



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Webový informa ní systém menší politické strany
Student:	Tomáš Doležálek
Vedoucí:	Ing. David Buchtela, Ph.D.
Studijní program:	Informatika
Studijní obor:	Informa ní systémy a management
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Cílem práce je návrh a implementace webového eviden ního systému menší politické strany umož ůující evidenci len strany, volebních komisí a výsledk voleb. Sou ástí návrhu je i volba vhodné platformy v etn stanovení rolí a zabezpe eného p ístupu uživatel .

1. Prove te analýzu funk ních a nefunk ních uživatelských požadavk .
2. Prove te návrh vlastního ešení systému dle doporu ených metod a technik softwarového inženýrství.
3. Návrh implementujte ve form pilotního systému v prost edí LAMPS (Linux, Apache, MySQL, PHP a SSL) s ohledem na požadavek zabezpe eného p ístupu uživatel .
4. Pilotní systém ádn otestujte a nasa te, po dohod s vedoucím práce, u vybrané politické strany.
5. Prove te finan ní zhodnocení náklad na zavedení a provoz aplikace a benefit pro koncového uživatele.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 16. listopadu 2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Webový informační systém menší politické strany

Vedoucí práce: Ing. David Buchtela PhD.

17. května 2016

Poděkování

Chtěl bych poděkovat své rodině za zázemí a podporu, které mi poskytla. Dále bych chtěl poděkovat vedoucímu práce za cenné rady, které mi při vypracování poskytl.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 17. května 2016

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2016 Tomáš Doležálek. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Doležálek, Tomáš. *Webový informační systém menší politické strany*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

V této bakalářské práci se čtenář dočte o analýze, návrhu a implementaci informačního systému. Součástí analýzy je popis funkčních a nefunkčních požadavků, případů užití a identifikace uživatelských rolí. V návrhu jsem se zabýval architekturou webové aplikace a uživatelským rozhraním. Tento informační systém byl vytvořen pro menší politickou stranu, aby nahradil nedostačující evidenci vedenou v programu Microsoft Excel. Informační systém je určen k evidenci členů strany, volebních výsledků a volebních komisí. Součástí informačního systému je MySQL databáze pro ukládání dat a webová aplikace napsaná v PHP. V bakalářské práci je také zahrnuto ekonomicko-manažerské zhodnocení nákladů na pořízení a provoz informačního systému, a popis přínosů pro politickou stranu. Přínosem informačního systému je zejména zjednodušení agendy politické strany a zabezpečení přístupu k evidovaným datům.

Klíčová slova webová evidence, informační systém, webová aplikace, politická strana, zabezpečený přístup, LAMPS, ekonomické zhodnocení, databáze, analýza, návrh

Abstract

This bachelor's thesis looks into the analysis, design and implementation of information system. Analysis contains description of functional and non-functional requirements, use case scenarios and user roles identification. In the design part I described the chosen architecture and created a basic design of user interface. This information system was created to replace ineffective Microsoft Excel tables for recording the data. Information system is therefore used to record data about political party members, election committees and results of elections. The information system consists of MySQL database for data storing and PHP web application. This bachelor's thesis also contains economical evaluation of the purchase and running expenses of the information system. I also described the benefits of the new information system to the political party. The main benefits so far is the simplification of data recording and safe access to data.

Keywords web application, information system, web register, political party, secured connection, LAMPS, economic evaluation, database, analysis, design

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza požadavků	5
2.1 Analýza	5
2.2 Současné řešení	5
2.3 Požadavky	5
2.4 Případy užití	9
2.5 Uživatelské role	13
3 Návrh	15
3.1 Doménový model	15
3.2 Architektura	16
3.3 Uživatelské rozhraní	17
4 Použité nástroje	25
4.1 HTML	25
4.2 CSS	25
4.3 Bootstrap	26
4.4 JQuery	26
4.5 DataTables	26
4.6 MySQL	26
4.7 PHP	27
4.8 PHPmailer	27
5 Realizace	29
5.1 Ukládání dat	29
5.2 Struktura webové aplikace	34
5.3 Popis webové aplikace	38

5.4 Testování	57
6 Ekonomicko manažerské zhodnocení	59
6.1 Zhodnocení nákladů	59
6.2 Přínosy informačního systému	60
Závěr	61
Literatura	63
A Seznam použitých zkratk	65
B Obsah příloženého CD	67

Seznam obrázků

3.1	Doménový model	15
3.2	Přihlašovací obrazovka	17
3.3	Uživatelský profil	18
3.4	Obrazovka moje příspěvky	19
3.5	Obrazovka členové	20
3.6	Obrazovka členové - detail člena	21
3.7	Obrazovka Volební komise	22
3.8	Obrazovka Volebních výsledků	23
5.1	Databázový model	30
5.2	Adresářová struktura	35
5.3	Přihlašovací obrazovka	38
5.4	Registrační obrazovka	39
5.5	Obrazovka obnovení hesla - zadání e-mailu pro obnovu	40
5.6	Obrazovka obnovení hesla - zadání nového hesla	40
5.7	Obrazovka obnovení aktivačního e-mailu - zadání e-mailu	41
5.8	Zobrazení uživatelského profilu	42
5.9	Zobrazení uživatelského profilu - formulář změny adresy	43
5.10	Obrazovka Moje příspěvky	44
5.11	Členská základna - základní zobrazení	45
5.12	Členská základna - přidání člena	46
5.13	Členská základna - zobrazení členů	47
5.14	Členská základna - přidání příspěvku členovi	48
5.15	Členská základna - zobrazení informací o členovi	49
5.16	Volební komise - základní zobrazení	50
5.17	Volební komise - přidání komise	51
5.18	Volební komise - zobrazení komisí a členů	52
5.19	Volební komise - přidání nového člena	53
5.20	Volební výsledky - základní zobrazení	53
5.21	Volební výsledky - přidání volebního výsledku	54

5.22	Volební výsledky - zobrazení volebních výsledků	55
5.23	Navigační menu - zobrazení pro běžného uživatele	56
5.24	Navigační menu - zobrazení pro pokročilého uživatele	56

Úvod

Politické strany jsou dnes důležitou součástí veřejného života. Politické strany a jejich členové formulují společné cíle a ideologie. Skrze politické strany se občané mohou podílet na důležitých rozhodnutích týkajících se například města, ve kterém žijí nebo se mohou podílet na politice státu. Politické strany mohou mít spousty členů, podléhají mnoha pravidlům a přísné evidenci. Vzhledem k těmto okolnostem je evidence činnosti velmi důležitá pro všechny politické strany.

Je to právě rozsáhlá oblast působení a množství členů, jež činí informační systémy zajímavé pro využití i malou politickou stranou. Pro politickou stranu je důležité sdružovat své členy, vyslechnout jejich názory, sestavovat kandidátní listiny a volební komise. Bez centrální správy informací o členech se tato činnost stává velmi obtížnou, zejména se zvyšujícím se počtem členů. Centrální správa je výhodná pro každou organizaci, protože umožňuje rychlejší a efektivnější vyhledávání a aktualizace důležitých informací. Informační systém navíc poskytne prostor pro budoucí růst strany, který by jinak mohl být obtížný.

Ve své práci se budu zabývat návrhem a vývojem informačního systému pro menší politickou stranu a to formou webové evidence. Výhodou webové evidence je přístup do systému pomocí webového prohlížeče, což klade nízké nároky na technické vybavení uživatelů. Součástí bakalářské práce je návrh a tvorba této webové evidence. Tento proces se skládá z analýzy požadavků, samotného návrhu a implementace. Provedu také otestování a manažersko ekonomické zhodnocení přínosů pro politickou stranu. Dle požadavků zadavatele proběhne tvorba webové evidence v prostředí LAMPS, což znamená využití Linuxu, Apache, MySQL databáze, PHP a zabezpečeného přístupu pomocí SSL.

Cíl práce

Cílem bakalářské práce je analýza, návrh a implementace webové evidence pro menší politickou stranu. Tato webová aplikace umožní evidenci členů politické strany, uchovávání volebních výsledků a volebních komisí. Bude se tedy v podstatě jednat o obecný informační systém, který se bude využívat při řízení činnosti politické strany.

Pro uživatele tohoto informačního systému bude vytvořena jednotná webová aplikace. Každý uživatel, který bude chtít systém používat, se do něj bude muset zaregistrovat. Systém bude o uživateli evidovat citlivé informace, jako například jejich bydliště, a proto je nutný zabezpečený přístup. Tyto informace jsou důležité pro činnost strany a z toho důvodu je nutná jejich evidence. Tyto informace lze například využít pro skládání volebních komisí při volbách do různých orgánů strany nebo můžeme tyto informace použít pro sestavení kandidátních listin.

Vytvoření informačního systému se neomezuje pouze na jeho návrh a implementaci, ale je také nutné provést ekonomické zhodnocení tohoto systému. Cílem je vytvořit systém, který podpoří a zjednoduší činnost strany a zároveň pro ni bude únosný z hlediska nákladů na budoucí údržbu a provoz.

Analýza požadavků

2.1 Analýza

Analýzou v tomto případě rozumíme rozbor budoucího systému. Snažíme se získat dostatek informací, abychom mohli provést vhodný návrh systému, který splní očekávání zákazníka a pomůže v jeho činnosti. Ve fázi analýzy sbíráme a analyzujeme požadavky na budoucí systém, popisujeme procesy, které bude systém zahrnovat, specifikujeme uživatelské role, případy užití.

2.2 Současné řešení

V oblasti informačních systémů pro politické strany se mi nepodařilo najít vhodné řešení dané problematiky, které by splnilo všechny požadavky. Existují sice různé webové aplikace pro zájmové spolky, ale tyto aplikace nepokrývají všechny požadavky kladené na webovou evidenci. Webové aplikace těchto spolků vznikají na popud právě těchto společenství a řeší problematikou spojenou s jejich činností. Proto jsem se rozhodl, že nejlepší bude vytvořit novou webovou aplikaci, která splní požadavky politické strany.

Například členové Pirátské strany řešili před léty obdobný problém a nakonec se jako nejlepší řešení ukázalo vyvinutí jejich vlastní aplikace [1].

2.3 Požadavky

Nyní se zaměřím na analýzu požadavků na webovou evidenci. Požadavky dělíme na funkční a nefunkční

2.3.1 Funkční požadavky

Funkčními požadavky rozumíme požadavky na funkce systému, které od něj budoucí uživatel očekává. Můžou se zde objevit například požadavky na ur-

číté zpracování dat, různé výpočty nebo třeba periodické generování reportů. Funkční požadavky se také dají zobrazit pomocí Use case diagramů.

2.3.1.1 Evidence členů

Jedním z nejdůležitějších požadavků na webovou evidenci je evidence členů politické strany. Informační systém umožní zadávání nových členů strany a uchování stávajících členů. Dále umožní vyhledávání a zobrazování členů v evidenci pro potřeby strany. O členech strany budou evidovány tyto údaje a položky:

1. Datum narození
2. Jméno
3. Příjmení
4. Adresa
5. E-mail
6. Členské číslo

Dále bude mít každý uživatel možnost vyplnit svou adresu. U každého uživatele bude také ukládána a sledována historie příspěvků, které straně složí.

2.3.1.2 Správa členství

Jakmile budeme mít v evidenci členy strany, bude třeba spravovat jejich členství. Identifikoval jsem následující typy členství:

1. Typ členství
 - Předseda
 - Člen předsednictva
 - Člen
 - Pozastaveno
 - Čekatel
 - Nečlen
 - Neplatné
 - Vyloučen

Typ Nečlen se bude používat v případě, že by bylo nutné evidovat v databázi osoby, jež nejsou členy strany. Toto může nastat například při evidenci volebních komisí.

2.3.1.3 Poplatky

Politická strana může vyžadovat po svých členech hrazení určitých členských poplatků, případně i členové strany mohou chtít podpořit svou stranu finančním darem. V takovém případě se zdá výhodné evidovat poplatky jednotlivých členů. U poplatků budou evidovány následující položky:

1. Výše poplatku
2. Datum připsání
3. Popis poplatku

2.3.1.4 Evidence volebních výsledků

Pro politickou stranu je samozřejmě důležité sledovat, jakých volebních výsledků v rámci svého působení dosahuje. Systém bude tudíž umožňovat uživatelům zadávat volební výsledky a bude udržovat jejich historii. V rámci volebních výsledků, jsem se rozhodl umožnit evidenci těchto položek:

1. Procentuální zisk hlasů
2. Typ voleb
3. Datum konání voleb

2.3.1.5 Evidence volebních komisí

Politická strana si rovněž přeje uchovávat informace o volebních komisích. V této situaci mohou nastat dva případy. Strana může evidovat členy ve volebních komisích ve státem vypsáných volbách a nebo členy v komisích ve vnitrostranických volbách. U volebních komisí budou evidovány následující položky:

1. Členové komise
2. Popis komise

2.3.2 Nefunkční požadavky

2.3.2.1 Uživatelské rozhraní

Součástí informačního systému bude uživatelské rozhraní, jež umožní uživatelům přístup do evidence a její využití. Před přístupem do tohoto rozhraní se uživatel bude muset nejdříve zaregistrovat. Po registraci mu bude zaslán aktivační email. Uživatel bude muset nadále vyčkat přiřazení typu členství některým pověřeným členem strany a až pak bude moci naplno využívat systém, samozřejmě s omezeními plynoucími z jeho pozice v rámci strany. Identifikoval jsem následující požadavky pro uživatelské rozhraní:

2. ANALÝZA POŽADAVKŮ

1. Uživatelské rozhraní umožní registraci nových uživatelů
2. Změna osobních údajů
3. Sledování uživatelských příspěvků
4. Umožní změnu nebo zaslání hesla při jeho zapomenutí
5. Umožní zadávání a prohlížení volebních výsledků strany
6. Umožní zadávat a prohlížet volební komise.
7. Umožní zadávat a prohlížet členy strany.
8. Umožní pohodlnou navigaci ve webové aplikaci.
9. Uživatelské rozhraní bude vytvořeno pro zobrazení desktopovými prohlížeči.

