

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2014

Jakub HROMÁDKO

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Shlukování poslanců
Jakub Hromádko

Bakalářská práce
2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub Hromádko**
Osobní číslo: **I08059**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Shlukování poslanců**
Zadávající katedra: **Katedra informačních technologií**

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

Úkolem shlukové analýzy je zařadit jedince do skupin podle jejich podobností a rozdílností a dále také redukovat informace o jedincích.

Jedním z důvodů pro redukcí dimenze dat jsou omezené možnosti grafického zobrazení mnohorozměrných datových souborů.

Pro shlukování byl navržen velký počet metod. Jejich principy se liší.

Některé metody vychází z matice vzdáleností (hierarchické metody) a jiné metody vychází přímo ze vstupní matice (metody rozkladu).

Existují ještě alternativní přístupy z oblasti neuronových sítí.

Shlukování poslanců či senátorů v této práci bude provedeno na základě jejich hlasování.

Cílem práce bude sestavení programu umožňujícího načtení dat o hlasování poslanců resp. senátorů z webových stránek parlamentu a senátu

a měření názorových vzdáleností mezi jednotlivými poslanci.

Hierarchické metody užívají dva přístupy - divizioní a aglomerativní. U divizionního přístupu předpokládáme, že všechny objekty tvoří jeden shluk, ten je pak dělen do více shluků až každý objekt tvoří jeden shluk.

U aglomerativního přístupu každý objekt tvoří jeden shluk a hledáme nejpodobnější jedince. Jejich sloučením dostáváme nový shluk a snižujeme počet shlukovaných jedinců o jedničku.

Postup shlukování se zakresluje pomocí dendrogramu.

Bude zajímavé zjistit, zda vzniklé shluky budou odpovídat rozložení politických stran. Může být také měřena loajalita členů strany např. pomocí vzdálenosti k předsedovi parlamentního klubu.

K měření vzdálenosti existuje velký počet měř podobnosti resp. nepodobnosti (euklidovská, Čebyševova, Manhattanská, Lanceyowa-Williamsova, Hammingova vzdálenost).

Při zpracování dat o hlasování bude třeba nalézt nejvhodnější míru podobnosti v kontextu s navrženými kvantitativními proměnnými pro zápis výsledků hlasování resp. neúčasti poslance a případné transformace.

Z metod rozkladu lze použít např. metodu k průměrů. Číslem k se určuje požadovaný počet výsledných shluků. Tímto číslem může být počet politických stran.

Hlavní myšlenka je definovat k centroidů, jeden pro každý shluk. Zde se nabízí použít jako centroidy předsedy poslaneckých stran.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

Řezánková, H., Húsek, D., Snášel, V.: Shluková analýza dat, Kamil Mařík-Profesional Publishing, Praha, 2007.

Hebák, P., Hustopecký, J., Malá, I.: Vicerozměrné statistické metody 3, Informatorium, Praha, 2005.

Vedoucí bakalářské práce:

Mgr. Jaroslav Marek, Ph.D.

Katedra matematiky a fyziky

Datum zadání bakalářské práce: **20. prosince 2013**

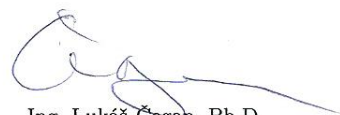
Termín odevzdání bakalářské práce: **9. května 2014**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2014

Prohlášení autora

Prohlašuji:

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 7.5.2014

Jakub Hromádko

Poděkování

Na tomto místě bych rád poděkoval zejména panu Mgr. Jaroslavu Markovi, Ph.D. za jeho cenné rady, připomínky a názory, které mi velmi pomohly při tvorbě této práce.

ANOTACE

Práce je věnována shlukové analýze, která je aplikována na datech tvořených z výsledků hlasování poslanců a senátorů České republiky v období 11/2013-04/2014. Cílem shlukování je najít v datovém souboru podobné jedince a zařadit je do skupin. Pomocí webové aplikace jsou zpracována data o hlasování poslanců z webových stránek poslanecké sněmovny a senátu České republiky. Provedené shlukování poslanců je graficky interpretováno pomocí diagramu.

KLÍČOVÁ SLOVA

shluková analýza, hlasování poslanců, hierarchické shlukování, ploché shlukování

TITLE

Clustering of deputies

ANNOTATION

The work is devoted to cluster analysis, which is applied to data consisting of the results of the vote of deputies and senators Czech Republic in the period 11/2013-04/2014. The goal of clustering is to find a data file similar to the individual and to classify them into groups. Using web application are processed data of voting deputies from the website Chamber of Deputies and the Senate of the Czech Republic. Performed clustering of deputies is graphically interpreted using the diagram.

KEYWORDS

cluster analyse, voting Members of Parliament, hierarchical clustering, flat clustering

OBSAH

SEZNAM ZKRATEK	9
SEZNAM ILUSTRACÍ A TABULEK.....	10
ÚVOD.....	11
1 SHLUKOVÁ ANALÝZA	12
1.1 Vstupní datová matice.....	12
1.2 Měření podobnosti a vzdálenosti	12
1.3 Míry vzdáleností.....	13
1.3.1 Příklad výpočtu vzdáleností.....	14
1.4 Hierarchické shlukování.....	14
1.4.1 Algoritmus hierarchického shlukování.....	14
1.4.2 Způsob hodnocení vzdáleností.....	15
1.4.3 Praktický příklad.....	16
1.4.4 Prezentace výsledku hierarchického shlukování	17
1.5 Nehierarchické (ploché, flat) shlukování	18
1.5.1 Algoritmus nehierarchického shlukování	18
1.5.2 Praktický příklad.....	18
2 APLIKACE.....	20
2.1 Teoretický základ	20
2.1.1 HTML	20
2.1.2 CSS	21
2.1.3 PHP	22
2.1.4 Databáze.....	22
2.2 Funkční požadavky	23
2.3 Softwarové požadavky	23
2.4 UML use case diagram.....	24
2.5 UML activity diagram.....	25

2.6	Návrh struktury databáze	26
2.6.1	Entity.....	26
2.6.2	ER diagram	29
2.6.3	Implementace databázového modelu.....	30
2.7	Vývoj aplikace	30
2.7.1	Uživatelské rozhraní a grafické prvky	31
2.7.2	Vytváření aplikace a funkcionality	32
3	INSTALACE	40
3.1.1	Databáze.....	40
3.1.2	Aplikace	40
3.1.3	Hlavní konfigurační soubory	40
4	UŽIVATELSKÁ PŘÍRUČKA	41
5	VÝSLEDKY APLIKACE	43
6	ZÁVĚR	47
7	POUŽITÁ LITERATURA	49
8	PŘÍLOHY	51

SEZNAM ZKRATEK

AJAX – Asynchronous JavaScript and XML

API – Application Programming Interface

CSS – Cascading Style Sheets

ER diagram – Entity–relationship diagram

HTML – Hypertext Markup Language

PHP – Hypertext Preprocessor

SQL – Structured Query Language

UML – Unified Modeling Language

XPath – XML Path Language

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1 – Znázornění vzdáleností objektů ve dvou dimenzích	13
Obrázek 2 – Zobrazení stromového diagramu se zvýrazněnými shluky	17
Obrázek 3 – Přiřazování objektů k vzorovým objektům ve dvourozměrném prostoru	18
Obrázek 4 – UML case diagram	24
Obrázek 5 – UML activity diagram	25
Obrázek 6 – ER diagram	29
Obrázek 7 – PhpMyAdmin	30
Obrázek 8 – Webová stránka s využitím Twitter Bootstrap frameworku	31
Obrázek 9 – Uživatelské rozhraní s odesláním požadavku na akci	38
Obrázek 10 – AJAXem vložený diagram	39
Obrázek 11 – Diagram zpracované flat analýzy	42
Obrázek 12 – Způsob seskupení objektů pro interpretaci výsledku	43
Obrázek 13 – Interpretace hierarchické analýzy s ikonami postav odlišené barvou	43
Obrázek 14 – Interpretace tří plochých analýz pro jeden vzorový objekt	44
Obrázek 15 – Interpretace jedné nehierarchické analýzy pro tři vzorové objekty	45
Tabulka 1 – Výpočet měř vzdáleností	14
Tabulka 2 – Entity a jejich atributy	27
Tabulka 3 – Vzdálenosti nejvíce a nejméně loajálních osob ke vzorovým objektům	45

ÚVOD

Tato práce bude věnována shlukové analýze a vytvoření aplikace umožňující provádění hierarchického a nehierarchického shlukování. Hlasování poslanců, respektive jejich statistiky, jsou velmi vhodným typem vstupních dat pro metody shlukové analýzy, neboť na výsledku můžeme zřetelně pozorovat podobnosti, respektive nepodobnosti jednotlivých poslanců při hlasování.

Shluková analýza umožňuje také porovnávat jednotlivce a jejich vzdálenost od předem určených vzorů. V případě poslanců budeme zkoumat vzdálenost k předsedům politických stran a v případě senátorů budeme zkoumat vzdálenost k předsedům senátorských klubů. Metody shlukové analýzy umožňují zkoumat, zda jednotlivé shluky vzniklé hierarchickým shlukováním odpovídají rozložení politických stran, respektive senátorských klubů.

Součástí teoretické části bude popis základních principů a metod shlukové analýzy. Vytvořená aplikace bude načítat data o hlasování poslanců a senátorů z webových stránek poslanecké sněmovny www.psp.cz a webových stránek senátu www.senat.cz. Aplikace umožní uživateli jednoduše zvolit požadované metody shlukové analýzy a po jejím zpracování bude výsledek prezentován jednoduchou a přehlednou formou.

1 SHLUKOVÁ ANALÝZA

Základním principem shlukové analýzy je zařazování objektů do skupin neboli shluků tak, aby shluk obsahoval nejpodobnější objekty. Zařazování jedince ze skupiny všech objektů do shluku provádíme na základě tzv. měř podobnosti (vzdálenosti), do shluku přiřazujeme dva nejpodobnější objekty. Může se stát, že do shluku bude zařazen jen jeden jedinec. Zkoumanými objekty rozumíme jakýkoliv objekt, kterému je možno přiřadit množinu charakteristických hodnot. Takovým objektem mohou být například rostliny s charakteristickou vlastností délka okvětního lístku, délka pestíku a výška rostliny, zvířata s vlastností barva srsti, váha, rychlost, délka těla, doba březosti, váha potomků. V neposlední řadě zkoumanými objekty mohou být také nejrůznější statistiky sportovních výsledků, burzovní výsledky, firemní informace, atd. Shluková analýza neřeší pouze podobnosti objektů, k dalším problémům shlukové analýzy patří také kritéria pro shlukování, stanovení počtu výsledných shluků a interpretace výsledku analýzy. [1]

Mezi základní principy shlukové analýzy řadíme hierarchické shlukování a nehierarchické (ploché) shlukování. Hierarchické shlukování vychází z jedné množiny vektorů (vstupní matice dat). Vypočítáme matici vzdáleností (podobnosti) a spojujeme nejpodobnější jedince a shluky. Při algoritmu nehierarchického shlukování je dána množina vektorů (vstupní matice dat) a dále množina vzorových objektů (centroidů), ke kterým přiřazujeme objekty z množiny vektorů.

1.1 Vstupní datová matice

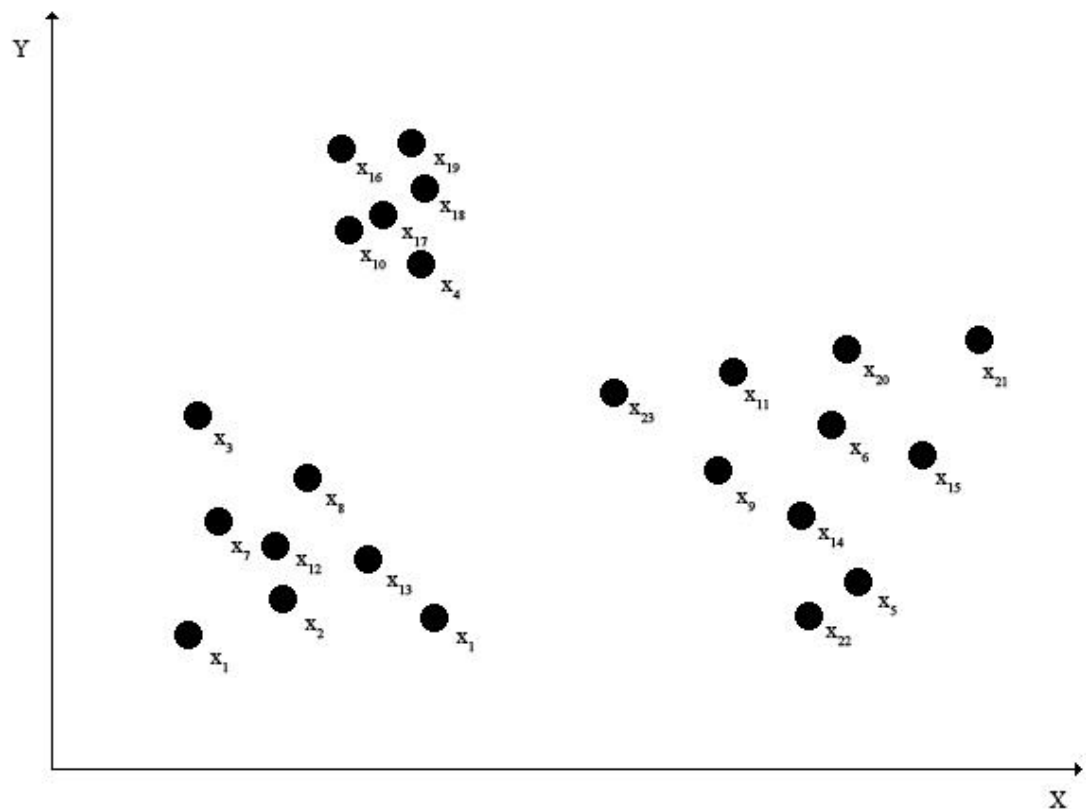
Základem statistické analýzy jsou m -rozměrná pozorování zkoumaných objektů, výsledkem pozorování je množina statistických jednotek či datových jednotek různých typů. Velikost množiny jednotek označujeme jako n . Ve výsledku tvoří základní asymetrickou matici pro vícerozměrnou statistickou analýzu označenou S a velikosti $n \times m$. [1]

