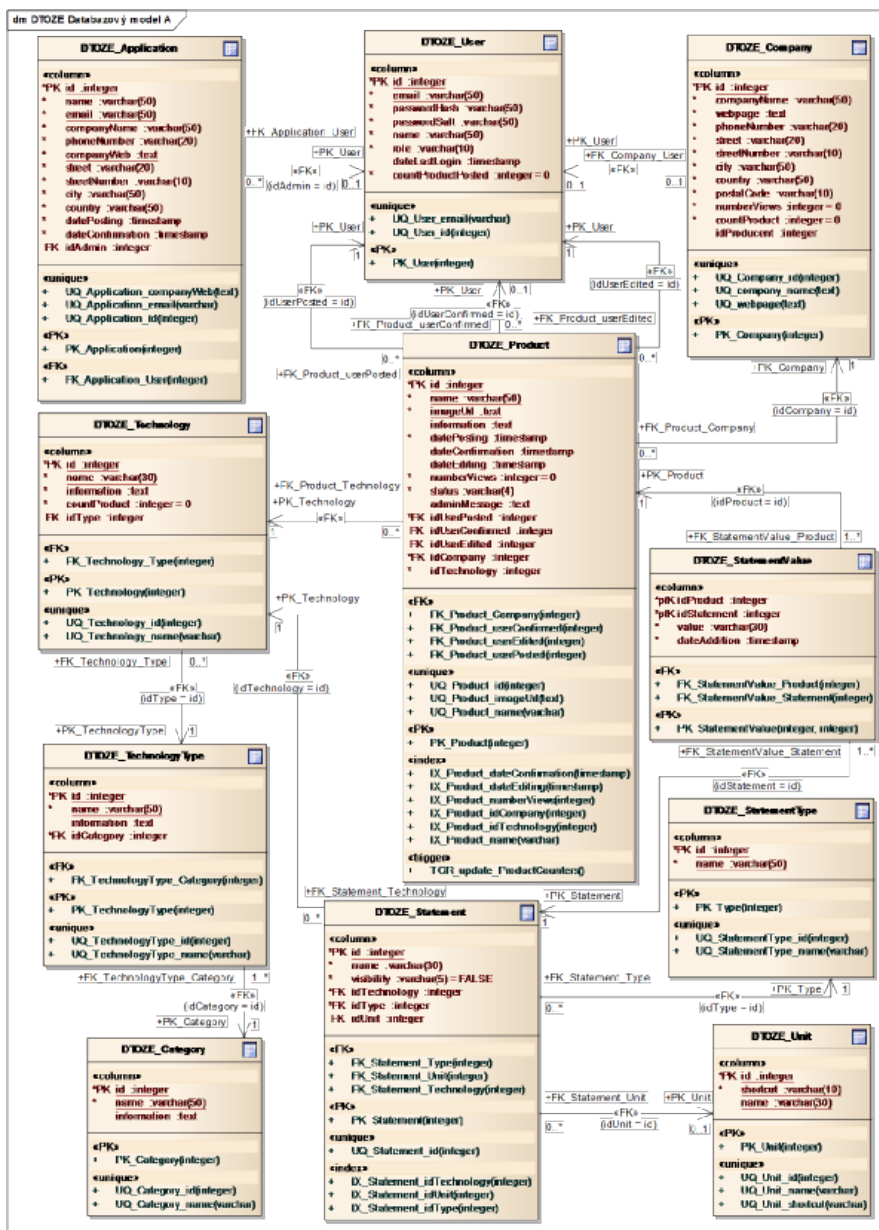
## C Finální ER model

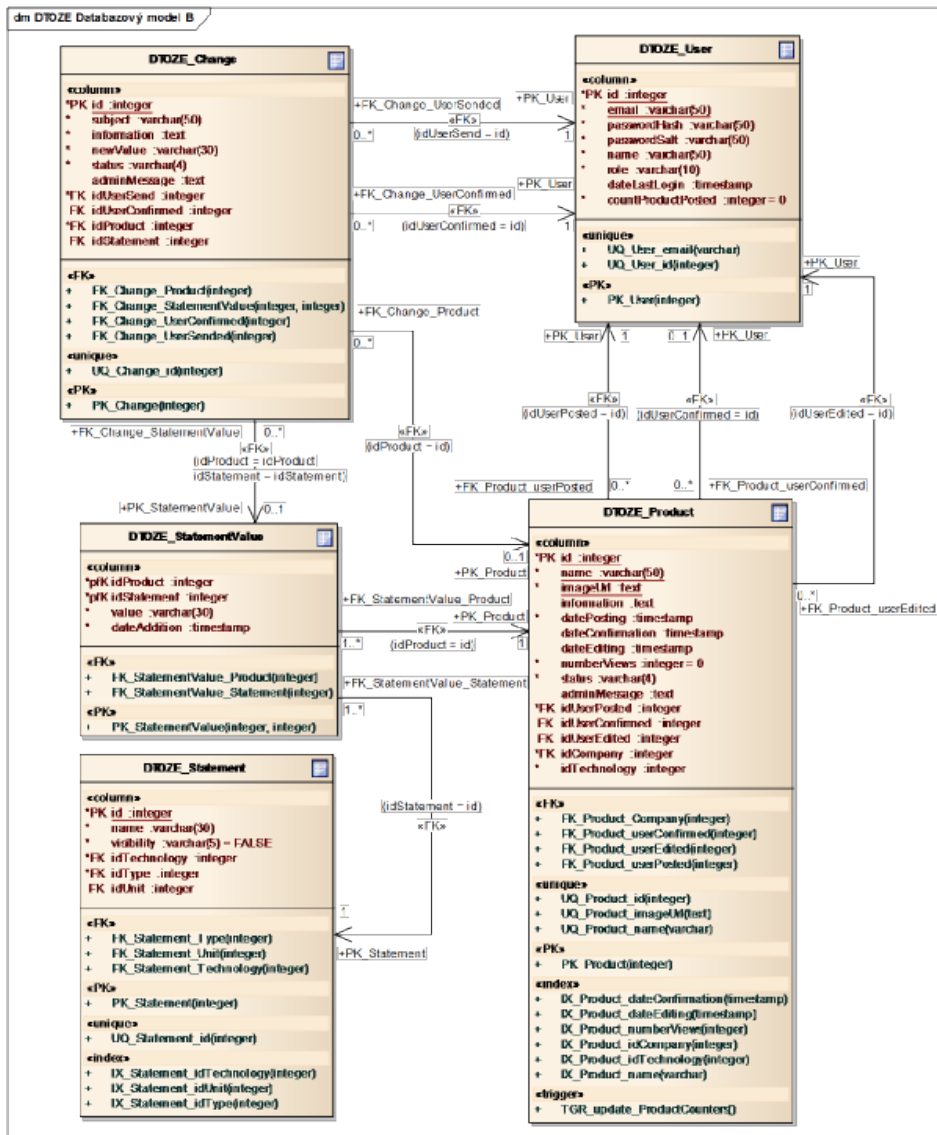


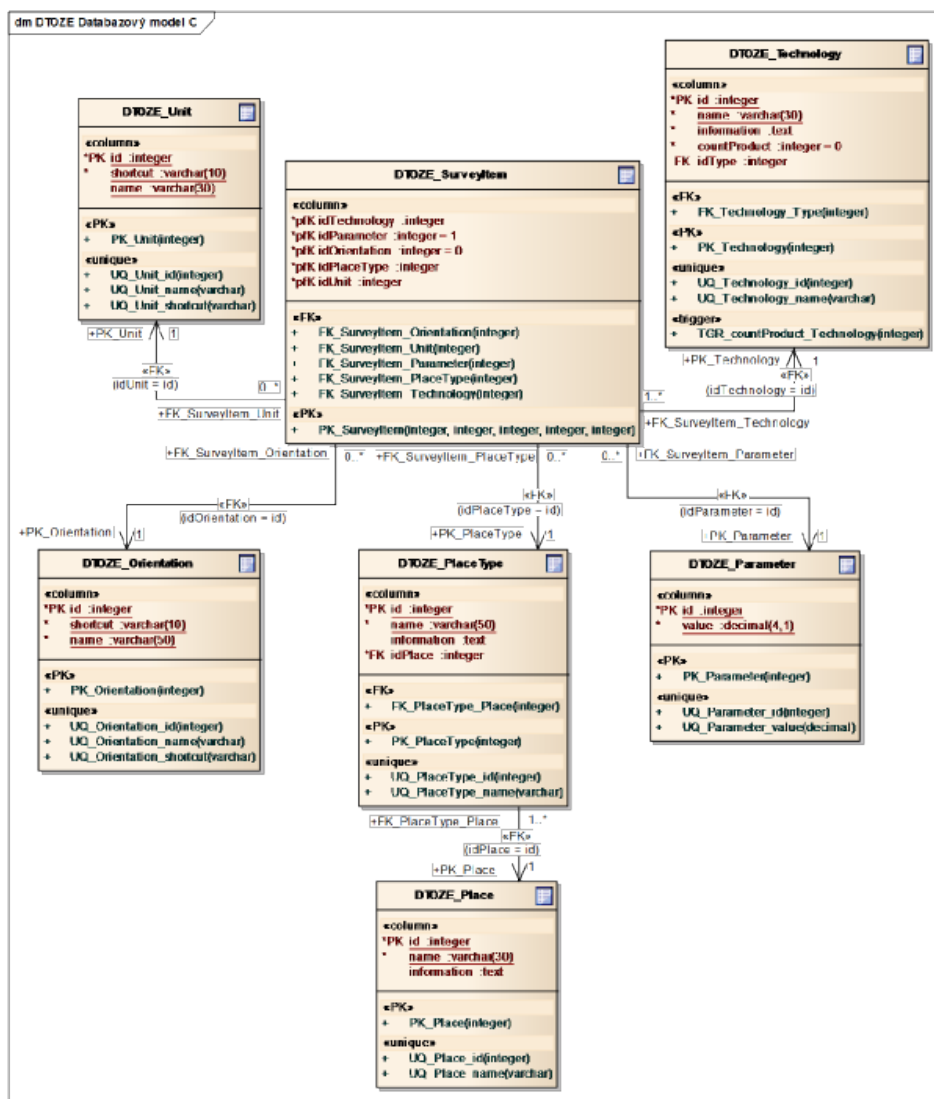
## D Finální logický model

Logický model DTOZE											
	klíč	unikátní	NULL	množina hodnot	max. délka		klíč	unikátní	NULL	množina hodnot	max. délka
<b>Firma</b>						<b>Parametr</b>					
ID firmy	PK	ANO	NE	{1..n}	-	ID parametru	PK	ANO	NE	{1..n}	-
název	-	NE	NE	text	50	hodnota	-	ANO	NE	{0,000..1,000}	3
webová stránka	-	ANO	ANO	NULL nebo text	neomezeno	<b>Přihláška</b>					
telefonní číslo	-	NE	ANO	NULL nebo text	20	ID přihlášky	PK	ANO	NE	{1..n}	-
ulice	-	NE	NE	krátký text	20	jméno	-	NE	NE	krátký text	50
číslo popisné	-	NE	NE	krátký text	10	e-mail	-	ANO	NE	krátký text	50
obec	-	NE	NE	krátký text	50	název firmy	-	NE	NE	krátký text	50
PSČ	-	NE	NE	krátký text	10	tel. na firmu	-	NE	NE	krátký text	20
stát	-	NE	NE	krátký text	50	web. stránka	-	ANO	ANO	NULL nebo text	neomezeno
počet zobrazení	-	NE	NE	{1..n}	-	ulice	-	NE	NE	krátký text	20
ID uživatele (producent)	FK	ANO	ANO	NULL nebo {1..n}	-	číslo popis.	-	NE	NE	krátký text	10
<b>Orientace</b>						obec	-	NE	NE	krátký text	50
ID orientace	PK	ANO	NE	{1..n}	-	PSČ	-	NE	NE	krátký text	10
zkratka	-	ANO	NE	krátký text	10	stát	-	NE	NE	krátký text	50
