

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Použití architektury klient-server pro vzdálené spouštění numerických modelů

Diplomová práce

Autor: Bc. **David Chvalkovský**

Vedoucí DP práce: Ing. Otto Severýn, Ph.D.

Konzultant: Ing. Dalibor Frydrych, Ph.D.

V Liberci 28. 5. 2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David CHVALKOVSKÝ**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Informační technologie**

Název tématu: **Použití architektury klient-server pro vzdálené spouštění numerických modelů.**

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


1. Seznamte se s numerickými modely podzemních procesů používaných a vyvíjených na NTI, jejich vstupními a výstupními datovými soubory a způsobem řízení jejich běhu.
2. Navrhněte síťovou aplikaci v architektuře klient-server, která umožní uživateli komfortním způsobem spouštět numerické modely na vzdáleném počítači.
3. Navrženou aplikaci implementujte a otestujte.
4. Získané poznatky zdokumentujte v technické zprávě.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: cca 40 až 50 stran
Forma zpracování diplomové práce: tištěná

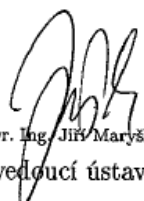
Seznam odborné literatury:

- [1] Maryška J., Severýn O., Tauchman M., Tondr D.: Modelling of the groundwater flow in fractured rock - a new approach, In Proceedings of Algoritmy 2005 (K. Mikula, ed.), Slovak Technical University, Bratislava, pp.113-122
- [2] Eckel B.: Myslíme v jazyku Java. Grada, Praha, 2007.

Vedoucí diplomové práce: **doc. Ing. Otto Severýn, Ph.D.**
Ústav nových technologií a aplikované informatiky
Konzultant diplomové práce: **doc. Ing. Dalibor Frydrych, Ph.D.**
Ústav nových technologií a aplikované informatiky
Datum zadání diplomové práce: **31. října 2008**
Termín odevzdání diplomové práce: **29. května 2009**


prof. Ing. Václav Kopecký, CSc.
děkan




prof. Dr. Ing. Jiří Maryška, CSc.
vedoucí ústavu

V Liberci dne 31. října 2008

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon číslo 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum 28. 5. 2009

Podpis

Abstrakt

Tato práce se zabývá spouštěním programů na dálku pomocí architektury klient-server. Cílem práce bylo vytvoření webové aplikace, která bude zajišťovat správu vstupních a výstupních souborů programů, spouštění, nastavení a administraci programů. Webová aplikace s názvem RemStat, je určena pro práci s programy pro modelování podzemních procesů a transportu látek, které jsou vyvíjeny na ústavu NTI. Tato aplikace slouží především pro jednoduchý přístup k těmto programům.

V rámci této práce jsou popsány programy GENFLOW a Flow123D pro výpočet modelů, použité webové technologie Javy EE, jejího frameworku Struts 2, serveru Apache Tomcat a další.

Klíčová slova: Webová aplikace, Java, Struts 2, vzdálené spouštění programů, modely podzemního proudění a transportu látek.

Abstract

This work dealt with the launching of remote programs with the architecture of the client-server. The aim of the work was to create application, which provides administrative input and output program files and launching, setting and administrating programs. The web application with the name RemStat is designed for working with programs for modeling underground processes and transport materials, which is being developed at the NTI institute. The application provides easy access to these programs.

The programs GENFLOW and Flow123D are described in the diploma work. They are used for computation of the models. Java EE web technologies, framework Struts 2, server Apache Tomcat and others are described and used for creating of the web application.

Keywords: Web application, Java, Struts 2, launching of remote programs, models, underground processes and transport of material.

Obsah

Zadání diplomové práce	Chyba! Záložka není definována.
Prohlášení	3
Abstrakt	4
Abstract.....	4
Obsah	5
Úvod	7
1 Modelování podzemního proudění.....	8
1.1 Programy vyvíjené na ústavu NTI	8
1.1.1 Program Flow123d.....	9
1.1.2 Program GENFLOW	9
2 Webové technologie	10
2.1 Webová aplikace	10
2.1.1 Architektura klient-server	10
2.1.2 HTML a CSS	11
2.1.3 JavaScript.....	11
2.1.4 Databáze MySQL.....	11
2.2 Java EE.....	12
2.2.1 Java Servlet	12
2.2.2 Soubor JSP	13
2.2.3 JavaBeans.....	13
2.2.4 Vývojová prostředí Javy	14
2.2.5 Server Apache Tomcat.....	14
2.3 Struts 2 framework.....	15
2.3.1 MVC model	15
2.3.2 Mapování Struts 2 akcí	15
2.3.3 Srovnání webových frameworků Javy.....	17
3 Spuštění aplikace.....	18
3.1 Přihlášení do aplikace	18
3.2 Hlavička a menu stránek.....	19
3.3 Práce s projekty.....	20
3.3.1 Seznam projektů.....	20
3.3.2 Editace souborů projektu	21
3.3.3 Spuštění projektu, výpis výsledků	22
3.4 Nastavení programů	24
3.5 Nastavení uživatelů.....	25
3.5.1 Výpis uživatelů	25
3.5.2 Založení a editace uživatele	26
3.6 Spuštění webové aplikace na serveru	27
3.6.2 Soubor RemStat.war	28
3.6.3 Spuštění aplikace	28
3.6.4 Bezpečnostní rizika.....	29
4 Popis aplikace	30
4.1 Adresář data	30
4.1.1 Jádro aplikace.....	31
4.1.2 JavaBeany pro přenos dat v rámci aplikace	31
4.1.3 Přístup k datům	32
4.1.4 Spuštění výpočtů.....	33

4.1.5 Vlákno pro výpočet.....	34
4.1.6 Třída ServletContextListener	34
4.2 Základní rozdělení akcí.....	34
4.3 Uživatelské akce	35
4.4 Akce projektů.....	36
4.4.1 Výpis projektů.....	36
4.4.2 Založení nového projektu	36
4.4.3 Úprava projektů, načtení souborů	37
4.4.4 Spouštění programů	37
4.4.5 Posílání souborů.....	38
4.5 Nastavení.....	38
4.5.1 Nastavení programů	39
4.5.2 Nastavení uživatelů	39
4.6 Další části projektu	39
4.6.1 Pomocné JSP soubory.....	39
4.6.2 Pomocné soubory pro generování HTML stránek.....	40
4.6.3 Příložené programy a jejich data v projektech.....	40
4.7 Nastavení aplikace	41
4.7.1 Soubory s popiskami.....	41
4.7.2 Nastavení programů a projektů.....	41
4.7.3 Soubory s nastavením aplikace.....	42
4.7.4 Použité knihovny Javy	42
Závěr	44
Použitá literatura	45
Přílohy	46
Příloha A – Adresářová struktura aplikace	46
Příloha B - Nastavení programů aplikace	47
Příloha C - Nastavení Struts 2.....	48
Příloha D - Nastavení souboru web.xml	49
Příloha E – Nastavení souboru context.xml.....	50

Úvod

S rozvojem komunikačních technologií a to zejména s rozvojem celosvětové počítačové sítě Internet, vzniká mnoho možností pro jejich využití. V dnešní době je Internet dostupný každému a rychlost připojení narůstá každým rokem nebývalým tempem. Firmy investují nemalé prostředky do rozvoje těchto technologií, protože vědí, že se jim tyto náklady brzy vrátí. Také školy si toto uvědomují, zapojují se do vývoje nových technologií, tvoří páteř Internetu a samy jej využívají pro prezentaci svých výsledků, práce studentů a zejména pro komunikaci s okolím.

Na Technické univerzitě v Liberci se vyvíjí velké množství užitečných programů, které nejsou v některých případech dostupné, tak jak bychom si přáli. Jsou různé možnosti, jak právě tyto programy poskytnout běžnému uživateli, avšak nejdostupnějším a nejrychlejším přístupem je právě internet a to zejména webové stránky univerzity, kam denně chodí studenti, učitelé, zaměstnanci a další návštěvníci.

Za tímto účelem vznikla webová aplikace RemStat, která slouží zejména pro vzdálené spouštění programů, které slouží pro modelování podzemního proudění. Úlohy řešené v těchto programech se vyznačují relativně velkou náročností výpočtu, proto je výhodnější pro tyto úlohy vyčlenit server k tomu vybavený. Pomocí aplikace RemStat je možné spouštět programy spustitelné pomocí příkazové řádky, které nevyžadují grafické rozhraní. S těmito možnostmi lze zpřístupnit mnoho různých programů, ale na druhé straně je zapotřebí zajistit zabezpečení proti neoprávněnému přístupu, zneužití prostředků serveru nebo proti neoprávněné manipulaci s daty.

Obsah této diplomové práce je rozdělen do čtyř hlavních kapitol. V první kapitole je shrnut popis programů vyvíjených na ústavu NTI. V druhé kapitole jsou rozepsány použité webové technologie jakými je framework Javy Struts 2, platforma Javy EE a další webové technologie, které byly použity pro vývoj webové aplikace s názvem RemStat, která byla vytvořena v rámci této diplomové práce.

Třetí kapitola popisuje vytvořené webové stránky aplikace RemStat, spuštění webové aplikace na serveru Apache Tomcat a ve vývojovém prostředí Eclipse nebo NetBeans a její nastavení. Poslední kapitola popisuje strukturu vytvořených tříd Javy pro zprávu dat, práci s programy a prezentační logiku vytvořenou za pomoci již zmíněných webových technologií.

1 Modelování podzemního proudění

Modelování podzemního proudění pomocí počítačových modelů umožňuje předpovídat směr a pohyb kapalin (podzemní vody), různých látek (ropy) a plynů (zemní plyn), které probíhá v různých typech podzemního prostředí a za různých podmínek. Při podzemním proudění může docházet k transportu, který je označován jako přenos částic ve vodním prostředí, ve vzduchu vlivem větru atd. Proudění také může probíhat bez účasti transportu, stejně tak jako transport bez účasti proudění.

Modely proudění a transportu látek slouží jako podklad pro zjištění rizika kontaminace spodních vod, pro stanovení velikosti ochranného pásma okolo vodních zdrojů, pro odhad škodlivosti úložišť nebezpečných odpadů, k zjištění vzájemného ovlivnění studní, vrtů, k zjištění průsaků hrází, k vybudování protipovodňové ochrany, při těžbě surovin a k dalším neméně důležitým výpočtům.

Výsledky těchto modelů slouží pro efektivnější práci, šetří vynaložené prostředky a odhalují možná rizika, která mohou nastat (viz [9]).

1.1 Programy vyvíjené na ústavu NTI

Na ústavu NTI se dlouhodobě vyvíjí programy pro modelování podzemního proudění. Mezi tyto programy patří program FLOW123D a donedávna i program GENFLOW, který se již přestal vyvíjet. Vstupem těchto programů je sada souborů, které popisují numerický model dané úlohy. Mezi tyto soubory patří například soubor elementů, okrajových podmínek, materiálových vlastností a další. Stejně tak výstupem je sada souborů, kam patří soubor příponou .log, ve kterém je vypsán poslední průběh dané úlohy (viz [6]).

Časová náročnost úloh řešených v těchto programech pro běžný počítač může být od těch zkušebních trvajících několik milisekund po ty nejnáročnější, které trvají v řádu několika minut, hodin, dnů či týdnů. Pro tyto úlohy v praxi většinou platí kvadratická závislost v čase na počtu použitých elementů. S tím souvisí i odhad pracnosti úlohy, se kterou by měl zadavatel počítat.

1.1.1 Program Flow123d

Program Flow123d slouží pro simulaci proudění podzemních vod v saturovaném prostředí. Tento program byl navržen a zkonstruován pro úlohy zaměřující se na problematiku uskladnění trvale vyhořelého jaderného paliva. Flow123d pracuje s modely založenými na kombinovaných sítích složených z 3D, 2D a 1D elementech (viz [6]).

Sadu vstupních souborů programu Flow123D potřebných k výpočtu tvoří soubor elementů MSH, materiálových vlastností MTR, soubor okrajových podmínek BCD, sousedství NGH a definice úlohy INI. Výsledkem výpočtu jsou soubory s příponou *pos*, které obsahují výsledné okrajové podmínky a materiálové vlastnosti zadaných elementů. Průběh výpočtu je zaznamenán v souboru s příponou *log* (viz [9]).

1.1.2 Program GENFLOW

GENFLOW je program sloužící pro modelování podzemního proudění, který pracuje s uzavřenou sítí prizmatických elementů, což je trojboký hranol se skloněnými podstavami (viz [6]).

Stejně jako program Flow123D obsahuje program GENFLOW také sadu vstupních souborů s podobným obsahem, ale v jiném formátu. Tato sada souborů popisuje model podzemního proudění a skládá se ze souboru definic sítě multielementů STE, sítě uzlů STU, materiálových vlastností MTR, okrajových podmínek OKE a dalších souborů. Výsledkem výpočtu numerického modelu je soubor vektorů, soubor skalárních veličin a logovací soubor (viz [9]).

