

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## SYSTÉM MONITOROVÁNÍ STAVU PLÁNOVACÍCH ÚLOH

BAKALÁŘSKÁ PRÁCE

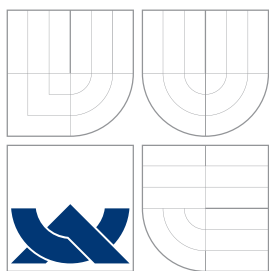
BACHELOR'S THESIS

AUTOR PRÁCE

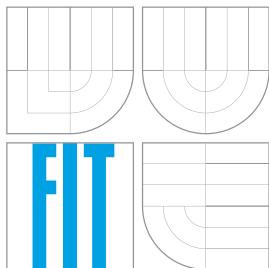
AUTHOR

MARTIN MAGA

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# SYSTÉM MONITOROVÁNÍ STAVU PLÁNOVACÍCH ÚLOH

PLANNING TASK MONITORING SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN MAGA

VEDOUCÍ PRÁCE

SUPERVISOR

ZDĚNEK LETKO SUPERVISOR.TITLE.P

BRNO 2014

## Abstrakt

Závěrečná práce prezentuje Systém monitorování stavu plánovacích úloh, který umožňuje najít optimální řešení pro NP-problém vzhledem k dostupnému času a dostupným algoritmy. V práci analyzujeme technologie k tvorbě uživatelského rozhraní pro tento systém s využitím open-source technologií, rovnako analyzujeme systém Optaplanner, který vykonává řešení plánovacích problémů použitím různých konfiguračních souborů, které definují zadání daného problému a použití algoritmy. Vypracovali sme návrh, který je uživatelský intuitivní a jednoduchý na pochopení s poměrně strmou učivou sa krivkou. Tento návrh sme předložili uživateli, který na základě vyplnění dotazníka poskytl spětnou vazbu na overení formálnosti a validity všech akcí. Zistili sme, že uživatelské rozhraní by mohlo obsahovat řadu rozšíření, které umožní uživateli jednoduchší orientaci v prostředí. Předpokládá sa použití tohoto projektu v rámci firemních požadavků, rovnako aj pro komunitní potřeby, které ho môžu ľubovoľne upravovať. Výsledok je řešení problematiky Systému monitorování stavu plánovacích úloh je uživatelské rozhraní je uživatelské rozhraní, které intuitivně umožňuje pracovat v rámci organizace, rovnako aj sledovat stav a vytvářet nové úlohy. Rovnako je možné definovat vlastní úlohy a overit si řešení různých NP problémů, které sú známe.

## Abstract

Výtah (abstrakt) práce v anglickém jazyce.

## Klíčová slova

Java EE 6, Java, Java Beans, Java Server Faces, Monitorování, Twitter, Bootstrap, Optaplanner, Webová služba, Enterprise Java Bean, JBoss, Rich Faces, Model, Komponenta, Maven, Arquillian, Plánování, MySQL, Uživatel, Uživatelská rola, Obmedzení, Plánovací problém, Úloha, Martin Večera, Zdeněk Letko, Red Hat .

## Keywords

Java EE 6, Java, Java Beans, Java Server Faces, Monitoring, Twitter, Bootstrap, Optaplanner, Web Services, Enterprise Java Bean, JBoss, Rich Faces, Model, Component, Maven, Arquillian, Planning, MySQL, User, User Role, Constraint, Planning problem, Martin Večera, Zdeněk Letko, Red Hat .

## Citace

Martin Maga: Systém monitorování stavu plánovacích úloh, bakalářská práce, Brno, FIT VUT v Brně, 2014

# System monitorování stavu plánovacích úloh

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Zdeňka Letka a Martina Večeře

.....  
Martin Maga  
26. dubna 2014

## Poděkování

Veľmi rád by som poďakoval za vedenie mojej bakalárskej práce pánovi Zdeňkovi Letkovi a pánovi Martinovi Večeřovi, ktorý mi poskytl rady a podali pomocnú ruku vždy, keď som narazil na problém.