1.2 Měření podobnosti a vzdálenosti

Prvním krokem po sestavení vstupní datové matice S je výběr vhodného způsobu hodnocení podobnosti každé dvojice objektů x_i a x_j ze vstupní matice S . K posuzování podobností slouží při shlukové analýze právě míry vzdáleností respektive podobností. Při realizaci shlukovacího algoritmu se provede výpočet příslušné míry vzdáleností (podobností) dvojic x_i a x_j . Výsledek každé vzdálenosti pak zapíšeme jako $S(x_i, x_j)$ nebo S_{ij} . Platí, že $S_{ij} = S_{ji}$ a $S_{ii} = 0$. [1], [2]

Výsledkem celého výpočtu je symetrická matice vzdáleností, kterou označíme jako D o velikosti $n \times n$, jedná se tedy o symetrickou matici, jejíž diagonálu tvoří hodnoty 0. Platí, že:

- $D_{ij} \geq 0$ – hodnoty v matici vzdáleností jsou vždy nezáporné,
- $D_{ij} = D_{ji}$ – matice je symetrická podle diagonály,
- $D_{ii} = 0$ – v matici vzdáleností jsou hodnoty na diagonále rovny 0. [1]



Obrázek 1 – Znázornění vzdáleností objektů ve dvou dimenzích, [zdroj: vlastní]

1.3 Míry vzdáleností

Mezi nejznámější míry vzdáleností patří:

1. Euklidovská vzdálenost:

$$D_E(x_i, x_j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2},$$

2. čtvercová Euklidovská vzdálenost:

$$D_{ES}(x_i, x_j) = \sum_{k=1}^n (x_{ik} - x_{jk})^2,$$

3. Manhattanská vzdálenost:

$$D_M(x_i, x_j) = \sqrt[q]{\sum_{k=1}^n \|x_{ik} - x_{jk}\|^q},$$

4. Čebyševova vzdálenost:

$$D_C(x_i, x_j) = \max_k(x_{ik} - x_{jk}),$$

5. Lancey-Williamsova vzdálenost:

$$D_{LW}(x_i, x_j) = \sum_{k=1}^n \frac{\|x_{ik} - x_{jk}\|}{\|x_{ik}\| + \|x_{jk}\|}.$$

[1]

1.3.1 Příklad výpočtu vzdáleností

Mějme tři vektory $x_1 = (0; 2)$, $x_2 = (5; 1)$ a $x_3 = (3; 1)$, pak jejich vzdálenosti jsou po výpočtu následující:

Tabulka 1 – Výpočet měr vzdáleností, [zdroj: vlastní]

míra	$x_1 - x_2$	$x_2 - x_3$
Euklidovská D_E	5,1	2
čtverec Euklidovská D_{ES}	26	4
Manhattanská D_M	6	2
Čebyševova D_C	5	2
Lancey-Williamsova D_{LW}	1,33	0,25

1.4 Hierarchické shlukování

Hierarchické shlukování se dále rozděluje na dva přístupy: aglomerativní a divizní. Při aglomerativním přístupu každý objekt tvoří jeden samostatný shluk a tyto shluky jsou následně při každém kroku slučovány do doby, než jsou všechny shluky sloučeny do jednoho. Méně užívaným přístupem je divizní hierarchické shlukování. Na začátku divizního přístupu tvoří všechny objekty jeden shluk a tento shluk je následně rozdělen v každém kroku na jednotlivé shluky, dokud všechny objekty netvoří samostatný shluk. [1], [2]

Výsledky hierarchického shlukování je možné přehledně prezentovat graficky, formou dendrogramu (stromového diagramu). Na vodorovné ose zobrazíme původní objekty (prvotní shluky) a na svislé ose zobrazíme stupnici hladiny propojení shluků. Dole začíná strom listy (jednotlivými objekty) a v každém kroku spojíme příslušné shluky na dané hladině. [2]

1.4.1 Algoritmus hierarchického shlukování

V tomto odstavci popíšeme algoritmus aglomerativního hierarchického shlukování:

1. Ze vstupní matice S vypočteme matici vzdáleností D , podle příslušné míry vzdáleností.
2. Začneme proces od n shluků, z nichž každý obsahuje právě jeden objekt.

3. Projdeme matici vzdáleností D a nalezneme i -tý a j -tý shluk, jejichž vzdálenost D_{ij} je minimální. Protože matice D je symetrická, procházíme pouze pravou část matice. Je-li v matici D více dvojic shluků s minimální hodnotou D_{ij} , vybereme náhodně jednu z těchto dvojic.
4. Nalezený i -tý a j -tý shluk s minimální hodnotou vzdálenosti spojíme do nového shluku označeného hodnotou k , i -tý a j -tý řádek a sloupec matice vzdáleností D odstraníme a nahradíme je jedním řádkem a sloupcem pro nový shluk označeného k , který definujeme jako minimální hodnotu z původních řádků k ostatním shlukům. Řád matice vzdáleností D se pak snížil o jedna.
5. Poznamenejme si hodnotu hladiny spojení a identifikátory i -tého a j -tého shluku, které jsme sloučili.
6. Opakujeme krok 3. a následující, dokud nejsou všechny shluky zařazeny do jediného shluku. [2]

1.4.2 Způsob hodnocení vzdáleností

Ve výše popsaném algoritmu jsme ve 4. kroku dva odstraněné shluky nahrazovali jedním, novým shlukem. Pro nový shluk jsme odvodili řádek a sloupec vzdáleností k ostatním shlukům v matici z nejmenších vzdáleností původního i -tého a j -tého shluku. Tento způsob hodnocení nazýváme metodou nejbližšího souseda. Stejně tak můžeme vzdálenosti odvozovat jako maximální hodnotu (metoda nejvzdálenějšího souseda) nebo průměrnou hodnotu (metoda průměrné vazby) z původních hodnot i -tého a j -tého shluku. [2]

Metoda nejbližšího souseda (nejbližší vazby)

Při přepočtu matice vzdáleností odvodíme pro všechny shluky a nový shluk vzdálenosti dle vzorce: $D_{kl} = \min(D_{il}, D_{jl})$. Při aplikaci této metody však vznikají tzv. mosty, kdy se i často vzdálené objekty spojí do stejného shluku. Tento efekt může být nevýhodný, požadujeme-li eliptický tvar shluků se zhuštěným jádrem. Mimo zápornou vlastnost tvorby mostů má metoda nejbližšího souseda spíše více příznivých vlastností. [2]

Metoda nejvzdálenějšího souseda (úplné vazby)

Metoda úplné vazby naopak využívá při odvozování nové vzdálenosti jako kritérium maximum z možných mezishlukových vzdáleností k danému shluku. Tedy, dle vzorce: $D_{kl} = \max(D_{il}, D_{jl})$. Tato metoda netvoří mosty, naopak je náchylnější k tvorbě kompaktních shluků. [2]

Metoda průměrné vazby

Tato metoda velmi často vede k podobným výsledkům shlukování jako metoda nejbližšího souseda. Jako kritérium pro odvozování mezishlukových vzdáleností se zde používá vzorec pro aritmetický průměr: $D_{kl} = \frac{D_{iu} + D_{jl}}{2}$. [1], [2]

1.4.3 Praktický příklad

Ukážeme si praktický příklad výpočtu aglomerativního hierarchického shlukového algoritmu s využitím metody průměrného souseda. Na počátku máme vstupní datovou matici S mající vektory: $x_1 = (0; 1; 1)$, $x_2 = (2; 1; 0)$, $x_3 = (2; 3; 1)$, $x_4 = (0; 0; 1)$, $x_5 = (2; 0; 2)$:

$$S = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 0 \\ 2 & 3 & 1 \\ 0 & 0 & 1 \\ 2 & 0 & 2 \end{bmatrix}$$

Ze vstupní matice S vypočítáme symetrickou matici vzdáleností s využitím euklidovské metriky:

$$D = \begin{bmatrix} 0 & 2,24 & 2,83 & 1 & 2,45 \\ & 0 & 2,24 & 2,45 & 2,24 \\ & & 0 & 3,6 & 3,16 \\ & & & 0 & 2,24 \\ & & & & 0 \end{bmatrix}$$

Nyní nalezneme v matici nejmenší hodnotu vzdálenosti, v tomto případě je minimem hodnota 1, která odpovídá vzdálenosti mezi shlukem x_1 a x_4 . Po tomto kroku zmenšíme dimenzi matice odstraněním řádků a sloupců reprezentující vzdálenosti k shlukům x_1 a x_4 . Dále odvodíme a vložíme do matice nové vzdálenosti nového shluku $x_6 = \{x_1, x_4\}$ k ostatním shlukům. Po tomto kroku dostaneme novou matici vzdáleností:

$$D = \begin{bmatrix} 0 & 2,24 & 2,24 & 2,34 \\ & 0 & 3,16 & 3,21 \\ & & 0 & 2,34 \\ & & & 0 \end{bmatrix}$$

Zkontrolujeme je-li řád matice roven jedné, pokud není pokračujeme v iteraci.

Vidíme, že minimální hodnotě 2,24 odpovídají vzdálenosti $D[x_2, x_3]$, $D[x_2, x_5]$. Vybere jednu shodnou minimální distanci, v tomto případě například $D[x_2, x_3]$, odstraníme nalezené shluky a přepočítáme matici D k novému shluku $x_7 = \{x_2, x_3\}$:

$$D = \begin{bmatrix} 0 & 2,34 & 2,69 \\ & 0 & 2,78 \\ & & 0 \end{bmatrix}$$

Zkontrolujeme zda je řád matice roven jedné, pokud není, pokračujeme v iteraci.

Minimální hodnotou je v této iteraci $D[x_5, x_6]$. Vybere shluky x_5 a x_6 a sloučíme je do nového shluku $x_8 = \{x_1, x_4, x_5\}$ a přepočítáme matici:

$$D = \begin{bmatrix} 0 & 2,74 \\ & 0 \end{bmatrix}.$$

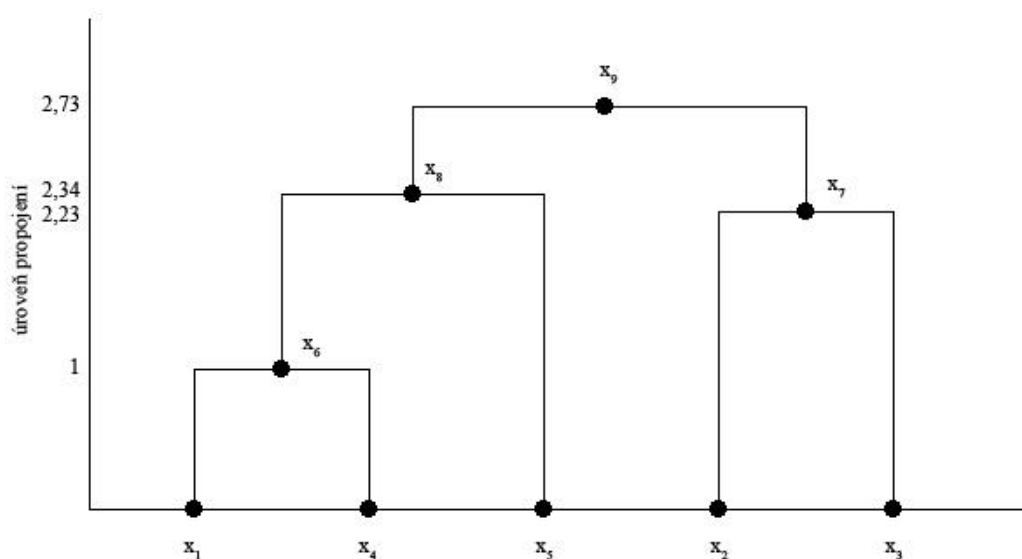
Zkontrolujeme je-li řád matice roven jedné, pokud není pokračujeme v iteraci.

V nové matici vzdáleností reprezentuje minimální vzdálenost $D[x_7, x_8] = 2,74$. Shluky x_7 a x_8 sloučíme do nového shluku $x_9 = \{x_1, x_2, x_3, x_4, x_5\}$. Po odstranění shluků x_7, x_8 a vložení nového shluku x_9 dostáváme novou matici D .

Po kontrole řádu matice zjišťujeme, že řád matice vzdáleností je roven jedné a tím algoritmus ukončujeme.

