



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**SPRÁVA MUNICIPALIT V SYSTÉMU HLASOVÁNÍ
MĚSTSKÝCH ZASTUPITELSTEV**

MUNICIPAL ADMINISTRATION IN THE VOTING SYSTEM OF CITY COUNCILS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ONDŘEJ KREJČÍ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. KRISTÝNA ZAKLOVÁ

BRNO 2024

Zadání diplomové práce



154362

Ústav: Ústav informačních systémů (UIFS)
Student: **Krejčí Ondřej, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Informační systémy a databáze
Název: **Správa municipalit v systému hlasování městských zastupitelstev**
Kategorie: Informační systémy
Akademický rok: 2023/24

Zadání:

1. Seznamte se s problematikou fungování obecních a krajských zastupitelstev a s daty, které ze své činnosti produkují. Zaměřte se na dostupné datové zdroje a systémy na zaznamenávání hlasování.
2. Prostudujte oblast tvorby informačních systémů a webových aplikací s důrazem na použitelnost uživatelských rozhraní.
3. Analyzujte požadavky na ruční vkládání informací o hlasování zastupitelstev a následnou správu jednotlivých municipalit. Analyzujte existující systém pro vizualizaci hlasování Zastupitelstva města Brna.
4. Navrhněte architekturu pro správu instancí informačního systému z bodu 3 a jejich databází. Dále navrhněte informační systém pro správu municipalit (jejich vytváření, úpravy, uživatelsky přívětivé vkládání, aktualizace záznamů z hlasování těchto orgánů a správa uživatelů).
5. Navržené řešení po konzultaci s vedoucí implementujte.
6. Otestujte funkčnost a použitelnost systému.

Literatura:

- Pereira, G.V., Macadar, M.A., Luciano, E.M. et al. *Delivering public value through open government data initiatives in a Smart City context*. Inf Syst Front 19, 213–229 (2017). DOI: <https://doi.org/10.1007/s10796-016-9673-7>
- Johnson, J. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Morgan Kaufmann Publishers/Elsevier, 2010, ISBN 9780123750303.
- Karvonen, K. (2000). *The beauty of simplicity*. In Proceedings on the 2000 conference on Universal Usability (CUU '00), pages 85–90. ACM.
- Kruger, R, Brosens, J. a Hattingh, M. (2020). *A Methodology to Compare the Usability of Information Systems*. DOI: https://doi.org/110.1007/978-3-030-45002-1_39.
- Zaklová, K. (2023). Analýza a vizualizace dat z hlasování Zastupitelstva města Brna. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/146480>

Při obhajobě semestrální části projektu je požadováno:
Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Zaklová Kristýna, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 31.7.2024
Datum schválení: 30.10.2023

Abstrakt

Cílem práce bylo rozšířit existující systém pro vizualizaci hlasování Zastupitelstva města Brna, umožnit jeho použití pro více municipalit a přidat možnost správy municipalit a informací o hlasování jejich zastupitelstev. Původní systém, implementovaný jako webová aplikace s architekturou klient–server za použití frameworku Flask a knihovny React, byl zobecněn pro vizualizaci hlasování zastupitelstev libovolného počtu municipalit. Pro správu dat byla do systému přidána nová administrační část. Funkčnost řešení byla demonstrována na datech několika municipalit. Správná funkčnost administrační části byla ověřována průběžně při vkládání dat nových municipalit, a také bylo provedeno testování s několika vybranými uživateli.

Abstract

The goal of this thesis was to extend an existing system for visualization of Brno City Council voting, enable its usage for multiple municipalities and add the option to manage municipalities and their council voting information. The original system, implemented as a web application with a client–server architecture using the Flask framework and React library, was generalized to enable visualizing data from multiple municipalities. New administration part was added to the system for data management. Functionality of the solution was demonstrated on data from several municipalities. Correct functionality of the administration part was continuously verified when entering data for new municipalities and testing was also carried out with a few selected users.

Klíčová slova

zastupitelstvo, hlasování, hlasování zastupitelstva, otevřená data, informační systémy, React, Flask

Keywords

city council, voting, council voting, open data, information systems, React, Flask

Citace

KREJČÍ, Ondřej. *Správa municipalit v systému hlasování městských zastupitelstev*. Brno, 2024. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Kristýna Zaklová

Správa municipalit v systému hlasování městských zastupitelstev

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Kristýny Zaklové. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Ondřej Krejčí

30. července 2024

Poděkování

Rád bych poděkoval Ing. Kristýně Zaklové za odborné vedení práce, poskytnuté rady a zpětnou vazbu a také celkovou vstřícnost. Dále děkuji Ing. Petru Johnovi za pomoc při nasazování jednotlivých verzí systému, a rovněž za poskytnuté rady a zpětnou vazbu.

Obsah

1	Úvod	3
2	Obecní a krajská zastupitelstva	4
2.1	Územní samosprávné celky	5
2.1.1	Obce a orgány obcí	5
2.1.2	Kraje a orgány krajů	8
2.2	Data zastupitelstev a jejich zdroje	9
2.3	Systémy na zaznamenávání hlasování	11
3	Průzkum dostupných dat z hlasování zastupitelstev	16
3.1	Česká republika	16
3.2	Zahraničí	19
3.2.1	Datové sady	20
3.2.2	Municipality	21
3.2.3	Projekt Parlameter	24
4	Informační systémy a použitelnost uživatelských rozhraní	26
4.1	Informační systémy	26
4.2	Webové aplikace	27
4.3	Uživatelská rozhraní a jejich použitelnost	30
5	Analýza problému a specifikace požadavků	34
5.1	Systém pro vizualizaci hlasování Zastupitelstva města Brna	34
5.2	Požadavky na rozšíření systému a cíloví uživatelé	41
5.3	Specifikace požadavků	42
6	Návrh	44
6.1	Architektura systému	44
6.2	Serverová část systému	45
6.2.1	Datový model	46
6.3	Uživatelské rozhraní	50
6.3.1	Administrační část	50
7	Implementace	57
7.1	Serverová část systému	57
7.1.1	Generalizace pro více municipalit	57
7.1.2	Administrační část	58
7.1.3	Změny schématu databáze municipalit	63

7.1.4	Další rozšíření	64
7.2	Uživatelské rozhraní	65
7.2.1	Administrační část	65
8	Testování	70
8.1	Testy API	70
8.2	Uživatelské testování	72
9	Závěr	75
	Literatura	76
A	Koncové body REST API	81
B	Schéma centrální databáze	86
C	Snímky obrazovky uživatelského rozhraní	87
D	Obsah příloženého paměťového média	89

Kapitola 1

Úvod

Transparentnost veřejné správy společně s důrazem na otevřená data nabývá v posledních dekáдах s rozvojem digitalizace na významu. Technologický pokrok usnadňuje zveřejňování dat a umožňuje rychlý a snadný přístup k nim pro případné další využití. Publikování otevřených dat je prosazováno mnoha nevládními, ale i státními organizacemi, a má napomoci informovanosti společnosti a inovacím. V kontextu vládních dat také umožňuje získat přehled nad činností volených zástupců a jejich případnou kontrolu.

Zapojení do zveřejňování dat roste i na úrovni menších územních celků než je stát. Jako příklad lze uvést město Brno, které mimo jiné publikuje data z jednání svého zastupitelstva. Na základě těchto dat byl v rámci předchozí diplomové práce vytvořen systém pro vizualizaci hlasování Zastupitelstva města Brna.

Cílem této práce je existující systém rozšířit o práci s více municipalitami a správu dat. Je třeba umožnit ukládání údajů o samotných municipalitách a upravit systém tak, aby jej bylo možné použít pro jednotnou vizualizaci hlasování zastupitelstev více municipalit. Dále je třeba systém rozšířit o možnost správy informací o hlasování zastupitelstev, které systém vizualizuje, a také municipalit a uživatelů systému.

Kapitola 2 představuje zastupitelstva obcí a krajů v České republice a poskytuje obecný pohled na data, která produkují. Navazující kapitola 3 obsahuje průzkum dostupných dat z hlasování zastupitelstev municipalit v České republice a v zahraničí. Průzkum se zabývá dostupnými datovými sadami a analýzou dat z jednání zastupitelstev u vybraných měst. Kapitola 4 představuje oblast informačních systémů, webových aplikací a použitelnosti uživatelských rozhraní.

Součástí kapitoly 5 je analýza existujícího systému pro vizualizaci hlasování Zastupitelstva města Brna a používaného datového modelu. Následně jsou definováni uživatelé a analyzovány požadavky na rozšíření systému. Návrh potřebných úprav existujícího systému a nové části systému pro správu dat je popsán v kapitole 6. Následující kapitola 7 popisuje konkrétní implementaci navržených úprav a rozšíření systému. Testování správné funkčnosti vytvořeného systému a ověření jeho použitelnosti uživatelským testováním je popsáno v kapitole 8. Závěrečná kapitola 9 shrnuje provedenou práci a představuje některé možnosti dalšího rozšiřování systému.

Kapitola 2

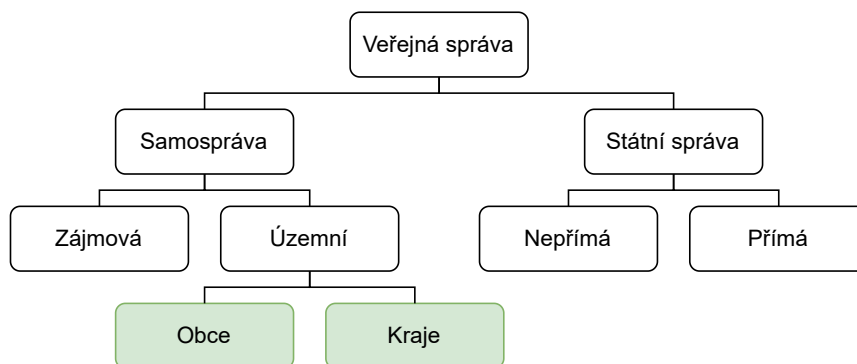
Obecní a krajská zastupitelstva

Zastupitelstva jsou volenými orgány územních celků, které jsou zodpovědné za jejich správu. V České republice existují zastupitelstva na dvou úrovních, a to obecní a krajská. V rámci této kapitoly jsou popsány obce a kraje v České republice, včetně zasazení do kontextu veřejné správy a relevantní legislativy. Součástí popisu je definice zastupitelstev jakožto orgánů těchto celků. Dále se kapitola zabývá daty produkovanými nebo souvisejícími se zastupitelstvy. Důležitá pro tuto práci jsou především hlasování zastupitelstev, jsou proto rozebrány i možnosti zaznamenávání hlasování a používané hlasovací systémy.

Veřejnou správou se rozumí správa veřejných záležitostí ve veřejném zájmu, která je vykonávána subjekty jako jejich právem uložená povinnost [4, 42, 44]. Jako opak veřejné správy je chápána soukromá správa, která je vykonávána soukromými subjekty na základě jejich vlastní vůle za účelem dosažení vlastních cílů. Pojem veřejná správa může být chápán v různých pojetích. Práce [44] uvádí organizační pojetí, kdy se pojmem myslí orgány veřejné správy, nebo funkční pojetí, kdy tento pojem reprezentuje činnost těchto orgánů, tedy vykonávání veřejné správy. Veřejná správa může být členěna na státní správu a samosprávu [4, 9]. Zjednodušené schéma veřejné správy obsahující popisované pojmy lze vidět v diagramu 2.1.

Státní správou se rozumí veřejná správa vykonávaná státem skrze jeho instituce [4]. Státní správa může být vykonávána přímo, skrze orgány státu, nebo nepřímo, kdy je správa delegována na jiné subjekty než je stát [44]. Přími vykonavatelé státní správy jsou např. vláda či ministerstva, ústřední státní úřady (např. Úřad pro ochranu hospodářské soutěže) nebo úřady podřízené ministerstvům (např. Česká obchodní inspekce). Nepřími vykonavatelé státní správy jsou orgány samosprávy, na které byla dle zákona delegována pravomoc výkonu státní správy. Zákonem je definováno, které územní celky mohou vykonávat státní správu a v jakém rozsahu [44]. Příkladem je přenesená působnost vykonávaná orgány územních samosprávných celků, např. obecními či krajskými úřady.

Samosprávu lze definovat jako formu veřejné správy, která je vykonávána orgány odlišnými od státu [4, 9, 23]. Jedná se o projev decentralizace, kdy je určitá oblast veřejné správy přenesena na samosprávný orgán, který ji následně vykonává samostatně. Nejedná se zde o vztah podřízenosti či zastupování státu. Souvisí s konceptem subsidiarity, kdy by mělo rozhodování a zodpovědnost náležet orgánu, který je nejbližší občanovi [9]. Samosprávu lze dále dělit na územní a zájmovou [4, 42]. Zájmová samospráva je reprezentována profesními komorami (např. Česká lékařská komora). **Územní samosprávu** lze popsat jako společenství občanů na určitém území, kteří mají právo na samosprávu [4, 44]. V českém právním systému jsou obce a kraje definovány jako územní samosprávné celky [44].



Obrázek 2.1: Zjednodušené schéma veřejné správy v České republice. Zvýrazněné jsou obce a kraje, kterými se tato práce především zabývá.

2.1 Územní samosprávné celky

Základy územní samosprávy jsou položeny přímo v Ústavě, konkrétně v hlavě sedmé – Územní samospráva [10]. Čl. 99 definuje dělení na územní samosprávné celky: „Česká republika se člení na obce, které jsou základními územními samosprávnými celky, a kraje, které jsou vyššími územními samosprávnými celky.“ Mezi další významné definice patří, že obec je vždy součástí kraje, územní samosprávné celky mohou mít vlastní majetek, hospodaří podle vlastního rozpočtu a jsou spravovány zastupitelstvem. Čl. 102 stanovuje, že členové zastupitelstva jsou voleni a mají čtyřleté funkční období (detaily specifikovány v zákoně, viz dále).

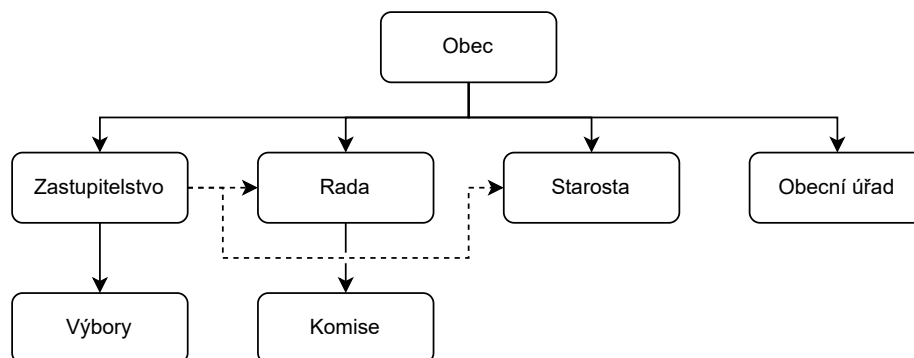
Pro úplnost je třeba ještě zmínit vojenské újezdy, které jsou odlišné od obcí a nejsou součástí území žádné obce [4]. K 1. 1. 2023 existovaly pouze 4 vojenské újezdy a 6258 obcí [18]. Nad úroveň krajů navíc existují v rámci České republiky jako vyšší územní celky ještě regiony soudržnosti, nejedná se však o územní správní celky [4]. Tyto územní celky jsou zařazeny v evropském systému NUTS (*Nomenclature of Units for Territorial Statistics*) na úrovni NUTS 2. Každý region soudržnosti je tvořen jedním nebo více kraji.

Fungování územních samosprávných celků je dále vymezeno zákony. Obcemi se zabývá zákon č. 128/2000 Sb. (Zákon o obcích) [13] a kraji zákon č. 129/2000 Sb. (Zákon o krajích) [14]. V následujících podsekcích je shrnuta relevantní část legislativy, především týkající se zastupitelstev.

2.1.1 Obce a orgány obcí

Primárním legislativním zdrojem pro fungování obcí je Zákon o obcích [13], odkazy v této podsekcí směřují do části 1 tohoto zákona, není-li uvedeno jinak. Základní informace ohledně obcí obsahuje hlava I. Existují různé druhy obcí, resp. základní územní samosprávné celky mohou nést i jiný název než obec. První možností je městys, kterým se obec může stát na vlastní žádost, pokud je tato schválena vládou. Dalším druhem je město, kde je postup podobný, ale existuje zde podmínka, že obec musí mít minimálně 3000 obyvatel. Obce se mohou dále stát statutárními městy, jejichž existence je dána zákonnou úpravou a jejich seznam je obsažen v zákonu. Znění zákona z 1. 7. 2023 [13] jich obsahuje celkem 26. Statutární města mají právo dále členit své území na městské obvody nebo části, které mají vlastní orgány samosprávy.

