

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Software pro vykazování výroby elektrické energie z
fotovoltaických panelů

Petr Vondrouš

Bakalářská práce

2012

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr Vondrouš**
Osobní číslo: **I070112**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Software pro vykazování výroby elektrické energie z fotovoltaických panelů**
Zadávací katedra: **Katedra informačních technologií**

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

Cílem práce je návrh a realizace aplikace pro evidenci, vykazování a fakturaci vyrobené elektrické energie z obnovitelných zdrojů na základě manuálního odečtu dat z elektroměru. Data budou ukládána lokálně na PC uživatele v xml souboru nebo do relační databáze. Aplikace bude psána v jazyce C nebo Delphi.

Mezi další funkce aplikace patří vedení statistik, generování měsíční hlášení o výrobě z obnovitelných zdrojů pro ČEZ a také čtvrtletní hlášení pro ERU. Aplikace bude technicky zabezpečena z hlediska licencování (generování licenčních klíčů nebo jiný princip).

V teoretické části aplikace bude provedena rešerše na trhu existujících řešení a jejich porovnání.

Odborný konzultant - Jaromír Kopecký, T-Solutions, s.r.o.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

1. HERNANDEZ, J., M., Viescas, L., J.: Myslíme v jazyku SQL: Tvorba dotazů. Grada, 2004. ISBN 978-80-247-0899-7
2. STEPHENS, Ryan K. et al. Naučte se SQL za 28 dní. Stačí hodina denně. Computer Press, 2010. 728 s. ISBN: 978-80-251-2700-1.
3. SVOBODA, L. 1001 tipů a triků pro Delphi. 2. vyd. Computer Press, 2003. 546 s. ISBN 80-7226-488-5

Vedoucí bakalářské práce:

RNDr. David Žák, Ph.D.

Katedra informačních technologií

Datum zadání bakalářské práce: **16. prosince 2011**

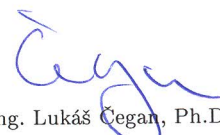
Termín odevzdání bakalářské práce: **11. května 2012**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 30. března 2012

Prohlášení autora

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10. 8. 2012

Petr Vondrouš

Poděkování

Chtěl bych poděkovat panu Jaromíru Kopeckému za rady a konzultace při tvorbě aplikace vznikající pro tuto práci.

Anotace

Cílem této práce je vytvoření softwaru pro evidenci, vykazování a fakturaci výroby elektrické energie z obnovitelných zdrojů. Po krátkém úvodu do legislativy výroby elektřiny z fotovoltaických panelů, jsou porovnány konkurenční produkty a nakonec je popsán vznik vlastního softwaru.

Klíčová slova

Výroba elektřiny, Energetický regulační úřad, fotovoltaika, Delphi, Firebird

Title

Software for reporting electricity production from photovoltaic panels

Annotation

The aim of this bachelor's thesis is creating software for reporting electricity production from photovoltaic panels. After a short brief of legislation problems, there is comparasion of competitive products.

Keywords

Electricity production, photovoltaic, Delphi, Firebird

Obsah

Seznam zkratk	9
Seznam obrázků	10
Úvod	11
1 Fotovoltaické elektrárny	12
1.1 Popis problematiky	12
1.2 Povinnosti provozovatele FVE	12
1.2.1 Energetický regulační úřad	12
1.2.2 Provozovatel distribuční sítě	13
1.3 Forma podpory	14
1.3.1 Výkupní ceny.....	14
1.3.2 Zelený bonus	14
1.3.3 Změna podpory.....	14
2 Existující nástroje pro FVE	15
2.1 Potřeby výrobce	15
2.2 Ruční evidence	15
2.3 CNE Program FAKTURACE pro FVE	15
2.4 Energy ForEver	16
2.5 Albert.....	17
2.6 Voltas.....	20
3 Návrh softwaru pro vykazování výroby elektrické energie	23
3.1 Definice zadání	23
3.2 Databáze	23
3.2.1 Firebird	24
3.3 Moduly aplikace	24
3.3.1 Číselníky.....	24
3.3.2 Fakturace	25
3.3.3 Sestavy.....	25
3.4 Rozložení objektů	25
3.5 Licence.....	26
4 Vytvoření softwaru pro vykazování	28
4.1 Databázová struktura	28

4.1.1	Nástroje pro správu databáze.....	28
4.1.2	Domény	28
4.1.3	Generátory	29
4.1.4	Triggery	29
4.1.5	Tabulky.....	29
4.1.6	Procedury.....	31
4.2	Potřebné komponenty pro Delphi.....	31
4.2.1	IceTabSet	31
4.2.2	SMDBGrid	32
4.2.3	Rave Reports	32
4.2.4	WinCrypt	32
4.3	Unity v Delphi	33
4.3.1	Generátor Licence.....	33
4.3.2	dmMain.....	33
4.3.3	ListFrame.....	33
4.3.4	EditFrame	34
4.3.5	ProdPlaceUnit.....	34
4.3.6	InvoiceUnit	35
4.3.7	DefUnit	35
4.3.8	LicenceUnit	35
4.3.9	IniUnit.....	35
	Závěr	36
	Literatura	37
	Příloha A – Diagram databázových tabulek	38
	Příloha B – Zdrojový kód souboru IniUnit.pas	39
	Příloha C – Ukázky obrazovky aplikace	43

Seznam zkratk

FVE	Fotovoltaická elektrárna
OZE	Obnovitelné zdroje
ERÚ	Energetický regulační úřad
DS	Distribuční síť
OS	Operační systém
SW	Software
DB	Databáze

Seznam obrázků

Obrázek 1 - CME program FAKTURA pro FVE - náhled	16
Obrázek 2 - Energy ForEver, údaje o výrobě	17
Obrázek 3 - Albert, úvodní obrazovka	18
Obrázek 4 – Albert, okno fakturace.....	19
Obrázek 5 - Voltas.....	21
Obrázek 6 - Voltas, editace vytvořené faktury	21
Obrázek 7 - zjednodušený návrh diagramu tříd.....	26
Obrázek 8 - FlameRobin, základní okno	28
Obrázek 9 - Generátor licence	33
Obrázek 10 - Diagram tříd unitu ProdPlaceUnit	34

Úvod

S rostoucími energetickými nároky dnešní společnosti vyvstala na povrch otázka, jak dlouho lze využívat nejrozšířenější zdroje energie, kterými jsou fosilní paliva, než budou vyčerpány. Částečnou odpovědí na tuto otázku by mohly být zdroje paliva, které nelze vyčerpávat, resp. které lze obnovovat. Odtud také přichází pojmenování obnovitelné zdroje energie. Tyto zdroje energie jsou hledány hlavně v okolní přírodě, kde je pak snaha řízeně přeměnit nevyužitou energii na jinou, kterou umíme využívat.

Možnosti získání energie znají lidé již dávno, se současným rozvojem technologií se tyto možnosti zefektivňují a také rozšiřují. Lepší se efektivita využití těchto způsobů také jejich větší uplatnění na trhu a vyšší podíl na celkovém zisku energie, se všemi svými výhodami i nevýhodami. Zvyšující se podíl OZE je ale také způsoben podporou těchto druhů energie od české vlády i ze strany Evropské unie, kdy výrobci z OZE jsou dotováni vyššími výkupními cenami.

Podpora OZE ze strany státu přispěla v České Republice k velkému rozvoji elektráren využívajících OZE, nejvíce pak energii ze slunce neboli fotovoltaickou energií. Fotovoltaické elektrárny jsou v posledních letech budovány v různých velikostech od velkých farem fotovoltaických panelů umístěných na rozsáhlých pozemcích, až po malé elektrárny s jedním či dvěma panely na střeše rodinného domu. Provozovatelé těchto FVE musejí, kvůli komunikaci s úřady a fakturaci jejich odběratelům, vést určité výkaznictví. A právě tímto se zabývá tato bakalářská práce, jejímž cílem je vytvoření softwaru pro vykazování výroby elektrické energie z fotovoltaických panelů.

V úvodní části se bakalářská práce věnuje obecnému popisu problematiky a legislativy FVE, která je s tím spojená a způsobu vykazování výroby elektrické energie. V další části porovnává tato práce již existující softwary, které mohou provozovatelé FVE používat pro vykazování. Nakonec popisuje bakalářská vznik softwaru pro evidenci, statistiku a vykazování výroby FVE. Na vytvoření zadání, jak by měl software vypadat a jak by měl pracovat, se velkou měrou podílel pan Jaromír Kopecký ze společnosti T-Solutions, s.r.o.

1 Fotovoltaické elektrárny

1.1 Popis problematiky

Jak již bylo v úvodu naznačeno, výroba elektrické energie z OZE prožívá v současnosti v České Republice obrovský boom. Je to dáno hlavně podporou OZE ze strany státu. Tato podpora má své kořeny v Bruselu v roce 1997, kdy si EU stanovila cíl do roku 2010 vyrábět 12 % hrubé domácí spotřeby energie z obnovitelných zdrojů. Tento cíl byl Evropskou komisí 10. ledna 2007 zrevidován a navýšen na 20 % podílu obnovitelných zdrojů energie na spotřebě energie v EU do roku 2020 a povinného minimálního cíle 10 % podílu pro biopaliva. Tyto cíle jsou uvedeny v Bílé knize o obnovitelných zdrojích energie.

Dne 27. září 2001 přijal Evropský parlament směrnici¹ o podpoře elektřiny vyrobené z obnovitelných zdrojů energie na vnitřním trhu s elektřinou. Tímto² usnesením se podpora OZE stala pro členské státy závazná a každá členská země má možnost zavést finanční opatření, jako daňové úlevy a různé dotace. Určeny jsou cíle celého Evropského společenství (20 % podílu OZE do roku 2020), tak dílčí cíle pro jednotlivé země (13 % podílu OZE pro Českou republiku do roku 2020)

Tato evropská legislativa vedla v České Republice k přijetí zákona č. 180/2005 Sb. o podpoře výroby elektřiny z obnovitelných zdrojů energie a o změně některých zákonů (zákon o podpoře využívání obnovitelných zdrojů). Dle tohoto zákona se „Obnovitelnými zdroji se rozumí obnovitelné nefosilní přírodní zdroje energie, jimiž jsou energie větru, energie slunečního záření, geotermální energie, energie vody, energie půdy, energie vzduchu, energie biomasy, energie skládkového plynu, energie kalového plynu a energie bioplynu“ (1).