1.4.4 Prezentace výsledku hierarchického shlukování

Výsledky hierarchického shlukování se nejlépe prezentují graficky v tzv. dendrogramu neboli stromovém diagramu, v případě velkého počtu jedinců to ale neplatí. Na svislé ose je nejčastěji stupnice hladiny spojení shluků a na vodorovné ose jsou označeny jednotlivé objekty, které shlukujeme. [1]



Obrázek 2 – Zobrazení stromového diagramu se zvýrazněnými shluky, [zdroj: vlastní]

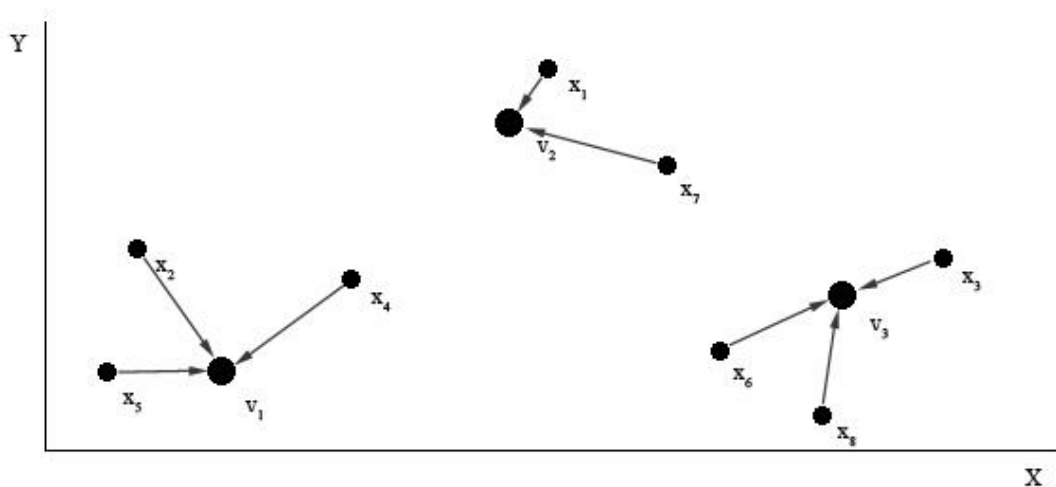
Jednotlivé shluky reprezentují objekty na vodorovné ose a také jednotlivá propojení shluků reprezentují další nový shluk vzniklý propojením dvou shluků na nižší úrovni propojení.

Prezentace výsledku shlukové analýzy prostřednictvím stromového diagramu však má i některé nevýhody. Při počtu nad cca 50 objektů začíná být diagram natolik "zhuštěný", že je jeho prezentace nepřehledná.

1.5 Nehierarchické (ploché, flat) shlukování

Nehierarchické shlukování, které se jinak nazývá ploché nebo také flat, se netýká výstavby stromového diagramu tak jako hierarchické shlukování. V nehierarchické metodě shlukování se objekty přidělují do předem známého počtu shluků. Počet konečných shluků je dán libovolně velkou množinou $V = \{v_1, v_2, v_3, \dots, v_n\}$ vzorových objektů neboli centroidů, ke kterým se přiřazují ostatní objekty podle jejich vzdáleností k těmto vzorovým shlukům. [3]

Vzorové objekty mohou být například vybrané objekty z množiny objektů na vstupu analýzy, ale stejně tak vzorovým objektem může být fiktivní objekt určitých vlastností, dle kterých chceme objekty ze vstupní matice rozdělit.



Obrázek 3 – Přiřazování objektů k vzorovým objektům ve dvourozměrném prostoru, [zdroj: vlastní]

1.5.1 Algoritmus nehierarchického shlukování

1. Stejně jako u hierarchického shlukování vypočteme matici vzdáleností D ze vstupní matice S . Na rozdíl od metody uvedené na straně 12 v podkapitole 1.4 ploché shlukování počítá matici vzdáleností k vzorovým objektům. To znamená, že pro každý objekt S_i vypočítáme dle vybrané metriky vzdálenosti se všemi vzorovacími objekty z množiny V . Výsledkem je asymetrická matice vzdáleností.
2. Procházíme matici vzdáleností a každý objekt S_i přiřadíme do shluku vzorovacího objektu, k němuž má zkoumaný objekt nejbližší, tedy jeho hodnota vzdálenosti k němu je nejmenší. Pokud má nejkratší vzdálenost k více než jednomu vzorovacímu objektu, přiřadíme jej náhodně k jednomu z těchto shluků. [3]

1.5.2 Praktický příklad

Na začátku algoritmu máme vstupní datovou matici S , která obsahuje vektory: $x_1 = (0 ; 1 ; 1)$, $x_2 = (2 ; 1 ; 0)$, $x_3 = (2 ; 3 ; 1)$, $x_4 = (0 ; 0 ; 1)$, $x_5 = (2 ; 0 ; 2)$, $x_6 = (2 ; 2 ; 1)$, $x_7 = (3 ; 3 ; 3)$,

$x_8 = (0; 0; 0)$ a množinu vzorových objektů $V = \{v_1, v_2, v_3\}$, kde $v_1 = (0; 0; 2)$, $v_2 = (3; 0; 2)$ a $v_3 = (3; 3; 2)$:

$$S = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 0 \\ 2 & 3 & 1 \\ 0 & 0 & 1 \\ 2 & 0 & 2 \\ 2 & 2 & 1 \\ 3 & 3 & 3 \\ 0 & 0 & 0 \end{bmatrix}.$$

Ze vstupní matice S a vzorových objektů z množiny V vypočítáme asymetrickou matici vzdáleností s využitím euklidovské metriky:

$$D = \begin{bmatrix} 1,41 & 3,32 & 3,74 \\ 3 & 2,45 & 3 \\ 3,74 & 3,32 & 1,41 \\ 1 & 3,16 & 4,36 \\ 2 & 1 & 3,16 \\ 3 & 2,45 & 1,73 \\ 4,36 & 3,16 & 1 \\ 2 & 3,6 & 4,69 \end{bmatrix}.$$

V dalším kroku nalezneme minimální hodnoty ke vzorovým objektům. Pro x_1 vidíme, že je nejbližší vzorovému objektu s hodnotou 1,41, proto ho přiřadíme do shluku $S_{v_1} = \{x_1\}$, pro x_2 nalézáme minimální vzdálenost s hodnotou 2,45 k vzorovacímu objektu v_2 a přiřadíme jej tedy do shluku $S_{v_2} = \{x_2\}$. Přiřazování do shluků provádíme do doby, než projdeme celou matici vzdáleností D .

Výsledkem jsou tři shluky $S_{v_1} = \{x_1, x_4, x_8\}$, $S_{v_2} = \{x_2, x_5\}$, $S_{v_3} = \{x_3, x_6, x_7\}$.

2 APLIKACE

Pro aplikaci shlukové analýzy poslanců do programové podoby byla zvolena forma webové aplikace pojmenovaná Hlasování poslanců. V podobě webové stránky nabízí program pro uživatele zejména snadný přístup a ovládání.

2.1 Teoretický základ

2.1.1 HTML

Zkratka HTML znamená Hypertext Markup Language, jedná se o hypertextový značkovací jazyk, který umožňuje tvorbu a zobrazení webových stránek na internetu. Důležitou vlastností tohoto jazyka je jeho univerzalita, protože HTML stránky lze v dnešní době zpracovat na všech počítačích a mobilních zařízeních (chytré telefony, tablety) prostřednictvím internetového prohlížeče. [6]

Webové stránky se prostřednictvím jazyka HTML tvoří pomocí tzv. tagů, které jsou umístěné ve špičatých závorkách <> a představují elementy webové stránky. Pro tvorbu webu v jazyce HTML stačí pouze textový editor, nicméně pro sofistikovanější tvorbu je vhodné využít jakýkoliv editor zvýrazňující syntaxi jazyka, například PSPad nebo NetBeans.

Základní strukturu HTML stránky tvoří záhlaví označené tagem <head> a tělo <body>. Obsah záhlaví není na webové stránce vidět, ale má za cíl určit vlastnosti stránky, například titulek stránky <title></title>. V samotném těle jsou zadány prvky, které jsou na stránce viditelné. Těmito prvky mohou být odstavce <p>...</p>, nadpisy <h1>...</h1>, odkazy <a>... a mnoho dalších prvků. Jednoduchá webová stránka v HTML pak může vypadat následovně: [6]

```
<html>
  <head>
    <title>Nová HTML stránka</title>
  </head>
  <body>
    <p>Ahoj světe!</p>
  </body>
</html>
```

Jazyk HTML obsahuje velké množství tagů různých elementů webové stránky a jejich popis by stačil na samostatnou publikaci, proto zde byl popsán jen velmi stručný základ jazyka HTML.

Aktuální verze HTML je verze 4.0.1, která se používá od roku 1999, je proto značně zastaralou verzí a pro tvorbu moderního webu nedostačující, z toho důvodu je chystána nová verze HTML 5, která splní veškeré požadavky moderních webových aplikací. [7]

2.1.2 CSS

Protože HTML samo o sobě pouze strukturu webové stránky vytvoří a již ji příliš vzhledově neupraví, slouží k tomuto účelu stylový jazyk Cascading Style Sheets ve zkratce CSS, neboli kaskádové styly. Stylový jazyk CSS obsahuje celou řadu selektorů, které je možné i kombinovat a které upravují vzhled HTML elementů. CSS je možné propojit s HTML třemi způsoby:

1. V elementu `<style>...</style>` ve kterém jsou definovány všechny CSS selektory a jejich vlastnosti a je vložen přímo v HTML dokumentu.
2. Přimo v HTML tagu s použitím atributu `style="..."`. Například pro jednoduchou změnu barvy textu pro nadpis: `<h1 style="color: red;" >...</h1>`.
3. V externím souboru kaskádových stylů s koncovkou `.css`. Tento způsob je nejpoužívanější pro svoji přehlednost a možností použít více kaskádových souborů pro jednu HTML stránku, či naopak jeden soubor `.css` pro více stránek. V záhlaví HTML kódu je na kaskádové styly odkazováno tagem: `<link href="style.css" rel="stylesheet" type="text/css">`.

Kaskádové styly mohou upravovat například vlastnosti typu:

- rozměry elementu: `width, height,`
- odsazení elementu: `margin, padding,`
- vlastnosti textu: `color, font-size, font-family,`
- vlastnosti bloku: `display,`
- okraje elementů: `border,`
- a další.

Nevýhodou kaskádových stylů je jejich rozdílné zobrazování v prohlížečích, způsobené chybným zobrazováním vlastností elementů. Například prohlížeč Internet Explorer kaskádový styl zobrazí jinak prohlížeč než Chrome, stejně tak je tomu i u ostatních prohlížečů. Proto je nutné některé elementy stylovat pro různé prohlížeče zvlášť.

Aktuální verze kaskádových stylů je CSS 2, ale stejně jako HTML 4.0.1 je tato verze zastaralá. Se zavedením nového HTML 5 bude CSS 2 nahrazeno kaskádovými styly CSS 3.

2.1.3 PHP

Zkratka PHP původně vznikla z názvu Personal Home Page, ale v současnosti se jedná o rekurzivní zkratku, která vyjadřuje Hypertext Processor, v překladu Hypertextový Procesor. PHP je nejrozšířenější skriptovací jazyk používaný k programování dynamického obsahu webových stránek a aplikací. Lze tedy pomocí PHP naprogramovat diskusní fórum, fotogalerii a další dynamické webové stránky. PHP skript je prováděn serverem a k uživateli je přenášen až výsledek skriptu, na rozdíl od HTML, které je prováděno na straně klienta. [8], [9]

Syntaxe jazyka PHP vychází z několika programovacích jazyků (Perl, C, Java). PHP skript je vždy ohraničen elementem `<?php ... ?>`. Na rozdíl od programovacích jazyků, ze kterých PHP vychází, jsou proměnné dynamicky typovány a při jejich deklaraci není nutné uvádět jejich datový typ. K identifikaci proměnné je vždy před jejím názvem uveden znak `$`. Aktuální verze jazyka PHP je verze 5.5. Nyní k ukázce jednoduchého PHP skriptu, který na obrazovku vypíše čísla 0-10: [8]

```
<?php
    for($i=0; $i<=10; $i++){
        echo $i;
    }
?>
```

2.1.4 Databáze

Databázi si lze představit jako soubor dat, které reprezentují objekty reálného světa, například evidence řidičů, správa parkoviště nebo agenda poslanců. Tyto objekty obsahují entity, představující konkrétní objekt reálného světa (člověk, student, zvíře, auto) s charakteristickými vlastnostmi (atributy), například jméno, rok narození, barva srsti, atd.

Jednotlivé entity mezi sebou vytvářejí vztahy (vazby) a jako celek tvoří relační databázový model. [10]

Pro práci s daty v databázi jsou používány databázové systémy jako MySQL či Oracle. Tyto databázové systémy jsou založeny na jazyce SQL (Structured Query Language), je to standardizovaný dotazovací jazyk, určený pro ovládání databází. SQL zahrnuje nástroje pro tvorbu databází a nástroje pro manipulaci s daty. SQL je tzv. deklarativní (neprocedurální) programovací jazyk, to znamená, že SQL kód není psán v samostatném programu, ale je vkládám do jiného programovacího jazyka, již procedurálního. V praxi SQL jazyk definuje pouze to, jaká data uživatel požaduje, to jakým způsobem tato data dostane, obstarává databázový systém. [10]

2.2 Funkční požadavky

Funkčními požadavky rozumíme seznam požadavků, které musí být do aplikace implementovány:

- načtení dat o hlasování poslanců a senátorů z webových stránek parlamentu a senátu,
- uložení načtených dat,
- volba metody shlukové analýzy,
- volba vlastností metod shlukové analýzy,
- zpracování dat o hlasování,
- zobrazení výsledku analýzy dat.

2.3 Softwarové požadavky

V případě webové aplikace klient/server klademe minimální požadavky zejména na webový server a také na klienta, který přistupuje na webovou aplikaci skrze webový prohlížeč.