© Martin Maga, 2014.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

# Kapitola 1

## Úvod

V úvode by som Vás rád krátko zoznámil s témou svojej bakalárskej práce, ktorá sa venuje téme Systému monitorovania stavu plánovacích úloh. Tento systém sa skladá z užívateľského rozhrania, ktoré je vytvorené prostredníctvom Java open-source technológií, ktoré bežia na javovskom serveri. Preto sa zameráme na všetky technológie, ktoré potrebujeme pre správne pochopenie a následnú implementáciu užívateľského rozhrania pre tento systém. Rovnako bližšie vysvetlím použitý open-source java server, ktorý je nevyhnutý pre beh aplikácie. Rovnako bude treba správne pochopiť celý plánovací open-source plánovací systém Optaplanner, pre ktoré je užívateľské rozhranie určené. Tento systém umožňuje spúšťať definované užívateľsky definované problémy, ktoré systém prostredníctvom správnych algoritmov naplánuje a dospeje k správnomu riešeniu vhl'adom na dostupný čas a dostupné algoritmy. Rovnako sa budem venovať testovaniu a vyhodnoteniu užívateľského rozhrania z hľadiska intuitívnosti, jednoduchosti a splnenia všetkých formálnych požiadavok. Rovnako uvediem testy potrebné k overeniu správnej činnosti aplikácie a použitý framework. Toto téma bolo vybraté z dôvodu môjho osobného záujmu o open-source technológie, rovnako o možnosti ich využitia a veľmi ma zaujala možnosť verejná zdieľania projektu medzi open-source komunitov, ktorá mi môže poskytnúť spätnú vazú, resp. môže túto prácu využívať v praxi, čo cieľ, ktorý by som rád prostredníctvom tejto práce dosiahol.

Dopísať podľa vyhodnotenie podľa testovania a dotazníka.

## Kapitola 2

# Java Enterprise edition 6

### 2.1 Motivácia

V posledných rokoch prevláda tendencia tvorby komplexných informačných systémov, ktoré spracovávajú veľké množstvo dát. Preto sa zvyšuje tlak na vývojárov na tvorbu prostriedkov, ktoré dokážu takéto systémy ľahko a rýchlo vytvárať. Jedným z takýchto prostriedkov je platforma Java Enterprise Edition (Java EE), ktorá použijeme vo verzii 6, ktorá nám postačuje pre implementáciu aplikácie. Java EE je platformou, ktorá rozširuje základné možnosti jazyka Java o enterprise technológie, ktoré umožňujú tvorbu komplexnejších systémov, ktoré bežia na rôznych aplikačných serveroch. Jazyk Java je open source, rovnako ako aj všetky poskytnuté technológie, preto som sa rozhodol využívať tento programovací jazyk. Platforma Java EE je ďalej tvorená špecifikáciami pre podporu webových technológií, webových aplikácií, podnikovej logiky a . . . . Nám budú postačovať prvé 3 špecifikácie tejto platformy, ktoré rozobereme v nasledujúcej časti spolu s technológiami, ktoré ich reprezentujú. Na základe Java boli implementované boli implementované rôzne Java EE kontajnery, ktoré sú potrebné pre správu a beh aplikácie. My sa zameriame na open-source riešenia z dôvodu šírenia projektu ako open-source. Ďalšou výhodou použitia tejto platformy je použitie anotácií, ktoré zjednodušujú implementáciu výslednej aplikácie a spôsobia konfiguráciu danej komponenty pri nasadzovaní a za behu. Rovnako je zdôraznený princíp POJO (Plain Old Java Objects)[?] a zjednodušenie tvorby balíkov. V poslednom rade musí spomenúť princíp „Convention over configuration“, ktorý minimalizuje počet konfigurácií pre daný projekt. V nasledujúcej časti rozoberiem všetky potrebné špecifikácie doplnené o rôzne frameworky, bez ktorých by sa vývojový cyklus aplikácie nezaobišiel.

### 2.2 Špecifikácia platformy

Java EE predstavuje platformu určenú na vývoj webových a podnikových aplikácií[?]. Tieto aplikácie sú viacvrstvové z dôvodu lepšej prenositeľnosti, nasaditeľnosti a modifikovateľnosti. Frontend, predstavujúci užívateľské rozhranie a logiku na jeho ovládanie, pozostáva z webových frameworkov, stredná vrstva poskytuje bezpečnosť a transakcie.