Přijetí zákona o podpoře výroby elektřiny z OZE vedlo k masivní výstavbě FVE, která způsobovala zvyšování cen elektrické energie a hrozbu dalšího zvyšování v budoucích letech až o desítky procent. Proto nechala v září 2010 vláda vypracovat a ve stavu legislativní nouze i parlamentem schválit novelu zákona č. 180/2008 Sb., která velmi omezuje podporu pro nově postavené FVE (po roce 2010). Výhodné podmínky pro již stojící provozovny zaručuje na 20 let.

1.2 Povinnosti provozovatele FVE

Každý výrobce elektřiny z FVE musí splnit několik legislativních požadavků. Mimo požadavků, které se týkají stavby elektrárny a podnikání samotného jsou to požadavky Energetického regulačního úřadu a provozovatele distribuční sítě.

1.2.1 Energetický regulační úřad

ERÚ má na starosti dohled nad energetickým průmyslem v ČR a do jeho působnosti patří:

¹ Směrnice Evropského parlamentu a Rady 2001/77/ES

² Směrnici 2001/77/ES a 2003/30/ES ruší a nahrazuje nová směrnice 2009/28/ES

- regulace cen,
- podpora využívání obnovitelných a druhotných zdrojů energie a kombinované výroby elektřiny a tepla,
- ochrana zájmů zákazníků a spotřebitelů,
- ochrana oprávněných zájmů držitelů licencí,
- šetření soutěžních podmínek,
- spolupráce s ÚOHS,
- podpora hospodářské soutěže v energetických odvětvích,
- výkon dohledu nad trhy v energetických odvětvích.

Každý nový výrobce elektřiny z OZE musí požádat ERÚ o přidělení licence na výrobu elektřiny. Licence je nutná již pro zkušební provoz zdroje elektřiny. Vlastnictví licence na výrobu elektřiny a současně smlouvy o dodávce elektřiny je nezbytnou podmínkou pro dodávku elektřiny do elektrizační soustavy.

Pro ERÚ musí výrobce předkládat ještě výkazy potřebné zejména kvůli statistice. Pro výrobce z ostatních OZE (větrná a fotovoltaická elektrárna) menších než 0,5 MWe je to čtvrtletní výkaz VA 1-12

1.2.2 Provozovatel distribuční sítě

Výrobce musí uzavřít smlouvu s nějakým provozovatelem distribuční soustavy, který od něho bude odebírat a vykupovat vyrobenou elektrickou energii. Žádosti, které požadují různí provozovatelé DS se mohou lišit.

Dále musí minimálně měsíc před plánovaným započítáním nahlásit provozovateli distribuční soustavy výběr formy podpory výroby z OZE. Což znamená, zda za výrobu bude dostávat podporu formou výkupních cen nebo formou zelených bonusů. K výběru formy podpory slouží oznámení, které je přílohou č. 1 vyhlášky č. 475/2005 Sb., v platném znění.

Nejpozději čtyři měsíce před plánovaným zahájením výroby předat hlášení o předpokládaném množství elektřiny vyrobené z OZE, které je přílohou č. 2 vyhlášky č. 475/2005 Sb. Hlášení o plánu výroby se poté odesílá každý rok nejpozději do 31. srpna.

Na konci každého měsíce odesílá výrobce provozovateli DS fakturu na dodanou elektrickou energii. Přílohou faktury je Měsíční výkaz o výrobě elektřiny z OZE, který je přílohou č. 6 vyhlášky č. 541/2005 Sb.

1.3 Forma podpory

Dle již zmiňovaného zákona 180/2005 Sb. (novelizovaného 180/2008 Sb.) je státem garantovaná podpora výroby elektrické energie z OZE. Způsoby podpory jsou dva a to buď formou výkupních cen, nebo formou zelených bonusů.

1.3.1 Výkupní ceny

V případě podpory formou výkupních cen prodává výrobce veškerou vyrobenou elektrickou energii svému regionálnímu distributorovi, který má ze zákona povinnost ji od výrobce odebírat. Výkupní cena, za kterou je elektřina prodávána je zafixována po dobu dvaceti let (15 let dle zákona, 5 let vyhláškou), což znamená, že výrobce prodává 20 let energii za cenu, která platila v době uvedení elektrárny do provozu.

Výkupní cena nesměla, dle zákona o podpoře výroby elektřiny z obnovitelných zdrojů, meziročně klesnout o více než 5 %, aby byl zachován princip návratnosti investice do 15 let. Jelikož to ale vedlo k velkému nárůstu počtu FVE a tím pádem i k velkému zdražování elektrické energie. Proto byla navržena již zmiňovaná novela zákona 180/2008 Sb., která umožňuje meziročně snížit výkupní ceny o více než 5 %.

Výkupní ceny jsou výhodné, pokud výrobce energii sám nespotřebovává.

1.3.2 Zelený bonus

Zvolí-li si výrobce podporu formou zeleného bonusu, nedodává veškerou vyrobenou energii do DS, ale může ji spotřebovat sám, nebo prodat nějakému třetímu subjektu. Zelený bonus je příplatek k tržní ceně elektřiny. Prodá-li výrobce elektřinu z OZE za smlouvenou tržní cenu účastníkovi trhu s elektřinou nebo vyrobenou elektřinu sám spotřebuje, má právo navíc inkasovat od provozovatele přenosové nebo regionální distribuční soustavy na základě předloženého výkazu zelené bonusu. Výše zeleného bonusu je pro každý druh OZE každoročně upravována a zveřejněna v cenovém rozhodnutí ERÚ. V současné době je platné cenové rozhodnutí č. 7/2011.

Ceny za zelený bonus jsou nižší než výkupní ceny, ale vyplatí se, pokud provozovatel FVE vyrobenou energii sám spotřebovává. Nemusí ale pak platit za elektřinu odebranou ze sítě. Výrobce si však musí sám najít odběratele, který od něho bude elektřinu vykupovat.

1.3.3 Změna podpory

Formu podpory může výrobce energie měnit a to jednou ročně. Změnu je nutné nahlásit do 30. listopadu a platná bude od 1. ledna následujícího roku. Při změně zůstává výrobce stále ve stejné kategorii, která platí k datum uvedení výroby do provozu.

2 Existující nástroje pro FVE

2.1 Potřeby výrobce

Jak již bylo řečeno, všichni výrobci elektřiny z OZE musejí o výrobě vést záznamy a z nich generovat výkazy pro ERÚ a pro distributora. Každý měsíc je potřeba provádět odečty stavu elektroměru a z odečtených hodnot vytvářet faktury pro odběratele elektřiny a sestavovat výkazy.

2.2 Ruční evidence

Někteří, hlavně menší, výrobci z FVE možná používají vlastní nástroje a postupy, jak evidovat vyrobenou elektřinu. Může to být například tabulka v tabulkovém procesoru, nebo nějaký podobný postup. Všechny potřebné výkazy, které musí výrobce vyplňovat, jsou k dispozici na internetu na stránkách ERÚ nebo distributora ve formátu *doc*, který lze ručně upravit a odeslat mailem.

Tento způsob výkaznictví má výhodu v ceně, ale je velmi náchylný na chybné vyplnění dat.

2.3 CNE Program FAKTURACE pro FVE

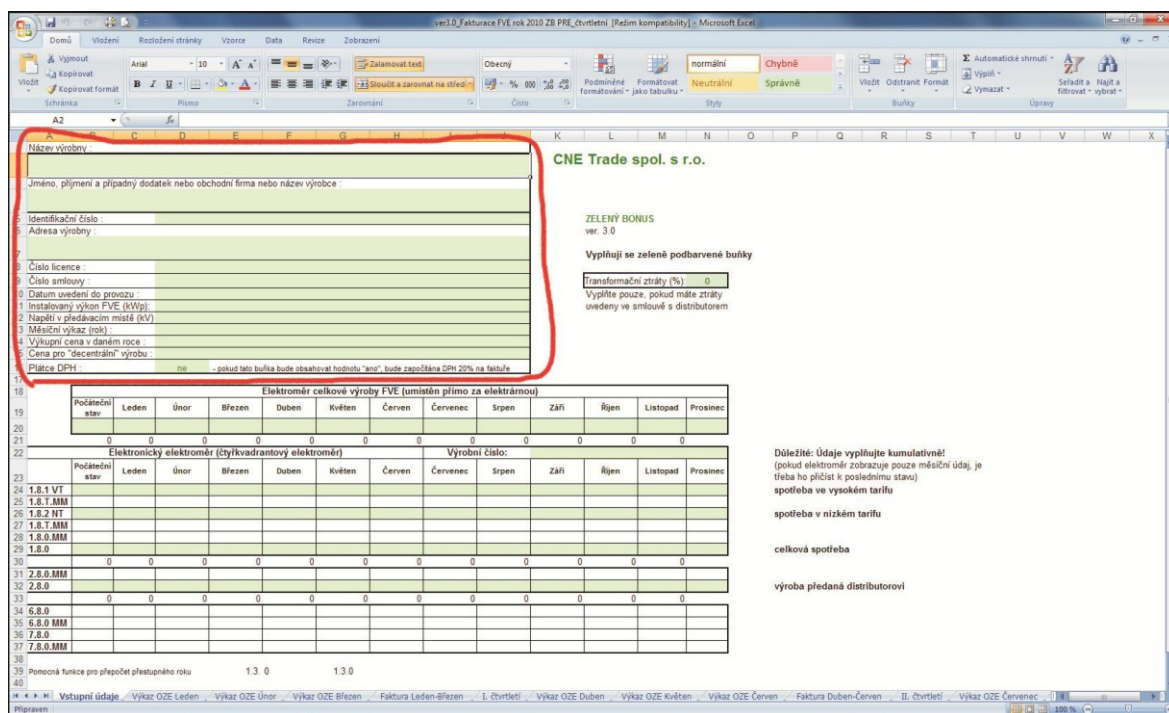
Firma Czech Nature Energy, a. s., která se mj. zabývá také výstavbou FVE, nabízí program pro fakturaci FVE³. Program je vytvořený v MS Excel a umožňuje jednou měsíčně zadávat stavy elektroměru a následně z těchto dat vytvoří fakturu, měsíční výkaz pro distributora a čtvrtletní výkaz pro ERÚ, které lze posléze vytisknout. Výrobce má na každý kalendářní rok jeden soubor *xls*, do kterého na jeden list zapisuje jednotlivé měsíční odečty, ze kterých se poté dotahují data na další listy, na kterých jsou faktury na všechny měsíce, výkazy na všechny měsíce a výkazy na všechna čtvrtletí.