2 Webové technologie

Základem diplomové práce je vytvoření webového serveru, který bude generovat dynamický obsah HTML stránek a zajišťovat spouštění programů pro modelování podzemních procesů.

K dynamickému vytváření obsahu stránek se používá celá řada technologií jakými je PHP, Perl, JSP, ASP, ASP .NET a další, které na základě požadavku klienta ze shromážděných dat (ze souboru, databáze nebo z jiných zdrojů), poskytují přístup klientovi, prostřednictvím prohlížeče. Tyto technologie většinou využívají MVC model pro jednodušší vývoj webových aplikací.

Obsah HTML stránek je obvykle doplněn o skriptovací jazyk, který se provádí na straně klienta. Nejznámějším skriptovacím jazykem je JavaScript, který doplňuje výslednou stránku o řadu dynamických prvků. Dále se využívá kaskádových stylů CSS, které popisují způsob zobrazení výsledných stránek

V této kapitole je dále popsána programovací platforma *Enterprise Editon* jazyka Javy, vývojové prostředí Eclipse a NetBeans, server Apache Tomcat, databáze MySQL, webový framework Javy Struts 2 a další webové technologie jakými jsou JavaBeany, Servlety, JSP soubory a další.

2.1 Webová aplikace

Je taková aplikace, která je poskytována uživatelům z webového serveru přes počítačovou síť Internet. Taková to aplikace je především oblíbená pro svoji jednoduchost a dostupnost na Internetu.

Webové aplikace generují dynamicky obsah webových stránek ve formátu HTML. Takto vytvořené webové stránky se používají například pro vlastní prezentaci podniků, škol nebo také internetové obchody, bankovníctví, sociální sítě, atd. (viz [10]).

2.1.1 Architektura klient-server

Architektura klient-server je síťová architektura, kde klienti (často aplikace s grafickým uživatelským rozhraním) komunikují se serverem, který je umístěn na vzdáleném počítači. Server poskytuje služby, které klient na požádání využívá. Klasickým příkladem je prohlížení webových stránek, kde uživatel přes webový

prohlížeč pošle požadavek serveru na vykreslení stránky a ten obratem pošle data pro vykreslení stránky v prohlížeči (viz [10]).

Charakteristika serveru:

1. Pasivní
2. Čeká na požadavky klienta
3. Při přijetí požadavku jej obslouží

Charakteristika klienta:

1. Aktivní
2. Posílá požadavky serveru
3. Čeká na odpovědi

2.1.2 HTML a CSS

HyperText Markup Language je označován zkratkou HTML, jedná se o značkovací jazyk pro hypertext. Používá se pro publikaci dokumentů www stránek. Jako popis způsobu zobrazení HTML stránek se používají kaskádové styly označované zkratkou CSS. Hlavním přínosem kaskádových stylů je oddělení vzhledu dokumentu od jeho obsahu a struktury (viz [10]).

2.1.3 JavaScript

JavaScript je nejčastěji používaný skriptovací programovací jazyk, který se zapisuje pomocí speciálních tagů přímo do HTML kódu. JavaScript se spouští většinou až po stažení stránky z internetu na straně klienta, čímž se šetří prostředky serveru, ale také z toho plynou bezpečnostní omezení (JavaScript nemůže pracovat se soubory, aby tím neohrozil klienta) (viz [10]).

2.1.4 Databáze MySQL

Zdrojem dat webové aplikace můžou být soubory na disku, databáze nebo jiný vnější zdroj. Nejčastěji se pak využívají pro ukládání dat databáze, mezi ty nejznámější patří MySQL, PostgreSQL, Oracle, Microsoft SQL Server a další. Jak většinou názvy

napovídají probíhá zde komunikace pomocí jazyka SQL. Nejvíce používaná databáze pro tvorbu webových aplikací je MySQL a to zejména díky své implementovatelnosti na různých operačních systémech, výkonnosti a dostupnosti. Tato databáze je optimalizovaná na rychlost, ale na druhou stranu tolik nepodporuje zálohování a některé funkce jakými jsou pohledy, trigger a uložené procedury.

2.2 Java EE

Java je open source objektově orientovaný programovací jazyk vyvinutý firmou Sun Microsystems, který byl představen v roce 1995. Java je jedním z nejrozšířenějších programovacích jazyků na světě. Díky své přenositelnosti nezáleží na jakém operačním systému je program vytvořen nebo spuštěn. Java je také jednoduchý, distribuovaný, robustní, bezpečný, dynamický a elegantní programovací jazyk. Její nevýhodou může být pomalejší start programů, protože je zapotřebí nejprve před spuštěním tyto programy přeložit z takzvaného mezikódu.

Existují různé platformy Javy uzpůsobené pro práci například s mobilními telefony a s různými zabudování zařízeními (Java ME) nebo s klasickými počítači (Java SE) a nebo určenou pro provoz podnikových aplikací a informačních systémů (Java EE).

Java Enterprise Edition (Java EE nebo také J2EE) je uzpůsobena zejména pro vývoj webových aplikací. Od standardní edice Javy se liší v přidáných knihovnách, obsahuje především JSP (Java Server Pages), Servlety, EJB (Java Enterprise Beans) a další technologie pro web (viz [10]).

2.2.1 Java Servlet

Servlety jsou objekty v programovacím jazyce Java, které dynamicky zpracovávají požadavky a odpovědi (requests, responses) webového serveru. Generovaný obsah je ve formátu HTML, ale může být např. i ve formátu XML. Servlety jsou protipól k dynamickým webovým technologiím jakými jsou PHP, ASP .NET a dalším, kde se neprogramuje v Javě. Servlety pracují se *session* (relace, trvající síťové spojení mezi klientem a serverem zahrnující výměnu paketů), http cookie (malé množství dat, které server pošle prohlížeči a ten si je uloží na počítači) a s vyhodnocováním, přesměrováním URL adres (viz [7]).

Servlet API (rozhraní pro programování aplikací) obsahuje balíček Java tříd *javax.servlet.**; definující interakci mezi webovým zásobníkem (kontainer) a servlety. Webový kontainer je zodpovědný za zprávu Servletů, mapování URL adres a za zprávu požadavků klienta.

2.2.2 Soubor JSP

Java ServletPages slouží jako rozšíření technologie Java Servlet. Jsou to soubory s příponou *.jsp*, které obsahují kombinaci statických HTML tagů, textů, scripletů a speciálních tagů, které zajišťují generování dynamického obsahu webových stránek.

JSP soubory se před použitím nejprve zpracují a převedou na soubory s příponou *.java*, které kompilátor Javy zkompileje do zdrojového souboru s příponou *.class*. Takto zkompilevané soubory slouží pro rychlejší opětovné spouštění(viz [7]).

2.2.3 JavaBeans

JavaBeans jsou třídy, které zapouzdřují více objektů do jednoho (bean). Zpravidla obsahují konstruktor bez argumentů, rozhraní pro serializaci nebo také rozhraní pro účely třídění (Comparable). JavaBean obsahuje sadu členských proměnných, ke kterým se přistupuje jednotlivě pomocí přístupových metod, takzvaných gettrů a settrů.

Příklad JavaBeanu:

```
public class ProjectDTO implements Comparable<ProjectDTO>,
    Serializable
{
    private String      projectName;
    private Date        start;
    private boolean     running;
    private List<FileDTO> files;

    public int compareTo(ProjectDTO project)
    {
        return this.projectName.compareTo(project.getProjectName());
    }

    public String getProjectName() {
        return projectName;
    }

    public void setProjectName(String projectName) {
        this.projectName = projectName;
    }

    public boolean isRunning () {
```

```

        return running;
    }

    public void setRunning(boolean running) {
        this.running = running;
    }
    ...
}

```

Na příkladu je vidět upravená třída z aplikace RemStat s názvem *ProjectDTO*, která v sobě obsahuje proměnné řetězec, datum, logický typ a kolekci JavaBeans, které nemají přístup zvenčí (privátní proměnné), ale pomocí přístupových metod, které jsou vidět zvenčí (viz [10]).

2.2.4 Vývojová prostředí Javy

Mezi nejvíce používané vývojové prostředí Javy patří Eclipse a NetBeans, které patří pod open source projekty. NetBeans je podporován firmou Sun Microsystems a Eclipse projekt vznikl uvolněním kódu IBM. Obě platformy podporují kromě Javy další programovací jazyky, mezi které patří C++, PHP a další.

V základní verzi obsahuje Eclipse pouze integrované prostředky pro vývoj Javy, které se dají rozšířit pomocí pluginů o další programovací jazyky nebo o návrh UML, psaní XML a další.

Oproti tomu NetBeans v základu obsahuje moduly, které se dají dále rozšiřovat. Tyto moduly zjednodušují práci s webovými aplikacemi, frameworkem Struts nebo s webovými službami a další (viz [10]).

2.2.5 Server Apache Tomcat

Apache Tomcat je Servlet/ JSP kontejner, který implementuje specifikaci Servletů a JSP. Obsahuje také další nástroje umožňující vývoj a nasazení webových aplikací a dalších webových služeb. Apache Tomcat je volně ke stažení na stránkách <http://tomcat.apache.org/>, kde poslední verze je 6.0.18 (viz [17]).

Po stažení a nainstalování je možné si jeho funkčnost otestovat na adrese <http://localhost:8080/>. Instalace Apache Tomcatu obsahuje tyto důležité adresáře *bin* pro spouštění serveru, adresář *conf*, kde se nacházejí soubory pro jeho konfiguraci (nejdůležitějším souborem je *server.xml* pro konfiguraci kontejneru), dále adresář kam

se ukládají logovací soubory *log* a adresář *webapps* pro ukládání webových aplikací (viz [12]). Apache Tomcat je v některých případech součástí vývojových prostředí Javy.

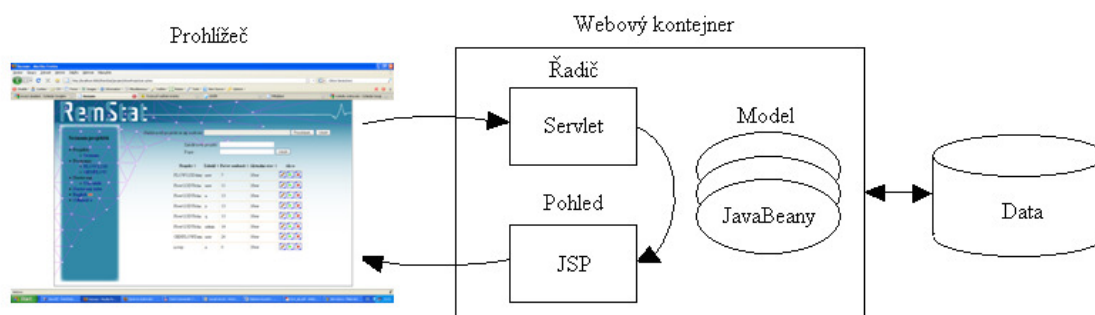
2.3 Struts 2 framework

Framework nebo-li rámec Jakarta Struts 2 je projekt Apache s volně přístupným kódem. Rámec Struts 2 slouží pro vytváření webových aplikací založených na technologiích Java Servlet a JSP. Dále podporuje vývoj aplikací založených na architektuře návrhového vzoru MVC (model-pohled-řadič).

Rámec Struts obsahuje mnoho tříd a komponent Javy pro vytváření webových aplikací. Definuje, jakým způsobem tyto třídy budou spolupracovat na řešení problému a jak bude vypadat organizace uvnitř aplikace (viz [7]).

2.3.1 MVC model

U návrhového vzoru MVC (model-pohled-řadič) je požadavek klienta (prohlížeč) nejprve zachycen řadičem, který má na starosti zpracování požadavku, ověření pravosti, autorizaci a poté určuje, která stránka (pohled) se zobrazí jako další, tak jak je tomu na obrázku 1. Za zprávu vnitřní struktury zodpovídá model, který je naprogramován v Javě a data se zde předávají pomocí speciálních tříd s názvem JavaBeans (viz [7]).



Obrázek 1: Architektura MVC modelu.

2.3.2 Mapování Struts 2 akcí

Úkolem řadiče v modelu MVC je vytváření instancí akčních tříd a rozdělovat jim práci. Pomocí XML souborů s názvem *struts* nebo k němu přidružených XML souborů se mapují všechny potřebné akční třídy na požadovanou URL adresu (například

../RemStat/project/allProjects.action) daných akcí. Je zvykem, že všechny akce končí v URL adrese koncovkou *.action*, což usnadňuje jejich přiřazení rámci Struts 2, jinak by se akce museli nastavovat přímo v souboru *web.xml* (viz [7]).