Požadavky na server

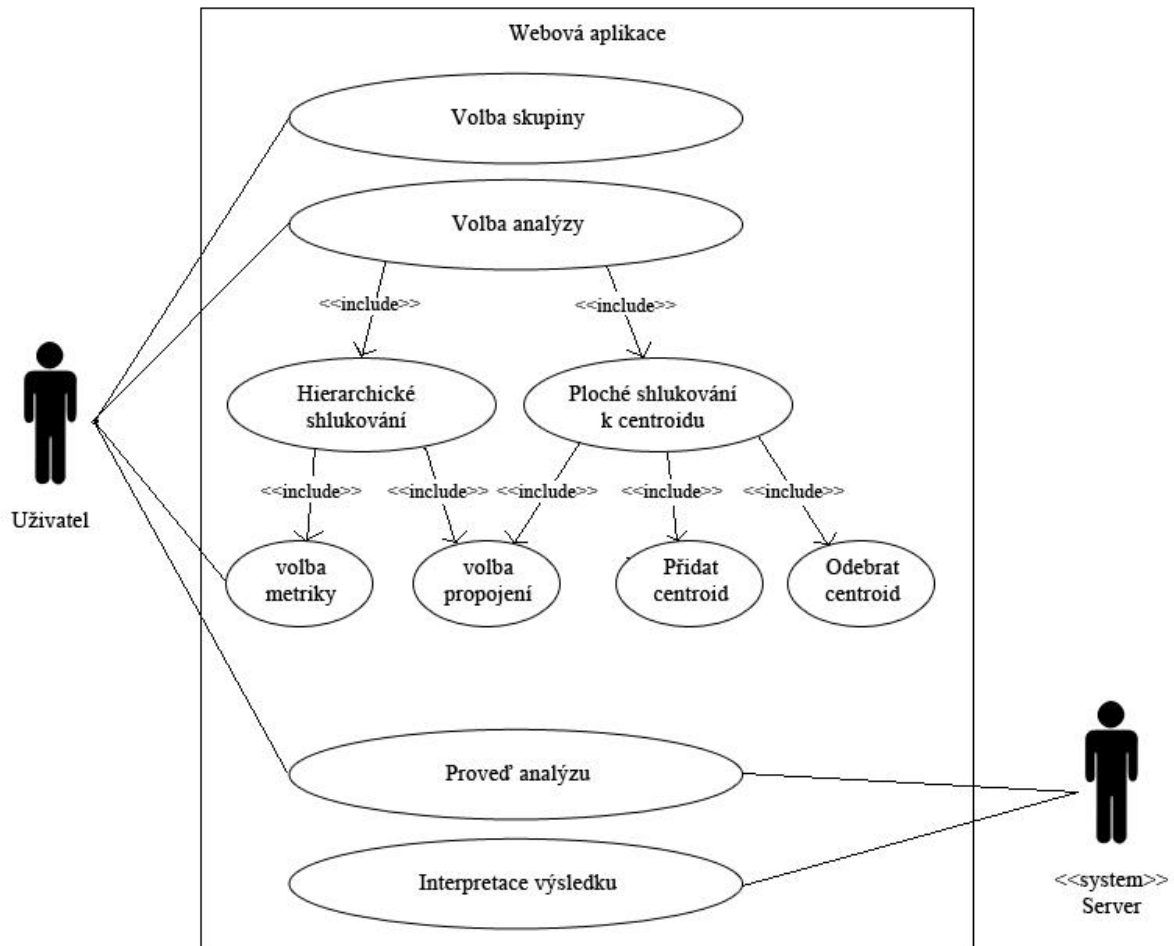
- Webový server: Apache 2.4.4 a novější,
- skriptovací jazyk: PHP 5.4.12 a novější,
- databázový systém: MySql 5.6.12 a novější.

Požadavky na klienta

- Webový prohlížeč s podporou JQuery: IE, Opera, Firefox, Chrome a další,
- stabilní připojení k internetu.

2.4 UML use case diagram

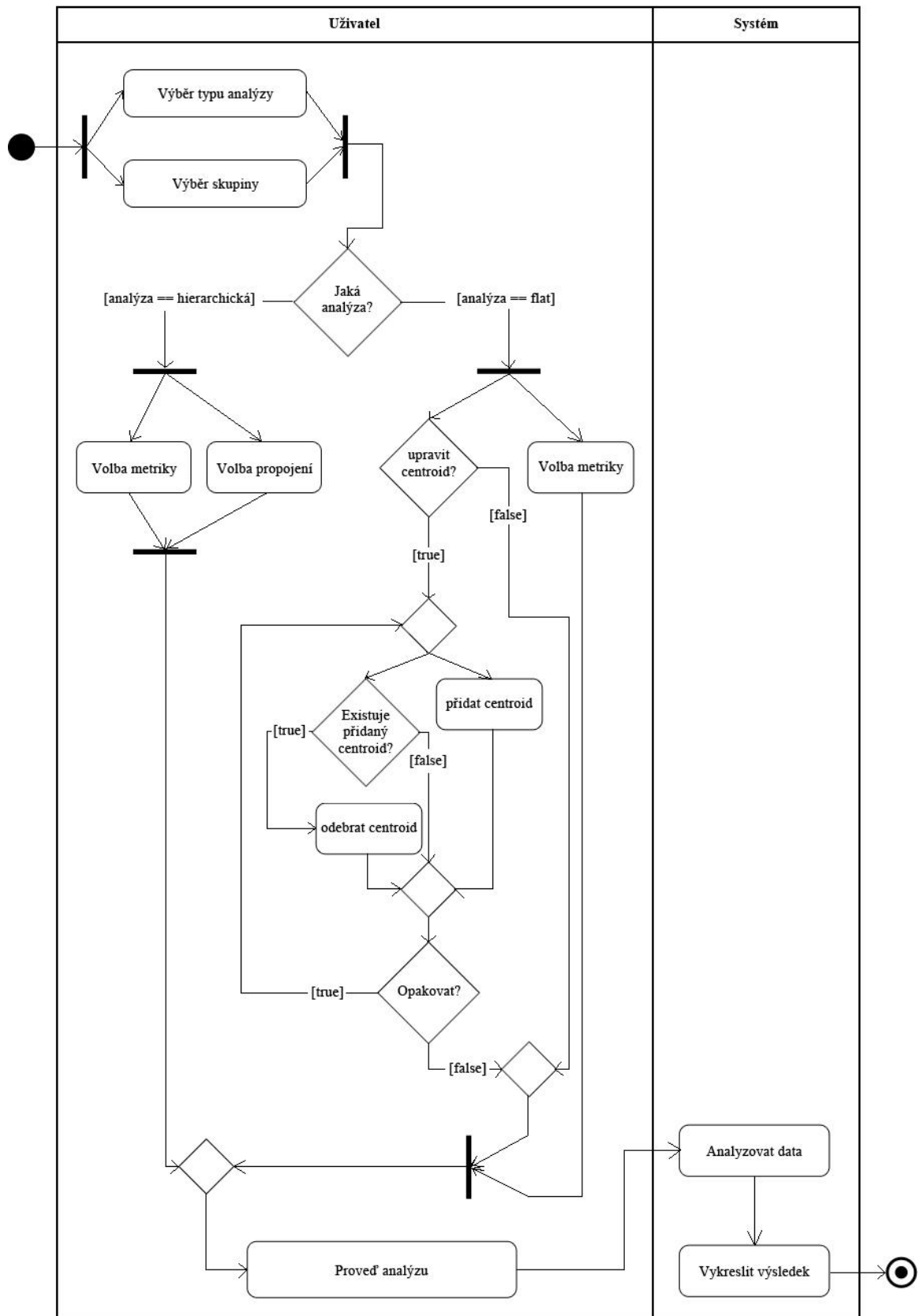
Diagram UML use case zobrazuje možnosti rolí: Uživatel, Server a úkony, které může vykonat ve webové aplikaci.



Obrázek 4 – UML case diagram, [zdroj: vlastní]

UML use case diagram zobrazuje návštěvníka, pojmenovaného Uživatel, který po vstupu na webovou stránku může vykonat volbu skupiny pro kterou provede analýzu, zároveň také může vykonat výběr druhu analýzy. Po dokončení výběru analýzy jsou Uživateli povoleny další možnosti vybrané analýzy. V případě hierarchického shlukování je Uživateli umožněno zvolit typ metriky a způsob propojení. Pro ploché shlukování smí nastavit způsob propojení a přidávat nebo odebírat vzorové objekty. V poslední části je Uživateli umožněno provést analýzu, kterou zpracuje role Server a vrátí výsledek analýzy.

2.5 UML activity diagram



Obrázek 5 – UML activity diagram, [zdroj: vlastní]

UML activity diagram vyjadřuje procedurální logiku webové aplikace, konkrétně akce, jež role (druh uživatele) mohou vykonávat a v jakém sledu tyto aktivity vykonávají. UML activity diagram, viz obrázek 5, obsahuje role dvě: Uživatel a Systém. [11]

Po vstupu do diagramu vybere Uživatel druh analýzy a současně zvolí i druh skupiny na kterou se vybraná analýza aplikuje. Po výběru druhu analýzy pokračuje k aktivitám nutným ve vybrané analýze:

Rozhodl-li se Uživatel pro hierarchickou analýzu, jsou dalšími možnými aktivitami: volba metriky a volba propojení. Po výběru Uživatel pokračuje dále.

Pokud byla zvolena flat analýza, může Uživatel zvolit aplikovanou metriku a pokud neupraví seznam centroidů pokračuje dále. V případě, že se Uživatel rozhodne pro vlastní seznam centroidů (upravit seznam centroidů), může přidat nový centroid nebo odebrat některý z centroidů ze seznamu, není-li seznam prázdný. Po dokončení úprav seznamu centroid pokračuje Uživatel dalším krokem.

Poslední aktivitou Uživatele je zvolit akci: Provést analýzu, která pokračuje do Systému, jenž provede aktivitu: Analyzovat data a po jejím dokončení vykoná aktivitu: Vykreslit výsledek a touto aktivitou procedurální logika skončí.

2.6 Návrh struktury databáze

Struktura databáze pro data o poslancích je z části převzatá a upravená struktura databázových souborů dostupných na webových stránkách www.psp.cz, protože převedení dat to jiné struktury by bylo vzhledem k množství entit a jejich atributů náročnou operací. Pro data senátorů byla tato databáze následně rozšířena o entity nutné k začlenění dat o senátorech v původní databázi.

Databázový model obsahuje celkem deset entit, z nichž každá splňuje 3. normální formu, databázi je proto možné v budoucnu rozšířit dle nových požadavků a potřeb.

2.6.1 Entity

V této podkapitole se seznámíme s entitami použitými v databázové struktuře:

Tabulka 2 – Entity a jejich atributy, [zdroj: vlastní]

Název entity	Atributy
Funkce	id, id_funkce, id_organ, id_typ_funkce, nazev_funkce_cz
Hlasovani_poslanec	id, id_poslanec, id_hlasovani, vysledek
Hlasovani_senator	id, id_osoba, id_hlasovani, vysledek
Hlasovani_url	id_hlasovani, url, used
Organy	id, id_organ, organ_id_organ, id_typ_organu, zkratka, nazev_organu_cz, nazev_organu_en, od_organ, do_organ, priorita, cl_organ_base
Osoby	id, id_osoba, jmeno, prijmeni, titul1, titul2, narozeni, pohlavi
Poslanec	id, id_poslanec, id_osoba, id_kraj, id_kandidatka, id_obdobi
Senatorske_kluby	id_klubu, nazev, zkratka
Zarazeni	id, id_osoba, id_of, cl_funkce, od_o, do_o
Zarazeni_senator	id, id_klub, id_osoba

Funkce

Tato entita představuje funkci, jež politik vykonává, je-li mu funkce přiřazena.

Hlasovani_poslanec

Entita Hlasovani_poslanec obsahuje informace o výsledku hlasování jednotlivého poslance, kterého představuje atribut id_poslanec. Hlasování poslanců reprezentuje atribut id_hlasovani.

Hlasovani_senator

Tato entita obsahuje informace o výsledku hlasování jednotlivého senátora, kterého v této entitě představuje entita Osoby s id_osoba. Hlasování senátorů dále navazuje atributem id_hlasovani na entitu Hlasovani_url.

Hlasovani_url

Entita představuje jednotlivá hlasování senátorů na určité webové stránce o které uchovává atribut url. Atribut used určuje, zda již bylo hlasování z dané url načteno (pokud bylo načteno: url = 1, pokud nebylo hlasování načteno: url = 0).

Organy

Podobná entita jako entita Funkce a představuje jednotlivé politické orgány.

Osoby

Jak název napovídá entita Osoby představuje reálné osoby.

Poslanec

Entita Poslanec navazuje na entitu Osoby, která představuje poslance v určitém volebním období a kraji.

Senatorske_kluby

Jedná se o entitu, která představuje jednotlivé senátorské kluby. Zkratka senátorského klubu je tvořena počátečními písmeny názvu klubu.

