

**České vysoké učení technické**

*Fakulta elektrotechnická*

**Návrh**

*Systém elektronických voleb*

*E-volby*

## Obsah

Architektura aplikace.....	3
Model tříd.....	5
CreatingElection.....	5
GenerateResult.....	5
Nominating.....	5
Voting.....	5
Použité technologie a knihovny.....	6
Glassfish.....	6
MySQL.....	6
Netbeans.....	6
Google Code.....	6
Google Wave.....	6

## Architektura aplikace

Diagramy struktury projektů jsou uloženy v souboru [ Logical architecture.pdf ]

Jak už bylo v bezpečnostních rizicích zmíněno, jednotlivé komponenty musí běžet na různých serverech. Proto je celý tento systém rozdělen do několik samostatných projektů, které je možné nahrát samostatně na různé servery. V rámci lokálních voleb je samozřejmé, že tyto jednotlivé moduly budou nahrány na jeden server ( viz Diagramy nasazení ).

Prvním modulem je Controller. Tento modul ovládá celkovou logiku aplikace. Má své webové rozhraní, ke kterému se připojují, jak administrator, tak komisaři a voliči. Aby systém rozeznal, kdo se k tomuto rozhraní hlásí, je zde použito J2EE rolí. Každý uživatel má v realmu u svého loginu a hesla uloženou J2EE roli, která říká jaké má daný uživatel oprávnění v systému a do kterých oblastí ho tento systém autorizuje. Z toho vyplývá, že v naší aplikaci se využívají tři role: admin, commissioner a voter.

Dále je tento modul rozdělen na tři vrstvy. Jsou to vrstvy prezentační ( webové rozhraní ), aplikační ( enterprise beany ), datová ( databáze ).

Prezentační vrstva je tvořena JSP stránkami a managed beanama, které slouží jako prostředník mezi stránkami a aplikační vrstvou.

Aplikační vrstvu tvoří entitní třídy a enterprise beany, které jsou schopné číst a ukládat data do relační databáze. Zde jsou naimplementovány všechny metody právě pro tuto činnost. Například beana CreateElectionSessionBean má v sobě implementovány metody na práci s volbami (vytváření nových voleb, přidávání komisařů do voleb, vytváření volebních událostí), ale také tato beana slouží pro načítání dat z databáze ( např. Komisařovi volby, volby, ve kterých může volič volit, atd. ). Všechny beany jsou popsány v diagramech v souboru určenému pro tuto část.

Pro tento modul potřebujeme tabulky Candidate, Commissioner, Election, ElectionEvent, ElectionResult, Vote, Voter.

Dalším modulem je Applet skrz který může volič hlasovat v tomto volebním systému. Tento applet je vytvořen v jazyku JAVA fx a má jednoduchou strukturu. Tento applet se skládá ze dvou vrstev. V prezentační vrstvě je vidět seznam kandidátů a dvě tlačítka u každého z nich (pro volbu zvolit kandidáta či nikoli). Dále jsou v zápatí další dvě tlačítka jedno slouží pro odeslání hlasu a druhé na vypnutí appletu. V aplikační vrstvě máme dvě třídy, první z nich je Receiver, která slouží pro získání kandidátů dané volební události a třída Sender slouží pro odeslání volebního lístku.

Třetím modulem je Validátor. Tento modul se skládá pouze ze dvou vrstev. Narozdíl od Controlleru tento modul nepotřebuje žádné webové rozhraní protože se veškerá jeho funkcionality nastavuje z modulu Controller. V aplikační vrstvě tohoto modulu jsou opět (stejně jako u Controlleru) entitní třídy a enterprise beana. Tento modul má

opět svou vlastní databázi.

Posledním modulem tohoto systému je modul Counter. Tento modul je hodně podobný modulu validátor. Skládá se opět ze dvou vrstev. V té aplikační se opět nacházejí entitní třídy a enterprise beany. Tento modul má opět svoji vlastní databázi.

## **Model tříd**

Diagramy modelu tříd jsou uloženy v souboru [ Class model.pdf ].

### ***CreatingElection***

Tvorba volební události, její spojení s komisí a následně i s komisáři.

### ***GenerateResult***

Výpočet výsledků volební události.

### ***Nominating***

Nominace kandidátů pro volební událost s jejich programem a schvalovacím řízením.

### ***Voting***

Samotný průběh volby voliče.

## Použité technologie a knihovny

### **Glassfish**

Aplikační server Glassfish pro platformu JavaEE ve verzi 2.1.1 jsem přejali od našich předchůdců pracujících dříve na tomto projektu. Sami vidíme mezi jeho nespornými výhodami skutečnost, že se jedná o open source. Tento server je navíc psán jako ukázková implementace nových rysů v poslední specifikaci platformy JavaEE, tudíž máme zajištěnu podporu nejmodernějších technik.

### **MySQL**

Stejně jako v případě Glassfish jsme i my zůstali u MySQL jako naši předchůdci. MySQL se vyznačuje velmi jednoduchou administrací a je velmi přizpůsobivé a multiplatformní, což nám umožní soustředit se více na samotný projekt.

### **Netbeans**

Jako vývojové prostředí jsme zvolili Netbeans, pro jednoduché prostředí a naši dobrou znalost tohoto nástroje.

Krom toho Netbeans skvěle zvládá spolupráci s Glassfish serverem a dokáže snadno komunikovat s MySQL databází.

### **Google Code**

Rozhodli jsme se projekt hostovat na Googlu, konkrétně na jeho službě Google Code. Mezi hlavní důvody vedoucí k tomuto rozhodnutí jistě patří dobré jméno Googlu a jistota, že služby jím nabízené jsou kvalitní, vždy dostupné a nehrozí v podstatě žádná ztráta dat. Navíc nám služba Google Code zastřešuje hned několik nástrojů nezbytných k vývoji projektů – SVN, Issues Tracker a Wiki.

### **Google Wave**

Pro komunikaci uvnitř týmu se nám osvědčil z již dřívějších projektů Google Wave, který nabízí velmi efektivní, snadné a rychlé řešení vzájemné komunikace mezi více uživateli. Hlavními výhodami je real-timeová komunikace, možnost vytvářet neomezené množství vláken vzájemně se vnořujících do sebe v rámci jednoho „Wavu“ či možnost editovat všechny příspěvky. Samozřejmostí je uchovávání historie změn, takže není problém dohledat jakékoliv změny zpětně a sledovat plnění „Wavu“.

Krom toho Google Wave vždy upozorní na nový příspěvek e-mail, případně se dá do Google Chrome či Firefoxu doinstalovat doplněk, který indikuje nové zprávy. Komunikace je tedy vskutku blesková.