Příklad mapování akce v XML souboru:

```
<package name="project" namespace="/project" extends="struts-default">
  <action name="getAllProjects" method="execute"
    class="remstat.actions.project.ProjectAction">

    <result name="success"/>jsp/project/project.jsp</result>
  </action>
</package>
```

Na příkladu je vidět akce s názvem *getAllProjects*, která se nalézá v balíčku *project*, který definuje některé její vlastnosti. Každá akce má svůj název, kterým se volá například pomocí URL adresy. Takto definovaná akce zpravidla obsahuje akční třídu, metodu akční třídy a výsledek *result*. Výsledek akce většinou reprezentuje soubor JSP, HTML soubor, obrázek, graf, či jiná akce atd.

Po spuštění akce *getAllProjects* se vytvoří instance třídy *ProjectAction* a zavolá se metoda s názvem *execute*, která načte data do kolekce a vrátí výsledek *SUCCESS*.

Data z akční třídy se předávají pomocí *gettrů* a *settrů* do JSP stránky. Tímto způsobem je možné načítat i další objekty, jakými jsou jednoduché datové typy, JavaBeans, různé kolekce dat a další.

Příklad akční třídy:

```
public class ProjectAction extends ActionSupport
{
    private ProjectDTO project;
    private List<ProjectDTO> projectList;
    private IProjectHandlerDAO projectHandler = new ProjectHandler();

    public String execute()
    {
        projectList = projectHandler.getAllProjects();
        return SUCCESS;
    }

    public List<ProjectDTO> getProjectList() {
        return projectList;
    }
}
```

Na příkladu je vidět akční třída s názvem *ProjectAction*, která po zavolání metody s názvem *execute*, naplní kolekci dat s názvem *projectList* a umožní jej načíst do JSP souboru, který je mapován jako výsledek akce v XML souboru.

Příklad výsledného JSP souboru:

```
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
    ...
    <s:form action="allProjects">
        <s:iterator value="projectList">
            <s:property value="projectName" />
        </s:iterator>
    </s:form>
    ...
</html>
```

JSP soubor *project* je definován jako výsledek akce *getAllProjects*. Struts 2 v JSP souborech umožňuje použití speciálních tagů (prefix *s*), který je definován na prvním řádku příkladu. Dále je zde formulář, který slouží pro odesílání dat zpět akční třídě a tag *iterator* sloužící k vypísání záznamů z kolekce (v tomto případě jmen projektů).

2.3.3 Srovnání webových frameworků Javy

Framework Struts 2 vychází z nejvíce používaného frameworku Javy Struts 1. Struts 2 přináší spoustu vylepšení a hlavně se vyhýbá chybám svých předchůdců. Struts 2 obsahuje zjednodušené akce, zrušení *ActionFormů*, využívá jazyka OGNL (přenos dat v rámci JSP souboru), lepší typovou konverzi, testování, lépe nastavené implicitní hodnoty, vylepšené značky (tagy v JSP souborech), anotaci, zjednodušenou tvorbu doplňků pomocí knihoven Javy, jednoduchou integraci do Springu, využívá technologie AJAX (například validaci na straně klienta) a také zkrácení některých názvů používaných ve Struts 1.

Struts 2 převyšuje Struts 1 skoro ve všech ohledech, ale najdou se i některé nevýhody. Jelikož Struts 2 vyšly na konci roku 2006, zatím si ještě nezískaly takovou základnu příznivců a ani dokumentace k nim není úplná.

Dalším frameworkem Javy je JavaServlet Faces (JSF), který je součástí Java EE a po Struts 1 je jedním z nejrozšířenějších frameworků Javy. Podobně jako ostatní frameworky, JSF používá speciálních tagů v JSP souborech.

Pro přenos dat se používají JavaBeany, které se nastavují pomocí souborů XML. Pro přechod mezi pohledy se používají navigační pravidla (podobně jako Akce Struts). Navíc každá stránka obsahuje takzvané JSF komponenty, které se mohou navzájem vnořovat (viz. [10]) .

3 Spuštění aplikace

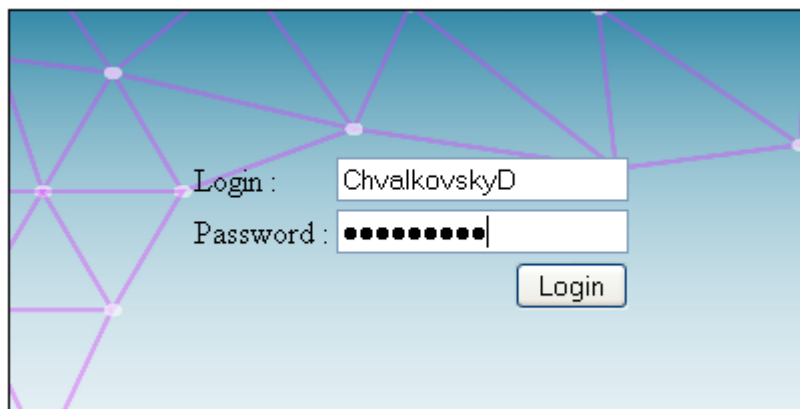
V rámci diplomové práce byla vyvinuta webová aplikace RemStat (*Remote Station*), neboť nejlépe a nejkratším způsobem vystihuje její podstatu. RemStat zajišťuje přenos a správu dat mezi klientem a serverem, spouštění a ovládání programů pro výpočet numerických modelů na straně serveru. Tato aplikace slouží především pro řešení náročných úloh, u kterých by výpočet na klasickém počítači zabral hodně času a pro které by bylo výhodnější vyčlenit počítač s velkou výpočetní kapacitou.

Výhodou RemStatu jako webové aplikace je dostupnost na internetové síti, kde záleží hlavně na rychlosti připojení, která hraje zejména roli při posílání souborů mezi klientem a serverem. Aplikace RemStat pracuje s programy, které je možné spouštět přes příkazovou řádku a nepotřebují žádné grafické rozhraní.

Ovládání aplikace RemStat z pohledu klienta je velice jednoduché a stačí k tomu webový prohlížeč, správcem nastavený uživatelský účet a správně zadaná webová adresa. V této kapitole jsou popsány všechny stránky, které jsou použity ve webové aplikaci. Dále je zde popsán postup spuštění aplikace na serveru Tomcat nebo ve vývojovém prostředí Javy.

3.1 Přihlášení do aplikace

Pro přihlášení do aplikace musí mít uživatel zřízen uživatelský účet s přihlašovacím jménem a heslem. Pokud uživatel není dosud přihlášen, tak po vypsání adresy aplikace (například <http://localhost:8080/RemStat/>) do prohlížeče se zobrazí přihlašovací okno viz obrázek 2.



Obrázek 2: Přihlašovací okno do aplikace RemStat.

3.2 Hlavička a menu stránek

Okna stránek obsahují vždy hlavičku a menu aplikace. Hlavička obsahuje název aplikace RemStat a nové logo Technické univerzity v Liberci. Menu obsahuje název aktuální stránky, pokud je načtený projekt, tak i název projektu a odkazy na projekty, programy, nastavení, jazyk a odhlášení z aplikace.

Práce s projekty je rozdělena do tří částí (seznam projektů, práce se soubory projektu a spouštění projektu, kde je zobrazen i výpis výsledků). V menu se dále zobrazuje název aktuálně načteného projektu a odkazy na další operace s ním spojené (nastavení projektu a jeho souborů, odkaz na spuštění výpočtu a po spuštění výpočtu odkaz na stažení výsledné sady výstupních souborů).

Další částí menu jsou použité programy. Obsah této části se načítá ze seznamu programů aplikace RemStat. Poslední částí menu je nastavení. Tato část slouží k výpisu uživatelů a k nastavení jejich účtů. Rozdělení obsahu stránek podle menu vypadá takto:

- Projekty (vlastní projekty, pouze administrátor vidí všechny)
 - Seznam
 - Soubory
 - Spuštění
- Nastavení programů (pouze administrátor)
 - FLOW123D
 - GENFLOW
- Nastavení (pouze administrátor)
 - Uživatelé (výpis)
 - Založení nového uživatele
 - Editace
- Nastavení účtu uživatele
- Jazyk (Česky nebo anglicky)
- Odhlášení

3.3 Práce s projekty

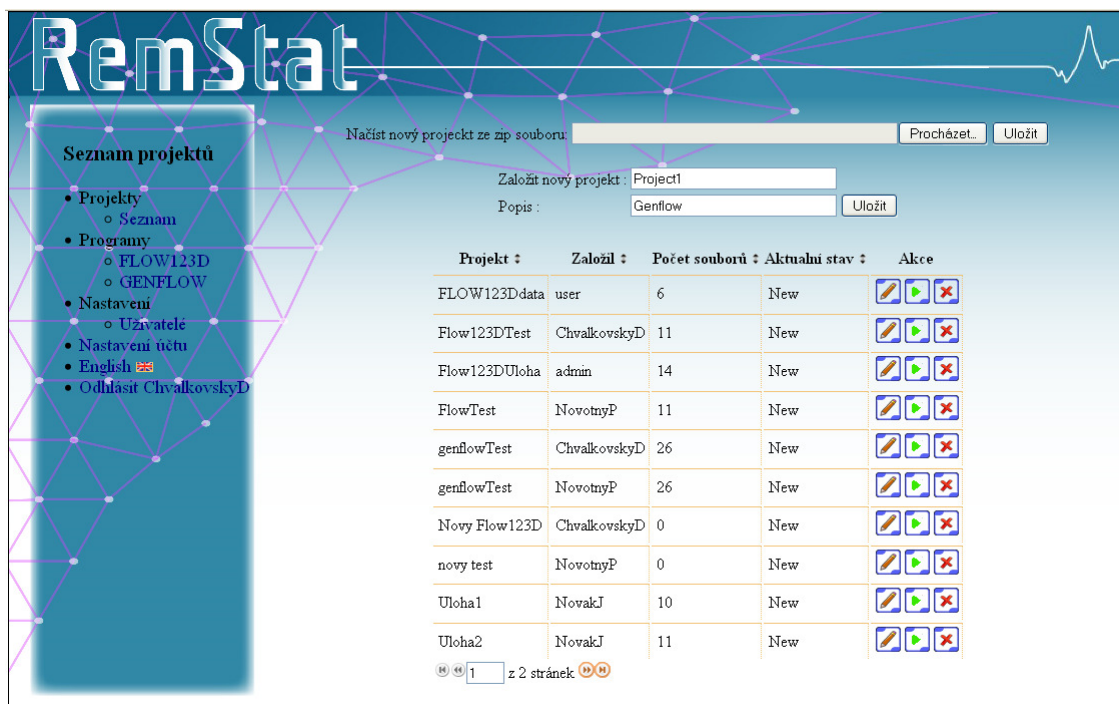
Tato část menu aplikace je nejrozšířenější a zahrnuje veškerou práci s projekty od jejich výpisu, nastavení, načítání, načítání jednotlivých souborů, až po spouštění, výpis výsledků a stažení výsledného komprimovaného souboru s výslednými soubory.

Taková to práce s projekty by měla být volně dostupná studentům a dalším zájemcům, kteří mají zájem na práci s numerickými modely. Pokud by se tento program používal pro výuku, museli by se upravit přístupová práva k daným projektům, protože by studenti mohli zasahovat do práce ostatních studentů.

3.3.1 Seznam projektů

V tomto seznamu se zobrazují projekty vytvořené daným uživatelem. Pouze administrátor má právo nahlížet do všech projektů. U každého z projektů je zobrazen jeho název projektu, kdo ho založil, počet souborů v adresáři, jeho aktuální stav a odkazy na editaci projektu a jeho souborů, spuštění projektu a jeho smazání (pouze probíhající projekt nelze smazat) viz obrázek 3.

Na této stránce je možné také založit prázdný projekt, u kterého je možné zadat popis projektu a pak načíst všechny soubory jednotlivě. Druhou možností jak založit nový projekt, je načíst celý nový projekt najednou z komprimovaného zip souboru. Tento soubor musí obsahovat složku, ve které je uložena sada vstupních souborů programu. Takto založený projekt ponese název složky v zabaleném souboru.



