

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra počítačů



diplomová práce

**Systém pro evidenci a vyhodnocení poruch
parních turbogenerátorů ŠKODA**

Jan Dohnal

Vedoucí práce: Ing. Ivan Halaška

Studijní program: Elektrotechnika a informatika

Obor: Výpočetní technika

květen 2009

Prohlašuji, že jsem tuto práci vypracoval samostatně a řádně uvedl všechny informační zdroje. Dále prohlašuji, že nemám námitek proti nekomerčnímu využití této práce, zdrojových souborů nebo jejich částí katedrou počítačů elektrotechnické fakulty ČVUT.

V Praze dne 20.5.2009

Jan Dohnal

Seznam použitých zkratek

- AJAX - platforma doplňující Javascript (Asynchronous JavaScript and XML)
- API - rozhraní pro programování aplikací (Application Programming Interface)
- ASP - skriptovací platforma společnosti Microsoft (Active Server Pages)
- CGI - technologie generující webové stránky (Common Gateway Interface)
- CSS - tabulky kaskádových stylů (Cascading Style Sheets)
- CSV - formát dat s hodnotami oddělenými čárkou (Comma Separated Values)
- DB - databáze (DataBase)
- GNU - projekt zaměřený na svobodný software; rekurzivní zkratka (GNU's Not Unix)
- GUI - grafické uživatelské rozhraní (Grafic User Inteface)
- HTML - hypertextový značkovací jazyk (HyperText Markup Language)
- HTTP - hypertextový přenosový protokol (HyperText Transfer Protokol)
- HTTPS - šifrované HTTP (HyperText Transfer Protokol - Secured)
- JSP - platforma pro tvorbu dynamických webových stránek (Java Server Pages)
- MD5 - jednocestný šifrovací algoritmus (Message Digest)
- OOP - objektově orientované programování (Object oriented programming)
- PHP - rekurzivní zkratka pro skriptovací platformu (PHP: Hypertext Preprocessor)
- PNG - grafický formát dat (Portable Network Graphics)
- SI - softwarové inženýrství
- SSJS - technologie umožňující spuštění JavaScriptu na serveru (Server Side JavaScript)
- SSL - šifrovaná vrstva pro komunikaci (Secure Socket Layer)
- SQL - strukturovaný dotazovací jazyk (Structured Query Language)
- TG - Turbogenerátor
- URL - adresa umístění zdrojů dle definované struktury (Uniform Resource Locator)
- VBScript - programovací jazyk (Visual Basic Script)
- WWW - služba internetových stránek (World Wide Web)
- XHTML - rozšiřitelné HTML (eXtensible HyperText Markup Language)
- XML - rozšiřitelný značkovací jazyk (eXtensible Markup Language)
- XP - extrémní programování (eXtreme Programming)

Obsah

1. Úvod	7
1.1. Historie	7
1.2. Zadání	7
1.3. Termín zadání a vypracování	8
2. Volba implementačních nástrojů	8
2.1. Výkonná část aplikace	8
2.1.1. <i>Vlastnosti jazyka PHP</i>	8
2.1.2. <i>Alternativy</i>	9
2.1.3. <i>Vlastnosti skriptovacího jazyka ASP</i>	9
2.1.4. <i>Zvolené řešení</i>	9
2.2. Uživatelské rozhraní (XHTML)	9
2.2.1. <i>Kaskádové styly (CSS)</i>	10
2.2.2. <i>Rozšíření uživatelského rozhraní (JavaScript)</i>	10
2.2.3. <i>Rozšíření uživatelského rozšíření (jQuery)</i>	10
2.2.4. <i>Komunikační rozšíření uživatelského rozhraní (AJAX)</i>	10
2.3. Databázová vrstva (Firebird SQL)	11
2.3.1. <i>Alternativy</i>	11
2.3.2. <i>Zvolené řešení</i>	12
2.4. Aplikační server	12
2.4.1. <i>Alternativy</i>	12
2.4.2. <i>Zvolené řešení</i>	12
2.5. Kódování znakové sady	13
2.5.1. <i>Alternativy</i>	13
2.5.2. <i>Zvolené řešení</i>	13
3. Použité implementační nástroje	13
3.1. Datová vrstva	14
3.2. Aplikační vrstva	14
3.3. Webová vrstva	15
4. Struktura práce	15
4.1. Výběr metody	15
4.1.1. <i>Klasický model tvorby softwaru</i>	16
4.1.2. <i>Spirálový model</i>	16

4.1.3.	<i>Agilní metody - extrémní programování (XP)</i>	17
4.1.4.	<i>Naše volba</i>	18
4.1.5.	<i>Manifest agilního vývoje softwaru</i>	18
4.2.	<i>Návrh</i>	19
4.3.	<i>Implementace</i>	19
4.4.	<i>Testování</i>	20
5.	<i>Existující implementace</i>	21
6.	<i>Popis implementace</i>	22
6.1.	<i>Databázová vrstva</i>	22
6.1.1.	<i>Tabulky</i>	23
6.1.2.	<i>Indexy, primární a cizí klíče, generátory</i>	25
6.1.3.	<i>Triggery</i>	26
6.1.4.	<i>Procedury</i>	26
6.1.5.	<i>Domény a výjimky</i>	26
6.2.	<i>Aplikační serverová část - PHP</i>	27
6.2.1.	<i>DB komunikační knihovna</i>	28
6.2.2.	<i>Hlavní skript</i>	29
6.2.3.	<i>Jednotlivé "akce"</i>	30
6.2.4.	<i>Pomocné skripty</i>	30
6.2.5.	<i>Knihovny funkcí</i>	31
6.3.	<i>Klientská část aplikace (GUI)</i>	36
6.3.1.	<i>Statická část (HTML)</i>	36
6.3.2.	<i>Dynamická část (Javascript)</i>	37
6.3.3.	<i>Kaskádové styly (CSS)</i>	37
7.	<i>Testování aplikace</i>	38
7.1.	<i>Testování pomocí hypotéz</i>	38
7.1.1.	<i>Hypotézy</i>	38
7.1.2.	<i>Cílová skupina</i>	39
7.1.3.	<i>Nastavení testu</i>	39
7.1.4.	<i>Pre-test</i>	40
7.1.5.	<i>Testovací úkoly</i>	40
7.1.6.	<i>Post-test</i>	41
7.1.7.	<i>Vyhodnocení účastníků</i>	41

7.1.8.	<i>Vyhodnocení hypotéz</i>	42
7.1.9.	<i>Závěr hypotéz</i>	42
7.2.	Testování uživatelského rozhraní	42
7.2.1.	<i>Akceptační testy</i>	43
7.3.	Statistické funkce	43
7.4.	Testování bezpečnosti aplikace	44
7.4.1.	<i>Proniknutí do systému bez přihlášení</i>	44
7.4.2.	<i>Odposlech spojení mezi klientem a serverem</i>	44
7.4.3.	<i>Útok pomocí dotazů (SQL injection)</i>	45
8.	Uživatelský návod	46
8.1.	Přehled použitých ikon	46
8.2.	Navigace v programu	46
8.3.	Funkce prohlížeče	48
8.4.	Tisk	48
8.5.	Chybová hlášení	49
8.6.	Hlavní nabídka	49
8.6.1.	<i>Data</i>	49
8.6.2.	<i>Výběr</i>	53
8.6.3.	<i>Statistiky</i>	54
8.6.4.	<i>Technické informace</i>	64
8.6.5.	<i>Další funkce</i>	65
8.6.6.	<i>Odhlášení</i>	67
9.	Zhodnocení funkčnosti aplikace	68
10.	Závěr a zhodnocení práce	68
11.	Použité zdroje	69

1. Úvod

Téma mé diplomové práce vzniklo z dlouhodobější spolupráce mezi firmou Computer Systems Consulting s.r.o., se kterou několik let spolupracuji, a společností ŠKODA POWER a.s. Spolupráce je založena na statistických analýzách a tvorbě programového vybavení pro evidenci a analýzu poruch parních turbogenerátorů.

1.1. Historie

Údaje o poruchách parních turbogenerátorů jsou ve ŠKODA POWER a.s. shromažďovány již od roku 1972. Papírové záznamy nejprve byly od roku 1988 převedeny postupně do databáze dBase IV. Od roku 1992 byla pro evidenci těchto záznamů vytvořena aplikace v relačním databázovém systému PC FAND. Tato aplikace byla provozována v prostředí MS DOS do roku 2003, kdy jsem byl pověřen tvorbou nové aplikace v systému Borland Delphi pro MS Windows. Tato aplikace byla vytvořena ve dvou variantách – pro správce databáze a v jednodušší verzi pro ostatní uživatele. Nicméně, její použití bylo možné pouze ve vnitřní síti ŠKODA ENERGO (dnes ŠKODA POWER a.s.).

1.2. Zadání

Vzhledem k současným potřebám a možnostem sběru dat bylo třeba vytvořit snadno přenositelnou aplikaci, nezávislou na operačním systému (MS Windows, Unix, Linux), která by byla dostupná i vzdáleným uživatelům prostřednictvím internetu. Aplikace musí splňovat přísná kritéria IT bezpečnosti v rámci ŠKODA Holding. Bylo třeba přepracovat strukturu databáze podle současných požadavků (řada údajů už není dostupná, některé údaje jsou nevěrohodné a nepoužívají se, některé údaje jsou naopak nově sledovány), přepracovat a doplnit číselníky. Statistické funkce programu bylo třeba rozšířit o možnosti vizualizace dat, odhady parametrů Weibullova modelu pro doby mezi poruchami a délek poruch. V souvislosti s tím bylo nutné vytvořit nástroj pro vyhledávání a tvorbu složených podmínek pro výběr dat.

Dle požadavků bylo tedy třeba provést zevrubnou revizi databázového úložiště a přepracovat strukturu databáze podle současných kritérií včetně přepracování a doplnění číselníků. Bylo zjištěno, že některé položky v evidenci jsou nepotřebné a naopak jiné chybí. Také byly zjištěny nedostatky v konzistenci dat, takže je třeba provést i úpravy struktury dat.

Dále bylo třeba přepracovat uživatelské rozhraní, které je neobratné a pomalé. Nesplňovalo aktuální funkční nároky. Bylo tedy rozhodnuto že se vytvoří webová aplikace pro evidenci poruch parních turbogenerátorů (TG). Systém by navíc měl dovolovat vzdálený přístup k datům pro klienty společnosti ŠKODA POWER a.s. (např. uživatele z elektráren, servisního střediska) při zachování bezpečnosti a integrity dat. Systém bude provozován na zařízení společnosti zadavatele a bude pod jeho správou. Samozřejmě musí systém dovolovat různé úrovně přístupu (práv uživatelů).

Nová aplikace musí sloužit také jako nástroj pro základní statistickou analýzu údajů o poruchách a vyhodnocení modelů dob mezi poruchami a délek poruch včetně grafických výstupů. V této části zadání jsem nebyl pověřen výběrem statistických metod, pouze jejich realizací a grafickým zpracováním.

Veškeré aplikační výstupy by měly být exportovatelné do jiných programových systémů, například k další statistické analýze, nebo pro tvorbu výročních zpráv a jiných dokumentů.

1.3. Termín zadání a vypracování

Vzhledem k následným modifikacím zadání bylo při prvním návrhu modifikace aplikace (květen 2007) rozhodnuto, že celý projekt bude řešen po jednotlivých etapách. V prvních se jednalo převážně pouze o evidenci dat včetně její editace. Do dalších byly postupně zařazovány filtrace dat a editace číselníků. V posledních fázích probíhalo řešení statistických výpočtů, které se byly časem ještě modifikovány. Závěrečné odevzdání proběhlo 1.9.2008.

2. Volba implementačních nástrojů

Vzhledem k dlouhodobé spolupráci jsme se účastnili i při tvorbě zadání. Bylo tedy i na nás zvolit implementační nástroje. V následující kapitole jsou objasněny důvody, které nás k této volbě vedly.

Hlavními kritérii bylo zrychlení aplikace vůči předchozí Win32 API verzi a zvýšení dostupnosti v rámci lokální sítě Škoda Power. Služba WWW tato kritéria splňuje. Rychlost je zvýšena jak díky tomu, že výkon je přesunut ze strany klienta na server, tak díky výkonnosti samotného skriptovacího jazyka PHP. Navíc nám tato služba zvyšuje dostupnost v podstatě na 100% a to bez nutnosti instalací na klientské stanice. Tato forma je také výhodná pro upravování aplikace za chodu.

2.1. Výkonná část aplikace

Další fází bylo vybrat nástroj, který by generování aplikačních stránek dovozoval. Hypertextový preprocesor PHP je dnes velmi rozšířený skriptovací jazyk pro implementaci serverové části webových aplikací. Skript napsaný v PHP vygeneruje stránku podle zadaných kritérií a výsledek odešle dotazujícímu počítači stejným způsobem, jakým se odesílají běžné statické stránky. Jedná se o objektově orientovaný programovací jazyk, který svojí syntaxí vychází z jazyka C++.

2.1.1. Vlastnosti jazyka PHP

Díky novým vlastnostem této verze se PHP posunulo o blíže objektově orientovaným programovacím jazykům. Tato nová generace PHP přinesla mnoho zásadních změn především co se týče OOP (Object oriented programming, objektově orientované programování). Objektově orientovaná syntaxe byla do PHP zabudována už od verze PHP3. Jednalo se ale pouze o vylepšení původního procedurálního základu. Stejně tak, ani PHP4 ještě nepřineslo žádné radikální vylepšení objektové syntaxe. Zásadní změny v objektovém přístupu přineslo PHP5, kdy byl přepřeprogramován celý objektový model od základu a vývojářům tak může nabídnout následující možnosti v OOP:

- konstruktory a destruktory
- specifikátory přístupu pro vlastnosti a metody (private, protected, public)
- abstraktní třídy

- konečné vlastnosti a metody (final)
- rozhraní (interface)
- specifikace typů objektů v parametrech (Class type Hints)
- statické vlastnosti a metody

2.1.2. Alternativy

Alternativními technologiemi jsou například JSP (Java Server Pages, technologie poskytující jednoduchý a rychlý způsob vytváření dynamického obsahu webových stránek, které jsou nezávislé na platformě), ASP (Active Server Pages, technologie určená zejména pro internetové stránky. Byla vyvíjena společností Microsoft, která je později nahradila systémem ASP.NET), ASP.NET, VB.NET, SSJS (Server Side JavaScript, technologie umožňující provedení JavaScriptu na straně serveru), CGI- skriptovací jazyky (Common Gateway Interface, technologie generující HTML stránky), atd.

2.1.3. Vlastnosti skriptovacího jazyka ASP

ASP je technologie vyvíjená firmou Microsoft. Základem ASP je skriptovací jazyk, přičemž není striktně určený který. Nejvíce používaný je např. VBScript (Visual Basic Script) nebo Script (implementace jazyka JavaScript od firmy Microsoft). Jedná se o technologii určenou zejména pro vývoj internetových stránek, kdy ASP soubory jsou de facto XHTML stránky obohacené o ASP skripty. Při požadavku na ASP soubor pomocí webového prohlížeče server nejdříve kód ASP interpretuje a výsledný XHTML kód odešle prohlížeči. Nevýhodou ASP je vázanost pouze na webový server IIS (Internet Information Server) firmy Microsoft, který ASP podporuje, a tedy na platformu Windows. Jako další nevýhodu shledávám skutečnost, že ASP není k dispozici zdarma.

2.1.4. Zvolené řešení

PHP shledávám jako nejvýhodnější programovací jazyk pro tvorbu aplikační logiky implementované webové aplikace především díky následujícím vlastnostem.

- nezávislost na platformě
- bezproblémová spolupráce s mnoha databázemi (Firebird, MySQL, Oracle, PostgreSQL)
- dobrá uživatelská podpora
- Open-source

2.2. Uživatelské rozhraní (XHTML)

XHTML (extensible hypertext markup language) je značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW vyvinutý konsorciem W3C. Jedná se o XML aplikaci, která je následníkem jazyka HTML. HTML (HyperText Markup Language) je dodnes (spolu s XML a následníkem HTML - XHTML) ústřední technologií webu. Poslední specifikovanou verzí je HTML 4.01., která je stále zpětně kompatibilní s původním HTML. V současné době je vyvíjeno HTML 5.0, na kterém se pracuje paralelně s XHTML 2.0.

2.2.1. Kaskádové styly (CSS)

CSS (Cascading Sytle Sheets) neboli kaskádové styly se využívají pro definování zobrazení stránek. Jedná se o souhrn metod pro úpravu vzhledu stránek, tedy formátování obsahu XHTML dokumentů. Styly umožňují přesně definovat, jak bude ten který element vypadat. Hlavním důvodem použití CSS je oddělení vzhledu stránek od jejich struktury. Tato definice vzhledu a formátování je uložena v externím souboru .css, čímž lze jednoduše zajistit jednotný vzhled elementu v rámci celého webového projektu. Při změně vzhledu webu pak stačí pouze upravit tento soubor. První návrh normy byl zveřejněn v roce 1994, nyní se pracuje na verzi CSS 3.

2.2.2. Rozšíření uživatelského rozhraní (JavaScript)

JavaScript je multiplatformní, user-side, interpretovaný, objektově orientovaný skriptovací jazyk, který lze vkládat do přímo do HTML kódu stránky. Tato technologie vznikla v roce 1996 a je vyvíjena firmou Netscape. Kód JavaScriptu může být zapsán přímo do XHTML nebo může být načítán z externího souboru .js. Díky tomu, že vykonávání skriptu probíhá na straně klienta je možné JavaScript využít k přesunu vykonávání skript ze serveru do webového prohlížeče a tím zrychlit odezvu např. jednoduchých akcí implementovaného systému. Dále se hojně využívá ke kontrole zadávaných dat do formulářů. Vzhledem k vykonávání skriptu na straně klienta není z důvodu bezpečnosti, u standardních nepodepisovaných skriptů, možné přistupovat k souborům ani k systémovým objektům. Další nevýhodou je, že uživatel může JavaScript zakázat. Důležitost a výhodnost použití JavaScriptu není jen z důvodu vylepšení interaktivnosti webové aplikace, je navíc vedle ActionScriptu jednou ze dvou technologií pomocí níž se realizuje asynchronní komunikace se serverem při používání AJAX.

2.2.3. Rozšíření uživatelského rozšíření (jQuery)

Framework jQuery je stručný, přehledný, ale přesto mocný nástroj na práci s Javascriptem. Umožní snadno vyhledávat elementy DOMu a následně je modifikovat, či vytvářet nové. K vyhledávání vám postačí znát ID (HTML atribut id="") nebo CLASS (HTML atribut class=""). Stejně tak vytváření nových HTML elementů je jednoduché, stačí zadat HTML kód a jQuery sám vytvoří patřičnou strukturu DOMu. Tento framework umí pracovat s událostmi, nabízí pokročilé funkce pro práci s poli, podpora AJAXu a animací.

Framework jQuery lze i rozšiřovat o další funkce. Těmto rozšířením se věnuje např. server <http://www.jqueryui.com>.

2.2.4. Komunikační rozšíření uživatelského rozhraní (AJAX)

AJAX není technologie. Jedná se o několik spolupracujících existujících technologií, každá zajišťuje to, co umí nejlépe a dohromady jdou novým výkonným směrem.

Hlavní předností technologie AJAX (Asynchronous JavaScript and XML) je umožnění, aby stránka zadala požadavek serveru a obdržela od něj XML data, analyzovala je a zobrazila v požadovaném formátu a to vše by se odehrávalo asynchronně. Asynchronně znamená v tomto smyslu oboustranná komunikace se serverem aniž by se musela celá stránka znovu

načíst (reload), jako je tomu při volání SQL dotazů pomocí PHP. Základním kamenem této asynchronní komunikace se serverem je JavaScript. Díky této vlastnosti lze velice zpříjemnit a zrychlit práci uživatele s aplikací.

Často se pomocí AJAXu realizují tzv. našeptávače. Když uživatel píše do textového políčka, váš text průběžně odesílají na server a dostávají od něj nápovědná slova, která by mohl chtít napsat. Ty se poté předkládají ve formě seznamu pod políčkem. Dalším využitím AJAXu je např. stránka, která umožňuje interaktivní zadávání požadavků uživatelem, kdy uživatel zmáčkne tlačítko a na stránce se po malé prodlevě objeví požadovaná informace z databáze na serveru, aniž by se znovu nahrávala celá stránka. Zobrazování výsledků dotazů databáze bez nutnosti opětovného načtení celé stránky se dá využít v mnoha dalších případech jak při práci s formuláři, s menu nebo obsahem stránky a zpříjemnit tak uživatelské rozhraní aplikace.

Jednou z největších výhod AJAXu je také šetření datovými přenosy. U klasické webové aplikace se s každým požadavkem musí uživateli posílat celý kód stránky, ačkoli se změní třeba jen nepatrná část stránky. Naopak s AJAXem se posílá jenom to co se mění. Hlavní výhodou AJAXu je bezesporu urychlení práce uživatele při používání aplikace. Nemusí se po každém úkonu znovu nahrávat nová stránka. Toto chování je daleko blíže tomu, co zná uživatel z klasických desktopových aplikací, což zvyšuje použitelnost aplikace z hlediska look & feel (standardizace jednotného vzhledu i funkčnosti).

2.3. Databázová vrstva (Firebird SQL)

Zvolení databázového serveru bylo možné, ovšem stávající systém bežel na aplikaci Borland Interbase, takže jsme tuto volbu ponechali, pouze s přechodem na OpenSource ve SQL Firebird. Tento SQL server vykazuje profesionální kvality i pro objemné databáze, je přenositelný na libovolnou platformu a je snadno ovladatelný.

2.3.1. Alternativy

Oracle (Oracle9i), Sybase (Sybase ASE), Microsoft (MS SQL, MS Access), IBM (DB2), Derby (Caudscape), Progress, PostgreSQL, LDAP, Informix, Solid, Firebird (Interbase).

2.3.1.1. Vlastnosti databázového serveru PostgreSQL

PostgreSQL je robustní Open-source relační databázový systém, který je produktem patnáctiletého aktivního vývoje. Jedná se o bezpečný a spolehlivý databázový systém, který není závislý na konkrétní platformě. PostgreSQL je šířen pod nejliberálnější Open-source licencí BSD2, díky níž může být neomezeně používán, distribuován a modifikován.

2.3.1.2. Vlastnosti databázového serveru Oracle

Oracle je v současné době jedním z nejpropracovanějších relačních databázových systémů. Podstatnou nevýhodou tohoto systému je jeho komerční politika a vysoká cena.