Základním orgánem obce je jeho zastupitelstvo, dalšími orgány jsou rada, starosta, obecní úřad a zvláštní orgány obce. U jiných druhů obcí existují obdobné orgány, ale liší se názvem, např. městský úřad namísto obecní úřad v případě města nebo primátor namísto starosta v případě statutárního města. Zastupitelstvo a další jmenované orgány existují i pro městské obvody nebo části. Orgány obce se blíže zabývá hlava IV a V. Zjednodušené schéma orgánů obce je k vidění v diagramu 2.2.



Obrázek 2.2: Zjednodušené schéma orgánů obce a jejich orgánů. Přerušované čáry značí členství zastupitelů v dalších orgánech.

Občanem obce je dle § 16 fyzická osoba s českým občanstvím, která má v obci přihlášený trvalý pobyt. Občan obce má právo volit a být volen do zastupitelstva obce. Dále má mj. právo nahlížet do rozpočtu obce, závěrečného účtu obce, usnesení a zápisů z jednání zastupitelstva, usnesení rady, výborů a komisí rady a pořizovat z těchto dokumentů výpisy.

Zastupitelstvo

Zastupitelstvu obce se věnuje díl 1 a 2 hlavy IV. Dle § 68 má zastupitelstvo 5 až 55 členů. Počet zastupitelů odpovídá počtu obyvatel obce, pro různý počet obyvatel je v zákoně definováno možné rozmezí počtu zastupitelů, např. obce do 500 obyvatel mohou mít 5 až 15 zastupitelů. Konkrétní počet zastupitelů obce, kteří mají být zvoleni, je před volbami stanoven zastupitelstvem a musí být oznámen na úřední desce.

Pravomoci zastupitelstva vymezuje především § 84. Patří mezi ně mj. schvalovat program rozvoje a rozpočet obce, zřizovat a rušit příspěvkové organizace a organizační složky obce, vydávat obecně závazné vyhlášky, zřizovat a rušit výbory a volit jejich členy, volit starostu, místostarosty a členy rady obce (z řad zastupitelů obce). Další pravomoci týkající se právního jednání a nakládání s majetkem jsou vymezeny v § 85.

Zasedáním zastupitelstva se zabývají § 92–96. Zasedání se konají dle potřeby, musí se ale konat minimálně každé 3 měsíce. Zpravidla je svolává a řídí starosta. Právo dávat návrhy k zařazení na pořad jednání mají zastupitelé, rada a výbory. Zasedání zastupitelstva obce jsou veřejná. Obecní úřad má povinnost informovat o místě, době a navrženém programu zasedání a tyto informace vyvěsit na úřední desce.

Dle § 95 se z jednání zastupitelstva pořizuje zápis. Zápis musí obsahovat počet přítomných zastupitelů, schválený pořad jednání, průběh a výsledek hlasování a přijatá usnesení. Podepisuje ho starosta nebo místostarosta a určení ověřovatelé. Musí být pořízen do 10 dní po skončení zasedání a uložen na obecním úřadu, kde je možné do něj nahlédnout. Obec dle § 96 vydává jednací řád, kterým se řídí jednání zastupitelstva. Upravuje např. pravidla pro zapojení občanů obce do jednání [44] nebo pravidla pro distanční účast na jednání [26].

Volby do zastupitelstev obcí řeší zákon č. 491/2001 Sb. [17]. Paragraf 22 uvádí požadavky na kandidátní listiny, které musí mj. obsahovat název volební strany a informace o kandidátech: jméno a příjmení, věk, povolání, obec či část obce, kde jsou přihlášení k trvalému pobytu, a název politické strany nebo hnutí, kterého jsou členy. Po registraci kandidátních listin jsou dle § 25 vytištěny hlasovací lístky obsahující tyto údaje, které jsou dodány voličům. Výsledek voleb zveřejní odpovědný obecní úřad na své úřední desce (§ 47), celkový výsledek voleb je zveřejněn ve Sbírce zákonů (§ 51).

Rada

Rada obce je výkonný orgán obce, který se zodpovídá zastupitelstvu. Zabývá se jí díl 3 hlavy IV. Radu obce tvoří starosta, místostarostové a další členové rady zvolení z řad zastupitelů. Počet členů rady je lichý a rada může mít 5 až 11 členů. Rada se nevolí v obcích, které mají méně než 15 zastupitelů. V případě, že rada neexistuje, přechází její pravomoci na starostu. Schůze rady se konají dle potřeby a jsou neveřejné. Ze schůze se pořizuje zápis, požadovaný obsah je podobný jako v případě zastupitelstva. Právo do něj nahlížet mají pouze zastupitelé.

Výbory a komise

Dalšími orgány obce se zabývá hlava V. Jedná se o výbory a komise, kterých může být zřízeno více. Je na uvážení vedení obce, které z těchto orgánů zřídí, a zda budou preferovat výbory či komise [44].

Výbory jsou iniciativní a kontrolní orgány zřizované a odpovědné zastupitelstvu obce. Vždy je zřízen alespoň finanční a kontrolní výbor. Komise jsou iniciativní a poradní orgány rady obce. Na rozdíl od rady nemusí být členy těchto orgánů pouze zastupitelé, v případě komisí je navíc zákonem daná podmínka, že předseda musí prokázat odbornou způsobilost v oblasti působnosti komise.

Statutární město

Statutární město je specifickým typem obce, kterým se dále zabývá hlava VIII. Především je zde definováno právo statutárního města upravit své vnitřní poměry pomocí statutu, kde může vymezit městské obvody nebo části a jejich území. Na své městské obvody nebo části poté může přenést některé pravomoci a svěřit jim majetek. Funkci místostarostů statutárního města zastávají náměstci primátora.

Statutárním městem se může změnou zákona stát libovolné město, jak ale uvádí [44], typicky se jedná o města s velkým počtem obyvatel, která jsou ekonomicky, nebo i kulturně významná. Možnost rozdělení statutárního města na menší celky také podporuje princip subsidiarity.

Hlavní město Praha

Postavení hlavního města Prahy je na rozdíl od ostatních obcí specifické v tom, že má vlastní právní úpravu. Zabývá se jím zákon č. 131/2000 Sb. [16] (Zákon o hlavním městě Praze). Orgány hlavního města Prahy jsou ale v zásadě identické jako u ostatních obcí [44].

Významný je rozdíl v počtu členů zastupitelstva, který je pro hlavní město definovaný v § 48 a činí 55 až 70 členů. Hlavní město Praha je zároveň obcí i krajem a primátor dle § 72 vykonává i funkci hejtmána. Volby se podle § 123 řídí zákonem pro volby do obecních

zastupitelstev. Zákon také přímo definuje, že je Praha rozdělena na městské části, které jsou dále vymezeny ve Statutu hlavního města Prahy, podobně jako je tomu u statutárních měst. Samotné hlavní město Praha ale dle zákona není statutárním městem a není proto uvedeno ve výčtu statutárních měst Zákona o obcích [13].

2.1.2 Kraje a orgány krajů

Kraje jsou v Ústavě definovány jako vyšší územní správní celky České republiky [10]. Konkrétní vymezení všech 14 krajů včetně hlavního města Prahy a jejich území je poté obsahem ústavního zákona č. 347/1997 Sb. [11]. Další popis krajů vychází ze Zákona o krajích [14], odkazy uvedené v této podsekcí směřují do části 1 tohoto zákona, není-li uvedeno jinak.

Základním orgánem kraje je zastupitelstvo, dalšími jsou rada, hejtman, krajský úřad a zvláštní orgány kraje. Blíže se jimi zabývá hlava IV a V. Je zjevná podobnost mezi orgány krajů a obcí, které se sice liší názvem, ale jejich funkce je z velké části podobná a mohou tak být popsány společně [23]. Většina sdílených aspektů již byla popsána v předchozí podsekcí 2.1.1 týkající se obcí. Dále jsou proto popisovány jen klíčové informace ohledně krajů a jejich orgánů včetně odkazů do zákona a jsou zdůrazněny odlišnosti oproti obcím.

Občanem kraje je dle § 12 fyzická osoba s českým občanstvím, která má trvalý pobyt v obci nebo na území vojenského újezdu na území kraje. Podobně jako v případě občana obce má občan kraje právo volit a být volen do krajského zastupitelstva. Také může nahlížet do rozpočtu kraje, usnesení a zápisů z jednání zastupitelstva, usnesení rady, výborů a komisí rady a pořizovat z nich výpisy, nebrání-li tomu jiné zákony.

Zastupitelstvo

Zastupitelstvu kraje se věnuje díl 1 hlavy IV. Dle § 31 má zastupitelstvo 45, 55 nebo 65 členů, dle specifikovaného rozmezí počtu obyvatel kraje. Počet zastupitelů, kteří mají být zvoleni, musí být oznámen na úřední desce.

Pravomoci zastupitelstva vymezuje především § 35. Patří mezi ně mj. vydávat obecně závazné vyhlášky kraje, schvalovat strategie rozvoje územního obvodu kraje a rozpočet kraje, zřizovat a rušit příspěvkové organizace a organizační složky kraje, vydávat obecně závazné vyhlášky, zřizovat a rušit výbory a volit jejich členy, volit hejtmana, náměstky hejtmana a další členy rady (z řad zastupitelů kraje). Oproti obcím mají mj. navíc právo předkládat návrhy zákonů Poslanecké sněmovně.

Zasedáním zastupitelstva se zabývají § 39–45. Zasedání krajského zastupitelstva se také konají dle potřeby, minimálně však každé 3 měsíce. Jsou svolávána a řízena hejtmanem. Rovněž jsou veřejná a krajský úřad o jejich konání musí informovat na úřední desce.

Dle § 43 se z jednání zastupitelstva pořizuje zápis. Povinný obsah je identický jako v případě obcí, svým podpisem jej místo starosty potvrzuje hejtman nebo náměstek hejtmana. Zápis musí být dostupný k nahlédnutí na krajském úřadu.

Volby do zastupitelstev krajů se řídí zákonem č. 130/2000 Sb. [15]. Údaje požadované na kandidátních listinách a následně vytištěné na hlasovacích lístcích (popsané v § 21 a 24) jsou shodné jako u obcí. Dle § 44 jsou výsledky voleb zveřejněny krajským úřadem vyvěšením na úřední desce a celkové výsledky jsou zveřejněny Státní volební komisí sdělením ve Sbírce zákonů.

Další orgány kraje

Radou se zabývá díl 2 hlavy IV. Radu kraje tvoří hejtman, náměstci hejtmána a další zvolení členové. Na rozdíl od rady obce existuje rada kraje vždy. Rada má 9 nebo 11 členů, podle počtu obyvatel kraje.

Výbory a komise se zabývá hlava V. Zastupitelstvo vždy zřizuje výbor finanční, kontrolní, pro výchovu, vzdělání a zaměstnanost. Komise jsou popsány v § 80 a jejich role je stejná jako v případě obcí.

2.2 Data zastupitelstev a jejich zdroje

Po představení zastupitelstev a jejich legislativního ukotvení je možné se na zastupitelstva podívat z pohledu dostupných informací a dat. Shrnutí se primárně týká dat produkovaných zastupitelstvy se zaměřením na hlasovací data. Jsou ale zmíněna i další data souvisejících orgánů. Zaměření je především na povinnost jejich zveřejňování, formát a dostupnost.

Ze zasedání zastupitelstva i rady vzniká zápis, který musí minimálně obsahovat položky vymezené zákonem. Není však vymezen žádný předepsaný formát. Pro tuto práci jsou významné především položky počet přítomných zastupitelů a průběh a výsledky všech hlasování. Jména přítomných zastupitelů nejsou požadována, ale je třeba zaznamenávat změny v jejich počtu [47]. Zápisy musí v obou případech být uloženy na úřadu (obecním či krajským), kde jsou k dispozici k nahlédnutí – z jednání zastupitelstva pro všechny občany obce či kraje, z jednání rady pouze pro zastupitele. Dle zákona č. 106/1999 Sb. (Zákon o svobodném přístupu k informacím) právo na informace ze zápisu z jednání zastupitelstva náleží i dalším osobám, které o ně požádají [12, 26, 44].

Výsledky hlasování jsou povinnou součástí zápisů, není ale kladen požadavek zaznamenat hlasy jednotlivých zastupitelů ani není předepsán formát. Výsledek by měl obsahovat počet hlasů pro možnost pro, proti, zdržel se a nehlasoval [47]. Také by mělo být uvedeno o jakém bodu se jednalo a je doporučeno popsat širěji celý průběh projednávání bodu. Existuje také doporučení [47] vycházet při formulaci přijatých usnesení ze Zákona o obcích, např. „Zastupitelstvo obce vydává Obecně závaznou vyhlášku obce...“ K hlasováním je dále třeba poznamenat, že navzdory tomu, že jednání zastupitelstva jsou veřejná, samotná hlasování mohou být tajná [26].

Data o hlasování mohou být dostupná i ve formě hlasovacích protokolů, jako je tomu např. u Zastupitelstva města Brna [52]. Hlasovací protokoly nejsou definované v zákoně, ale některá města s tímto termínem operují. Lze je definovat jako dokumenty obsahující údaje týkající se jednotlivých hlasování. Typicky jsou výstupem systémů pro zaznamenávání hlasování (viz sekce 2.3) a mohou tak obsahovat podrobnější data než zápis, především hlasování jednotlivých zastupitelů.

Existují různé přístupy k zaznamenávání hlasování s odlišnou detailností. Významný je také formát záznamu, který udává obtížnost strojového zpracování. Z tohoto hlediska jsou nejvhodnější plnohodnotné datové sady hlasování, čímž jsou míněny datové sady obsahující volby jednotlivých zastupitelů. Jeden záznam v takové datové sadě typicky reprezentuje jedno hlasování (hlasování k jednomu bodu) nebo volbu individuálního zastupitele pro dané hlasování [51]. Je však třeba poznamenat, že může existovat i datová sada, která obsahuje pouze méně detailní záznamy výsledků jednotlivých hlasování. V zápisu ze zasedání je ekvivalentem plnohodnotné datové sady reprezentace každého hlasování jmenným seznamem zastupitelů pro každou hlasovací možnost. Méně detailní záznamy o hlasování pak mohou obsahovat pouze sumární čísla pro každou hlasovací možnost nebo pouze výsledek

hlasování [52]. Popsané možnosti zaznamenávání hlasování lze seřadit sestupně podle detailnosti, ostatním možnostem je navíc nadřazena plnohodnotná datová sada na základě vhodnosti ke strojovému zpracování:

- datová sada (plnohodnotná)
- jmenné seznamy
- sumární čísla
- výsledek hlasování

Zákon dále definuje právo občanů obce či kraje nahlížet do dalších dokumentů a porizovat si z nich výpisy, jedná se např. o rozpočet nebo usnesení rady, výborů a komisí. Zákon zároveň ukládá povinnost zveřejňovat některé údaje na úřední desce, např. se jedná o informace o nadcházejícím jednání nebo informace týkající se voleb. Pro krajské úřady a obecní úřady obcí s rozšířenou působností existuje dle zákona č. 261/2021 Sb. povinnost metadata informací zveřejněných na úředních deskách zveřejňovat jako otevřená data [39].

Určitá data týkající se kandidátů a zvolených zastupitelů jsou také vymezena volebními zákony. Kandidátní listiny obsahují informace o kandidátech, které lze poskytnout poté, co proběhne jejich registrace [49]. Výsledky voleb jsou dle zákona zveřejněny ve Sbírce zákonů. Rovněž jsou zveřejněny na webových stránkách Českého statistického úřadu¹, kde lze vyhledat výsledky pro jednotlivé obce či kraje. Je možné zobrazit seznam všech kandidátů s jejich jménem a politickou příslušností. Pro každého kandidáta je také uveden počet získaných hlasů a příznak zisku mandátu.

Informace týkající se zastupitelstev typicky bývají k dispozici na webových stránkách municipalit [51]. Webové stránky také mohou v elektronické podobě obsahovat některé z dokumentů, které mají obce či kraje povinnost zveřejňovat nebo nabízet k nahlédnutí. Pokud zastupitelstva zveřejňují nějaká data ve formě datové sady, typicky k tomu používají datové portály nebo katalogy [51]. Příklady dostupných hlasovacích dat pro konkrétní municipality jsou dostupné v kapitole 3.