2.3.2.2 Bezpečnost

Vzhledem k povaze uchovávaných dat v evidenci je nutné myslet na bezpečnost přístupu k těmto datům. Jedním z hlavních požadavků na webovou evidenci je tudíž zprostředkování zabezpečeného připojení pomocí OpenSSL s využitím protokolu HTTPS místo obyčejného HTTP. Informační systém umožní připojení uživatelům pouze pomocí tohoto zabezpečeného protokolu.

2.3.2.3 Singularita členství

V České republice mohou být členy politických stran pouze státní občané starší 18 let a tyto osoby mohou být členy pouze jedné politické strany [2]. Bohužel jsem nenalezl žádný registr či nějakou jinou možnost, jak zjistit, jestli je občan již členem nějaké jiné strany. Jako jediné možné řešení se jeví podepsání či potvrzení čestného prohlášení žadatelem o členství ve straně.

2.3.2.4 Použití prostředí LAMPS

Informační systém poběží na serveru politické strany. Jedním z požadavků je běh webové evidence v prostředí LAMPS. Zkratka LAMPS znamená Linux, Apache, MySQL, PHP a SSL. Pro vytvoření webové evidence budu tedy muset předpokládat nasazení v Linuxovém prostředí a pro běh webového serveru Apache. Samotný informační systém se bude skládat z webové aplikace, která bude vytvořena v programovacím jazyce PHP a datovým úložištěm ve formě MySQL databáze. Z důvodu požadavků na bezpečnost připojení budu muset využít zabezpečené připojení pomocí HTTPS.

2.4 Případy užití

Případy užití neboli Use case jsou součástí analýzy informačního systému. Případy užití modelují typické situace, které mohou nastat při používání systému. Typicky se používají k popsání jedné funkcionality systému. Případy užití nicméně neříkají, jakým způsobem se dané scénáře realizují při implementaci. Dané situace popisujeme z hlediska uživatele a kroků, které musí v rámci situace učinit.[3]

2.4.1 Registrace nového uživatele

Jelikož uživatelé bez registrace nemají možnost přístupu, bude se muset každý nový uživatel zaregistrovat. Scénář registrace uživatele probíhá v následujících krocích:

1. Uživatel zvolí tlačítko „Registrace“ nebo je přesměrován
2. Systém umožní registraci pouze členům, jež jsou uvedeni v evidenci. Propojení s členským profilem bude zajištěno přihlašovacím e-mailem.
3. Pokud uživatel nevyplní některá z povinných polí formuláře, systém uživatele upozorní a bude vyžadovat opravu.
4. Pokud bude uživatelem vyplněná e-mailová adresa již uložena v systému (tzn. účet s tímto e-mailem již existuje), ten na tuto chybu upozorní a umožní uživateli přejít do formuláře pro obnovení hesla.
5. V případě, že registrace proběhne úspěšně, bude zaslán novému uživateli aktivační e-mail.
6. Nakonec uživatel aktivuje svůj účet dle instrukcí v zasláném e-mailu.

2.4.2 Nové zaslání aktivačního e-mailu

Tento scénář nastane v případě, že uživatel bude potřebovat nový aktivační e-mail. Toto může nastat v případě, že aktivační e-mail uživateli nedorazí.

1. Uživatel klikne na odkaz „Znovu zaslat aktivační e-mail“.
2. Uživatel zadá svůj přihlašovací e-mail.
3. Pokud tento e-mail není zaregistrovaný nebo je aktivovaný, dojde k vypsání chybové hlášky.
4. V případě, že je zadaný e-mail v pořádku, proběhne zaslání nového aktivačního e-mailu.

2.4.3 Přihlášení uživatele do systému

Pokud se uživatel registroval, může se přihlásit do webové evidence pomocí přihlašovací obrazovky.

1. Uživatel vyplní svůj přihlašovací e-mail a heslo.
2. Po odeslání přihlašovacích údajů uživatelem, systém ověří zadané údaje a případně vpustí uživatele do systému.
3. Pokud uživatel zadá chybné přihlašovací údaje, systém vypíše chybu.
4. Z přihlašovací obrazovky bude uživateli umožněno přejít na obrazovku pro změnu ztraceného hesla.

2.4.4 Odhlášení uživatele

Tento scénář se odehrává v případě, že se uživatel rozhodne odhlásit z webové aplikace. Samozřejmě se ze systému mohou odhlásit pouze přihlášení uživatelé.

1. Uživatel zvolí odkaz „Odhlásit se“.
2. Po úspěšném odhlášení uživatele je uživatel přesměrován zpět na přihlašovací obrazovku.

2.4.5 Změna údajů

Uživateli bude umožněno změnit si svůj e-mail nebo přihlašovací heslo.

1. Uživatel ve svém profilu zvolí tlačítko „Změnit Heslo“ nebo „Změnit E-mail“.
2. Uživateli se zobrazí příslušný formulář pro změnu údaje. Formulář bude vyžadovat zadání stávajícího hesla.
3. Systém vyhodnotí správnost zadaných údajů a případně vypíše chybovou hlášku.
4. Pokud jsou všechny údaje zadány správně, bude provedena změna osobních údajů.

2.4.6 Obnovení přihlašovacího hesla

Tento scénář nastává v případě, že se uživatel rozhodne obnovit své přihlašovací údaje, například z důvodu zapomenutí tohoto hesla.

1. Na přihlašovací obrazovce uživatel klikne na odkaz pro obnovu hesla nebo je přesměrován.

2. Uživatel zadá svůj přihlašovací e-mail.
3. Uživatel dále pokračuje instrukcemi v zasláném e-mailu.
4. Po úspěšné změně hesla je uživatel přesměrován na přihlašovací obrazovku, kde se může nyní přihlásit tímto novým heslem.

2.4.7 Zadání volebních výsledků

Systém umožní zadávání volebních výsledků uživatelům s dostatečným oprávněním.

1. Uživatel zvolí položku „Volební výsledky“ a zde zvolí „Přidat výsledek“.
2. Zobrazí se formulář pro zadání volebního výsledku.
3. Uživatel vyplní povinné údaje, například získaný počet hlasů, typ voleb, kterých se strana účastnila, apod.
4. Uživatel provede odeslání nového výsledku stiskem tlačítka „Odeslat“.
5. Systém zkontroluje správnost vyplnění a případně vypíše chybovou hlášku.
6. V případě správného vyplnění, bude nový výsledek přidán do evidence.

2.4.8 Zobrazení volebních výsledků

Uživatelé budou moci zobrazit volební výsledky dosažené stranou a zjistit tak, jak si jejich strana vede.

1. Uživatel zvolí položku „Volební výsledky“ a zde klikne na tlačítko „Zobrazit“.
2. Uživateli se zobrazí seznam volebních výsledků.
3. Uživatel bude moci tyto výsledky filtrovat a řadit.
4. Po rozkliknutí odkazu konkrétního výsledku se uživatel bude moci podívat na detaily tohoto volebního výsledku.

2.4.9 Zobrazení evidovaných členů

Webová evidence umožní pověřeným uživatelům zobrazení informací o členech.

1. Uživatel zvolí položku „Členové“ a zde klikne na tlačítko „Zobrazit členy“.
2. Uživateli bude zobrazen seznam evidovaných členů.

2.4.10 Evidence členů

Informační systém umožní evidenci členů politické strany.

1. Uživatel zvolí položku „Členové“ a zde klikne na tlačítko „Přidat člena“.
2. Uživateli bude zobrazen formulář pro přidání nového člena.
3. Uživatel vyplní údaje nově evidovaného člena - jméno, příjmení, datum narození, členské číslo, e-mailovou adresu a funkci v rámci politické strany.
4. Formulář bude po odeslání zvalidován a v případě špatného vstupu umožní opravu.
5. Po přidání člena je uživatel přesměrován na zobrazení členů a je informován o výsledku operace.

2.4.11 Členské příspěvky

Webová evidence umožní spravovat příspěvky členů strany. Příspěvky bude spravovat vybraný člen strany.

1. Uživatel klikne na tlačítko „Členové“ a zde zvolí „Zobrazit členy“.
2. Uživateli bude zobrazen seznam členů strany zaregistrovaných v evidenci.
3. Uživatel si rozklikne kartu člena strany a zde klikne na odkaz členské příspěvky.
4. Zobrazí se formulář pro zadání členského příspěvku.
5. Vyplňující zadá všechny potřebné údaje a zvolí typ příspěvku.
6. Po odeslání systém zkontroluje zadané údaje, v případě chybného vyplnění bude uživatel upozorněn.
7. Po úspěšném odeslání formuláře bude člen, jemuž je příspěvek připsán, informován například e-mailovou zprávou.

2.4.12 Zobrazení členských příspěvků

Každý uživatel bude moci sledovat historii zaplacených příspěvků.

1. Uživatel vybere položku „Příspěvky“ a zvolí odkaz „Moje příspěvky“.
2. Uživateli se zobrazí seznam jemu připsaných členských příspěvků.
3. Příspěvky bude moci filtrovat a řadit.

2.4.13 Zobrazení členského profilu

Každý registrovaný uživatel bude mít možnost zobrazit si svůj profil. Zde si bude moci zobrazit osobní údaje, které jsou uloženy v informačním systému. Pro zobrazení těchto informací uživatel klikne na odkaz „Profil“.

2.4.14 Vytvoření volební komise

Tento scénář nastává v případě, že uživatel chce vytvořit novou volební komisi v systému.

1. Uživatel zvolí položku „Volební komise“ a zde zvolí tlačítko „Přidat komisi“.
2. Zobrazí se formulář pro vyplnění údajů o volební komisi.
3. Uživatel přidá členy komise a popíše jejich roli v komisi.
4. Uživatel odešle formulář. Systém nyní zkontroluje správnost vyplnění.
5. Pokud je formulář vyplněn chybně, systém vypíše chybovou hlášku a umožní opravu.
6. Správně vyplněný formulář je zpracován a výsledek uložen do databáze.

2.5 Uživatelské role

Důležitou součástí analýzy je také určení vhodných uživatelských rolí v připravovaném systému. Uživatelské role propůjčují uživatelům oprávnění k vykonávání činností v rámci systému. Například omezují, k jakým informacím má uživatel přístup a jaká data může sám do systému zadávat. S uživatelskými rolmi a právy se setkáváme často, například operační systém nám může zakázat provádět změny v nastavení, dokud se nepřihlásíme s účtem s dostatečným oprávněním (např. administrátorský účet ve Windows). Použití uživatelských rolí se hodí zejména v systémech s přístupem většího množství uživatelů, kde by se uživatelé mohli dostat k pro informacím, které pro ně nejsou určeny nebo by mohli necitlivým zásahem změnit důležitá nastavení nebo upravit či poškodit důležitá data. Uživatelská oprávnění volíme tak, aby každý uživatel měl přístup jen k těm informacím a funkcím systému, které potřebuje pro svou činnost.

Každý uživatel bude mít po registraci roli běžného uživatele.

2.5.1 Běžný uživatel

Tato role bude zpřístupněna všem uživatelům systému po úspěšné registraci. Umožní uživatelům zobrazit a upravovat svůj profil. Uživatel si bude dále moci zobrazit historii svých členských příspěvků. Tito uživatelé mají také

oprávnění prohledávat a zobrazovat si volební výsledky strany. Tato role je určena zejména běžným členům strany.

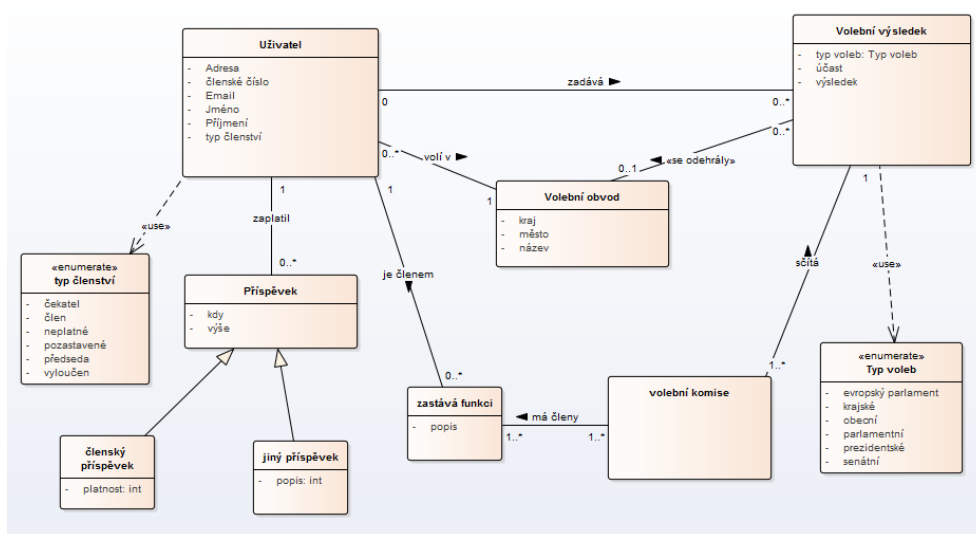
2.5.2 Pokročilý uživatel

Tato uživatelská role je určena těm členům strany, jež budou provádět změny ve webové evidenci. Tito uživatelé budou moci přidávat nové členy do systému, přidávat příspěvky již evidovaným členům. Dále budou tito uživatelé moci vytvářet nové volební komise a tyto komise upravovat. Uživatelé oprávnění touto rolí budou moci také řešit agendu spojenou s evidencí volebních výsledků.