2.3.1.3. Vlastnosti databázového serveru Microsoft SQL Server

Microsoft SQL Server je výkonný databázový systém vyvíjený společností Microsoft. Zásadním omezením je jeho závislost na platformě Microsoft Windows. Další podstatnou nevýhodou je jeho cena. Existuje bezplatně distribuovatelná verze systému MS SQL Serveru, jejíž výkon je však zaručen pouze pro maximálně 5 současných uživatelských přístupů. Tato verze je vydána pod názvem MSDE (Microsoft Data Engine) a je využitelná v menších projektech.

2.3.2. Zvolené řešení

Z nabízených možností jsme vybrali systém Firebird z kromě výše uvedených důvodů i proto, že nabízí:

- podpora přístupu z PHP
- stabilita
- rychlost
- podpora Foreign Keys, Procedur, Triggerů
- je standardně zdarma
- dobrá uživatelská podpora
- nezávislost na platformě

2.4. Aplikační server

Apache je softwarový webový server šířený pod open-source licencí. Je použitelný na množství moderních operačních systémů – Unix, Microsoft Windows, Linux. Jeho vývoj začal v roce 1993 v NCSA (National Center for SuperComputing Applications) na Illinoiské univerzitě. V dnešní době je Apache nejpopulárnějším serverem na internetu. Dle průzkumů firmy Netcraft dosáhla používanost serveru Apache v listopadu 2005 69% ze všech webových serverů.

2.4.1. Alternativy

Z na trhu dostupných aplikací bych rád zmínil: PWS (Personal Web Server - volně dostupná verze IIS od společnosti Microsoft), IBM (WebSphere), SUN (Sun ONE), Microsoft (MS IIS (Internet Information Server)), Macromedia (JRun) a Novell (eXtend App server).

2.4.1.1. Vlastnosti IIS

IIS (Internet Information Services) je sada internetových služeb pro servery běžících na systému Microsoft Windows. Dle statistik se jedná o druhý nejpoužívanější webový server Internetu (prvním je Apache). IIS je komerčním produktem, který je distribuován pouze spolu s operačním systémem Microsoft Windows Server 2003.

2.4.2. Zvolené řešení

Pro naši aplikaci jsme vybrali volně dostupný server Apache HTTP mimo jiné proto, že je podporován většinou webových serverů. Nabízí podporu PHP a Firebird SQL. Je nabízen jako Open-source software. Mimo to také nabízí velmi dobré zabezpečení.

2.5. Kódování znakové sady

Pro korektní zobrazování českých znaků ve webovém prostředí a bezproblémové ukládání a čtení těchto znaků z databáze je třeba zvolit vhodné kódování. A to jak pro zdrojové kódy, tak pro databázi.

2.5.1. Alternativy

Znaková sada definuje množinu znaků, které se mohou vyskytovat v textu zobrazovaném na počítači. Každý znak je až do samotného zobrazení reprezentován číselným kódem, který je definován právě ve znakové sadě. Znakových sad je velké množství a jednou z nejznámějších je ASCII, která definuje 128 znaků. Zahrnuje tak všechny znaky anglické abecedy, číslice, interpunkci a speciální znaky. Pro zobrazení české diakritiky bylo však nutné tuto znakovou sadu rozšířit na 256 znaků, kde prvních 128 znaků je stejných s původní ASCII a zbylé znaky jsou použity pro národní znakové sady. Znakových sad pro českou diakritiku vzniklo několik, přičemž v dnešní době nejpoužívanější jsou ISO 8859-2 a Windows 125x.

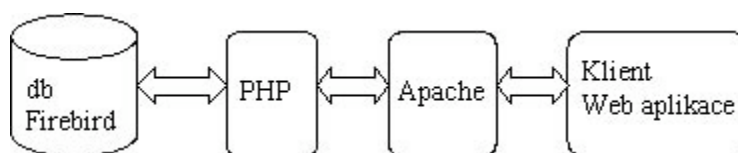
2.5.2. Zvolené řešení

Kódování UTF-8 je pro kódování znakové sady Unicode v prostředí Internetu v dnešní době nejpoužívanější. To také díky tomu, že kódování UTF-8 je podporované všemi známými webovými prohlížeči. UTF-8 kóduje znaky do 1 až 4 Byt. Znaky s kódem nižším než 128 kóduje jako jeden Byte dle ASCII. Znaky vyššími kódy jsou reprezentovány více Byty.

3. Použité implementační nástroje

Všechny softwarové prostředky, potřebné pro provoz systému jsou vyvíjeny v rámci projektu Open Source, tedy jsou určeny k bezplatnému použití. Nevznikají tím uživatelům tohoto systému žádné další náklady s pořizováním databázového systému, programového prostředí či jiných programových prostředků.

Aplikace je založena na třívrstvé architektuře server – klient – tenký klient. Ke komunikaci používá jako tzv. "tenkého klienta" prostředí internetového prohlížeče.



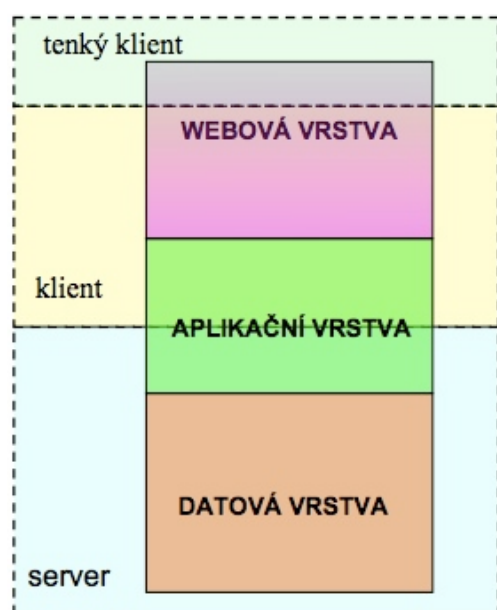
Zvolená softwarová architektura přináší několik výhod současně:

- nízké náklady na použité programové prostředky
- volně šířitelné programové prostředky v rámci licence GNU (což je rekurzivní zkratka pro GNU's Not Unix)
- minimální hardwarové nároky na klientské stanice
- jednoduchá údržba systému (je možná i na dálku)

- žádná instalace u klienta, stačí pouze připojení k intranetu/internetu
- možnost řízení uživatelských přístupů
- používání systému není omezeno místně – lze jej používat dislokovaně v různých místech, podmínkou je opět pouze připojení k síti intranet/internet

Po funkční stránce lze rozlišit tři základní vrstvy: datovou, aplikační a webovou. Aplikační vrstva je dále rozdělena na dvě samostatné vrstvy: aplikační vrstvu PHP & Apache a aplikační vrstvu PHP pro komunikaci s DB (aplikace je pak nezávislá na typu SQL DB).

Datová vrstva komunikuje s aplikační vrstvou na bázi server – klient buď prostřednictvím protokolu TCP/IP nebo v rámci nativního připojení na jednom stroji. První způsob komunikace dovoluje i fyzické oddělení obou vrstev na různých (i vzdálených) strojích.



Obr. 1: Struktura systému.

Vrstvy se vzájemně překrývají: zatímco datová vrstva je celá umístěna na serverové části systému (fyzicky je umístěna na datovém serveru), aplikační vrstva je součástí jak serveru (aplikační server), tak i klientské aplikace (webový server). V této architektuře je klientská část, obsahující zčásti jak aplikační, tak i webovou (prezentační) vrstvu, fyzicky umístěna na webovém serveru. Uživatel s tímto systémem komunikuje prostřednictvím tenkého klienta (webového prohlížeče), umístěného fyzicky na jeho pracovní stanici. Tenký klient (uživatel) nemusí mít žádný přístup k datům – ten zprostředkuje aplikační vrstva klientské části systému.

3.1. Datová vrstva

Základem systému je databázový server Firebird. Tento server je k dispozici například na adrese <http://www.firebirdsql.org/> v rámci sdružení Open Source. Jedná se o moderní databázový systém, který využívá jazyk SQL (dle normy ISO/IEC 9075:1992 - Database Language SQL (SQL-92)) pro definici dat a manipulaci s nimi.

Databáze je implementována pomocí jazyka SQL a DB serveru Firebird SQL (verze 2.0). Tato aplikace je vyvíjena skupinou Firebird Project a je odvozena z aplikace Interbase od společnosti Borland. Jak ovšem bylo výše řečeno, program je třívrstvý, takže provést změnu systému lze pouze upravením knihovny funkcí pro přístup k DB.

3.2. Aplikační vrstva

Aplikační vrstva je tvořena webovým serverem a skriptovacím systémem PHP (verze 5.1.0), který je v dnešní době jedním z nejrozšířenějších serverových skriptovacích jazyků, doplněná funkcemi DB serveru (uživatelské funkce a trigger).

3.3. Webová vrstva

Webová aplikace generuje dynamicky sérii webových stránek ve standardním formátu HTML (dle standardu konsorcia W3C - XHTML 1.0), který je podporován běžnými prohlížeči. Obecně je každá jednotlivá webová stránka dodána prohlížeči jako statický dokument, nicméně díky reakci na vstup uživatele do formulářových prvků a následném zpracování pomocí Javascriptu, popř. pomocí knihovny Ajax, se stránka stává dynamickou. Webový prohlížeč v průběhu běhu aplikace interpretuje a zobrazuje stránky a funguje jako *univerzální klient* pro libovolnou webovou aplikaci.

Webová vrstva je vrstvou prezentační. Je zodpovědná za komunikaci aplikace s uživatelem. Poskytuje výstupy ve formátu HTML generovaném aplikační vrstvou. Webová vrstva má dále funkci autorizační – zasílá aplikační vrstvě uživatelský přístup. Webová vrstva aplikace je navržena co nejužší, to je kompenzováno pomocí dynamicky dočítaných informací (pomocí knihoven AJAX). Obsahuje navíc ovládací logiku aplikace.

Zobrazení prezentační vrstvy by nemělo záviset na použitém webovém prohlížeči, ale bohužel i v dnešní době existuje mnoho odlišností ve zpracovávání HTML standardu. Aby byla zajištěna co možná nejspolehlivější funkce aplikace v různých prostředích (na různých prohlížečích), jsou v aplikaci použity pouze ty knihovny funkcí (např. Ajax, WindowWidget), které jsou tzv. „crossbrowser“, neboli jejich funkce je stejná na různých prohlížečích. Samozřejmostí je dnes, že generované stránky by měly splňovat požadavky mezinárodně uznávaných standardů W3C.

Vykreslování HTML stránek prohlížečem probíhá za důsledného použití kaskádových stylů (CSS). Výhodou použití stylů CSS je, že oddělíme obsah stránek od jejich vzhledu. Veškerý vzhled a formátování se načítá z jednoho souboru .css, který je společný pro celou aplikaci. To umožňuje snadnou změnu designu webu, stačí změnit pouze jeden soubor .css a změna se aplikuje na celou aplikaci. Soubor CSS se ukládá do mezipaměti prohlížeče. Pokud není změněn, načítá se pouze jednou a zobrazení stránek se tím velmi urychlí.

4. Struktura práce

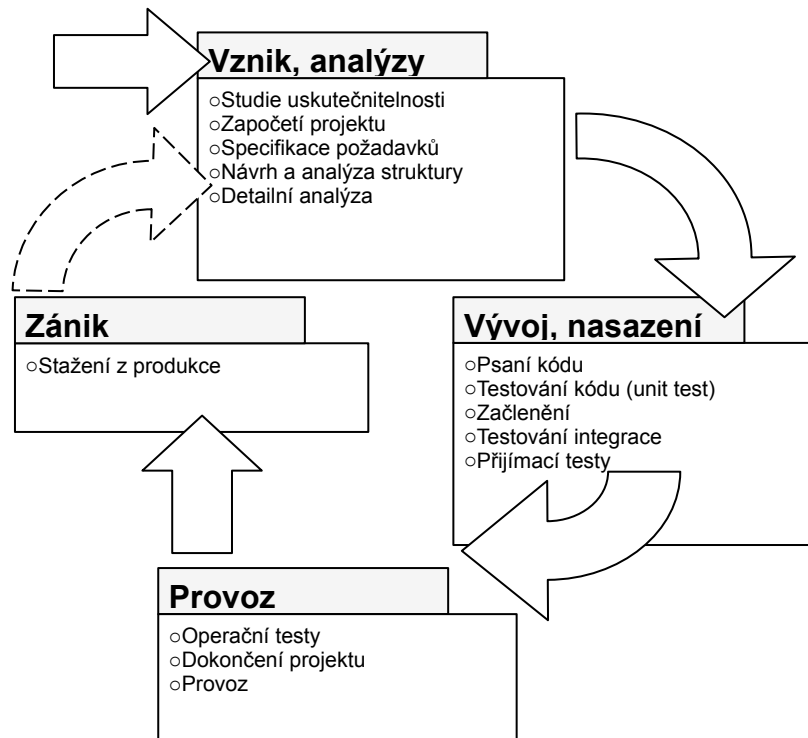
Na počátku práce bylo třeba zvolit metodu, kterou bude vývoj aplikace řízen. Nebylo tedy možno zvolit standartní model vývoje aplikace.

4.1. Výběr metody

Vývoj aplikace byl značně ovlivněn problémem, který se v softwarovém inženýrství objevoval vždy: změny a nestabilita požadavků zadavatele. Podle klasického schématu jsou na počátku vývoje stanoveny požadavky, které jsou nadále neměnné, a následně se provede kompletní analýza a implementace produktu. Než je ale vývoj dokončen, zákazník zjistí, že si některou součást systému představoval jinak. Při klasickém přístupu se tato situace často řeší zapracováním dodatečných požadavků až v následných verzích programu. Existuje však jiný, takzvaný agilní přístup, který tuto situaci řeší operativně. Tento přístup bere za neměnné čas a zdroje, v závislosti na tom se mění funkcionality produktu.

4.1.1. Klasický model tvorby softwaru

Následující diagram popisuje klasický model tvorby softwaru s důrazem na analýzu.



Standardní metoda programování

4.1.2. Spirálový model

Další z možností je spirálový model vývoje software [5]. Vývoj podle spirálového modelu probíhá po spirále a to od středu směrem ven. V tomto modelu vývoje je hlavní důraz kladen na rizika a jejich analýzu.

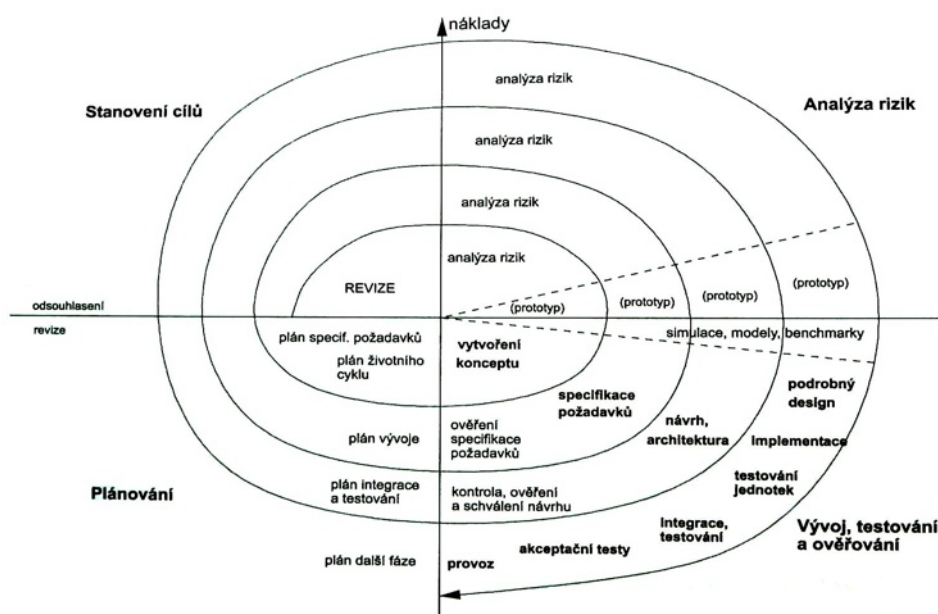


Schéma spirálového modelu životního cyklu

Mezi rizika jsou započítány všechny myslitelné situace nebo události, které mohou způsobit selhání aplikace. Úkolem je zjistit všechna tato rizika předem a připravit reakce na tyto události.

Nejprve se stanoví cíle, stanoví se také alternativy řešení a omezující podmínky (levý horní kvadrant). Pak se analyzují jednotlivá rizika a výsledkem je dostatečně funkční a robustní prototyp, který napomůže vyhodnotit alternativy anebo analyzovat velikost rizik. Pak se prochází vývojem testováním a ověřováním a nakonec plánováním.

Jednotlivé cykly mají své definované významy:

- První cyklus – najít globální rizika
- Druhý cyklus – konstrukce a ověření specifikace požadavků
- Třetí cyklus – vytvoření a ověření detailního designu
- Čtvrtý cyklus – implementace, testování a integrace

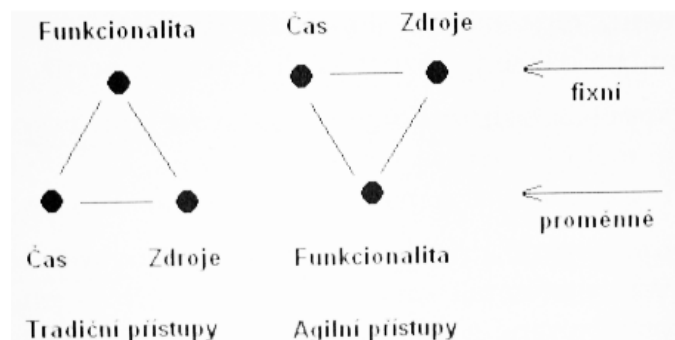
4.1.3. Agilní metody - extrémní programování (XP)

Extrémní programování [17] je ve skutečnosti promyšlený a disciplinovaný přístup k vývoji softwaru. Tato metoda se osvědčila i ve společnostech bedlivě sledujících své náklady, jakými jsou například Bayerische Landesbank, Credit Swiss Life, DaimlerChrysler, First Union National Bank, Ford Motor Company a UBS.

XP je úspěšné, jelikož klade důraz na uspokojení zadavatele. Metodologie je navržena k dodání softwaru zadavateli tehdy, kdy ho potřebuje. XP umožňuje vývojáři spolehlivě reagovat na měnící se potřeby zadavatele i v pozdějších fázích životního cyklu. Rovněž klade důraz na týmovou spolupráci. Manažeři, zákazníci a vývojáři jsou všichni součástí týmu, jehož cílem je dodat kvalitní software. XP implementuje snadný a přesto efektivní způsob umožňující vývoj v týmu.

XP zlepšuje softwarový projekt ve čtyřech ohledech: komunikaci, jednoduchosti, zpětné vazbě a kuráži (courage). XP programátoři komunikují se svými zadavateli a kolegy programátory. Svůj návrh udržují jednoduchý a jasný. Dostávají zpětnou vazbu testováním softwaru počínaje hned od začátku vývoje. Dodávají systém zadavateli tak brzy, jak je to jen možné, a implementují navržené změny. S tímto základem mohou XP programátoři s kuráží čelit měnícím se požadavkům a technologiím.

XP je jiné. V mnohém se podobá skládance. Skládá se z mnoha malých částí. Jednotlivě tyto části nedávají žádný smysl, ale když se mezi sebou propojí, je možné spatřit celkový obraz. Toto je výrazný odklon od klasických metod vývoje softwaru, jenž vede ke změně způsobu, jakým programujeme.



Rozdíly v přístupu SI

4.1.4. Naše volba

V našem případě, kdy bylo třeba navázat na stávající systém a dodržet zpětnou kompatibilitu, jsme museli použít kombinaci všech tří uvedených přístupů. Základní analýza včetně návrhu základní struktury databáze byla již provedena v předchozích verzích. Jejich dosavadní používání si vyžádalo opakovanou revizi funkcí systému. Realizace posledního zadání a nových požadavků na rozšíření funkcionality systému už probíhalo plně v duchu agilního přístupu, který se jevil i z hlediska zavedení nových metod pro statistické vyhodnocení dat jako nejvýhodnější.

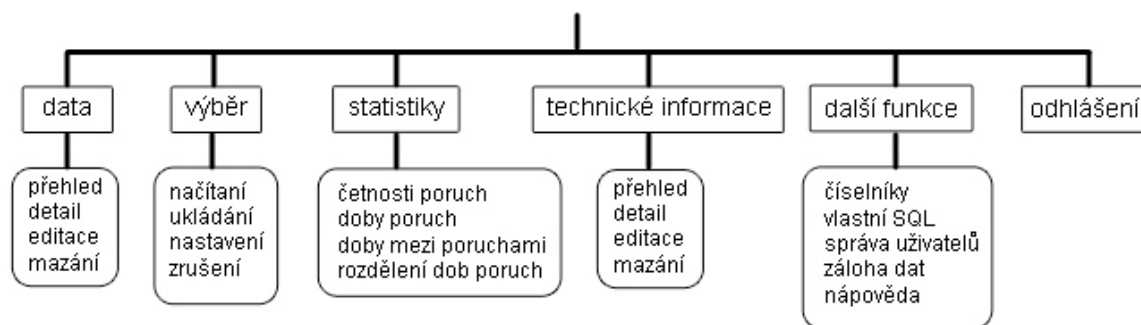
4.1.5. Manifest agilního vývoje softwaru

Přířem agilních metodik vývoje softwaru je manifest z roku 2001 [15]. Manifest říká, že přijmout a umožnit změnu je mnohem efektivnější, než pokoušet se ji zabránit. Je třeba být připraven reagovat na nepředvídatelné události, neboť zaručeně nastanou, tj. jedinou jistotou je změna. Na základě manifestu se dává přednost:

- Individualitám a komunikaci před procesy a nástroji
- Provozuschopnému softwaru před obsáhlými, objemnými dokumentacemi
- Spolupráci se zákazníkem před uzavíráním mnoha smluv
- Reakci na změnu před striktním plněním plánu

4.2. Návrh

Období návrhu je při této metodě značně zjednodušené. Na počátku návrhu aplikace jsme vycházeli z původní aplikace.



návrh struktury programu

Dále bylo třeba dbát na několik základních pravidel extrémního programování:

- *Jednoduchost*: Dokončení jednoduchého návrhu vždy zabere méně času než dokončení návrhu komplexního. Proto jsme vždy pracovali na co nejjednodušší samostatně fungující části.
- *Zvolit systémovou metaforu (system metaphor)*: Abychom udrželi konzistentní pojmenovávání tříd a metod, zvolili jsme systémovou metaforu. Pojmenování objektů je důležité pro porozumění celkového návrhu a znovupoužitelnosti kódu. Zvolili jsme si systém pojmenovávání objektů, které je srozumitelné.
- *Žádná funkcionalita není přidána dříve než je použita*: Nepřidávat do systému nic navíc. Z těchto pomocných funkcí se v budoucnu využije jen asi 10%, takže 90% času ztrácíte.
- *Přepřepínejte (refactor) kdykoliv a kdekoliv je to možné*: Myšlenku, že uchovávání a znovuužívání starých návrhů je opravdu efektivní, XP popírá. Eliminace nepoužitých funkcionalit a omlazování starých návrhů se nazývá přepřepínání (refactoring). Přepřepínání během celého životního cyklu projektu šetří čas a zvyšuje kvalitu.