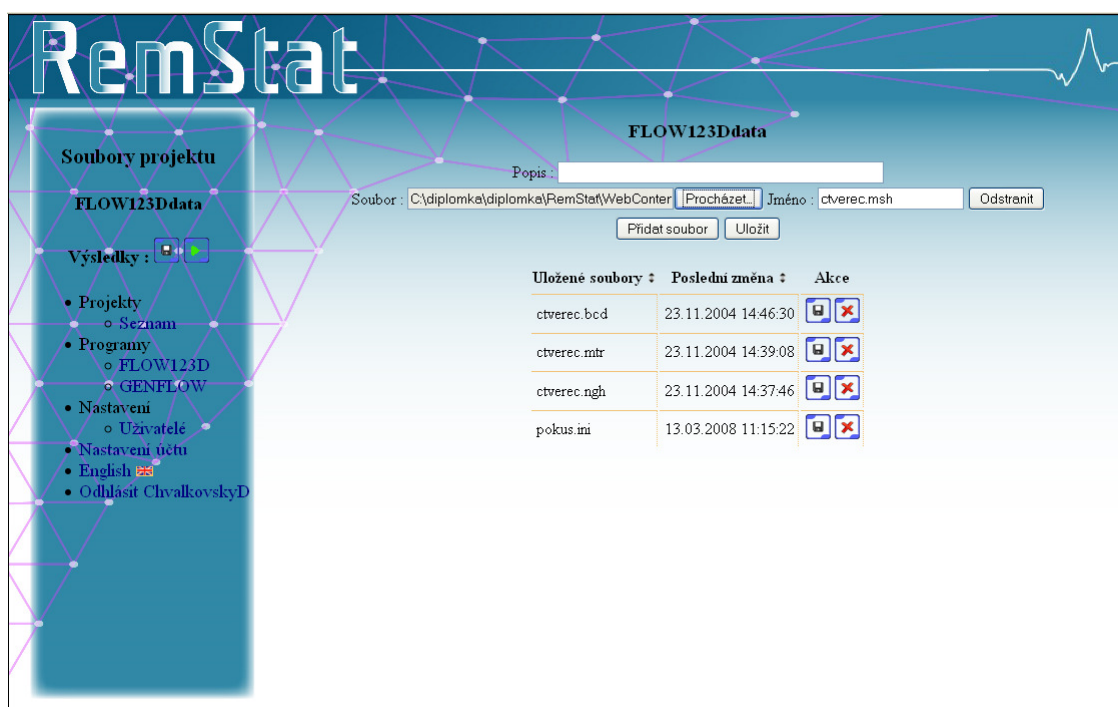
Obrázek 3: Stránka obsahující výpis projektů a jejich zakládání.

3.3.2 Editace souborů projektu

Tato stránka se zaměřuje na práci se soubory projektu a je možné se na ni dostat z předešlé stránky z odkazu pro editaci projektu (obrázek tužky). Na této stránce je možné přidávat, mazat a nebo stahovat soubory daného projektu.

Pro uložení více souborů najednou je zapotřebí přidávat nová pole pro jejich načítání a do nich vyplnit cestu k souboru procházením adresářů. Při vybrání souboru, se implicitně vyplní jméno souboru i s koncovkou do pole se jménem. V tomto poli je možné upravit název souboru, pod kterým se soubor uloží do projektu.

Pokud jsou již načtené soubory uloženy na serveru, zobrazí se ve spodní tabulce, kde je možné s nimi dále pracovat viz obrázek 4.



Obrázek 4: Editace souborů projektu.

3.3.3 Spuštění projektu, výpis výsledků

Stránka určená pro spouštění programů se objeví po kliknutí na odkaz pro spuštění projektu, který znázorňuje zelená šipka. Dále je zapotřebí z nabídky programů zvolit program, který bude pracovat s daným projektem a nastavit příkaz, který se pošle programu viz obrázek 5. Pro lepší přehled se příkaz skládá z několika částí vypsanych zvlášť do oken, které se běžně oddělují mezerou. Tyto okna lze libovolně přidávat nebo ubírat, prázdná okna se nezpracovávají. Pro jednodušší práci s příkazy a kvůli bezpečnosti se cesta k programu nezadá a je automaticky přidělena aplikací.

Jednou z výhod je načtení základního příkazu, který se vyplní po stisku tlačítka. Tento příkaz slouží pro jednodušší vyplňování a pro časté vkládání stejného nebo podobného textu. Základní příkaz je uložen pro každý program v souboru *settings.properties* a za běhu aplikace je možné jej měnit pokaždé v menu nastavení programu.

Příkaz se obvykle skládá z názvu spustitelného souboru, který se může psát s příponou i bez přípony (např. s příponou *.exe*). Další kolonky jsou určeny pro parametry úlohy jakými jsou speciální konstanty programu určující parametry obvykle začínající pomlčkou a obsahující jedno písmeno udávající specifikaci úlohy

následovanou například názvem načteného souboru nebo název některého z parametrů použitých v programu.

Pokud jde o parametr, který se skládá z více slov oddělených mezerou je zapotřebí tento název umístit do dvojitého uvozovky. Všechny parametry bývají popsány v nápovědě programu a tu lze spustit stejně jako běžnou úlohu příkazem obsahujícím název spustitelného souboru, za kterým následuje lomítko a otazník nebo pomlčka a h (-h nebo také /?).

Příklad spuštění nápovědy programu Flow123d příkazem *flow123d /h*:

```
This is 1-2-3-FLOW, version 1.3.2 Build #1; Builded on Mar 29 2006 at 10:37:37.
```

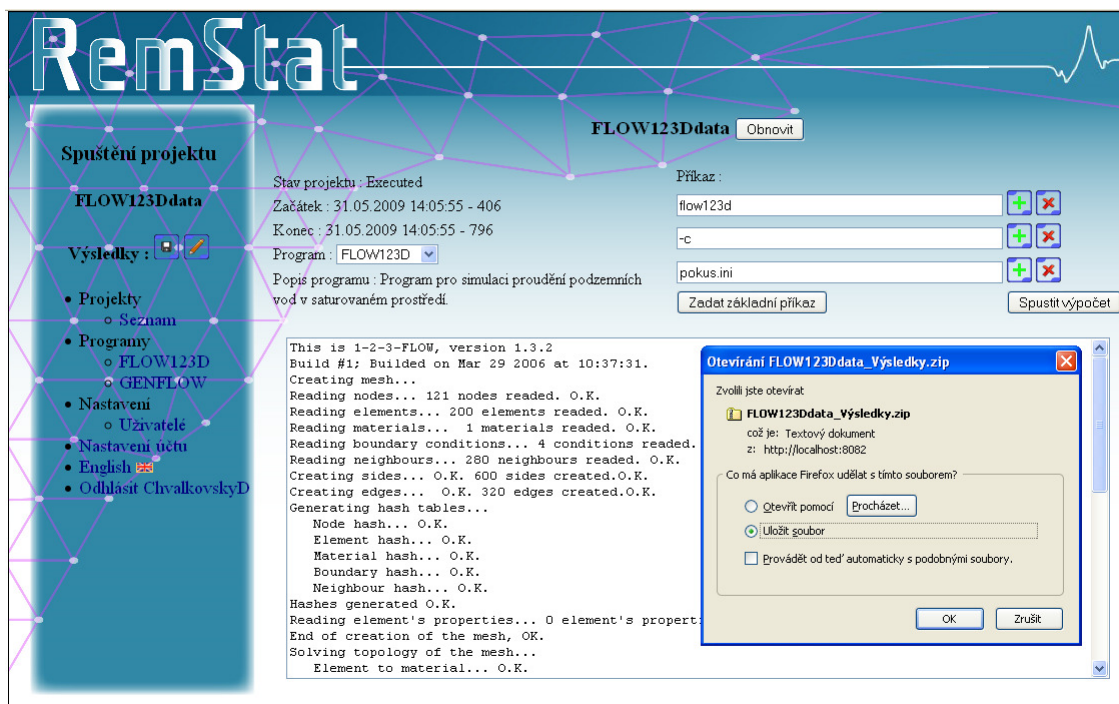
```
Usage: flow123d [options] ini_file Options:
```

- s Compute MH problem
- c Convert flow data files into Gmsh parsed post-processing file format

Po zadání příkazu a spuštění programu se na stránce zobrazí stav projektu, přesný čas začátku a po ukončení úlohy i konec výpočtu. Projekt může nabývat čtyřech stavů a to nový, vykonává se, provedený, zastaven a chyba (v aplikaci je popsán takto: new, running, executed, stoped a error). Poslední částí je výpis událostí z programu, kam se zapisují všechny výstupy programu.

Po vykonání programu se zobrazí ikona s disketou v menu. Po kliknutí na tuto ikonu je možné si stáhnout vygenerované soubory (soubory, které byly vytvořeny nebo pozměněny po spuštění programu) i s výsledky vypsányými v log souboru podobně jako je tomu v editaci souborů projektu.

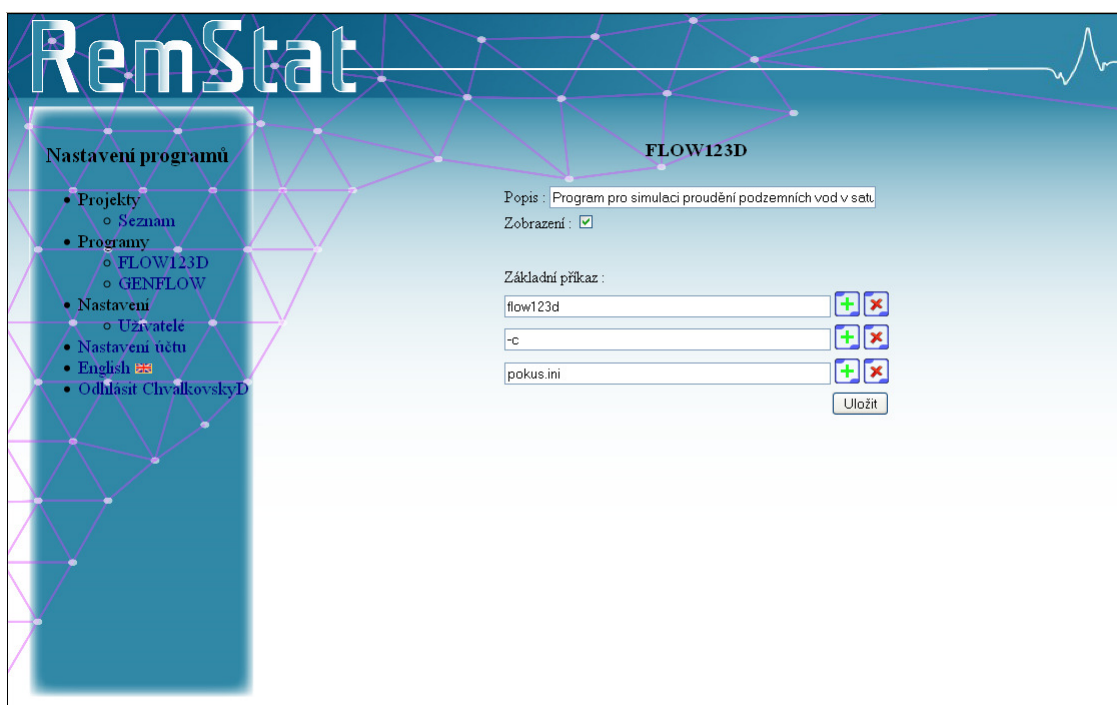
Informace o průběhu výpočtu, použité příkazy a informace o výsledných souborech se udržují v paměti do té doby, než je projekt smazán nebo spuštěn, další výpočet s daným projektem nebo při restartu serveru.



Obrázek 5: Spuštění programů.

3.4 Nastavení programů

Aplikace RemStat umožňuje za běhu aplikace měnit nastavení programů uživatelům s administrátorskými právy. U programů lze měnit popis, zobrazení v aplikaci (Program může odstranit z nabídky programů pro spuštění výpočtů), obsah základního příkazu, který je popsán v předešlé kapitole 3.3.3. Výhodou nastavení zobrazení je rychlé přidání nebo ubrání programu z nabídky spustitelných programů bez nutnosti jejich mazání viz obrázek 6.