Návrh

V této kapitole se budu zabývat návrhem informačního systému. Popisem návrhu webové aplikace bych chtěl upřesnit provedenou analýzu. Budu se zabývat například architekturou systému a použitými vývojovými nástroji.

3.1 Doménový model



Obrázek 3.1: Doménový model

Jedná se o obecný, na platformě nezávislý diagram. Právě proto, že je model nezávislý na platformě, neobjevují se v něm konkrétní datové typy u atributů. Používá se pro zobrazení business entit budoucího systému a vztahů mezi nimi. V takovém případě je možné i atributy těchto entit vynechat [4], já

jsem však rozhodl je pro větší názornost do modelu zahrnout. Tento model se nevyužívá přímo pro implementaci, ale spíše pro lepší pochopení problémové domény. Model informačního systému je zobrazen na obrázku 3.1.

3.2 Architektura

Pro webovou evidenci jsem zvolil tzv. třívrstvou architekturu. Při využití třívrstvé architektury rozdělíme aplikaci do tří logických celků, které spolu vzájemně spolupracují. Vzhledem k rozsahu práce se mi jeví rozdělení na tři vrstvy jako postačující. Cílem rozdělení aplikace na vrstvy je oddělení logických celků a také lepší pochopení daných částí. Dále toto rozdělení umožňuje větší flexibilitu v případě budoucích změn, například pokud budeme chtít změnit rozložení uživatelského rozhraní či jeho vzhled, pozměníme prezenční vrstvu a nebudeme muset například zasahovat do vrstvy datové. Třívrstvá architektura je nejběžnější aplikací vícevrstvé architektury a je často využívána při návrhu webových aplikací [5].

3.2.1 Prezenční vrstva

Prezenční vrstva slouží k zobrazování informací koncovým uživatelům a pomocí této vrstvy bude uživatel posílat své požadavky aplikaci. Cílem této vrstvy je předávat informace od uživatele ke zpracování business vrstvou a zobrazování těchto informací pro uživatele srozumitelnou cestou. Tato vrstva je jediná, jež je dostupná pro uživatele přímým odkazem.

Ná této vrstvě jsem se rozhodl využít návrhový vzor MVC - Model View Controller. Prezenční vrstva bude rozdělena právě na View a Controller. View se stará o zobrazení dat uživateli, může například obsahovat různé filtry, které ale nejsou v datové vrstvě dostupné. Cílem View je zobrazování informací a dat modelu. Controller je další částí vzoru MVC, cílem Controlleru v aplikaci je spouštět procesy a reagovat na požadavky View. Controller může například zadat Modelu změnu dat nebo aktualizovat View. [6] Například, když uživatel klikne na odkaz, je mu zobrazena nová obrazovka. Samotný model se v prezenční vrstvě nevyskytuje, nicméně v kontextu této aplikace lze za model považovat business a datovou vrstvu.

3.2.2 Business vrstva

V této vrstvě se nachází business logika aplikace. Tato vrstva přebírá požadavky a informace z prezenční vrstvy a zpracovává je. Dále má business vrstva přístup k datové vrstvě, dává této vrstvě požadavky a získaná data zpracovává. Takto zpracovaná data jsou pak poskytnuta prezenční vrstvě k zobrazení uživateli. Tato vrstva v podstatě spojuje prezenční a datovou vrstvu.

3.2.3 Datová vrstva

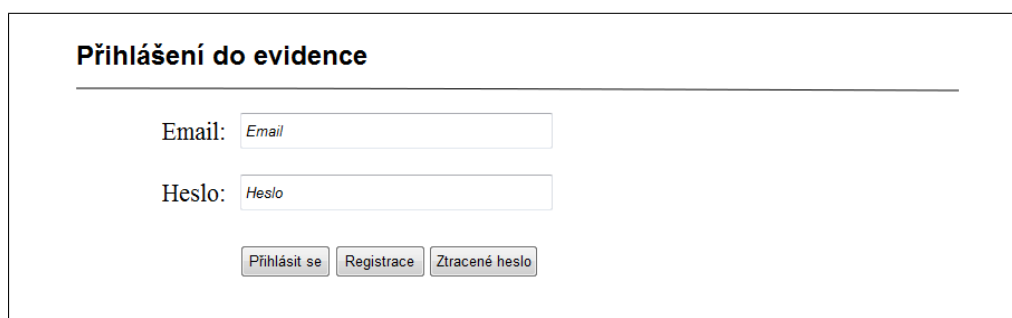
Úkolem této vrstvy je zprostředkování propojení mezi webovou aplikací a databází. Tato vrstva zpracovává požadavky business vrstvy na práci s daty. Příkladem požadavku business vrstvy může být například vyhledání určitých dat v databázi, datová vrstva pak sestaví patřičný příkaz, například SQL dotaz, a získaná data předá business vrstvě ke zpracování. Vzhledem k rozsahu webové aplikace bude tato vrstva pracovat pouze s jednou MySQL databází, a proto nebude potřeba řešit synchronizaci dat a rozdělení úkolů mezi více databází.

3.3 Uživatelské rozhraní

Do návrhu své práce jsem se rozhodl zahrnout také návrh uživatelského rozhraní webové aplikace. V této sekci tedy popíšu obrazovky budoucí aplikace a jejich vzájemné návaznosti. Uživatelské rozhraní je důležitou součástí webové aplikace a mělo by zohlednit funkční požadavky popsané v kapitole Analýza.

3.3.1 Přihlašovací obrazovka

První obrazovku, kterou uživatel uvidí při pokusu vstoupit do webové evidence, bude právě přihlašovací obrazovka. Přihlašovací obrazovka obsahuje dvě textová pole, kam uživatel vyplní svůj přihlašovací e-mail a heslo. Po stisknutí tlačítka přihlásit je provedena validace formuláře. Pokud je některé z textových polí vyplněno nesprávně, objeví se u něj chybová hláška s popisem chyby. V případě úspěšné autentifikace uživatele bude uživatel přesměrován na hlavní stránku webové evidence. Uživatel má dále možnost přejít z této obrazovky na registrační formulář, případně na formulář pro změnu zapomenutého hesla.



Přihlášení do evidence

Email:

Heslo:

Obrázek 3.2: Přihlašovací obrazovka

3.3.2 Uživatelský profil

Na této obrazovce má uživatel možnost nechat si zobrazit své údaje v evidenci. Na obrazovce se nacházejí dvě tabulky, první zobrazuje osobní údaje uživatele, druhá pak jeho adresu. Ke každé tabulce je také přidruжено tlačítko, jež umožní uživateli změnit své osobní údaje. Navigační panel v horní části umožní pohodlnou navigaci skrze systém a zobrazuje právě zobrazenou obrazovku stínem. Tato obrazovka je dostupná všem uživatelům webové aplikace.

Webová evidence Index **Profil** Moje příspěvky Členové Volební komise Volební výsledky

Uživatelský profil

Osobní údaje

Jméno:	Petr	Příjmení:	Novák
Členské číslo:	0000	Typ členství:	Člen
Datum narození:	01. 12. 1900	Email:	example@email.com

Upravit

Adresa

Ulice:	Pod lipami 25		
Město:	Praha	PSČ:	16000
Okres:	Praha 1	Kraj:	Praha

Upravit

Obrázek 3.3: Uživatelský profil

3.3.3 Obrazovka Moje příspěvky

Na této obrazovce se uživatelům zobrazí jim připsané členské příspěvky. Hlavní součástí této obrazovky je interaktivní tabulka, která umožňuje uživateli pohodlné filtrování příspěvků. Vyhledávací box umožňuje vyhledávání ve všech sloupcích. Uživateli je dále umožněno změnit počet zobrazených výsledků v tabulce a přecházet mezi jednotlivými stranami výsledků pomocí navigačních tlačítek ve spodní pravé straně. Tato obrazovka je také dostupná všem uživatelům.

Webová evidence Index Profil **Moje příspěvky** Členové Volební komise Volební výsledky

Moje příspěvky

Zobrazit záznamu na stranu Vyhledat

Datum	Částka	Popis	Přidal
1. 4. 2016:	500	Členský příspěvek	5
1. 5. 2016:	500	Členský příspěvek	5
Datum	Částka	Popis	Přidal

Předchozí 1 Další

Obrázek 3.4: Obrazovka moje příspěvky

3.3.4 Obrazovka Členové

Tato obrazovka je dostupná pouze pro uživatele s rolí Pokročilý uživatel viz. 2.5.2. Na této obrazovce uživatelé řeší agendu spojenou s členy strany. V počátečním stavu tato obrazovka zobrazuje členy politické strany v interaktivní tabulce. Po klepnutí na řádek tabulky se tento zvýrazní a uživateli se objeví nové možnosti volby ve formě tlačítek 3.5. Po klepnutí na tlačítko „Zobrazit details“ se uživateli zobrazí profil zvoleného člena a historie jeho členských příspěvků 3.6. Příspěvky jsou zobrazeny formou interaktivní tabulky, jež po vybrání příspěvku zvýrazní řádek a umožní uživateli příspěvek odstranit. Stisknutím tlačítka „Zavřít“ se uživatel vrátí zpět na zobrazení všech členů v evidenci.

3. NÁVRH

Webová evidence Index Profil Moje příspěvky **Členové** Volební komise Volební výsledky

Členská základna

Přidat člena

Zobraz detaily Přidat příspěvek Změnit členství

Členové

Zobrazit 10 záznamu na stranu Vyhledat

Jméno	Příjmení	Typ členství	Členské číslo	Email	Datum narození
Antonín	Těžký	Nečlen	175	anton@email.com	1. 5. 1967
Jana	Nová	Člen předsednictva	2719	jnova@email.com	25.4.1974
Jméno	Příjmení	Typ členství	Členské číslo	Email	Datum narození

Předchozí 1 Další

Obrázek 3.5: Obrazovka členové

Webová evidence	Index	Profil	Moje příspěvky	Členové	Volební komise	Volební výsledky
------------------------	-------	--------	----------------	----------------	----------------	------------------

Členská základna

Zavít

Členský profil

Jméno:	Petr	Příjmení:	Novák
Členské číslo:	0000	Typ členství:	Člen
Datum narození:	01. 12. 1900	Email:	example@email.com

Adresa

Ulice:	Pod lipami 25		
Město:	Praha	PSČ:	16000
Okres:	Praha 1	Kraj:	Praha

Členské příspěvky

Zobrazit záznamu na stranu

Datum	Částka	Popis	Přidal
1. 4. 2016:	500	Členský příspěvek	5
1. 5. 2016:	500	Členský příspěvek	5
Datum	Částka	Popis	Přidal

Obrázek 3.6: Obrazovka členové - detail člena

3. NÁVRH

3.3.5 Obrazovka - Volební komise

Tato obrazovka je opět dostupná pouze pro uživatele s rolí Pokročilý uživatel 2.5.2. Součástí této obrazovky jsou dvě interaktivní tabulky. První tabulka zobrazuje seznam volebních komisí a po kliknutí na řádek tabulky se tento zvýrazní a umožní uživateli zvolit nové možnosti - přidání nového člena do komise, případně upravit popis komise. Stiskem tlačítka „Přidat komisi“ se místo tabulky zobrazí formulář pro přidání komise.

Druhá tabulka zobrazuje členy jednotlivých komisí. Opět se jedná o interaktivní tabulku, jež umožňuje vyhledávání ve všech sloupcích a řazení řádků dle jednotlivých sloupců. Po kliknutí do řádku tabulky, se obdobně jako u první tabulky tento řádek zvýrazní a zobrazí uživateli možnost odstranit člena z komise.

Webová evidence Index Profil Moje příspěvky Členové **Volební komise** Volební výsledky

Volební komise

Zobrazit záznamu na stranu

ID	Popis	Počet členů
1	2. sjezd strany	2
2	Volby 2010, Okrsek č. 5	2

Zobrazit záznamu na stranu

Členské číslo	Jméno	Příjmení	Funkce v komisi	Popis	Číslo komise
175	Jana	Nová	Předseda	Sčítá hlasy	1
2719	Antonín	Těžký	Člen	Kontroluje OP	1

Obrázek 3.7: Obrazovka Volební komise

3.3.6 Obrazovka - Volební výsledky

Tato obrazovka je dostupná pouze pro uživatele s rolí Pokročilý uživatel 2.5.2. Hlavní součástí této obrazovky tvoří interaktivní tabulka, jež se zobrazí po stisknutí tlačítka „Zobrazit volební výsledky“. Po kliknutí do řádku tabulky je zvolený řádek zvýrazněn a objeví se tlačítko umožňující smazat zvolený výsledek. Pro přidání nového volebního výsledku musí uživatel stisknout tlačítko „Přidat volební výsledek“ se místo tabulky zobrazí formulář pro přidání volebního výsledku.

Webová evidence Index Profil Moje příspěvky Členové Volební komise **Volební výsledky**

Volební výsledky

Zobrazit volební výsledky Přidat volební výsledek Zavřít

Odstranit výsledek

Zobrazit 10 záznamů na stranu Vyhledat:

Datum voleb	Zisk hlasů v %	Popis	Typ voleb	Přidán členem č.
2010-01-01	32.25	Volby do místních samospráv	Obecní	1234
2011-10-15	37.25	Krajské volby	Krajské	1234
Datum voleb	Zisk hlasů v %	Popis	Typ voleb	Přidán členem č.

Předchozí 1 Další

Obrázek 3.8: Obrazovka Volebních výsledků

Použité nástroje

Nyní popíšu zvolené vývojové nástroje, velká část použitých technologií byla dána požadavkem na vývoj v prostředí LAMP. Další nástroje jsem vybral na základě předchozích zkušeností ze studia a požadavků na funkčnost systému.