4.3. Implementace

Oproti fázi návrhu je v XP na tuto fázi kladen mnohem větší důraz. Je také mnohem náročnější pro zadavatele než při standardních metodách SI. Opět je dáno několik základních pravidel XP:

- *Zadavatel je vždy k dispozici*: Jedna z mála podmínek XP je mít zadavatele vždy k dispozici. A to nejen, aby vývojářskému týmu pomáhal, nýbrž aby byl jeho součástí. Píše požadavky uživatele. Volí pořadí zapracování požadavků uživatele do distribucí, definuje „malé inkrementální distribuce“ (small incremental releases), podává podrobnosti během programování úloh. Pomoc zadavatele je třeba při funkčních testech (functional testing). Dodává testovací data a verifikuje výsledek.
- *Kód musí vyhovovat podmínkám standardu*: Kódovací standard udržuje kód konzistentní a čitelný celému týmu včetně jeho snadného způsobu přepřepínání.

- *Integrujte často*: Integruje se tak často jak to jen jde (nejlépe každých den). Každý programátor je zodpovědný za integraci svého kódu, kdykoliv se v něm vyskytne významnější změna.

- Použijte *kolektivní vlastnictví kódu (collective code ownership)*: Kolektivní vlastnictví kódu dodává programátorovi odvalu přinést nové myšlenky do kteréhokoliv segmentu projektu. Kdokoliv z vývojářů smí změnit kterýkoliv řádek kódu, opravit chyby nebo jej přepracovat. Nikdo není úzkoprofilovým prvkem při provádění změn.

- *Optimalizaci* odložte až nakonec.

- *Žádné přesčasy*: Pracování přesčas vyčerpává pracovního ducha týmu a odebírá mu motivaci. Projekty, které vyžadují přesčasy, aby se stihly termíny, budou opožděny tak jako tak. Je lepší svolat mítink pro plánování distribucí a změnit aktuální zaměření vývoje projektu či jeho časový plán. Nebo přidat další lidi do projektu.

4.4. Testování

V extrémním programování je většina testování prováděna souběžně s vývojem, resp. rychle se střídá fáze kódování a testování. Každá nalezená chyba v praxi vyžaduje napsání akceptačního testu, aby bylo možno se jí v budoucnu vyvarovat. Vytvoření akceptačního testu před laděním (debugging) pomáhá zadavateli jasně definovat problém a komunikovat jej s programátory.

Akceptační testy (acceptance tests) jsou prováděny často a jejich výsledky jsou zveřejňovány: Akceptační testy se vytvářejí na základě požadavků zadavatele. Během iterace se požadavky uživatele zvolené při mítinku pro plánování iterací přeloží do akceptačních testů. Zákazník specifikuje testovací scénáře po úspěšné implementaci požadavku uživatele. Požadavek může mít jeden nebo více akceptačních testů, všechny sledují správnou funkcionálníitu.

Akceptační testy jsou systémy černé skříňky. Každý akceptační test představuje nějaký očekávaný výsledek systému. Zadavatel je zodpovědný za ověření správnosti akceptačního testu a za určení priorit pro zpracování nalezených chyb. Akceptační testy se rovněž používají jako regresní testy před distribucí.

Požadavek uživatele se nepovažuje za dokončený, dokud neprojde akceptačním testem. Zajištění kvality (quality assurance) (QA) je základním prvkem XP procesu.

5. Existující implementace

Vzhledem k specifickému zadání byla jedinou existující aplikací, kterou jsme vyvinuli již před několika lety. Tato aplikace byla implementována v aplikaci Borland Delphi 6 a byla v podstatě předlohou k této práci.

Editor Databáze poruch parních turbogenerátorů

ŠKODA **Poruchy turbogenerátorů** **Data Statistika Editace číselníků**

Obsahuje vybrané údaje důležité pro vyhodnocení četnosti poruch parních turbogenerátorů a dob mezi nimi

Elektrárna	Blok	Výkon	Datum poruchy	Výpadek
EPOC-II	6	200 MW	5.1.1997	2645
EPOC-II	6	200 MW	6.1.1997	1046
EDE-I	4	200 MW	7.1.1997	1470
EME-III	11	500 MW	16.1.1997	3183.3
EDE-I	4	200 MW	18.1.1997	400
EDE-I	4	200 MW	25.1.1997	3200
EDE-I	4	200 MW	25.1.1997	3200
EDE-I	4	200 MW	31.1.1997	0
ETU-II	1	200 MW	1.2.1997	12319.9
EME-III	11	500 MW	3.2.1997	2600
ETU-II	3	200 MW	5.2.1997	0
EDE-I	4	200 MW	12.2.1997	3200
EPOC-I	3	200 MW	1.3.1997	7521

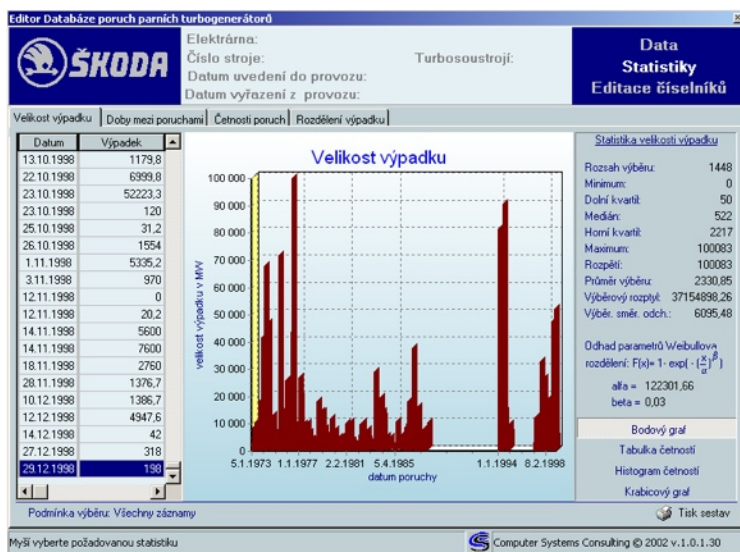
Kód	Skupina součástí	Položení	Výpadek
3570	olejová armatura	první	1046
3160	Součásti regulace	následné	1046

Elektrárna: EPOC-II
 Číslo bloku: 6
 Číslo vývodu rotoru: 0
 Nominální výkon bloku: 200 MW
 KKS kód: -0 0 0 - 0
 Datum poruchy: 6.1.1997
 Výkon před poruchou: 1
 Pravidelné zkoušky ochrany: neprováděny
 Datum poslední GO: 0
 Velikost výpadku: 1046
 Příčina poruchy: nečistota v oleji
 Projev poruchy: nízký tlak
 Kód zavinění poruchy: provozovatel
 Důvod poruchy: stav způsobený běžným prov
 Ovlivnitelnost poruchy: jinak ovlivnitelná
 Únik média: ostatní
 Regenerace: ostatní
 Napájecí: Záloha EN
 Způsob odstavení: přetížení najždění bloku
 Typ odstávky: porucha provozu úplná (odst
 Způsob opravy: vyčištění
 Kombinovaná oprava: ano - opravováno více kompo
 Příloha 1 Příloha 2 Příloha 3

Podmínka výběru: Všechny záznamy
 Výběr z hlavní nabídky programu

Computer Systems Consulting © 2002 v.1.0.1.33

Kromě evidence obsahovala již některé statistické funkce. Jejich provedení bylo ovšem příliš neobratné a neumožňovala exporty.



Původní aplikace byla vytvořena ve dvou variantách a to plnohodnotné (kde byly přístupné všechny funkce) a omezené verzi. Omezená verze byla dostupná volně všem zaměstnancům Škoda Power a.s. a neumožňovala editaci, vkládání či mazání záznamů.

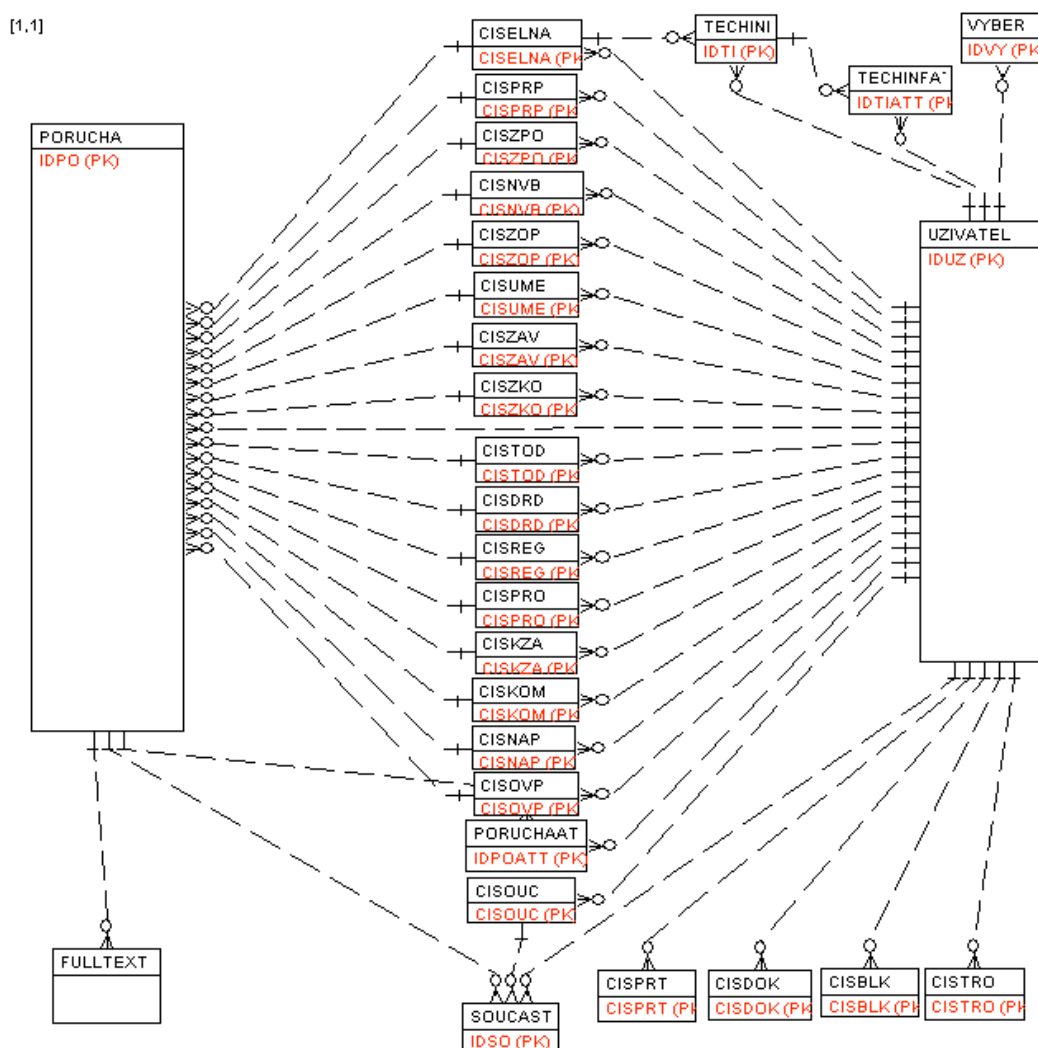
Nová aplikace pokrývá všechny funkce původní aplikace a přidává několik dalších. To vše je přístupné pro všechny zaměstnance Škoda Power a.s. dle jejich přidělených práv.

6. Popis implementace

Aplikace byla vyvinuta s důrazem na modularitu. Skládá se z několika dominantních částí. První z nich je databáze (DB) jako taková, která kromě ukládání dat má zabudovány funkce pro kontrolu a základní údržbu dat. Významná část aplikace je vytvořena v jazyce PHP. Tu lze rozdělit na vrstvu komunikující s DB, hlavní část spojující jednotlivé funkce do jednoho celku, jednotlivé dílčí funkce aplikace a podpůrnou část aplikace (login - přihlášení, null - Ajaxové funkce). Poslední částí je tzv. GUI tvořená v HTML a CSS.

6.1. Databázová vrstva

Tato vrstva (implementována pomocí SQL - Structured Query Language) obsahuje tabulky dat, indexy (pro rychlejší vyhledávání), generátory (čítače klíčů), triggerly (funkce automaticky volané při změně dat), procedury, domény (definované typy dat) a výjimky (vlastní ošetření chybových hlášení).



6.1.1. Tabulky

Hlavní funkcí databáze je uchovávání dat. To zajišťují tabulky, které obsahují neomezené množství záznamů a každý záznam omezené množství položek.

6.1.1.1. Číselníky

Tyto tabulky se používají pro pokrytí položek záznamů typu množina, tedy typů kdy máme na výběr z omezeného množství hodnot. Těmto hodnotám jsou pro zmenšení objemu dat a lepší manipulaci doplněny číselné klíče (indexy). Tyto tabulky jsou většinou ve struktuře: identifikační číslo, popis, zkratka. Seznam číselníků:

- CISBLK - bloky elektráren (nepoužívaná záloha)
- CISDOK - typ přiložených dokumentů (Word, Excel ...)
- CISDRD - druh dokumentu
- CISELNA - seznam elektráren
- CISKOM - typ poruchy na komponentě
- CISKZA - kód zavinění poruchy
- CISNAP - provoz napáječek (nepoužívaná záloha)
- CISNVB - nominální výkon bloku (nepoužívaná záloha)
- CISOUC - seznam součástí s kódem
- CISOVP - ovlivnitelnost poruchy (nepoužívaná záloha)
- CISPRO - projev poruchy
- CISPRP - příčina poruchy
- CISREG - regenerace (nepoužívaná záloha)
- CISTOD - typ odstávky stroje
- CISUME - únik média
- CISZAV - důvod poruchy (nepoužívaná záloha)
- CISZKO - pravidelné zkoušky ochrany (nepoužívaná záloha)
- CISZOP - způsob opravy
- CISZPO - způsob odstavení

6.1.1.2. Hlavní tabulky

Hlavní část databáze tvoří následující tabulky, které obsahují informace o poruchách, porouchaných součástech a o provozovaných strojích:

- PORUCHA - hlavní aplikační tabulka obsahující záznamy o poruchách turbogenerátorů. Obsahuje následující položky (v závorce je uvedeno jméno v případě odkazované hodnoty):

- IDPO - primární klíč bez jiné informační hodnoty
- POELE (CISELNA) - elektrárna
- POBLOK - číslo bloku v elektrárně
- PONVB - nominální výkon blok (odpovídá hodnotě z tabulky TECHINF, zachovává ovšem historii)
- POSLGO - datum poslední generální opravy před poruchou
- PODATUM - datum poruchy (archivace kvůli zpětné kompatibilitě se starou aplikací)
- POVYPADEK - množství ušlého výkonu v MWh
- POPRP (CISPRP) - příčina poruchy
- POPRO (CISPRO) - projev poruchy
- POZPO (CISZPO) - způsob odstavení
- POKZA (CISKZA) - kód zavinění poruchy
- POTOD (CISTOD) - typ odstávky stroje
- POUME (CISUME) - únik média
- POZOP (CISZOP) - způsob opravy
- POKOM (CISKOM) - typ poruchy na komponentě
- POKKS?? - KKS kód resp. jeho příslušná část
- PO_KKS?? - vypočítaná hodnota z položek POKKS??
- PODRUHDOK (CISDRD) - druh dokumentu
- POCAS - čas kdy porucha nastala
- PODELKA - délka výpadku v důsledku poruchy (v sekundách)
- POPOZN - poznámka k poruše
- POANOTACE - anotace k poruše
- POSTARI - celková doba provozu stroje do začátku poruchy (v sekundách)
- POSLBO - datum poslední běžné opravy
- TAG - pomocná položka
- DMR - datum a čas poslední změny záznamu
- IDUZ (UZIVATEL) - identifikace uživatele, který provedl poslední změnu
- ostatní položky slouží pouze k archivaci po předchozích aplikacích, které používali stávající DB
- SOUCASTI
 - IDSO - primární klíč bez jiné informační hodnoty
 - SOIDPO (PORUCHA) - identifikace poruchy
 - SOUCAST (CISOUC) - typ porouchané součásti
 - SONASLED - typ následku (prvotní x následovné poškození)

- SOVYPADEK - podíl na výpadku
- TAG - pomocná položka
- DMR - datum a čas poslední změny záznamu
- IDUZ (UZIVATEL) - identifikace uživatele, který provedl poslední změnu
- TECHINF - Tato tabulka byla původně určena pouze k evidenci technických informací. Následně však bylo třeba evidovat datum spuštění stroje. Tudíž byla přidána hodnota TISTART a TIVYKON, dle kterých se řídí některé statistiky.
 - IDTI - primární klíč bez jiné informační hodnoty
 - TICISLO - evidenční číslo technické informace
 - TIIDELNA (CISELNA) - elektrárna, kde je stroj instalován
 - TIBLOK - číslo bloku, kde je instalován stroj
 - TIDATUM - datum technické informace
 - TITEXT - poznámka technické informace
 - TISTART - datum uvedení stroje do provozu
 - TIVYKON - momentálně instalovaný výkon na stroji
 - TAG - pomocná položka
 - DMR - datum a čas poslední změny záznamu
 - IDUZ (UZIVATEL) - identifikace uživatele, který provedl poslední změnu

6.1.1.3. Pomocné tabulky

Další tabulky lze popsat jako pomocné. Ať už uchovávají informaci o přiložených souborech, pomocných hodnotách pro vyhledávání, nebo seznamu uživatelů.

- PORUCHAATT - odkazy na přiložené soubory (dokumentaci) k poruchám
- TECHINFATT - odkazy na přiložené soubory (dokumentaci) k provozovaným strojům
- FULLTEXT - pomocná tabulka pro předuložení vyhledávání textu v celé DB
- VYBER - pomocná tabulka, pro ukládání podmínek výběrů. Podmínky jsou uloženy v zahashovaném textu. Kromě toho se ukládá pouze název, identifikační číslo výběru a uživatele.
- UZIVATEL - seznam uživatelů systému. U každého uživatele evidujeme jeho: identifikační číslo, uživatelské jméno, jméno (a příjmení), úroveň práv, telefon a heslo (zašifrované pomocí MD5).

6.1.2. Indexy, primární a cizí klíče, generátory

Indexy jsou vytvořeny automaticky pro každé klíčované položky. Každá tabulka obsahuje primární klíč s indexem. Tabulky, které mají odkazované položky obsahují příslušné cizí

klíče, čím jednak udržují konzistenci dat a zároveň nabízejí možnost kaskádních úprav a to až při editaci klíčové položky nebo mazání celého odkazovaného záznamu.

Každý primární klíč má vytvořen příslušný generátor posloupnosti čísel. Generátor je v paměti uložen pouze jako číslo, které se funkcemi inkrementuje.

6.1.3. Triggery

Triggery (neboli spouště) jsou výborným nástrojem pro udržování konzistence dat. V této aplikaci jsou použity především pro udržování vypočítaných položek a tím urychlení dalších výpočtu nad všemi daty. Následuje stručný výpis, kde první část názvu určuje tabulku, kde je trigger uplatněn a písmena za podtržítkem kdy se trigger vykonává (After, Before, Delete, Update, Insert):

- PORUCHA_AD - úprava stáří stroje u ostatních záznamů daného stroje
- PORUCHA_AI - přepočítání stáří stroje u všech záznamů daného stroje
- PORUCHA_AU - úprava stáří stroje u ostatních záznamů daného stroje (popř. dvou při přesunu poruchy k jinému stroji)
- TECHINF_AD - vynulování stáří u všech záznamů daného stroje
- TECHINF_AI - přepočítání stáří stroje u všech záznamů daného stroje
- TECHINF_AU - úprava stáří stroje u ostatních záznamů daného stroje (popř. dvou při přesunu poruchy k jinému stroji)

6.1.4. Procedury

Procedury jsou v této aplikaci vytvořeny především pro hromadné úpravy dat. Kromě toho zde zůstali v záloze funkce ovládající některé generátory. Jsou však pouze kvůli zpětné kompatibilitě. Místo nich se používá přímo funkce systému Firebird `gen_id(ID_GEN,INC)`.

- GEN_PO (vrací ID INTEGER) - vrací a navyšuje hodnotu generátoru GEN_IDPO
- GEN_SO (vrací ID INTEGER) - vrací a navyšuje hodnotu generátoru GEN_IDSO
- GEN_TI (vrací ID INTEGER) - vrací a navyšuje hodnotu generátoru GEN_IDTI
- GEN_UZ (vrací ID INTEGER) - vrací a navyšuje hodnotu generátoru GEN_IDUZ
- GEN_POSTARI (IDELNA INTEGER, BLOK INTEGER) - pro všechny záznamy poruch dané elektrárny (IDELNA) a bloku (BLOK) přepočítá hodnotu doby provozu stroje (POSTARI)
- PREPOCTI_STARI - zavolá výše uvedenou proceduru (GEN_POSTARI) pro všechny stroje, resp. všechny existující elektrárny a bloky.

6.1.5. Domény a výjimky

Domény jsou ekvivalentem datových typů ve standardních programovacích jazycích. I zde jsou samozřejmě aplikovány. Znakové kódy jsou vytvořeny jako CHAR, tedy pevný počet znaků (doplňuje mezerami). Celočíselné hodnoty jsou buď definovány jako INTEGER, nebo s pevným počtem řádů, tedy DECIMAL(?,0), kde ? je počet řádů. Logická hodnota je

definována jako jednociferné číslo (v rozsahu 0-9). Ostatní textové položky jsou typu VARCHAR, což je znakový typ s proměnnou délkou. Záměrně není použit typ BLOB (tedy binárně uložená data), kvůli špatné ovladatelnosti, pomocí jazyka PHP.