Součástí digitalizačních iniciativ ve státní správě je publikování otevřených vládních dat (*open government data*, OGD) přístupných pomocí digitálních platform [41]. Pro organizaci dostupných dat slouží **datové katalogy**, které lze jednoduše popsat jako seznam datových sad [22, 35]. Katalog je tvořen kolekcí metadat popisujících datové sady, které umožňuje vyhledávat a používat [22]. V oblasti otevřených vládních dat slouží datové katalogy pro přístup k datovým sadám státních organizací [35]. Dalším pojmem jsou otevřené **datové portály**, což jsou rovněž rozhraní sloužící ke zveřejňování dat. Na rozdíl od katalogů se však může jednat o celou platformu sloužící pro interakci s otevřenými daty, např. poskytováním vizualizací, statistik, analýz či sdílením článků [33, 43]. Základní vlastností každého portálu je však poskytování přístupu k samotným datům skrze datový katalog.

V České republice je pro účel zveřejňování vládních dat ve formě datových sad k dispozici **Národní katalog otevřených dat (NKOD)**². NKOD je vymezen § 4c Zákona o svobodném přístupu k informacím jako informační systém sloužící k evidování informací zveřejňovaných jako otevřená data [12]. Tyto informace musí být poskytovány způsobem umožňujícím dálkový přístup. Subjekty, které jsou povinny zveřejňovat otevřená data, tak povinně činí prostřednictvím NKOD. Katalog je součástí webového Portálu otevřených dat³.

¹Český statistický úřad – Výsledky voleb – <https://www.volby.cz/>

²Národní katalog otevřených dat – <https://data.gov.cz/datov%C3%A9-sady>

³Portál otevřených dat – <https://data.gov.cz/>

2.3 Systémy na zaznamenávání hlasování

Způsob zaznamenávání hlasování ani formát těchto záznamů není zákonem stanoven. Zastupitelstva proto mohou využívat různé přístupy k vytváření těchto záznamů, včetně jejich ručního vytváření, což může být použito v menších obcích. Další možností je použití hlasovacích zařízení [26]. Použití hlasovacích systémů je typické pro větší města [51].

V dřívější práci byly identifikovány dva systémy používané v České republice: H.E.R. Systém a systém Ministr [52]. Dále v Česku existuje např. systém Usneseni.cz⁴ nebo aplikace UZO⁵ (Usnesení zastupitelstva a rady obce). Jako obecnější řešení lze uvést konferenční systémy společnosti Bosch. Nicméně existují i další možné alternativy, včetně proprietárních řešení (využívá např. Zastupitelstvo města Brna [29]).

Dále je blíže zkoumán H.E.R. Systém, systémy Ministr a systémy společnosti Bosch. Pro každý systém jsou také na základě konkrétního příkladu municipality, která systém používá, popsány produkované hlasovací protokoly. Ukázky zobrazení výsledků hlasování v jednotlivých systémech jsou k vidění na obrázcích 2.3, 2.4 a 2.5.

H.E.R. Systém

H.E.R. Systém je hlasovací a konferenční systém nabízený v Česku firmou BitEST⁶. Výrobce je slovenská firma A.S.Partner⁷, která má zastoupení ještě v Polsku. Pro shrnutí deklarovaných vlastností systému je možné vyjít z dostupné uživatelské příručky k bezdrátové verzi systému [2] a webových stránek společnosti BitEST [3].

Základní části systému jsou řídicí počítač s nainstalovaným softwarem, řídicí jednotka a jednotlivá hlasovací zařízení a mikrofony. Existuje více provedení hlasovacích zařízení, od zabudovaných po jednodušší bezdrátová. Prezence zastupitelů je kontrolována pomocí jejich hlasovacího zařízení, primárně pomocí zasunutí hlasovací karty. Systém eviduje 5 možností hlasování – pro, proti, zdržel se, nehlasoval a nepřítomen.

Schůzi pomocí nainstalovaného programu řídí předsedající. Systém udržuje pro každé volební období databázi zastupitelů, kterou je možné editovat a měnit délku mandátu zastupitelů. Dále se do systému zadávají jednotlivá zasedání, ke kterým se vážou údaje o hlasování. Je možné zadat jednotlivé body jednání, které se následně použijí pro vytvoření hlasování. Případně se zadají přímo při vytvoření hlasování. Systém nabízí pro každé hlasování možnost exportu výsledků ve formátu XML, HTML a PDF. Pro konfiguraci systému se používá soubor `her.ini`, kde je možné nastavit např. délku hlasování, číslo volebního období nebo automatický tisk výsledků hlasování. Také je zde možné změnit výchozí hodnoty textů v aplikaci. Pro systém jsou nabízeny další rozšiřující moduly, např. pro distanční jednání.

Systém je v současné chvíli v České republice nainstalován např. ve statutárních městech Ostrava a Třinec, městech Sokolov a Kutná Hora, městských obvodech Ostrava-Jih a Plzeň 3 a v Kraji Vysočina [3].

⁴Systém Usneseni.cz – <https://www.usneseni.cz/>

⁵Aplikace UZO – <https://uzob.cz/>

⁶H.E.R Systém na stránkách společnosti BitEST – <https://www.bitest.cz/hlasovaci-system/>

⁷Výrobce H.E.R. Systému, společnost A.S.Partner – <https://www.aspartner.sk/>

Zastupitelstvo Ostravy ke každému jednotlivému hlasování zveřejňuje hlasovací protokoly (nazývané *Výsledky hlasování*)⁸. Jedná se o dokumenty ve formátu HTML⁹. V hlavičce je uvedeno číslo zasedání, datum a číslo a název bodu ke kterému se hlasuje. Následně je uveden text usnesení. Pro hlasování jsou uvedena sumární čísla pro všech 5 hlasovacích možností. Dále jsou po jednotlivých klubech uvedeny hlasy jednotlivých členů. Pro každého člena je uvedeno jméno a hlasovací možnost. Pro jednotlivé kluby jsou také uvedena sumární čísla.

HLASOVANIE č. 12 - Výsledek			9:46
Dr. V. Groß	Ing. T.Kirchler	Dr. J. Berger	
F.Grünwald	F.Schwarzenberg	J. Ziegler	
Ing. F.Grossman	Ing. P.Berkovit	Ing. F.Grün	
Mgr. F. Herzog	Mgr. Guldenburg	Ing. H.Motyka	
Mgr. A. Hartman	J. Piltz	Ing.G.Lobkowitz	
S. Hauptman	J. Vavricek	Dr. H.Novacek	
M. Zweig	E. Klein	Mgr.A.Deimler	
Ing. J.Gruber	T. Schülze	Gros-Albenhouse	
Ing. K.Hoffer	Ing.H.Schöller	Dr. M.Klimek	
Ing. A.Schuster	Ing. H.Klein	Ing. K. Benz	
Dr. Popowitz	Ing. L.Berg	Dr.phil.G.Enxin	
J. Tischler			
Za: 25	Proti: 1	Zdržalo sa: 3	

Obrázek 2.3: Snímek obrazovky výsledků hlasování v H.E.R Systému. Převzato z [2].

Ministr

Společnost Ministr Systems vyrábí hlasovací a konferenční systémy Ministr nabízené v několika různých verzích [38]. Pro systémy není veřejně dostupný žádný manuál či dokumentace, ale je možné vyjít z popisu jednotlivých verzí systému na webu společnosti.

Základní složky systému jsou v zásadě stejné, jednotlivá hlasovací zařízení jsou připojena k řídicí jednotce a počítači. Existuje bezdrátová verze systému jak s dedikovanými hlasovacími zařízení, tak s použitím tabletů či notebooků pro hlasování. Hlasovací jednotky mají 3 základní možnosti hlasování – pro, proti, zdržel se. Dále systém dle uvedených snímků obrazovky pracuje s možností nehlasoval.

Systém obdobně umožňuje řízení diskuze a elektronické hlasování. Detaily nejsou uvedeny, ale z dostupných snímků obrazovky lze vidět, že systém má uložené informace o zastupitelích včetně jejich klubů. Dále umožňuje hlasovat k jednotlivým bodům a výsledky

⁸Statutární město Ostrava – Zastupitelstvo města – Návrh programu, materiály, usnesení, výsledky hlasování, audiovizuální záznamy, zápisy – <https://www.ostrava.cz/cs/urad/mesto-a-jeho-organy/zastupitelstvo-mesta/usneseni>

⁹Pro několik novějších hlasování je dostupný pouze dokument ve formátu PDF obsahující všechna hlasování, formát je odlišný, ale pro jednotlivá hlasování obsahuje stejně podrobné údaje.

archivuje. Systém navíc umožňuje zobrazování obrazových materiálů a i v základní verzi nabízí pořizování zvukového záznamu indexovaného podle bodů jednání a diskutujících. Systém archivuje všechna data jednání včetně zvukového záznamu a umožňuje jejich export, pro univerzální verzi systému Ministr 7u jsou uvedeny formáty PDF a Microsoft Excel. Pro systém je k dispozici rozšíření o video systém, který řídí kamerový záznam a online vysílání zasedání.

Společnost poskytuje seznam uživatelů svého systému rozdělený na kabelovou a bezdrátovou verzi. Poslední verze z 15. 12. 2022 uvádí pro kabelovou verzi např. statutární města Jablonec nad Nisou a Olomouc, města Mariánské Lázně a Domažlice, několik městských částí Prahy a Brna a Olomoucký kraj. Pro bezdrátovou verzi je Olomoucký kraj uveden taktéž, dále Moravskoslezský kraj, Rada Pardubického kraje a další municipality.

Zastupitelstvo Olomouckého kraje pro každé zasedání zveřejňuje hlasovací protokol (nazvaný *Přehled jmenovitého hlasování*) obsahující výsledky všech hlasování ve formátu PDF¹⁰. Pro každé hlasování je v hlavičce uvedeno číslo zasedání, datum, číslo a výsledek hlasování a číslo a název bodu hlasování. Poté následuje seznam všech zastupitelů, který obsahuje 3 sloupce – jméno, politický klub a hlasovací možnost. Kromě 4 výše popsaných hlasovacích možností jsou dále uvedeny možnosti omluven a nepřítomen. V patičce je uvedeno kvorum, sumární čísla a procenta pro 4 hlasovací možnosti a celkový počet omluvených a nepřítomných.

Je třeba poznamenat, že dokument neobsahuje explicitní zmínku o systému Ministr, nicméně použití systému je možné vidět na záznamech z jednání¹¹. Město Jablonec nad Nisou publikuje pro každé zasedání zastupitelstva hlasovací protokoly (*Hlasování zastupitelů*) k jednotlivým hlasováním ve formátu TXT¹². Obsah dokumentů a jejich formát je stejný jako v případě popsaných dokumentů v PDF.

Hradiště

25. zasedání Zastupitelstva města
Předseda: Ing. Stanislav Orság

HLASOVÁNÍ č. 1

2.1.2015 10:26:24
01_schválení programu jednání

Orság S.	Sedlák J.	Kaňáková T.	Kropáčová B.	Oborný M.
Zemánek J.	Konečný P.	Nerušil J.	Svoboda L.	Stonawský P.
Pospíšil P.	Puszkailer L.	Vojkůvková A.	Knopf J.	Papica L.
Šrámek K.	Czabe L.	Šnajdr A.	Vepřek B.	Bejdák L.
Endrůdy P.				

Ke schválení je potřeba 11 hlasů
Hlasování o usnesení
Celkem zastupitelů: 21
Omluveno: 0

Pro: 13 Proti: 5 Zdrželo se: 3 Nehlasovalo: 0

Obrázek 2.4: Snímek obrazovky výsledků hlasování v systému Ministr. Převzato z [38].

¹⁰Olomoucký kraj – Zastupitelstvo – Zápisy ZOK – <https://www.olkraj.cz/zapisy-zok-cl-4995.html>

¹¹Olomoucký kraj – Archiv vysílání – <https://live.publicstream.cz/olomouc-kraj/archiv>

¹²Statutární město Jablonec nad Nisou – Zastupitelstvo – <https://www.mestojablonec.cz/cs/samosprava/zastupitelstvo/>

Konferenční systémy Bosch

Jednou z možných mezinárodních alternativ českých systémů jsou konferenční systémy vyráběné firmou Bosch. Následující přehled vychází z manuálu pro balík softwaru pro systémy DCN Next Generation [5]. Architektura systému je složitější než u předchozích systémů, např. je oddělen řídicí server, konfigurační aplikace, aplikace pro řízení zasedání a tiskový server. Každá aplikace může, ale nemusí být nasazena na odlišném počítači. Systém je složen z různých částí, které jsou aktivovány na základě licencí, např. hlasování, čtení identifikačních karet nebo streaming dat ve formátu XML dalším částem systému.

Konfigurační aplikace umožňuje správu uživatelů a jejich skupin s různými právy. Odděleně od uživatelů jsou definováni „delegáti“ (*delegates*), což jsou účastníci jednání, tedy např. zastupitelé. Aplikace dále umožňuje vytvoření šablon pro různé typy hlasování (např. požadovaná většina, tajné hlasování, délka hlasování), je možné definovat zasedání, která se mohou dělit na části (*sessions*) a jednotlivá hlasování. Hlasování mohou používat různé hlasovací možnosti dle nastavení, např. pouze ano/ne nebo 4 možnosti – navíc možnosti „zdržel se“ a „neúčastnit se hlasování“.

Tisková aplikace umožňuje tisk výsledků hlasování. Export dat se zde neřeší, ale je okrajově zmíněn v dalších dokumentech [6]. Vzhledem k existenci modulu pro streaming dat ve formátu XML lze předpokládat, že data je možné exportovat do strojově zpracovatelného formátu.



The screenshot shows a voting interface with a dark blue background. At the top, there is a header bar with the date '01-11-2010', a subject line 'This is line one of the subject for this voting' and 'This is line two of the subject for this voting', and the time '11:29'. Below the header, there is a list of participants with their names and the option they selected. At the bottom, there is a summary of the votes.

Participant	Vote
Algra Rendert	CDA
Aptroot Charlie	VVD
Atsma Joop	CDA
Balkenende Jan-Peter	CDA
Bashir Farshad	SP
Besselink Marianne	PvdA
Bilder Eddy	CDA
Biskop Jack	CDA
Bommel Harrie van	SP
Bos Wouter	PvdA

Total: 49 Yes: 44 Abstain: 3
Not voted: 1 No: 2

Obrázek 2.5: Snímek obrazovky s příkladem možného zobrazení výsledků hlasování ze systému Bosch DCN. Převzato z <https://community.boschsecurity.com/t5/Conference-Discussion/What-integration-options-are-available-for-DCN-NG/ta-p/8354> [cit. 19. 2. 2024].

Systém DCN byl používán ve městě Birmingham, jak bylo zjištěno v průzkumu dostupných hlasovacích dat (viz kapitola 3). Dále je třeba zmínit, že v současné době Bosch nabízí i nový konferenční systém DICENTIS¹³. Tento systém používá např. zastupitelstvo města Pisa [7].

Ze zastupitelstva města Birmingham jsou hlasovací protokoly dostupné u některých starších hlasování ve formátu PDF, jedná se o naskenované vytištěné dokumenty ze systému

¹³Bosch – Konferenční systémy – <https://www.boschsecurity.com/cz/en/solutions/conference-solutions/>

Bosch DCN. Digitální verze dokumentů (tj. nativní PDF) jsou dostupné z parlamentu Fidži¹⁴. Dokument v hlavičce obsahuje název zasedání, číslo a název hlasování a datum. Poté je uveden celkový počet přítomných členů, kteří hlasovali a nehlasovali, následovaný sumárními čísly pro 4 hlasovací možnosti. Následuje shrnutí po politických skupinách obsahující celkový počet zástupců a sumární čísla. Poté následuje seznam hlasování jednotlivých zástupců podle jmen. V případě zastupitelstva města Birmingham jsou zastupitelé seskupeni podle hlasovacích možností, pro každou je uveden celkový počet hlasů, a pro každého je uvedeno členství v politické straně. Protokoly z fidžijského parlamentu používají seskupení po politických skupinách a pro každého poslance je uvedena hlasovací možnost.

Shrnutí

Zastupitelstva v České republice jsou orgány obcí nebo krajů, což jsou územní samosprávné celky. Právně vycházejí především ze Zákona o obcích a Zákona o krajích, byly ale zmíněny i další související zákony. Zastupitelé jsou voleni na čtyřletá funkční období. Jejich hlasování probíhá na zasedáních zastupitelstva, která se konají minimálně každé 3 měsíce. Kromě zastupitelstva mají územně správní celky i další orgány, např. radu nebo starostu či hejtmána, kteří jsou voleni z řad zastupitelů.