název	-	ANO	NE	krátký text	30	datum posláni	-	NE	NE	datum	-
<b>Dotazníková položka</b>						datum přijetí	-	NE	ANO	NULL nebo datum	-
ID technologie	PK, FK	NE	NE	{1..n}	-	ID uživatele (správce)	FK	NE	ANO	NULL nebo {1..n}	-
ID parametru	PK, FK	NE	NE	0 nebo {1..n}	-	<b>Hodnota údaje</b>					
ID orientace	PK, FK	NE	NE	1 nebo {1..n}	-	ID produktu	PK, FK	NE	NE	{1..n}	-
ID druhu umístění	PK, FK	NE	NE	{1..n}	-	ID údaje	PK, FK	NE	NE	{1..n}	s
ID jednotky	PK, FK	NE	NE	{1..n}	-	hodnota	-	NE	NE	krátký text	30
<b>Uživatel</b>						datum přidání	-	NE	NE	datum	-
ID uživatele	PK	ANO	NE	{1..n}	-	<b>Údaj</b>					
email	-	ANO	NE	krátký text	50	ID údaje	PK	ANO	NE	{1..n}	-
jméno	-	NE	NE	krátký text	50	název	-	NE	NE	krátký text	30
hash hesla	-	NE	NE	krátký text	50	viditelnost	-	NE	NE	{FALSE, TRUE}	5
súl hesla	-	NE	NE	krátký text	50	ID technologie	FK	NE	NE	{1..n}	-
role	-	NE	NE	{admin, producent}	10	ID typu údaje	FK	NE	NE	{1..n}	-
datum posl. přihláš.	-	NE	ANO	datum	-	ID jednotky	FK	NE	ANO	NULL nebo {1..n}	-
počet přidanych produktů	-	NE	NE	0 nebo {1..n}	-	<b>Druh technologie</b>					
<b>Umístění</b>						ID druhu	PK	ANO	NE	{1..n}	-
ID umístění	PK	ANO	NE	{1..n}	-	název	-	ANO	NE	krátký text	30