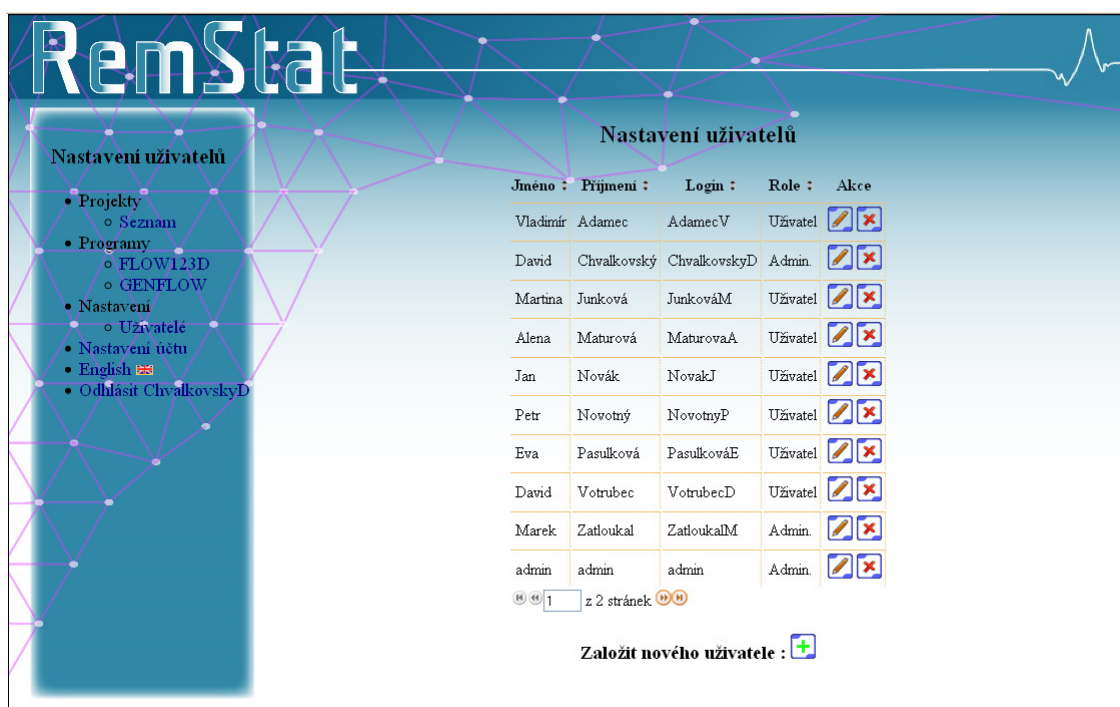
Obrázek 6: Nastavení programů v aplikaci.

3.5 Nastavení uživatelů

Kromě nastavení programů může administrátor nastavovat dále účty uživatelům, přidávat nové nebo je mazat. K tomu slouží stránka s výpisem všech uživatelů a stránka pro jejich přidání a editaci.

3.5.1 Výpis uživatelů

Výpis uživatelů obsahuje údaje o uživatelích a odkazy na přidání nového uživatele či administrátora a odkazy na jejich editaci a smazání z databáze viz obrázek 7.



Obrázek 7: Výpis uživatelů.

3.5.2 Založení a editace uživatele

Tato stránka slouží pro založení nového uživatele i pro jeho editaci. V případě vytvoření nového uživatele je zapotřebí vyplnit všechny údaje. Pokud je uživatel editován, nemusí se znovu zadávat heslo viz obrázek 8.

Podobně jako je možné nastavit administrátorem jakéhokoli uživatele aplikace, je možné nastavit vlastní údaje v nastavení svého účtu, kromě administrátorských práv. Také je zde zapotřebí pro změnu hesla zadat to staré.



Obrázek 8: Přidání nového uživatele a editace stávajícího.

3.6 Spuštění webové aplikace na serveru

Oproti tomu spuštění webové aplikace na serveru Apache Tomcat je o něco složitější. Apache Tomcat je zapotřebí nejdříve stáhnout ze stránek Jacarta projektu a posléze nainstalovat a spustit. Dále je zapotřebí mít nainstalovanou databázi MySQL a v ní mít uloženou tabulku s uživateli. Dále do nainstalované složky Apache do adresáře s názvem *lib* zkopírovat knihovnu ovladačů databáze MySQL.

Pokud je zapotřebí aplikaci upravit, nebo přidat další části, je možné si kromě Apache Tomcatu nainstalovat vývojové prostředí Javy a to Eclipse nebo Netbeans. Tato vývojová prostředí si může každý stáhnout zdarma na internetu a nainstalovat na svůj počítač. Po nainstalování a spuštění vývojového prostředí je zapotřebí importovat aplikaci RemStat a hlavně přiřadit server Apache Tomcat tomuto projektu.

3.6.1 Softwarová konfigurace

Pro spuštění webové aplikace RemStat na serveru a pro její vývoj je potřeba těchto programů:

- Vývojové prostředí Javy Eclipse Ganymede verze 3.4.x nebo NetBeans verze 6.x
- Java verze 6

- MySQL verze 5.x
- Apache Tomcat verze 6.x

A je podporována operačními systémy, kde lze spustit JVM (platí pro server):

- Windows
- Linux

3.6.2 Soubor RemStat.war

Tento soubor je součástí diplomové práce a obsahuje uloženou webovou aplikaci RemStat, která obsahuje třídy Javy a další pomocné soubory nezbytné pro spuštění aplikace pod serverem Apache Tomcat nebo k otevření projektu ve vývojovém prostředí Eclipse nebo NetBeans. Dále v sobě obsahuje potřebné knihovny Javy, programy GENFLOW a FLOW123D a vstupní data těchto programů, uložená v projektech (viz. Příloha A s obsahem adresářů aplikace RemStat). Ke spuštění a k vyzkoušení aplikace není nutné přikládat další soubory nebo upravovat nastavení aplikace.

V případě nasazení aplikace k běžnému použití je zapotřebí si rozmyslet, zda-li tyto data (Programy a projekty) oddělit od aplikace nebo je ponechat v rámci aplikace. Pokud by tyto data zůstala uložena v rámci aplikace, mohlo by dojít k přepsání celého jejího obsahu po nasazení nové verze aplikace.

3.6.3 Spuštění aplikace

a) Postup spuštění aplikace RemStat ve vývojovém prostředí Javy:

- 1) Importovat projekt ze souboru *RemStat.war* do Eclipse nebo do NetBeans.
- 2) Použít platformu Javy EE
- 3) Nainstalovat zvlášť server Apache pokud není součástí vývojového prostředí (nechat vypnutý)
- 4) V Eclipse přidat server Apache mezi servery (u Netbeans je součástí vývojového prostředí).
- 5) Přiřadit aplikaci RemStat k serveru.
- 6) Spustit Apache server v Eclipse (Netbeans) a v prohlížeči si pustit aplikaci na adrese <http://localhost:8080/RemStat/> (pokud jste si nezadali jiný port nebo jiný název aplikace).

b) Spuštění aplikace pouze na serveru *Apache*:

- 1) Po nainstalování a spuštění *Apache* serveru, zadat adresu <http://localhost:8080/> do prohlížeče.
- 2) Pokud je server spuštěn, tak po zadání adresy do prohlížeče se zobrazí stránka, s odkazem na Tomcat manager. Po zvolení odkazu na manager je zapotřebí vyplnit heslo, které bylo nastaveno při instalaci.
- 3) V Tomcat manageru po kliknutí na odkaz *deploy* se zobrazí nabídka, kam se zadá cesta k uloženému souboru *RemStat.war*, který se na serveru rozbalí a umožní v tabulce aplikací spustit aplikaci RemStat.
- 4) Výsledná adresa aplikace by měla být <http://localhost:8080/RemStat/> pokud nebylo změněno číslo portu 8080.

c) Nastavení databáze MySQL:

- 1) Po instalaci MySQL, je potřeba vytvořit novou databázi pomocí skriptu *mysqlDatabase.sql* a nastavit parametry souboru *context.xml* (viz příloha E) v aplikaci RemStat.
- 2) Pro spuštění aplikace je potřeba vložit soubor s ovladači MySQL mezi knihovny *Apache*.

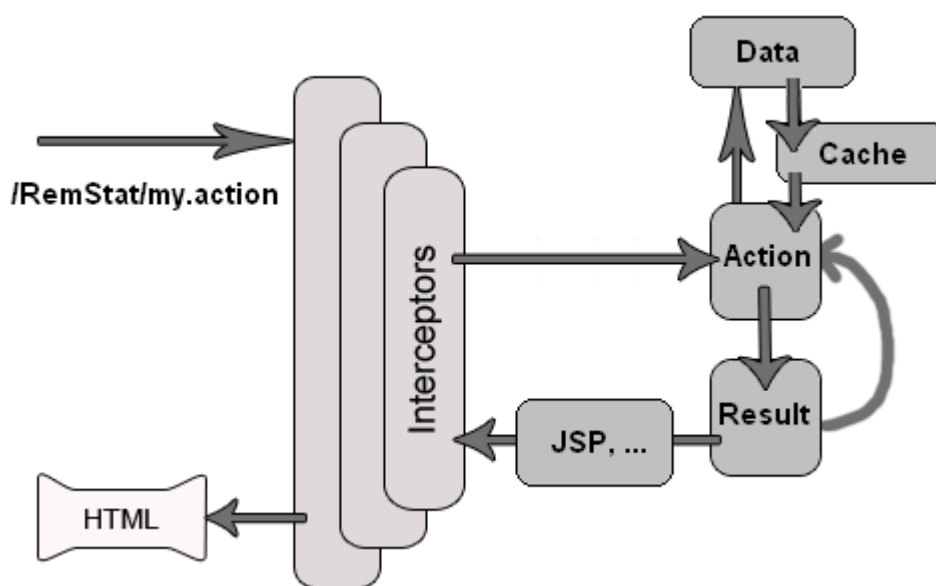
3.6.4 Bezpečnostní rizika

Pro bezpečný chod webové aplikace RemStat, by nemělo být možné, aby si běžný uživatel pomocí příkazů sloužících pro spouštění programů, mohl zadat takový příkaz, který by mohl spustit program s jakým by mohl zasahovat či jinak manipulovat s citlivými daty nebo se soubory uloženými na serveru kromě těch v projektech. Proto je zapotřebí dbát zřetel na výběr použitých programů.

Dalším rizikem může být velké zatížení serveru v případě načítání objemných souborů a to z pohledu zatížení internetové sítě a také z hlediska zaplnění vyhrazeného místa na disku serveru nebo také vytížení serveru vykonáváním náročných úloh. V tomto případě je věcí administrátorů, aby si ohlíželi obsah projektů a úlohy spouštěné na serveru, které patří daným uživatelům.

4 Popis aplikace

Webová aplikace RemStat je rozdělena do několika funkčních částí. První část aplikace zajišťuje zprávu dat. Jedná se o takzvanou bussines logiku tvořenou třídami Javy umístěnými v adresáři *data*. Druhou částí je prezentační logika, která zajišťuje prezentaci dat na webu tak, jak je tomu v MVC modelu viz kapitola 2.3.1. Do prezentační logiky patří třídy umístěné do adresáře s názvem *web* (ten představují zejména akce) a z části i *resources* (nastavení, popisky). Do této části také patří obsah složky *WebContent*, kde se vyskytují XML soubory s nastavením webové aplikace, JSP soubory, CSS, JavaScript a další pomocné soubory.



Obrázek 2: Architektura Struts 2 frameworku tvořící prezentační vrstvu komunikující se zdrojem dat (třídy Javy v balíčku *data*) tvořící bussines logiku (viz [13]).

4.1 Adresář *data*

Všechny třídy Javy se sdružují do balíčků (packages), což jsou ve skutečnosti adresáře ve stromové struktuře odděleny tečkou. Tyto balíčky tříd jsou rozděleny v aplikaci RemStat do třech základních adresářů s názvy *data*, *web* (pracuje s rámcem *Struts 2*) a *resources* (reprezentuje nastavení, použité konstanty a popisky použité k tvorbě webových stránek) viz. Příloha A.

Adresář (balíček) s názvem *data* obsahuje jádro aplikace (aplikační logika). Tyto třídy jsou nezávislé na zbytku aplikace, umožňují přístup k datům uložených na serveru. Tato část kódu je vyměnitelná, její třídy se dají použít i v jiném projektu, nebo se

mohou vzít a použít pro jinou aplikaci se stejným využitím, která bude mít jiné uživatelské rozhraní, například by mohla přenášet data jako tlustý klient (ve *Swingu*), přes protokol TCP/IP.

4.1.1 Jádru aplikace

V jádře aplikace je obsaženo několik balíčků a to balíček se zásobárnou dat (*cache*), který obsahuje všechna důležitá data o programech, projektech a uživateli, která se načítají při startu serveru a mění se v průběhu chodu aplikace. Tato zásobárna slouží jako rychlá paměť pro časté načítání stejných dat. Tyto data by se jinak musela pokaždé načítat z úložiště dat (ze souborů či z databáze), což by zpomalovalo načítání stránek webové aplikace. Do této paměti se ukládají informace o spustitelných programech, uživateli, o obsahu projektů, o průběhu výpočtů a o jejich výsledcích.

DataCore je třída, která slouží pro inicializaci ostatních tříd v balíčku *data.cache* při startu aplikace a sama je inicializována třídou *ServletContextListener*. Při startu aplikace je potřeba inicializovat programy, které jsou nastaveny pomocí souboru *settings.properties*, což zajišťuje třída s názvem *ProgramsData*. Tato třída si uloží všechna potřebná data ze souboru do kolekce JavaBeanů s názvem *ProgramDTO*, popsaných v následující kapitole. Tyto data ze souboru říkají jaké programy se mají načíst a jsou uchovávány po celou dobu běhu aplikace.

Další třídou, která je inicializována třídou *DataCore*, je třída *ProjectsData*, která při své inicializaci, načítá všechny projekty, které se nacházejí v adresáři projektů do seznamu JavaBeanů s názvem *ProjectDTO*, popsaným též v následující kapitole. Seznam těchto JavaBeanů se v průběhu aplikace mění v závislosti na práci s projekty uvnitř aplikace.