Dále byla představena data produkovaná zastupitelstvy se zaměřením na hlasovací data. Ze zasedání zastupitelstev je povinnost pořizovat zápis, do kterého mají občané právo nahlížet. V zápisu musí být uvedeny výsledky hlasování, není však předepsaný konkrétní formát tohoto dokumentu, pouze minimální požadovaný obsah. Byly uvedeny různé možnosti zaznamenání hlasování a jejich výsledků. Pro další zpracování je vhodné publikování dat ve formě datových sad nebo ve strojově zpracovatelných formátech, což však není povinné a vytváření a zveřejňování takových dat probíhá pouze dobrovolně.

V souvislosti se zaznamenáváním hlasování byly představeny hlasovací systémy a uvedeno několik příkladů. Obecně je možné o zmíněných systémech konstatovat, že umožňují minimálně zadávání a ukládání dat o zastupitelích, zasedáních a hlasováních včetně jejich výsledků. Každý také umožňuje nějakou míru konfigurace. Data ze systémů je možné exportovat, ale možnosti exportu jsou zmíněny spíše okrajově a není popsáno schéma exportovaných dat. Při výběru systému lze zvažovat, zda vybrat ucelené řešení složené z řídicího softwaru i hlasovacích zařízení, nebo naopak zvolit modulárnější řešení, kde je možné vytvořit řešení na míru složené z různých komponent. Dalším hlediskem může být cena nebo dostupnost lokálního řešení namísto nutnosti konfigurovat a lokalizovat univerzální systém. Také je možné použít pouze hlasovací zařízení a výsledky hlasování zpracovat libovolným dalším softwarem.

¹⁴Parlament Fidži – Výsledky hlasování – <https://www.parliament.gov.fj/voting-results/>

Kapitola 3

Průzkum dostupných dat z hlasování zastupitelstev

Hlasování představuje jednu z klíčových činností zastupitelstev, při které zastupitelé využívají politickou moc k ovlivnění života občanů nebo k nakládání s veřejnými prostředky. Povinnost zveřejňovat zápisy ze zasedání zastupitelstev napomáhá občanům kontrolovat činnost jimi volených zástupců. Nicméně, formát pro tyto záznamy není daný, stejně jako není dána povinnost uvést hlasování jednotlivých zastupitelů nebo zápisy zveřejňovat v elektronické podobě.

Problém analýzy hlasování volených zástupců je dlouhodobě řešen na vyšší úrovni národních parlamentů [24, 25]. Specificky otevřeností parlamentů se zabývá např. i iniciativa Open Government Partnership (OGP, *Partnerství pro otevřené vládnutí*)¹.

Cílem této kapitoly je provést průzkum dostupných dat z hlasování zastupitelstev. V první sekci jsou shrnuty dostupné datové sady z hlasování zastupitelstev obcí v České republice a jsou popsány dostupné informace o hlasování zastupitelstev na příkladu konkrétních měst. Poté je popsán průzkum dostupných datových sad a dostupnosti výsledků hlasování vybraných municipalit v zahraničí. Pro identifikaci používaného typu záznamu o výsledcích hlasování jsou používány termíny představené v sekci 2.2.

3.1 Česká republika

Vzhledem k počtu obcí v České republice není v rozumném čase možné průzkum provést individuálně pro každou z nich. Nicméně, cílem je především zhodnotit stav a formát dostupných dat hlasování zastupitelstev, proto se stačí zaměřit především na obce, které jsou v přístupu k otevřeným datům napřed.

Národní katalog otevřených dat

Pro nalezení dostupných datových sad hlasování zastupitelstev je možné použít NKOD. Tento průzkum byl již proveden [53] a následně aktualizován v roce 2022 [52]. Celkem bylo nalezeno 9 datových sad poskytovaných 6 municipalitami – Praha, Praha 6, Děčín, Hradec Králové, Karviná, Brno.

Situace k 7. 1. 2024 je podobná, počet datových sad hlasování zastupitelstev se zvýšil na 10 (při použití stejného dotazu *hlasování zastupitelstva*). Přibyla datová sada sta-

¹OGP – Open Parliaments – <https://www.opengovpartnership.org/policy-area/open-parliaments/>

tutárního města Most, které rovněž zveřejnilo datovou sadu hlasování své rady. Jedná se o jedinou datovou sadu hlasování rady dostupnou v NKOD. Stav datových sad hlasování zastupitelstev v NKOD a porovnání s předchozím průzkumem je pro jednotlivé municipality následující:

Praha Publikováno největší množství datových sad, celkem 4, počínaje funkčním obdobím 2010–2014 až po současné zastupitelstvo. Sady jsou ve formátu CSV, je k dispozici i jejich schéma, které je ve všech případech stejné. Frekvence aktualizací není uvedena, ale data jsou zřejmě aktualizována po každém zasedání, jelikož datová sada obsahuje data včetně posledního zasedání.

Praha 6 Existuje jediná datová sada pro období 2014–2018, nicméně jsou dostupná pouze metadata a nefunkční odkaz na dokumentaci. Dříve existovaly odkazy pro stažení dat ve formátu PDF a XML, stažení ovšem kvůli chybě nebylo možné [52, 53]. Nyní už na stránce žádné odkazy ani informace o formátu dat nejsou.

Děčín K dispozici jsou dvě datové sady. První je popsána „Statistika hlasování zastupitelů města Děčína“. Data jsou ve formátu CSV a není k nim k dispozici další popis, schéma ani hlavička CSV souboru. Zdánlivě se jedná o hlasování jednotlivých zastupitelů z jednání zastupitelstva konaného 28. 2. 2019.

Druhá datová sada je složena z 50 souborů, 49 z nich obsahuje data ve formátu CSV, poslední je textový soubor s metadaty. Každý soubor obsahuje záznamy o hlasování z jednoho zasedání, stejně jako u první datové sady. Nejstarší data jsou z 28. 2. 2019², nejnovější z 29. 6. 2023. Datová sada tedy obsahuje data zasahující do funkčního období dvou zastupitelstev. Oproti předchozímu průzkumu přibyly záznamy za rok 2022 a 2023. Není ale jasné, jaká je procedura a frekvence provádění aktualizací. V době minulého průzkumu chyběla data za období zhruba 1 roku, mezitím byla data doplněna, ale v současné chvíli opět chybí data za 6 měsíců. Lze proto předpokládat, že se aktualizace provádí manuálně a pouze několikrát za funkční období zastupitelstva.

Hradec Králové V dřívějších průzkumech byla nalezena datová sada, u které z důvodu chyby nešla stáhnout data. Tato datová sada již není k dispozici vůbec, město poskytuje jedinou datovou sadu, a to svoji úřední desku.

Karviná Jedna datová sada ve formátu XML, bez schématu. Celkem obsahuje 28 záznamů z jednání zastupitelstva, první z 29. 4. 2019, poslední z 11. 12. 2023. Datová sada tedy obsahuje data z funkčního období dvou zastupitelstev (2018–2022 a 2022–2026) a je pravidelně aktualizována. Dle popisu jsou data exportována z hlasovacího systému.

Brno Jedna datová sada ve formátu JSON. K dispozici je i schéma, odkaz je však pouze na datovém portálu města Brna, v NKOD chybí. Datová sada obsahuje data pro všechna funkční období zastupitelstev, počínaje obdobím 2018–2022. Data jsou aktualizována do týdne od skončení zastupitelstva. Pro detaily ohledně datové sady a jejího vzniku viz [29].

Most Podobné řešení jako Brno. Je k dispozici jedna datová sada ve formátu JSON včetně odkazu na datový katalog města, chybí ale schéma. Datová sada obsahuje data ze

²Data pro zasedání zastupitelstva Děčína z 28. 2. 2019 ve druhé datové sadě jsou identická s těmi v první, až na upravený bod hlasování v jednom řádku. Lze spekulovat, že první datová sada sloužila pouze pro zkušební první zveřejnění dat.

dvou funkčních období zastupitelstva, nejstarší z 18. 11. 2021. Datová sada hlasování rady je řešena identicky, nejstarší záznam je z 14. 10. 2021. Obě datové sady jsou dle popisu aktualizovány do týdne od jednání daného orgánu.

Celkem tedy existuje 5 municipalit, které publikují svá hlasovací data v NKOD – Praha, Děčín, Karviná, Brno a Most. Je vhodné poznamenat, že se jedná exkluzivně o statutární města, což podporuje myšlenku, že zaznamenávání a případné publikování hlasovacích dat je prováděno především většími městy. S výjimkou Děčína jsou data vcelku pravidelně aktualizována. Všechny tyto datové sady by mělo být možné dále zpracovat strojovým způsobem, byť náročnost potřebného předzpracování by byla různá dle použitého schématu, členění dat atd.

Je možné vidět značné rozdíly ve formátu poskytovaných dat. Schéma datových sad je v každém případě odlišné, liší se i detailnost popisu použitého schématu. Pro Prahu a Brno je schéma k dispozici (i když u Brna chybí odkaz a je nutné ho dohledat), pro Most a Karvinou chybí, v případě Děčína se jedná pouze o názvy sloupců v textovém souboru. Dále se liší použitý formát datové sady – CSV, JSON, XML, kdy data jsou vždy poskytována jen v jednom formátu. Dále se datové sady liší i svojí granularitou. Praha používá jednu datovou sadu obsahující jeden soubor pro každé funkční období zastupitelstva. Děčín má jednu (relevantní) datovou sadu rozčleněnou do souborů po jednotlivých zasedáních. Ostatní datové sady jsou kompletní a obsahují data pro všechna zastupitelstva.

Zajímavé jsou též samotné názvy datových sad a použitá klíčová slova. Název datové sady vždy obsahuje slovo hlasování a většinou slovo zastupitelstvo, v případě Prahy a Prahy 6 je nahrazeno zkratkou konkrétního zastupitelstva. Samotný název municipality není v názvu vždy uveden, např. datové sady pro Brno, Most i Děčín (obě) se jmenují „Hlasování zastupitelstva“. Vzhledem k tomu, že součástí vyhledávání v NKOD není název poskytovatele, tak je možné tyto datové sady rozlišit jen pomocí popisu. Ani v případě klíčových slov není žádná shoda, nejčastější je kombinace „zastupitelstvo“ a „hlasování“, tato klíčová slova jsou kromě Karviné, Prahy 6 a první datové sady Děčína použita vždy. V případě Karviné chybí klíčové slovo „hlasování“, je použito „zastupitelstvo“ a „hlasování zastupitelstva“. Brno jako jediné používá klíčová slova i v angličtině, např. „city council“ a „voting“.

Ve srovnání s předchozím průzkumem začalo nově poskytovat data město Most, jehož datová sada se až na problém s chybějícím schématem řadí spolu s Brnem a Prahou k těm nejkvalitnějším. Město Děčín po delší odmlce přidalo nová data, ale v současné chvíli již opět chybí data za téměř půl roku. Co se nedostupných dat týče, pro Prahu 6 byly odstraněny odkazy na data a zůstala pouze metadata datové sady, datová sada pro Hradec Králové byla odstraněna kompletně. Obecně se z hlediska množství publikovaných hlasovacích dat situace od roku 2022 mírně zlepšila díky průběžnému zveřejňování nových dat a přidání datových sad Mostu.

Webové stránky

Fakt, že obec nezveřejňuje svá hlasovací data v NKOD, neznámá, že žádné informace o hlasování zastupitelstva nejsou k dispozici. Jak již bylo zmíněno dříve, obce mají povinnost zveřejňovat zápisy ze zasedání obsahující výsledky hlasování a téměř každá obec má k dispozici oficiální webové stránky. Lze proto předpokládat, že mnohé využijí možnost zveřejnit tyto informace elektronicky. Ve většině případů je nutné počítat s tím, že data na rozdíl od plnohodnotných datových sad, nebudou ve strojově zpracovatelném formátu, typické je použití dokumentových formátů, např. PDF.

Pro analýzu stavu dostupných dat o hlasování je nutné individuálně prohledat oficiální webové stránky jednotlivých obcí. Výběr obcí opět vychází z předchozích prací, u statutárních měst Liberec a Zlín byla provedena analýza prezentace hlasovacích dat [52] a město Jilemnice bylo analyzováno v rámci průzkumu otevřených dat ve veřejné správě [42].

Liberec Na webových stránkách města³ je sekce s informacemi o jednotlivých orgánech města. Dokumenty z jednání zastupitelstva a rady od roku 2016 jsou dostupné v odděleném systému⁴. Pro zastupitelstvo je dostupný zápis a oddělený dokument pouze s výsledky hlasování, oba ve formátu PDF. Dokument s výsledky hlasování je oskenovaný vytištěný hlasovací protokol z H.E.R. Systému. V archivu dokumentů na webu města jsou stejné dokumenty dostupné již od roku 2009. Výsledky hlasování rady jsou v systému dostupné jen v rámci zápisů z jednání, kde jsou reprezentovány sumárními čísly.

Dále je na stránkách města k dispozici odkaz na živé vysílání zasedání zastupitelstva a zápisy z jednání výborů, které také obsahují sumární čísla pro výsledky hlasování. Vzhledem k tomu, že výsledky hlasování minimálně od roku 2009 zaznamenává hlasovací systém, mělo by být možné tato data publikovat ve strojově zpracovatelném formátu i vytvořit datovou sadu, byla-li by pro to vůle ze strany města.

Zlín Výsledky hlasování ani zápisy z jednání zastupitelstva nejsou dostupné⁵, pouze výpis usnesení s výsledkem (schváleno, neschváleno atp.) ve formátu PDF. K dispozici jsou záznamy z jednání zastupitelstva za rok 2023, na záznamu je možné vidět použití H.E.R. Systému.

Jilemnice Webové stránky města⁶ obsahují sekci pro zastupitelstvo, radu i zápisy z jednání výborů a komisí. Pro zastupitelstvo je k dispozici online přenos i záznamy zasedání a zápisy ve formátu PDF. Zápisy jsou ve formátu naskenovaných vytištěných dokumentů vytvořených v aplikaci UZOb, výsledky hlasování jsou ve formě jmenných seznamů.

Shrnutí

Existuje několik měst, která již publikují hlasovací data svého zastupitelstva ve formě datových sad. Množství zveřejněných dat se pomalu zvyšuje, zároveň ale dochází k odstranění některých datových sad minulých zastupitelstev, které zřejmě byly v minulosti dostupné. Data jsou typicky zveřejňována většími městy, ale není to pravidlem, jak je vidět na příkladu statutárního města Zlín. Na příkladech bylo také ukázáno dobrovolné zveřejňování zápisů ze zasedání zastupitelstva na webových stránkách měst, zároveň se jedná o příklady použití dříve popsaných hlasovacích systémů. Problémem pro další zpracování jsou odlišné formáty zaznamenaných hlasování a zveřejňování výsledků pouze ve formě zápisů ve formátu PDF.

3.2 Zahraničí

Tato sekce se zabývá průzkumem dostupných dat o hlasování zastupitelstev či podobných orgánů municipalit v zahraničí. Cílem je získat širší rozhled o stavu zveřejňování dat o hla-

³Statutární město Liberec – Orgány města – <https://www.liberec.cz/cz/mesto-samosprava/dalsi-organy-mesta/>

⁴Statutární město Liberec – Veřejné podklady – <https://podklady.liberec.cz/>

⁵Statutární město Zlín – Volené orgány – <https://www.zlin.eu/volene-organy-0>

⁶Město Jilemnice – Samospráva – <https://www.mestojilemnice.cz/samosprava/>

sování těchto orgánů a případně o formátu, ve kterém jsou dostupná. Nejedná se, a ani se nemá jednat, o kompletní přehled všech dostupných datových sad týkajících se hlasování zastupitelstev, jelikož vytvoření takového přehledu je problematické i na mnohem menší úrovni České republiky a není žádoucí ho provádět v celosvětovém měřítku. Navíc se vychází z předpokladu, že od otevřených dat je možné očekávat i jejich dohledatelnost. Proto se pro účely této sekce předpokládá, že pokud hledaná data existují, pak by měla být dostupná v rámci datového katalogu či odkazovaná z relevantní sekce webových stránek municipality.

Určitým problémem při vyhledávání bylo při použití klíčových slov jako `vote` nebo `voting` velké množství nálezů týkajících se voleb do obecních zastupitelstev, jejich výsledků, organizace atp. Volební výsledky jsou v demokraciích z principu veřejné a typické je i jejich další zpracování a vizualizace médií [34]. Na základě výsledků vyhledávání je možné usuzovat, že jsou i častěji vyhledávané a dostupné ve formě datových sad. K jejich dostupnosti může také přispívat nižší frekvence jejich vzniku, jelikož volby na rozdíl od hlasování orgánů probíhají typicky jednou za několik let. Po původním vytvoření datové sady voleb pak také není nutné provádět její průběžnou aktualizaci.