- T_AKTDAT - DATE (default 'today', NOT NULL)
- T_DATUM - DATE
- T_CAS - TIMESTAMP
- T_IDENT - INTEGER (NOT NULL)
- T_IDENTZ - INTEGER (default 0)
- T_LOGICAL - NUMERIC(1,0) (default 0)
- T_INT? - DECIMAL(?,0) (default 0)
- T_PROCENTA - DECIMAL(10,2) (default 0)
- T_KOD? - CHAR(?)
- T_ZKRATKA - CHAR(4)
- T_SNAZEV - VARCHAR(12)
- T_NAZEV - VARCHAR(30)
- T_LNAZEV - VARCHAR(50)
- T_LINE - VARCHAR(80)
- T_MEMO - VARCHAR(1024)
- T_LMEMO - VARCHAR(8000)

Dále obsahuje aplikace pár ošetřených výjimek (exceptions - chyby) již na úrovni DB. Které se vyvolají při operacích na které uživatel buď nemá právo, nebo kvůli zachování integrity dat je nelze provést. Nicméně, aplikace má ošetřeny pokud možno všechny chyby již v PHP vrstvě, takže níže zmíněné výjimky jsou opět uplatňovány pouze ve starších aplikacích, popř. při přímé editaci databáze:

- E_NODEL 'Záznam nelze vymazat!';
- E_NOINS 'Nelze přidat nový záznam! Pouze lze upravit stávající záznam!';
- E_NOTRIG 'Trigger dosud není definován!';
- E_NOUPD 'Záznam již nelze měnit!';
- E_NOUPDO 'Tento záznam nelze měnit!';
- E_OLDDATE 'Příliš staré datum poruchy !';

6.2. Aplikační serverová část - PHP

Hlavní část programu tvoří skripty PHP (s koncovkou .php), které jsou umístěny v kořenovém adresáři aplikace:

- *index.php* - hlavní skript
- *null.php* - pomocný skript pro AJAXem volané dotazy (HTML bez hlaviček)

- *graf.php* - pomocný skript pro generování obrázků grafů
- *login.php* - přihlašovací formulář
- *ajax_login.php* - skript obsluhující přihlášení uživatele
- *csv.php* - pomocný skript pro exportování dat do formátu CSV (MS Excel)
- *rtfclass.php* - doplňující knihovna pro práci s formátem RTF
- *functions_DB.php* - knihovna pro komunikaci s databází (DB komunikační knihovna)
- *functions_form.php* - knihovna pro vytváření editačních formulářů
- *functions_get.php* - knihovna funkcí pro konverzi, či získávání jednotlivých hodnot
- *functions_global.php* - knihovna s funkcemi globálního použití
- *functions_graf.php* - knihovna s funkcemi kreslení různých typů grafů
- *functions_post.php* - knihovna zajišťující zápis dat do databáze
- *functions_preview.php* - knihovna s funkcemi hromadných tabulkových výpisů
- *functions_show.php* - knihovna poskytující funkce pro výpis jednotlivých záznamů
- *functions_stats.php* - knihovna s jednotlivými statistickými metodami
- *functions.php* - knihovna skládající předchozí knihovny s globální hlavičkou aplikace
- *ajaxroutine.js.php* - pomocná knihovna pro podporu AJAX (Javascript generovaný PHP)
- *Lilliefors.csv* - datový soubor obsahující kritické hodnoty Lillieforsova testu
- *Weibull.csv* - datový soubor obsahující kritické hodnoty pro Chi² test

Kromě nich je zde adresář “*body*” ve kterém jsou uloženy skripty jednotlivých stránek aplikace.

6.2.1. DB komunikační knihovna

Tato knihovna (*functions_DB.php*) je součástí databázové vrstvy, protože jejím prostřednictvím se provádí veškerá komunikace s DB. V praxi to má za následek to, že v případě změny databázové aplikace za jinou SQL (např. PostgreSQL, MS SQL, atd.) stačí zaměnit pouze tuto knihovnu, která má jako jediná součást PHP aplikace návaznost na konkrétní volbu DB vrstvy.

Knihovna obsahuje pouze několik základních funkcí pro komunikaci s databází a nastavovací konstanty pro DB. V podstatě se jedná pouze o jakési “aliasy” na konkrétní funkce PHP. Některé funkce mají možnost nastavení výpisu či skrytí chybového registru databáze.

- *\$DATABASE_NAME* - obsahuje adresu databáze ("localhost:/home/DB.fdb")
- *\$DATABASE_USER* - uživatelské jméno pro přístup aplikace do DB ("SYSDBA")
- *\$DATABASE_PASS* - uživatelské heslo pro přístup aplikace do DB ("masterkey")

- *function ConnectDB()* - v případě, že tato funkce nenalezne otevřené spojení do DB, naváže nové. Zároveň zajišťuje nastavení parametrů komunikace PHP s DB (formát předávání data, atd.)
- *function DisconnectDB()* - pokud existuje spojení s DB tak jej ukončí (vždy na konci celého skriptu)
- *function GetSQL(\$sql,\$quiet=0)* - provede SQL dotaz a vrátí výsledek. Přitom zajišťuje spojení s DB a nabízí možnost omezení výpisu chyb.
- *function GetRow(\$select,\$quiet=0)* - vybere záznam z výběru a přejde na další záznam. Funkce vrací pole s indexy pojmenovanými dle názvů položek výběru. Opět nabízí možnost omezení výpisu chyb.
- *function GetRowI(\$select,\$quiet=0)* - jedná se o stejnou funkci jako předchozí, pouze jednotlivé položky jsou indexovány čísly od 0.
- *function GetFieldInfo(\$select,\$field,\$quiet=0)* - vrací pole datových parametrů pro daný výběr a položku. S možností omezení výpisu chyb.
- *function GetFieldCount(\$select,\$quiet=0)* - vrací počet položek ve výběru.
- *function GetGEN(\$gen_id,\$incre=0)* - získá hodnotu generátoru *\$gen_id* (čítač pro primární klíče tabulek). Popř. nabízí možnost následovného navýšení hodnoty o *\$incre*.
- *function GetRowCount(\$table,\$where="1=1")* - vrací počet záznamů tabulky *\$table* splňující podmínku *\$where*. V případě že podmínka není dána nahradí se logickou pravdou a funkce vrací celkový počet záznamů.
- *function RunMySQL(\$sql,\$quiet=0)* - funkce vracející výpis libovolného skriptu *\$sql*. Popř. zformátovaného do přehledné tabulky. Funkce je dostupná pouze administrátorům a má možnosti omezení výpisu chyb.
- *function TestDB()* - pomocná funkce, která detekuje verzi databáze. Vrací tedy logickou hodnotu na základě které se provede či neprovede update struktury DB.
- *function ModifyOldDB()* - funkce pro úpravu struktury databáze. Tato funkce je využívána k automatické bezztrátové úpravě původní verze DB na novou.
- *function GetBlob(\$data)* - funkce pro získávání hodnot z položky *\$data* typu BLOB (pouze pro účely předchozí funkce)

6.2.2. Hlavní skript

Hlavním skriptem aplikace je soubor “*index.php*”, který obsahuje HTML kód základní obrazovky aplikace. Obsahuje hlavní i podřízená menu. Každá volba znamená změnu předávaného parametru *\$action* (popř. *\$subaction*). Tyto proměnné se vyhodnotí a zvolí se příslušný obsah stránky (výběrem odpovídajícího souboru z adresáře *body*). Na závěr skript ukončí spojení s databází. Na stránce se vždy zobrazuje jméno přihlášeného uživatele a formulář pro výběr dat (načítaný pomocí AJAXu).

6.2.3. Jednotlivé “akce”

Tyto “akce” jsou reprezentovány jednotlivými soubory příkazů uložených v adresáři *body* s koncovkou *.body*. Tyto soubory jsou vkládány do hlavního skriptu a na uživatelské úrovni je nelze zobrazovat samostatně nebo bez jiného pomocného skriptu.

- *data.body* - hlavní přehledová tabulka poruch
- *editporucha.body* - editace poruchy a její zápis do DB
- *doby_poruch.body* - výpočet statistických charakteristik vybrané podmnožiny dat dle dob trvání poruch
- *doby_mezi_poruchami.body* - výpočet statistických charakteristik vybrané podmnožiny dat dle dob mezi poruchami
- *cetnosti_poruch.body* - statistické funkce pro četnosti poruch
- *rozdeleni_dob.body* - statistické funkce pro rozdělení dob trvání poruch
- *technicke_info.body* - přehled technických informací (resp. strojů)
- *edittechinf.body* - editace technické informace a zápis do DB
- *editace_ciselniku.body* - administrátorská funkce editace číselníků
- *modifikace.body* - administrační funkce automatické úpravy struktury DB
- *mySQL.body* - administrační funkce volání vlastních SQL příkazu nad DB
- *sprava_uzivatelu.body* - administrační funkce pro správu uživatelů
- *zaloha_dat.body* - administrační nástroj pro zálohování DB
- *help.body* - on-line návod k aplikaci

6.2.4. Pomocné skripty

Pomocné skripty jsou samostatně volané objekty. Ať už se jedná o skripty generující obrázky, nebo CSV či samotné HTML, mají vždy své hlavičky a vložení potřebných knihoven.

- *null.php* - pomocný skript pro AJAXem volané dotazy (HTML bez hlaviček). Obsahuje volání jednotlivých funkcí dle předávaného parametru *\$naction*.
- *graf.php* - pomocný skript pro generování obrázků grafů. Nejdříve si načte potřebná data ze SESSION (globální pole proměnných pro dané HTTP spojení) a nadefinuje si rozměry a barvy obrázku. Následně zavolá funkci pro daný typ grafu. Data odešle na výstup ve formátu PNG.
- *login.php* - skript s přihlašovacím formulářem s hlavičkou HTML. Pomocí JQuery odesílá formulář data následujícímu skriptu, který je vyhodnotí a vrátí buď 1 (tedy přihlášen), nebo text (chybovou hlášku), která je zobrazena uživateli.
- *ajax_login.php* - skript obsluhující přihlášení uživatele. Kontroluje zda uživatel existuje v tabulce UZIVATEL v DB, popř. zda má správné heslo. V případě, že uživatel

existuje a je-li uloženo prázdné heslo, funkce uloží heslo aktuálně zadané z přihlašovacího formuláře. Hesla jsou uložena v zašifrované formě (MD5).

- *csv.php* - pomocný skript pro exportování dat z hlavní tabulky a technických informací do formátu CSV (MS Excel). Obsahuje hlavičky CSV a překódování dat do CP1250 (kódování češtiny Windows).

6.2.5. Knihovny funkcí

Nejdůležitější výkonnou součástí aplikace jsou knihovny funkcí. Tyto knihovny seskupují funkce do logických celků.

- *functions.php* - soubor, který do sebe vkládá ostatní knihovny (a to ve správném pořadí). Obsahuje aplikační HTTP hlavičky (vypnutí vyrovnávací paměti, kódování, typ dokumentu). Nastavuje PHP konfiguraci (úroveň vypisování chybových hlášení, atd.). Soubor také ověřuje přihlášení a práva uživatele. V případě, že není správně přihlášen, je aplikace přeměrována na přihlašovací skript *login.php*. Kromě toho obsahuje soubor funkce pro zápis a načítání nastavení výběrů.
- *functions_form.php* - knihovna pro vytváření editačních formulářů. Obsahuje funkce, které vrací HTML kód daného formuláře. Obsahuje následující funkce:
 - *function EditSELECT(\$table,\$id,\$scaption,\$name,\$sel_id="", \$style="")* - Univerzální funkce, která vrací HTML objekt `<select>` naplněný hodnotami z tabulky *\$table*, s identifikátorem *\$id*, položkou pro popis hodnoty *\$scaption*, jménem HTML objektu *\$name*, předvyplněnou položkou *\$sel_id* (je-li) a upraveným stylem *\$style* (je-li třeba)
 - *function EditSELECTBLOK(\$scislna,\$sel_id="")* - Podobná funkce jako předchozí ovšem pouze pro výběr bloku dané elektrárny *\$scislna*.
 - *function EditINPUT(\$name,\$value="", \$style="", \$type="text")* - Opět univerzální funkce formátující položku do HTML objektu `<input>` s daným jménem *\$name*, předvyplněnou hodnotou *\$value*, upraveným stylem *\$style* a daným typem *\$type*.
 - *function EditPORUCHA(\$IDPO)* - vrací formulář v HTML pro editaci záznamu o poruše.
 - *function GetPoruchaFileUpload(\$IDPO)* - vrací formulář pro přikládání souborů k záznamu o poruše.
 - *function EditTECHINF(\$IDTI)* - vrací formulář v HTML pro editaci záznamu o technické informaci (resp. o daném stroji a elektrárně).
 - *function GetTechinfFileUpload(\$IDTI)* - vrací formulář pro přikládání souborů k záznamu technické informace.
 - *function EditSFIRST()* - vrací formulář pro změnu hodnoty *\$sfirst*, která udává počet zobrazovaných záznamů na stránce.

- *functions_get.php* - knihovna funkcí pro konverzi, či získávání jednotlivých hodnot.
Obsahuje:
 - *function GetUserCaption()* - výpis přihlášeného uživatele
 - *function GetTableTop(\$d,\$o,\$w=0)* - generátor záhlaví HTML tabulky
 - *function GetOrderTitle(\$caption,\$orderby)* - generátor názvu záhlaví s možností řazení a jeho detekcí
 - *function DecodeWhere(\$w)* - načítání systémového pole pro globální výběr dat
 - *function GetVyberForm()* - složení výběrového formuláře z následujících funkcí.
 - *function GetVyberPO_____(\$size,\$id=0)* - výběr pro danou položku. Funkce je volaná pomocí AJAX. Ve dvou variantách: malá a velká.
 - *function GetVyberCISELNIK(\$field,\$ciselnik,\$order,\$size,\$id=0)* - předchozí funkce volají tuto univerzální pro položku *\$field*, číselník *\$ciselnik*, v pořadí *\$order*
 - *function GetVyberIDSO(\$size,\$id=0)* - výběr součástí
 - *function GetPoruchaIDSO(\$size,\$IDPO,\$id=0,\$isadded=-1)* - výběr součástí pro poruchu *\$IDPO*, nebo inverzní výběr *\$isadded*
 - *function GetSelectVYBER(\$IDVY=0)* - vrací HTML <select> pro výběry z DB
 - *function GetSelectCISELNA(\$ciselna="")* - vrací HTML <select> pro elektrárnu
 - *function GetSelectCISNVB(\$cisnvb="")* - vrací HTML <select> pro nominální výkn bloku
 - *function GetVYBER(\$id)* - načítá uložený výběr z DB
 - *function GetEditCiselnik(\$ciselnik,\$wherecis,\$edit=1)* - editační / přehledová (*\$edit*) tabulka pro číselník *\$ciselnik* a podmínku *\$wherecis*
 - *function GetFile(\$soubor)* - načtení obsahu souboru
 - *function cacheSearch(\$forced=false)* - fulltextové vyhledávání s vyrovnávací pamětí v DB s možností posíleného přepisu
- *functions_global.php* - knihovna s funkcemi globalního použití a registrací (definicí) globálních proměnných:
 - *function HSTrim(\$s)* - náhrada mezer za tzv. pevné mezery
 - *function Date2Int(\$datum)* - převod datumu na tzv. Unix Time
 - *function RegMyGlobal(\$name)* - registrace (definice) dané proměnné
 - *function Int2HourCount(\$time)* - převod sekund na hodiny

- *function Int2Hour2Count(\$time)* - převod sekund na hodiny² (tedy na druhou)
- *function stripDiakritika(\$text)* - odstraní háčky a čárky na daném textu
- *function Int2TimeCount(\$time)* - převod sekund na strukturovaný výpis ve formátu D dní HH:MM:SS
- *function Int2Time2Count(\$time)* - převod sekund na strukturovaný výpis ve formátu D dní² HH²:MM²:SS²
- *function SetWhere(\$w)* - zápis hodnot do globálního výběru dat
- *functions_graf.php* - knihovna s funkcemi kreslení různých typů grafů z hodnot načtených ze \$_SESSION, pomocných funkcí a definice šířky a výšky grafu
 - *function GetColor(\$img=false,\$kolikata,\$celkem)* - pomocná funkce pro generování barvy ze škály daného stupně.
 - *function imagelineM(\$image, \$x1, \$y1, \$x2, \$y2, \$color)* - funkce kreslení úsečky s omezením souřadnic
 - *function GrafDateCaption(\$date)* - převodní funkce ze sekund na datum
 - *function ConvertValue(\$value,\$typeK="", \$unit="")* - převede hodnotu dle daného formátu a přidá jednotky
 - *function MapHistogram(\$titles)* - vytváří mapu (HTML - <map>) pro histogram
 - *function MapKolac(\$data,\$titles)* - vytváří mapu (HTML) pro koláčový graf
 - *function MapPP(\$c,\$d,\$alfa,\$beta,\$gama,\$delta)* - vytváří mapu (HTML) pro graf podmíněné spolehlivostní funkce
 - *function DrawHistogram(\$img)* - kreslí graf histogramu
 - *function DrawHistogramExt(\$img)* - kreslí graf rozdílových histogramů
 - *function DrawKolac(\$img)* - kreslí koláčový graf
 - *function DrawKrabice(\$img)* - kreslí krabicový graf
 - *function paintPoint(\$img,\$x,\$y,\$color)* - kreslí bod do bodového grafu
 - *function DrawBodovy(\$img)* - kreslí bodový graf
 - *function DrawLinkovy(\$img)* - kreslí linkový graf (bodový s propojením bodů)
 - *function DrawLinkovyPas(\$img)* - kreslí linkový graf se směrodatnou odchylkou
 - *function DrawIPLinkovy(\$img)* - kreslí graf intenzity s teoretickými křivkami
 - *function DrawIP(\$img)* - kreslí graf intenzity
 - *function DrawHP(\$img)* - kreslí graf hustoty pravděpodobnosti

- *function DrawFN(\$img,\$rev=0)* - kreslí graf distribuční (spolehlivostní - v případě reverze) funkce
- *function DrawFND(\$img,\$rev=0)* - kreslí graf rozdílů empirické a teoretických distribučních funkcí
- *function DrawQQ()* - kreslí Q-Q graf s teoretickými přímkami
- *function DrawPP(\$img)* - kreslí graf podmíněné spolehlivostní funkce
- *functions_post.php* - knihovna zajišťující zápis dat do databáze. Hodnoty jsou předávány pomocí globálního pole `$_POST`. V případě funkcí `Post_____` se dělí funkce dle hodnoty primárního klíče (pokud je definován a je větší než 0, tak se jedná o modifikaci, jinak se jedná o vložení). U funkcí `Drop_____` je mazání realizováno změnou hodnoty primárního klíče na zápornou hodnotu. Díky tomu není třeba řešit archivaci.
 - *function PostPORUCHA()* - úprava či vložení záznamu o poruše
 - *function PostTECHINF()* - úprava či vložení záznamu o technické informaci
 - *function PostTECHINFATT()* - úprava či vložení záznamu o příloze tech. info.
 - *function PostPORUCHAATT()* - úprava či vložení záznamu o příloze k poruše
 - *function PostVYBER(\$IDVY)* - úprava či vložení záznamu o výběru
 - *function DropVYBER(\$IDVY)* - smazání záznamu o výběru
 - *function DropPORUCHA(\$IDPO)* - smazání záznamu o poruše
 - *function DropTECHINF(\$IDTI)* - smazání záznamu o technické informaci
 - *function DropTECHINFATT(\$IDTIATT)* - smazání přílohy k technické informaci
 - *function DropPORUCHAATT(\$IDPOATT)* - smazání přílohy k poruše
 - *function DropCISELNIK(\$wherecis,\$ciselnik)* - smazání záznamu z číselníku
 - *function PostCISELNIK(\$ciselnik)* - úprava či vložení záznamu z číselníku
 - *function PostSOUCAST()* - úprava či vložení záznamu o součásti
 - *function DropSOUCAST()* - smazání záznamu o součásti
- *functions_preview.php* - knihovna s funkcemi hromadných tabulkových výpisů (HTML - `<table>`) a jejich CSV verzí
 - *function PreviewPORUCHY()* - přehledová tabulka poruch (HTML)
 - *function PreviewPORUCHYcsv()* - přehledová tabulka poruch (CSV)
 - *function PreviewTECHINFcsv()* - přehledová tabulka tech. informací (CSV)

- *function PreviewTECHINF()* - přehledová tabulka tech. informací (HTML)
- *function PreviewCISELNIK(\$ciselnik)* - přehledová tabulka číselníku (HTML)
- *functions_show.php* - knihovna poskytující funkce pro výpis jednotlivých záznamů dle předávané proměnné. Funkce jsou většinou volané funkcemi knihovny AJAX.
 - *function ShowCISELNA(\$CISELNA)* - detail záznamu o elektrárně
 - *function ShowCISDRD(\$CISDRD)* - detail z číselníku druhu dokumentů
 - *function ShowCISBLK(\$CISBLK)* - detail z číselníku bloků
 - *function ShowCISNVB(\$CISNVB)* - detail z číselníku nom.výkonů bloku
 - *function ShowCISOUC(\$CISOUC)* - detail záznamu o součásti
 - *function ShowPORUCHA(\$IDPO)* - detail záznamu o poruše
- *functions_stats.php* - knihovna s jednotlivými statistickými metodami. Funkce jsou rozděleny dle hodnot, se kterými pracují. Uspořádání se tedy snaží šetřit výpočetní kapacitu. Pro kreslení grafů jsou již jednou vypočítané hodnoty uloženy do SESSION, aby nemuseli být počítány opětovně. Samotné grafy se pro urychlení načítají pomocí Javascriptu až na žádost uživatele o zobrazení grafu. Každý graf lze také zobrazit v libovolné velikosti.
 - *function GetChi2p(\$chi2,\$df)* - funkce pro získání p-hodnoty pro Chi²-odhad (pomocí dat Weibull.csv)
 - *function GetChi2pLill(\$chi2,\$pocet)* - funkce pro získání p-hodnoty pro Lillieforsův test (pomocí dat Lilliefors.csv)
 - *function GetCetnostiPoruch(\$sql)* - statistiky četnosti poruch dle daného kritéria (pomocí předávaného SQL dotazu - viz. *body/cetnosti_poruch.body*)
 - *function GetRozdeleniDelek(\$sql)* - statistiky rozdělení délek trvání poruch dle daného kritéria (pomocí předávaného SQL dotazu - viz. *body/rozdeleni_dob.body*)
 - *function GetCharakteristikyDobyVypadku(\$dolni_mez=-1,\$sirka=-1)* - vytváří statistiky pro délky trvání poruch
 - *function GetCharakteristikyDobyMeziPoruchami(\$dolni_mez=-1,\$sirka=-1)* - vytváří statistiky pro doby mezi jednotlivými poruchami
 - *function GetIntenzitaStariDobyMeziPoruchami(\$sid=0)* - počítání hodnot průměrné kumulativní funkce a intenzity pro doby mezi poruchami podle stáří stroje v době poruch
 - *function GetIntenzitaStariDobyPoruch(\$sid=0)* - počítání dat průměrné kumulativní funkce a intenzity pro trvání podle stáří stroje v době poruch

- *function GetIntenzitaDataPoruch(\$sid=0)* - počítání dat průměrné kumulativní funkce a intenzity opakování dle data poruchy
- *function GetIntenzitaDobyPoruch(\$sid=0)* - počítání dat průměrné kumulativní funkce a intenzity trvání pro doby poruch

6.3. Klientská část aplikace (GUI)

Uživatelské rozhraní bylo vyvíjeno s ohledem na co největší přehlednost a dynamiku načítání pro rozložení výpočetní kapacity serveru. K implementaci byla zvolena kombinace XHTML, Javascript a CSS. Tedy standartní kombinace pro tvorbu webových aplikací.