Poslední třída, která se nachází v balíčku *data.cache* je třída *UserData*. Obsah této třídy se plní z databáze a je tvořen informacemi o uživateli, kteří pracují s aplikací RemStat.

4.1.2 JavaBeany pro přenos dat v rámci aplikace

Balíček s třídami DTO (*data transfer objekty*) obsahuje JavaBeany (speciální třídy Javy viz kapitola 2.2.3) *FileDTO*, *ProjectDTO*, *ProgramDTO* a *UserDTO*. Tyto třídy slouží pro přenos dat v rámci aplikace, dále obsahují důležitá data, která se

předávají v rámci dalších tříd, ale také v rámci akčních tříd a JSP souborů, ze kterých se generují HTML stránky.

Třída *FileDTO* obsahuje jméno souboru, čas poslední změny, (tento čas slouží především pro rozlišení souborů, které vznikly před a po spuštění výpočtu, takto se z jedné složky získá sada výstupních souborů programu). Tato třída také v sobě obsahuje datový typ soubor, který slouží pro přenos obsahu souboru v rámci aplikace.

Podobně jako třída *FileDTO* slouží pro přenos dat a informací o souborech, tak třída *ProjectDTO* se používá pro jednodušší přenos informací o projektech v rámci aplikace. Třída *ProjectDTO* obsahuje název projektu, jméno spuštěného programu, příkaz zaslaný programu, čas počátku a konce výpočtu, informace a výpisy o jeho průběhu, ale také obsahuje vlákno, na kterém výpočet běží. Dále tato třída obsahuje jména a atributy všech souborů, které jsou v projektu, potřebné k výpisu na stránce editace souborů projektu.

Dalším JavaBeanem v aplikaci je třída s názvem *ProgramDTO*, pomocí které se pracuje s daty potřebnými ke spuštění programů ovládaných aplikací RemStat. Tato třída obsahuje název a popis programu, sadu základních příkazů pro spuštění aplikace. Dále obsahuje sadu ukončovacích hesel, která jsou potřeba pro programy, které se samy neukončí po dokončení úlohy (například program Flow123d, který po vykonání úlohy čeká na stisk klávesy *enter*).

Poslední třída DTO se nazývá *UserDTO* a slouží k přenosu informací o uživateli v rámci aplikace. Tato třída obsahuje jméno, příjmení, heslo uživatele, nové heslo, ověření hesla, login, který musí být jedinečný, používaný jazyk a dále informaci o tom, zda-li je uživatel administrátor.

4.1.3 Přístup k datům

Další položkou v adresáři data jsou DAO (*data access object*) objekty pro přístup k datům, což jsou rozhraní (*interfacy*) tříd uložených v balíčku *persistence*, které pracují s daty uloženými na disku. Rozhraní definuje metody, které daná třída musí splňovat (načítání, ukládání, mazání, kompresi a dekompresi projektů a jejich souborů), aby mohla být zařazena do projektu. Tyto DAO objekty slouží především k oddělení vrstev v aplikaci, k jejich zpřehlednění, jednodušší výměně a znovu použití v jiném projektu.

Třídy obsahující metody, které jsou popsány DAO objekty se nalézají v balíčku s názvem *persistence*, což znamená něco trvalého, tedy práce s trvale uloženými daty na disku nebo v databázi. Mezi tyto třídy patří třída s názvem *FileHandler*, která obsahuje metody pro ukládání a mazání souborů, dále výpis všech souborů, uložených ve složce a metodu pro načtení výsledných souborů výpočtu.

Další třídou v balíku *persistence* je třída s názvem *ProjectHandler*, ve které se nacházejí metody pro práci s projektem, který reprezentuje adresář s názvem projektu. Tyto projekty se nacházejí v adresářích uživatelů. Tyto adresáře uživatelů nesou jméno přihlašovacího jména uživatele a slouží k přiřazení projektů jednotlivým uživatelům. Stejně tak obsahují adresáře projektů názvy projektů. Výpisem všech adresářů ve vyhrazeném adresáři pro projekty se získají všechny projekty v aplikaci RemStat. Třída *ProjectHandler* dále obsahuje metodu pro získání potřebných dat o projektu, metodu pro založení nového a metodu pro smazání projektu i s jeho obsahem.

Předposlední třída z balíku *persistence* se nazývá *ZipProjectHandler*. Tato třída zajišťuje komprese a dekompresi dat. Komprese dat slouží především pro posílání výsledných souborů, či celého projektu uživateli. Aby nedocházelo k ukládání komprimovaných dat na serveru, které by zabíralo místo a zpomalovalo načítání dat, ukládá se obsah těchto komprimovaných souborů do *ByteOutputStreamu*, který se dále načítá do *InputStreamu*, který slouží pro posílání dat uživateli pomocí Struts 2.

Dekomprese dat probíhá v aplikaci RemStat následujícím způsobem. Přijatá komprimovaná data od uživatele se neukládají přímo na disk, ale nejprve se zjistí, jestli uživatel poslal komprimovaný adresář ve správném formátu. Komprimovaný soubor musí v sobě obsahovat jednu složku, od které se převezme název projektu. Tato složka může mít libovolný obsah. Pokud je vše správně nastaveno pokračuje se v rozbalování přidružených souborů a jejich následné ukládání na disk serveru.

Poslední třídou balíku *persistence* je třída *UserJdbc*, která slouží pro načítání informací o uživatelích z databáze. Tato třída obsahuje metody pro načtení uživatele, všech uživatelů, vložení nového uživatele, změna záznamů a vymazání záznamů o uživateli.

4.1.4 Spouštění výpočtů

Předposlední položkou aplikační logiky je třída *ProgramExecute*, která slouží pro spouštění programů pro numerické výpočty modelů umístěných na serveru.

Vstupními parametry metody této třídy je JavaBean s názvem *ProjectDTO* popsaný v předešlých kapitolách, který v sobě nese informace o projektu a o daném programu, který bude použit k výpočtu.

Před spuštěním nového vlákna, na kterém poběží výpočet je zapotřebí zjistit, zda-li projekt již není spuštěn a pokud není, tak nastavit jeho parametry, tak aby nedošlo k opětovnému spuštění. Pokud je nové vlákno vytvořeno, nastaví se pracovní složka projektu, připraví se příkaz rozčleněný do několika částí a nakonec se spustí vlákno pro výpočet.

4.1.5 Vlákno pro výpočet

Pro každý výpočet je potřeba vytvořit v aplikaci RemStat jedno samostatné vlákno s názvem *ExecuteProgramThread*. V rámci vlákna se vytvoří instance třídy *ProcessBuilder*, která zajišťuje spuštění programů. Vstupními parametry této třídy je adresář, ve kterém se nachází vstupní sada souborů projektu a příkaz rozdělený na jednotlivé části potřebných pro spuštění programu.

Po spuštění instance *ProcessBuilderu* se načítají zprávy ze spuštěného programu a ukládají se do JavaBeanu *ProjectDTO*, stejně tak se zachytávají chybová hlášení, která se pak objeví na stránkách projektu. Započatý výpočet je také možný ukončit zrušením spuštěného procesu.

4.1.6 Třída *ServletContextListener*

Poslední třídou v tomto balíku je třída *ServletContextListener*, která slouží pro spuštění inicializace jádra aplikace (načtení souborů, programů, projektů a uživatelů do zásobárny dat). Tato třída je definována jako poslouchač (*listener*) v souboru s názvem *web.xml*, který leží v adresáři *WEB-INF*. Třída *ServletContextListener* je pak inicializována při spuštění aplikace na serveru Apache Tomcat (viz. Příloha D).

4.2 Základní rozdělení akcí

Webové rozhraní, nebo-li prezentační logika, je tvořena třídami akcí (*actions*). Všechny akční třídy v této aplikaci dědí od třídy *GenericAction*, která leží v základním balíku, akcí *remstat.web.action*. Třída *GenericAction* dále dědí od třídy Struts 2

ActionSupport patřící do balíku knihoven Struts 2. Výsledné akční třídy slouží pro provádění akcí, které jsou mapovány v XML souborech. Hlavním mapovacím souborem akcí je XML soubor s názvem *struts*, který dále poukazuje na další XML soubory s nastavením akcí projektu (*project*) a nastavení (*settings*).

Akce jsou rozděleny do XML souborů pro přehlednost, tak jak je tomu v menu. Stejně tak, jako jsou rozděleny akce do XML souborů, jsou rozděleny akční třídy do balíčků a k nim přiřazené jednotlivé JSP soubory se stejným názvem a se stejným uspořádáním adresářů. Příklad mapování akčních třídy v XML souboru, akční třída a JSP soubor je popsán v kapitole 2.3 Struts 2 framework.

Mapovací XML soubory obsahují také balíčky, které většinou nesou název souboru. Výjimkou je soubor *struts.xml*, který v sobě obsahuje dva balíčky. První balíček je abstraktní a obsahuje základní nastavení *interceptorů* a výsledků akcí (přesměrování na soubory *error.jsp* a *login.jsp*). Od tohoto balíčku pak dědí všechny ostatní. Druhým balíčkem je balíček sloužící pro přihlášení a odhlášení uživatele.

Každý z balíčků, kromě toho abstraktního obsahuje *namespace*, který označuje rozdělení balíčku akcí v URL adrese do složek oddělených lomítkem, podobně jako je tomu v adresářové struktuře.

4.3 Uživatelské akce

Základní akce jsou mapovány v hlavním mapovacím souboru *struts.xml* a tvoří je akce přihlášení a odhlášení uživatele do aplikace, nastavení jazyka a uživatelského účtu. Při úspěšném přihlášení uživatele nebo při přepnutí jazyka či nastavení účtu uživatele, je stránka automaticky přesměrována na akci s výpisem projektů.

Akční třídy uživatelských akcí jsou uloženy stejně jako předek všech akčních tříd této aplikace s názvem *GenericAction* v základním balíku akcí *remstat.web.action*. Akční třída přihlášení uživatele do aplikace, ověřuje při odeslání přihlašovacího formuláře správnost přihlašovacích údajů a pokud jsou správné, načte data o uživateli do proměnné *session* s názvem *loginData*, která dále v aplikaci slouží k ověřování a k ukládání informací o uživateli, jakými je jeho role (administrátor), použitý jazyk a další.

Session s těmito daty o uživateli zaniká při odhlášení uživatele nebo při jeho delší nečinnosti.

4.4 Akce projektů

Část akcí s názvem *project* je nejrozsáhlejší a zaměřuje se na práci s projekty. Tuto část tvoří třídy v balíku *remstat.web.struts.actions.project*, XML mapovací soubor s názvem *project* a JSP soubory ve stejnojmenném adresáři. Druhou část tvoří veškeré akce pracující s projekty, jakými je jejich výpis, nastavení, načítání nebo mazání (celého projektu a také jednotlivých souborů), spouštění programů (programy pracující s těmito projekty), zprávu příkazů, výpis výsledků, informace o průběhu nebo ukončení výpočtu, posílání výsledných souborů a další akce, spojené s projekty popsané v následujících podkapitolách.

4.4.1 Výpis projektů

Výpis projektů je tvořen akční třídou *ProjectListAction* a JSP souborem *ProjectList*. Jedná se o vybrání projektů, jejich seřazení a následné zobrazení v tabulce projektů. Projekty, jsou vybrány na základě přihlášeného uživatele, pokud je uživatel je přihlášen jako administrátor, zobrazí se mu všechny projekty a pokud není, tak se mu zobrazí pouze ty projekty, které sám založil.

Výpis projektů do tabulky JSP souboru se provádí pomocí Struts 2 komponent (tagů), jakými jsou *s:iterator*, *s:property* a další. Dále se také vypisují pomocí Struts důležité parametry do JavaScriptu, který dále slouží pro vykonávání akcí na straně klienta. Mezi tyto akce JavaScriptu patří práce se sortováním a stránkováním v tabulce, validace odesílaných dat při zakládání nových projektů a dále zpráva aktivních tlačítek, které mění svůj obrázek při přiblížení kurzoru a další.

4.4.2 Založení nového projektu

Založení projektu je možné dvěma způsoby. První způsob je založení projektu bez jeho obsahu. V tomto případě se vytvoří pouze složka s názvem projektu, do které se následně mohou kopírovat soubory. K tomuto účelu slouží formulář (*s:form*) s názvem *projectListForm* v JSP souboru, který obsahuje skryté vstupy, kam se uloží parametry Projektu a ty jsou pak dále převzaty přístupovými metodami třídy *ProjectListAction*.

Tento formulář dále slouží i pro odeslání dat dalších akcí projektů, jakými je editace, mazání nebo spuštění projektu, což je zajištěno JavaScriptem, který po kliknutí uživatelem na tlačítko, nastaví všechny potřebné parametry vstupů v rámci JSP souboru.