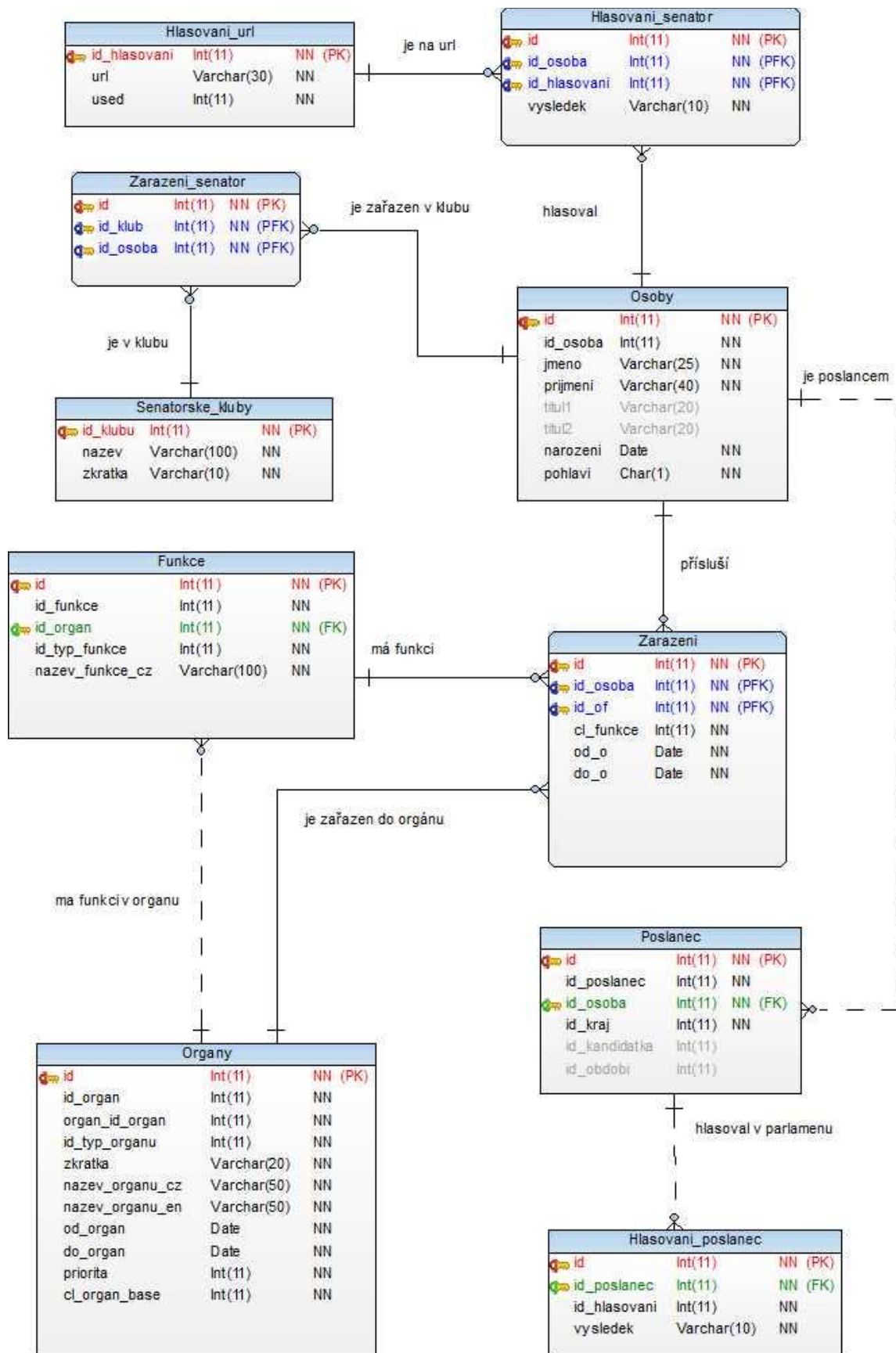
Zarazeni

Tato entita přiřazuje entitu Osoby k entitě Orgány. Pokud atribut `cl_funkce == 0`, `Zarazeni.id_o` odpovídá `Organy.id_organ`. Entitu Osoby přiřazuje k entitě Funkce, pokud `cl_funkce == 1`, potom atribut `Zarazeni.id_o` odpovídá `Funkce.id_funkce`. [12]

Zarazeni_senator

Entita `Zarazeni_senator` přiřazuje entitu Osoby, představující senátora, k entitě `Senatorske_kluby`.

2.6.2 ER diagram



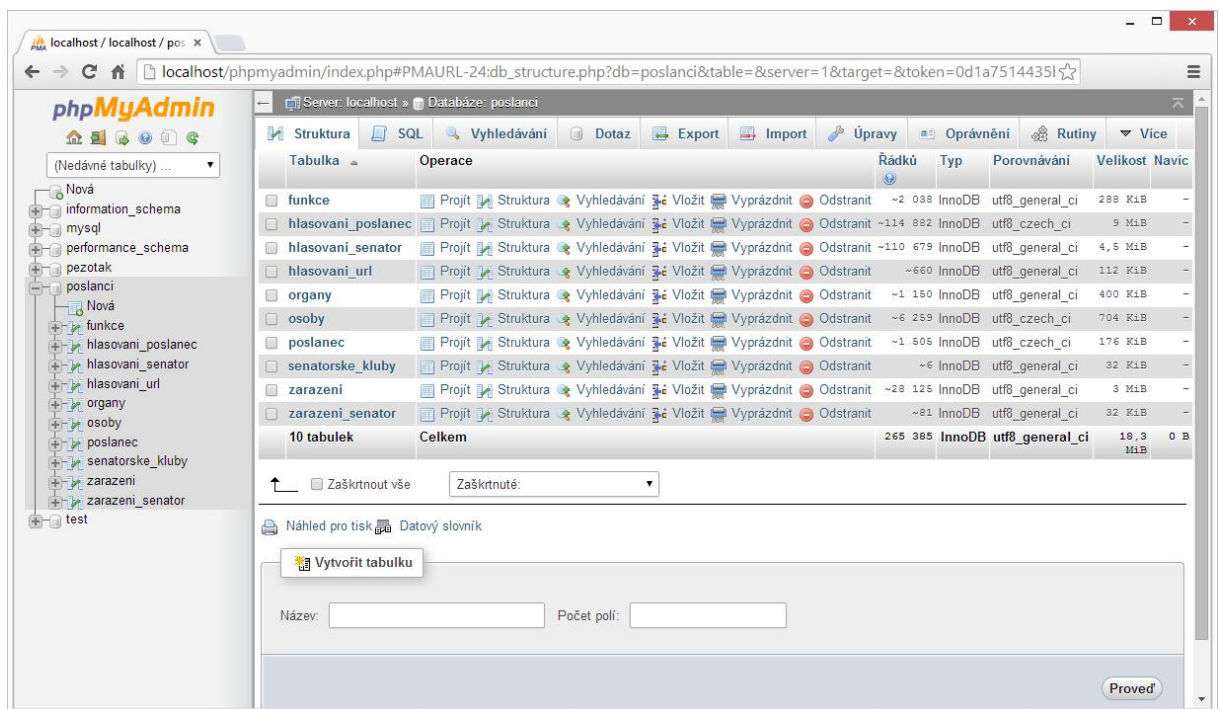
Obrázek 6 – ER diagram, [zdroj: vlastní]

2.6.3 Implementace databázového modelu

Pro implementaci databázového modelu byl použit databázový systém MySQL verze 5.6.12, který splňuje všechny požadavky kladené pro použití databázového modelu viz obrázek 6. Databázový systém MySQL byl pro vývoj provozován na lokálním serveru Apache 2.

Správu databázového systému MySQL při vývoji umožňoval správce PhpMyAdmin verze 4.1.7, který umožňuje správu databáze prostřednictvím webového rozhraní a poskytuje jednoduché nástroje pro správu databáze:

- uživatelské role s přístupovými právy a jejich správa,
- správa databází,
- správa tabulek, jejich sloupců a řádků,
- export, import databází a tabulek,
- vyhledávání v databázi,
- spouštění SQL příkazů,
- hromadné úpravy (vyprazdňování, odstraňování, optimalizace). [13]



Obrázek 7 – PhpMyAdmin, [zdroj: vlastní]

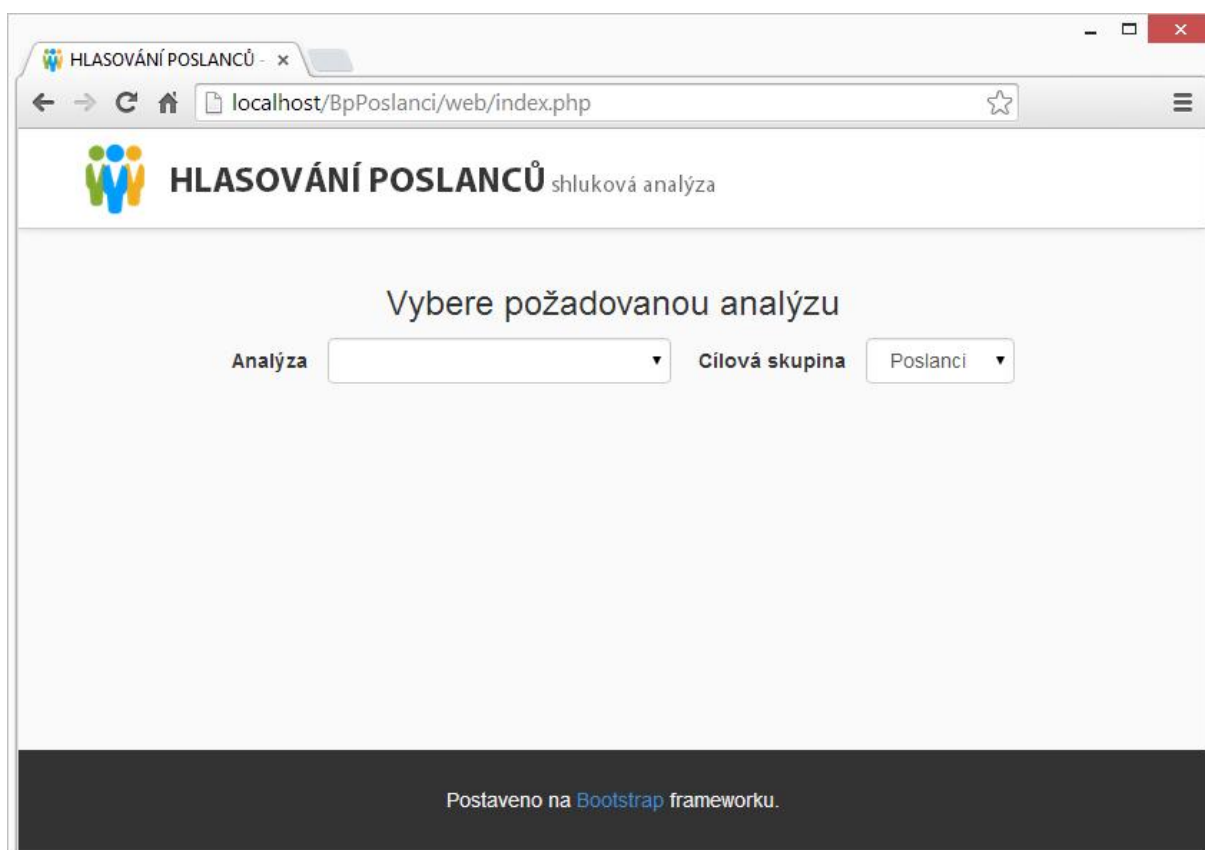
2.7 Vývoj aplikace

Pro vývoj aplikace byl z mnoha důvodů zvolen lokální server Apache 2, namísto využití některé placené nebo neplacené webhostingové služby. Výhodou lokálního serveru při vývoji

webové aplikace je plná kontrola činnosti serveru. Při programování skriptů se může vlivem chyby v programu vytvořit nekonečná smyčka a server je nutné restartovat, v případě, že nedojde k jeho "pádu". Vytvářet aplikaci na webhostingovém serveru by bylo proto nepraktické. Dalšími výhodami vývoje aplikace na lokálním serveru je rychlá aktualizace dat, vývoj lze provádět bez připojení k internetu, možnost upravovat nastavení serveru a jeho funkcí dle potřeb vyvíjené aplikace. Na vývojovém lokálním serveru byl implementován databázovým systémem MySQL 5 a scriptovací jazyk PHP 5.

2.7.1 Uživatelské rozhraní a grafické prvky

Uživatelské rozhraní (webová stránka) bylo vytvořeno s využitím Twitter Bootstrap frameworku. Jedná se o volně dostupný a jednoduchý soubor nástrojů pro vytváření webových stránek a webových aplikací. Twitter Bootstrap podporuje širokou škálu moderních webových technologií: HTML, CSS a JavaScript. Navíc obsahuje další prvky, které usnadní tvorbu webových stránek. Pro práci s tímto front-end frameworkem je nutná alespoň minimální znalost HTML a CSS. [4]



Obrázek 8 – Webová stránka s využitím Twitter Bootstrap frameworku, [zdroj: vlastní]

Ve zkratce je možné Twitter Bootstrap framework definovat jako předem připravené prvky, z nichž lze skládat požadovanou webovou aplikaci. Předpřipravené prvky je možné upravovat

podle potřeb, nutná je jen znalost použitých technologií. Pro tvorbu specifických webů však nemusí být framework Twitter Bootstrap ideálním řešením. Pro uživatelské rozhraní aplikace Hlasování poslanců se ukázalo toto řešení vhodnou volbou.

Instalace frameworku je velmi jednoduchá, zdrojové soubory Twitter Bootstrap stažené z domovského webu: <http://getbootstrap.com> se umístí na server a poté stačí na tyto soubory odkázat v záhlaví webové stránky elementem `<link>`.

Všechny soubory související s webovou stránkou Hlasování poslanců se nacházejí v adresáři: web, jedná se o zdrojové soubory Twitter Bootstrap, vlastní grafické prvky, CSS soubory a JavaScript knihovny.

Grafické prvky aplikace

Pro tvorbu grafických prvků, které nejsou součástí frameworku, byl využit volně dostupný grafický software Gimp. Grafickými prvky je myšleno logo v horní části webu a obrázky postav použité pro prezentaci výsledku analýzy.

2.7.2 Vytváření aplikace a funkcionality

V této podkapitole bude popsána tvorba samotné aplikace a ukázka některých vytvořených funkcí s popisem úkonů, které tyto funkce vykonávají.

Práce s MySql

Po implementaci databáze bylo nutné vytvořit vrstvu, která implementuje práci s databázovým systémem MySql. Pro tuto úlohu vznikla statická třída `TDb`, jež uchovává spojení s databází a obsahuje funkce pro vytváření, kontrolu a zpracování MySql příkazů. Lze tak jednoduše vytvořit SQL příkaz typu `select`, `insert` nebo `update`.

Práce s tabulkami

Pro snadnější práci s tabulkami jsou v adresáři `scripts/TDObject` vytvořeny statické třídy reprezentující jednotlivé tabulky, které dědí rodičovskou třídu `TDBObjectBaseClass` implementující základní funkce:

- `TDHlasovaniPoslanec.php`,
- `TDHlasovaniSenator.php`,
- `TDHlasovaniUrl.php`,

- TDOrgany.php,
- TDOsoby.php,
- TDPoslanec.php,
- TDSenator.php,
- TDSenatorskeKluby.php,
- TDZarazeni.php,
- TDZarazeniSenator.php.