6.3.1. Statická část (HTML)

Hlavní součást *index.php* obsahuje strukturu a rozvržení základní stránky aplikace. Jako první v dokumentu musí být definice typu dokumentu:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

V hlavičce obsahuje odkazy na všechny javascriptové knihovny a na soubor se sadou příslušných stylů. Tělo dokumentu je tvořeno několika základními prvky:

- *WaitDlg (id)* - je objekt, který se zobrazuje před dokončením práce serveru. Což zajišťuje funkce *flush()*, která odesílá data okamžitě.
- *userdiv (id)* - obsahuje jméno uživatele.
- *menu (class)* - obsahuje hlavní nabídku a výběrový formulář, který je ovšem prázdný.
 - *searchForm (id)* - tedy formulář pro fulltextové vyhledávání
 - *submenu (class)* - obsahuje jednotlivé podřadné nabídky
 - *vyberDiv (id)* - slouží k naplnění výběrovým formulářem (pomocí AJAX funkcí)
- *main (id)* - hlavní část obrazovky, která obsahuje vždy konkrétní skript zvolené funkce
- *bottominfo (id)* - závěrečné informace o produktu a jeho základních parametrech.

V průběhu vývoje aplikace byl zjištěn nedostatek prohlížeče Internet Explorer (společnosti Microsoft), která prokázala neschopnost zobrazení objemnějších tabulek *<table>*, takže bylo třeba zvolit stránkování dat (tedy zobrazování pouze omezeného počtu řádek) a vyřazení detailních zobrazení z hlavní tabulky poruch. Tento pohled je nyní načítán pomocí funkcí AJAX.

Aplikace je testována a vyvíjena primárně pro prohlížeč Mozilla Firefox, s ohledem na funkčnost pod všemi ostatními prohlížeči s podporou Javascriptu. Optimalizováno pro rozlišení 1024x768.

6.3.2. Dynamická část (Javascript)

Tento skriptovací jazyk je díky své podpoře ve většině webových prohlížečích v podstatě jedinou možností rozšíření uživatelského rozhraní o dynamické prvky. K tomu slouží několik funkčních knihoven, které jsou vyvíjeny velkými skupinami webových programátorů. Tato aplikace obsahuje následující knihovny:

- *ajaxroutine.js.php* - pomocná knihovna pro podporu AJAX (generovaná PHP). Doplnjuje rutiny XML do všech prohlížečů, které AJAX vyžaduje.
- *fixEvents.js* - pomocná knihovna zajišťující podporu některých Javascriptových událostí
- *jquery.js* - pomocná knihovna pro podporu JQuery frameworku
- *webtoolkit.aim.js* - pomocný skript pro AJAX
- *login_form.js* - skript pro přihlašování do aplikace, který zpracovává informace z přihlašovacího formuláře a pomocí PHP skriptu ověřuje oprávnění do aplikace. Výsledek ověření je přitom schopen zobrazit (bez opětovného načítání stránky) v přehledném okně s barevným odlišením chybových hlášení. V případě oprávněného přihlášení přesměruje aplikaci na hlavní obrazovku. Celý skript pracuje s frameworkem JQuery.
- *functions.js* - knihovna Javascriptových funkcí
 - *function TabOrder(e,trgE)* - funkce doplňující HTML objekt (*e*) o možnost přechodu mezi aktivními prvky formuláře jinou klávesou než [Tab] na jiný HTML objekt (*trgE*)
 - *function run(e)* - funkce rozšiřující objekty formuláře o možnost odeslání formuláře stiskem kláves [Ctrl+Enter]
 - *function zobrazSkryj(idecko,visible)* - funkce měnící styl (parametr “display”) HTML objektu (*idecko*) na hodnotu “none” nebo “block” dle parametru funkce (*visible* - 1=block; -1=none; 0=opačná hodnota)
 - *function confirmDel(href,id)* - potvrzovací dialogové okno při mazání záznamu. Pokud uživatel potvrdí smazání záznamu (*id*) přesměruje funkce aplikaci na odkaz (*href*)

6.3.3. Kaskádové styly (CSS)

Kaskádové styly jsou výborným způsobem, jak oddělit vzhled aplikace. Lze tak jednoduše zaměnit vzhled za jiný. Adresování stylů probíhá především přes HTML atribut *id*, tak přes třídy stylů (HTML atribut *class*). Aplikace využívá dvou různých stylů:

- *style.css* - styl aplikovaný na celou aplikaci
- *print_style.css* - styl použitý při tisku

Tyto styly lze měnit, bez dalších úprav kódu. Lze tak tedy vytvořit např. vánoční styl aplikace jak to známe třeba z vyhledávače Google.

7. Testování aplikace

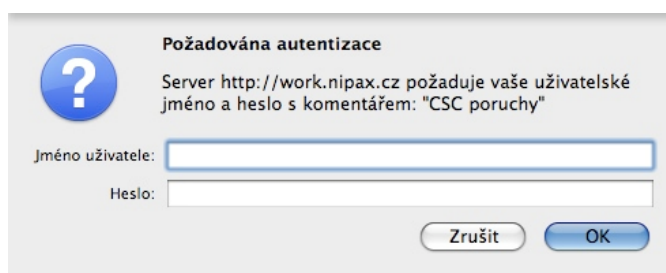
Testování probíhalo v několika fázích. Aplikace byla testována jak pomocí hypotéz (testování programátorem a několika pomocnými silami), tak s pomocí pracovníka Škody Power a.s., který průběžně testoval funkčnost aplikace. Zároveň byla aplikace testována na správnost statistických funkcí pomocí statistických programů (MiniTAB, Statistika, Matlab).

7.1. Testování pomocí hypotéz

Testování pomocí hypotéz pomáhá při rozhodování mezi několika volbami. V našem případě jsme ustanovili 4 hypotézy a zkušeli, která volba bude pro uživatele nejlepší. Tím dochází k vylepšení provozuschopnosti GUI aplikace.

7.1.1. Hypotézy

I. Přihlašování do aplikace je dostačující a nevyžaduje formulář.



II. Pro přehledovou tabulku stačí stránkování po 100 záznamech.

přehled dat

[porucha](#) | [protokol o přejímce](#) | [re](#)

zobrazit v tabulce [1605 záznamů]: << | < | vše | > | >>

typ	elektrárna	blok	výkon	čas poruchy	dobu poruchy	porouchané součásti	D, A, P, P	
PO	ECHVA	2	200 MW	01.03.2008 00:00	10h			
PO	ECHVA	2	200 MW	01.03.2008 00:00	10h			
PO	Dunamenti	17	neuvedeno	01.01.2008 10:35	12h	1130 - Přehřívák (P)		
PO	ECHVA	0	neuvedeno	01.01.2008 00:00	0h			
PO	Dunamenti	17	60 MW	01.01.2008 00:00	256h			
PO	ELE-II	2	110 MW	01.01.2008 00:00	10h			
PO	Třebovice	16	72 MW	04.05.2005 00:00	0h	3690 - Přístroje pro měření a řízení (P)		
PO	Třebovice	16	72 MW	22.04.2005 00:00	0h	7634 - Příslušenství bloku - přístrojové vybavení (P)		
PO	JETE	0	neuvedeno	20.04.2005 00:00	0h	3111 - VT rotor oběžné lopatky (P)		
PO	Třebovice	16	72 MW	18.04.2005 00:00	0h	7533 - Cirkulační čerpadla (P)		
PO	Třebovice	16	72 MW	15.04.2005 00:00	6.39h	8610 - Příslušenství elektrárny - spojovací vedení (P)		
PO	Třebovice	16	72 MW	07.04.2005 00:00	0h	3560 - Olejové hospodářství (P)		
PO	Třebovice	16	72 MW	06.04.2005 00:00	0h	3690 - Přístroje pro měření a řízení (P)		
PO	Třebovice	16	72 MW	04.04.2005 00:00	0h	7425 - Příslušenství bloku - armatury (P)		

III. Výběrový formulář je přehledný a pochopitelný.

IV. Submenu je lepší zobrazovat na kliknutí, nikoliv na přjetí kurzorem po hlavním menu.

ŠKODA POWER a.s.

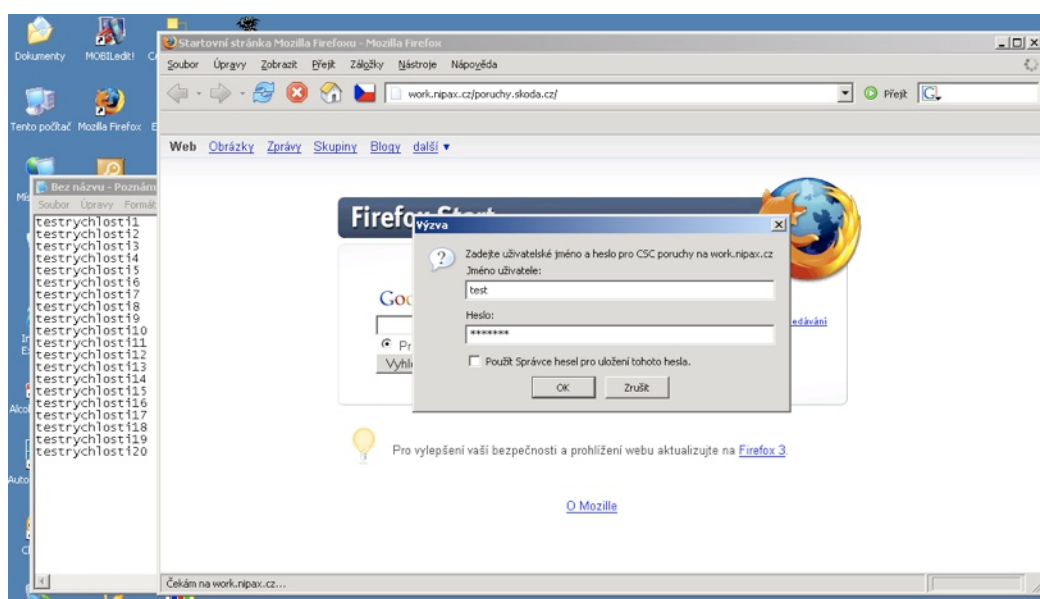
Administrátor

7.1.2. Cílová skupina

Cílovou skupinou v tomto případě jsou analytici, technici a vývojáři. Mohou to být muži i ženy ve věku od 23 do 55 let, kteří již pracovali v prostředí MS Windows a umí používat internetový prohlížeč.

7.1.3. Nastavení testu

Test probíhal jednotlivě u stolního PC (MS Windows XP) s myší a klávesnicí. Pro testování probíhalo v aplikacích Notepad a Firefox 3. Při testu byl v místnosti s testovanou osobou ještě testující, který ovšem při testu ani úkolech nepomáhal.



7.1.4. Pre-test

Tento test, by měl pomoci rozhodnout o relevanci jednotlivých testovaných uživatelů. Určit jejich schopnosti při práci s počítačem.

1. Kolik Vám je let?
2. Jak dlouho již používáte počítač? (pracovně)
3. Používáte radši myš než klávesnici?
4. Používáte klávesové zkratky?
5. Jaký používáte internetový prohlížeč?
6. Ohodnoťte Vaši rychlost psaní na klávesnici. [A-F ... jako známky A>100úhozů/min]

testovaná osoba	1.otázka	2.otázka	3.otázka	4.otázka	5.otázka	6.otázka
Adam	25let	5let	ano	ne	ie	E
Blanka	45let	15let	ne	ano	ff	B
Bedřich	27let	7let	ne	ano	ff	C
Cecílie	24let	2roky	ano	ano	ff	D
Franta	51let	16let	ano	ne	ie	E
Josef	23let	5let	ne	ano	ff	E
Martin	28let	12let	ano	ne	ie	D
Tonda	55let	25let	ne	ano	ie	B
Václav	38let	19let	ne	ano	ff	D
Zdenka	31let	9let	ano	ano	ie	D

7.1.5. Testovací úkoly

Úkoly jsou připraveny tak, aby uživatel nemusel být seznámen s testovanými hypotézami a zároveň je otestoval. Tyto úkoly musí být jednoduché a musí být znám "správný" postup jejich vykonání. V tomto případě jsou ještě předsunuty úkoly na otestování schopností testovaných.

1. Změřte čas za který stihnete napsat v otevřeném textové editoru NotePad 20x na samostatný řádek "testrychlosti" a pořadové číslo řádku. Můžete použít všech možností editoru.
2. Změřte čas, za který otevřete prohlížeč a dobu na otevření aplikace.
3. Přihlašte se jako "test", odhlašte se a znovu přihlašte jako "test2".
4. Proveďte výběr libovolné elektrárny pomocí hlavní nabídky.
5. Nalezněte 102. záznam se zapnutým stránkováním.
6. Zrušte stránkování a zkuste nalézt 98. záznam.
7. Uložte předchozí výběr jako "muj první vyber" a výběr zrušte.
8. Proveďte přejmenování "druhu dokumentu" s id=1 na "test".
9. Projděte si aplikaci dle libosti alespoň po dobu 5min.

7.1.6. Post-test

Tato část testu je zaměřena na sběr informací o průběhu vypracovávání úkolů. Uživatelé jsou dotazováni, tak abychom se dozvěděli o všech jejich problémech a úspěších při plnění úkolů.

1. Jaký čas jste si naměřili v 1.úkolů?
2. Jak dlouho Vám trvalo spuštění prohlížeče?
3. Měli jste problém s 3.úkolů? (A=bez problému ... F=nepovedlo se)
4. Obsahuje hlavní nabídku vše co potřebujete?
5. Vyhovovalo by Vám více zobrazování podmenu bez nutnosti kliknutí? (A=ano rozhodně ... F=vůbec ne)
6. Odhalili jste funkci skrytí podmenu opětovným kliknutím na hlavní nabídku?
7. Jaké stránkování by Vám vyhovovalo nejvíce? (stránkování: A=vše ...E=200, F=100)
8. Měli jste problém s vytvořením výběru? (A=bez problému ... F=nepovedlo se)
9. Měli jste problém s uložením výběru? (A=bez problému ... F=nepovedlo se)
10. Přišli Vám testovací úkoly složité? (A=jednoduché ... F=obtížné)

7.1.7. Vyhodnocení účastníků

Na základě pre-testu a úkolů vyhodnotíme testované uživatele a určíme jejich přínos pro celý test. Většinou nás budou zajímat především méně zdatní, popř. kritičtější uživatelé. Oproti tomu zkušený uživatelé, kteří se snadno vyrovnají s překážkami, jsou pro nás méně přínosní (jsou tolerantnější).

- **Adam** - 25let, pracuje s PC relativně krátce a tudíž pro nás bude představovat významnou část (kritičtější) uživatelů.
- **Blanka** - 45let, dlouholetý zaměstnanec firmy a s 15letou praxí s PC, velmi schopný uživatel. Přesto celkem zajímavý přínos k testu vzhledem k věku.
- **Bedřich** - 27let, mladší schopný zaměstnanec se zájmem o IT. Pro testování tvoří součást méně atraktivní skupiny uživatelů.
- **Cecílie** - 24let, mladý zaměstnec, bez hlubších znalostí v oblasti IT. Schopná rychle se učit. Z hlediska testování celkem přínosná.
- **Franta** - 51let, jeden z nejstarších testovaných, PC používá jen jako poslední možnost. IT příliš neholduje. Velmi kritická část testovaných.
- **Josef** - 23let, naopak nejmladší testovaný, se zájmem o IT. Pro testování tvoří součást méně přínosnou skupiny uživatelů.
- **Martin** - 28let, průměrný zaměstnanec firmy, bez většího zájmu o IT. Tvoří středně významnou část testované skupiny.
- **Tonda** - 55let, nejstarší testovaná osoba. Zvyklý pracovat v textově orientovaném prostředí. Celkem zajímavý přínos vzhledem k odlišným nárokům na ovládání.
- **Václav** - 38let, schopný starší zaměstnanec, se zájmem o IT. Méně náročný uživatel.
- **Zdenka** - 31let, průměrný zaměstnanec firmy, bez většího zájmu o IT. Tvoří středně významnou část testované skupiny.

7.1.8. Vyhodnocení hypotéz

Otázky 1,2 a 10 jsou spíše obecným testováním schopností uživatele. Nebudu je tedy zmiňovat u jednotlivých hypotéz. Ve výsledkové tabulce jsou zvýrazněny váhy odpovědí červenou barvou (červená = vyšší váha), dle předchozího hodnocení účastníků.

testovaná osoba	1.otázka	2.otázka	3.otázka	4.otázka	5.otázka	6.otázka	7.otázka	8.otázka	9.otázka	10.otázka
Adam	50s	7s	D	ano	C	ano	F	A	B	A
Blanka	35s	5s	E	ano	E	ne	E	B	A	B
Bedřich	41s	4s	D	ano	D	ne	E	A	B	B
Cecilie	43s	5s	E	ano	C	ano	F	C	C	B
Franta	50s	4s	D	ano	F	ne	E	B	C	A
Josef	62s	12s	C	ano	F	ne	D	D	B	C
Martin	40s	6s	C	ano	E	ne	E	C	B	B
Tonda	57s	9s	D	ano	F	ne	E	A	A	A
Václav	46s	6s	B	ano	F	ano	F	B	B	B
Zdenka	56s	8s	E	ano	F	ne	E	A	B	A

1. Přihlašování do aplikace je dostačující a nevyžaduje formulář. (otázky: 3)
Z tabulky vyplývá, že přihlašování je dostačující (všichni zvládli úkol). Nicméně spíše doporučuji upravit na formulář, vzhledem k podprůměrnému hodnocení (všichni účastníci testu měli problémy s odhlášením).
2. Pro přehledovou tabulku stačí stránkování po 100 záznamech. (otázky: 7)
Zde je patrné, že výrazná část dotazovaných by ocenilo zobrazení po 200 záznamech. Přesto i zde lze hypotézu potvrdit.
3. Výběrový formulář je přehledný a pochopitelný. (otázky: 8,9)
Lze říct, že nikdo z účastníků neměl větší problémy. Lze konstatovat, že formulář je pochopitelný a přehledný.
4. Submenu je lepší zobrazovat na kliknutí, nikoliv na přejetí kurzorem po hlavním menu. (otázky: 4,5,6)
Všichni účastníci souhlasili s obsahem hlavní nabídky, takže většinou není vůbec třeba podmenu zobrazovat. Navíc mechanismus zobrazení a skrytí kliknutím na položku hlavní nabídky je také pochopitelný. Což potvrdily i výsledky 5.otázky.

7.1.9. Závěr hypotéz

Dle závěrů z hypotéz, doporučuji předělat přihlašování do aplikace (hlavně tedy odhlásování). Předělat stránkování po 200 záznamech, nebo přidat možnost uživatelského nastavení. Co se týká výběrového formuláře, byl při testování shledán menší problém s vybíráním z delších číselníků (elektráren apod.). Ostatní testované funkce lze ponechat v současném stavu.

7.2. Testování uživatelského rozhraní

Webové PHP aplikace mají velkou výhodu při testování, protože skriptovací jazyk obecně je stabilnější při provozu. Díky tomu velkou část chyb může odstranit programátor již

při ladění aplikace. V našem případě jsme měli k dispozici dalšího uživatele, který s námi velmi úzce spolupracoval přímo v prostředí ŠKODA Power a.s. v Plzni. Měli jsme tedy přímou vazbu a mohli jsme se pomocí jeho úrovněového připomínkování rychleji dopracovat k finální podobě programu.

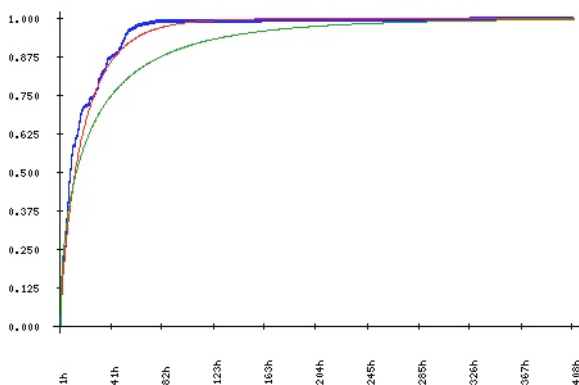
7.2.1. Akceptační testy

Vzhledem k tomu, že se celý projekt byl řešen postupně. Bylo testování prováděno postupně. Začali jsme tedy postupovat dle hlavního menu. V první fázi jsme zpracovali pohled na **data** v přehledové tabulce s detailními náhledy (7.-9.2007). Ve druhé fázi jsme upravili a opravili funkce **výběru** (9.-11.2007). Třetí fáze byla věnována **technickým informacím** a **dalším funkcím** s číselníky a správou uživatelů (11.-12.2007). Ve čtvrté fázi proběhla kontrola a úprava modulu **statistiky** (1.-6.2008). V poslední fázi proběhly závěrečné opravy v uživatelském rozhraní aplikace.

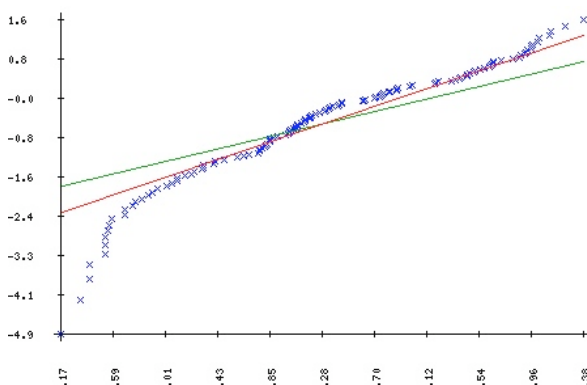
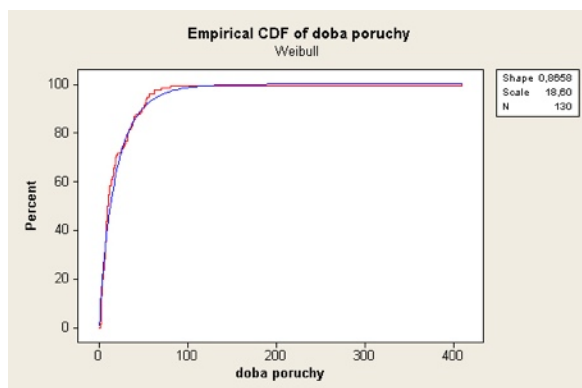
7.3. Statistické funkce

Testovány byly i statistické funkce aplikace. Pro porovnání grafů byly vytvořeny exporty dat do formátu CSV a následně zpracovány aplikacemi Microsoft Excel, Minitab a Statistica. Bohužel funkce, které tato aplikace nabízí z velké části nejsou podporovány ani těmito statistickými nástroji.