název	-	ANO	NE	krátký text	30	informace	-	NE	ANO	NULL nebo text	neomezeno
informace	-	NE	ANO	NULL nebo text	neomezeno	ID kategorie	FK	NE	NE	{1..n}	-
<b>Technologie</b>						<b>Kategorie</b>					
ID technologie	PK	ANO	NE	{1..n}	-	ID kategorie	PK	ANO	NE	{1..n}	-
název	-	ANO	NE	krátký text	30	název	-	ANO	NE	krátký text	30
informace	-	NE	ANO	NULL nebo text	neomezeno	informace	-	NE	ANO	NULL nebo text	neomezeno
počet technologií	-	NE	NE	0 nebo {1..n}	-	<b>Změna</b>					
počet produktů	-	NE	NE	0 nebo {1..n}	-	ID změny	PK	ANO	NE	{1..n}	-
ID druhu technol.	FK	NE	NE	{1..n}	-	předmět	-	NE	NE	krátký text	50
<b>Typ údaje</b>						popis	-	NE	NE	text	neomezeno
ID typu údaje	PK	ANO	NE	{1..n}	-	nová hodnota	-	NE	NE	krátký text	30
název	-	ANO	NE	text	30	stav	-	NE	NE	{OK, KO, WAIT}	4
<b>Produkt</b>						zpráva od správce	-	NE	NE	text	neomezeno
ID produktu	PK	ANO	NE	{1..n}	-	ID uživatele (producent)	FK	NE	NE	{1..n}	-
název	-	ANO	NE	krátký text	30	ID uživatele (správce)	FK	NE	ANO	{1..n}	-
url fotky	-	NE	ANO	NULL nebo text	neomezeno	ID produktu	FK	NE	NE	{1..n}	-
informace	-	NE	ANO	NULL nebo text	neomezeno	ID údaje	FK	NE	ANO	NULL nebo {1..n}	-
datum přidání	-	NE	NE	datum	-	datum přidání	-	NE	NE	datum	-
datum schválení	-	NE	ANO	NULL nebo datum	-	datum schválení	-	NE	ANO	NULL nebo datum	-