Tyto statické třídy o tabulkách uchovávají informace a obsahují funkce, které přistupují do databáze a vykonávají nad tabulkami SQL příkazy. Například funkce `insert()`, která vkládá data do tabulky:

```
public static function insert(Array $dataArray){
    if(count($dataArray) > 0){
        try{
            self::getTData()->checkRelevant($dataArray, true);
        }catch (Exception $e){
            echo $e->getMessage()." ".$e->getFile()." ".$e->getLine();
        }
    }
    return self::getTData()->insert($dataArray);
}
```

Do parametru funkce je vkládáno pole s hodnotami pro vložení do tabulky. Toto pole může obsahovat jeden řádek hodnot, potom indexy pole představují názvy sloupců tabulky nebo můžeme vkládat více řádků najednou potom do parametru vkládáme dvourozměrné pole. V prvním kroku skript zkontroluje, zda všechny názvy indexů v poli souhlasí s atributy tabulky, souhlasí-li sestaví SQL příkaz typu `insert` a provede jej. Funkce vrací Boolean hodnotu úspěchu provedení SQL příkazu nad databází.

Optimalizace práce s daty

Při vývoji aplikace, bylo nutné provést optimalizace procesů zpracovávajících data, hlavním důvodem bylo velké množství dat, s nimiž se pracuje. Výchozí nastavení serveru pro Php je paměťový limit o velikosti 256 MB, ten byl často překračován a skript ukončen chybovou

zprávou. Řešením problému s paměťovým limitem pro PHP, aniž by muselo být upraveno výchozí nastavení serveru, bylo zpracování dat po částech tak, aby byla paměť využívána efektivněji.

Dalším důvodem pro optimalizaci byla délka časového intervalu zpracování SQL příkazů insert v databázi. Při prvním spuštění, musí aplikace do databáze vložit velké množství dat a sbyl-li každý řádek (záznam) ukládán samostatným SQL příkazem, byl časový interval od spuštění SQL dotazů až po jejich dokončení nepřijatelný (pro 100 000 záznamů cca 45 minut). Řešením pomalého zpracování dat bylo vkládání více záznamů (řádků) v jednom SQL dotazu. Podle provedených pokusných měření, je hodnota 3000 záznamů (řádků) v jednom SQL dotazu vyhodnocena jako nejefektivnější pro takto velký objem dat.

Získávání dat z webových stránek

Jedním z úkolů této práce je získání dat o hlasování poslanců a senátorů z webových stránek poslanecké sněmovny respektive senátu. Na webových stránkách poslanecké sněmovny (<http://www.psp.cz>) v menu: Veřejnost a média, sekci: Data agend PS, jsou volně dostupné datové soubory (.unl) s agendou hlasování poslanců. Zpracování těchto souborů po stažení z webové stránky není složitou operací. Pro práci se soubory byla vytvořena třída `File` implementující všechny nutné funkce.

Získat data z webových stránek lze také parsováním webu. Parsováním se rozumí uložení HTML kódu webové stránky, ze kterého jsou vybrána potřebná data. Parsování lze provést různými metodami, v této práci bude popsán pouze způsob parsování použitý v aplikaci Hlasování poslanců pro získání dat o hlasování senátorů z webových stránek senátu ČR:

1. Nejdříve je nutné stáhnout zdrojový kód stránky, k tomu je použita knihovna `cURL`, která umožňuje komunikaci s mnoha různými servery skrze velké množství protokolů: `http`, `https`, `ftp` a další. V aplikaci implementuje parsování třída `Parser`. Při vyvolání konstruktoru třídy `Parser` se zavolá funkce `getWebsiteContent()`, která pracuje s výše zmiňovanou knihovnou `cURL`: [5]

```
function getWebsiteContent($sUrl, $charset = 'utf-8') {
    $ch = curl_init($sUrl);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
```

```

$output = curl_exec($ch);
curl_close($ch);
$output = iconv($charset, 'UTF-8', $output);
return $output;
}

```

Funkce `getWebsiteContent()` získá HTML kód webové stránky ve formě datového typu `String`, ten je následně funkcí `iconv()` převeden z původního kódování do kódování UTF-8 a takto upravený `String` vrací na výstupu.

2. V dalším kroku je volána funkce `getXpathFromContent()`, které je předán upravený obsah webu parametrem:

```

function getXpathFromContent($content) {
    $dom = new DOMDocument();
    @$dom->loadHTML(mb_convert_encoding($content, 'HTML-ENTITIES',
    'utf-8'));
    $xpath = new DomXPath($dom);
    return $xpath;
}

```

HTML kód je vložen do instance třídy `DOMDocument`, která reprezentuje HTML dokument a tento objekt je v dalším kroku předán nové instanci třídy `DomXPath`.

Pro nalezení požadovaných dat v získaném HTML kódu je nutné použít výraz jazyka XPath, který v dokumentu odpovídá cestě k určitým elementům. V aplikaci Hlasování poslanců je sestaven XPath pro parsování jednotlivých hlasování sestaven následně:

```
//article/div[3]/div/table/tr/td
```

Z výrazu je patrné, že odkazuje na prvky ve struktuře dokumentu. Tento konkrétní XPath výraz vyhledá všechny elementy `<td>`, uvnitř každého elementu `<article>` obsahující v pořadí třetí element `<div>`. Pro názornou ukázkou, výraz nalezne všechny elementy obsahující text "shoda":

```

<article>
    <div>...</div>
    <div>...</div>

```

```

<div>
  <div>
    <table>
      <tr><td>shoda</td><td>shoda</td></tr>
      <tr><td>shoda</td></tr>
      ...
    </table>
  </div>
</div>
</article>

```

3. Po sestavení výrazu jej zpracuje funkce implementována v třídě `DomXPath->query()`. Je-li nalezena shoda v dokumentu, funkce vrátí instanci třídy `DomNodeList`, která obsahuje všechny nalezené prvky v dokumentu.

Implementace shlukové analýzy

Další součástí práce je naprogramování shlukové analýzy a jejím prostřednictvím zpracovat získaná data. Všechny funkcionality shlukové analýzy se nachází ve `scripts/Analyse`. Pro shlukovou analýzu byla vytvořena rodičovská třída `ClusterAnalyse`. Rodičovskou třídu dědí třídy konkrétních metod shlukové analýzy: `HierarchicalClustering` a `FlatClustering`, ty implementují funkci `analyseData()`, která obsahuje konkrétní algoritmus pro danou metodu analýzy popsaných v kapitolách 1.4 a 1.5. Pro vykreslení výsledku analýzy implementují navíc tyto třídy funkci `draw()`.

Shluková analýza pracuje s daty v matici a bylo nasnadě vytvořit třídy, které práci s takto uspořádanými daty ulehčí. Pro matici a operace s ní proto vznikla třída `Matice`, jednou z často používaných operací nad maticí ve shlukové analýze je hledání minimální hodnoty, proto byla implementována i funkce `Matice->getMinElement()`:

```

public function getMinElement(){
    if(count($this->vectorArr) == 0){
        throw new Exception("Prazdna matice");
    }
    $min = pow(2,28);
    $minX = null;
    $minY = null;

```

```

foreach($this->vectorArr as $x => $row){
    foreach($row as $y => $value){
        if(($this->isSymetric && $x < $y) || !$this->isSymetric){
            if($min > $value){
                $min = $value;
                $minX = $x;
                $minY = $y;
            }
        }
    }
}
return array("x" => $minX, "y" => $minY, "min" => $min);
}

```

Funkce v prvním kroku zkontroluje naplněnost matice, následně nastaví počáteční minimální hodnotu na číslo 2^{28} a souřadnice umístění minimální hodnoty na null. V cyklech prochází řádky a sloupce matice a současně porovnává současné minimum s hodnotami matice na aktuálním řádku a sloupci. Je-li matice symetrická podle nulové diagonály, prochází algoritmus pouze pravou polovinu matice. Na výstupu vrací funkce minimální hodnotu se souřadnicemi výskytu.

Třidu `Matice` dědí třída `DistMat`, která představuje matici vzdáleností a mimo implementaci funkcí rodičovské třídy obsahuje vlastní metody pro výpočet matice vzdáleností ze vstupních dat pro analýzu. Funkce pro výpočet symetrické matice vzdáleností vypadá následovně:

```

public function calcDistMat_MxM(Matice $vectorMat, $metrics){
    if(is_null($vectorMat)){
        throw new Exception("Prazdna matice vektoru");
    }
    foreach($vectorMat->getVectorArr() as $rowKey => $vector1){
        foreach($vectorMat->getVectorArr() as $colKey => $vector2){
            parent::setElement($rowKey, $colKey,
                Metrics::getDistance($vector1, $vector2, $metrics));
        }
    }
}

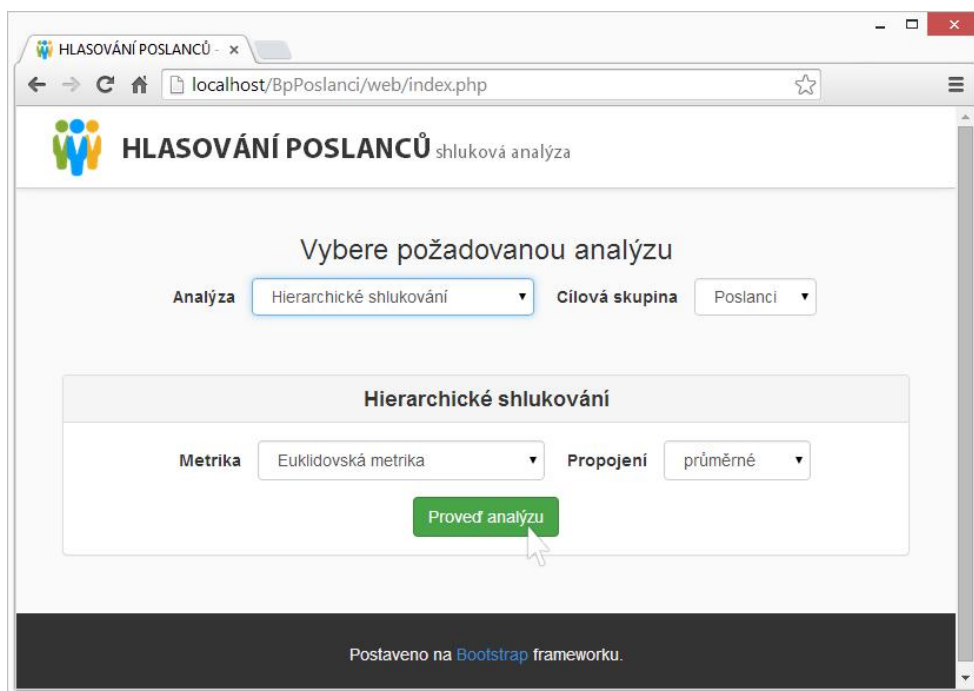
```

Parametrem této funkce je instance třídy `Matice` obsahující vstupní matici dat pro analýzu a konstanta třídy `Metrics`, určující aplikovanou metodu výpočtu vzdáleností mezi objekty ze vstupní datové matice. Pro každý řádek matice v cyklu spočítá funkce vzdálenosti ke všem ostatním řádkům ze vstupní matice a uloží tyto vypočtené vzdálenosti do matice vzdáleností.

Propojení uživatelského rozhraní a logické části

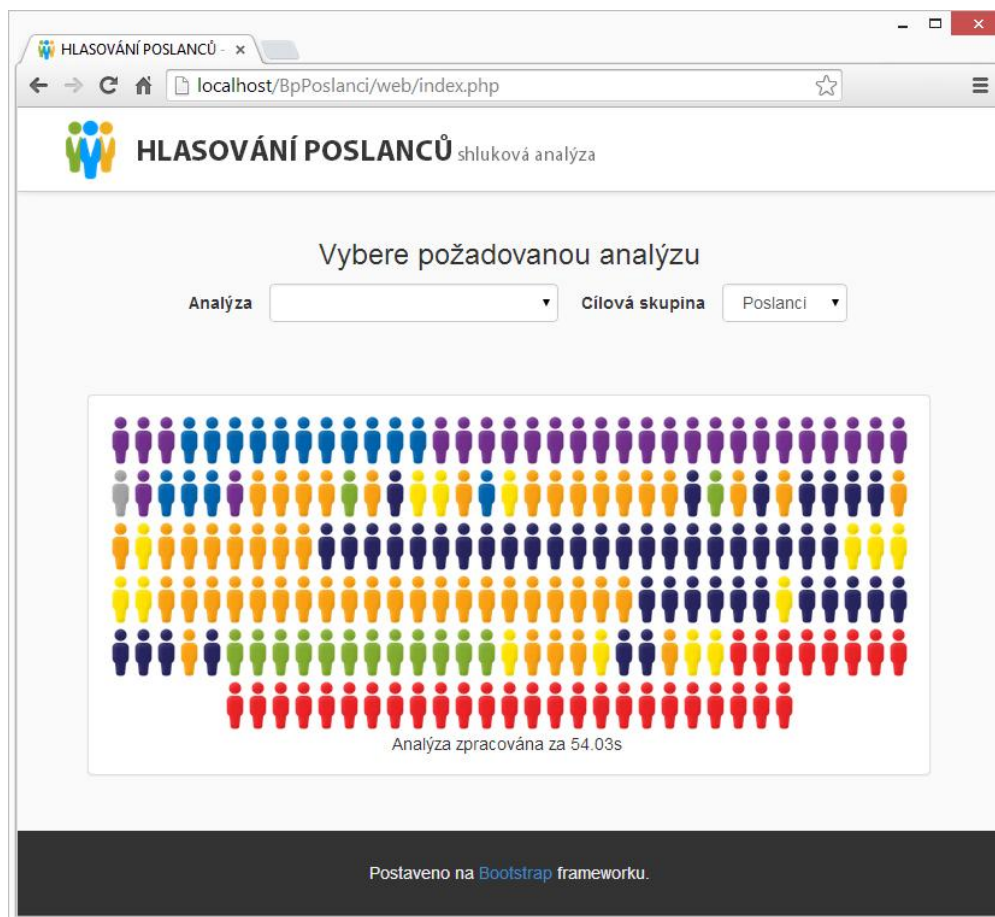
Uživatelské rozhraní a logickou část aplikace rozděluje tzv. Controller, v tomto případě je Controller třída implementující funkce, které může uživatelské rozhraní vyvolat. Celé propojení obstarává aplikační interface neboli API (application programming interface), v této aplikaci se API nachází v souboru `ajax_api.php`, kterému jsou předávány akce z uživatelského rozhraní (API) prostřednictvím AJAXu.