Všechny tyto výkazy jsou použitelné pouze pro elektrárny o instalovaném výkonu do 500 kW, což platí i pro součet výkonů elektráren pod jednou licenci. V současné době není měsíční výkaz pro ERÚ z instalací nad 0,5 MW tímto programem podporován.

Cena je 999 Kč (+ DPH 20 %) + poštovné a balné. Legislativní aktualizace jsou stávajícím uživatelům programu k dispozici za cenu ročního udržovacího poplatku ve výši 200 Kč bez DPH.

Tvůrce nabízí též možnost vedení programu za zákazníka, kdy výrobce energie z OZE sdělí firmě Czech Nature Energy, a. s. pouze stavy elektroměru a ta na základě plné moci vytvoří a odešle fakturu, měsíční výkaz pro distributora i čtvrtletní výkaz pro ERÚ. Tato služba je poskytována za cenu licence programu a ročního paušálu za vedení fakturace. Výše paušálu je stanovena individuálně dle používaných služeb.

³ <http://www.cne.cz/fotovoltaicke-systemy/program-fakturace-pro-fve/>



Obrázek 1- CME program FAKTURA pro FVE - náhled

Podobný sešit si mohou provozovatelé FVE vytvořit sami doma a tento nástroj se i blíží postupu popsanému v předchozí kapitole, jen za cenu, za kterou je možné získat robustnější a propracovanější nástroj. Výhodou naopak je jeho snadná přenositelnost mezi více počítači např. na nějakém externím mediu. Hlavní výhodou je možnost spuštění na Linuxových operačních systémech.

2.4 Energy ForEver

Další firma, která se zabývá budováním FVE, vytvořila svůj vlastní nástroj⁴ na evidenci fakturace a výkazů. Systém fakturace a výkazů společnosti Energy ForEver pro správu a tisk formulářů nutných k provozu fotovoltaických elektráren je webová aplikace, která umožňuje po přihlášení uživatele zadávání dat a odeslání faktur a výkazů.

Webové rozhraní umožňuje zadání dat o výrobci, jeho výrobně a odečtu elektroměru. Z jednotlivých odečtu lze vygenerovat měsíční výkaz a vytvořit fakturu, které se vygenerují do formátu *pdf*, odkud je možné je vytisknout nebo uložit do PC. Na základě odečtů lze též vytvořit Čtvrtletní výkaz pro ERÚ, který je možné opět vygenerovat jako *pdf* soubor, nebo přímo odeslat mailem na ERÚ. Všechny vytvořené výkazy lze zobrazit v historii.

⁴ <http://fakturace.energyforever.cz>

Energy ForEver s.r.o. | fakturace a výkazy - Mozilla Firefox

http://fakturace.energyforever.cz/admin/ moje-vyrobn

Energy ForEver Fakturace a výkazy

Přihlášen: Slunicko Josef

Moje údaje o výrobně elektřiny

Tučně vyznačená pole jsou povinná.

Název: FVE_Studenec_372

Ulice: ČP: 372

Město: Studenec PSČ: 51233

Režim: Zelený bonus

Číslo předávacího místa: 1000123456

Uvedení do provozu: Duben 2010

Výkon: 9.6 kWp

Napětí: 0.4 kV

Transformační ztráty: 0 %

Tarif: Dvoutarif, Vyberte, Jednotarif

Počáteční hodnoty elektroměru

Pokud uvádíte do provozu novou FVE, musíte vytvořit první výkaz a první fakturu, počáteční stavy elektroměru jsou hodnoty elektroměru při osazení (pokud elektroměry nebyly vynulovány). Datum nastavte na měsíc předcházející měsíci uvedením elektrárny do provozu.

Vztahující se k: Zář 2010

Pokud již FVE nějakou dobu provozujete a v minulosti jste tvořili výkazy sami, uveďte jako počáteční

Obrázek 2 - Energy ForEver, údaje o výrobně

Cena za registraci do služby Energy ForEver je 177 Kč včetně DPH na 12 měsíců.

Mezi výhody této webové aplikace patří jednoduché rozhraní s pouze několika základními, ale nejdůležitějšími, funkcemi. Náповěda je ve formě videí, které provází většinou scénářů, podle kterých bude uživatel postupovat při práci. Jednotlivé pole pro zadávání dat mají vedle sebe informační ikonu pro zobrazení popisu pole. Díky přítomnosti na serveru je aplikace vždy aktuální z hlediska legislativních a cenových změn. Že se jedná o online službu, kterou lze spustit z kteréhokoliv počítače připojeného k internetu, může být pro někoho velkou výhodou, pro někoho nevýhodou kvůli nutnosti internetového připojení. Mezi záporné vlastnosti může též patřit možnost pouze jedné výroby.

2.5 Albert

Již obsáhlejší produktem pro výkaznictví FVE je program Albert od Appro⁵. Po spuštění programu se zobrazí okno, jehož velikost nelze změnit, s několika tlačítky, pomocí níž se lze dostat ke všem funkcím programu. Z tohoto okna se spouštějí další modální okna s pevnou velikostí.

⁵ <http://www.volny.cz/appro/albert.htm>



Obrázek 3 - Albert, úvodní obrazovka

Program obsahuje několik funkčních modulů, kterými jsou:

- fakturace,
- tiskové sestavy,
- statistické grafy,
- editace údajů výrobce,
- číselník distributorů,
- seznam výroben daného výrobce.

Po zadání údajů o výrobci a jeho výrobnách, se v modulu fakturace tvoří podklady pro výkaznictví. Na každý měsíc se vytvoří jedna faktura (jedna pro distribuci a jedna pro přebytky), do které stačí zadat stavy elektroměru na konci měsíce.

Fakturace

Rok	Měs	Výrobné	FA distribuce	FA přebytky	Vystave.Σ	Výroba	Dodávka	Odběr	Odběr VNT	Distribuce	Přebytky
2011	5	1	2011050110	2011050120	1.6.2011	481	388	113		7 199,00	0,00
2011	4	1	2011040110	2011040120	1.5.2011	362	285	107		5 417,00	0,00
2011	3	1	2011030110	2011030120	1.4.2011	306	257	153		4 580,00	0,00
2011	2	1	2011020110	2011020120	2.3.2011	198	130	152		2 961,00	0,00
2011	1	1	2011010110	2011010120	1.2.2011	87	52	205		1 301,00	0,00
2009	12	1	2009120110	2009120120	27.1.2011	28	28	253		402,00	0,00

Fakturační období 2011 - 1
Období na fakturě 2011 - 01
Výrobné FVE Kopecký 1
Číslo faktury - distribuce 2011010110
Číslo faktury - přebytky 2011010120 Zelený bonus

Datumy
Datum vystavení 01.02.2011
Datum zdan. plnění 31.01.2011
Datum splatnosti 26.02.2011

Výkupní ceny
Výkupní 12,420 [Kč / kWh]
Za decentralní výrobu 0,064 [Kč / kWh]
Za výkup přebytků 0,000 [Kč / kWh]

Stavy elektroměrů
Výroba 96113677 **Odběr 96113677**

	Poč. stav	Kon. stav	Za měsíc	Za rok
Svorková výroba	3 148	3 235	87	87
Dodávka 2.8.0	2 417	2 469	52	52
Odběr 1.8.0	2 226	2 431	205	691
Odběr VT 1.8.1	0			
Odběr NT 1.8.2	0			

Zelené bonusy
Zelené bonusy
za měsíc 1 080,54 [Kč]
za rok 1 080,54 [Kč]
Vlastní spotřeba
za měsíc 35 [kWh]
za rok 35 [kWh]

Fakturováno
Sazba DPH 20 %
Základ DPH 1 083,87
DPH 216,77
Celkem s DPH 1 300,64
K úhradě 1 301,00

Distribuce 0,00
Přebytky 0,00

Poznámka

Zavřít, Opravit, Přidat, Kopírovat, Smazat, Tisk

Obrázek 4 – Albert, okno fakturace

Výstupní sestavy se vytvoří v okně fakturací, odkud lze vytisknout fakturu, měsíční a čtvrtletní hlášení o výrobě elektřiny, roční přehled o výrobě elektřiny a roční přehled o fakturaci. Další sestavy lze vytisknout v hlavním okně aplikace stiskem tlačítka sestavy, které nabízí tisk výstupů pro ERÚ, což jsou čtvrtletní výkazy o výrobě elektřiny a hlášení o předpokládané výrobě elektřiny a výstupy pro potřeby výrobce, jako porovnání plánu a skutečnosti, roční přehled o fakturaci a roční přehled o výrobě.

Z hlavního okna softwaru Albert je též přístupný modul grafů, který umožňuje zobrazení grafů se statistickými údaji. Údaje lze zobrazit jako čárový, sloupcový nebo rovinný graf, zadat všechny výroby nebo vybrat pouze jednu. Data je možné omezit rozsahem roků od – do a výběrem jedné nebo více výroben. Zobrazené grafy je možné posléze vytisknout. Na výběr jsou grafy výroby po měsících nebo rocích, nebo porovnání předpokládaného plánu výroby se skutečností. Zajímavým grafem je graf návratnosti investice, kde je vidět jak se k investičním nákladům postupně přibližuje zisk z provozu FVE.

Samotná aplikace je se skládá z jednoho spustitelného *exe* souboru, souboru nápovědy, *ini* souboru a několika datovými *dat* soubory se záznamy pro výrobce, výrobní, distributory a faktury. Albert není potřeba instalovat, stačí spustit *exe* soubor. Díky tomu je možné přenášet program mezi více počítači, stačí přenést celou složku s programem, který lze spustit na jakémkoliv PC s OS Windows.

Cena programu ve verzi pro koncového výrobce elektřiny je 1000 Kč bez DPH a manipulačního poplatku 100 Kč. Zneužití licence je hlídáno tak, že je každému zákazníkovi přeloženo *exe*, které již obsahuje jeho název firmy, IČ, DIČ, číslo licence a fakturační údaje a sídlo finančního úřadu. Tím je zabezpečeno, že aplikaci nemůže plnohodnotně využívat jiný výrobce.

Ovládání programu je velmi jednoduché a intuitivní. Uživatel hned ví, co se od něho očekává. Jednoduchost programu je oproti tomu vyvážena určitou strohostí grafického vzhledu, která nemusí být hodnocena jako záporná vlastnost. Jen se odlišuje od současného trendu tvorby aplikací, kdy je kladen velký důraz i na vzhled uživatelského rozhraní.