Pro nalezení relevantních dat byly použity dva přístupy, podobně jako u České republiky. Prvním přístupem byla snaha o nalezení dat o hlasování zastupitelstev, která jsou publikována ve formě datové sady. Pro tento účel byl použit vyhledávač Google, do kterého byly zadány dotazy s následující kombinací klíčových slov: `city council <vote(s) | voting> <open data | data set>`. Poté byly manuálně zkontrolovány nejrelevantnější nalezené stránky a vybrány ty skutečně obsahující relevantní data. V případě, že výsledkem vyhledávání byl datový katalog municipality, bylo dále provedeno vyhledávání v rámci něj. Výsledky jsou popsány v podsekcí 3.2.1.

Druhým přístupem byl pokus o dohledání výsledků hlasování zastupitelstev konkrétních municipalit. Bylo cíleno na větší města, u kterých je předpokládána větší pravděpodobnost systematického zveřejňování dat, potenciálně i v rámci vlastního datového katalogu. Pro jednotlivá města byla data hledána na oficiálních webových stránkách a pro zjištění používaného typu záznamu výsledků hlasování se vycházelo z posledních dostupných zápisů z jednání zastupitelstva. Výsledky jsou popsány v podsekcí 3.2.2.

3.2.1 Datové sady

Primárním cílem průzkumu bylo nalézt datové sady hlasování, jsou ale zmíněny i další nalezené výsledky, které jsou relevantní z hlediska strojově zpracovatelných dat nebo ucelených vizualizací hlasování. Shrnutí výsledků se nachází v tabulce 3.1. Celkem je uvedeno 8 měst, z nichž pro 6 jsou dostupná hlasovací data ve formě datové sady a s výjimkou jednoho města jsou data stále aktualizována.

Město Victoria nabízí dashboard hlasování vytvořený pomocí nástroje Microsoft Power BI, který umožňuje data zobrazit na různé úrovni podrobnosti (princip datových kostek). Vizualizace vytvořená stejným nástrojem nabízí i město White Rock, které ale data zároveň poskytuje i ve formě datové sady.

Datové sady poskytované městem Edmonton jsou rozdělené po jednotlivých funkčních obdobích. Zajímavé je, že pro novější volební období jsou navíc dostupné ve dvou verzích, kde jeden řádek odpovídá buď celému hlasování (s hlasy všech zastupitelů), nebo volbě jednoho zastupitele.

V případě města Cambridge se jedná explicitně o datové sady obsahující pouze jmenovitá hlasování (*roll call votes*). V popisu je uvedeno, že město používá dva typy hlasování,

druhou možností je „hlasové“ hlasování (*voice vote*), kdy se výsledek hlasování rozhodne na základě zjevné převahy jedné hlasovací možnosti. Město Cambridge je zároveň jediné město v průzkumu, které přestalo datové sady zveřejňovat. Dostupné jsou 2 datové sady pro různá funkční období, nejnovější data jsou z roku 2020.

Město Wigan neposkytuje ucelená data formou datové sady ani vizualizace. Je zajímavé tím, že na svých webových stránkách pro jednotlivá zasedání nabízí nejen zápis ve formě dokumentu ve formátu PDF, ale i zápis přímo na webové stránce. Z detailu zasedání je dále možné přejít na stránku pro každé hlasování, kde jsou hlasy reprezentované tabulkou. Nejedná se sice o ideální formát, ale oproti dokumentovým formátům se stále jedná o strukturovaná data, která je teoreticky možné strojově zpracovat bez nutnosti dalšího předzpracování, jako je např. použití technologií pro optické rozpoznávání textu (OCR, *Optical Character Recognition*) pro naskenované zápisy.

Datové sady každé město zveřejňuje ve svém městském datovém katalogu. Je u nich opět možné vidět rozdíly ve schématu, formátu i granularitě dat. Rozdíly mezi státy jsou pochopitelné, ale i na příkladu Kanady je možné vidět, že datové sady jsou odlišné. Ve srovnání s Českem je také častější možnost stáhnout data v různých formátech.

stát	municipalita	typ	členění	akt.	formát
Kanada	Vancouver	datová sada		ano	CSV, JSON, XLSX
	Victoria	vizualizace		ano	-
	Toronto	datová sada	funkční období soubory	ano	CSV, JSON, XML
	Edmonton	datová sada	funkční období datové sady	ano	CSV, RDF, XML
USA	Cambridge	datová sada		ne	CSV, RDF, XML
	Dallas	datová sada		ano	CSV, RDF, XML
	White Rock	datová sada, vizualizace	funkční období datové sady	ano	CSV
UK	Wigan	webová stránka	hlasování webové stránky	ano	HTML

Tabulka 3.1: Výsledky průzkumu dostupných dat o hlasování zastupitelstev v zahraničí pomocí vyhledávání na základě klíčových slov. Každý řádek obsahuje informace týkající se jedné municipality. Ve sloupci **členění** je uvedeno, zda jsou data rozdělena do více částí a jakým způsobem. Sloupec **akt.** uvádí, jestli jsou data v daném zdroji stále aktivně aktualizována.

3.2.2 Municipality

Tato část průzkumu se zabývá dostupnými hlasovacími daty vybraných municipalit. Cílem bylo především nalézt údaje o výsledcích hlasování, ať už ve formě datové sady, nebo v zápisu z jednání zastupitelstva. Okrajově jsou ale zmíněny i datové katalogy nebo další publikované dokumenty municipalit.

Spojené království Velké Británie a Severního Irska a Nový Zéland

Spojené království bylo pro průzkum zvoleno jako první a bylo pro něj prozkoumáno nejvíce municipalit. Důvodem je, že se jedná o zemi, která je často uváděna jako vzor v oblasti otevřených dat a dlouhodobě se umísťuje na vrcholu mezinárodních žebříků [42, 50].

V této souvislosti je často zmiňován britský katalog otevřených dat⁷. Katalog je v principu srovnatelný s NKOD. Je možné v něm vyhledávat záznamy o datových sadách, které typicky umožňují stažení dat, ale v některých případech mohou nabízet pouze metadata. Jsou k dispozici volební výsledky, včetně voleb do zastupitelstev. Žádná datová sada hlasování těchto orgánů však není k dispozici.

Pro průzkum webových stránek byla vybrána větší města a byli vybráni zástupci z Anglie, Skotska i Walesu. Shrnutí výsledků je k vidění v tabulce 3.2 (část **UK**). Je uveden primární způsob zaznamenávání výsledků hlasování v prohlížených zápisech a případně i další nalezené použité způsoby.

stát	municipalita	kat.	sep.	typ	poznámka
UK	Londýn	ano	ne	sumární čísla, jmenné seznamy	jmenné seznamy např. pro hlasování o rozpočtu
	Birmingham	ano	ano	výsledky, jmenné seznamy	pro starší zasedání naskenované výstupy z hlasovacího systému
	Edinburgh	ano	ano	výsledky, jmenné seznamy	volitelná sekce Voting pro bod jednání vždy obsahuje jmenný seznam
	Glasgow	ano	ano	sumární čísla	
	Cardiff	ne	ano	výsledky	webové stránky mají sekci pro otevřená data
NZ	Auckland	ano	ano	výsledky	občas nedostupné kvůli chybě serveru
	Christchurch	ano	ano	výsledky, jmenné seznamy	

Tabulka 3.2: Výsledky průzkumu dostupných dat o hlasování zastupitelstev vybraných municipalit ze Spojeného království Velké Británie a Severního Irska a Nového Zélandu. Nalezená data jsou dostupná z webových stránek municipalit skrze detail jednotlivých zasedání zastupitelstva. Sloupec **kat.** uvádí, jestli má město vlastní datový katalog. Sloupec **sep.** uvádí, zda jsou data týkající se zasedání oddělená od stránek města, např. na subdoméně nebo ve webovém informačním systému.

Město Birmingham má jako jediné u některých zasedání dostupné dokumenty týkající se pouze hlasování. Jedná se o naskenované hlasovací protokoly z hlasovacího systému společnosti Bosch. Nejnovější ale pochází z roku 2020 a u novějších zasedání již nejsou k dispozici.

Dále byla do průzkumu zahrnuta 2 největší města **Nového Zélandu** – Auckland a Christchurch. Výsledky jsou uvedeny v části **NZ** tabulky 3.2. Obě města používají da-

⁷UK – Katalog otevřených dat – <https://www.data.gov.uk/>

tové portály vytvořené v systému ArcGIS. Obě také používají stejný systém⁸ pro ukládání dokumentů z jednání orgánů municipality a videozáznamů. Zápisy z jednání zastupitelstva jsou dostupné ve formátu PDF i HTML.

Pro každé město je dostupné množství dat týkající se voleného zastupitelstva, případně také starosty nebo jednotlivých komisí. Jedná se ale v naprosté většině o dokumenty ve formátu PDF. Dokumenty se vždy vztahují k jednotlivým entitám a událostem, např. konkrétní jednání dané komise. Všechna zkoumaná města také v nějaké formě publikují otevřená data ve vlastním datovém katalogu nebo na webových stránkách. V žádném případě ale nejsou hlasovací data dostupná v ucelené datové sadě ani ve strojově zpracovatelném formátu.

Je k dispozici seznam zasedání zastupitelstva, ze kterého je možné navigovat na detail každého zasedání, kde jsou dostupné dokumenty. Jedním z dokumentů je zápis ze zasedání, který obsahuje informace o proběhlých hlasováních. Pro zaznamenání výsledků hlasování jsou použity různé způsoby. Někdy je použito více způsobů i v rámci zápisů ze zastupitelstva jednoho města, typicky pokud je vyžadováno jmenné hlasování. Výsledky zaznamenané ve formě jmenného seznamu byly v některých případech použity v zápisu např. ve městech Londýn nebo Christchurch, ale lze předpokládat, že tomu tak může být i u ostatních měst.

Rakousko a Německo

Pro srovnání byl dále proveden průzkum i v německy mluvících zemích. Ve státních datových katalogích^{9, 10} nebyly nalezeny žádné relevantní datové sady. Data z hlasování zastupitelstev byla hledána na stránkách jednotlivých měst, příp. daného voleného orgánu. Vybrána byla města Vídeň, Graz, Salzburg a Linz pro Rakousko, a Berlín, Hamburk a Brémy pro Německo.

Ve všech případech je možné výsledky hlasování nalézt v rámci nějaké formy zápisu ze zasedání. Částečnou výjimkou je město Graz, kde tyto dokumenty nejsou pro stažení k dispozici, ale popis jednotlivých hlasování i s výsledky je dostupný na webových stránkách. Ve více případech jsou dostupné zápisy ve formě kompletního přepisu jednání (stenoprotokoly). V některých případech jsou dostupné i kratší zápisy bližší těm používaným v ČR. Případně mohou informace o hlasování být dostupné i v seznamu usnesení projednaných na zasedání.

Standardním způsobem zaznamenání výsledků hlasování je shrnutí hlasovacích možností po politických subjektech, případně se zmínkou o rebelujících zastupitelích. Tento způsob záznamu zapadá z hlediska podrobnosti mezi sumární čísla a jmenné seznamy, jelikož lze teoreticky odvodit volbu jednotlivých členů, pokud je k dispozici seznam přítomných zastupitelů v době hlasování. Pokud je k dispozici i přepis jednání a dojde ke jmenovitému hlasování (*namentliche Abstimmung*), pak je možné v dokumentu najít i data na úrovni jmenného seznamu. Samotný záznam o výsledcích často v dokumentech není výrazně typograficky oddělený (především u kompletního přepisu jednání, kde je to dáno formou dokumentu). Dále se také mohou lišit použité fráze pro zaznamenání hlasovacích možností a výsledku.

Jako zajímavý příklad lze uvést město **Linz**¹¹, pro které jsou na webu města dostupné dokumenty s kompletním přepisem jednání i detail zasedání včetně výsledků hlasování

⁸Systém Infocouncil – <https://www.infocouncil.com.au/>

⁹Rakousko – Datový portál – <https://www.data.gv.at/>

¹⁰Německo – Datový portál – <https://www.govdata.de/>

¹¹Město Linz – Zasedání zastupitelstva – <https://www.linz.at/Politik/GRSitzungen/>

v HTML podobě. Zápisy město také publikuje jako datové sady dokumentů ve formátu PDF a TXT na městském datovém portálu.

Předchozí závěry platí i pro vybraná města v **Německu**. Je ale třeba zmínit, že situace těchto měst je specifická tím, že se jedná o městské státy – zastupitelstvo je tudíž parlamentem daného státu. Ve všech případech existuje systém pro vyhledávání dokumentů z jednání parlamentu i stenoprotokoly, které jsou dostupné přímo v detailu zasedání na webových stránkách, nebo dohledatelné v systému.

Dokumenty s výsledky jmenovitých hlasování jsou dostupné ze zemského parlamentu spolkové země Bádensko-Württembersko¹². Pro každé hlasování je k dispozici hlasovací protokol ve formátu PDF, kde je volba každého poslance zaznamenána v tabulce.

3.2.3 Projekt Parlameter

V rámci průzkumu byl také nalezen systém pro vizualizaci hlasování zastupitelstev projektu Parlameter¹³, což je slovinský projekt pro digitalizaci dat z jednání volených orgánů [19]. Poskytuje analytické nástroje pro sběr dat z jednání volených institucí [21]. Dále umožňuje vytváření systémů pro prezentaci informací a dokumentů a vizualizaci dat pro volené instituce, jejich členy a jejich hlasování. Koncept tohoto projektu je tak z části podobný systému řešenému v rámci této práce.

Projekt se primárně zaměřuje na parlamenty, jak vyplývá i z jeho názvu. V současné chvíli jej využívají parlamenty Slovinska, Chorvatska, Bosny a Hercegoviny a Ukrajiny [19]. Vizualizace jsou ale použité i pro několik slovinských obcí, včetně hlavního města Lublaně¹⁴. Systém umožňuje pro jednotlivá hlasování stáhnout data ve formátu JSON a CSV, ve formě kompletní datové sady není možné data stáhnout.

Shrnutí

Na základě průzkumu je možné potvrdit, že větší města v České republice i zahraničí typicky publikují na svých webových stránkách dokumenty z jednání svých orgánů, včetně nějaké formy zápisu z jednání zastupitelstva. Je tak možné dohledat výsledky hlasování. Existují ale velké rozdíly v použitých formátech i podrobnosti. Výsledky mohou být dostupné přímo na webových stránkách ve formátu HTML nebo je naopak může být nutné dohledat v dokumentu obsahujícím přepis jednání. Vzhledem k zaznamenaným rozdílům formátu záznamů mezi jednotlivými zkoumanými městy lze usuzovat, že ani v zahraničí neexistuje předepsaný standard pro zveřejňování hlasovacích dat zastupitelstev a jejich zveřejňování je dobrovolné.

U některých anglicky mluvících zemí se je možné v zápisu setkat se zaznamenáním pouze výsledku hlasování na rozdíl od zaznamenání výsledku pomocí sumárních čísel povinného v České republice. Podrobnější výsledky v těchto případech vůbec nemusí existovat, kvůli použití způsobu hlasování, kdy se pouze identifikuje existence většiny pro nebo proti návrhu (viz *voice vote*).

Byly nalezeny jednotky datových sad hlasování zastupitelstva. Pro Českou republiku jsou dostupné v rámci NKOD a potenciálně i v lokálním katalogu daného města. Všechny zahraniční datové sady byly dostupné z datových katalogů jednotlivých municipalit a naopak nebyly nalezeny v rámci státních katalogů. Datové sady obsahují údaje o hlasování

¹²Jmenovitá hlasování zemského parlamentu spolkové země Bádensko-Württembersko – <https://www.la-ndtag-bw.de/home/aktuelles/namentliche-abstimmungen.html>

¹³Projekt Parlameter – <https://parlameter.org/>

¹⁴Parlameter – Lublaň – <https://ljublana.parlameter.si/>

jednotlivých členů, ale nemusí obsahovat informace o všech proběhlých hlasování, protože obsahují jen jmenovitá hlasování.

Použité technologie se rovněž mezi jednotlivými municipalitami liší. Pro dohledání údajů o hlasování jsou vždy základem webové stránky municipality. Hlasování mohou být dostupná přímo na stránkách nebo jako dokumenty ke stažení, nebo může být použit externí systém pro správu dokumentů. Použití různých existujících řešení je možné vidět také u nástrojů na vizualizaci dat a u datových katalogů. Také byl představen existující systém pro vizualizaci hlasování projektu Parlameter.

Kapitola 4

Informační systémy a použitelnost uživatelských rozhraní

Tato kapitola představuje informační systémy a jejich architekturu. Vzhledem k zaměření práce se dále blíže zabývá technologií webových aplikací. Nakonec jsou představena uživatelská rozhraní se zaměřením na jejich použitelnost a možnosti vyhodnocování použitelnosti.