4.1 HTML

HTML je zkratka pro HyperText Markup Language. Jedná se o značkovací jazyk, jež se používá pro tvorbu a strukturování webových dokumentů. Základem jazyka jsou značky, tzv. tagy. Jednotlivé tagy určují webovému prohlížeči, jakým způsobem má zobrazit zpracovaný text. Značky tak umožňují v dokumentu vytvářet nadpisy, odstavce, tabulky, odkazy na další webové dokumenty apod. Cílem HTML není vytvořit výsledný vzhled webové stránky, ale pouze strukturu dokumentu.[7]

4.2 CSS

CSS, neboli Cascading Style Sheets, se využívá pro vytvoření výsledného rozvržení a vzhledu HTML dokumentu. Existuje více způsobů, jak aplikovat CSS kód na HTML dokument, nicméně nejčastěji se využívá oddělený soubor s příponou .css. Výhodou tohoto řešení je možnost aplikovat formátování pomocí jednoho .css souboru na více HTML dokumentů, propojení se pak realizuje pomocí speciálního odkazu. Tímto způsobem můžeme zajistit jednotný vzhled více HTML dokumentům. Navíc ušetříme čas, kdy změnu vzhledu provedeme změnou v příslušném CSS souboru, namísto abychom měnili jednotlivé HTML soubory.[8][9]

4.3 Bootstrap

Twitter Bootstrap je framework pro tvorbu vzhledu webových stránek. Jedná se o framework, který v sobě kombinuje CSS, javascript a připravené fonty. Vývojář pak může využít sadu předpřipravených tříd pro vytvoření celkového vzhledu webové stránky. Použití Bootstrapu je jednodušší a intuitivnější než využití běžného CSS a umožňuje vytvořit moderní webové stránky i vývojářům, jež se tolik nevyznají v designu. Výhodou je i jistá multiplatformnost, kdy je možné s využitím Bootstrapu vytvořit vzhled nejen pro desktopové prohlížeče, ale i pro mobilní zařízení.[10]

4.4 JQuery

Jedná se o javascriptovou knihovnu, která značně zjednodušuje práci s javascriptem. JQuery umožňuje vývojáři odchylovat události v dokumentu (například kliknutí na tlačítko), i když tyto události nejsou přímo v HTML dokumentu definovány. JQuery tedy umožňuje oddělit funkčnost od struktury dokumentu. JQuery také nabízí možnost dynamicky měnit vzhled stránky na základě uživatelských akcí a to bez nutnosti stránku znovu načíst ze serveru.[11]

4.5 DataTables

Jedná se o JQuery plugin, který z běžné HTML tabulky vytvoří interaktivní tabulku s možností vyhledávání, srovnávání výsledků v jednotlivých sloupcích, omezení počtu řádků na stránce atd. Výhodou je značná konfigurovatelnost funkčnosti výsledné tabulky, včetně možnosti lokalizace. Výslednou tabulku lze také stylizovat pomocí Bootstrapu a tím zajistit konzistentní vzhled webové aplikace.[12]

4.6 MySQL

Jedná se o databázový stroj sloužící k ukládání dat. Pro práci s databází využíváme jazyka SQL. MySQL je relační databáze, to znamená, že jednotlivá data jsou uložena do tabulek a vztahy mezi jednotlivými tabulkami jsou zajištěny pomocí společných atributů, tzv. cizích klíčů. Pro administraci MySQL databáze se používá webová aplikace phpMyAdmin. PhpMyAdmin umožňuje vytvářet účty pro připojení k databázi, spouštět SQL příkazy, měnit strukturu tabulek atd.[13]

Samotná MySQL databáze je velmi populární a multiplatformní [14], a proto je široce podporována knihovnami programovacích jazyků včetně PHP.

4.7 PHP

PHP: Hypertext Preprocessor je programovací jazyk, jež nalézá nejvíce uplatnění při vývoji dynamických webových stránek a webových aplikací. PHP script běží na serverové straně, při připojení uživatele se spustí příslušný script, který vygeneruje požadovaný HTML dokument. Uživatel v podstatě o běhu PHP netuší, dostává se mu totiž stejné odpovědi, jako kdyby na serveru byla umístěna obyčejná HTML stránka. Pro běh PHP je potřeba využít některý z dostupných PHP serverů, například Apache [15].

4.8 PHPmailer

Jedná se o PHP knihovnu sloužící k odesílání e-mailů. Pomocí této knihovny lze odeslat jak běžné e-maily, tak i předpřipravené HTML e-maily. Před odesláním e-mailu lze také nastavit SMTP server, pomocí kterého dojde k odeslání zprávy. Takto je možné využít stávající e-mailovou adresu bez nutnosti zřizovat si vlastní SMTP server.[16]

Realizace

V této kapitole se budu zabývat samotnou implementací webové evidence. Nejdříve se zaměřím na popis databáze a poté na webovou aplikaci.

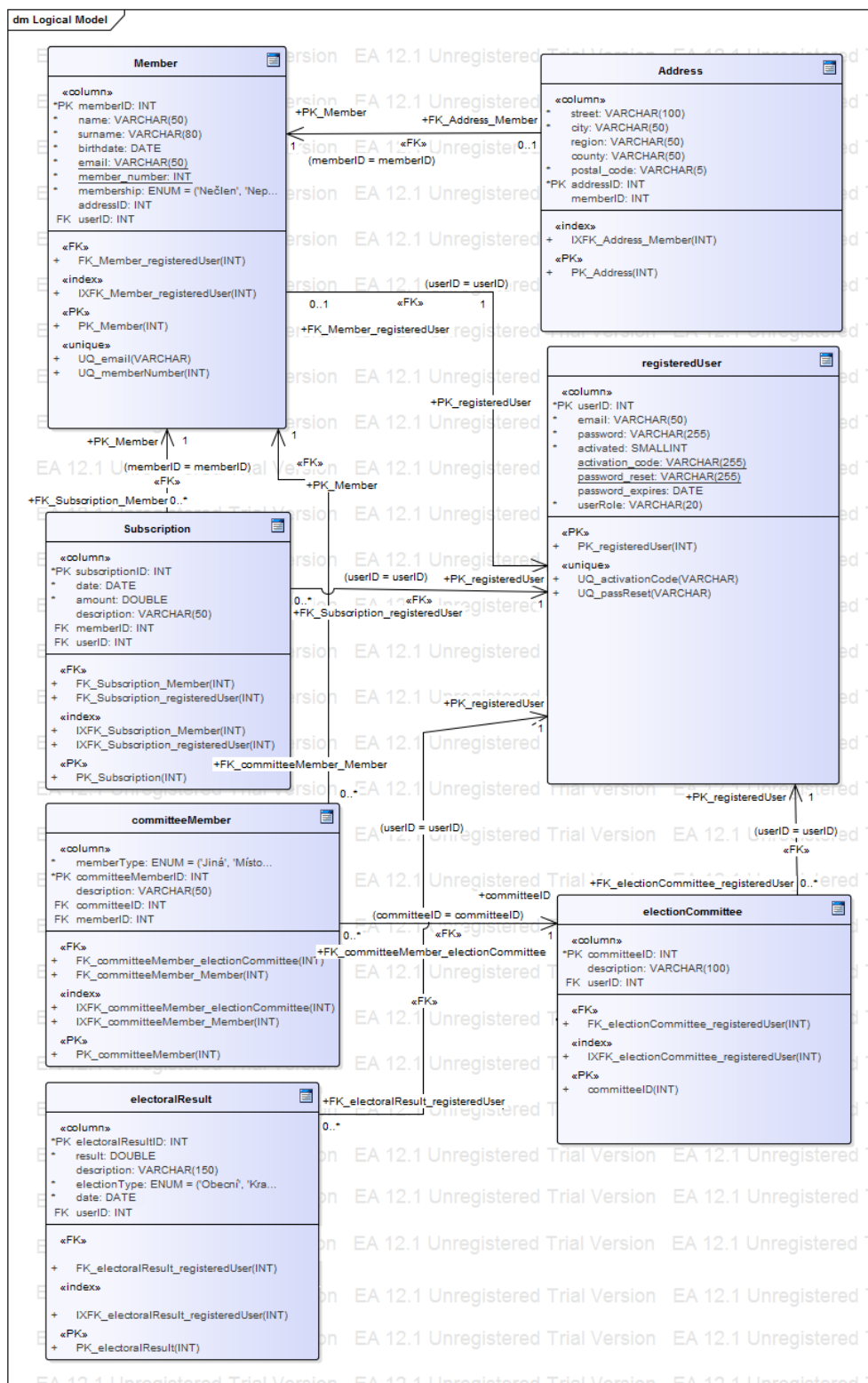
5.1 Ukládání dat

Jak jsem již popsal v analytické části 2.3.2.4, data budou v rámci informačního systému ukládána do MySQL databáze. Následuje popis logického modelu databáze a jednotlivých tabulek.

5.1.1 Databázový model

Na obrázku 5.1 je zobrazen logický model databáze, která bude sloužit pro ukládání dat v rámci informačního systému. V podstatě se jedná o implementaci doménového modelu 3.1 formou relační databáze. Z důvodu podpory cizích klíčů je potřeba nastavit typ tabulek na InnoDB. MySQL používá InnoDB tabulky implicitně od verze 5.5 [17]. Volba MySQL databáze mi umožnila použít výčtový typ ENUM. Protože z modelu není přímo vidět jakých hodnot mohou jednotlivé výčty nabývat, rozhodl jsem se je popsat u jednotlivých tabulek. Výčtový typ Enum jsem v jednotlivých případech využil pro jednodušší vyhledávání a filtraci dat uživatelem. Navíc je možné jednotlivé hodnoty výčtu načíst rovnou z databáze a tím například umožnit uživateli jednodušší vyplňování formulářů pomocí výberových polí (například 5.21).

5. REALIZACE



5.1.2 Tabulky databáze

5.1.2.1 Tabulka Member

V této tabulce jsou ukládána data o jednotlivých členech a tím je zajištěn funkční požadavek 2.3.1.1.

- **memberID** Identifikátor uloženého člena, primární klíč
- **name** Jméno evidované osoby
- **surname** Příjmení evidované osoby
- **birthdate** Datum narození evidované osoby
- **email** E-mailová adresa evidované osoby
- **member_number** Členské číslo
- **addressID** Identifikátor adresy, jež náleží tomuto členu, cizí klíč

Dále tabulka obsahuje výčet typu členství, jež může evidovaný člen v systému nabývat. Ve výčtu je zahrnut i typ Nečlen, jež se používá pro členy komisí, kteří ale nejsou členy strany. Díky tomuto atributu může strana spravovat členství jednotlivých členů 2.3.1.2. Výčet může nabývat následujících hodnot:

1. Předseda
2. Člen předsednictva
3. Člen
4. Pozastaveno
5. Čekatel
6. Nečlen
7. Neplatné
8. Vyloučen

5.1.2.2 Tabulka Address

Tato tabulka slouží k ukládání dat o adrese jednotlivých členů. Tabulka obsahuje následující atributy:

- **addressID** Identifikátor uložené adresy, primární klíč
- **street** Ulice a číslo domu, kde člen bydlí

- **city** Město
- **region** Kraj
- **county** Okres
- **postal_code** PSČ

5.1.2.3 Tabulka `committeMember`

Tato tabulka slouží k vytvoření vazby člena na komisi v níž zastává funkci. Jedná se o dekompozici stavu M:N mezi tabulkami `Member` a `electionCommittee`. Jedna komise tudíž může mít více členů a evidovaný člen může být součástí více komisí.

- **committeMemberID** Identifikátor člena komise, primární klíč
- **description** Popis působení člena v komisi
- **committeeID** Identifikátor komise, ke které člen patří, cizí klíč
- **memberID** Identifikátor členského profilu, cizí klíč

Pro jednodušší odlišení funkce v rámci komise jsem se rozhodl zde také zavést výčet pozic, jež může člen komise zastávat:

1. Předseda
2. Místopředseda
3. Člen
4. Jiná

5.1.2.4 Tabulka `electionCommittee`

Tato entita slouží pro ukládání dat o volebních komisích 2.3.1.5. Tabulka obsahuje následující sloupce:

- **committeeID** Identifikátor komise, primární klíč
- **description** Popis této volební komise
- **userID** Identifikátor uživatele, jež přidal tuto komisi, cizí klíč

5.1.2.5 Tabulka `electoralResult`

Tato tabulka slouží pro uchovávání volebních výsledků 2.3.1.4.

- **electoralResultID** Identifikátor volebního výsledku, primární klíč
- **result** Stranou dosažený volební výsledek
- **description** Popis volebního výsledku
- **date** Datum konání voleb
- **userID** Uživatel, jež přidal výsledek, cizí klíč

Dále tato tabulka obsahuje výčtový atribut `electionType`. Tento výčet nalezne využití při vytváření volebních výsledků, kdy se pomocí tohoto výčtu dají jednoduše oddělit různé druhy voleb. `electionType` nabývá následujících hodnot:

1. Obecní
2. Krajské
3. Celostátní
4. Evropské
5. Prezidentské
6. Stranické

5.1.2.6 Tabulka `Subscription`

Tato tabulka slouží pro ukládání členských příspěvků 2.3.1.3.