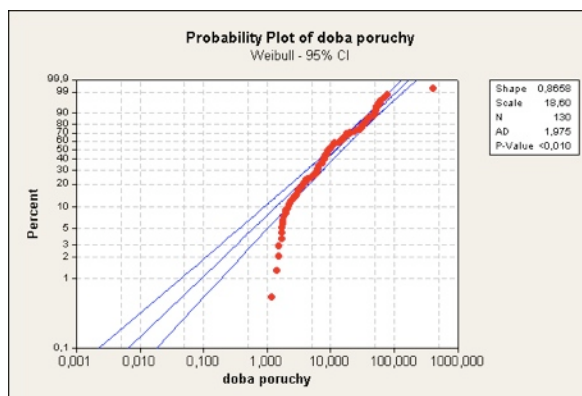
Testování probíhalo na menším výběru dat (cca 130 záznamů). Pro ten jsme provedli statistické výpočty souběžně ve všech aplikacích.

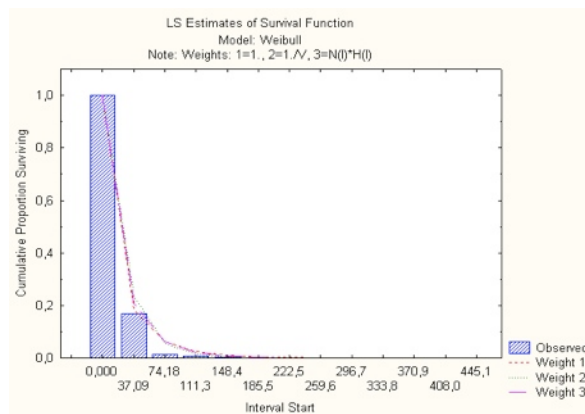
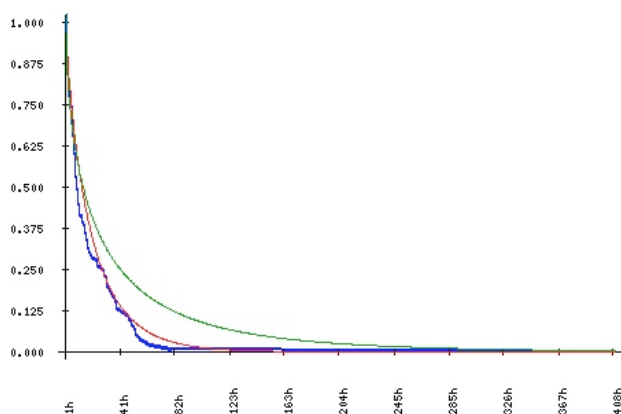


distribuční funkce (doba poruch) aplikace vs. MiniTAB

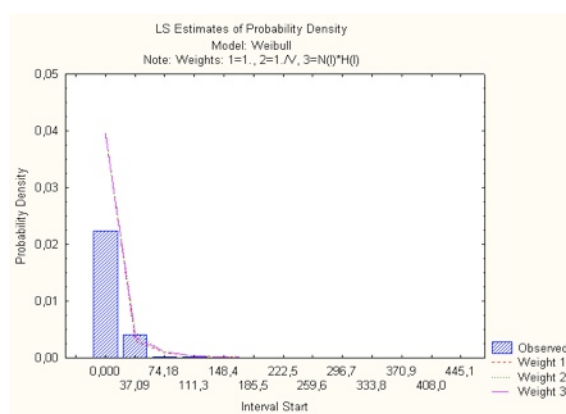
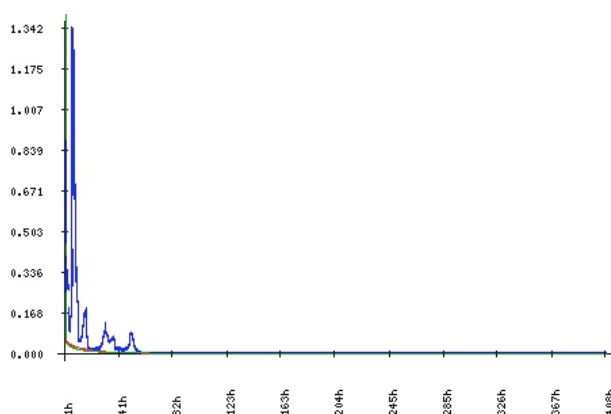


Q-Q graf (doba poruch) aplikace vs. MiniTAB





Spolehlivostní funkce (doba poruch) aplikace vs. Statistica



Hustota pravděpodobnosti (doba poruch) aplikace vs. Statistica

Testy shody byly provedeny jednak na úrovni vypočtených hodnot, jednak na úrovni grafické. Grafické srovnávání bylo síše orientační, neboť různé aplikace zobrazují výsledky různě a některé grafy (například průměrná kumulativní funkce a intenzita opakování) ve standardních statistických programech nejsou vůbec.

7.4. Testování bezpečnosti aplikace

Přestože je aplikace vyvíjena pouze pro provoz na vnitřní síti zadavatele, je nutné zajistit bezpečnost i v případě, že je útočník již v této síti. Proto proběhlo několik bezpečnostních opatření. Nicméně, předpokládá se, že serverová část aplikace a datový server by měly být kryty před případnými útoky zvenčí firewallem zadavatele.

7.4.1. Proniknutí do systému bez přihlášení

Při testování bezpečnosti bylo ověřeno, že do systému nelze proniknout bez znalosti přihlašovacích informací uživatele (jména a hesla). Bylo testováno, zda pokus o přístup na jinou než úvodní stránku bez přihlášení neposkytne žádné informace a pouze zobrazí přihlašovací formulář a chybové hlášení.

7.4.2. Odposlech spojení mezi klientem a serverem

Vzhledem k tomu, že aplikace je navrhována především pro provoz ve vnitřní síti zadavatele a nejedná se o příliš citlivá data, není třeba řešit zabezpečení spojení mezi klientem

a serverem. Přesto doporučuji provoz na HTTP serveru s SSL vrstvou. Pokud aplikace využívá šifrovanou komunikaci mezi klientem a serverem není možné odposlechnout libovolnou část komunikace, tedy ani přihlášení do aplikace. HTTPS spojení pomáhá např. k omezení možností útoku typu session stealing.

7.4.3. Útok pomocí dotazů (SQL injection)

Tento typ útoku byl uvažován v případě přihlašovacího formuláře, kde jsou jiné než alfanumerické hodnoty odfiltrovány, takže neumožňují SQL injection útok. Uvnitř aplikace tento útok není uvažován, protože předávané parametry jsou již ve formátu SQL dotazů a nelze tedy rozlišit správné hodnoty od SQL injection útoků.

8. Uživatelský návod

Uživatelský návod je důležitou součástí systému. Uživatel by v něm měl nalézt všechny potřebné informace pro ovládání aplikace. Na závěr návodu jsou popsána možná chybová hlášení, nicméně při správné instalaci by se neměla objevit.

8.1. Přehled použitých ikon



indikátor existence anotace či poznámky k záznamu o poruše. Kliknutím na tuto ikonu se objeví okno s obsahem.



indikátor existence přílohy. Podrobnosti o příloze se zobrazí při najetí kurzorem na tuto ikonu v okénku nápovědy (hintu). Kliknutím otevřete přílohu v asociované aplikaci (MS Office a další, pokud jsou odpovídající asociace nastaveny v operačním systému).



zobrazení detailu. V případě tabulky dat se zobrazí detail záznamu v okénku, v případě grafů se zobrazí graf na celou obrazovku, navíc s možností jeho zvětšení (kurzor s lupou).



editace záznamu v tabulce dat. Tato ikona je použita i pro vytváření seznamů (ve výběru, při výběru poškozených součástí).



ukončení vytváření seznamu (při nastavení výběru nebo při sestavování seznamu poškozených součástí).



zrušení záznamu. Kliknutí na tuto ikonu vyřadí záznam z tabulky.



export dat do CSV souboru (viz Export dat).



indikátor uspořádání tabulky podle daného atributu (sloupce, položky)



skok na začátek stránky (například v této nápovědě jsou tyto ikony při pravém okraji obrazovky)



uzavření okna - používá se při zobrazení detailů záznamů, anotací, poznámek. Je zpravidla umístěna v pravém horním rohu okna.



ikona pro spuštění vyhledávání textu na stránce (fulltextové vyhledávání)



zobrazení skrytého grafu. Pro urychlení zobrazování stránek jsou některé grafy skryté a zobrazují se pouze na vyžádání kliknutím na tuto ikonu.

8.2. Navigace v programu

Po spuštění se na obrazovce objeví formulář pro prohlížení záznamů o poruchách. Základní pohyb po programu je zajištěn **hlavní nabídkou** v programové liště. Tato hlavní nabídka obsahuje navigaci do základních funkcí programu:

Pro rychlé přecházení z jedné části programu do jiné je výhodné otevřít tyto části v různých panelech prohlížeče (pokud to umožňuje) a pouze mezi nimi přepínat.

Jednotlivé části programu mají další, vlastní **podnabídku funkcí**. Tato podnabídka se objevuje pod programovou lištou. Tak například funkce statistiky mají vlastní podnabídku:

četnosti poruch | doby poruch | doby mezi poruchami | rozdělení dob poruch | zpět

Teprve po výběru z podnabídky se objeví odpovídající obrazovka. Podnabídka přitom zmizí a pro její opětovné vyvolání je třeba znovu vybrat základní funkci z hlavní nabídky. Tedy chceme-li přejít ze statistiky pro dobu poruchy na statistiku dob mezi poruchami, musíme znovu volit funkci **statistiky** v hlavní nabídce a poté **doby mezi poruchami** v podnabídce.

V některých částech programu (data, statistiky) je kromě podnabídky, sloužící k upřesnění vybrané funkce, ještě i **nabídka pro užší výběr**. Tak například v prohlížení záznamů o poruchách (**data**) lze vybrat z této nabídky pro užší výběr které z následujících záznamů chcete zobrazit:

Porucha | Reklamace | Hlášení o neshodě | Protokol o přejímce

Podobně ve **statistiky > četnosti poruch** můžete tímto způsobem vybrat faktor, podle kterého budete data třídit.

Pohyb po datech a mezi grafy se provádí obvyklým způsobem pomocí myši či kurzorových kláves. Některé údaje v tabulce dat mají speciální funkci. Tyto údaje jsou obvykle zobrazeny jako hypertextové položky (po najetí kurzorem změna barvy či podtržení). Zpravidla neodkazují na jiné místo v programu, ale jejich zvláštní funkce spočívá v seřazení dat (názvy sloupců v tabulce) nebo ke zúžení výběru na všechny záznamy s touto hodnotou (v tabulce například název elektrárny nebo výkon).

Speciální způsob navigace obsahuje tabulka záznamů o poruchách v oddíle **data**. Nad tabulkou - pod názvem tabulky, nad levým horním rohem - a v jejím posledním řádku je **navigační řádek** :

zobrazit v tabulce: << | < | **vše** | > | >>

V takovémto případě je v tabulce zobrazeno najednou nejvýše 100 záznamů a šipky vlevo nebo vpravo umožňují posun o další stovku vpřed nebo vzad, na konec nebo na začátek. Kliknutím na slovo "**vše**" se změni způsob zobrazení tak, že v tabulce se zobrazí všechny záznamy najednou, což může být časově náročné. Navigační řádek potom bude mít tvar

zobrazit v tabulce: pouze **100**

Kliknutím na číslo **100** přepnete zpět na režim zobrazení po stovce záznamů.

8.3. Funkce prohlížeče

Program PORUCHY je koncipován jako třívrstvá aplikace. Tou nejvyšší - třetí vrstvou je takzvaný "tenký klient", tvořený internetovým prohlížečem. Tím se internetový prohlížeč stává součástí systému a je třeba do jisté míry znát i jeho funkce a nastavení. Protože různé prohlížeče a jejich různé verze mají různé ovládání a nastavení, uvedeme zde popis nejdůležitějších funkcí, které mohou bezprostředně ovlivnit práci s programem. Jejich nastavení či ovládání lze nalézt v nápovědě k prohlížeči samotnému (zpravidla v hlavní nabídce pod položkou **Nápověda**).

Nejdůležitějším nastavením prohlížeče je povolení tzv. cookies. Názvem "cookies" se označuje malé množství dat, která webový server pošle prohlížeči, který je uloží na vašem počítači. Při každém spuštění programu pak prohlížeč tato data posílá zpět serveru. Cookies slouží k rozlišování jednotlivých uživatelů, ukládají se do nich uživatelské předvolby apod. Protože cookies mohou představovat jisté nebezpečí pro ochranu osobních údajů, na některých počítačích může být jejich používání vypnuté. To však vylučuje použití tohoto programu. Zapnutí cookies pro různé prohlížeče je popsáno v nápovědě prohlížeče.

Další funkcí prohlížeče, nutnou pro bezproblémovou funkci programu, je podpora JavaScriptu. Aplikací část programu má dvě části - jedna, která je napsána v jazyce PHP, běží na aplikačním (webovém) serveru a na uživatele nekladou žádné požadavky. Druhá část programu, tvořena malými samostatnými procedurami napsanými v javascriptu, musí běžet na počítači uživatele. Je to část programu, vkládaná přímo do kódu HTML stránky, která ovládá různé interaktivní prvky uživatelského prostředí (tlačítka, textová políčka) nebo některé efekty obrázků. Proto program potřebuje podporu tohoto jazyka. Tu poskytují internetové prohlížeče. I zde je třeba jisté opatrnosti vzhledem k bezpečnosti počítače a proto může být tato podpora vypnuta. V této nápovědě naleznete návod, jak JavaScript zapnout v různých prohlížečích.

Program PORUCHY nemá vlastní ovládání tiskárny. Veškeré tisky se provádějí přes funkci prohlížeče, kterou najdete v hlavní nabídce pod položkou **Soubor > Tisk**. Použití této funkce v programu je popsáno v dalších odstavcích této nápovědy. Pro tisk tabulek v barvách je třeba v nastavení prohlížeče povolit tisk pozadí a obrázků. Většina prohlížečů dovolí nastavit text zápatí, záhlaví, velikost okrajů a číslování stránek.

Možnosti prohlížeče mohou usnadnit a zrychlit práci s programem. Takové usnadnění může přinést například možnost uložení přístupových údajů či používání záložek.

8.4. Tisk

Tisk je v aplikaci řešeno pomocí webového prohlížeče. Aplikace je navržena tak, že skrývá pro tisk veškeré ovládací prvky. Tím se vytiskne vždy jen část okna. Navíc lze tisknout libovolnou aplikační obrazovku.

Pro tisk tabulek v barvách je třeba povolit tisk pozadí a obrázků. Při tisku lze v prohlížeči nastavit text zápatí a záhlaví, stejně jako okraje a číslování stránek.

8.5. Chybová hlášení

V aplikaci je většina chybových hlášení ošetřena. Pokud nastane v programu chyba, zkuste obnovit stránku, teprve pak prosím kontaktujte správce Vašeho serveru. Děkujeme.

Nepodařilo se připojit k databázi!!

Chyba připojení aplikace k databázi. Příčinou může být nedostupný DB server, nebo chybný DB soubor či přihlašovací údaje.

Chyba SQL příkazu!!

Chyba při načítání dat z databáze. Při najetí na hlášku se zobrazí chybný SQL dotaz.

Nelze najít editovaný záznam!!

Chyba při vyvolání editace libovolného záznamu. Je způsobena buď chybným předáním parametrů dotazu, nebo nedostupností záznamu z databáze.

Nelze zkopírovat soubor!!

Chyba při nahrávání přílohy k záznamu o poruše, nebo technické informace.

....

Další chybové hlášky mohou být způsobeny nedostupností aplikace na server, popř. chybnou instalací aplikace.

8.6. Hlavní nabídka

8.6.1. Data



8.6.1.1. Prohlížení a editace záznamů o poruchách

Jednou ze základních funkcí programu je evidence údajů o poruchách. Program nabízí dva základní typy dat: záznamy o poruchách (zahrnující také záznamy o reklamacích, hlášení o neshodách, protokoly o přejímkách) a technické informace o elektrárnách a jejich blocích. Tato data jsou dostupná z hlavní nabídky volbou **data** nebo **technické informace**. V obou těchto částech programu lze prohlížet záznamy, upravovat je, doplňovat a odebírat. Pro uživatele je na této stránce několik důležitých prvků:

- **Lokální nabídka typu zobrazovaných dokumentů** v pravém horním rohu obrazovky (pod lištou s hlavní nabídkou) obsahuje tyto položky:

[Porucha](#) | [Reklamace](#) | [Hlášení o neshodě](#) | [Protokol o přejímce](#)

V tabulce záznamů se zobrazí pouze vybrané typy dokumentů (po vybrání jsou zvýrazněny červeně), přičemž jich lze vybrat i více najednou. Typ dokumentu (záznamu) je v tabulce uveden zkratkou v prvním sloupci.

Po spuštění programu není tento výběr nastaven a je třeba začít právě zde!

- **Navigační řádek** pod názvem tabulky, nad levým horním rohem tabulky se záznamy o poruchách:

zobrazit v tabulce: << | < | **vše** | > | >>

V takovémto případě je v tabulce zobrazeno najednou nejvýše 100 záznamů a šipky vlevo nebo vpravo umožňují posun o další stovku vpřed nebo vzad, na konec nebo na začátek. Kliknutím na slovo "**vše**" se změní způsob zobrazení tak, že v tabulce se zobrazí všechny záznamy najednou, což může být časově náročné. Navigační řádek potom bude mít tvar

zobrazit v tabulce: pouze **100**

Kliknutím na číslo **100** přepnete zpět na režim zobrazení po stovce záznamů.

8.6.1.2. Funkce tabulky

Tabulka záznamů o poruchách obsahuje prvky pro editaci, vkládání a mazání záznamů a další funkce pro zobrazení. V tabulce odpovídá každé poruše jeden řádek, sloupce odpovídají vybraným údajům o poruše:

typ - typ záznamu označuje jednu z variant: porucha (PO), reklamace (RE), hlášení o neshodě (HN) nebo protokol o přejímce (PR). Kliknutím na vybraný typ záznamu lze provést rychlý dočasný výběr všech záznamů pro tento typ.

elektrárna - zkratka názvu elektrárny, v níž došlo k poruše. Zkratka je uvedena podle číselníku elektráren. Kliknutím na název elektrárny lze provést rychlý dočasný výběr všech záznamů pro tuto elektrárnu.

blok - číslo bloku elektrárny, k němuž se záznam vztahuje. Kliknutím na číslo bloku lze provést rychlý dočasný výběr všech záznamů pro všechny bloky stejného čísla.

výkon - nominální výkon bloku. Výkon je uveden v MWh. Zhruba lze po vynásobení tohoto výkonu délkou poruchy (výpadku) spočítat velikost výpadku v MW. Kliknutím lze provést rychlý dočasný výběr všech záznamů se stejným výkonem.

čas poruchy - datum a čas, kdy došlo k poruše ve formátu dd.mm.rrrr hh:mm (den.měsíc.rok hodina:minuta). Přidržením kurzoru na tomto údaji se v okénku objeví datum a čas konce poruchy ve stejném formátu.

dobu poruchy - délka trvání poruchy v hodinách. Přidržením kurzoru na tomto údaji se v okénku objeví délka poruchy přepočítaná na dny, hodiny a minuty a výkon v MWh, vypočítaný podle výkonu bloku před vznikem poruchy, pokud je znám.

porouchané součásti - seznam obsahuje názvy součástí podle číselníku. Pokud nejsou v číselníku uvedeny, potom je v seznamu pouze jejich číslo. Kliknutím na název nebo číslo součásti lze provést rychlý dočasný výběr všech záznamů o poruchách, při nichž došlo k poškození vybrané součásti.

D,A,P,P - ikony pro zobrazení detailního záznamu ( - zobrazí všechny uložené údaje záznamu), anotace () , poznámky () a případných příloh ( ), uložených na datovém

serveru. Pokud záznam neobsahuje poznámku, anotaci nebo přílohu, potom příslušná ikona u tohoto záznamu chybí. Kliknutím na ikonu přílohy se tato zobrazí v samostatném okně aplikace, která je s touto přílohou asociována v operačním systému (zpravidla MS Word, Excel, grafický editor a podobně).

sloupec s ikonami pro editaci () a mazání () záznamů. Kliknutím na ikonu  v záhlaví tabulky lze provést export dat z tabulky do textového souboru, v němž jsou jednotlivé položky odděleny středníkem (tzv. "csv" soubor). Taskovýto soubor lze prohlížet přímo v tabulkovém editoru MS EXCEL.

Podvýběr, který vytvoříte kliknutím na obsah políčka v prvních čtyřech sloupcích tabulky, se zobrazí nad navigačním řádkem spolu s textem [Zrušit podvýběr]. Kliknutím na tento text bude příslušný podvýběr zrušen.

První řádek tabulky obsahuje názvy sloupců. Kliknutím myši na název některého z prvních šesti sloupců provedete setřídění záznamů vzestupně nebo sestupně podle hodnot tohoto sloupce. První kliknutí setřídí vzestupně, druhé sestupně a třetí toto setřídění ruší. Po kliknutí se na obrazovce zpravidla objeví hlášení "... *prosím vyčkejte, server zpracovává Váš dotaz ...*". Toto hlášení se objevuje vždy, když jsou data na serveru zpracovávána a klient (Váš prohlížeč) čeká na odpověď. Setřídění je indikováno ikonou  nebo  u názvu sloupce.

Poslední řádek tabulky obsahuje opět navigační řádek a v pravém sloupci poskytuje možnost [přidat záznam]. Kliknutím na tato slova otevřete editační formulář pro vytvoření nového záznamu.