Co ovšem může být hodnoceno záporně, jsou občasné chyby aplikace. Jak je vidět v okně fakturace (Obrázek 4), je faktura z ledna 2011, což je první faktura v roce 2011, která má ve stavu elektroměru odběr registr 1.8.0 stav za měsíc 205, ale za rok 691. To je jistě špatně, neboť se tento údaj tiskne na měsíční výkaz o výrobě elektřiny. Chyba mohla vzniknout neplynulým zadáváním dat, kdy je vidět, že během ledna 2011 byly zadány další dvě starší fakturace.

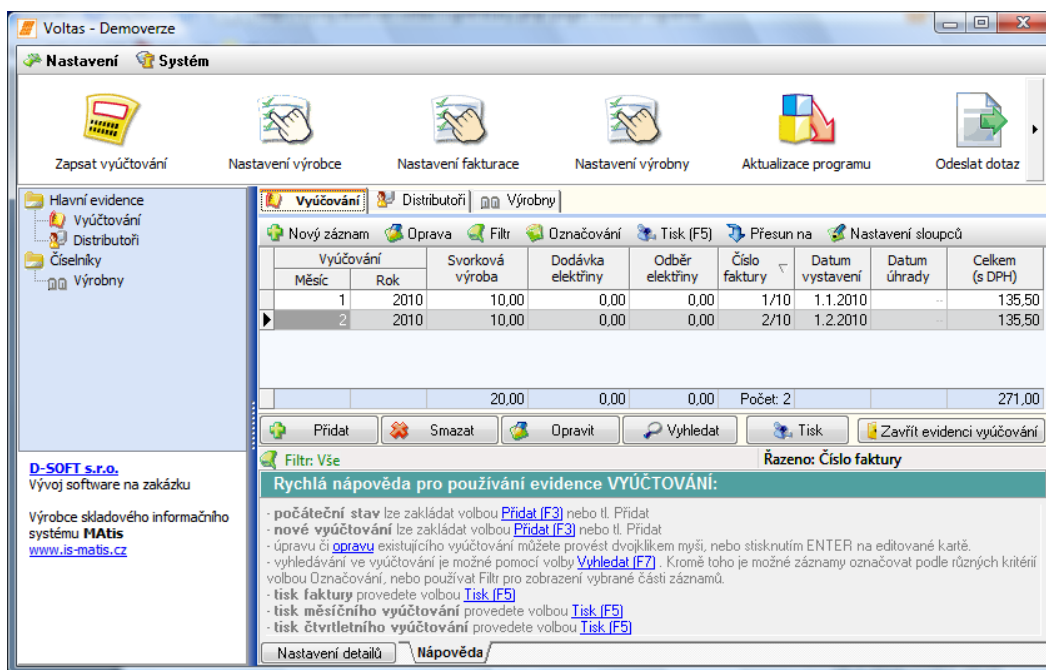
2.6 Voltas

Posledním popisovaným nástrojem pro fakturaci a výkaznictví FVE je software s názvem Voltas⁶ z dílny společnosti D-SOFT s.r.o. Voltas se dodává jako instalační balík, který je potřeba nainstalovat. Jako zdroj dat využíváno MS Access, tiskové sestavy jsou tvořeny pomocí nástroje FastReport.

Uživatelské rozhraní používá pro ovládání programu dva prvky. Těmi jsou panel v horní části okna, který obsahuje velké ikony s přístupem k hlavním funkcím programu, což je vytvoření vyúčtování, nastavení výrobce, nastavení výroben, nastavení fakturace a aktualizace programu. Nad těmito ikonami je ještě jednoduché menu, které obsahuje hlavně nastavení programu, licence, dotazy na podporu dodavatele softwaru, atp. Druhý způsob přístupu k ovládání aplikace je pomocí treeview v levé části okna, které stromově rozčleněné na funkční oblasti (hlavní evidence, číselníky, tisky a informační funkce). Hlavní evidence obsahuje přístup do seznamů vydaných faktur, distributorů a výroby. Seznam výroby se používá u výrobny, která má faktura nastavenou na delší období než měsíční. Při měsíčním období se výroba zapisuje pouze do fakturace. Oblast číselníky obsahuje pouze seznam výroben. Tisky obsahují tiskové sestavy ročního hlášení předpokládané výroby, měsíční a čtvrtletní výkaz pro ERÚ a dvě položky menu (faktura a měsíční výkaz), které však pouze zobrazují informaci o možnosti tisku v seznamu faktur. Informační funkce obsahují hodnocení výroby a zisku. Jsou to přehledy dat za určité období zobrazené v tabulce. Ze zobrazených dat lze vybrat konkrétní údaje do grafu.

Na zbývající ploše okna programu mohou být zobrazeny gridy s různými seznamy dat, např. vydané faktury. Těchto seznamů může být otevřeno více, neboť jsou zobrazovány pomocí záložek. Jednotlivým gridům lze nastavit zobrazené sloupce, jejich pořadí a třídění dat.

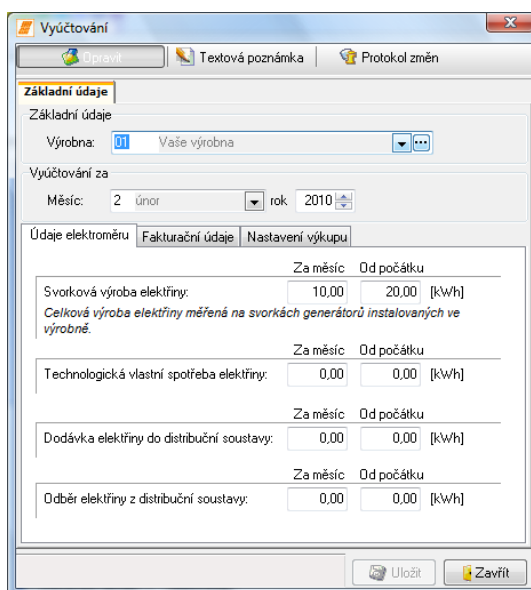
⁶ <http://www.voltas.cz/>



Obrázek 5 - Voltas

Editace konkrétního záznamu už probíhá v modálním okně. Jednotlivé údaje jsou v editačních oknech uspořádány opět pomocí záložek na logické celky. Každé editační pole obsahu podrobnou nápovědu, která se průběžně zobrazuje v k tomu vyhrazené části okna.

Tvorba faktury pracuje v režimu průvodce, kdy jsou postupně zadávány data, a stiskem tlačítka *Další* se uživatel posouvá na další krok. Po uložení faktury je uživateli zobrazeno okno, kde si může vytisknout všechny potřebné údaje (faktura, měsíční výkaz, atd.).



Obrázek 6 - Voltas, editace vytvořené faktury

Licence programu Voltas stojí 1800 Kč bez DPH a je včetně všech následujících aktualizací a obsahuje také emailovou podporu od výrobce. Po zakoupení se instaluje, ale na další PC lze přenést prostým zkopírováním složky. Musí se však spouštět ze stejného adresáře jako v původním počítači. Přínosnou funkcí aplikace je tvorba záloh. Do nastaveného adresáře se vytvářejí zálohy databázového souboru, které je možné v případě potřeby použít k obnově dat.

Voltas je velmi dobrý software, který se intuitivně ovládá a uživateli pomáhá propracovanou průběžně se zobrazující nápovědou.

3 Návrh softwaru pro vykazování výroby elektrické energie

3.1 Definice zadání

Hlavním cílem této bakalářské práce je vytvoření softwaru, který bude umožňovat evidenci, vykazování a fakturaci výroby elektrické energie. Návrh vznikl studiem konkurenčních produktů popsanych v předchozí kapitole. Jelikož to má být aplikace pro spuštění v operačním systému, vychází návrh z produktu Albert a Voltas. Obě aplikace funkčně plní svůj účel, ale trochu rozdílné je jejich uživatelské rozhraní. Albert sází spíše na jednoduchost, jeho okna mají pevnou velikost. Oproti tomu Voltas je na první o něco složitější na ovládání a má více možností na ovládání.

Prozkoumáním jejich kladných a záporných vlastností konkurence se postupně vykrytalizovaly požadovaná charakteristika budoucí aplikace.

Hlavními požadavky byly:

- programovací jazyk Delphi,
- možnost spuštění aplikace bez instalace, jako portable aplikace,
- zdroj dat databáze bez nutnosti instalace,
- jednoduchost uživatelského rozhraní s ovládáním pomocí ikon na panelu v horní části okna,
- možnost licenčního klíče.

Těmito body se upřesnilo hlavně technologické zaměření, ale ani tento výběr nebyl jednoduchý.

3.2 Databáze

Při výběru datového uložení aplikace bylo dle zadání na výběr datový soubor (používáno SW Albert), *xml* soubor nebo databáze. Souborové uložení je výhodné pro požadavek portable aplikace, ale práce s nimi není tak komfortní jako s relační databází. Aby se však nemuselo nic instalovat na PC, na kterém aplikace poběží, musí být databáze ve verzi embedded, tzn. pouze databázový soubor, případně nějaké *dll* soubory jakožto engine databáze bez potřeby instalace.

Produktů, které by se k tomuto účelu dalo použít je několik, z neznámějších např. *SQLite*, *MySQL*, databázový soubor *MS Access*, *Firebird*. Ale jelikož Firebird je odvozen z Borland InterbaseBase 6.0, který byl součástí vývojového prostředí Delphi, padla volba na něho.

3.2.1 Firebird

Databáze Firebird, jejíž poslední verze je 2.5.1, má několik edicí (SuperServer, SuperClassic, Classic a Embedded), z nichž každá má své specifické vlastnosti a je určena k jinému účelu. Pro účely vývoje software pro vykazování se hodí Firebird Embedded, jedná se o klientskou knihovnu, která v sobě již obsahuje samotný engine serveru.

V této edici se Firebird skládá z databázového *fdb* souboru, několika dll (*gds32.dll*, *ib_util.dll*, *icudt30.dll*, *icuin30.dll*, *icuuc30.dll*, *fbintl.dll* a *fbudf.dll*) a konfiguračních souborů (*firebird.conf* a *firebird.msg*). Všechny soubory stačí umístit do stejné složky jako spustitelný soubor aplikace, která se k databázi připojuje, ta si načte knihovnu *gds32.dll* (někdy může být pojmenovaný jako *fbclient.dll*) se již postará o správnou funkčnost serveru databáze. Pokud by se soubory Firebirdu, kromě klientského *gds32.dll*, ten musí zůstat u aplikace, nacházeli v jiném adresáři, je ještě potřeba v konfiguračním souboru *firebird.conf* nastavit cestu k adresáři Firebirdu direktivou *RootDirectory = c:\adresar*.