- **subscriptionID** Identifikátor příspěvku, primární klíč
- **date** Datum zapsání příspěvku
- **description** Popis příspěvku
- **memberID** Člen, jemuž je příspěvek připsán
- **userID** Uživatel, který příspěvek přidal, cizí klíč

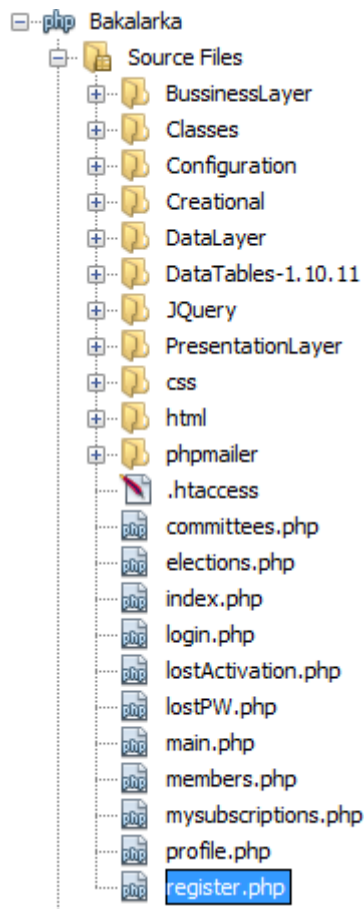
5.1.2.7 Tabulka `registeredUser`

Tato tabulka přímo nesouvisí s evidencí činnosti politické strany a používá se pro vytváření uživatelských účtů.

- **userID** Identifikátor uživatele, primární klíč
- **email** Přihlašovací email uživatele
- **password** Zahashované heslo uživatele se solí
- **activated** Tento atribut nabývá hodnoty 0 nebo 1, 0 znamená, že účet nebyl aktivován
- **activation_code** Aktivační kód, jež se zasílá uživateli pro aktivaci účtu
- **password_reset** Kód jež je zaslán uživateli při resetu zapomenutého hesla
- **password_expires** Datum, jímž lze omezit platnost hesla
- **userRole** Uživatelská role, opravňuje uživatele používat určité části systému

5.2 Struktura webové aplikace

V této části popíšu strukturu webové aplikace, rozdělení jednotlivých tříd a souborů do příslušných človek. Toto členění jsem volil dle logických celků implementace webové aplikace. Například soubory datové vrstvy architektury jsou sdruženy ve složce `DataLayer`. Stejně tak externí knihovny mají své oddělené adresáře.



Obrázek 5.2: Adresářová struktura

5.2.1 Architektura

Jak už jsem zmínil v návrhu viz. 3.2 při vývoji webové aplikace jsem využil třívrstvou architekturu. Jednotlivé vrstvy architektury jsou rozděleny do příslušných adresářů a to PresentationLayer, BussinessLayer a DataLayer.

5.2.1.1 Složka PresentationLayer

Prezentační vrstva se stará o uživatelské rozhraní. Zobrazuje informace z webové evidence a reaguje na uživatelské vstupy a požadavky. Prezentační vrstva se nachází v adresáři PresentationLayer. V tomto adresáři se nacházejí php soubory, které řeší operace uživatelského rozhraní. Soubory lze rozdělit na dva druhy a to *View a *Controller, například loginView a loginController. Soubory typu View se starají o zobrazení dat uživateli a obsahují potřebnou logiku. Soubory typu Controller definují třídy, jež se pak starají zejména o nastavení odpovídajícího View, tedy to co bude v dané situaci zobrazovat. Třídy typu

Controller dále obsahují metody, jež se volají v případě uživatelského vstupu a řeší operace s ním spojené. Prezentační vrstva byla vytvořena pro použití desktopovými internetovými prohlížeči a není optimalizována pro zobrazení na přenosných zařízeních.

Dále se zde nacházejí adresáře `PresentationLayer/Forms` a `PresentationLayer/Tables`. Ty obsahují formuláře a tabulky, z nichž se pak vytváří složitější View, jako například `membersView`.

5.2.1.2 Složka `BussinessLayer`

Součástí této vrstvy je aplikační logika. Business vrstva se nachází v adresáři `BussinessLayer` a sestává se z php souborů typu `*Managment` například `userManagement`. Vyjímkou je soubor `session.php`, v tomto souboru se kontroluje, zda-li je uživatel přihlášený a dále obsahuje logiku spojenou s odhlášením uživatele v případě nečinnosti. Management soubory obsahují logiku vztahenou k určitému celku aplikace, například soubor `userManagement` obsahuje třídu `userManager`, jež řeší logiku spojenou s uživatelskými účty, což je například přihlašování do evidence nebo registrace nových účtů. V případě potřeby jsou z této vrstvy volány metody tříd datové vrstvy k provedení změn v databázi, případně pro získání dat z databáze.

5.2.1.3 Složka `DataLayer`

Datová vrstva obsahuje třídy, které vyvolávají databázové operace. Soubory datové vrstvy jsou uloženy v adresáři `DataLayer` a jsou pojmenovány `*DAO`, například `userDAO`. Na základě požadavků business vrstvy zasílá datová vrstva požadavky MySQL databázi. Pro práci s databází se používá ovladač PHP PDO. Po vykonání požadavku datová vrstva informuje business vrstvu o výsledku operace, případně zašle data získaná z databáze. Vytvoření připojení k databázi je řešeno třídou `dbFactory`, jež popíšu dále v této kapitole 5.2.3.1.

5.2.2 Složka `Classess`

V této složce najdeme soubory s definicí tříd. Tyto třídy svým způsobem odpovídají určitým tabulkám z databázového modelu. Tyto třídy jsou v aplikaci využívány pro přenos dat mezi jednotlivými vrstvami. Instancie objektů těchto tříd zpravidla probíhá pomocí příslušných továren. Nyní jsou v tomto adresáři soubory s definicí následujících tříd: `address`, `committeeMember`, `electionCommittee`, `member` a `subscription`.

Také se zde nachází soubor `phpFileAutoload`, který obsahuje definice metod pro slinkování jednotlivých souborů napříč celou aplikací.

5.2.3 Složka Creational

V této složce se nacházejí php soubory, jež definují tovární třídy. Jedná se o jednoduché třídy sloužící k instanciaci objektů. Momentálně se v tomto adresáři nacházejí následující soubory:

5.2.3.1 dbFactory

Tento soubor obsahuje stejnojmennou třídu dbFactory. Tato továrna je určena pro vytvoření databázového spojení pro jednotlivé DAO objekty datové vrstvy aplikace. Databázové připojení je řešeno návrhovým vzorem Singleton, a to z toho důvodu, aby každému uživateli bylo otevřeno pouze jedno databázové spojení a neotevírala se zbytečně nová připojení pro každý požadavek uživatele. Výhodou je také sjednocení logiky ohledně připojení k databázi na jednom místě aplikace.

5.2.4 electionFactory

Třída electionFactory řeší logiku okolo vytváření objektů vztahujících se k volebním komisím a volebním výsledkům. Vytvářejí se zde objekty typu committeeMember, electionCommittee a electionResult.

5.2.5 memberFactory

Třída memberFactory se používá k instanciaci objektů vztahujících se ke správě členů. Pomocí této továrny se vytvářejí objekty member, address a subscription.

5.2.6 userFactory

Úkolem této třídy je instanciovat třídu userManager pro správu uživatelských účtů.

5.2.7 Složka Configuration

Tato složka obsahuje soubor cfg.php, který se používá pro nastavení webové aplikace. Zde lze nastavit připojení k databázi a také informace potřebné pro odesílání e-mailů z aplikace.

5.2.8 Omezení vnějšího přístupu do složek

V rámci přístupu do webové aplikace jsem omezil přístup do většiny složek aplikace pomocí souboru .htaccess [18]. Pokud se uživatel pokusí připojit pomocí odkazu, jež by vedl do jedné ze složek: BusinessLayer, Classes, Configuration, Creational, DataLayer, PresentationLayer, phpMailer nebo jejich podložek, bude mu přístup odmítnut s HTTP kódem 403.

5.3 Popis webové aplikace

5.3.1 Přihlášení

Přihlášení do aplikace je řešeno přihlašovací obrazovkou. Pokud se uživatel odhlásí z webové aplikace nebo je odhlášen z důvodu neaktivity, je automaticky přesměrován na tuto obrazovku. Z této obrazovky je možné dále přejít na registrační obrazovku stiskem tlačítka „Registrace“, případně na obrazovku pro obnovu hesla stiskem tlačítka „Ztracené heslo“. Dále je možné přejít na obrazovku pro obnovu aktivačního e-mailu pomocí tlačítka „Nový aktivační e-mail“. Samotná funkcionality se nachází v souborech `loginView.php` a `loginController.php` ve složce `PresentationLayer`. Před odesláním přihlašovacího formuláře probíhá na straně uživatele validace pomocí HTML5, validuje se:

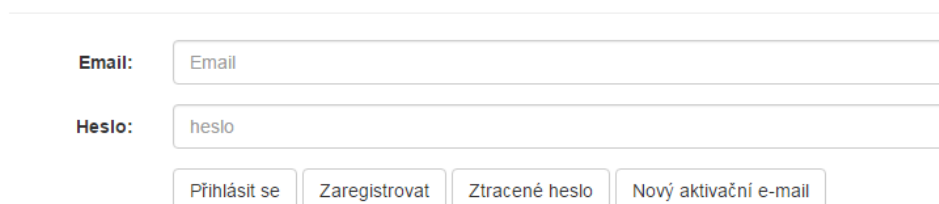
1. Vyplnění povinných polí
2. Správný formát e-mailu

V tomto případě jsou obě pole povinná. Stejná validace probíhá i na serverové straně. V případě úspěšné validace `loginController` požádá business vrstvu o autentizaci uživatele. Autentizace je řešena metodou třídy `userManager`. V případě selhání autentizace je uživateli vypsána chybová hláška. Autentizace může selhat v následujících případech:

- Neexistující uživatelský účet
- Špatně zadané přihlašovací údaje
- Účet nebyl doposud aktivován

Pokud uživatel zadá správné přihlašovací údaje, je vpuštěn do webové aplikace. Dále dojde k nastavení proměnné `$_SESSION`, do které se uloží id uživatele a jeho uživatelská role. Tato obrazovka řeší případ užití 2.4.3.

Přihlášení do evidence



The image shows a login form with two input fields: "Email" and "Heslo". Below the fields are four buttons: "Přihlásit se", "Zaregistrovat", "Ztracené heslo", and "Nový aktivační e-mail".

Obrázek 5.3: Přihlašovací obrazovka

5.3.2 Registrace uživatele

Každý uživatel, který chce používat webovou aplikaci se musí zaregistrovat 2.4.1. Registrace probíhá na registrační stránce `register.php`, z tohoto souboru jsou načteny soubory `registerView.php` a `registerController.php`, jež obsahují samotný vzhled a funkcionalitu. Validace probíhá obdobně jako v případě přihlašovací obrazovky. Na serverové straně se navíc kontroluje správnost zadání stejného hesla. Minimální délku hesla jsem stanovil na 6 znaků. V případě úspěšné validace třída `registerController` zavolá metodu třídy `userManager`, jež provede registraci uživatele. Po dokončení registrace je uživateli zobrazena jedna z následujících zpráv:

- Zadaný email již existuje!
- Účet se nepodařilo vytvořit.
- Účet vytvořen, aktivační e-mail se nepodařilo odeslat.
- Účet zaregistrován, aktivujte účet pomocí aktivačního e-mailu.

Uživatel dokončí svou registraci aktivačním e-mailem. Aktivační e-mail obsahuje odkaz, který po kliknutí provede aktivaci uživatelského účtu. Tímto je zajištěno, že se mohou registrovat pouze uživatelé s platným e-mailem.

Registrace do evidence

* povinné pole



*Email:

*Heslo:

*Zopakujte heslo:

Obrázek 5.4: Registrační obrazovka

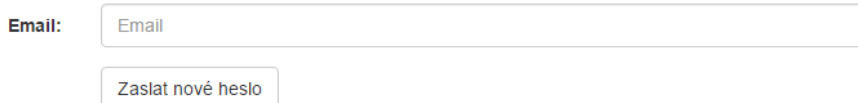
5.3.3 Obnovení hesla

Obnovení hesla 2.4.6 probíhá pomocí formuláře na obrazovce pro obnovení hesla, která se nachází na stránce `lostPW.php`. O samotný vzhled se stará

soubor `lostPWView.php` a o funkcionalitu pak třída v souboru `lostPWController.php`. Validace opět probíhá stejně jako v předchozích případech. Na uživatelské straně se zkontroluje vyplnění a správný formát e-mailu a na serverové straně se odehrává totéž. Proces obnovy hesla začíná třídou `lostPWController` jež volá příslušnou metodu třídy `userManager`. V případě, že zadaný e-mail neodpovídá žádnému registrovanému účtu, je uživateli vypsána chybová hláška a uživatel může e-mail zadat znovu.

Pokud uživatel zadal svůj e-mail správně, zobrazí se mu zpráva potvrzující zaslání e-mailu s pokyny pro změnu hesla. Po kliknutí na odkaz v e-mailu je uživatel přesměrován na obrazovku obnovení hesla, nově se zde zobrazuje formulář pro změnu hesla 5.6. Uživatel zde vyplní nové heslo a po úspěšné změně je přesměrován na přihlašovací obrazovku, kde je o provedení změny informován. Pokud se změna hesla nepovede, například z důvodu nesprávně zadaného hesla, uživatel není přesměrován a zobrazí se mu chybová hláška. Validace probíhá stejně jako validace hesla při registraci.

Reset hesla



The screenshot shows a web form titled "Reset hesla". It features a label "Email:" followed by a text input field containing the placeholder text "Email". Below the input field is a button labeled "Zaslat nové heslo".

Obrázek 5.5: Obrazovka obnovení hesla - zadání e-mailu pro obnovu

Reset hesla



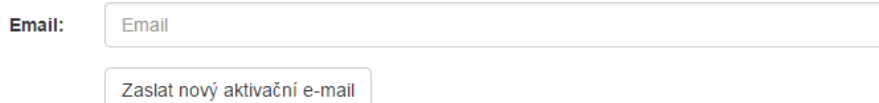
The screenshot shows a web form titled "Reset hesla". It features two labels: "Heslo:" and "Zopakujte heslo:". The "Heslo:" label is followed by a text input field containing the placeholder text "heslo". The "Zopakujte heslo:" label is followed by a text input field containing the placeholder text "heslo znovu". Below the input fields is a button labeled "Změnit heslo".