8.6.1.3. Editace záznamu

Editaci (úpravu, změnu) stávajícího záznamu vyvoláte kliknutím na ikonu pro editaci  v pravém sloupci na odpovídajícím řádku tabulky. Po tomto výběru se na obrazovce objeví formulář pro editaci. Tento formulář je stejný jak pro vstup nového záznamu, tak i pro editaci stávajících. Formulář je rozdělen do čtyř částí:

- v první části formuláře zadáte typ dokumentu (porucha, hlášení o neshodě, protokol o přejímce nebo reklamace) a číslo záznamu
- do druhé části zadáváte další údaje, které jsou součástí záznamu. Tyto údaje se vkládají obvyklým způsobem do vstupních polí buď přímo z klávesnice nebo výběrem z roletkové nabídky (tzv. "combo box").
- do třetí části formuláře lze zadat seznam poškozených součástí. Zadávání seznamu je poněkud nestandardní. Začneme tak, že klikneme na ikonu  napravo. Tím se otevře seznam součástí, z něhož
 - kliknutím na nevybranou součást (modrou) tuto vybereme a kliknutím na [OK] v dotazovacím okénku ji označíme jako poškozenou prvotně (P),
 - kliknutím na nevybranou součást (modrou) tuto vybereme a kliknutím na [Zrušit] v dotazovacím okénku ji označíme jako poškozenou druhotně (N),
 - kliknutím na vybranou součást (červenou) tuto odebereme ze seznamu poškozených součástí k danému záznamu.

Při výběru se ikona pro editaci změní na . Výběr ukončíme kliknutím na tuto přeškrtnutou tužku. Ve formuláři zůstanou pouze vybrané součásti s indikací primárního (P) či sekundárního (N) poškození.

- čtvrtá, poslední část formuláře obsahuje údaje o přílohách. Informace o přidružené příloze má dvě části: název přílohy (v levém sloupci) a odkaz na soubor s přílohou (pravý sloupec). Odkaz na přílohu se vkládá výběrem souboru z disku po stisku tlačítka [Procházet]. Máte-li výběr přílohy dokončen, do databáze ji vložíte stiskem tlačítka [přidat]

Na spodním řádku editačního formuláře je několik tlačítek. Ty slouží - podle označení - k **odeslání** záznamu do databáze, ke **zrušení** editace beze změny záznamů v databázi, pro vložení **anotace** či **poznámky**. Pro vkládání textu anotace či poznámky je nabídnuto okno s editorem textu, umožňujícím jakési základní formátování (Rich Text Editor).

8.6.1.4. Vložení nového záznamu

Nový záznam lze vložit použitím funkce [přidat záznam], jejíž odkaz je v pravém dolním rohu tabulky (v posledním řádku). Po kliknutí na text [přidat záznam] se objeví formulář pro vstup nového záznamu. Tento formulář je stejný jako pro editaci. Formulář je rozdělen do čtyř částí:

- v první části formuláře zadáte typ dokumentu (porucha, hlášení o neshodě, protokol o převímce nebo reklamace) a číslo záznamu
- do druhé části zadáváte další údaje, které jsou součástí záznamu. Tyto údaje se vkládají obvyklým způsobem do vstupních polí buď přímo z klávesnice nebo výběrem z roletkové nabídky (tzv. "combo box").
- do třetí části formuláře lze zadat seznam poškozených součástí. Zadávání seznamu je poněkud nestandardní. Začneme tak, že klikneme na ikonu  napravo. Tím se otevře seznam součástí, z něhož
 - kliknutím na nevybranou součást (modrou) tuto vybereme a kliknutím na [OK] v dotazovacím okénku ji označíme jako poškozeno prvotně (P),
 - kliknutím na nevybranou součást (modrou) tuto vybereme a kliknutím na [Zrušit] v dotazovacím okénku ji označíme jako poškozenou druhotně (N),
 - kliknutím na vybranou součást (červenou) tuto odebereme ze seznamu poškozených součástí k danému záznamu.
- Při výběru se ikona pro editaci změní na . Výběr ukončíme kliknutím na tuto přeškrtnutou tužku. Ve formuláři zůstanou pouze vybrané součásti s indikací primárního (P) či sekundárního (N) poškození.
- čtvrtá, poslední část formuláře obsahuje údaje o přílohách. Informace o přidružené příloze má dvě části: název přílohy (v levém sloupci) a odkaz na soubor s přílohou (pravý sloupec). Odkaz na přílohu se vkládá výběrem souboru z disku po stisku tlačítka [Procházet]. Máte-li výběr přílohy dokončen, do databáze ji vložíte stiskem tlačítka [přidat]

Na spodním řádku editačního formuláře je několik tlačítek. Ty slouží - podle označení - k **odeslání** záznamu do databáze, ke **zrušení** editace beze změny záznamů v databázi, pro vložení **anotace** či **poznámky**. Pro vkládání textu anotace či poznámky je nabídnuto okno s editorem textu, umožňujícím jakési základní formátování (Rich Text Editor).

8.6.1.5. Zrušení záznamu

Pro zrušení záznamu slouží ikona ✕. Kliknutím myši na tuto ikonu budete vyzváni k potvrzení úmyslu zrušit daný záznam. Teprve po tomto potvrzení dojde k jeho vyřazení z tabulky a tabulka se znovu překreslí.

8.6.2. Výběr



8.6.2.1. Výběr a vyhledávání záznamů

Výběr záznamů pro zobrazení a výpočty lze v programu provádět v zásadě třemi způsoby:

- **rychlým podvýběrem** přímo z tabulky dat,
- **fulltextovým vyhledáváním** záznamů podle zadaného textového řetězce,
- na základě zadání výběrových kritérií funkcí **výběr** v hlavní nabídce.

Zatímco první dva případy nemají vliv na statistické výpočty, třetí typ výběru se projeví i ve statistických výpočtech. Proto je třeba při jeho používání dbát na to, aby výsledný počet záznamů ve výběru byl dostatečný pro statistická zobrazení a odhady v části **statistiky**.

Formulář pro **výběr** se od ostatních liší barvou (je modrý) a je rozdělen do tří částí:

- horní a největší část pro nastavení podmínek výběru,
- prostřední část pro manipulaci s výběrem - nastavenou podmínku lze uložit pod jménem a takto uloženou opět načíst do formuláře,
- spodní řádek s nabídkou funkcí výběru.

8.6.2.2. Nastavení výběru

Výběrem se zde rozumí podmnožina uložených dat, vyhovující zadaným podmínkám. Takto vytvořená podmnožina dat se potom používá i ke statistickým výpočtům, dokud není zrušena. Pro vytvoření podmínky výběru slouží speciální editor, dostupný z hlavní nabídky jako **výběr**. Podmínky lze nastavovat v zásadě dvěma způsoby:

- **číselné podle hodnoty**: do vstupního pole lze zadat požadovanou hodnotu pro výběr dané položky (například *číslo stroje*).
- **číselné od - do**: do dvou vstupních polí lze zadat minimální a maximální hodnotu pro výběr dané položky. Přitom stačí zadat pouze první z nich (minimální). V takovém případě se druhá (maximální) hodnota automaticky doplní první hodnotou a výběr bude proveden pouze pro tuto jedinou hodnotu. Tímto způsobem se například zadává výběrová podmínka pro *datum poruchy* a *délku výpadku*.

- výběrem ze seznamu: nabídku přípustných hodnot vyvoláte kliknutím na ikonu . Výběr požadovaných hodnot provedete kliknutím myši na tyto hodnoty (změní barvu). Výběr hodnot do podmínky ukončíte kliknutím na ikonu . Tímto způsobem můžete nastavit podmínku například pro *číslo bloku* nebo *příčinu poruchy*. Pokud je seznam možných hodnot delší, je možno tento seznam zúžit. K tomu slouží editační okénko na začátku seznamu s malou ikonou lupy (například *elektrárna*). Po zadání části textu kliknutí na ikonu bude nabídka možností zúžena pouze na ty hodnoty, které obsahují zadaný podřetězec. Tak například pro výběr podle poškozené součásti, zadáte-li do okénka znak "3" a kliknete na , v seznamu k výběru se objeví pouze součásti, jejichž číslo začíná na "3". Z nich lze nyní vybrat pouze ty, podle nichž chceme vybírat. Chcete-li vybrat všechny položky ze seznamu, stačí použít ikonu  napravo od lupy. Zrušení označení položek pro výběr lze provést kliknutím na ikonu . (Tyto ikony jsou pouze u nastavení výběru podle poškozené součásti.)

8.6.2.3. Manipulace s výběrem

Pro usnadnění práce s výběry, které se často opakují, lze využít možnost uložení nastavených podmínek výběru a jejich opětovné načtení. To může zrychlit práci, případně pomoci při tvorbě složitých podmínek. Nabídka funkcí pro práci s výběrem je ve dvou řádcích:

- funkce pro práci s uloženými podmínkami výběru - v tomto řádku je třeba nejprve vybrat z roletové nabídky vpravo název uloženého výběru. Tento uložený výběr lze
 - **načíst** do formuláře;
 - přepsat aktuálně nastavenými podmínkami výběru volbou funkce **uložit jako** (původní obsah uložených podmínek se přepíše aktuálními);
 - **smazat**.
- funkce **uložit jako nový výběr** umožní aktuálně nastavené podmínky uložit na disk pod zadaným jménem (jméno zadejte do editačního okénka napravo).

8.6.2.4. Funkce výběru

Po nastavení podmínek výběru lze provést několik operací, charakterizovaných nabídkou funkcí v posledním řádku:

- **aplikovat** výběr - tato funkce provede vlastní výběr záznamů podle zadaného kritéria.
- **vyčistit** nastavení výběru - smaže všechny nastavené podmínky ve formuláři; tato funkce neovlivní již provedený výběr.
- **zrušit** výběr - zruší výběr záznamů; v záložce **data** se zobrazí opět všechny uložené záznamy.
- **zpět** - skryje formulář pro nastavení výběrových podmínek.

8.6.3. Statistiky



8.6.3.1. Četnosti poruch

Tato část statistické analýzy záznamů o poruchách poskytuje informace o rozdělení počtu poruch podle různých hledisek: podle elektrárny, bloku, nominálního výkonu, poškozené součásti a dalších. Hledisko pro třídění lze vybrat kliknutím na název položky v pravé horní části obrazovky, pod nabídkovou lištou. Obrazovka obsahuje **tabulku četností** poruch v jednotlivých třídách, **histogram** absolutních četností a **kruhový graf** relativních četností poruch.

- **Tabulka četností** je rozdělena do čtyř sloupců:
 - *číslo třídy*, které se objevuje i v popisu sloupců histogramu
 - *název třídy*, odpovídající hodnotám třídícího znaku,
 - *absolutní četnost* poruch pro odpovídající třídu,
 - *relativní četnost* poruch pro odpovídající třídu (procentuální zastoupení vzhledem k celkovému počtu poruch).
- Nad tabulkou je **okno pro nastavení počtu zobrazovaných tříd** počínaje nejčtenější v sestupném uspořádání. V případě, že tabulka obsahuje velký počet tříd, je tato funkce užitečná. Stačí do editačního okna napsat požadovaný počet a kliknout na tlačítko *Aplikovat*.
Jednotlivé třídy (podle hodnot vybraného znaku, například jednotlivé typy odstávky) jsou na obrazovce charakterizovány různými barvami. To umožňuje využít četnostní tabulku zároveň jako legendu ke grafům v pravé části obrazovky.
- **Histogram absolutních četností** (třída x absolutní četnost) zobrazuje absolutní četnosti poruch pro jednotlivé třídy pomocí sloupkového grafu. Barvy sloupků (tříd) odpovídají řádkům v tabulce četností poruch. Po najetí kurzoru na jednotlivé sloupce grafu se objeví malé okénko s informací o odpovídající třídě (hint).
- **Kruhový graf relativních četností** zobrazuje relativní četnosti poruch pro jednotlivé třídy vzhledem k celkovému počtu všech poruch v aktuálním výběru. Barvy výsečí (tříd) odpovídají řádkům v tabulce četností. Po najetí kurzoru na jednotlivé kruhové výseče se objeví malé okénko s informací o odpovídající třídě (hint).

V pravém dolním rohu každého grafu je ikona  pro jeho celoobrazovkové zobrazení. To lze použít, chceme-li obrázek například tisknout nebo kopírovat.

8.6.3.2. Doby poruch

Výpočet charakteristik a četnostní analýza délky poruchy. V levé části obrazovky je **tabulka se zdrojovými daty**, vpravo nahoře je **výsledková tabulka**. Ta obsahuje různé informace, mezi kterými lze vybírat pomocí lokální nabídky:

[charakteristiky](#) | [histogram](#) | [kumulativní histogram](#)

Pod výsledkovou tabulkou jsou umístěny jednotlivé **grafy**.

V pravém dolním rohu každého grafu je ikona  pro jeho celoobrazovkové zobrazení. To lze použít, chceme-li obrázek například tisknout nebo kopírovat.

- **Tabulka se zdrojovými daty** obsahuje údaje o identifikaci poruchy z hlediska
 - *elektrárny,*
 - *bloku,*
 - *data a času vzniku poruchy,*
 - *dobou poruchy* v hodinách.
- Řádky tabulky jsou uspořádány vzestupně podle doby poruchy a barevně odpovídají jednotlivým třídám v tabulce četností (ve výsledkové tabulce v pravé horní části obrazovky po volbě *histogram*).
 Po najetí kurzoru na datum a čas vzniku poruchy v libovolném řádku tabulky se objeví pod kurzorem okénko (hint) s informací o datu a času ukončení poruchy. Doba poruchy je potom rozdílem časů konce a začátku poruchy v hodinách.
- **Výsledková tabulka - charakteristiky** obsahuje
 - ***popisné statistiky datového souboru:***
 - počet záznamů ve výběru, ze kterého jsou charakteristiky počítány
 - rozsah hodnot - rozdíl maximální a minimální hodnoty,
 - aritmetický (výběrový) průměr dob poruch ve výběru,
 - výběrový rozptyl,
 - (výběrovou) směrodatnou odchylku,
 - minimum,
 - dolní kvartil,
 - horní kvartil,
 - medián,
 - maximum.
 - ***odhady parametrů Weibullova modelu***, to jest parametru měřítka (alfa) v hodinách a parametru tvaru (beta)
 - metodou maximální věrohodnosti (ML)
 - metodou nejmenších čtverců (LS)
 - ***výsledky testů shody*** odhadnutého Weibullova modelu a naměřených dat v aktuálním výběru. Pro ověření modelu jsou použity dvě metody:
 - Andersonův-Darlingův test a
 - Lillieforsův test.
 - U každého testu je uvedena hodnota testové statistiky a její *p*-hodnota.
 - ***Grafy.*** V nabídce je celkem 10 různých statistických a pravděpodobnostních grafů:
 - ***bodový (rozptylový) graf*** (datum x doba poruchy).
 Zobrazuje zdrojová (netransformovaná) data. Na ose *x* je datum vzniku poruchy, ve směru osy *y* je zobrazena délka poruchy. Každé poruše odpovídá jeden křížek.
 - ***Q-Q graf - graf*** transformovaných hodnot.
 Na ose *x* je zobrazena hodnota logaritmu délky poruchy ($\ln(x)$), na ose *y* jsou hodnoty dvojitého logaritmu převrácené hodnoty doplňku empirické distribuční funkce do jedničky ($\ln(\ln(1/(1-Fn(x))))$). Pokud zdrojová data odpovídají Weibullovu modelu, měla by všechna ležet v přímce. Odchylka od přímky znamená odchylku od teoretického Weibullova rozdělení. Do grafu jsou zakresleny přímky, odpovídající odhadnutému Weibullovu rozdělení pravděpodobnosti (červená

pro odhad metodou maximální věrohodnosti a zelená pro odhad metodou nejmenších čtverců). Rovnice těchto přímek jsou uvedeny pod grafem.

- graf distribuční funkce (doba poruchy x pravděpodobnost)
Empirická distribuční funkce odhadnutá z napozorovaných dat je zobrazena modrou barvou. Vedle toho jsou zobrazeny i distribuční funkce Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).
 - graf spolehlivostní funkce (doba poruchy x pravděpodobnost)
Empirická spolehlivostní funkce odhadnutá z napozorovaných dat (pomocí empirické distribuční funkce) je zobrazena modrou barvou. Vedle toho jsou zobrazeny i funkce spolehlivosti Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).
 - graf hustoty pravděpodobnosti (doba poruchy x intenzita)
Hustota odhadnutá z napozorovaných dat (jako numerická derivace empirické distribuční funkce) je zobrazena modrou barvou. Vedle toho jsou zobrazeny i hustoty Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).
 - graf intenzity doby poruchy (doba poruchy x intenzita)
Intenzita odhadnutá z napozorovaných dat (jako podíl empirické hustoty a empirické funkce spolehlivosti) je zobrazena modrou barvou. Vedle toho jsou zobrazeny i intenzity Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).
 - rozdílový graf cdf (doba poruchy x pravděpodobnost)
Graf zobrazuje odchylky empirické distribuční funkce od distribuční funkce (cdf) Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).
 - graf průměrné kumulativní funkce (doba poruchy x relativní počet poruch)
Zobrazuje vývoj kumulativní četnosti poruch vzhledem k době poruchy. Graf obsahuje pás 95% spolehlivosti pro odhad počtu poruch.
 - graf intenzity trvání (doba poruchy x intenzita)
Numerická derivace průměrné kumulativní funkce. Zobrazuje rychlost přírůstku kumulativního počtu poruch pro danou dobu poruchy na ose x .
- **Výsledková tabulka - histogram** obsahuje četnostní analýzu dob trvání poruchy. Zahrnuje
 - **výpočet frekvenční tabulky**, to znamená
 - rozřídění do tříd (počet tříd lze nastavit v dolním řádku tabulky). Třídy jsou charakterizovány pořadovým číslem, dolní a horní mezí v hodinách.

- výpočet absolutních četností (prostých i kumulativních) pro jednotlivé třídy (počty poruch).
- výpočet relativních četností (prostých i kumulativních) pro jednotlivé třídy (v %).
- výpočet teoretických (prostých) relativních četností za předpokladu Weibullova rozdělení a difference mezi empirickými a teoretickými četnostmi. Výpočet je proveden pro Weibullovo rozdělení s parametry odhadnutými buď metodou maximální věrohodnosti (ML) nebo metodou nejmenších čtverců (LS). Mezi těmito modely lze přepínat v dolním řádku tabulky.
- V posledním řádku tabulky lze
 - v levé části nastavit požadovaný počet třídních intervalů zapsáním příslušného čísla do editačního okna a kliknutím na tlačítko *Aktivovat*,
 - v pravé části přepnout mezi způsobem odhadu parametrů Weibullova rozdělení - mezi metodou nejmenších čtverců (LS) a metodou maximální věrohodnosti (ML).
- **výpočet chí-kvadrát testu dobré shody** empirického a teoretického Weibullova rozdělení, odhadnutého buď metodou nejmenších čtverců nebo maximální věrohodnosti, podle nastavení v pravé části posledního řádku tabulky. Výsledky testu jsou zobrazeny ve střední části posledního řádku tabulky a obsahují informaci o
 - počtu stupňů volnosti (v závorce),
 - hodnotě testové statistiky a
 - odpovídající *p*-hodnotě.
- **grafické znázornění**
 - histogramu četností (třída x relativní četnost), v němž jednotlivé sloupce odpovídají barevně třídám v tabulce.
 - rozdílového histogramu (třída x rozdíl relativních četností) - zobrazuje odchylky empirických a teoretických relativních četností pro jednotlivé Weibullovské modely (ML) a (LS).
- **Výsledková tabulka - kumulativní histogram** obsahuje četnostní analýzu dob trvání poruchy. Zahrnuje
 - **výpočet frekvenční tabulky**, to znamená
 - roztrídění do tříd (počet tříd lze nastavit v dolním řádku tabulky). Třídy jsou charakterizovány pořadovým číslem, dolní a horní mezí v hodinách.
 - výpočet absolutních četností (prostých i kumulativních) pro jednotlivé třídy (počty poruch).
 - výpočet relativních četností (prostých i kumulativních) pro jednotlivé třídy (v %).
 - výpočet teoretických (kumulativních) relativních četností za předpokladu Weibullova rozdělení a difference mezi empirickými a teoretickými četnostmi. Výpočet je proveden pro Weibullovo rozdělení s parametry odhadnutými buď metodou maximální věrohodnosti (ML)

nebo metodou nejmenších čtverců (LS). Mezi těmito modely lze přepínat v dolním řádku tabulky.

- V posledním řádku tabulky lze
 - v levé části nastavit požadovaný počet třídních intervalů zapsáním příslušného čísla do editačního okna a kliknutím na tlačítko *Aktivovat*,
 - v pravé části přepnout mezi způsobem odhadu parametrů Weibullova rozdělení - mezi metodou nejmenších čtverců (LS) a metodou maximální věrohodnosti (ML).
- **výpočet chí-kvadrát testu dobré shody** empirického a teoretického Weibullova rozdělení, odhadnutého buď metodou nejmenších čtverců nebo maximální věrohodnosti, podle nastavení v pravé části posledního řádku tabulky. Výsledky testu jsou zobrazeny ve střední části posledního řádku tabulky a obsahují informaci o
 - počtu stupňů volnosti (v závorce),
 - hodnotě testové statistiky a
 - odpovídající *p*-hodnotě.
- **grafické znázornění**
 - histogramu četností (třída x relativní četnost), v němž jednotlivé sloupce odpovídají barevně třídám v tabulce.
 - rozdílového histogramu (třída x rozdíl relativních četností) - zobrazuje odchylky empirických a teoretických relativních četností pro jednotlivé Weibullovské modely (ML) a (LS).

8.6.3.3. Doby mezi poruchami

Výpočet charakteristik a četnostní analýza doby mezi poruchami. Doby mezi poruchami jsou počítány vždy pro odpovídající elektrárnu a blok od doby jejího uvedení do provozu. V levé části obrazovky je **tabulka se zdrojovými daty**, vpravo nahoře je **výsledková tabulka**. Ta obsahuje různé informace, mezi kterými lze vybírat pomocí lokální nabídky:

[charakteristiky](#) | [histogram](#) | [kumulativní histogram](#)

Pod výsledkovou tabulkou jsou umístěny jednotlivé **grafy**.

V pravém dolním rohu každého grafu je ikona  pro jeho celoobrazovkové zobrazení. To lze použít, chceme-li obrázek například tisknout nebo kopírovat.

- **Tabulka se zdrojovými daty** obsahuje údaje o identifikaci poruchy z hlediska
 - *elektrárny*,
 - *bloku*,
 - *data a času vzniku poruchy*,
 - *dobou mezi poruchami* v hodinách.
- Řádky tabulky jsou uspořádány vzestupně podle doby mezi poruchami a barevně odpovídají jednotlivým třídám v tabulce četností (ve výsledkové tabulce v pravé horní části obrazovky po volbě *histogram*).

Po najetí myší na datum a čas vzniku poruchy v libovolném řádku tabulky se objeví pod kurzorem okénko (hint) s informací o datu a času ukončení poruchy.