Embedded verze Firebirdu má oproti běžným verzím několik omezení, které však vyplývají již ze samotné podstaty embedded.

- K serveru se lze připojit pouze pomocí lokálního přístupu k souboru, nikoliv pomocí TCP/IP připojení s názvem serveru.
- Nepoužívá se bezpečnostní databáze, což znamená, že uživatelské jméno slouží pouze k určení přístupu uživatele k databázovým objektům.
- Není možné přistupovat k jednomu databázovému souboru z více procesů, neboť je při přístupu k němu klientskou knihovnou exkluzivně uzamknut jen pro danou aplikaci.

Pro daný účel aplikace, ke které přistupuje vždy jen jeden uživatel a databáze slouží jen jako uložisko dat, je Firebird Embedded velmi vhodný nástroj.

3.3 Moduly aplikace

Z hlediska funkčnosti by aplikace měla být rozdělena do několika modulů, které budou sloužit každý svému účelu.

3.3.1 Číselníky

Již na první pohled je zřejmé, že aplikace by měla obsahovat několik jakýchsi číselníků, neboli seznamů dat, které se nastaví na začátku užívání aplikace a nebudou se příliš měnit.

Prvním z nich je seznam *výrobců*. Není to sice číselník v pravém slova smyslu, neboť takto koncipovaná aplikace obsahuje vždy pouze jednoho výrobce, ale přesto je ho možné funkčností zařadit mezi číselníky.

Na záznam výrobce navazuje seznam jeho *výroben*. Číselník výrobců obsahuje jednotlivé záznamy výroben, které výrobce provozuje a údaje o nich. Číselník výroben se také nebude příliš měnit a pro menší výrobce bude časté, že zde bude pouze jeden záznam.

Posledním číselníkem je seznam *distributorů*, který obsahuje údaje o provozovatelích DS důležitých hlavně pro fakturaci vyrobené elektrické energie. Ani zde se data nebudou příliš měnit a naopak by mohla být vyplněna v již prázdné databázi určené k distribuci koncovým zákazníkům.

3.3.2 Fakturace

Modul *fakturace* by měl sloužit k zadávání dat jednotlivých odečtů stavů elektroměru a vytvářet z nich počítat množství vyrobené elektrické energie. Podle těchto dat a v číselníku výroben nastavených výkupních cen se vypočítá fakturovaná částka za výrobu.

3.3.3 Sestavy

Po vytvoření faktury, je potřeba ji předat dále distributorovi. To by měl zajišťovat modul *sestav*. Dále by měl umět generovat další výstupní sestavy, které musí výrobce elektřiny z OZE odevzdávat distributorovi a ERÚ.

Výrobce musí odevzdávat distributorovi:

- fakturu,
- měsíční výkaz o výrobě elektrické energie z OZE

a ERÚ posílá:

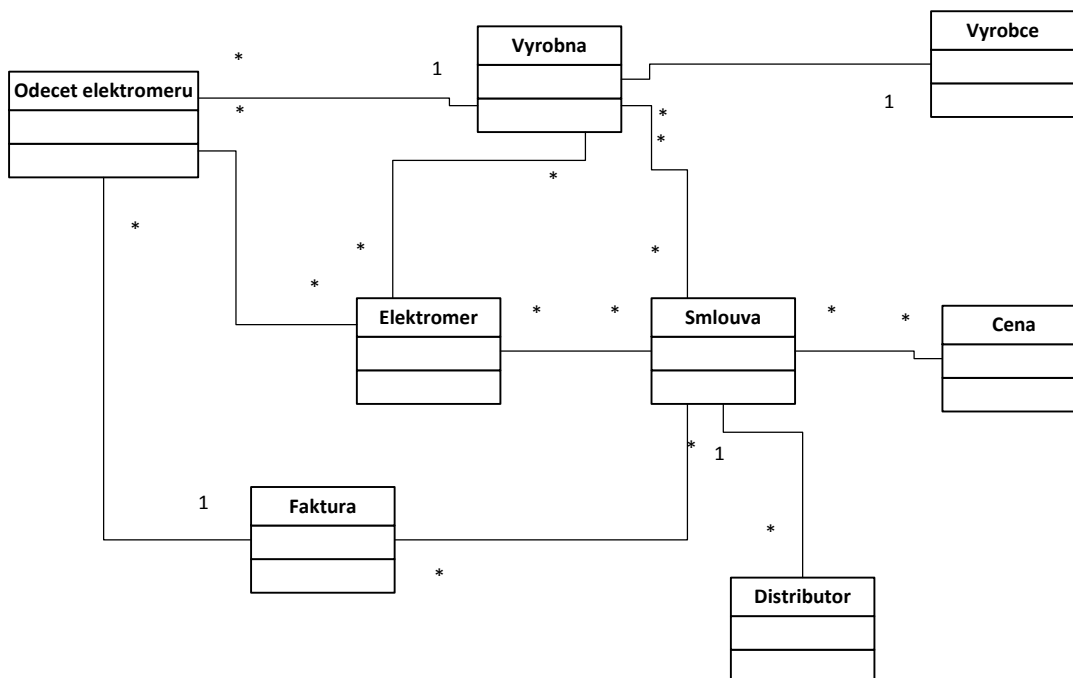
- čtvrtletní výkaz o výrobě elektrické energie z OZE VA 1-12,
- hlášení o předpokládaném množství elektřiny vyrobené z OZE,
- výrobce z OZE větší než 0,5 MWe vyplňuje měsíční výkaz ERÚ 1-12

Modul sestav by dále mohl umět generovat výstupy ve formě sestav či grafů, které budou ukazovat přehledy výroby, či porovnávat plánovanou a skutečnou výrobu.

3.4 Rozložení objektů

Částečně i z rozdělení aplikace na moduly vyplývají již nějaké objekty, z kterých by se měl celý software skládat. Provozovatelem FVE a tím i této aplikace je jeden *výrobce*, což je jeden objekt. Výrobce má jednu nebo několik *výroben*. Každá z těchto výroben má svůj *elektroměr*. Výrobce uzavírá pro každou ze svých výroben s *distributorem smlouvu*, která se váže ke konkrétnímu číslu elektroměru. Ve smlouvě s distributorem jsou dané *ceny* za elektřinu. Výrobce na konci každého měsíce, ale může i častěji, provádí odečet *stavu elektroměru*, kterým zaznamenává údaje jedné výroby s konkrétním elektroměrem. Z odečtu stavu elektroměru, který je z posledního dne v měsíci potom výrobce vytváří

fakturu s celkovou sumou výroby, případně spotřebou, za daný měsíc. Fakturu zasílá výrobce distributorovi na základě s ním uzavřené smlouvy. Takto popsané vztahy jsou graficky znázorněné ve zjednodušeném diagramu tříd (Obrázek 7), který ukazuje pouze objekty a vztahy mezi nimi, nikoliv však už jejich vlastnosti nebo metody.



Obrázek 7 - zjednodušený návrh diagramu tříd

Na takto připravený návrh objektů je možné navázat při definování databázové struktury. Vazby mezi objekty jsou již popsány a již stačí jen jednotlivým objektům definovat jejich vlastnosti.

3.5 Licence

Vytvářená aplikace by měla umět zabezpečení licencování pomocí nějakého mechanismu. Podle vydefinovaných požadavků by měla být licence určena jednomu výrobcí, ale ten by měl mít možnost spouštět SW na více počítačích.

Jednou možností, jak zabránit užívání aplikace někomu jinému nežli výrobcí, kterému je licence určena je přístup, který zvolil konkurenční SW Albert. Každý zákazník získá aplikaci přeloženou speciálně pro něho s již vyplněnými identifikačními údaji o něm. Takto je aplikace velmi dobře ochráněna proti zneužití, ale pro tvůrce SW přináší práci navíc s generování exe pro každého zákazníka.

Další způsobem je licencování kontrolování licenčního klíče. Ten je vhodné generovat na základě nějakého údaje. Může jím být třeba nějaká identifikace PC, ale tím by nebyl splněn požadavek na přenositelnost aplikace mezi více počítači. Protože je licence vydávána konkrétnímu výrobcí, je možné navázat klíč na nějaký identifikační údaj výrobce, nebo

ještě lépe na více údajů. Identifikační data výrobce, které by se neměla nikdy změnit, jsou název výrobce, IČ, DIČ a číslo licence ERÚ.

Ochrana aplikace tvořené pro tuto bakalářskou práci bude tedy tvořena licenčním klíčem, který bude generovaný na základě IČ, DIČ a čísla licence výrobce. Aplikace nebude mít žádný demo mód, proto ji by bez zadání povinných údajů výrobce a licenčního klíče nemělo jít spustit. Při prvním spuštění by měl být uživatel v průvodci postupně vyzván k zadání všech potřebných údajů.

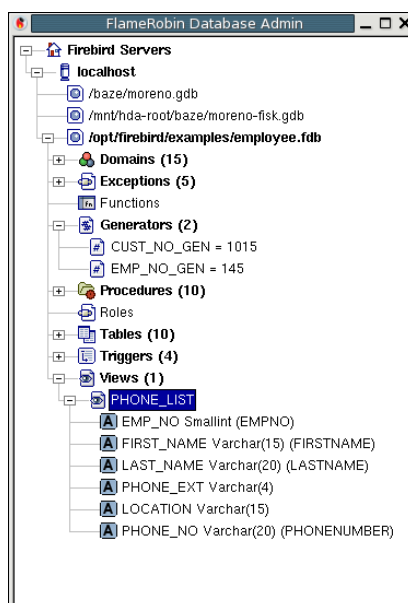
4 Vytvoření softwaru pro vykazování

4.1 Databázová struktura

Jako uložení dat je tedy pro vyvíjený software použita databáze Firebird Embedded 2.5.1. Struktura databázových tabulek je dána zjednodušeným objektovým modelem. Proto je možné přistoupit k návrhu databázové struktury.