datum posl. změny	-	NE	NE	datum	-	<b>Jednotka</b>					
počet zobrazení	-	NE	NE	0 nebo {1..n}	-	ID jednotky	PK	ANO	NE	{1..n}	-
stav	-	NE	NE	{OK, WAIT, KO}	4	zkratka	-	ANO	NE	krátký text	10
zpráva od správce	-	NE	NE	text	neomezeno	název	-	ANO	ANO	NULL nebo text	30
ID uživatele (přidal)	FK	NE	NE	{1..n}	-	<b>Druh umístění</b>					
ID uživatele (schválil)	FK	NE	NE	{1..n}	-	ID umístění	PK	ANO	NE	{1..n}	-
ID uživatele (upravil)	FK	NE	NE	{1..n}	-	název	-	ANO	NE	krátký text	30
ID firmy	FK	NE	NE	{1..n}	-	informace	-	NE	ANO	NULL nebo text	neomezeno
ID technologie	FK	NE	NE	{1..n}	-	ID umístění	FK	NE	NE	{1..n}	-

## E Finální databázový model







## F SQL skript DTOZE

Skript je dostupný v elektronické podobě na nosiči společně s elektronickou verzí práce pod názvem DTOZE\_SQL\_script.sql.

## G Obsah přiloženého CD

	readme.txt .....	stručný popis obsahu CD
	appendix .....	adresář s přílohami
	_ DTOZE_SQL_script.sql .....	SQL skript pro vytvoření databáze
	_ DTOZE_vision_v1.pdf .....	Vize webové aplikace DTOZE
	src .....	adresář se zdrojovým kódem práce
	_ thesis.tex .....	zdrojová forma práce ve formátu $\text{\LaTeX}$
	text .....	adresář s textem práce
	_ maurerova_veronika.pdf .....	text práce ve formátu PDF