Najnižšia vrstva poskytuje pripojenie k databázam. Java EE je platformou, ktorá poskytuje širokú škálu aplikačných programových rozhraní(API), ktoré zjednodušujú, zkracujú a znižujú komplexnosť vývoja a nasadenia výslednej aplikácie. Jej vývoj neustále napreduje a je spravovaný Java Community process(JCP). Aplikácie pre platformu Java EE sú vyvíjané prostredníctvom API, ktoré táto platforma poskytuje. Medzi tieto API patrí napríklad: Java Server Faces, Java Persistence API, Enterprise Java Bean, . . . . Behovým prostredím sú aplikačné servery, ktoré pozostávajú zo servletov, JavaServer Pages, EnterpriseJavaBeans a iných technológií, ktoré sa starajú o správu aplikácie a jej nasadenie. Keďže je Java označovaná ako multiplatformovaná musí poskytovať prostriedky, ktoré je možné nasaďovať naprieč rôznymi aplikačnými serverami. Medzi takéto prostriedok patrí bezpečnosť, ktorá je v riešená pomocou prístupových pravidiel, ktoré sú interpretované za behu aplikácie. V ďalších kapitolách si rozoberieme aplikovaný model jazyka, ktoré je veľmi dôležitý pre pochopenie princípu činnosti aplikácií vyvinutých touto platformou. V ďalšej kapitole rozobereme aplikačný model platformy Java EE.

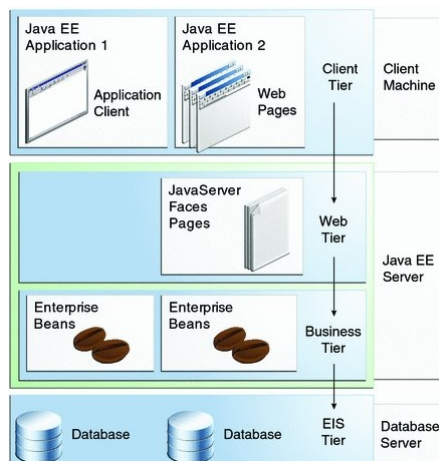
## 2.3 Aplikačný model

Java EE definuje aplikácie, ktoré sú viacvrstvové(multitier). Aplikačná logika je rozdelená medzi komponenty podľa ich funkcie[?]. Jednotlivé komponenty sa následne rôzne inštalujú na rôzne zariadenia v závislosti, do ktorého stupňa patria(Keďže každý stupeň môže byť fyzicky na inom aplikačnom serveri). Jednotlivé stupne sa skladajú z rôznych komponent, pričom stupne sú rozdelené nasledovne:

- Klientský stupeň sa skladá z klientských komponent, ktoré bežia na klientskom počítači
- Java EE server sa skladá z webových a podnikových komponent, ktoré bežia na Java EE serveri
- Databázový server ktorý sa skladá z enterprise information system komponent

Typicky beží medzi klientskom a databázou častou viac-vláknový Java EE server, ktorý býva označovaný skratkou EIS. Viacstupňovérozloženie môžete názorne vidieť na obrázku č. ?? . Java EE aplikácia beží na klientskej stanici, býva obvykle reprezentovaná tenkým klientom(webovým prehliadačom), nazývaným „thin client“(pretože sa nedotazuje priamo na databázový server), alebo hrubým klientom, do ktorého je čiastočne vložená logika aplikácia. Klient môže byť reprezentovaný ako webový alebo aplikačný. Typický webový klient pritom pozostáva z: Webové prehliadača, ktorý zobrazuje stránky a dynamických webových stránok pozostávajúceho z rôzneho značkovaciehojazyka(HTML,XHTML), ktoré sú generované webovými komponentami. Zložitá logika je vykonávaná strednou vrstvou, pričom klient len posiela požiadavky na Java EE server a ten prípadne sa dotazuje databázové servera a následne predáva výsledok. Klient môže poskytovať aj bohatšie užívateľské rozhranie, ktorá býva vytvárané technológiou Swing alebo Abstract Window Toolkit[?], po prípade sa vyskytuje aj prístup prostredníctvom príkazového riadku. V





Obrázek 2.1: Model Java EE [<http://docs.oracle.com/javasee/6/tutorial/doc/>]