4.1 Informační systémy

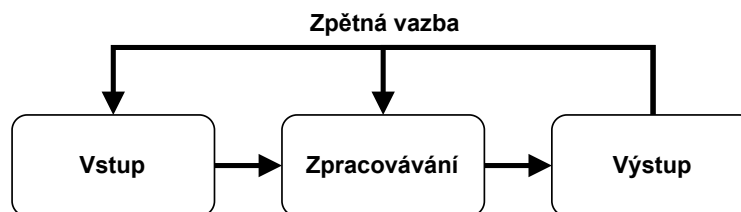
Informační systémy (IS) jsou v dnešní době běžnou součástí lidského života a jsou využívány velkými i malými společnostmi. Informační systémy lze dle [32] definovat jako množinu souvisejících komponent, které sbírají, zpracovávají, ukládají a distribují informace, které pomáhají rozhodování a řízení organizace. Obecněji lze IS definovat jako množinu komponent, které sbírají, manipulují, ukládají a šíří data a informace a poskytují mechanismus zpětné vazby za účelem splnění cílů [48].

Důležitým pojmem v kontextu informačních systémů je informace. Informaci lze definovat jako data upravená do formy, která má význam a užitek pro člověka [32]. Data jsou naopak nezpracovaná fakta reprezentující události v organizaci či fyzickém prostředí. Tato fakta ještě nebyla zpracována do formy, které lidé mohou porozumět a použít [32]. Je možné rozlišovat různé kategorie dat, např. textová, obrazová nebo zvuková. Složením nebo uspořádáním dat tak, aby měla význam a přidanou hodnotu oproti individuálním faktům, se stanou informacemi [48].

Lze identifikovat tři základní aktivity, které informační systém provádí pro splnění svého účelu [32, 48]. Prvním krokem je **vstup**, kdy jsou sbírána data. Následuje **zpracovávání**, kdy jsou vstupní data konvertována nebo transformována do smysluplné podoby. Zpracováním dat vzniká užitečný výstup, který je typicky uložen. Posledním krokem je **výstup**, kdy jsou produkovány užitečné informace, např. ve formě dokumentů [48]. Případně jsou informace ve vhodné formě předány členům organizace [32]. K těmto třem základním aktivitám je navíc třeba definovat **zpětnou vazbu**, kterou je myšlen výstup, který je opět vrácen do systému a vede ke změnám ve vstupu nebo zpracovávání. Schéma aktivit systému je zobrazeno v diagramu 4.1.

Informační systémy je možné z hlediska jejich použití dělit na dva základní typy [31]:

- **operativní IS (OIS):** Slouží pro řízení každodenních operací. Mají spíše menší rozsah dat a je jednodušší specifikovat požadavky na takový systém. Mají význam primárně pro vnitřní chod organizace.



Obrázek 4.1: Schéma základních aktivit informačního systému. Adaptováno z [48].

- **manažerské IS (MIS):** Slouží pro podporu rozhodování managementu, poskytují strategické výhody díky zlepšování kvality a rychlosti rozhodování. Je těžší dopředu vymezit potřebná data a operace, častěji používají *ad hoc* dotazy.

Dosud byly informační systémy popisovány obecně, typické, a pro tuto práci relevantní, jsou ale především informační systémy založené na počítačích. Takové IS jsou složeny z hardwaru, softwaru, databází, telekomunikací, lidí a procedur [48].

Infrastruktura a komponenty systému

Základem pro provoz informačních systémů je infrastruktura informačních technologií (*information technology infrastructure*) [32]. Zahrnuje v sobě hardware, software a služby a poskytuje platformu pro provoz IT systémů.

Infrastruktura prošla dlouhým vývojem, z přístupů používaných v současné době je třeba uvést především architekturu **klient–server**, která nabyla na významu s rozšířením osobních počítačů a rozvojem velkých informačních systémů připojených k síti [31, 32]. Klienty představují počítače připojené k výkonému serveru, který jim nabízí služby. Klient slouží pro uživatelský vstup a server zpracovává a ukládá data. Nejjednodušší variantou tohoto přístupu je dvouvrstvá architektura (*two-tiered architecture*), kdy je klient připojen přes síť k jedinému serveru. Na základě požadavků organizace mohou být přidány další úrovně serverů s různými odpovědnostmi. Tento přístup se nazývá vícevrstvá či n-vrstvá architektura.

Z hlediska softwaru a komunikace nabývají na významu **webové služby**, což jsou komponenty, které spolu komunikují na základě univerzálních webových standardů [32]. Základní technologií umožňující tento vývoj byl jazyk XML, který posloužil jako univerzální jazyk pro zasílání zpráv přes webové protokoly.

Běžným prostředkem pro ukládání dat jsou **databáze** [48]. Databáze jsou kolekce dat, které mohou sloužit mnoha aplikacím. Pro práci s databázemi slouží systém řízení báze dat (*Database Management System*, DBMS), software, který umožňuje efektivní správu dat a poskytuje aplikacím přístup k nim [32]. DBMS poskytuje rozhraní pro přístup k datům v databázích, díky tomu aplikace nemusí řešit detaily fyzického uložení dat.

4.2 Webové aplikace

Intuitivně lze webové aplikace chápat jako aplikace dostupné na webu. Zdroj [8] jako webové aplikace uvádí tradiční webové stránky, jednostránkové webové aplikace i API (*application programming interface*, aplikační programové rozhraní). Definice dle [1] je striktnější a říká, že webová aplikace je software běžící ve webovém prohlížeči. Zdroj dále odlišuje webové

aplikace od statických webových stránek, zároveň však uvádí, že webové stránky mohou být poskytovány webovými aplikacemi.

Z hlediska webových služeb a API je možné použít členění na „lidský“ a programovatelný web [46]. Programovatelným webem je míněna část webu, sloužící pro zpracování programy namísto zobrazení lidmi. Může k tomu používat odlišné formáty, např. XML. Lidský web je složen z webových stránek a aplikací. Jelikož jsou v obou případech doručovány přes webové technologie, jedná se o webové služby a jsou také součástí programovatelného webu.

Popularita webových aplikací je dle [37] dána tím, že různá zařízení připojená k internetu mají nainstalovaný webový prohlížeč. Ten implementuje všechny základní webové technologie a webové aplikace tyto klientské technologie mohou používat a díky tomu jsou poměrně platformě nezávislé.

Jako webové aplikace je tedy možné chápat všechny aplikace, které jsou dostupné prostřednictvím webu nebo používající webové technologie, bez ohledu na to, jestli poskytují uživatelské rozhraní nebo API.

Technologie

Na web je možné nahlížet skrze model klient–server [20]. Server je stále připojen k internetu a dostává požadavky, na základě kterých vrací požadované webové stránky (nebo obecněji zdroje). Klient zasílá požadavky serverům, typicky skrze webový prohlížeč.

Mezi základní klientské technologie patří HTML, CSS a JavaScript [37]. **HTML** (*Hypertext Markup Language*) je jazyk sloužící k popisu struktury webové stránky. Soubory ve formátu HTML obsahují jednotlivé elementy, které reprezentují části obsahu stránky. **CSS** (*Cascading Style Sheets*) je jazyk sloužící k popisu stylu stránek, udává jak mají jednotlivé elementy vypadat.

JavaScript je programovací jazyk sloužící pro implementaci interaktivních webových stránek. Existuje pro něj značné množství aplikačních rámců (*frameworks*) a knihoven pro vytváření klientských aplikací (*frontend*), např. **React**, **Angular** nebo **Vue.js** [8]. Webové stránky jsou reprezentovány pomocí **DOM** (*Document Object Model*), což je objekt se stromovou strukturou, který slouží jako rozhraní pro interakci s webovými dokumenty [20, 37].

Na serverové straně (*backend*) nejsou technologie pevně dány. Teoreticky je možné použít libovolné programovací jazyky, které umožní vytvořit a odeslat obsah, který bude přijatý na klientské straně. Prakticky je třeba zhodnotit podporu jazyka pro práci se zprávami protokolu HTTP, HTML dokumenty, serializací a deserializací formátů jako JSON a XML atd. Jako příklady používaných technologií je možné uvést frameworky **Ruby on Rails**, **Django** (Python) [46] a **ASP.NET Core** (C#) [37]. Na serverové straně je také možné použít JavaScript, např. framework **Express** [8].

Protokol **HTTP** (*Hypertext Transfer Protocol*) složí pro výměnu souborů na webu, typicky je používán servery a webovými prohlížeči [20]. Jedná se o dokumentový protokol, kterým klient formuluje své požadavky na server [46]. Základní části HTTP požadavku jsou metoda, cesta, hlavička a tělo. Metoda udává typ operace požadovaný od serveru, možné metody jsou definovány standardem. Cesta udává, ke kterému zdroji je přistupováno. Hlavičky jsou páry klíč/hodnota obsahující další informace. Tělo požadavku může obsahovat libovolná data, dle použité metody může a nemusí existovat. HTTP odpověď je složena z kódu, hlaviček a těla. Kód je číslo udávající výsledek zpracování požadavku, hodnoty a význam kódů jsou standardizovány.

Pro identifikování zdrojů v protokolu HTTP slouží **URI** (*Uniform Resource Identifier*). **URL** (*Uniform Resource Locator*) je typ URI reprezentující umístění zdrojů na internetu [20].

Základní složky URL jsou protokol, identifikace serveru (*host*), např. `google.com`, a cesta k požadovanému zdroji na serveru, např. `/resources/file.txt` [8]. Součástí URL může volitelně být dotaz (*query*) složený z párů klíč/hodnota.

Dělení webových aplikací

Aplikace je možné dělit na serverové (*server-side*) a klientské (*client-side*) [8]. Serverová generuje stránky aplikace na serveru a následně je odesílá klientovi. Klientská provádí většinu generování uživatelského rozhraní na klientské straně. Ze serveru typicky dojde k jednomu načtení aplikace, která potom dynamicky mění obsah webové stránky na straně klienta. Dříve byly standardem serverové aplikace, díky rozvoji technologií bylo umožněno přesunout logiku do klientských technologií, kde aplikace mohou použít JavaScript pro manipulaci DOM a měnit tak uživatelské rozhraní přímo v prohlížeči.

Klientské aplikace jsou také nazývány jednostránkové aplikace (*Single-Page Application*, SPA) [8]. Slovo jednostránkové v názvu nutně neznamená, že aplikace klientovi poskytuje jen jednu webovou stránku. Z pohledu uživatele může v aplikaci existovat libovolné množství webových stránek. Rozdíl je v tom, že individuální stránky nejsou načítány ze serveru. V praxi také nemusí platit, že klientské aplikace načtou ze serveru pouze jednu HTML šablonu, kterou dále upravují. Mohou např. existovat oddělené balíčky používané pro jednotlivé části aplikace. Pro jejich vývoj jsou typicky používány frontend frameworky nebo knihovny jazyka JavaScript.

Pro serverové aplikace také existuje pojem aplikace renderované na serveru (*Server-Side Rendered*, SSR). Rozdělení však není pevně dané, SPA mohou používat SSR pro urychlení načítání stránek.

SPA jsou v dnešní době dominantním přístupem k vývoji webových aplikací [8]. Dřívější přístup k vytváření webových aplikací vyžadoval, aby server posílal všechny potřebné části stránky, tedy HTML, CSS a JavaScript, jako odpověď na každý požadavek. Což mělo za následek, že se velká část dat posílala redundantně, jelikož významná část HTML a CSS zůstávala beze změny. Aby se předešlo této redundantní činnosti zpomalující načítání stránek, začaly se ze serveru načítat pouze data ve formátu JSON nebo XML. Úpravu uživatelského rozhraní pak provedl klientský JavaScript. Postupným vývojem se tak došlo právě k jednostránkovým aplikacím.

Webové služby a REST API

Pro fungování SPA je nezbytné mít možnost načítat ze serveru data namísto dokumentů (celých webových stránek). K tomu slouží **webové služby**, což je obecnější termín, který lze chápat jako API, které je přístupné přes protokol HTTP [8]. Pro implementaci webových služeb již delší dobu existují technologie jako WSDL a SOAP, jako problém těchto technologií je však uváděna přílišná komplexnost [8, 46].

Novějším a jednodušším přístupem je **REST** (*representational state transfer*) [8]. REST je pojem reprezentující sadu požadavků nebo principů, které webová služba musí splňovat. Služba, která je splňuje, může být označena přídatným jménem *RESTful*, tedy odpovídající REST, běžně se pro ni používá název REST API.

REST je bezstavové spojení mezi klientem a serverem, což je vhodné pro použití s protokolem HTTP, který je také bezstavový [8]. Architektury webových služeb odpovídajících REST používají protokol HTTP pro přístup ke zdrojům, metoda je udávána HTTP metodou a cesta odpovídá identifikaci zdroje [46].

Jedním z možných způsobů pro implementování REST API je architektura orientovaná na zdroje (*Resource-Oriented Architecture*) [46]. Zdroje jsou entity, které jsou dostatečně důležité na to, aby na ně bylo odkazováno nezávisle. Každý zdroj je identifikovaný URI. Pro interakci s entitou je možné použít HTTP metody. Základní 4 metody jsou následující:

- GET: načtení zdroje
- PUT: upravení existujícího zdroje nebo vytvoření zdroje se zvoleným URI
- POST: vytvoření nového zdroje
- DELETE: smazání existujícího zdroje

Mimo zmíněných metod existují ještě další, např. HEAD. HTTP metody mají dvě vlastnosti, které mohou být užitečné pro REST API, pokud jsou korektně používány. **Bezpečné** (*safe*) metody GET a HEAD slouží pro čtení dat, a nikoli pro změnu stavu serveru. Provedení několika bezpečných metod by mělo mít stejný efekt jako neprovedení žádné. Prakticky tyto metody mohou mít vedlejší efekty, např. uložení záznamu do logu, ale klient by je neměl provádět kvůli vykonání daného vedlejšího efektu.

Idempotentní (*idempotent*) metody by měly vždy vést ke stejnému výsledku, bez ohledu na to, kolikrát za sebou je identická metoda použita. Bezpečné metody GET a HEAD jsou proto automaticky idempotentní, dále se jedná o metody PUT a DELETE. U metody PUT tímto vzniká omezení, aby metoda modifikovala stav zdroje nahrazením novým stavem, a neprováděla např. inkrementování hodnoty.

4.3 Uživatelská rozhraní a jejich použitelnost

V předchozích sekcích byla přímo i nepřímo zmíněna interakce uživatele se systémem skrze uživatelské rozhraní. **Uživatelské rozhraní** (*user interface*, UI) je možné definovat jako elementy (operačního) systému, které umožňují přístup k počítačovému systému a jeho řízení [48]. **Grafická uživatelská rozhraní** (GUI) pro interakci se systémem používají grafické elementy zobrazené na obrazovce jako jsou ikony nebo menu [48]. Primární způsob interakce s GUI využívá vizuální reprezentace objektů relevantních pro uživatele, který s nimi manipuluje prostřednictvím myši [40]. GUI se na základě akcí uživatele odpovídajícím způsobem aktualizuje, např. posouvání objektu myší. V definici je sice použito slovo myš, prakticky však v dnešní době interakce může probíhat i pomocí jiného zařízení, které ovládá kurzor, případně pomocí dotykové obrazovky.

Uživatelská rozhraní umožňují uživatelům pracovat se systémem a jejich kvalita tak ovlivňuje efektivitu práce se systémem. Zvýšení použitelnosti UI může vést k úspoře času a finančních prostředků [40]. Půžitelnost je pojem, se kterým se je možné setkat u vyhodnocování uživatelských rozhraní informačních systémů i webových aplikací [8, 31]. Zjednodušeně lze říci, že použitelnost udává, jak jednoduché je systém použít.

V oblasti hodnocení interakce mezi systémem a uživatelem existuje i pojem uživatelské příjemnosti/přívětivosti (*user-friendly*). Tento pojem však dle [40] není vhodný. Jeden z důvodů je spíše terminologický – uživatelé nepotřebují, aby k nim systém byl přátelský (pozn. *friendly*), ale aby sloužil svému účelu. Druhým důvodem je, že pojem je příliš jednorozměrný (míra přívětivosti). Pro oblast zabývající se tvorbou uživatelských rozhraní se proto začaly používat i jiné názvy, např. HCI (*Human-Computer Interaction*, interakce člověk-počítač). Jako alternativní termín pro hodnocení uživatelských rozhraní je zavedena právě použitelnost.

Použitelnost (*usability*) je jedním z aspektů akceptovatelnosti systému [40]. S tímto pojmem pracují i mezinárodní standardy ISO. Bylo identifikováno 5 standardů, které se zabývají použitelností software nebo interaktivního systému [36]. Použitelnost je z pohledu HCI podle ISO definována jako míra, do jaké může být produkt používán uživateli k dosažení specifikovaných cílů s efektivností¹ a spokojeností v daném kontextu použití [28, 36].