Obrázek 5.6: Obrazovka obnovení hesla - zadání nového hesla

5.3.4 Nový aktivační e-mail

Uživatel dosud neaktivovaného účtu má možnost nechat si zaslat nový aktivační e-mail. Uživatel je poté přesměrován na obrazovku pro obnovu aktivačního e-mailu. Zde uživatel zadá e-mail na nějž si přeje zaslat nový aktivační e-mail. Pokud je účet s tímto e-mail již aktivován nebo není vůbec zaregistrován viz 2.4.2, dojde k vypsání chybové hlášky. Zadání e-mailu je validováno na straně uživatele pomocí HTML5, tak na straně serveru. Pokud je zadaný e-mail v pořádku, informační systém vygeneruje nový aktivační e-mail a zašle ho uživateli. Uživatel pak ztrácí možnost aktivovat účet pomocí starého aktivačního e-mailu.

Nový aktivační e-mail



The image shows a web form titled "Nový aktivační e-mail". It consists of a label "Email:" on the left, followed by a text input field with the placeholder text "Email". Below the input field is a button with the text "Zaslat nový aktivační e-mail".

Obrázek 5.7: Obrazovka obnovení aktivačního e-mailu - zadání e-mailu

5.3.5 Zobrazení uživatelského profilu

Každý uživatel webové aplikace má možnost zobrazení svého uživatelského profilu 2.4.13. Zde má uživatel možnost změnit si svůj přihlašovací e-mail a heslo. Dále má možnost uživatel změnit svou adresu. Pokud se nepodaří získat uživatelská data z databáze, místo tabulek se zobrazí pouze chybová hláška. Obrazovka uživatelského profilu je dostupná ze stránky profile.php. O vzhled se stará soubor profileView.php. O načtení stránky v požadovaném stavu se stará třída profileController. V případě, že uživatel klikne například na tlačítko „Změnit Adresu“, třída profileController aktualizuje profileView tak, že se místo tabulky s adresou zobrazí formulář pro změnu adresy. Navigační pruh v horní části obrazovky se načítá ze souboru navbar.php.

Data do jednotlivých tabulek načítá profileController pomocí metod třídy memberManager. Data jsou do profileView předávána formou asociativních polí.

5. REALIZACE

[Webová evidence](#) [Úvod](#) [Profil](#) [Moje příspěvky](#) [Členové](#) [Volební komise](#) [Volební výsledky](#) [Odhlásit se](#)

Uživatelský profil

Osobní údaje

Jméno:	Jonáš	Příjmení:	Plukovník
Členské číslo:	2568	Typ členství:	Člen předsednictva
Datum narození:	04. May 1987	Emailová adresa:	activeUser@test.com

Vaše Adresa

Ulice:	Kolobanova		
Město:	Město	PSČ:	10004
Okres:	Okres	Kraj:	Kraj

Obrázek 5.8: Zobrazení uživatelského profilu

5.3.5.1 Změna adresy

Po kliknutí na tlačítko „Změnit adresu“ třída `profileController` změní nastavení `profileView` a dojde k překreslení stránky. Uživatel může vyplnit svou novou adresu v nově zobrazeném formuláři 5.9. Tento formulář je načten ze souboru `addAddressForm.php`, soubor `profileView.php` je díky tomu přehlednější. Validace probíhá pouze na straně uživatele pomocí HTML5 a je vyžadováno vyplnění všech polí. U pole PSČ je zároveň zkontrolován pětimístný formát čísla. Odeslání formuláře probíhá metodou POST. Po úspěšné validaci formuláře je provedena změna adresy v databázi a uživateli je zobrazena v profilu jeho nová adresa. Stisknutím tlačítka „Zrušit“ dojde k zavření formuláře a překreslení stránky do původního stavu.

Webová evidence	Úvod	Profil	Moje příspěvky	Členové	Volební komise	Volební výsledky	Odhlásit se
-----------------	------	---------------	----------------	---------	----------------	------------------	-------------

Uživatelský profil

Osobní údaje

Jméno:	Jonáš	Příjmení:	Plukovník
Členské číslo:	2568	Typ členství:	Člen předsednictva
Datum narození:	04. May 1987	Emailová adresa:	activeUser@test.com

Vyplňte prosím adresu:

Ulice:
Město:
Okres:
Kraj:
PSČ:

Obrázek 5.9: Zobrazení uživatelského profilu - formulář změny adresy

5.3.5.2 Změna hesla nebo e-mailu

Po stisknutí příslušného tlačítka se uživateli zobrazí formulář pro změnu hesla nebo e-mailu 2.4.5. Z důvodu větší bezpečnosti jsem se rozhodl při těchto změnách vyžadovat po uživateli zadání současného hesla. Validace těchto formulářů probíhá jak na straně klienta, tak na serverové straně. V případě správného vyplnění formuláře volá třída profileController metodu třídy userManager, jež provede příslušné změny.

5.3.6 Zobrazení vlastních příspěvků

V rámci webové aplikace má každý uživatel právo nechat si zobrazit členské příspěvky, jež mu byly v rámci evidence připsány. Na rozdíl od případu užití 2.4.12 jsem se rozhodl pro větší pohodlí uživatele vyčlenit pro členské příspěvky vlastní záložku v navigačním menu. Uživateli nyní stačí pouze kliknout na odkaz „Moje Příspěvky“ v navigačním menu a rovnou se mu zobrazí členské příspěvky. Tato obrazovka je dostupná ze souboru mysubscriptions.php, odtud je pomocí třídy mySubscriptionsController načten mySubscriptionView.php,

5. REALIZACE

jež se stará o zobrazení dat uživateli. Data jsou opět načítána pomocí třídy `mySubscriptionsController`, který volá metodu třídy `memberManager` pro načtení příspěvků uživatele. Data jsou do View předána formou asociativního pole a zde jsou vložena do HTML tabulky. Funkce, jakými je například vyhledávání nebo řazení, jsou tabulce přidány pomocí knihovny `DataTables 4.5`. Vyhledávání, řazení apod. v tabulce je prováděno na straně klienta v prohlížeči, tímto nedochází ke zbytečnému zatěžování serveru a reakce tabulky jsou rychlejší. Toto řešení jsem zvolil z důvodu, že neočekávám takové množství záznamů, jež by zpomalovalo zobrazení dat.

Webová evidence Úvod Profil **Moje příspěvky** Členové Volební komise Volební výsledky Odhlásit se

Moje příspěvky

Zde je seznam všech Vám připsaných příspěvků.

Zobraz záznamů na stránce Vyhledat

Datum	Částka	Popis	Přidal člen č.
2016-05-03	500	Duben	2568
Datum	Částka	Popis	Přidal člen č.

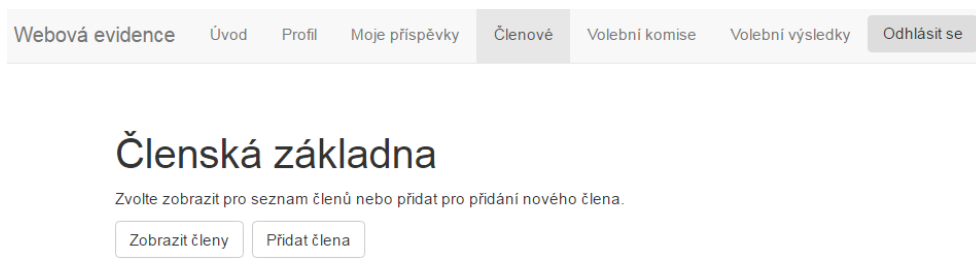
Strana 1 z 1 Předchozí 1 Další

Obrázek 5.10: Obrazovka Moje příspěvky

5.3.7 Členská základna

V této části aplikace má uživatel možnost spravovat evidenci členů. Opět jsem využil oddělení funkcionality od zobrazení, o zobrazení informací se stará `membersView.php`, zpracování uživatelských vstupů a aktualizaci obrazovky se stará třída `membersController`. Zobrazení členů je možné přímým odkazem na soubor `members.php`, odkud se pomocí Controlleru načítá View. Tato část aplikace je dostupná pouze uživatelům s rolí `Pokročilý uživatel 2.5.2`. V případě, že se tuto stránku pokusí zobrazit uživatel s nižším oprávněním, je automaticky přesměrován na úvodní obrazovku.

V základním stavu se na této obrazovce zobrazují pouze dvě tlačítka: „Zobrazit členy“, „Přidat členu“. Toto řešení jsem zvolil z toho důvodu, aby se zbytečně nenačítala data o členech, pokud chce uživatel například pouze přidat nového členu.



Obrázek 5.11: Členská základna - základní zobrazení

5.3.7.1 Přidání nového člena

Po stisknutí tlačítka „Přidat člena“ se uživateli zobrazí formulář pro přidání člena do evidence 2.4.10. Tento formulář je načten ze souboru `addMemberForm.php`. Jednotlivá pole formuláře jsou dostatečně pojmenována, aby uživateli zjednodušila zadávání informací. Položka „Typ členství“ obsahuje možné typy členství načtené přímo z databáze, uživatel si tak nemusí pamatovat jednotlivá členství. Formulář vyžaduje zadání unikátního členského čísla. Číslování členů je vnitřní záležitostí politické strany, a proto jsem se rozhodl negenerovat členská čísla automaticky. Formulář vyžaduje vyplnění všech položek, na straně klienta jsem k validaci formuláře využil HTML5. Na serverové straně kontrolovuji pouze správný formát e-mailu, zadaného data a členského čísla. Vzhledem k tomu, že do této části evidence budou mít přístup pouze prověřeni členové strany, jsem nepředpokládal úmyslné pokusy o uložení špatných dat. Po úspěšném odeslání formuláře je uživateli zobrazena evidence členů v základním stavu navíc se zprávou o výsledku přidání nového člena.

Webová evidence Úvod Profil Moje příspěvky **Členové** Volební komise Volební výsledky Odhlásit se

Členská základna

Zrušit

Jméno:

Příjmení:

Datum narození:

Emailová adresa:

Členské číslo:

Typ členství:

Odeslat

Obrázek 5.12: Členská základna - přidání člena

5.3.7.2 Zobrazení členské základny

Po stisku tlačítka „Zobrazit členy“ se uživateli zobrazí v tabulce členská základna strany 2.4.9. Data jsou do tabulky načtena pomocí třídy `membersController`, jež volá metodu příslušné třídy na business vrstvě - `memberManager`. Samotnou tabulku jsem se rozhodl oddělit do souboru `membersTable.php`. Tabulka je obohacena o funkčnost pomocí knihovny `DataTables 4.5`. Data se načítají ze serveru najednou, operace s daty, jakými jsou například vyhledávání nebo srovnávání výsledků, probíhá až na straně klienta. Pro další práci s jednotlivými členy je potřeba kliknout na příslušný řádek tabulky.

Jak je patrné z obrázku 5.13 lze rovnou z tohoto zobrazení jednoduše změnit typ členství. Po kliknutí na tlačítko „Zobrazit detaily“ se uživateli místo tabulky členů zobrazí informace o vybraném členovi včetně historie jeho členských příspěvků.

Webová evidence Úvod Profil Moje příspěvky **Členové** Volební komise Volební výsledky Odhlásit se

Členská základna

Zvolte zobrazit pro seznam členů nebo přidat pro přidání nového člena.

Zavřít

Zobraz detaily Přidat příspěvek

Vyloučen

Změnit typ členství

Zobraz 10 záznamů na stránce Vyhledat člen

Jméno ↓↑	Příjmení ↓↑	Členské číslo ↓↑	Typ členství ↓↑	Email ↓↑	Datum narození ↓↑
Chandler	Bing	1234	Člen	c.bing@super.com	1974-07-15
Jonáš	Plukovník	2568	Člen předsednictva	activeUser@test.com	1987-05-04
Ross	Geller	1235	Člen	r.geller@super.com	1972-10-05
Jméno	Příjmení	Členské číslo	Typ členství	Email	Datum narození

Strana 1 z 1 (Filtrováno ze 9 záznamů) Předchozí 1 Další

Obrázek 5.13: Členská základna - zobrazení členů

5.3.7.3 Přidání příspěvku členovi

Na rozdíl od případu užití 2.4.11 jsem se rozhodl umožnit uživateli přidat členský příspěvek bez nutnosti otevření členské karty. Myslím si, že toto řešení bude pro uživatele přínosnější a rychlejší, zejména pokud by přidávali příspěvky postupně více členům. Po kliknutí na tlačítko „Přidat příspěvek“ se namísto tabulky zobrazí formulář pro přidání členského příspěvku. Tento formulář je definován v souboru addSubscriptionForm.php. Uživateli se zde zobrazí členské číslo člena, jemuž má být příspěvek zaevidován, uživateli tak stačí vyplnit pouze částku a popis příspěvku. Datum přidání příspěvku do evidence vygeneruje informační systém. Validaci formuláře opět provádím pomocí HTML5, na serverové straně kontroluji vyplnění požadovaných polí a zda-li částka je opravdu číslo. Po odeslání formuláře je uživatel přesměrován zpět na základní zobrazení obrazovky Členové spolu se zprávou o úspěchu operace.