Druhým způsobem uložení projektu je odeslání komprimovaného souboru na server, který se následně rozbalí mezi projekty, tak jak byl poslán. Proto je zapotřebí, aby komprimovaný soubor obsahoval složku s názvem projektu, ve které se budou nacházet vstupní soubory programů. Pokud uživatel toto pravidlo poruší a například pošle komprimovaný soubor, který bude obsahovat pouze soubory bez složky bude uživatel upozorněn po pokusu načíst data.

4.4.3 Úprava projektů, načtení souborů

Po založení projektu je někdy zapotřebí upravit projekt, například změnit popis, přidat, načíst zpět nebo smazat soubory. K tomuto účelu slouží třída *ProjectFilesAction*, která načítá dat z JSP souboru *editProject*, které zahrnuje kromě menu aplikace také JSP soubor *projectFiles* potřebný pro načítání souborů.

Pro načítání více souborů najednou bylo zapotřebí najít způsob, jak tyto soubory poslat v celku. Nejprve bylo zapotřebí pomocí JavaScriptu připravit vstupní pole pro zadání cesty a názvu souborů. Tyto pole, patřící jednomu souboru, jsou očíslovány stejným číslem, které je převedeno na název, aby se dalo použít při načítání JavaBeanů *FileDTO* do *HashMapy*. Obsah této *HashMapy* je pak načten akční třídou a uložen na disk serveru.

V rámci editace projektů je vypsána tabulka všech souborů, uložených ve složce projektu. V této tabulce je možné načítání a mazání souborů ze serveru. Každý záznam v tabulce je podřízen jednomu souboru s unikátním názvem. Takto lze pomocí JavaScriptu ovládat přesun parametrů do ukrytých vstupů (*s:hidden*) JSP souboru, potřebných pro přesun dat do Akční třídy.

4.4.4 Spouštění programů

Pro spouštění programů je vyhrazena třída *ExecuteProjectAction*, která si předává data s JSP souborem *executeProject* a spolu vytvářejí stránku, která umožňuje zvolení si projektu, programu, zapsání nového příkazu i toho, který je přednastaven

v nastavení programů. Dále se zde zobrazují informace o průběhu výpočtu a jeho výsledky.

Na této stránce není možné, aby například uživatel spustil jeden projekt v jeden čas vícekrát, ale je možné každý výpočet v jeho průběhu ukončit a spustit znovu. Tyto informace o projektech se ukládají do JavaBeanu *ProjectDTO*. Každému projektu přísluší jeden JavaBean, proto při jeho spuštění probíhá kontrola parametru akce. Stejně tak se nastaví parametr akce při jeho dokončení. Podle tohoto parametru se dále mění obsah stránky.

4.4.5 Posílání souborů

Důležitou částí akcí jsou dvě speciální třídy pro posílání souborů uživateli. Třída *DynamicStreamResult* je definována jako výstup akce Struts 2, která obsahuje výstup obsahu souboru v podobě vstupního proudu bytů (*inputstream*) a další parametry jakými je název, typ a velikost stahovaného souboru. Tento výstup akcí je definován v mapovacím XML souboru *project* a slouží pro posílání jednotlivých souborů nebo komprimovaných výsledků klientovy, které jsou načítány v akční třídě *DownloadFileAction*.

Třída *DownloadFileAction* zprostředkovává načtení souboru nebo komprimované sady výstupních souborů na základě použité metody a vstupních parametrech dotazu. V případě poslání samostatného souboru je potřeba název projektu a souboru, pro stažení komprimovaného výsledku stačí název projektu. Tyto parametry se předají metodám tříd v aplikační vrstvě sloužícím k načítání dat a jejich výsledky se předají pomocí *gettrů* a *settrů* instanci třídy *DynamicStreamResult*.

4.5 Nastavení

Do nastavení aplikace RemStat mají přístup pouze administrátoři a skládá se z nastavení programů a uživatelů. Akce, které slouží k tomuto nastavení jsou nastaveny v souboru *setting.xml* a patří k nim také JSP soubory uložené v adresáři *settings* a akční třídy ve stejnojmenném balíčku akcí.

Tato část aplikace slouží k zobrazení a k nastavení použitých programů a k nastavení a zobrazení účtů uživatelů.

4.5.1 Nastavení programů

Pro nastavení programů je vyhrazena akční třída *ProgramAction*, pomocí které se načítají všechny data o programech do JavaBeanů *ProgramDTO*. V tomto případě se přenášejí data do JSP souboru *editProgram*, kde je možné tyto programy editovat a měnit jejich nastavení.

4.5.2 Nastavení uživatelů

Tuto část tvoří tabulka s výpisem všech uživatelů a formulář pro editaci účtů uživatelů. Výpis uživatelů i editace jsou tvořeny akční třídou *UserAction*, která slouží jak pro načítání všech dat, tak i k jejich ukládání v případě založení nebo editace uživatele. Podobně jako u dalších výpisů záznamů do tabulky je možné stránkovat a sortovat záznamy v tabulce pomocí třídy *UsedlostPaging* a také pomocí JavaScriptu který se přidává k JSP souboru s názvem *userList*.

Pokud je vybrán uživatel kliknutím na tlačítko editace v tabulce, pak na základě unikátního přihlašovacího jména, které se odešle jako žádost na editaci, se zobrazí na stránce vygenerovaný obsah z JSP souboru *userEdit* s načtenými daty. Tyto data jsou po odeslání zpět tlačítkem uložit odeslána zpět třídě *UserAction*, která je pak dále uloží mezi ostatní záznamy.

4.6 Další části projektu

Dalšími částmi projektu jsou adresáře a soubory, které tvoří součást uživatelského rozhraní. Mezi tyto součásti patří JSP soubory, soubory s kaskádovými styly, obrázky, JavaScripty a další soubory, které se nacházejí v adresáři *WebContent* (viz. Příloha A). Do tohoto adresáře také patří adresáře s programy a projekty, které se zde nacházejí jako součást aplikace RemStat.

4.6.1 Pomocné JSP soubory

Mezi hlavní JSP soubory, které nebyli zmíněny v předešlých podkapitolách jsou JSP soubory uložené v adresáři *controls*. Tyto soubory slouží jako doplněk těch, které zajišťují hlavní funkci webové aplikace. Hlavním představitelem těchto JSP souborů je menu, které je součástí všech pohledů (kromě přihlášení).

Menu je tvořeno odkazy na akční třídy, patří sem také odkazy na nastavení programů. Tyto odkazy jsou generovány v závislosti na načtených programech aplikací. Dále jsou zde odkazy na nastavení jazyka a odhlášení. Tyto dvě akce pracují s daty o uživateli uloženými v session, tato data jsou uložena po dobu komunikace s uživatelem. Pokud se uživatel odhlásí, jsou tato data smazána a stránka je následně přesměrována na přihlášení. V případě změny jazyka je tato změna uložena do session a odtamtud brána, při každé další akci.

Dalším představitelem je JSP soubor, který je součástí všech tabulek se sortováním a stránkováním. V tomto souboru jsou nejprve načteny JavaScriptové soubory *paging.js* a *sorting.js*, potřebné k načtení JavaScriptu pro sortování a stránkování záznamů v tabulkách. Dále se zde nacházejí uložené hodnoty atributů potřebné k uložení a k poslání dat akční třídě. Poslední součástí tohoto souboru je JavaScript, který generuje obsah ovládacího panelu sortování, který se vkládá jako *tfoot* tabulky.

Posledníma dvěma JSP soubory je soubor, s obsahem loga stránky a soubor pro editaci částí příkazu pro spouštění programů, který obsahuje převážně JavaScript, u kterého je potřeba doplnit některé části kódu pomocí Struts 2 elementů (zejména pole adresy použitých obrázků odkazů).

4.6.2 Pomocné soubory pro generování HTML stránek

V adresáři *WebContent* se nacházejí adresáře se soubory obsahující kaskádové styly (adresář *css*), které určují vzhled všech stránek, dále obrázky (adresář *img*) použité na tlačítka a pozadí aplikace tvořící celkový obraz stránek a soubory JavaScriptu (adresář *js*), které byly použity ve většině stránek. JavaScript, v těchto souborech byl zejména použit pro změnu obrázku tlačítka při přejetí obrázku kurzorem myši nebo při stránkování a řazení záznamů v tabulkách.

4.6.3 Příložené programy a jejich data v projektech

Mezi data přiložená k aplikaci RemStat patří i adresář s programy GENFLOW a FLOW123D. V tomto adresáři se nalézají adresáře s kompletní sadou souborů, potřebných pro spuštění těchto programů.

Další složka nese název projekty. Do této složky se ukládají adresáře se soubory potřebné pro spuštění projektů. Také tyto adresáře jsou součástí aplikace RemStat a objeví se v nabídce při spuštění serveru Apache Tomcat. Tyto soubory se při prvním spuštění Tomcatu překopírují spolu s celou aplikací na místo, kam si Apache Tomcat ukládá všechny své aplikace.

4.7 Nastavení aplikace

Nastavení webové aplikace RemStat se dělí do několika úrovní. První úrovní je nastavení webových komponent v souboru *web.xml* (viz. Příloha D) jakými jsou filtry a mapování frameworku Struts 2. Další úrovní je nastavení komponent Struts 2 v souboru *struts.properties* (viz. Příloha C). Pro nastavení připojení k databázi se používá soubor *context.xml*. Dále je tu nastavení programů použitých v aplikaci. U nastavení aplikace je také zapotřebí počítat s nastavením použitých knihoven Struts 2, nastavení serveru Apache Tomcat, verze Javy a další.

V adresáři (balíčku Javy) s názvem *resources* se nachází nastavení programů a projektů, použité konstanty, soubor s českými a anglickými popisky a třídu, která obsahuje metody, pro načítání dat ze souborů s příponou *properties*

4.7.1 Soubory s popiskami

Tyto soubory se nacházejí v balíčku *remstat.resources* a obsahují popisky v různých jazykových překladech. Jako základní jazyk je nastavena čeština. Další soubory jsou odlišeny zkratkou jazyka uvedenou v koncovce názvu souboru oddělenou podtržítkem. Změnou lokalizace v aplikaci pro daného uživatele se pak přiřazují automaticky soubory, z kterých se bude načítat obsah popisek.

4.7.2 Nastavení programů a projektů

Nastavení projektů a programů je uloženo v souboru *settings.properties*. V tomto souboru se nastavují názvy adresářů, kde jsou uloženy spustitelné programy a adresář kam se ukládají projekty. U programů je zapotřebí nastavit název programu, jeho adresář, části příkazu kterým se spouští a pokud se program sám neukončí po

spuštění (např. program FLOW123D, který čeká po ukončení na stisk libovolné klávesy), je potřeba zadat ukončovací řetězec, který vypíše program po jeho ukončení do konzole.

V souboru *captions.properties* jsou uloženy popisky pro výchozí jazyk (čeština). Další jazykové verze (angličtina *captions_en.properties*) textů použitých v aplikaci jsou rozlišeny zkratkou v názvu podle jejich lokalizace. Každému jazyku odpovídá jeden soubor. Struts 2 si sám dokáže vybrat z jakého souboru si bude popisky brát a to podle toho, jaký jazyk má klient nastavený v prohlížeči. Pro jednodušší nastavení z pohledu klienta je v menu odkaz na změnu používaného jazyka a tato změna je uložena v session.

4.7.3 Soubory s nastavením aplikace

Mezi tyto soubory patří soubor *log4j*, který slouží pro logování. Tento soubor je důležitý zejména pro vývoj aplikace, ale také pro zprávu aplikace, kdy se mohou vyskytnout nečekané chyby. Výskyt těchto chyb se obvykle ukládá do logovacích souborů.

Pro nastavení parametrů frameworku Struts 2 slouží soubor *struts.properties*, ve kterém se nastavují velikosti načítaných souborů, což je velice důležité pro nastavení aplikace RemStat, protože se předpokládá, že se budou stahovat soubory v řádu několika GB. Dále se zde nastavuje přístup k *properties* souborům, obsahující popisky aplikace a další parametry Struts 2 aplikace (viz. Příloha C).

V adresáři WEB-INF se nachází soubor *web.xml*, který je velmi důležitý pro nastavení webové aplikace v Javě, nastavují se zde webové komponenty jakými jsou filtry, naslouchače, chybové a uvítací stránky a spousta dalších důležitých věcí (viz. Příloha D).

4.7.4 Použité knihovny Javy

V nastavení vývojového prostředí Eclipse aplikace RemStat (pod položkou Java build path) jsou přiřazeny knihovny Javy v adresáři WEB-INF/lib. Dále se zde nacházejí knihovny serveru Apache a knihovny Javy obsažené v adresáři s názvem *jre* nebo *jdk* verze 1.5 a vyšší potřebné ke spuštění jakéhokoli programu v Javě.