4.1.1 Nástroje pro správu databáze

Jelikož Firebird Embedded je tvořen pouze několika dll soubory a ani pro jiné edice nenabízí Firebird, kromě konzol, žádný nástroj pro úpravu struktury databázových objektů. Je potřeba se porozhlédnout po jiném produktu třetí strany. Na webových stránkách Firebirdu je zmiňován open-source projekt FlameRobin⁷. Poskytuje velmi dobrý a jednoduchý nástroj pro správu Firebird databáze. Je ovšem potřeba znát syntaxi SQL příkazů ve Firebirdu, která se mírně odlišuje od SQL standardu.



Obrázek 8 - FlameRobin, základní okno

Na svou jednoduchost to je ale velmi mocný nástroj umožňující správu databázových objektů i dat.

4.1.2 Domény

Firebird neumožňuje tvorbu vlastních datových typů, ale místo toho má *domény*. Je to vlastně pojmenování, které musí být jedinečné, určité množiny vlastností, jako je datový typ, implicitní hodnota, možnost sloupce být NULL, znaková sada nebo omezení hodnot. Při definici sloupce tabulky je možné místo datového typu zadat doménu a sloupec poté přebírá všechny vlastnosti domény.

⁷ <http://www.flamerobin.org/>

Pro lepší přehlednost a popisnost obsahu tabulek vzniknuly ve vyvíjené DB dvě domény:

- money – datový typ decimal(12,4)
- boolean – datový typ smallint, defaultní hodnota 0.

4.1.3 Generátory

V některých databázových systémech je možné nastavit číselným sloupcům tabulky nastavit příznak autoinkrement, což znamená, že při každém přidání záznamu sloupci přiřazena postupně se zvětšující hodnota z číselné řady. To je výhodné hlavně u primárních klíčů.

Tuto vlastnost Firebird neposkytuje, místo toho má však tzv. generátory. Což jsou pojmenované databázové objekty, které v sobě uchovávají naposledy použitou hodnotu z číselné řady a pomocí funkce *GEN_ID()*, která zvýší hodnotu generátoru a tuto hodnotu vrátí, umožňují autoinkrementálně nastavovat hodnotu sloupce. Toto se děje nejčastěji v *before insert triggeru*.

Příklad vytvoření generátoru:

```
CREATE GENERATOR VYROBNA_ID_VYROBNA_GEN;
```

4.1.4 Triggery

Firebird umožňuje nastavit tabulce trigger před nebo po vložení, smazání nebo změně dat. Triggerů je využíváno hlavně v součinnosti s generátory pro nastavování autoinkrementální hodnoty sloupce.

Příklad vytvoření triggeru:

```
SET TERM ^ ;
CREATE TRIGGER BI_VYROBNA_ID_VYROBNA FOR VYROBNA ACTIVE
BEFORE INSERT POSITION 0
AS
BEGIN
    IF (NEW.ID_VYROBNA IS NULL) THEN
        NEW.ID_VYROBNA = GEN_ID(VYROBNA_ID_VYROBNA_GEN, 1);
END^
SET TERM ; ^
```

4.1.5 Tabulky

Struktura tabulek softwaru pro FVE vychází z předběžného návrhu popsaného v kapitole 3.4. Všechny takto vytvořené tabulky obsahují primární klíč s ID záznamu, jehož hodnota je triggerem a generátorem autoinkrementálně nastavována. Jedná se o tabulky:

- VYROBCE – tabulka pro záznam výrobce, jeho identifikační údaje, kontaktní a fakturační údaje, atd. Tabulka obsahuje sloupce PODPIS a RAZITKO, typu BLOB

pro uložení obrázků razítka a podpisu, které by mohlo být možné tisknout na výstupní sestavy.

- VYROBNA – záznamy s daty jednotlivých výroben, obsahují cizí klíč do tabulky VYROBCE.
- ELETROMER – obsahuje číslo, počáteční stavy elektroměru a tarifní sazbu. Cizím klíčem se odkazuje na tabulku VYROBNA.
- DISTRIBUTOR – údaje o distributorech, které jsou potřebné pro zobrazení v tiskových sestavách.
- SMLOUVA – cizí klíče se odkazují na tabulky VYROBNA, DISTRIBUTOR a ELEKTROMER. Obsahem tabulky jsou čísla smlouvy a objednávky, EAN předávacího místa a logickou hodnotu určující, zda je smlouva na podporu výroby formou zeleného bonusu nebo výkupních cen.
- CENA – cizí klíč ukazuje na tabulku SMLOUVA a určuje samotnou výši nasmlouvaných cen.
- ODECET – stavy elektroměru k datu, cizí klíč do tabulek VYROBNA a ELEKTROMER.
- FAKTURA – cizími klíči se váže na ODECET a SMLOUVA. Obsahem jsou sumy výroby, či spotřeby na daný měsíc a z těchto údajů vypočítané fakturované ceny.

Některé tabulky obsahují sloupec s NEAKT, doména BOOLEAN, který určuje, zda je daný záznam platný či nikoliv. Nastavením sloupce NEAKT na hodnotu 1 a založením nového záznamu je možné vést historii změn.

V databázi je ještě jedna tabulka s názvem INI, ve které jsou ve sloupcích SEKCE, NAZEV a HODNOTA uloženy hodnoty nastavení, která bývají ukládána do ini souboru. Je zde uložen např. nastavený licenční klíč. Primárním klíčem jsou sloupce SEKCE a NAZEV.

Příklad vytvoření tabulky:

```
CREATE TABLE CENA
(
  ID_CENA Integer NOT NULL,
  ID_SMLOUVA Smallint,
  VYKUPNI MONEY,
  DECENTRALNI_VYROBA MONEY,
  PREBYTKY MONEY,
  PRIMARY KEY (ID_CENA)
);
ALTER TABLE CENA ADD CONSTRAINT FK_CENA
FOREIGN KEY (ID_SMLOUVA) REFERENCES SMLOUVA (ID_SMLOUVA);
GRANT DELETE, INSERT, REFERENCES, SELECT, UPDATE
ON CENA TO SYSDBA WITH GRANT OPTION;
```

4.1.6 Procedurey

V databázovém systému Firebird je možné vytvářet i vnořené procedury. Syntaxe je opět trochu odlišná od standardu, kdy mezi definicí názvu procedury a samotným beginem je definice výstupních proměnných a definice proměnných aplikace. Pro potřeby této práce vznikly procedury dvě:

- IU_INI_ZAZNAM – Procedura pro zápis do tabulky INI. Pokud záznam již existuje, tak se updatuje na novou hodnotu, jinak se vytvoří nový záznam.
- NOVA_FAKTURA – Počítá data nutná pro vytvoření nové faktury. Výsledné hodnoty vrací ve výstupních proměnných. Po kontrole uživatel v aplikaci se teprve založí nový záznam o faktuře (už z vývojového prostředí Delphi).

Příklad definice procedury:

```
SET TERM ^ ;
CREATE PROCEDURE IU_INI_ZAZNAM (
    SEKCE Varchar(50),
    NAZEV Varchar(50),
    HODNOTA Varchar(50) )
AS
BEGIN
    if (exists(select * from INI where SEKCE = :sekce and NAZEV = :nazev))
    then
        update INI set HODNOTA = :hodnota where SEKCE = :sekce and NAZEV =
        :nazev;
    else
        insert into INI(SEKCE, NAZEV, HODNOTA) values (:sekce, :nazev,
        :hodnota);
    SUSPEND;
END^
SET TERM ; ^
```

4.2 Potřebné komponenty pro Delphi

Samotná spustitelná aplikace vznikala ve vývojovém prostředí Delphi 2009, které již samo o sobě nabízí dostatek prostředků pro vývoj dle navržených požadavků. Ale bylo potřeba dodat další komponenty a unity, spíše kvůli lepšímu vzhledu, než kvůli funkčnosti.

4.2.1 IceTabSet

Na navržený způsob zobrazování v záložkách, podobně jako v současných internetových prohlížečích bylo potřeba sehnat nějaký vhodný systém tabu, protože PageControl, který je obsažený již v prostředí Delphi tak líbivý vzhled, který by odpovídal současným trendům. Náhradou se nestal jiný page control, ale tab control s názvem IceTabSet⁸, který má vzhled podobný prohlížeči Google Chrome.

Funkčnosti podobné jako u page control je docíleno tím, že místo stránek jsou pod lištu IceTabSetu vloženy Framy, kterým se postupně nastavuje vlastnost visible podle toho,

⁸ <http://sourceforge.net/projects/icetabset/>

který tab je zrovna vybraný. Způsob zobrazování dat ve Framech je výhodný z toho důvodu, že je možné mít stejný vzhled např. editačního okna výrobce jak na hlavním formuláři, tak v nějakém jiném modálním okně.

Pro správné nainstalování komponenty je potřebná ještě potřeba Delphi GDI+ API⁹. Dále samotná komponenta IceTabSet obsahuje chybu, kdy při zavírání tabu kliknutím na křížek při pohybu myši se tab nezavře, ale přepne se do režimu přesunu. Tuto chybu lze odstranit upravením podmínky v metodě MouseMove. Místo chybného podmínky:

```
if fCanDragTabs and (ssLeft in Shift) and (fIxTabStartDrag >= 0) then  
lépe funguje tato:
```

```
if fCanDragTabs and (ssLeft in Shift) and (fIxTabStartDrag >= 0) and not  
IsInCloseButton(Tab, X, Y) then
```

4.2.2 SMDBGrid

Ani DBGrid, který je v prostředí Delphi nemá zrovna nejmodernější vzhled a také nemá příliš možností nastavení. Proto použita komponenta SMDBGrid¹⁰, která je součástí balíku komponent SMComponents. Pouze stačí otevřít, přeložit a instalovat projekt určený pro příslušnou verzi Delphi.

Kromě vylepšení designu přináší SMDBGrid i lepší možnosti úpravy vzhledu gridu, ale i způsobu zobrazení dat. Grid umožňuje zalamování textu v řádcích s daty i v titulku, pro boolean data zobrazuje checkbox, atd.

4.2.3 Rave Reports

Výsledné výstupy dat z aplikace jsou prováděny pomocí tiskových sestav. Tyto sestavy musí být vytvořeny v nějaké reportovací službě. Přímo v prostředí Delphi 2009 je obsažen Rave Reports. Sestavy vytvořené pomocí Rave Reports je možné mít jako samostatný soubor v adresáři s aplikací, nebo jako součást přeloženého exe.

Rave Reports poskytují přehledný designer pro tvorbu sestav. Data do sestav lze předávat pomocí textových parametrů, přímým přístupem k jednotlivým komponentám sestavy přímo ze zdrojového kódu Delphi nebo napojením sestavy na databázi.