Vyplňte prosím údaje:

Členské číslo:	<input type="text" value="1111"/>
Částka:	<input type="text" value="Částka"/>
Popis:	<input type="text" value="Popis"/>
	<input type="button" value="Odeslat"/>

Obrázek 5.14: Členská základna - přidání příspěvku členovi

5.3.7.4 Zobrazení detailů o členovi

Na této obrazovce si uživatel může nechat zobrazit detailní informace i členovi strany. Jedná se o podobné zobrazení, jako když si člen zobrazí svůj vlastní profil. Ve spodní části obrazovky se nachází tabulka s historií členských příspěvků. Tyto příspěvky jsou načteny najednou serverem. Zpracování formou interaktivní tabulky probíhá až na klientské straně v prohlížeči, opět se jedná o obyčejnou HTML tabulku k níž byly přidány funkce pomocí knihovny DataTables 4.5. Po kliknutí na příslušný řádek tabulky se tento zvýrazní a umožní uživateli smazat příspěvek pomocí tlačítka „Odstranit Příspěvek“. Před smazáním příspěvku na uživatele vyskočí okno vyžadující potvrzení operace.

Členský profil

Jméno:	Jonáš	Příjmení:	Plukovník
Členské číslo:	2568	Typ členství:	Člen předsednictva
Datum narození::	04. May 1987	Emailová adresa:	activeUser@test.com

Adresa:

Ulice:	Ulice č.5		
Město:	Střední město	PSČ:	12345
Okres:	Okres na severu	Kraj:	Můj Kraj

Členské příspěvky

[Odstranit příspěvek](#)

Zobraz záznamů na stránce

Vyhledat

ID	Datum přidání do evidence	Částka	Popis	Přidal člen č.
1	2016-05-03	500	Duben	2568
ID	Datum	Částka	Popis	Přidal člen č.

Strana 1 z 1

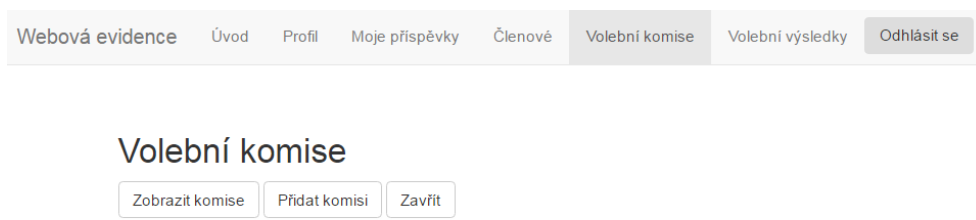
[Předchozí](#) [1](#) [Další](#)

Obrázek 5.15: Členská základna - zobrazení informací o členovi

5.3.8 Volební komise

Tato část aplikace slouží uživateli pro zobrazování a přidávání volebních komisí včetně členů těchto komisí a je dostupná pouze pro uživatele s rolí „Pokročilý uživatel“ 2.5.2. Pokud má uživatel pouze roli běžného uživatele, je automaticky přeměrován na úvodní obrazovku. Tato část aplikace je dostupná z navigačního menu odkazem „Volební komise“ a je načtena ze souboru `committees.php`. Pomocí třídy `committeesController` je načten soubor `committeeView.php`, jež obsahuje vzhled této části webové aplikace.

V základním zobrazení volebních komisí si uživatel může nechat zobrazit evidované volební komise stiskem tlačítka „Zobrazit komise“ nebo přidat novou komisi stiskem tlačítka „Přidat komisi“.



Obrázek 5.16: Volební komise - základní zobrazení

5.3.8.1 Přidání volební komise

Po stisknutí tlačítka „Přidat komisi“ 2.4.14 dojde k překreslení stránky na serveru a zobrazení formuláře pro přidání volební komise. Tento formulář je načten ze souboru `committeeAddForm.php`. Do komise lze přidat libovolný počet členů, stiskem tlačítka „Přidat dalšího člena“ se zobrazí další část formuláře pro přidání dalšího člena do komise. Stejným způsobem lze člena odebrat stiskem tlačítka „Odebrat člena“. Toto zobrazování nových položek ve formuláři se odehrává v prohlížeči uživatele pomocí JQuery 4.4. Formulář nepovažuje zadání jednoho člena strany vícekrát v jedné komisi za chybu, aby bylo možné zaevidovat i různé funkce jednoho člena v komisi. Po odeslání formuláře se zobrazení komisí překreslí zpátky do základního zobrazení a uživatel je zobrazena zpráva o úspěchu přidání komise.

Webová evidence Úvod Profil Moje příspěvky Členové **Volební komise** Volební výsledky Odhlásit se

Volební komise

Zobrazit komise Přidat komisi Zavřít

Popis komise:

Člen:

Popis funkce v komisi

Člen:

Popis funkce v komisi

Odebrat člena

Přidat dalšího člena

Odeslat

Obrázek 5.17: Volební komise - přidání komise

5.3.8.2 Zobrazení volebních komisí

K zobrazení evidovaných volebních komisí dojde po stisku tlačítka „Zobrazit komise“. Třída `committeesController` v tomto případě zajistí načtení tabulek pro zobrazení komisí a jednotlivých členů ze souboru `committeesTable.php`. Uživateli se zobrazí dvě oddělené tabulky, v horní tabulce se zobrazují jednotlivé volební komise a ve druhé tabulce členové všech volebních komisí. Toto řešení jsem zvolil, aby uživatel mohl rychleji filtrovat a vyhledávat požadované informace. Data jsou zaslána uživateli najednou a zpracování dat (filtrování, řazení dle sloupců, vyhledávání) probíhá na straně uživatele v prohlížeči. Knihovna `DataTables` si je schopna poradit i s velkým množstvím dat [19]. Proto jsem se rozhodl data načíst ze serveru najednou, aby při jejich využití klientem nebyl server zbytečně vytěžován.

5. REALIZACE

Webová evidence Úvod Profil Moje příspěvky Členové **Volební komise** Volební výsledky Odhlásit se

Volební komise

Zobrazit komise Přidat komisi Zavřít

Zobraz 10 záznamů na stránce Vyhledat

ID komise	Popis	Počet členů
5	1. sjezd strany	2
ID komise	Popis	Počet členů

Strana 1 z 1 Předchozí 1 Další

Zobraz 10 záznamů na stránce Vyhledat

Členské číslo	Jméno	Příjmení	Funkce v komisi	Popis člena	Číslo komise
1234	Chandler	Bing	Člen	sčítá hlasy	5
1874	Karolína	Světlá	Předseda	řečník	5
Členské číslo	Jméno	Příjmení	Funkce v komisi	Popis člena	Číslo komise

Strana 1 z 1 Předchozí 1 Další

Obrázek 5.18: Volební komise - zobrazení komisí a členů

5.3.8.3 Práce s komisemi

Při kliknutí na volební komisi v tabulce dojde k zvýraznění vybraného řádku. Nad tabulkou volebních komisí se zobrazí možnost smazat volební komisi a všechny členy stiskem tlačítka „Smazat komisi“, případně může uživatel vyplněním drobného formuláře přidat nového člena do komise. K zobrazení tohoto formuláře spolu s tlačítkem „Smazat komisi“ jsem využil JQuery, aby zbytečně nedocházelo k zatěžování serveru požadavky.

Při kliknutí na člena volební komise ve druhé tabulce dojde ke zvýraznění zvoleného řádku a nad touto tabulkou se zobrazí možnost smazat člena volební komise stiskem tlačítka „Odebrat člena“. Před odebráním člena se uživateli zobrazí dialogové okno vyžadující potvrzení operace. Při smazání všech členů komise nedojde k odstranění komise samotné, toto se může hodit v případě, že bylo nutné vyměnit všechny členy komise. Komisi je nutno smazat ručně.

Webová evidence Úvod Profil Moje příspěvky Členové **Volební komise** Volební výsledky Odhlásit se

Volební komise

Zobrazit komise Přidat komisi Zavřít

Smazat komisi

Přidat člena: Jonáš Plukovník, členské číslo: 2568, Člen předsednictva

Jiná

Popis funkce v komisi

Přidat člena

Zobraz 10 záznamů na stránce Vyhledat

ID komise	Popis	Počet členů
5	1. sjezd strany	2
ID komise	Popis	Počet členů

Strana 1 z 1 Předchozí 1 Další

Obrázek 5.19: Volební komise - přidání nového člena

5.3.9 Volební výsledky

Webová aplikace umožňuje zadávat, prohlížet 2.3.1.4 a mazat volební výsledky. Volební výsledky jsou uživateli dostupné z navigačního menu po kliknutí na odkaz „Volební výsledky“. Sekce volební výsledky je dostupná pouze pro uživatele s rolí Pokročilý uživatel 2.5.2. Volební výsledky jsou dostupné na stránce elections.php, o funkcionalitu se stará třída electionsController a o samotné zobrazení informací soubor electionsView.php.

Webová evidence Úvod Profil Moje příspěvky Členové Volební komise **Volební výsledky** Odhlásit se

Volební výsledky

Zobrazit volební výsledky Přidat volební výsledek Zavřít

Obrázek 5.20: Volební výsledky - základní zobrazení

5.3.9.1 Přidání volebního výsledku

Uživatel může přidat volební výsledek stiskem tlačítka „Přidat volební výsledek“, uživateli se následně zobrazí formulář pro přidání volebního výsledku 2.4.7. Po stisku tlačítka třída electionsController zajistí překreslení stránky a zobrazení formuláře. Formulář je uložen v souboru addResultForm.php. Stiskem tlačítka „Zavřít“ dojde k překreslení stránky zpět na základní zobrazení. Formulář vyžaduje vyplnění všech položek, pro počet hlasů jsem zvolil desetinné číslo s přesností na dvě místa, což se mi zdá dostačující. Uživatel dále zadá datum konání voleb, popis volebního výsledku a typ voleb. Možné typy voleb jsou načteny přímo z databáze, takže si uživatel nemusí všechny typy pamatovat.

Validace probíhá na straně uživatele pomocí HTML5 vstupu a po odeslání formuláře i na serverové straně. Na serverové straně nekontroluji pouze zadání povinných údajů, ale také správný formát. Například jestli zadané číslo je opravdu číslem. Pro tuto kontrolu jsem se rozhodl, protože některé prohlížeče nepodporují všechny druhy HTML5 vstupních polí.

Po správné validaci formuláře třída electionsController zavolá metodu třídy electionManager, jež provede požadované změny. Uživatel je přesměrován zpět na základní zobrazení obrazovky volebních výsledků a je informován o úspěchu přidání volebního výsledku.

The screenshot shows a web application interface with a navigation bar at the top containing links: 'Webová evidence', 'Úvod', 'Profil', 'Moje příspěvky', 'Členové', 'Volební komise', 'Volební výsledky', and 'Odhlásit se'. The main content area is titled 'Volební výsledky' and contains three buttons: 'Zobrazit volební výsledky', 'Přidat volební výsledek', and 'Zavřít'. Below these buttons is a form with the following fields:

- Datum konání voleb:** A text input field with the placeholder 'mm/dd/yyyy'.
- Procentuální zisk hlasů:** A text input field with the placeholder 'Zisk hlasů (%)'.
- Popis volebního výsledku:** A text input field with the placeholder 'Popis volebního výsledku'.
- Typ voleb:** A dropdown menu with 'Obecní' selected.

At the bottom of the form is an 'Odeslat' button.

Obrázek 5.21: Volební výsledky - přidání volebního výsledku

5.3.9.2 Zobrazení volebních výsledků

Po kliknutí na tlačítko „Zobrazit volební výsledky“ se uživateli zobrazí evidované volební výsledky 2.4.8. Třída electionsController zavolá metodu třídy electionsManager, jež předá zpátky evidovaná data volebních výsledků. Třída electionsController pak aktualizuje electionsView.php tak, aby zobrazoval tabulku s volebními výsledky. Tato tabulka je definována v souboru electionResultTable.php.

Data jsou načtena serverem naráz a zaslána klientovi, k zobrazení dat jsem použil tabulku obohacenou o funkce pomocí knihovny DataTables 4.5. Po kliknutí na řádek tabulky je tento zviditelněn pomocí JQuery a navíc se uživateli zobrazí tlačítko „Odstranit výsledek“. Po kliknutí na toto tlačítko musí uživatel potvrdit smazání volebního výsledku v dialogovém okně, tímto je zabráněno nechtěnému smazání volebního výsledku.

Webová evidence Úvod Profil Moje příspěvky Členové Volební komise **Volební výsledky** Odhlásit se

Volební výsledky

Zobrazit volební výsledky Přidat volební výsledek Zavřít

Odstranit výsledek

Zobrazit 10 záznamů na stránce Vyhledat

Datum voleb	Počet hlasů(%)	Popis výsledku	Typ voleb	Přidal člen č.
2006-03-01	55.67	Naše 1. volby	Celostátní	2568
2010-03-01	66.01	Naše 2. volby	Celostátní	2568

Datum voleb Počet hlasů(%) Popis výsledku Typ voleb Přidal člen č.

Strana 1 z 1 Předchozí 1 Další

Obrázek 5.22: Volební výsledky - zobrazení volebních výsledků

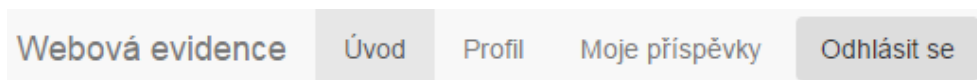
5.3.10 Odhlášení uživatele

Odhlášení uživatele je možné odkudkoliv z webové aplikace stisknutím tlačítka „Odhlásit se“. Po odhlášení uživatele dojde ke smazání dat uložených do proměnné \$_SESSION zavoláním PHP funkce session_destroy(). Uživatel je poté přeměrován na přihlašovací obrazovku v souladu s 2.4.4.

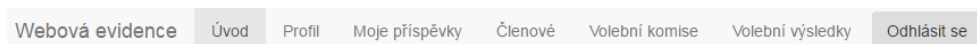
Dále jsem v aplikaci zavedl automatické odhlášení uživatele při delší nečinnosti. Dobu potřebnou k odhlášení jsem stanovil na 30 minut. Automatické odhlášení jsem implementoval z důvodu vyššího zabezpečení aplikace.