- **Výsledková tabulka - charakteristiky** obsahuje
 - **popisné statistiky datového souboru:**
 - počet záznamů ve výběru, ze kterého jsou charakteristiky počítány
 - rozsah hodnot - rozdíl maximální a minimální hodnoty,
 - aritmetický (výběrový) průměr dob poruch ve výběru,
 - výběrový rozptyl,
 - (výběrovou) směrodatnou odchylku,
 - minimum,
 - dolní kvartil,
 - horní kvartil,
 - medián,
 - maximum.
 - **odhady parametrů Weibullova modelu**, to jest parametru měřítka (alfa) v hodinách a parametru tvaru (beta)
 - metodou maximální věrohodnosti (ML)
 - metodou nejmenších čtverců (LS)
 - **výsledky testů shody** odhadnutého Weibullova modelu a naměřených dat v aktuálním výběru. Pro ověření modelu jsou použity dvě metody:
 - Andersonův-Darlingův test a
 - Lillieforsův test.
 - U každého testu je uvedena hodnota testové statistiky a její p -hodnota.
 - **Grafy.** V nabídce je celkem 10 různých statistických a pravděpodobnostních grafů:
 - **bodový (rozptylový) graf** (datum x doba mezi poruchami).
Zobrazuje zdrojová (netransformovaná) data. Na ose x je datum vzniku poruchy, ve směru osy y je zobrazena doba mezi poruchami. Každé poruše odpovídá jeden křížek.
 - **$Q-Q$ graf - graf transformovaných hodnot.**
Na ose x je zobrazena hodnota logaritmu doba mezi poruchami ($\ln(x)$), na ose y jsou hodnoty dvojitého logaritmu převrácené hodnoty doplňku empirické distribuční funkce do jedničky ($\ln(\ln(1/(1-F_n(x))))$). Pokud zdrojová data odpovídají Weibullovu modelu, měla by všechna ležet v přímce. Odchylka od přímky znamená odchylku od teoretického Weibullova rozdělení. Do grafu jsou zakresleny přímky, odpovídající odhadnutému Weibullovu rozdělení pravděpodobnosti (červená pro odhad metodou maximální věrohodnosti a zelená pro odhad metodou nejmenších čtverců). Rovnice těchto přímek jsou uvedeny pod grafem.
 - **graf distribuční funkce** (doba mezi poruchami x pravděpodobnost)
Empirická distribuční funkce odhadnutá z napozorovaných dat je zobrazena modrou barvou. Vedle toho jsou zobrazeny i distribuční funkce Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).

- graf spolehlivostní funkce (doba mezi poruchami x pravděpodobnost)
Empirická spolehlivostní funkce odhadnutá z napozorovaných dat (pomocí empirické distribuční funkce) je zobrazena modrou barvou. Vedle toho jsou zobrazeny i funkce spolehlivosti Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).
- graf hustoty pravděpodobnosti (doba mezi poruchami x intenzita)
Hustota odhadnutá z napozorovaných dat (jako numerická derivace empirické distribuční funkce) je zobrazena modrou barvou. Vedle toho jsou zobrazeny i hustoty Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).
- graf intenzity doby poruchy (doba mezi poruchami x intenzita)
Intenzita odhadnutá z napozorovaných dat (jako podíl empirické hustoty a empirické funkce spolehlivosti) je zobrazena modrou barvou. Vedle toho jsou zobrazeny i intenzity Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).
- rozdílový graf cdf (doba mezi poruchami x pravděpodobnost)
Graf zobrazuje odchylky empirické distribuční funkce od distribuční funkce (cdf) Weibullova rozdělení (červeně pro odhad metodou maximální věrohodnosti a zeleně pro odhad metodou nejmenších čtverců).
- graf průměrné kumulativní funkce (doba mezi poruchami x relativní počet poruch)
Zobrazuje vývoj kumulativní četnosti poruch vzhledem k době mezi poruchami. Graf obsahuje pás 95% spolehlivosti pro odhad počtu poruch.
- graf intenzity trvání (doba mezi poruchami x intenzita)
Numerická derivace průměrné kumulativní funkce. Zobrazuje rychlost přírůstku kumulativního počtu poruch pro danou dobu mezi poruchami na ose x .
- **Výsledková tabulka - histogram** obsahuje četnostní analýzu dob mezi poruchami. Zahrnuje
 - **výpočet frekvenční tabulky**, to znamená
 - roztržení do tříd (počet tříd lze nastavit v dolním řádku tabulky). Třídy jsou charakterizovány pořadovým číslem, dolní a horní mezí v hodinách.
 - výpočet absolutních četností (prostých i kumulativních) pro jednotlivé třídy (počty poruch).
 - výpočet relativních četností (prostých i kumulativních) pro jednotlivé třídy (v %).
 - výpočet teoretických (prostých) relativních četností za předpokladu Weibullova rozdělení a difference mezi empirickými a teoretickými četnostmi. Výpočet je proveden pro Weibullovo rozdělení s parametry odhadnutými buď metodou maximální věrohodnosti (ML) nebo

- metodou nejmenších čtverců (LS). Mezi těmito modely lze přepínat v dolním řádku tabulky.
- V posledním řádku tabulky lze
 - v levé části nastavit požadovaný počet třídních intervalů zapsáním příslušného čísla do editačního okna a kliknutím na tlačítko *Aktivovat*,
 - v pravé části přepnout mezi způsobem odhadu parametrů Weibullova rozdělení - mezi metodou nejmenších čtverců (LS) a metodou maximální věrohodnosti (ML).
 - **výpočet chí-kvadrát testu dobré shody** empirického a teoretického Weibullova rozdělení, odhadnutého buď metodou nejmenších čtverců nebo maximální věrohodnosti, podle nastavení v pravé části posledního řádku tabulky. Výsledky testu jsou zobrazeny ve střední části posledního řádku tabulky a obsahují informaci o
 - počtu stupňů volnosti (v závorce),
 - hodnotě testové statistiky a
 - odpovídající *p*-hodnotě.
 - **grafické znázornění**
 - histogramu četností (třída x relativní četnost), v němž jednotlivé sloupce odpovídají barevně třídám v tabulce.
 - rozdílového histogramu (třída x rozdíl relativních četností) - zobrazuje odchylky empirických a teoretických relativních četností pro jednotlivé Weibullovské modely (ML) a (LS).
 - **Výsledková tabulka - kumulativní histogram** obsahuje četnostní analýzu dob trvání poruchy. Zahrnuje
 - **výpočet frekvenční tabulky**, to znamená
 - roztrídění do tříd (počet tříd lze nastavit v dolním řádku tabulky). Třídy jsou charakterizovány pořadovým číslem, dolní a horní mezí v hodinách.
 - výpočet absolutních četností (prostých i kumulativních) pro jednotlivé třídy (počty poruch).
 - výpočet relativních četností (prostých i kumulativních) pro jednotlivé třídy (v %).
 - výpočet teoretických (kumulativních) relativních četností za předpokladu Weibullova rozdělení a difference mezi empirickými a teoretickými četnostmi. Výpočet je proveden pro Weibullovo rozdělení s parametry odhadnutými buď metodou maximální věrohodnosti (ML) nebo metodou nejmenších čtverců (LS). Mezi těmito modely lze přepínat v dolním řádku tabulky.
 - V posledním řádku tabulky lze
 - v levé části nastavit požadovaný počet třídních intervalů zapsáním příslušného čísla do editačního okna a kliknutím na tlačítko *Aktivovat*,
 - v pravé části přepnout mezi způsobem odhadu parametrů Weibullova rozdělení - mezi metodou nejmenších čtverců (LS) a metodou maximální věrohodnosti (ML).

- **výpočet chí-kvadrát testu dobré shody** empirického a teoretického Weibullova rozdělení, odhadnutého buď metodou nejmenších čtverců nebo maximální věrohodnosti, podle nastavení v pravé části posledního řádku tabulky. Výsledky testu jsou zobrazeny ve střední části posledního řádku tabulky a obsahují informaci o
 - počtu stupňů volnosti (v závorce),
 - hodnotě testové statistiky a
 - odpovídající p -hodnotě.
- **grafické znázornění**
 - histogramu četností (třída x relativní četnost), v němž jednotlivé sloupce odpovídají barevně třídám v tabulce.
 - rozdílového histogramu (třída x rozdíl relativních četností) - zobrazuje odchylky empirických a teoretických relativních četností pro jednotlivé Weibullovské modely (ML) a (LS).

8.6.3.4. Rozdělení dob poruch

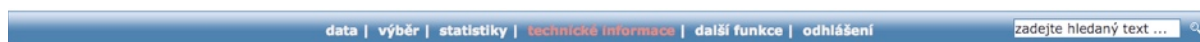
Tato část statistické analýzy záznamů o poruchách poskytuje informace o rozdělení úhrnu délek poruch podle různých hledisek: podle elektrárny, bloku, nominálního výkonu, poškozené součásti a dalších. Hledisko pro třídění lze vybrat kliknutím na název položky v pravé horní části obrazovky, pod nabídkovou lištou. Obrazovka obsahuje **tabulku úhrnů dob poruch** v jednotlivých třídách, **histogram** absolutních úhrnů a **kruhový graf** relativních úhrnů dob poruch.

- **Tabulka úhrnů dob poruch** je rozdělena do čtyř sloupců:
 - *číslo třídy*, které se objevuje i v popisu sloupců histogramu
 - *název třídy*, odpovídající hodnotám třídícího znaku,
 - *úhrn* dob poruch pro odpovídající třídu,
 - *relativní úhrn* dob poruch pro odpovídající třídu (procentuální zastoupení vzhledem k celkovému úhrnu dob poruch).
- Nad tabulkou je **okno pro nastavení počtu zobrazovaných tříd** počínaje nejvyšším úhrnem dob poruch v sestupném uspořádání. V případě, že tabulka obsahuje velký počet tříd, je tato funkce užitečná. Stačí do editačního okna napsat požadovaný počet a kliknout na tlačítko *Aplikovat*.
Jednotlivé třídy (podle hodnot vybraného znaku, například jednotlivé typy odstávky) jsou na obrazovce charakterizovány různými barvami. To umožňuje využít tabulku úhrnů dob poruch zároveň jako legendu ke grafům v pravé části obrazovky.
- **Histogram absolutních úhrnů dob poruch** (třída x absolutní úhrn doby poruch) zobrazuje absolutní úhrny dob poruch pro jednotlivé třídy pomocí sloupkového grafu. Barvy sloupců (tříd) odpovídají řádkům v tabulce úhrnů dob poruch. Po najetí myši na jednotlivé sloupce grafu se objeví malé okénko s informací o odpovídající třídě (hint).
- **Kruhový graf relativních úhrnů** zobrazuje relativní úhrny dob poruch pro jednotlivé třídy vzhledem k celkovému úhrnu dob všech poruch v aktuálním výběru. Barvy

výšečí (tříd) odpovídají řádkům v tabulce úhrnů. Po najetí myši na jednotlivé kruhové výšeče se objeví malé okénko s informací o odpovídající třídě (hint).

V pravém dolním rohu každého grafu je ikona  pro jeho celoobrazovkové zobrazení. To lze použít, chceme-li obrázek například tisknout nebo kopírovat.

8.6.4. Technické informace



8.6.4.1. Informace o turbogenerátorech

V této části programu jsou k dispozici údaje o jednotlivých turbogenerátorech. Záznamy zde lze prohlížet, upravovat, doplňovat a odebírat.

Tabulka technických informací obsahuje prvky pro editaci, vkládání a mazání záznamů a další funkce pro zobrazení. V tabulce odpovídá každému stroji jeden řádek, sloupce odpovídají vybraným údajům o stroji:

- **elektrárna** - zkratka názvu elektrárny, v níž došlo k poruše. Zkratka je uvedena podle číselníku elektráren.
- **blok** - číslo bloku elektrárny, k němuž se záznam vztahuje.
- **výkon** - nominální výkon bloku. Výkon je uveden v MWh.
- **datum uvedení bloku do provozu** ve formátu dd.mm.rrrr (den.měsíc.rok).
- **P,P** - ikony pro zobrazení poznámky () a případných příloh (), uložených na datovém serveru. Pokud záznam neobsahuje poznámku nebo přílohu, potom příslušná ikona u tohoto záznamu chybí. Kliknutím na ikonu přílohy se tato zobrazí v samostatném okně aplikace, která je s touto přílohou asociována v operačním systému (zpravidla MS Word, Excel, grafický editor a podobně).
- **sloupec s ikonami** pro editaci () a mazání () záznamů. Kliknutím na ikonu  v záhlaví sloupce lze provést export dat z tabulky do textového souboru, v němž jsou jednotlivé položky odděleny středníkem (tzv. "csv" soubor). Taskovýto soubor lze prohlížet přímo v tabulkovém editoru MS EXCEL.

První řádek tabulky obsahuje názvy sloupců. Kliknutím myši na název některého z prvních čtyř sloupců provedete seřazení záznamů vzestupně nebo sestupně podle hodnot tohoto sloupce. První kliknutí seřadí vzestupně, druhé sestupně a třetí toto seřazení ruší. Po kliknutí se na obrazovce zpravidla objeví hlášení "... prosím vyčkejte, server zpracovává Váš dotaz ...". Toto hlášení se objevuje vždy, když jsou data na serveru zpracovávána a klient (Váš prohlížeč) čeká na odpověď. Seřazení je indikováno ikonou ▲ nebo ▼ u názvu sloupce.

Poslední řádek tabulky v pravém sloupci poskytuje možnost [\[přidat záznam\]](#). Kliknutím na tato slova otevřete editační formulář pro vytvoření nového záznamu.

8.6.4.2. Editace záznamu

Editaci (úpravu, změnu) stávajícího záznamu vyvoláte kliknutím na ikonu pro editaci  v pravém sloupci na odpovídajícím řádku tabulky. Po tomto výběru se na obrazovce objeví formulář pro editaci. Tento formulář je stejný jak pro vstup nového záznamu, tak i pro editaci stávajících. Formulář je rozdělen do tří částí:

- v první části formuláře zadáváte údaje, které jsou součástí záznamu. Tyto údaje se vkládají obvyklým způsobem do vstupních polí buď přímo z klávesnice nebo výběrem z roletkové nabídky (elektrárna).
- druhá část formuláře obsahuje tlačítka pro uložení záznamu ([odeslat]), ukončení editace bez uložení ([zrušit]) a doplnění poznámky ([poznámka]). Poznámka se zadává do samostatného okna s editorem textu, umožňujícím základní formátování (Rich Text Editor).
- ve třetí části formuláře lze zadat údaje o přílohách. Informace o přidružené příloze má dvě části: název přílohy (v levém sloupci) a odkaz na soubor s přílohou (pravý sloupec). Odkaz na přílohu se vkládá výběrem souboru z disku po stisku tlačítka [Procházet]. Máte-li výběr přílohy dokončen, do databáze ji vložíte stiskem tlačítka [přidat]

8.6.4.3. Vložení nového záznamu

Nový záznam lze vložit použitím funkce [přidat záznam], jejíž odkaz je v pravém dolním rohu tabulky (v posledním řádku). Po kliknutí na text [přidat záznam] se objeví formulář pro vstup nového záznamu. Tento formulář je stejný jako pro editaci stávajících záznamů.

8.6.4.4. Zrušení záznamu

Pro zrušení záznamu slouží ikona . Kliknutím myši na tuto ikonu budete vyzváni k potvrzení úmyslu zrušit daný záznam. Teprve po tomto potvrzení dojde k jeho vyřazení z tabulky a tabulka se znovu překreslí.

8.6.5. Další funkce



8.6.5.1. Prohlížení a editace číselníků

Řada položek obsahuje hodnoty, které se opakují a v databázi jsou vyjádřeny pomocí svých číselných kódů. Převod mezi těmito kódy a jim odpovídajícími textovými hodnotami zprostředkovávají takzvané "číselníky". Pro běžného uživatele obsahují seznam hodnot, kterých mohou některé položky nabývat. Tak například druh záznamu může nabývat hodnot "porucha" (zkratkou PO), "reklamace" (RE), "hlášení o neshodě" (HN) a "protokol o přejímce" (PR). Tyto texty nalézne běžný uživatel v číselníku druhu záznamu. Uživatel s právem správce dat může tyto číselníky navíc doplňovat o další hodnoty a upravovat.

Po volbě funkce **číselníky** z podnabídky dalších funkcí lze vybrat z roletové nabídky některý z číselníků a zobrazit jej. Přehled použitých číselníků:

- druh dokumentu (CISDRD)
- zkratky a názvy elektráren (CISELNA)
- označení bloků (CISBLK)
- označení nominálních výkonů bloků (CISNVB)
- evidované příčiny poruch (CISPRP)
- projevy poruch (CISPRO)
- kódy zavinění (CISKZA)
- počátek poruchy (CISZAV)
- únik média (CISUME)
- způsoby odstavení (CISZPO)
- typy odstávky (CISTOD)
- způsoby opravy (CISZOP)
- druhy kombinované opravy (CISKOM)
- číselník součástí turbosoustrojí (CISOUC)

Záznamy v číselnících lze samozřejmě i editovat, mazat či doplňovat, pokud k tomu máte příslušné oprávnění.

8.6.5.2. Vlastní dotazy SQL

Uživatelé, kteří mají práva administrátora, mohou pracovat s daty přímo pomocí příkazů SQL (Structured Query Language). Program dovoluje přímo vkládat příkazy DML (Data Manipulation Language), jako je *select*, *insert*, *delete*.

8.6.5.3. Správa uživatelů

Program dovoluje rozlišovat různé uživatele podle práv přístupu k datům. Tato práva přiděluje administrátor programu a tato funkce mu dovolí přidávat nové uživatele, editovat jejich údaje, měnit hesla a přístupová práva, ale také uživatele rušit. V případě, že zapomenete přístupové heslo, administrátor může právě zde vaše heslo resetovat, což znamená nastavit jej pro další přihlášení.

Přístupová práva mohou být odstupňována dle požadavků provozovatele programu. Příkladem může být obvyklé třístupňové nastavení práv, kdy běžný uživatel má právo pouze data prohlížet, provádět výběry a statistické odhady, oprávnění uživatelé mají právo zápisu a editace záznamů, výjimečně rušení záznamů. Administrátor potom může vkládat, editovat i rušit záznamy, včetně záznamů v číselnících, provádět zálohu databáze, vést evidenci uživatelů a další činnosti.

8.6.5.4. Nápověda

Vývolá nápovědu ve formátu HTML.

8.6.6. Odhlášení



Po ukončení práce s databází poruch by se měl každý uživatel odhlásit. Chceme-li ukončit práci s programem, zvolíme tuto funkci z hlavní nabídky. Na obrazovce se objeví následující text:

Jste odhlášen(a)! Pro nové přihlášení pokračujte ... [zde](#).

Pokud se chcete vrátit do programu, klikněte myší na slovo [zde](#) a dostanete možnost se znovu přihlásit do programu.

9. Zhodnocení funkčnosti aplikace

Aplikace je dokončena a zadavatel ji provozuje na svém intranetovém serveru v Plzni. Byli implementovány všechny požadované prvky. Aplikace byla oproti předchozí verzi vylepšena především:

- zrychlením odezvy rozhraní i databázové vrstvy
- zpřehledněním GUI
- převedením na aplikační platformu nezávislou na operačním systému
- doplněním o nové statistické funkce
- možnosti exportu dat
- rozšířením možností výběru
- přidáním administračních nástrojů
- lepší distribuce informací v rámci interní sítě zákazníka
- jednodušší udržitelnost systému
- zvýšení bezpečnosti systému

Spolupráce se Škoda Plzeň a.s. i nadále pokračuje. Aplikace by měla být rozšiřována o funkce importu dat od provozovatelů TG Škoda a navíc by měla proběhnout úprava uživatelského prostředí.

10. Závěr a zhodnocení práce

Tato práce mi především dala možnost praktické aplikace softwarového inženýrství v rámci spolupráce s větším zákazníkem. Tento systém má již třetí přepracování. Mohl jsem tedy pozorovat evoluci v programovacích nástrojích. Od textově orientované aplikace vytvořené systémem PC Fand, přes MS Windows platformu až k moderní třívrstvé multiplatformní aplikaci. Zároveň jsem si vyzkoušel netradiční formu SI - extrémní programování. Účastnil jsem se mnoha sezení se zákazníkem a mohl se tak aktivně podílet při formulování požadavků. Uplatnil jsem zde především znalosti z oboru softwarového inženýrství a databázových systémů. K testování systému byli použity metody z předmětu “Styk člověka s počítačem”.

Chtěl bych tímto poděkovat svému vedoucímu práce Ing. Ivanu Halaškovi a především Ing. Václavu Černému, který byl přínosem pro vývojový tým a to především jako tester a analytik. Na závěr bych rád poděkoval i celé své rodině a přítelkyni za trpělivost v období psaní této práce.

11. Použité zdroje

- [1] Richta, Sochor: Softwarové inženýrství I. Skripta FEL ČVUT, Praha 1998
- [2] Pokorný, J., Halaška, I.: Databázové systémy. Vydavatelství ČVUT Praha, 1998
- [3] Václav Kadlec: Agilní programování-metodiky efektivního vývoje softwaru. Computer Press. Brno. 2004. ISBN 80-251-0342-0.
- [4] Jiří Kosek: PHP - tvorba interaktivních internetových aplikací, Grada Publishing, s.r.o. 1998
- [5] Jan Adam Strouhal - Disertační práce "Spolehlivostní modely při tvorbě softwaru", ČVUT FS, Praha. 2007
- [6] Session v PHP [online]. <<http://interval.cz/clanky/pracujeme-se-session-v-php/>>
- [7] Dušan Janovský. Jak psát web, příručka html, css a javascriptu [online] <<http://www.jakpsatweb.cz>>
- [8] Manuál php [online] <<http://cz.php.net/manual/cs/>>
- [9] Interval.cz [online]. <<http://interval.cz/vyvoj-aplikaci/ajax/>>
- [10] W3.org, specifikace xhtml 1.0 [online] <<http://www.w3.org/TR/xhtml1/#xhtml>>
- [11] Wikipedia.org, xmlhttprequest [online] <<http://en.wikipedia.org/wiki/XMLHttpRequest>>
- [12] Wikipedia.org, HTTP cookie [online] <http://cs.wikipedia.org/wiki/HTTP_cookie>
- [13] Wikipedia.org, php [online] <<http://cs.wikipedia.org/wiki/PHP>>
- [14] Wikipedia.org, wysiwyg [online] <<http://cs.wikipedia.org/wiki/WYSIWYG>>
- [15] The Agile Manifesto <<http://www.agile-manifesto.org>>. 2001
- [16] JQuery [online]. <<http://jquery.com/>>
- [17] Extrémní programování [online]. <<http://www.extremeprogramming.org>>
- [18] Styk člověka s počítačem [online] <<http://service.felk.cvut.cz/courses/X36SCP>>