Výčet použitých knihoven Javy:

Knihovny Struts2:

- commons-collections.jar
- commons-io.jar
- commons-logging.jar
- freemarker.jar
- ognl.jar
- struts2-core.jar
- xwork.jar

Knihovna pro nastavení rozsahu platnosti atributů (např. session)

- struts2-scope-plugin-1.0.4.jar

Knihovna pro logování:

- log4j.jar

Knihovna potřebná pro stahování souborů:

- commons-fileupload.jar

Využití třídy `ByteInputStream` pro přímé posílání souborů bez nutnosti ukládání na disk u komprimace souborů uložené v knihovně:

- saaj-impl.jar

Závěr

Cílem práce bylo vytvoření aplikace v Javě, která bude spouštět programy na dálku a spravovat vstupní a výstupní soubory těchto programů. Pro realizaci tohoto úkolu si autor vybral framework Javy Struts 2. Bylo zapotřebí se podrobně seznámit jak s použitými webovými technologiemi, tak i s programy používanými na ústavu NTI.

Při tvorbě webové aplikace bylo zapotřebí navíc vytvořit správu uživatelů, kteří by měli k aplikaci přístup zvenčí a administrátorská práva pro uživatele, kteří by měli na starost chod aplikace. Správa uživatelských účtů v aplikaci je řešena pomocí jednoduchého připojení k MySQL databázi. Takové to ověřování uživatelů, by mohlo být do budoucna nahrazeno jiným zdrojem, jakým je například ověřování uživatelů v rámci TUL.

Mezi další drobnější vylepšení, by se dalo zařadit například odesílání e-mailů po skončení úlohy, zobrazit průběh výsledků posílání souborů na server, odhad trvání řešené úlohy pro některé z programů, různé statistiky vytížení serveru, správa volného místa vyhrazeného pro projekty atd.

Na závěr by chtěl autor poděkovat všem, kteří poskytli podporu a technickou pomoc při vývoji aplikace a to zejména vedoucímu práce panu Ing. Ottovi Severýnovi, Ph.D a za konzultace panu Ing. Daliboru Frydrychovi, Ph.D.

Použitá literatura

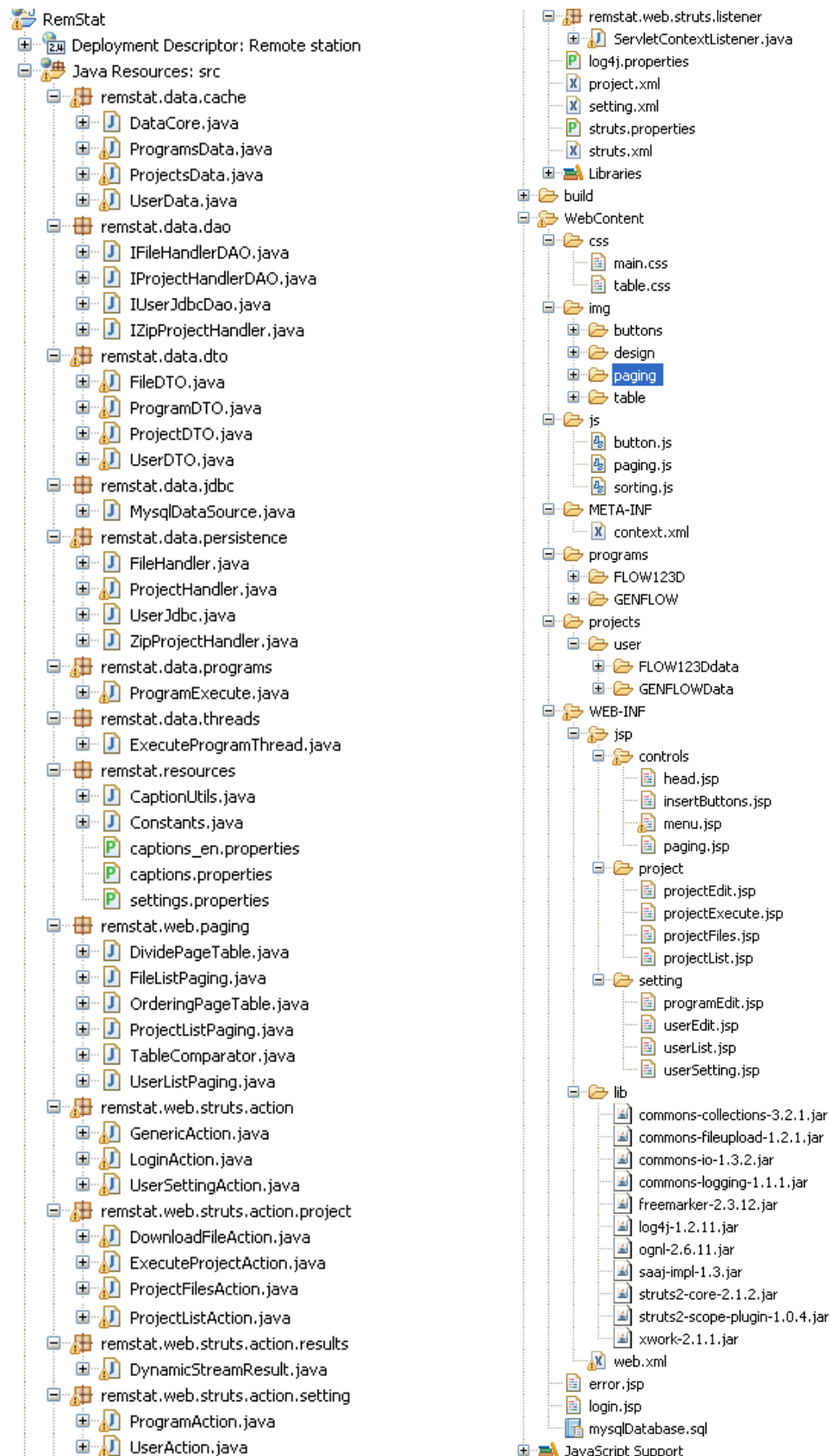
- [1] Maryška J., Severýn O., Tauchman M., Tondr.: Modelling of the groundwater flow in fractured rock – a new approach, In Proceedings of Algoritmy 2005 (K. Mikula, ed.), Slovak Technical University, Bratislava, pp. 113-122
- [2] Eckel B.: Myslíme v Jazyku Java. Grada, Praha, 2007.
- [3] John Pollock: JavaScript Příručka programátora, Softpress s.r.o 2001
- [4] Kris Jamsa, Suleiman Lalami, Steve Weakley: Programování na webu, UNIS 1996
- [5] Scott Isaacs: Dynamické HTML, Computerpress 1998
- [6] Dokumentace programů Flow123D a GENFLOW (Interní materiály ústavu NTI)
- [7] Chudl Cavaness: Programujeme Jakarta Struts. Nakladatelství O'Reilly & Associates, Inc 2003
- [8] Donald Brown, Chad Michael Davis, Scot Stanlic: Ebook Struts2 in action, Manning publication co. (<http://www.manning.com/>)

Zdroje na internetu:

- [9] Stránky výzkumného centra pro pokročilé sanační technologie a procesy
<http://centrum-sanace.cs.cas.cz>
- [10] Česká a anglická verze Wikipedie <http://wikipedia.org/>
- [11] Stránky Apache Jakarta projektu <http://jakarta.apache.org/>
- [12] Stránky Apache Struts projektu <http://struts.apache.org>
- [13] RoseIndia, Struts2 tutoriál <http://www.roseindia.net/struts/struts2/index.shtml>
- [14] Stránky Sun Microsystems, Java tutoriál <http://java.sun.com/docs/books/tutorial/>
- [15] Dušan Janovský, jak psát web <http://www.jakpsatweb.cz/>
- [16] Stránky Linuxsoft, JavaScript tutorial <http://www.linuxsoft.cz/javascript/>
- [17] David Měrka, článek Apache Tomcat – konfigurace, srovnání s jinými servery
<http://nb.vse.cz/~zelenyj/it380/eseje/xmerd04/Tomcat.htm>
- [18] Stránky Apache Struts projektu <http://struts.apache.org>

Přílohy

Příloha A – Adresářová struktura aplikace



Příloha B - Nastavení programů aplikace

Obsah souboru *settings.properties*:

```
#projects and programs directory names in WebContent directory
projects.directory.name=projects
programs.directory.name=programs

#all programs separated with comma(,):
programs = FLOW123D, GENFLOW

#Description:
program.FLOW123D.description=Program pro simulaci proudění podzemních
vod v saturovaném prostředí.
program.GENFLOW.description=Program pro modelování podzemního
proudění.

#If visibility is true program is visible, else hidden
program.FLOW123D.visible=true
program.GENFLOW.visible=true

#orders:
program.FLOW123D.order.1=flow123d.exe
program.FLOW123D.order.2=-c
program.FLOW123D.order.3=pokus.ini

program.GENFLOW.order.1=gen-flow.exe
program.GENFLOW.order.2=Cube.mmf

#If program does'nt finished, write finished strings separated with ;
program.FLOW123D.end.with= Bye, bye.
```

Soubor obsahuje nastavení použitých programů Flow123D a GENFLOW. Jsou zde nastaveny postupně jména adresářů projektů a programů, názvy programů (musí se shodovat s názvem jejich adresáře), popisků programů, zobrazení, základní příkazy a nepovinně ukončovací hesla, pokud se program sám neukončí. Zeleně jsou značeny komentáře.

Příloha C - Nastavení Struts 2

Obsah souboru *struts.properties*:

```
#file size upload limit in byte
struts.multipart.maxSize=100000000000

#captions
struts.custom.i18n.resources=remstat.resources/captions

struts.url.includeParams=none

struts.multipart.parser=org.apache.struts2.dispatcher.multipart.JakartaMultiPartRequest

### Change this to reflect which path should be used for JSP control
tag templates by default
struts.ui.theme=simple
```

Nejdůležitější částí tohoto nastavení je limit velikosti souboru v bytech, jaký může server přijmout (v tomto případě 100 GB). Další důležitou částí je cesta k souborům *captions*, kde se mohou nacházet různé jazykové verze textů, použitých na webových stránkách. V základním nastavení jsou tagy Struts 2 v JSP souborech obalovány různými částmi tabulky a tomu zabrání nastavením jednoduchého formátu, jak je tomu na posledním řádku tohoto souboru.

Příloha D - Nastavení souboru web.xml

Soubor *web.xml* v sobě zahrnuje nastavení webových komponent použitých ve webové aplikaci. V aplikaci RemStat má následující strukturu:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

  <display-name>Remote station</display-name>

  <resource-ref>
    <description>MySQL DB</description>
    <res-ref-name>jdbc/remstat</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>

  <listener>
    <listener-class>
      remstat.web.struts.listener.ServletContextListener
    </listener-class>
  </listener>

  <filter>
    <filter-name>struts2</filter-name>
    <filter class>
      org.apache.struts2.dispatcher.FilterDispatcher
    </filter-class>
  </filter>

  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <welcome-file-list>
    <welcome-file>login.jsp</welcome-file>
  </welcome-file-list>

  <session-config>
    <session-timeout>20</session-timeout>
  </session-config>

</web-app>
```

První řádek souboru reprezentuje použité kódování XML souboru. Uvnitř nastavení je název webové aplikace, odkaz na nastavení databáze, nastavení třídy posluchače (instance této třídy se vytvoří při startu serveru), nastavení filtru (Struts 2) a mapování Struts akcí. Do tohoto souboru lze začlenit například uvítací stránku (v tomto případě přihlášení uživatele), nastavit chybové stránky a další. Poslední položkou tohoto

souboru je nastavení trvání session v minutách, to je zejména důležité proto, jak dlouho uživatel bez žádné činnosti bude přihlášen.

Příloha E – Nastavení souboru context.xml

Tento soubor se nachází v adresáři *META-INF* a slouží k nastavení aplikace v rámci serveru a k nastavení zdrojů dat, tedy databáze MySQL. Pro připojení k databázi je zapotřebí vyplnit jméno uživatele, heslo, ovladač a adresu databáze. Důležitým parametrem je také jméno zdroje dat, s kterým se pak dále pracuje v rámci aplikace. Dále je možné nastavit parametry připojení.

Obsah souboru *context.xml*:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration
    2.0//EN" "http://struts.apache.org/dtds/struts-2.0.dtd">

<Context path="/RemStat" docBase="RemStat"
    reloadable="true" crossContext="true">

    <Resource name="jdbc/remstat" auth="Container"
        type="javax.sql.DataSource"
        maxActive="30" maxIdle="4" maxWait="1000"
        username="remstat" password=""
        driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:3306/test?autoReconnect=false&
        mp;useUnicode=true&characterEncoding=UTF8">
    </Resource>
</Context>
```