Použitelnost není jedna vlastnost systému, ale spíše se jedná o soubor různých atributů [40]. Konkrétně zdroj uvádí 5 tradičních atributů:

- **snadnost naučení** (*learnability*): Jak jednoduché je pro uživatele naučit se pracovat se systémem. Cílem je, aby uživatel byl schopný, co nejdříve vykonávat práci.
- **efektivnost** (*efficiency*): Systém by měl uživateli, který s ním umí pracovat, umožnit dosáhnout vysoké míry produktivity.
- **zapamatovatelnost** (*memorability*): Mělo by být jednoduché zapamatovat si, jak se systémem pracovat. Pro uživatele by nemělo být nutné se po pauze v používání systému znova učit, jak s ním pracovat.
- **chyby** (*errors*): Uživatelé by měli při práci se systémem dělat minimum chyb, mělo by být jednoduché se z nich zotavit a neměly by nastávat katastrofické chyby.
- **spokojenost** (*satisfaction*): Používání systému by mělo být příjemné, uživatelé by měli být (subjektivně) spokojeni s jeho používáním.

Použitelnost je sama o sobě abstraktní koncept, a právě proto je pro její vyhodnocení nutné blíže vymezit její jednotlivé komponenty, které mohou být vyhodnoceny nebo měřeny [40]. Použitelnost je typicky měřena pomocí testovacích uživatelů, kteří používají systém pro vykonání předdefinovaných úkolů. Pro určení celkové použitelnosti systému je možné použít průměrné hodnoty měření a porovnat je se stanoveným minimem, také lze stanovit požadavky na celou distribuci získaných měření, např. podmínka, že jen určité malé množství uživatelů dalo systému nejnižší možné hodnocení.

Konkrétní příklady hodnocení použitelnosti lze demonstrovat na některých zmíněných attributech použitelnosti. Pro atribut snadnost naučení lze měřit za jakou dobu je uživatel, který poprvé pracuje se systémem, schopný jej používat pro splnění definovaných úkolů. Pro chyby lze měřit, kolika se uživatel dopustí za dobu, kdy je sledován. Je ovšem nutné přiřadit chybám váhu podle jejich závažnosti. Základní chybou může být jakákoli akce, která nevede ke splnění cíle, naopak katastrofické chyby jsou takové, které mohou vést ke zničení práce uživatele apod. Spokojenost je více subjektivním atributem, jedním ze způsobů jejího hodnocení je použití dotazníků, kde uživatelé typicky hodnotí spokojenost se systémem na numerické stupnici, např. 1–5.

Na hodnocení uživatelských rozhraní lze nahlížet i z pohledu estetiky a krásy. Jak uvádí [28], pojmy používané při hodnocení použitelnosti jako jednoduchost, kvalita designu a příjemnost vycházejí z estetiky. Důvěra ve webové stránky může být založena na tom, jestli uživatel považuje design stránky za přehledný nebo čistý, je tedy hodnocen na základě estetického citění uživatele. Jedno z doporučení návrhu uživatelských rozhraní – jednoduchost, tak nemá pouze technický dopad, kdy je pro uživatele jednodušší systém použít, ale může ovlivnit jeho názor na systém bez ohledu na efektivitu jeho používání. Požadavky na estetiku se mohou lišit mezi jednotlivými skupinami uživatelů, např. technicky zkušení uživatelé mohou docenit i technické aspekty stojící za implementací uživatelského rozhraní.

¹V originále *effectiveness* a *efficiency*, je tedy myšlena efektivita z hlediska splnění požadavků i minimalizace nároků na čas, prostředky atd.

Hodnocení použitelnosti systémů

Testování použitelnosti systémů vyžaduje sběr informací o používání systému a potenciálních problémech, základní metodou je především uživatelské testování [40].

uživatelské testování Umožňuje přímo zjistit, jak uživatelé interagují se systémem a jaké nastávají problémy. Práce s uživateli však přináší nevýhody se spolehlivostí a validitou dosažených výsledků. Mezi uživateli mohou existovat výrazné rozdíly, nelze proto činit závěry z rozdílu ve výsledcích stejného testu pro různé uživatele. Také je třeba zohlednit, zda jsou testovací uživatelé relevantní uživatelé systému a zda zvolené úkoly odpovídají reálným podmínkám používání systému.

přemýšlení nahlas Jedná se o metodu, kdy testovací uživatel nahlas říká své myšlenky při práci se systémem. Umožňuje pochopit, jak uživatel vidí systém a identifikovat chyby v jeho uvažování. Přemýšlení nahlas je však nepřírozené a nemusí být pro uživatele snadné.

pozorování Pozorovatel pasivně sleduje uživatele v jejich činnosti a zaznamenává údaje. V případě, že je to nutné, může požádat o vysvětlení nějaké činnosti. Jinak by však neměl zasahovat do činnosti uživatelů, neměl by tedy např. vysvětlovat, jak systém používat. Metoda může pomoci odhalit neočekávané způsoby používání systému.

dotazníky a interview Jedná se o nepřímé metody, jelikož uživatel přímo neinteraguje s uživatelským rozhraním, pouze odpovídá na připravené otázky. Použití dotazníků je jednodušší, jelikož je snadné je distribuovat uživatelům. Interview jsou interaktivní, typicky obsahují více otevřených otázek a je možné reagovat na problémy, pokud např. uživatel otázku nepochopí.

skupinové diskuze (*focus groups*) Jedná se o techniku, kdy jsou s uživateli diskutovány jejich pocity a požadavky na systém. Problémem jsou nároky na výběr reprezentativních uživatelů. Není zde přímo sledována interakce uživatelů se systémem, což vede k riziku, že názor uživatele nemusí odpovídat reálnému požadavku na systém.

logy Systém může automaticky ukládat údaje o svém používání, které lze později analyzovat. Také je možné zaznamenat kompletní interakci uživatele se systémem (*transcript logging*), kterou lze později pro účely analýzy zrekonstruovat.

zpětná vazba Uživatelé mohou sami poskytnout zpětnou vazbu. Výhodou je, že se jedná o akci iniciovanou uživateli a vychází tak z jejich aktuálních problémů. Do této kategorie spadá i beta testování, kdy je vybraným uživatelům umožněna práce se systémem před jeho vydáním, za účelem získání zpětné vazby.

Článek [30] představuje **metodologii CIUE** (*Comparative Inter-Organisational Usability Evaluation*) sloužící pro porovnání použitelnosti informačních systémů. Deklarovaný cíl je zlepšení použitelnosti IS demonstrováním, že z pohledu uživatele nedosahuje stejného standardu jako jiný systém. Cílem metodologie je používat objektivní data, metody pracující se subjektivními názory jako např. skupinové diskuze jsou proto nevhodné. Je doporučeno používat kontrolované pozorování uživatelů a kvantitativní metody sběru a analýzy dat. Proces je poté rozdělen do 5 kroků:

1. **plánování:** Jsou zvoleny relevantní metriky, vybrány systémy pro porovnávání a je naplánováno uživatelské testování včetně výběru uživatelů.

2. **vyhodnocení použitelnosti:** Provedení uživatelského testování systémů. Je doporučeno použít kontrolované prostředí a zaznamenat maximální množství dat.
3. **extrakce dat:** Přenos dat z nástrojů pro záznam do analytických nástrojů.
4. **analýza:** Je provedena kvantitativní analýza při které jsou porovnány podobné metрики jednotlivých systémů.
5. **prezentace:** Výsledky jsou prezentovány účastníkům CIUE (organizace, jejichž IS byly porovnávány).

Jako konkrétní příklad vyhodnocování použitelnosti IS je možné uvést použití metody **dotazníku SUMI** (*Software Usability Measurement Inventory*) [45]. Dotazník hodnotí spokojenost uživatele se systémem v 5 oblastech: náklonnost (*affection*, pozitivní emoční reakce na systém), efektivita (*efficiency*), nápomocnost (*helpfulness*), kontrola (*control*, zda má uživatel pocit, že má systém pod kontrolou) a snadnost naučení (*learnability*). Uživatelé hodnotí systém u otázek spadajících do uvedených oblastí na stupnici 1–3. Pro následné vyhodnocení získaných dat je k dispozici analytický nástroj a referenční databáze pro porovnání dosažených výsledků.

Pravidla pro návrh uživatelských rozhraní

Použitelnost a její testování slouží pro vyhodnocení a případné detekování problémů při práci se systémem. Je však vhodné snažit se uživatelské rozhraní od začátku navrhovat způsobem, kterým se předejde maximálnímu množství chyb. Způsobem jak toho dosáhnout je použití pravidel pro návrh uživatelského rozhraní (*design rules*) [27]. Zdrojem těchto pravidel je předchozí výzkum vycházející především z lidské psychologie. Nejedná se o konkrétní předpisy, jak se má uživatelské rozhraní implementovat. Spíše jde o obecné návody či principy, které je nutné aplikovat pro konkrétní případy. Jedná se například o konzistentní rozložení elementů, členění pomocí vizuální hierarchie [27], jasně označené možnosti ukončení interakce nebo použití zkratk (ikony nebo klávesové zkratky) [40].

Návrhová pravidla lze rovněž použít pro **heuristické vyhodnocování** [40], kdy je kvalita uživatelského rozhraní hodnocena na základě vybraných principů použitelnosti. Vyhodnocení provádí malá skupina hodnotitelů, kteří zkoumají UI a posuzují, zda odpovídá vybraným heuristikám (pravidlům).

Shrnutí

Byly představeny informační systémy a webové aplikace, jako jeden ze způsobů, jak je implementovat. Pro webové aplikace je relevantní rozvoj knihoven jazyka JavaScript pro vytváření klientské části systému a komunikace se serverem pomocí API. Pro implementaci API je populární přístup REST. Jako možný přístup pro jeho implementaci byla uvedena architektura orientovaná na zdroje a byly popsány relevantní metody protokolu HTTP.

Následně byla zkoumána oblast uživatelských rozhraní a hodnocení jejich kvality, především na základě aspektu použitelnosti. Byly popsány různé atributy tvořící použitelnost, způsoby získávání dat pro její hodnocení a byly uvedeny příklady konkrétních přístupů pro hodnocení.

Kapitola 5

Analýza problému a specifikace požadavků

Tato kapitola se zabývá analýzou požadavků na zadávání hlasovacích dat a správy municipalit v systému pro vizualizaci hlasování. Jako první je analyzován existující systém pro vizualizaci hlasování Zastupitelstva města Brna, který tato práce dále rozšiřuje. Následně jsou definováni cíloví uživatelé systému a specifikovány požadavky na úpravy systému a jeho rozšíření pro správu dat.

Zdrojem informací byla analýza existujícího systému (ve verzi k 19. 1. 2024), analýza fungování zastupitelstev a průzkum dostupných dat z hlasování zastupitelstev municipalit. Cíle a požadavky na systém byly dále konzultovány s vedoucí práce a dalšími členy projektu.

5.1 Systém pro vizualizaci hlasování Zastupitelstva města Brna

Systém byl vytvořen v rámci diplomové práce [52] a v současné chvíli je nasazen na serveru města Brna. Cílem této práce je přidat do původního systému podporu pro vizualizaci hlasování zastupitelstev více municipalit a dále přidat možnost spravovat municipality a informace o hlasování jejich zastupitelstev. Pro tyto účely mi byl poskytnut přístup k repozitáři se zdrojovým kódem systému.

Primárním účelem existujícího systému je vizualizace dat z hlasování Zastupitelstva města Brna. Při vytváření systému se však předpokládala možná rozšíření, byl proto kladen důraz i na použitelnost pro jiné municipality a také byl vytvořen obecný model vstupních dat. Další část této sekce slouží k seznámení s existujícím systémem, vychází z textu diplomové práce [52] i zdrojového kódu systému. Je popsána architektura systému, použité technologie, uživatelé a změny oproti původní implementaci.

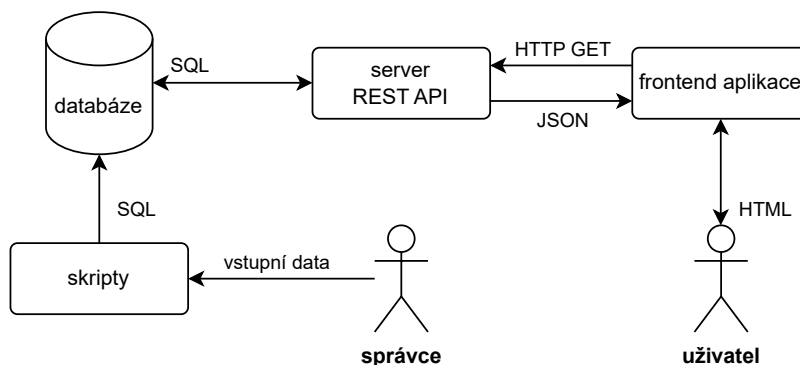
Systém je implementován jako webová aplikace s dvouvrstvou architekturou klient–server. Architektura systému je k vidění v diagramu 5.1. Základní komponenty systému jsou následující:

- **klient (frontend aplikace):** Uživatelské rozhraní je poskytováno webovou aplikací implementovanou v programovacím jazyce JavaScript a knihovně React¹. Aplikace je rozdělena na jednotlivé stránky, které poskytují detaily či vizualizace týkající se

¹Knihovna React – <https://react.dev/>

jednotlivých entit systému, např. zastupitelů či hlasování. Aplikace komunikuje se serverem pomocí REST API.

- **server (backend aplikace):** Implementován v programovacím jazyce Python za použití mikro webového frameworku Flask². Na základě požadavků přijatých přes API načítá a předzpracovává požadovaná data z databáze a vrací je ve formátu JSON. Přímo na serveru jsou také pomocí skriptů vkládána nová data do databáze.
- **databáze:** Pro uložení dat je použita relační databáze MariaDB³. Pro vytvoření schématu databáze je k dispozici inicializační SQL skript.



Obrázek 5.1: Diagram architektury systému pro vizualizaci hlasování Zastupitelstva města Brna. Adaptováno z [52].

Cíloví uživatelé a role v systému

Cílem projektu je zvýšit zájem občanů o komunální politiku a poskytnout jim vizualizace hlasování Zastupitelstva města Brna. Systém slouží pro zobrazení hlasovacích dat pro libovolné uživatele bez omezení. Cílová skupina uživatelů systému je proto poměrně široká. Systém slouží pouze pro prezentaci dat a nepracuje s uživatelskými účty, každý **uživatel** v cílové skupině tak má v systému stejnou roli. Jedná se o občany s volebním právem, tedy starší 18 let. Předpokládá se, že úroveň jejich technických znalostí může být různá. Systém je proto navržen pro jednoduché pochopení včetně možnosti zobrazení nápověd.

Navíc je třeba zmínit roli **správce**, což je administrátor s přístupem k serveru, kde je aplikace nasazená. Správce je jediný, kdo má možnost a právo modifikovat data uložená v databázi a ovlivnit tak data zobrazovaná systémem.

Hlasovací data a obecný model vstupních dat

Pro vkládání dat do systému jsou dostupné dvě možnosti. První je specifická pro zastupitelstvo města Brna a zahrnuje i předzpracování dat v poskytnutém formátu, tato metoda je prakticky používána pro nasazený systém pro město Brno. Druhá metoda používá obecný model dat a slouží pro případné doplnění dalších municipalit do systému. Navíc je využíváno modifikace dat přímo v databázi v případech, kdy je třeba provést korekce chyb

²Framework Flask – <https://flask.palletsprojects.com/>

³V původní práci byl uveden databázový systém MySQL, ale pro nasazení se v současné chvíli používá MariaDB. Nicméně pro účely systému jsou oba databázové systémy ekvivalentní.

ve vstupních datech. Tento přístup je akceptovatelný, jelikož data jsou v rámci systému jinak statická a až na opravy chyb se v čase nemění.

Hlasovací data zastupitelstva města Brna jsou získávána z hlasovacích protokolů z jednání zastupitelstva ve formátu HTML. Pro jejich zpracování slouží skript, jehož vstupem jsou složky obsahující jednotlivé hlasovací protokoly. Každá složka odpovídá jednomu zasedání zastupitelstva a protokol v ní jednomu hlasování. Skript provede syntaktickou analýzu HTML souborů pomocí balíčku *Beautiful Soup*, zkontroluje údaje, dopočítá další odvozené hodnoty a nahraje data do databáze. Pro nahrání dat ve formátu JSON odpovídajících obecnému modelu je také k dispozici skript, který pracuje s jedním vstupním souborem.