Využití AJAXu má ve webových aplikacích a webových stránkách své opodstatnění, před jeho existencí, bylo nutné pro každou změnu jejich obsahu načíst znovu celý obsah stránky. Taková akce zbytečně načítá již zobrazený obsah a z pohledu uživatele je i nepříjemná. Proto vznikl AJAX, jehož prostřednictvím se mění obsah webu, bez nutnosti opětovného načtení celé stránky. Jedinou nevýhodou AJAXu je nutnost novější verze prohlížeče s podporou této technologie. Například uživatel vybere z webového rozhraní požadovanou analýzu, metody a poté stiskne tlačítko: Proved' analýzu, viz obrázek 9.



Obrázek 9 – Uživatelské rozhraní s odesláním požadavku na akci, [zdroj: vlastní]

AJAX tuto akci zachytí a odešle metodou GET požadavek do API na příslušnou akci. V tomto případě je akcí hierarchická shluková analýza, aplikovaná na skupinu poslanci, s Euklidovskou metrikou a průměrným propojením, viz obrázek 9. Aplikační interface (API), zkontroluje relevantnost požadované akce a pokud není irelevantní vyžádá si od Controlleru danou akci. Controller akci provede (zpracuje analýzu) a vrátí její výsledek. V konečném kroku API předá výsledek AJAXu, který provede změnu v obsahu webu bez nutnosti načtení stránky. V toto případě API vrátí AJAXu výsledek analýzy formou diagramu, viz obrázek 10.



Obrázek 10 – AJAXem vložený diagram, [zdroj: vlastní]

3 INSTALACE

Kapitola Instalace charakterizuje postupy potřebné ke zprovoznění aplikace Hlasování poslanců na webovém serveru, zprovoznění databáze a konfigurace základních nastavení aplikace.

3.1.1 Databáze

Ke zprovoznění databáze je nutné použít webové uživatelské rozhraní pro správu databáze PhpMyAdmin. Po vstupu do uživatelského rozhraní přejdeme do sekce Import a vybereme soubor s exportovanou databází poslanci.sql v příloze A a provedeme import do databáze. Soubor poslanci.sql obsahuje mimo struktury databáze také data o hlasování, proto chcete-li začít s prázdnou databází vybereme v levém sloupci okna rozhraní databázi Poslanci, v hlavním sloupci zaškrtneme všechny tabulky a v nabídce zvolíme volbu Vyprázdnit.

3.1.2 Aplikace

Instalace samotné aplikace Hlasování poslanců obnáší pouze nahrání zdrojových souborů aplikace na server na kterém bude aplikace provozována, soubory nalezneme v příloze A.

3.1.3 Hlavní konfigurační soubory

V adresáři config nalezneme hlavní konfigurační soubory obsahující základní nastavení nutné k funkci aplikace Hlasování poslanců. Adresář obsahuje dva soubory, z nichž nás bude zajímat pouze soubor Db_config.php, který obsahuje proměnné nutné k propojení aplikace a databáze:

- username – uživatelské jméno pro přihlášení do databáze,
- pass – heslo pro přihlášení do databáze,
- host – server, kde je databáze provozována.

Po nastavení proměnných v konfiguračním souboru pro databázi je instalace dokončena a můžeme přejít ke spuštění aplikace prostřednictvím prohlížeče.

4 UŽIVATELSKÁ PŘÍRUČKA

Pro stručné popsání úkonů v uživatelském rozhraní je příručka popsána v jednotlivých krocích. Po vstupu na webovou stránku můžete jako návštěvník vykonat v zásadě pouze dvě akce, hierarchickou analýzu a nehierarchickou analýzu:

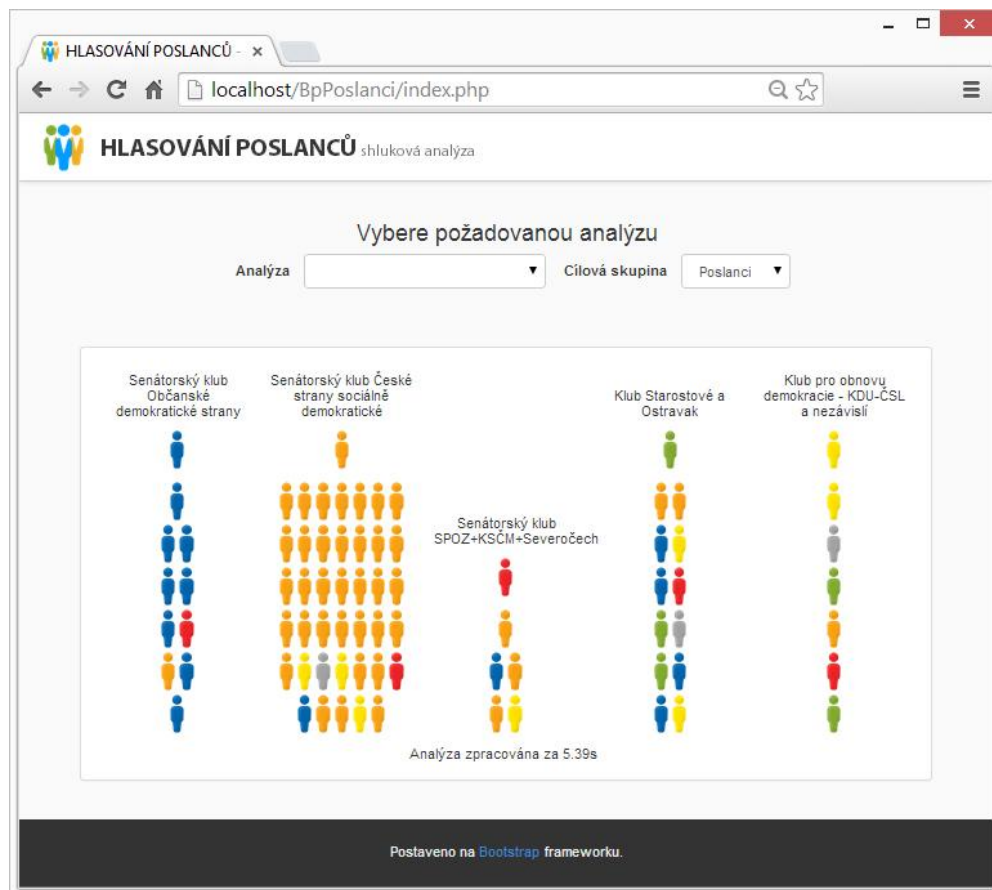
Hierarchická analýza

1. Z nabídky Analýza vyberte volbu Hierarchické shlukování.
2. Jako výchozí skupina pro analýzu je v nabídce Cílová skupina vybraná skupina Poslanci, pokud chcete analyzovat hlasování senátorů zvolte nabídku Senátoři.
3. Hierarchická analýza nabízí dvě další nastavení metod, jakými bude analýza provedena. Jako výchozí metrika v nabídce Metrika je volba Euklidovská metrika, pokud chcete aplikovat jinou metodu, proveďte změnu v nabídce Metrika.
4. Druhým možným nastavením je nabídka Propojení, která je nastavena na výchozí hodnotě průměrné. Pokud chcete použít jiné propojení, změňte volbu v nabídce Propojení.
5. Nyní můžete nastavenou analýzu provést stiskem tlačítka Proved' analýzu.
6. Je vám zobrazena animace značící zpracování analýzy. V této chvíli vám není dovoleno konat další volbu analýzy. Po zpracování je animace zpracování analýzy skryta a nahrazena diagramem, který prezentuje názorovou vzdálenost jednotlivých poslanců.
7. Nyní můžete vykonat další analýzu nebo ukončit okno prohlížeče.

Nehierarchická analýza

1. Z nabídky Analýza vyberte volbu Nehierarchické shlukování.
2. Jako výchozí skupina pro analýzu je v nabídce Cílová skupina vybraná skupina Poslanci, pokud chcete analyzovat hlasování senátorů, zvolte nabídku Senátoři.
3. Nehierarchická analýza nabízí dvě další nastavení. Jako výchozí metrika v nabídce Metrika je volba Euklidovská metrika, pokud chcete aplikovat jinou metodu proveďte změnu v nabídce Metrika.
4. Druhým možným nastavením je nastavení centroidů. Pokud nevyberete vlastní seznam centroidů, jsou centroidy předsedové stran respektive senátorských klubů.
5. Pokud přidáte do seznamu centroidů osobu ze seznamu, můžete ji odebrat červeným tlačítkem zobrazeného u jména.
6. Nyní můžete nastavenou analýzu provést stiskem tlačítka Proved' analýzu.

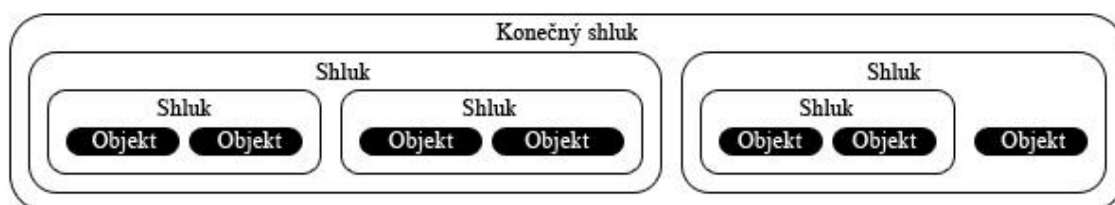
7. Je vám zobrazena animace značící zpracování analýzy. V této chvíli vám není dovoleno konat další volbu analýzy. Po zpracování je animace zpracování analýzy skryta a nahrazena diagramem, prezentující loajalitu osob k předsedům či vybraným osobám, není-li k některému z vybraných vzorů přiřazena žádná osoba, není tato osoba zobrazena.
8. Nyní můžete vykonat další analýzu nebo ukončit okno prohlížeče.



Obrázek 11 – Diagram zpracované flat analýzy, [zdroj: vlastní]

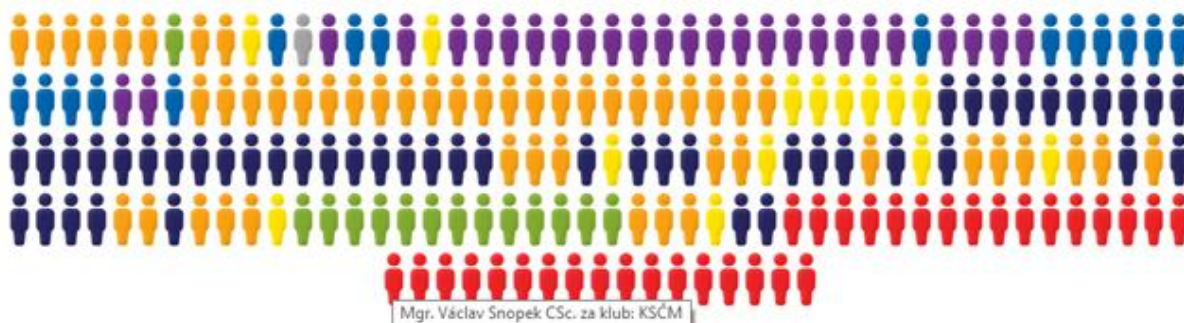
5 VÝSLEDKY APLIKACE

V předchozích kapitolách byla popsána kompletní aplikace s výjimkou interpretace výsledků analýzy, které bude věnována tato kapitola. V teoretické části, podkapitole 1.4.4, byla popsána interpretace hierarchické shlukové analýzy formou stromového diagramu, také bylo uvedeno, že stromový diagram (dendrogram) není vhodný pro větší množství objektů, to je případ i aplikace Hlasování poslanců. Zobrazení výsledku analýzy dendrogramem pro větší počet objektů by bylo nepřehledné, proto bylo nutné nalézt jinou formu interpretace výsledku hierarchické analýzy. Nejjednodušším řešením bylo vykreslit na řádek jednotlivé objekty, tento řádek představoval konečný shluk s prvky seřazenými tak, aby vedle sebe byly objekty z levého shluku a vedle nich objekty z pravého shluku, takto jsou objekty řazeny dle shluků až k základním shlukům, které obsahují jediný objekt, pro ilustraci popsané interpretace viz obrázek 12.



Obrázek 12 – Způsob seskupení objektů pro interpretaci výsledku, [zdroj: vlastní]

Tento způsob zobrazení se ukázal pro aplikaci Hlasování poslanců jako vhodný, protože velmi dobře a jednoduše zobrazuje rozložení jednotlivých politických stran, respektive senátorských klubů, přiřadí-li se objektům barva politické strany. Na druhou stranu pro přesnější interpretaci jednotlivých shluků není použita interpretace vhodná, neboť nevykresluje osoby v jednotlivých shlucích viz obrázek 12.