4.2.4 WinCrypt

Pro generování licenčního klíče jsou potřeba nějaké hashovací funkce. Delphi bohužel žádné neobsahuje, proto je potřeba použít cizí řešení. Hashovací funkce jsou obsaženy přímo v OS Windows a pro přístup k nim existuje se používá unit WinCrypt.pas. Jedná se o přepis hlavičkového souboru jazyka C do Delphi.

⁹ <http://www.prodigy.com/files/gdiplus.zip>

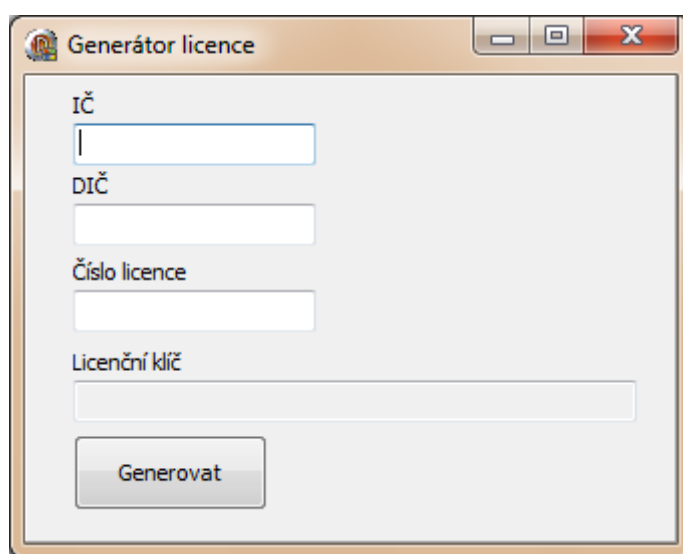
¹⁰ <http://www.scalabium.com/smdbgrid.htm>

4.3 Unity v Delphi

Samotný software je tvořený dvěma aplikacemi. Jednou hlavní a druhou podpůrnou, což je generátor licence. Hlavní aplikace se skládá z datového modulu, hlavního formuláře a několika framů umístěných na hlavním formuláři. Po prvním spuštění se zobrazuje modální okno s průvodcem nastavení.

4.3.1 Generátor Licence

Generátor licence je jednoduché okno, které na základě zadaných dat IČ, DIČ a čísla licence ERÚ vygeneruje licenční klíč. Generování klíče provádí unit *LicenceUnit*, která je popsána níže.



Obrázek 9 - Generátor licence

4.3.2 dmMain

Datový modul aplikace zajišťuje připojení k databázi a jsou na něm umístěny TibTable, případně TibQuery a objekty pro výrobu a faktur, jejichž data jsou takto sdílena pro ostatní unity aplikace. Dále obsahuje komponenty pro přístup k tiskovým sestavám Rave Reports. Na dmMain je možné ještě najít TImageList, který používají všechny akce (TAction) v celé aplikaci. TImageList je zatím prázdný, ale tímto způsobem bude možné přidat do aplikace ikony při případném komerčním nasazení.

4.3.3 ListFrame

Framy, které se zobrazují v hlavním okně programu, jsou dvou typů, seznam dat s gridem a editační okno, nebo spíše editační frame.

Všechny framy s gridem jsou odvozeny od stejného předka a tím je *ListFrame*. Na framu je zobrazený grid, který zobrazuje data z TDataSetu dsData a ve spodní části je panel s ovládacími tlačítky pro přidání, změnu a smazání záznamu.

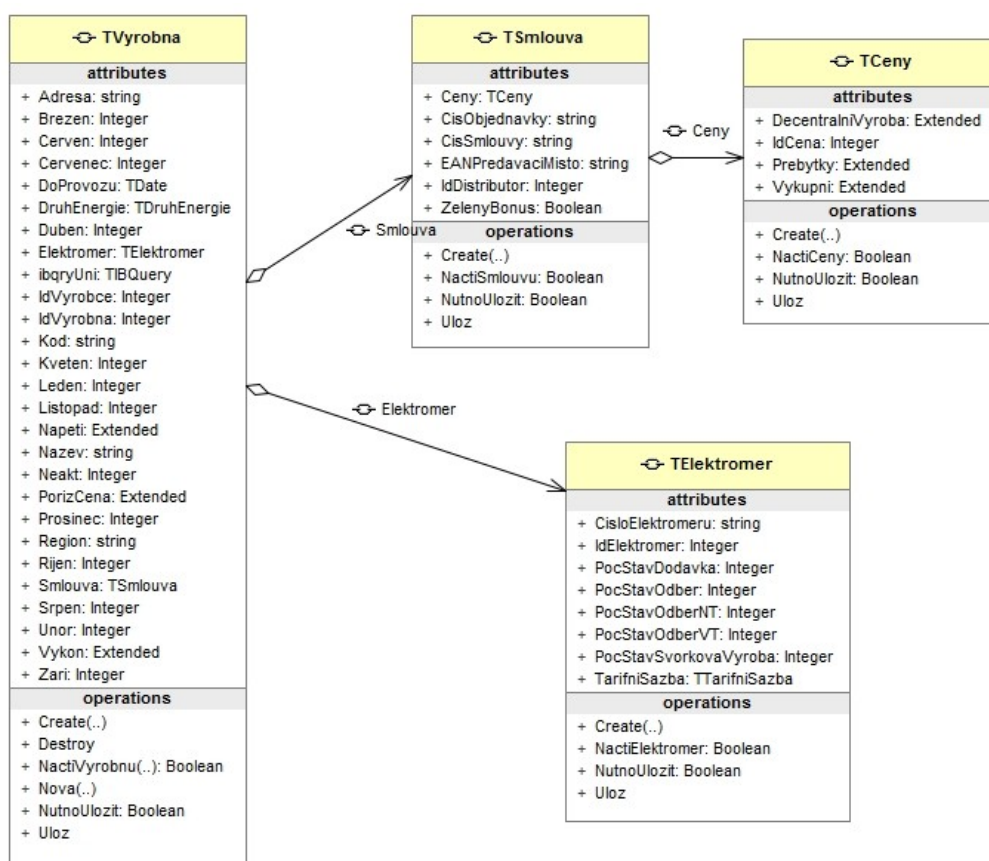
4.3.4 EditFrame

Společným předkem pro všechny editační framy je EditFrame. Ve spodní části je panel s tlačítky pro uložení dat (Uložit), uložení dat se zavřením otevřeného tabu (Ok) a uzavření tabu bez uložení dat (Storno).

EditFrame je pracuje s TDataSetem dsData, který mu přiřazuje potomek ListFramu, jehož data edituje. Metody pro uložení dat, zrušení změn dat, atd. pracují právě s dsData. V některých potomcích (InvoiceEditFrame a ProdPlaceEditFrame) se nepracuje s dsData, nýbrž s objekty TFaktura a TVyrobná, proto se nepoužívá poděděná funkcionalita, ale je přepsána jinou.

4.3.5 ProdPlaceUnit

Unit s jedním záznamem konkrétní výroby. Obsahuje objekty TVyrobná, TElektromer, TSmlouva a TCeny. Všechny objekty jsou na sebe navazující stejně jako databázové tabulky. Všechny třídy obsahují metody pro načtení dat z DB, uložení dat do DB, metodu pro zjištění, jestli se změnil nějaké atributy, atd.



Obrázek 10 - Diagram tříd unitu ProdPlaceUnit

4.3.6 InvoiceUnit

Třída TFaktura pro práci s jednou fakturou. Obsahuje metody pro načtení existující faktury z DB, uložení faktury do DB a vytvoření nové faktury (zavoláním SQL procedury NOVA_FAKTURA).

4.3.7 DefUnit

Definiční unit, který obsahuje definice společné pro celou aplikaci. Jsou zde definované výčtové typy, několik resourcestringů a funkce, např. funkce MyMessageDlg, která na rozdíl od MessageDlg zobrazuje popisy okna a tlačítek dialogu česky.

4.3.8 LicenceUnit

LicenceUnit obsahuje třídu TLicence, starající se o vygenerování licenčního klíče. Třída obsahuje metodu H, typu string, která zahashuje zadaný řetězec pomocí zadaného hashe. Pro vytvoření hashe jsou použity funkce z unitu WinCrypt.pas. Další metodou použitou pro generování klíče je Sh, typu string. Ta zadanému stringu zrcadlově převrátí znaky, ne však všechny, ale každý druhý.

Výsledný vygenerovaný licenční klíč vznikne postupným používáním metod H a Sh.

```
FLicKlic := Sh(H(Sh(H(FICO,CALG_SHA) + H(FCisLicence,CALG_MD5) +  
H(FDIC,CALG_MD4)), CALG_SHA1));  
FLicKlic := Sh(FLicKlic + H(FLicKlic,CALG_MD5));  
FLicKlic := Sh(FLicKlic + H(FLicKlic,CALG_SHA));
```

Takto vygenerovaný řetězec je příliš dlouhý, proto jsou z něho do licenčního klíče vybrány jen některé znaky (vždy ze stejných pozic).

4.3.9 IniUnit

Obsahuje TMyIni, což je kolekce ini záznamů. Tyto záznamy jsou při startu programu načteny z DB a při další práci už se pracuje pouze s kolekcí. Při zápisu ini záznamu jsou data hned sesynchronizována do databáze.

Zdrojový kód unitu je v příloze B.

Závěr

Cílem práce bylo vytvoření softwaru pro vykazování výroby elektřiny z OZE, který by se měl vyrovnat svým konkurentům a případně by měl nabídnout ještě něco navíc. U svých dvou konkurentů, Albert a Voltas, se software inspiroval hlavně v množství potřebných údajů a ve způsobu ovládání. Tyto SW mají také své chyby a právě jich se snažila vytvořená aplikace vyvarovat.

Základní předpoklady na aplikaci byly splněny. Software je vytvořený v programovacím jazyku Delphi a jako zdroj dat používá databázi Firebird. Pro své spuštění nepotřebuje nic instalovat do systému, proto může fungovat jako portable aplikace.

V současném stavu splňuje aplikace všechny legislativní požadavky na výkaznictví výrobce elektrické energie z fotovoltaických panelů. Nepodařilo se dopracovat doplňující statistické výstupy, jako jsou sestavy zobrazující návratnost investice, či porovnání plánu a skutečnosti. Tyto a jiné podobné výstupy by mohly být i ve formě grafů.