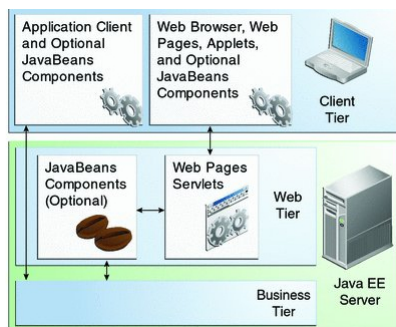
strednej časti obrázku sa nachádza Java EE server, na ktorom môžu bežať rôzne technológie v závislosti od požiadavky výslednej aplikácie a možností daného servera. Stredná vrstva sa ešte delí na webový stupeň, ktorý je prezentovaný technológiami JavaServer Faces a Pages. Druhá časť strednej vrstvy takzvaná podniková vrstva býva reprezentovaná technológiu EnterpriseJava Beans, ktoré vytvárajú logiku aplikácie. Java EE server môže byť reprezentovaný, ešte okrem spomenutých technológií, rôznymi inými dostupnými technológiami, v závislosti od možnosti aplikačného servera, ktorý môže byť open-source (JBoss, Tomcat, GlassFish) alebo komerčný (IBM WebSphere, BEA WebLogic), ten obsahuje rôzne komponenty, ktoré so sebou rôzne komunikujú a interagujú na požiadavky klienta a na druhej strane komunikujú s databázovým systémom a starajú sa o beh aplikácie a jej nasadenie. Posledná časť predstavuje databázový server, ktorý obsahuje dáta, ktoré klient požaduje pri svojom požiadavku, tento server sa nazýva "EIS". Pre prístup k nemu sa používa buď nový prístup, ktorý sa nazýva objektovo-relačné mapovanie, ktoré využíva rozličné ovládače pre prístup k databázovému systému (napr. JDBC).

V nasledujúcej kapitole sa zameriame na technológie strednej vrstvy, ktoré sú nevyhnutné pre tvorbu a pochopenie činnosti navrhnutej aplikácie.

## 2.4 Webové komponenty

Java EE webové komponenty sú softwarové komponenty, ktoré spracovávajú prichádzajúci HTTP požiadok a poskytujú naň odpoveď. Všetky Java EE webové komponenty sú postavené na servletoch. Servlety sú javovské triedy, ktoré dynamicky spracovávajú požiadavky a tvoria odpovede. Súčasťou servletov alebo webových stránok, ktoré sú technológiu JavaServer Faces technológiu (JSF) and JavaServer pages (JSP). Servlety podporujú automatickú správu sedenia, prostriedky pre vytváranie a ničenie servletov. Technológia JavaServer Faces a JavaServer Pages podporujú spracovanie užívateľských vstupov a ich

predanie a spracovanie podnikovou logikou. Pre implementáciu výslednej aplikácie bola použitá JavaServer Faces technológia, ktorá poskytuje dostatočné možnosti pri tvorbe webových stránok. V rámci webových komponent spomeniem technológiu, ktorá je potrebná pre pochopenie funkčnosti aplikácie. Ide o technológiu Web Service.



Obrázek 2.2: Webové komponenty [<http://docs.oracle.com/javase/6/tutorial/doc/>]

Na nasledujúcom obrázku č.?? je ukázaný princíp fungovania webových komponent. V hornej časti obrázku sa nachádza klientská vrstva, ktorá obsahuje buď len webový prehliadač po prípade Applety alebo JavaBean komponenty, ktoré čiastočne obsahujú logiku aplikácie. Na druhej strane môže byť klient reprezentovaný aplikačným klientom, ktorý obsahuje úplnú prezentačnú logiku aplikácie a teda v tom prípade, odpadá potreba spracovania vstupov po prípade nejaké generovania html stránky. Takýto klient komunikuje už len priamo s Java EE serverom, konkrétne podnikovým stupňom, ktorý implementuje zvyšnú logiku aplikácie a je reprezentovaný technológiou Enterprise Java Beans. V prípade, že máme k dispozícii tenkého klienta, klient komunikuje prostredníctvom webového prehliadača s HTML alebo XHTML stránky, ktoré sú vytvorené technológiou, ktoré spracovávajú požiadavky od klienta (vstupy) a následne komunikuje s podnikovým stupňom, ktorý obsahuje logiku reprezentovanú Enterprise Java Beans technológiou, ktorý následne môže komunikovať s databázovým serverom. Odpoveď je následne „predaná“ stránkam vytvorené prostredníctvom JavaServer Faces alebo JavaServer Pages technológiou a následne zobrazená užívateľovi v podobe výstupu na webovú stránku. V nasledujúcich dvoch podkapitolách sa bližšie pozrieme na technológie JavaServer Faces a JavaServer Pages.

### 2.4.1 JavaServer Faces

JavaServer Faces (JSF) je framework pre tvorbu užívateľských rozhraní webových aplikácií. Tento framework beží na Java EE serveri. Tento framework poskytuje sa skladá z ďalšieho frameworku, ktorý obsahuje rôzne komponenty pre zobrazenie informácií, užívateľských vstupov, spracovanie udalostí, navigáciu medzi stránkami. JSF vytvára aplikácie na základe MVC - Model-View Controller. Aplikácia, ktorá je vytvára týmto frameworkom pozostáva z webových stránok, grafických komponent, sadou komponent naviazané