5.3.11 Navigační menu

Součástí webové aplikace je navigační menu. Zobrazení položek v tomto menu se liší podle role uživatele. Pokud nemá uživatel dostatečná oprávnění pro využívání některých částí webové aplikace, nejsou mu zobrazeny ani příslušné položky v navigačním menu. Odhlašovací tlačítko je dostupné všem uživatelům.



Obrázek 5.23: Navigační menu - zobrazení pro běžného uživatele



Obrázek 5.24: Navigační menu - zobrazení pro pokročilého uživatele

5.3.12 Připojení přes HTTPS

Webová aplikace vyžaduje po uživateli připojení pomocí šifrovaného protokolu HTTPS. Pokud se uživatel připojí pouze standardním protokolem HTTP je automaticky přesměrován na připojení zajištěné protoklem HTTPS. Toto přesměrování jsem realizoval za pomoci podmínky a pravidla v souboru .htaccess v kořenovém adresáři aplikace. Uživatel je přesměrován HTTP stavovým kódem 301 [20].

5.3.13 Uživatelská hesla

Pro ukládání uživatelských hesel do databáze jsem použil knihovní funkci PHP `password_hash()`. Tato funkce generuje hashe pro uložení hesel do databáze, k vytvoření těchto hashí používá náhodně generovanou sůl. Tímto je zajištěno, že uživatelé se stejným heslem nebudou mít v databázi stejné hashe tohoto hesla. K ověření hesla při autentifikaci uživatele se pak používá funkce `password_verify()`. [21]

5.3.14 Generování aktivačních kódů

V aplikaci dochází ke generování dvou aktivačních kódů: jeden pro potřeby aktivace uživatelského účtu a druhý v případě obnovení hesla. Tyto kódy jsou pak uživateli zaslány e-mailem a pomocí nich může provést potřebnou operaci. Uživatel nicméně kód ručně zadávat nemusí, na to je myšleno předem už při

vygenerování celého aktivačního odkazu, jehož je aktivační kód součástí. Řetězec pro vytvoření hashe jsem získal náhodně generovanou hodnotou z funkce `mt_rand()`, dále se tato hodnota zřetězí s uživatelským e-mailem a časem. Takto vzniklý řetězec je nakonec zpracován hashovací funkcí SHA1. Čas a e-mail jsem přidal z důvodu snížení pravděpodobnosti vzniklých kolizí.

5.4 Testování

Spolu s implementací aplikace jsem prováděl testování jednotlivých přidaných funkcí. Tímto způsobem pro mne bylo jednodušší odhalit vzniklé problémy a rychleji nalézt vhodné řešení. Jednotlivé části aplikace jsem testoval na různé vstupy, které by odpovídali reálnému provozu aplikace. Dále jsem zkoušel reakce na neočekávané a nevhodné vstupy. V rámci testů jsem také kontroloval správnost zápisu dat do databáze. I když nemá smysl webovou aplikaci používat v případě nedostupné databáze, otestoval jsem reakce aplikace na výpadky databáze.

Zabezpečení aplikace jsem testoval pokusy o připojení pomocí HTTP místo HTTPS, dále jsem zkoušel přístup do jednotlivých částí aplikace pomocí přímých odkazů místo navigace umožněné přímo aplikací. Tímto způsobem jsem otestoval možnosti neautorizovaného přístupu do webové aplikace.

Nakonec jsem celou webovou aplikaci otestoval jako celek. Při tomto testování jsem využil případy užití, které jsem definoval v analýze informačního systému.

Ekonomicko manažerské zhodnocení

6.1 Zhodnocení nákladů

6.1.1 Náklady na systém

Nyní se budu zabývat náklady, které plynou z pořízení informačního systému a jeho následného provozu. Do nákladu na pořízení systému jsem započítal náklady na analýzu, návrh, implementaci a testování. Tyto náklady budou vyjádřeny počtem odpracovaných hodin - tzv. man hours. Dále je potřeba do nákladů započítat potřebné licence, certifikáty apod.

Původně jsem předpokládal pracnost okolo 130 pracovních hodin, nicméně tento odhad nebyl zcela přesný a celková pracnost aplikace se vyšplhala na 150 pracovních hodin. Jednu pracovní hodinu jsem nacenil částkou 150 Kč. Z toho plyne celková cena informačního systému a to 22 500 Kč.

6.1.2 Náklady na licence

Velkou výhodou se ukázal být požadavek na implementaci v rámci systému LAMPS, čímž se snížily náklady na potřebné licence. Celý informační systém tak využívá volně přístupného systému, čímž odpadá nutnost jednorázového případně periodického poplatku za licence. Informační systém vznikl v rámci bakalářské práce, a proto jej kterýkoliv subjekt může využívat zdarma. Pro získání výhradních práv na použití systému by bylo potřeba složit poplatky ve výši 7000 Kč [22]. Tohoto práva však nebylo využito.

6.1.3 Náklady na provoz

Politická strana již vlastní server, kde provozuje webové stránky. Vzhledem k předpokládanému počtu uživatelů webové aplikace, není nutno navyšovat

kapacitu tohoto serveru. Používáním webové evidence se cena za provoz nezvýší. Politická strana také vlastní potřebné certifikáty, které bude nadále využívat, jak pro webové stránky, tak pro webovou evidenci. Ani zde nedojde ke zvýšení nákladů na provoz. Správa webové evidence bude probíhat v režii politické strany, pracnost odhaduji na 3 hodiny měsíčně, což může zastat současný administrátor webových stránek. Jak je patrné, se zavedením webové evidence nelze očekávat velké zvýšení nákladů na provoz informačních technologií politické strany.

6.2 Přínosy informačního systému

Politická strana používá pro evidenci své činnosti tabulky v MS Excelu. Toto řešení způsobuje problémy se sdílením těchto dat a také zvyšují nároky na údržbu a celkovou správu. Informační systém podpoří činnost politické strany tím, že zjednoduší a posílí procesy evidence její činnosti. Data z evidence jsou nově dostupná všem členům z jednoho místa a jsou automaticky aktualizována. Informační systém umožňuje pracovat s evidovanými daty více členům najednou aniž by museli řešit problémy spojené s posíláním si dat mezi sebou, toto za ně řeší nový informační systém.

Díky tomuto zjednodušení činnosti se zlepší pracovní podmínky členů pověřených evidováním dat a dojde ke zjednodušení těchto podnikových procesů. Dalším přínosem informačního systému je vyšší úroveň zabezpečení zpracovávaných dat, která už nemusí být uložena v počítačích jednotlivých členů, ale na jednom místě v databázi. Informační systém také umožňuje snadnou manipulaci s daty samotnými, vyhledávání a práce s daty je přímo součástí systému. Uživatelé se tak nemusí učit funkce v MS Excelu, když za ně manipulaci s daty vyřeší samotný informační systém.

Díky zvoleným technologiím bylo možné udržet minimální náklady a zároveň zajistit informační systém s takovou funkčností, jež politické straně chyběla. Budoucí podpora systému by neměla být složitá vzhledem díky využití stále se vyvíjejících technologií, které jsou volně dostupné.

Závěr

V rámci bakalářské práce jsem vytvořil informační systém, jež nalezne využití při evidenci činnosti politické strany. Tento informační systém jsem vytvořil dle požadavků politické strany a umožňuje evidenci členů politické strany, členských příspěvků, volebních komisí a volebních výsledků.

Proces tvorby informačního systému jsem rozdělil na několik částí, jež popisují v této bakalářské práci. Provedl jsem analýzu problémové domény, definoval jsem funkční požadavky, nefunkční požadavky a případy užití informačního systému. Dále jsem popsal vhodné uživatelské role. Na základě této analýzy jsem provedl návrh informačního systému, zvolil jsem vhodnou architekturu a provedl návrh uživatelského rozhraní. Následně jsem provedl samotnou implementaci a otestování aplikace. Součástí bakalářské práce je shrnutí přínosů informačního systému pro politickou stranu a vyčíslení nákladů na pořízení systému a jeho budoucí používání a údržbu.

Myslím si, že by bylo možné v budoucnu rozšířit funkcionalitu informačního systému pro podporu další činnosti politické strany. Například by bylo možné využít uložená data pro vytváření kandidátních listin nebo systém využít pro automatické podávání členských přihlášek do strany.

Literatura

- [1] Pirátská strana: Návrh na vytvoření aplikace pro evidenci členů. [online], [cit. 2016-10-03]. Dostupné z: <https://forum.pirati.cf/administrativni-odbor-f182/navrh-na-vytvoreni-aplikace-pro-evidenci-clenu-t4773.html>
- [2] Ministerstvo vnitra ČR: Občanské aktivity. [online], [cit. 2016-11-03]. Dostupné z: <http://www.mvcr.cz/clanek/obcanske-aktivity-118893.aspx?q=Y2hudW09Ng%3D%3D>
- [3] Čápka, D.: Use Case Diagram. [online], [cit. 2016-27-03]. Dostupné z: <https://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>
- [4] Rejnková, P.: Příklady použití diagramů UML 2.0. [online], [cit. 2016-11-03]. Dostupné z: http://uml.czweb.org/objekto_diagram.htm
- [5] Clever and Smart: Vícevrstvá architektura: popis vrstev. [online], [cit. 2016-27-3]. Dostupné z: <http://www.cleverandsmart.cz/vicestva-architektura-popis-vrstev/>
- [6] Bernard, B.: Úvod do architektury MVC. [online], [cit. 2016-11-03]. Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/uml/uml-use-case-diagram/>
- [7] Shannon, R.: What is HTML? [online], [cit. 2016-29-04]. Dostupné z: <http://www.yourhtmlsource.com/starthere/whatishtml.html/>
- [8] HTML.net: What is CSS? [online], [cit. 2016-29-04]. Dostupné z: <http://html.net/tutorials/css/lesson1.php>
- [9] Stohwasser, P.: What is CSS? [online], [cit. 2016-29-04]. Dostupné z: <http://www.pestujemeweb.cz/obsah/css/co-je-css.php>
- [10] Bootstrap Core Team: Getting started. [online], [cit. 2016-29-04]. Dostupné z: <http://getbootstrap.com/getting-started/>

- [11] Jadrný, T.: Úvodní článek. [online], [cit. 2016-29-04]. Dostupné z: <http://jquery-navod.cz/kategorie-ostatni-clanky/1-uvodni-clanek>
- [12] SpryMedia Ltd: Manual. [online], [cit. 2016-29-04]. Dostupné z: <https://www.datatables.net/manual/installation>
- [13] Čápka, D.: Úvod do MySQL a příprava prostředí. [online], [cit. 2016-29-04]. Dostupné z: <http://www.itnetwork.cz/mysql/mysql-tutorial-uvod-a-priprava-prostredi>
- [14] Zajíc, P.: Úvod do MySQL a příprava prostředí. [online], [cit. 2016-29-04]]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=731
- [15] Brožek, E.: Server Apache, co je vlastně zač a co dokáže. [online], [cit. 2016-29-04]. Dostupné z: <http://www.zive.sk/clanok/3188/server-apache-co-je-vlastne-zac-a-co-dokaze>
- [16] Večera, Z.: PHPMailer: Snadné posílání (HTML) e-mailů z PHP. [online], [2016-30-04]. Dostupné z: <http://www.zive.sk/clanok/3188/server-apache-co-je-vlastne-zac-a-co-dokaze>
- [17] MySQL: Creating InnoDB Tables. [online], [cit. 2016-30-04]. Dostupné z: <http://dev.mysql.com/doc/refman/5.7/en/using-innodb-tables.html>
- [18] Janovský, D.: Soubor .htaccess. [online], [cit. 2016-01-05]. Dostupné z: <http://www.jakpsatweb.cz/server/htaccess.html>
- [19] SpryMedia Ltd: Client-side data source (50,000 rows). [online], [cit. 2016-02-05]. Dostupné z: https://datatables.net/extensions/scroller/examples/initialisation/large_js_source.html
- [20] CheckUpDown: HTTP Error 301 - Moved permanently. [online], [cit. 2016-02-05]. Dostupné z: <http://www.checkupdown.com/status/E301.html>
- [21] The PHP Group: password_hash. [online], [cit. 2016-02-05]. Dostupné z: <http://php.net/manual/en/function.password-verify.php>
- [22] Valenta, M.: Bakalářská práce (BP) a předmět BI-PPR. [online], [cit. 2016-05-05]. Dostupné z: <https://users.fit.cvut.cz/valenta/doku/doku.php/fitzp/start>

Seznam použitých zkratk

- CSS** Cascading Style Sheets
- DAO** Data access Object
- HTML** HyperText Markup language
- HTTP** Hypertext Transport Protocol
- HTTPS** Hypertext Transfer Protocol Secure
- LAMPS** Linux, Apache, MySQL, PHP, SSL
- MVC** Model View Controller
- PHP** PHP: Hypertext Preprocessor
- PHP PDO** PHP Data Objects
- SHA1** Secure Hash Algorithm 1
- SQL** Structured Query Language
- SSL** Secure Sockets Layer

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
install.txt.....	Popis instalace
webApp.....	adresář se spustitelnou formou implementace
sql.....	adresář s sql scripty
├─ create.sql.....	script pro vytvoření databáze
src	
├─ thesis.....	zdrojová forma práce ve formátu \LaTeX
text.....	text práce
├─ thesis.pdf.....	text práce ve formátu PDF
UML.....	Soubory programu Enterprise Architect z nichž jsem vytvořil
diagramy	
webApp	
├─ Bakalarka.zip.....	Archív s implementací bakalářské práce