Z hlediska licencování je aplikace chráněna licenčním klíčem, který je generovaný na základě IČ, DIČ a čísla licence ERÚ výrobce. Tento způsob ochrany by měl být pro tento typ aplikace dostačující, avšak třída generátoru licence je připravena na případnou změnu způsobu licencování. Vhodné by mohlo být nějaké asymetrické šifrování údajů o výrobcí. Rozšíření by se mohl dočkat i samotný generátor, aby mohl poskytovat též seznam zákazníků, kteří si produkt zakoupili.

Grafické rozhraní aplikace je prozatím trochu strohé, bez jakýchkoliv ikon či obrázků. Aplikace však počítá s případným komerčním nasazením, kdy si pak prodávající firma může dodat svou vlastní grafiku. Připravený je i formulář zobrazující logo softwaru před spuštěním.

Zadaní práce aplikace určitě splnila, ale aby mohla být konkurence schopná na trhu, který mají již z velké míry rozdělený konkurenti, měla by poskytovat větší výhody. Těmito výhodami by mohlo být třeba automatické odečítání stavu elektroměru, nebo odeslání vygenerovaných sestav ve formátu pdf přímo distributorovi, či ERÚ.

Literatura

1. Sbírka zákonů. [Online] 05. Květen 2008.
<http://aplikace.mvcr.cz/archiv2008/sbirka/2005/sb066-05.pdf>.
2. Energetický regulační úřad. [Online] <http://www.eru.cz/>.
3. Firebird. [Online] <http://www.firebirdsql.org/>.
4. Program FAKTURACE pro fotovoltaické elektrárny. *Czech Nature Energy, a.s.*
[Online] <http://www.cne.cz/fotovoltaiicke-systemy/program-fakturace-pro-fve/>.
5. VOLTAS - softwara fotovoltaika, výkazy a fakturace. [Online] <http://www.voltas.cz/>.
6. Energy ForEver, s.r.o. [Online] <http://fakturace.energyforever.cz>.
7. Albert - obnovitelné zdroje FVE. *Appro aplikační programování.* [Online]
<http://www.volny.cz/appro/albert.htm>.

Příloha A – Diagram databázových tabulek

Příloha B – Zdrojový kód souboru IniUnit.pas

```
unit IniUnit;

interface

uses Classes, SysUtils, IBQuery, IBDatabase;

Type
  TMyIni = class;

  TMyIniSekce = class(TCollectionItem)
  private
    FSekce: string;
    FNazev: string;
    FHodnota: string;
  public
    constructor Create(AIni: TCollection); override;
    destructor Destroy; override;
  end;

  TMyIni = class(TCollection)
  private
    FOwner: TComponent;
    FibqryUni: TIBQuery;

    procedure SetItem(Index: integer; Value: TMyIniSekce);
    procedure UlozItem(Index: integer);
    function GetItem(Index: integer): TMyIniSekce;
    function GetSekce(ASekce, ANazev: string): TMyIniSekce;
  public
    constructor Create(AOwner: TComponent; ADatabase: TIBDatabase;
      ATrans: TIBTransaction);
    destructor Destroy; override;

    property SekceItem[index: integer]: TMyIniSekce read GetItem write
      SetItem;

    procedure NacistZDb;
    procedure WriteString(ASekce, ANazev, AHodnota: string);
    procedure WriteInteger(ASekce, ANazev: string; AHodnota: integer);
    function AddSekce: TMyIniSekce;
    function ReadString(ASekce, ANazev: string): string;
    function ReadInteger(ASekce, ANazev: string): integer;
  end;
implementation

uses DefUnit, Dialogs;

resourcestring
  sSQLSelectINI = 'select SEKCE, NAZEV, HODNOTA from INI';
  sSQLInsertINI = 'execute procedure IU_INI_ZAZNAM :SEKCE, :NAZEV,
:HODNOTA;';
  { TMyIniSekce }

{*****}
*****}
constructor TMyIniSekce.Create(AIni: TCollection);
begin
```

```

    inherited Create(AIni);
end;

{*****}
destructor TMyIniSekce.Destroy;
begin
    inherited Destroy;
end;

{ TMyIni }

{*****}
function TMyIni.AddSekce: TMyIniSekce;
begin
    Result := Add as TMyIniSekce
end;

{*****}
constructor TMyIni.Create(AOwner: TComponent; ADatabase: TIBDatabase;
ATrans: TIBTransaction);
begin
    inherited Create(TMyIniSekce);

    FOwner := AOwner;

    FibqryUni := TIBQuery.Create(AOwner);
    FibqryUni.Database := ADatabase;
    FibqryUni.Transaction := ATrans;
end;

{*****}
destructor TMyIni.Destroy;
begin
    try
        FreeAndNil(FibqryUni);
    finally
        inherited Destroy;
    end;
end;

{*****}
function TMyIni.GetItem(Index: integer): TMyIniSekce;
begin
    Result := TMyIniSekce(inherited GetItem(Index))
end;

{*****}
function TMyIni.GetSekce(ASekce, ANazev: string): TMyIniSekce;
var
    i: Integer;
begin
    Result := nil;
    for i := 0 to Count - 1 do
        begin

```



```

        if (SekceItem[i].FSekce = ASekce) and (SekceItem[i].FNazev = ANazev)
then
    begin
        Result := SekceItem[i];
        Exit;
    end;
end;

end;

{*****}
*****}
procedure TMyIni.NacistZDb;
var lSekce: TMyIniSekce;
begin
    Clear;
    with FibqryUni do
    begin
        if Active then
            Close;

        SQL.Text := sSQLSelectINI;

    try
        Open;

        while not eof do
        begin
            lSekce := AddSekce;

            lSekce.FSekce := FieldByName('SEKCE').AsString;
            lSekce.FNazev := FieldByName('NAZEV').AsString;
            lSekce.FHodnota := FieldByName('HODNOTA').AsString;
            Next;
        end;
    except
        MyMessageDlg(sChybaDat, mtWarning, [mbOk], 0);
    end;
end;

end;

{*****}
*****}
function TMyIni.ReadInteger(ASekce, ANazev: string): integer;
begin
    Result := StrToInt(ReadString(ASekce, ANazev));
end;

{*****}
*****}
function TMyIni.ReadString(ASekce, ANazev: string): string;
var lSekce: TMyIniSekce;
begin
    Result := EmptyStr;
    lSekce := GetSekce(ASekce, ANazev);

    if lSekce <> nil then
        Result := lSekce.FHodnota;
end;

```

```

{*****}
*****}
procedure TMyIni.SetItem(Index: integer; Value: TMyIniSekce);
begin
  inherited SetItem(Index, Value);
end;

{*****}
*****}
procedure TMyIni.UlozItem(Index: integer);
begin
  with FibqryUni do
  begin
    if Active then
      Close;
    SQL.Text := sSQLInsertINI;

    ParamByName('SEKCE').AsString := SekceItem[Index].FSekce;
    ParamByName('NAZEV').AsString := SekceItem[Index].FNazev;
    ParamByName('HODNOTA').AsString := SekceItem[Index].FHodnota;
    try
      ExecSQL;
    except
      MyMessageDlg(sChybaDat, mtWarning, [mbOk], 0);
    end;
  end;
end;

{*****}
*****}
procedure TMyIni.WriteInteger(ASekce, ANazev: string; AHodnota: integer);
begin
  WriteString(ASekce, ANazev, IntToStr(AHodnota));
end;

{*****}
*****}
procedure TMyIni.WriteString(ASekce, ANazev, AHodnota: string);
var lSekce: TMyIniSekce;
begin
  lSekce := GetSekce(ASekce, ANazev);
  if lSekce = nil then
  begin
    lSekce := AddSekce;
    lSekce.FSekce := ASekce;
    lSekce.FNazev := ANazev;
  end;

  lSekce.FHodnota := AHodnota;
  UlozItem(lSekce.Index);
end;

end.

```

Příloha C – Ukázky obrazovky aplikace

Software pro vykazování výroby elektrické energie z fotovoltaických panelů

Výrobný Výrobce Distributoři Odečet stavu elektroměru Fakturace

Uvítací obrazovka × Výrobce × Výrobný × Faktury ×

Období			Výroba						Cena	
Rok	Měsíc	Číslo faktury	Svorková výroba	Dodávka	Odběr	Odběr NT	Odběr VT	K úhradě	Cena s DPH	Cena bez DPH
2011	2	201102	198	130	152	0	0	2961	2960,976	
2011	1	201101	87	52	205	0	0	1301	1300,6416	

Přidat Změnit Smazat Tisk

Výrobna: Testovací výrobná 2

Software pro vykazování výroby elektrické energie z fotovoltaických panelů

Výrobní Výrobce Distributoři Odečet stavu elektroměru Fakturace

Uvítací obrazovka x Výrobce x Výrobní x Faktury x Výrobna x

Základní údaje

Kód výrobní 002

Název výrobní Testovací výrobná 2

Adresa výrobní Adresa 123314, 53002 Pardubice

Region (kraj) Prdubický

Druh energie

Fotovoltaická
 Větrná
 Vodní

Technické informace

Uvedeno do provozu 22. 2.2012 Napětí [kV] 0,40 Výkon [kWp] 3,40 Pořizovací cena [Kč] 399886,00

Smlouva

Distributor ČEZ Distribuce

Číslo objednávky 4600005714

Číslo smlouvy 2150793/0071/OZE/09

Předávací místo EAN EAN859182400707463103

Elektroměr

Výr. číslo elektroměru 96113677

Tarifní sazba
 Jednotarifní Dvoutarifní

Počáteční stavy:

Svorková výroba 3148

Dodávka do DS [2.8.0] 2417

Odběr z DS [1.8.0] 2226

Odběr z DS VT [1.8.1] 0

Odběr z DS NT [1.8.2] 0

Ceny

Výkupní cena [Kč / kWh] 12,42

Cena za decentralní výrobu [Kč / kWh] 0,06 Zelený bonus

Cena za výkup přebytků [Kč / kWh] 0,00

Plán výroby

Předpokládaná výroba v průběhu roku [kWh]

Leden
Únor
Březen
Duben
Květen
Červen
Červenec
Srpen
Září
Říjen
Listopad
Prosinec

Tisk plánu

Ok Uložit Storno

Výrobna: Testovací výrobná 2

Nový zápis stavu elektroměru

Datum odečtu 17.8.2012

Svorková výroba

Dodávka do DS [2.8.0]

Odběr z DS [1.8.0]

Odběr z DS VT [1.8.1]

Odběr z DS NT [1.8.2]

Uložit Storno