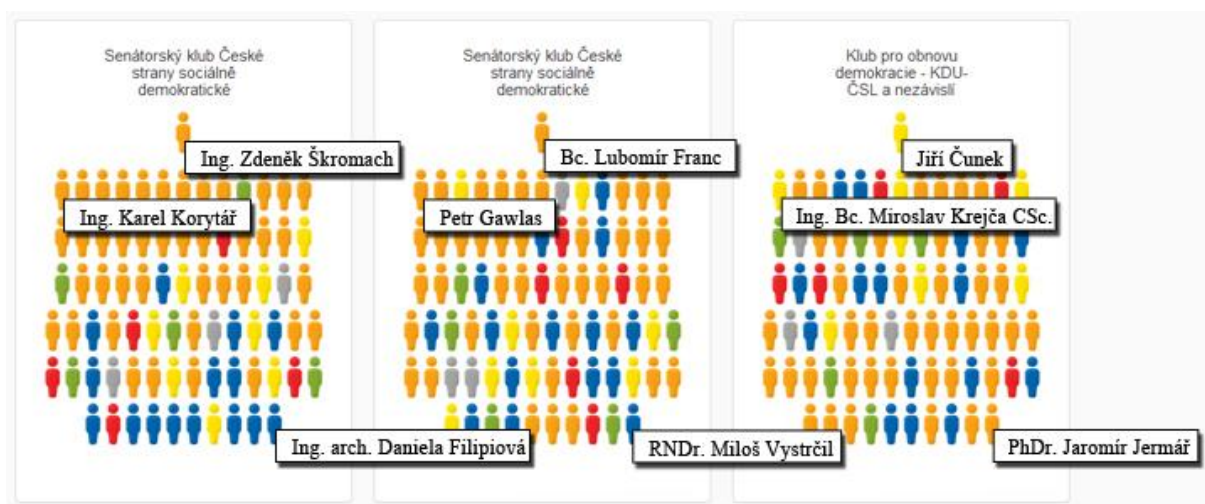
Obrázek 13 – Interpretace hierarchické analýzy s ikonami postav odlišené barvou, [zdroj: vlastní]

Na obrázku 13 je názorně vidět vykreslení výsledku hierarchické analýzy dat o hlasování poslanců za použití Euklidovské metriky a průměrné vazby. Aplikace je omezena rozlišením

monitoru na straně klienta, proto není možné zobrazit všechny ikony osob na jeden řádek, proto je zalomen stejně jako běžný text. Po najetí kurzoru myši nad ikonu postavy se zobrazí jméno osoby a její zařazení.

Z obrázku 13 je dále možné usuzovat, že například červeně označená strana KSČM a její poslanci jsou názorově velmi podobní, zatímco oranžově označená politická strana ČSSD a tmavě modrá strana ANO jsou do určité míry smíšený, lze tedy usuzovat, že strana ČSSD a ANO mohou tvořit koalici.

Nehierarchické shlukování je vzhledem k interpretaci výsledku jednodušší než hierarchické shlukování, v horní části jsou zobrazeny osoby zvolené jako vzorové objekty s popisem politické příslušnosti, pod nimi v řádcích seřazené osoby od nejloajálnějších k nejméně loajálním a ikony osob jsou opět orientované ve směru textu.

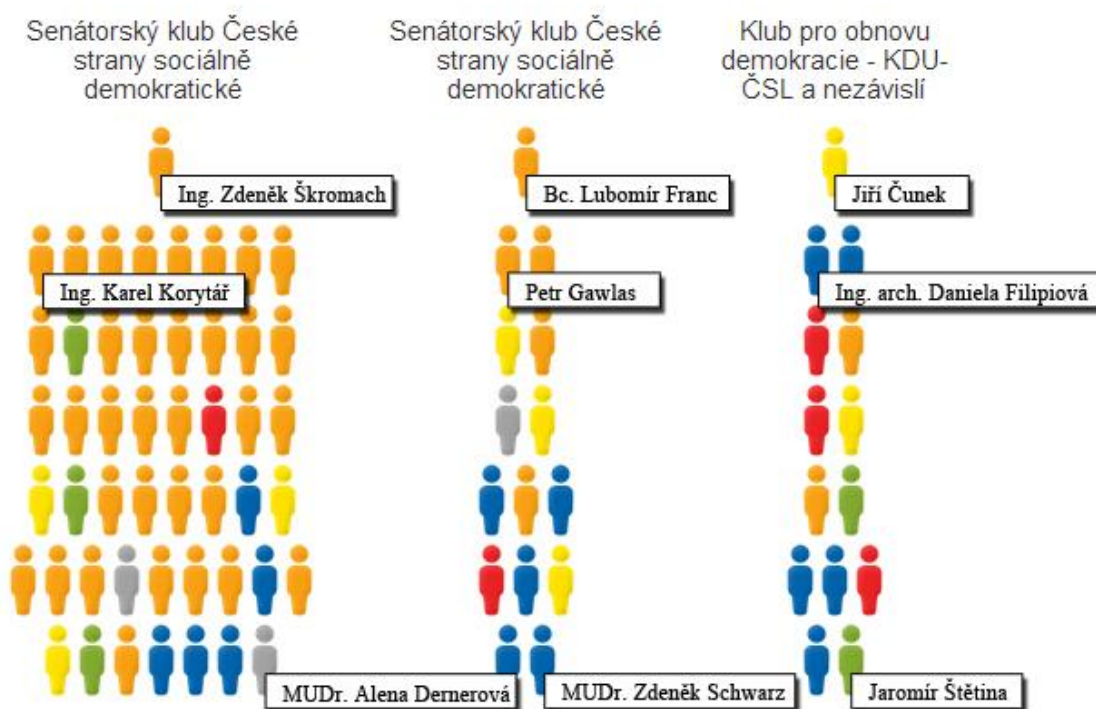


Obrázek 14 – Interpretace tří plochých analýz pro jeden vzorový objekt, [zdroj: vlastní]

Na obrázku 14 lze vidět interpretace výsledků tří nehierarchických (plochých) analýz vždy pro jeden vzorový objekt. Za vzorové objekty byly pro ukázkou zvoleny známé osobnosti české politické sféry: Ing. Zdeněk Škromach, Bc. Lubomír Franc a Jiří Čunek. Pokud zvolíme pro plochou analýzu pouze jeden vzorový objekt, diagram zobrazuje loajalitu všech senátorů k zvolenému vzoru (na obrázku jsou zobrazena jména vzorových objektů, jména osob nejvíce loajálních a nejméně loajálních vzhledem ke zvolenému vzoru).

Rozdílný výsledek analýzy lze získat zvolíme-li vzorové objekty v jedné ploché analýze (s Euklidovskou metrikou a průměrnou vazbou), viz obrázek 15. K vzorovému objektu jsou

přiřazeny pouze osoby, jejichž loajalita je větší k tomuto vzoru než k vzorům ostatním, objekty jsou seřazeny opět od nejloajálnějších k méně loajálním.



Obrázek 15 – Interpretace jedné nehierarchické analýzy pro tři vzorové objekty, [zdroj: vlastní]

Je vhodné také uvést vzdálenosti jednotlivých osob ke vzorovým objektům, protože tabulka vzdáleností obsahuje velké množství hodnot jsou vypsány pouze vzdálenosti k nejloajálnějšímu a nejméně loajálním osobám, viz tabulka 3. Nejmenší hodnota ve sloupci (min) určuje vzdálenost nejloajálnějšího jedince a větší hodnota (max) určuje vzdálenost nejméně loajálního jedince přiřazeného k vzoru:

Tabulka 3 – Vzdálenosti nejvíce a nejméně loajálních osob k vzorovým objektům, [zdroj: vlastní]

	Jiří Čunek	Bc. Lubomír Franc	Ing. Zdeněk Škromach
Ing. Karel Korytář			11,56 (min)
MUDr. Alena Dernerová			15,87 (max)
Petr Gawlas		13,04 (min)	
MUDr. Zdeněk Schwarz		16,07 (max)	
Ing. arch. Daniela Filipiová	14,9 (min)		
Jaromír Štětina	16,55 (max)		

Z tabulky 3 a obrázku 15 si lze vytvořit představu o tom, jak podobní jsou jednotliví senátoři vzorovému objektu respektive ke kterému vzoru má blíže více senátorů. Zatímco k Ing. Zdeňku Škromachovi je nejvíce podobný Ing. Karel Korytář s vzdáleností 11,56, kterou můžeme v rámci tabulky 3 považovat za nejkratší vzdálenost, naopak k Jiřímu Čunkovi je nejloajálnější senátorka Ing. arch. Daniela Filipiová "až" s hodnotou vzdálenosti 14,9. Rozdílné vzdálenosti nejloajálnějších senátorů ke svým vzorům a počet přiřazených senátorů k vzorům by mohly napovídat také o "názorových sympatiích" v senátu k těmto osobám.

6 ZÁVĚR

Cílem této bakalářské práce bylo vytvoření programu, který získá data o hlasování poslanců a senátorů z webových stránek poslanecké sněmovny a senátu České republiky, získaná data program zpracuje shlukovou analýzou a interpretuje výsledek analýzy.

První kapitola je věnována teoretického základu shlukové analýzy, který charakterizuje počáteční data pro shlukovou analýzu, metody měření vzdáleností mezi zkoumanými objekty, druhy shlukové analýzy a jejich algoritmy. Kromě teoretického základu obsahuje práce podkapitoly s praktickými příklady výpočtu měř vzdáleností a ukázky použití algoritmů shlukové analýzy.

Další částí je tvorba webové aplikace, která implementuje metody shlukové analýzy. Pro pochopení použitých technologií aplikace je první část této kapitoly věnována seznámení s těmito technologiemi, zejména se jedná o značkovací jazyk HTML, stylový jazyk CSS, databázový systém MySQL a skriptovací jazyk PHP. Dále popisuje funkční a softwarové požadavky kladené na aplikaci a návrh databázového modelu, tvorbu ER diagramu a popis jednotlivých entit. Po seznámení s teoretickým základem stavby struktury aplikace, je popsána samotná tvorba aplikace a řešení některých problémů při tvorbě vzniklých.

V posledních kapitolách jsou popsány úkony nutné ke zprovoznění aplikace a možnosti ovládání webového rozhraní.

Vytvořená aplikace není v konečném stádiu vývoje, pro "ostrý" provoz je nutné přizpůsobit webové uživatelské rozhraní majoritním webovým prohlížečům, protože vývoj probíhal s testováním aplikace v prohlížeči Chrome a správná funkce AJAXu ani kaskádových stylů v jiných prohlížečích není zaručena. Aplikaci je možné rozšířit o další funkce a metody shlukové analýzy a způsoby jejich interpretace, kterými by mohly být například možnosti exportu a importu dat shlukové analýzy do respektive z datových souborů.

Ačkoli je aplikace stále ve stádiu vývoje, lze ji shledat použitelnou a v rámci bakalářské práce splňuje všechny požadavky vypsáné v jejím zadání. V testovacím provozu je aplikace Hlasování poslanců dostupná na webové adrese: <http://www.hlasovaniposlancu.eu/>. Pro uplatnění aplikace by bylo vhodné ji začlenit k webové aplikaci zpracovávající další

statistiky politické sféry České republiky, ale je velmi zajímavé vidět na diagramech vytvořené aplikace provázání mezi jednotlivými politiky i mimo rámec politické strany.

7 POUŽITÁ LITERATURA

[1] ŘEZANKOVÁ, Csc., Doc. Ing. Hana, Ing. Dušan HÚSEK, Csc. a Prof. RNDr. Václav SNÁŠEL, Csc. *Shluková analýza dat*. Příbram: Professional Publishing, 2007. ISBN 978-80-86946-26-9.

[2] HEBÁK, Petr, Jiří HUSTOPECKÝ, Iva PECKOVÁ, Miroslav PLAŠIL, Milan PRŮŠA, Hana ŘEZANKOVÁ, Petr VLACH a Alžběta SVOBODOVÁ. *Vícerozměrné statistické metody [3]*. 2. doplněné vydání. Praha: Informatorium, spol. s.r.o., 2007. ISBN 978-80-7333-001-9.

[3] MELOUN, Milan a Jiří MILITKÝ. *Kompendium statistického zpracování dat: Metody a řešené úlohy*. Vyd. 2., přeprac. a rozš. Praha: Academia, 2006. ISBN 80-200-1396-2.

[4] Twitter Bootstrap. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001, 2013-10-15. [cit. 2014-04-29]. Dostupné z: http://cs.wikipedia.org/wiki/Twitter_Bootstrap

[5] PHP.net. *PHP: Introduction - Manual* [online]. © 2001-2014 [cit. 2014-04-29]. Dostupné z: <http://www.php.net/manual/en/intro.curl.php>

[6] DELLWIG, Ingo. *HTML 4: příručka tvůrce webu*. Praha: Grada Publishing, spol. s.r.o., 2002. ISBN 80-247-0297-5.

[7] HyperText Markup Language. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001, 2014-02-18 [cit. 2014-04-29]. Dostupné z: http://cs.wikipedia.org/wiki/HyperText_Markup_Language

[8] PHP. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001, 2014-04-22 [cit. 2014-04-29]. Dostupné z: <http://cs.wikipedia.org/wiki/PHP>

[9] PHP: základy. *Tvorba-webu.cz* [online]. © 2003 - 2008 [cit. 2014-04-29]. Dostupné z: <http://www.tvorba-webu.cz/php/>

[10] SKŘIVAN, Jaromír. Interval.cz: Databáze a jazyk SQL. *Interval.cz*/ [online]. 2000 [cit. 2014-04-29]. ISSN 1212-8651. Dostupné z: <http://interval.cz/clanky/databaze-a-jazyk-sql/>

[11] Diagram aktivit. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001, 2014-03-12 [cit. 2014-04-29]. Dostupné z: http://cs.wikipedia.org/wiki/Diagram_aktivit

[12] Poslanci a osoby. *Psp.cz* [online]. [cit. 2014-04-29]. Dostupné z: <http://www.psp.cz/sqw/hp.sqw?k=1301>

[13] PhpMyAdmin. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001, 2014-1-21 [cit. 2014-04-29]. Dostupné z: <http://cs.wikipedia.org/wiki/PhpMyAdmin>

8 PŘÍLOHY

Příloha A - CD-ROM s aplikací Hlasování poslanců