Obecný model vstupních dat je definován jako soubor, případně objekt, ve formátu JSON. Definice modelu je k dispozici ve dvou formátech, jako schéma (*JSON Schema*) a jako definice v programovacím jazyce TypeScript. Oba přístupy je možné použít pro případnou validaci vstupních dat, první je běžnější pro uživatelské použití, schéma v tomto formátu je použito např. u několika datových sad hlasování v NKOD.

Model používá české identifikátory, aby se vyhnul pokřivení významu jednotlivých údajů a aby korespondoval se jmény použitými v modelu entit systému. Na nejvyšší úrovni je model rozdělen na 4 základní části:

- **municipalita**: Objekt se základními informacemi o dané municipalitě.
- **politickeSubjekty**: Seznam politických subjektů, což mohou být politické strany nebo hnutí, jejich koalice nebo, jako v případě Brna, zastupitelské kluby⁴.
- **zastupitele**: Seznam členů zastupitelstva, také obsahuje informace o členství zastupitelů v politických subjektech.
- **zastupitelstva**: Informace o jednotlivých zastupitelstvech v seznamu po jednotlivých funkčních obdobích. Obsahuje jednotlivá zasedání a data o proběhlých hlasováních.

Entity v modelu mají přiřazeny unikátní identifikátory, které slouží např. pro odkaz na politický subjekt, jehož je zastupitel členem, nebo pro referenci na zastupitele pro jednotlivé hlasy zastupitelů u hlasování. Identifikátory také slouží pro jednotnou identifikaci zastupitelů a politických subjektů přes více funkčních období. Položky datového typu řetězec mají omezenou délku dle odpovídající definice v databázi. Model také definuje číselník pro hlasovací možnosti, který obsahuje následujících 6 možností:

- | | |
|-----------------|-------------------------------|
| • A: ano | • Z: zdržel se |
| • N: ne | • 0: omluven ⁵ |
| • X: nehlasoval | • - ⁶ : nepřítomen |

Obecný model zároveň slouží jako schéma pro datové sady. V rámci API je k dispozici koncový bod pro stažení datové sady pro jedno funkční období zastupitelstva v tomto formátu, který je pro uživatele dostupný ze stránky *O projektu*. Takto vytvořená datová sada byla rovněž použita pro ověření správné funkčnosti skriptu na vkládání dat v tomto

⁴Existence klubů je definována v jednacím řádu Zastupitelstva města Brna, viz <https://www.bрно.cz/w/zastupitelstvo-mesta-brna>.

⁵Hlasovací možnost omluven se jako jediná v rámci systému nepoužívá a ani není definována v číselníku v databázi, protože ji Zastupitelstvo města Brna nepoužívá.

⁶Pozn.: jedná se o znak krátká pomlčka (*en dash*, Unicode znak U+2013), nikoli spojovník (Unicode znak U+002D, číslo 45 v ASCII).

formátu. V rámci testování byla také ručně ověřena správnost hlasovacích dat uložených v databázi a vytvořené datové sady. Systém obsahuje data počínaje funkčním obdobím zastupitelstva 2018–2022 a nová data jsou stále přidávána, celkem tedy obsahuje data za dvě funkční období.

Uživatelské rozhraní

Webová aplikace je implementována jako SPA v knihovně React a programovacím jazyku JavaScript. Projekt je vytvořený pomocí nástroje **Create React App**, který poskytuje výchozí konfiguraci nástrojů pro překlad, vytváření balíčku atd, kterou tak není třeba vytvářet a spravovat ručně.

Mezi hlavní závislosti použité v aplikaci patří knihovna **React Router** pro směrování v aplikaci a členění aplikace na stránky. Pro komunikaci se serverem se používá knihovna **axios** poskytující klienta pro provádění HTTP požadavků. Aplikace dále využívá komponentovou knihovnu **PrimeReact** poskytující širokou škálu komponent pro jednotlivé prvky uživatelského rozhraní. Pro komponenty se z knihovny rovněž používá sada stylů (*theme*), dostupná ve formě CSS souboru, která udává výchozí vzhled komponent.

Aplikace je rozdělena na jednotlivé stránky, mezi kterými lze navigovat pomocí odkazů v obsahu stránky nebo hlavním menu. Struktura aplikace a jednotlivé stránky jsou definovány pomocí komponenty **BrowserRouter** z knihovny **React Router**. Komponenta obsahuje definici mapování mezi URL cestami a komponentami uživatelského rozhraní implementovanými v knihovně React. Komponenty, které mají definovanou cestu a z pohledu uživatele tedy odpovídají jedné webové stránce, jsou dále nazývány stránkami.

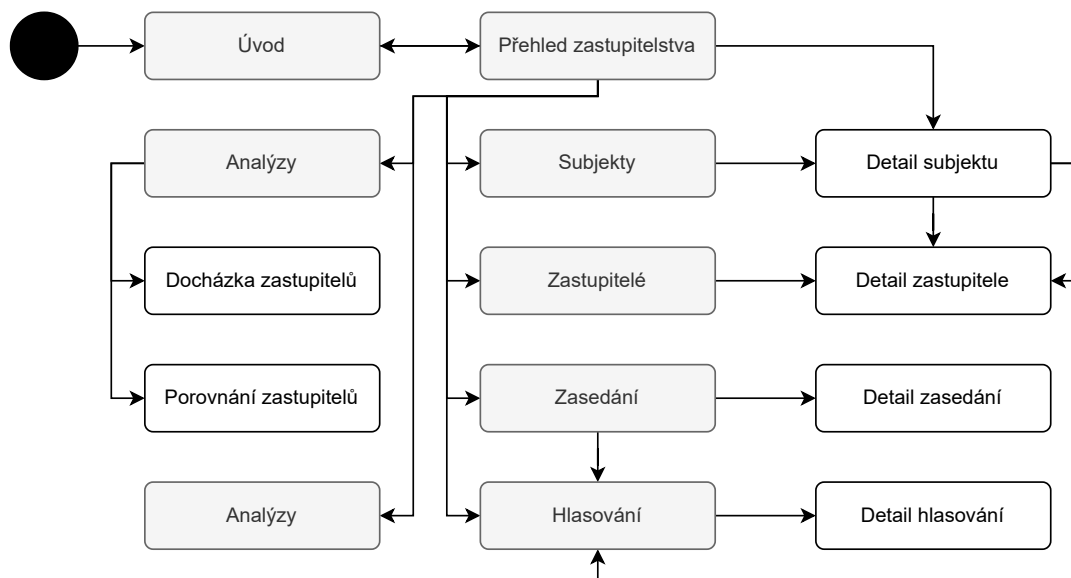
S výjimkou úvodní a chybové stránky se pro všechny stránky aplikace používá stejná šablona, která poskytuje rozložení stránky a opakující se komponenty. Šablona dělí stránku na 3 základní části – záhlaví obsahující menu s odkazy na vybrané stránky, zápatí s informacemi o projektu a tělo stránky. Obsah těla stránky je daný konkrétní stránkou. Pro implementaci stránek se používají komponenty založené na třídách. Plánuje se však přechod projektu na funkcionální přístup, kdy se pro definici komponent využívají funkce. V rámci funkcí se pro interakci s funkcionalitou knihovny React (např. pro aktualizaci stavu nebo reakce na jeho změny) používají speciální funkce nazývané *hooks*.

Stránky aplikace lze rozdělit do 4 základních skupin – přehledy pro jednotlivé entity, detaily, analýzy, a ostatní stránky jako úvod, informace o projektu atd. Jednotlivé stránky a možnosti navigace mezi nimi jsou vizualizovány v diagramu návaznosti obrazovek 5.2. Každý stav reprezentuje jednu webovou stránku v systému. Je vynechána chybová stránka, na kterou je přesměrováno v případě chybného URL nebo nesprávných parametrů (např. pro filtr).

Jednotlivé stránky i soubor `index.html` (základní HTML šablona sloužící jako vstupní bod React aplikace) obsahují poměrně velké množství statického textu, který je specifický pro Zastupitelstvo města Brna, např. název města a zastupitelstva, místo zasedání, titul primátorka atp.

Backend a REST API

Pro komunikaci s backend částí systému je poskytováno REST API, které na základě přijatých požadavků vrací statická data z databáze ve formátu JSON. Pro implementaci je použit programovací jazyk Python a framework Flask. Mezi nejdůležitější závislosti (balíčky) patří právě framework Flask a další rozšiřující balíček **Flask-RESTful**, které byly použity pro vytvoření aplikace poskytující REST API. Funkcionalita pro jednotlivé kon-



Obrázek 5.2: Diagram návaznosti obrazovek v systému pro vizualizaci hlasování Zastupitelstva města Brna. Na stránky s šedým pozadím je možné přejít pomocí odkazu v záhlaví stránky ze všech stránek kromě stránky Úvod, pro zjednodušení jsou tyto odkazy uvedeny pouze ve směru od stránky Přehled zastupitelstva. Také jsou vynechány některé další odkazy mezi ostatními stránkami a chybová stránka.

kové body je definována pomocí tříd, jejichž metody odpovídají jednotlivým metodám HTTP požadavků. Koncové body jsou následně vytvořeny propojením třídy s URL cestou. Koncové body REST API používají pouze bezpečnou metodu **GET**. Na základě žádného dotazu tedy nedochází ke změně dat ani stavu serveru.

Každý koncový bod má URL cestu začínající prefixem `/flask/`. Vzhledem k použití pouze metody **GET** se v žádném případě nepracuje s tělem HTTP požadavků. Koncové body lze na základě práce s parametry rozdělit na dvě základní skupiny:

- **bezparametrické:** Slouží pro načtení kompletního seznamu určitého typu dat, např. koncový bod `/flask/councils` pro načtení seznamu zastupitelstev (funkčních období) pro municipalitu.
- **parametrické:** Obsahují parametr v rámci URL cesty a případně ještě v dotazu. Většina koncových bodů je parametrických, jelikož je ve většině případů vyžadována vazba minimálně na konkrétní zastupitelstvo. Tyto koncové body mohou vracet jak seznamy, tak data pro konkrétní entitu. Jako příklad je možné uvést koncový bod `/flask/<int:council_id>/vote/<int:vote_id>` pro načtení dat konkrétního hlasování. Použitá notace odpovídá definici koncového bodu používaného v rámci balíčku **Flask-RESTful**. Ve špičatých závorkách jsou uvedeny parametry v cestě s datovým typem a názvem.

Parametry v dotazu jsou použity pro koncový bod filtrování hlasování a porovnání dvou členů zastupitelstva.

Pro interakci s databází se používá balíček **mariadb** (resp. **mysql-connector-python**). Při vykonávání akce pro každý koncový bod se vytvoří připojení k databázi na základě konfigurace serveru. Pro načítání dat se používají parametrizované **SQL** dotazy a databázový

kurzor. Většina koncových bodů nevrací přímo data konkrétní entity ale předzpracovaná data pro konkrétní komponentu frontend části aplikace, např. koncový bod `sessionsData`, který vrací všechna potřebná data pro zobrazení seznamu zasedání namísto pouhého seznamu entit zasedání.

Databázový model byl vytvořen s ohledem na rozšiřitelnost, je proto obecný a lze použít pro zastupitelstvo libovolné municipality. Jediným drobným problémem je použití zkratky ZMB (Zastupitelstvo města Brna) v názvech některých klíčů, což však nemá efekt na použitelnost schématu pro jiné municipality.

Zjednodušená reprezentace schématu databáze je k vidění v diagramu 5.3. Mezi základní entity systému patří zastupitelstvo, politický subjekt, člen (zastupitel), zasedání, hlasování (jeden hlasovací bod) a hlasování člena. Každá je reprezentována vlastní tabulkou v databázi. Dále existují tabulky s číselníky, např. pro výsledky hlasování (přijato, nepřijato) nebo hlasovací možnosti (viz popis obecného modelu výše). Schéma také obsahuje několik tabulek, které aktuálně v systému nejsou používány, ale jsou připraveny pro potenciální budoucí rozšíření. V databázi jsou dále obsaženy tabulky, případně sloupce, obsahující statistiky – sumární hodnoty vypočítané při vkládání dat sloužící pro urychlení načítání dat pro účely vizualizace. Tyto statistiky jsou v diagramu 5.3 zvýrazněny.

Významná je entita `PrisluSiK` reprezentující členství člena v politické straně pro určité funkční období. Jedná se o časový interval ohraničený dvěma daty, který je vytvořen na základě dat konání zasedání při zpracování vstupních hlasovacích dat. Počátek intervalu je dán datem prvního zasedání, na kterém se člen účastnil hlasování. Koncové odpovídá datu posledního zasedání, kterého se člen zúčastnil. V případě účasti po celé funkční období je tak interval definován od data prvního zasedání po datum posledního zasedání. Nepracuje se tedy např. s datem začátku a konce funkčního období, které typicky není z hlasovacích protokolů dostupné. V případě, že člen v průběhu funkčního období změnil členství v politickém subjektu, pak má několik záznamů v tabulce členství, původní končící k datu posledního zasedání s členstvím v původním subjektu a následující začínající datem dalšího zasedání se členstvím v novém subjektu (členství tedy nejsou spojitá).

Členství určují účast člena na zasedání, kdy všichni členové, jejichž interval členství obsahuje datum zasedání, jsou členy pro dané zasedání. Tato vazba není přímo reprezentována v databázovém schématu, ale je zajištěna na aplikační úrovni při vkládání dat, jelikož interval členství je odvozen od členství člena v politickém subjektu pro konkrétní zasedání ve zdrojových datech. Člen zasedání má vždy záznamy v tabulce statistik účasti (`BylPritomen`) a hlas (`HlasovaniClena`) pro každé hlasování na daném zasedání⁷.

Rozšíření původního systému

Obsah této sekce vychází jak z původní práce, tak i ze zdrojového kódu, ke kterému jsem měl přístup od října 2023. Popis proto již zohledňuje některé změny provedené od dokončení původní práce. Některé aspekty dalšího vývoje je však vhodné zmínit explicitně.

Po nasazení původního systému pro město Brno na serveru města byla navázána spolupráce s městem Most, které začalo nově publikovat datové sady (jak bylo popsáno v sekci 3.1 o průzkumu dostupných hlasovacích dat v České republice). Pro tyto účely byla provedena úprava systému, při které byly nahrazeny řetězce specifické pro město Brno v uživatelském rozhraní. Nově se tyto hodnoty načítají z konfiguračního souboru. Následně byla vytvořena nová instance systému s konfiguračním souborem obsahujícím hodnoty pro město Most

⁷Záznamy o hlasování člena existují i pro zasedání, kterých se člen nezúčastnil. Hlas je reprezentován pomocí hlasovací možnosti *nepřítomen*.

navržen tak, aby byl rozšiřitelný pro více municipalit. Databázový model je proto poměrně obecný a byl vytvořen i obecný model dat ve formátu JSON. Tento model je zároveň použit pro export dat ze systému ve formě datových sad.

Systém byl také nasazen pro další municipalitu – město Most, přičemž byly odstraněny některé konstanty v systému platné pouze pro město Brno. Všechny změny však byly prováděny manuálně a systém zatím nemá podporu pro více municipalit, jedná se o dvě instance systému s různou konfigurací.

Existující model pokrývá všechny relevantní entity pro reprezentaci hlasování jednoho zastupitelstva. V případě, že budou odhaleny další možnosti hlasování či jiná specifika pro další zastupitelstva, mělo by ho být možné rozšířit, stejně jako v případě úprav pro tajná hlasování. Systém je především třeba rozšířit o entitu municipalit. Potřebná konfigurace a statický text z frontend aplikace se tím přesune do databáze a municipality bude možné spravovat v rámci systému bez dalších ručních zásahů.

5.2 Požadavky na rozšíření systému a cíloví uživatelé

Cílem systému pro vizualizace je poskytnout občanům informace o hlasování zastupitelstev municipalit, vzbudit tak větší zájem o komunální politiku a umožnit jim kontrolovat hlasování jimi volených zástupců. Jednání zastupitelstva jsou ze zákona veřejná, občané tak mají možnost se jich osobně zúčastnit, což však nelze považovat za jednoduché nebo praktické pro každého občana. Zákon dále dává občanům právo nahlédnout na úřadu do zápisu z jednání zastupitelstva, kde však musí být obsaženy pouze výsledky hlasování, ne konkrétní volby jednotlivých zastupitelů. Navíc opět není praktické muset úřad osobně navštívit či si od něj dokument vyžádat.

Problematika otevřených a otevřených vládních dat je však ve veřejném prostoru řešena již delší dobu a mnohé municipality se snaží svá data zveřejňovat. Zápisy proto bývají často dostupné na webových stránkách municipalit, dostupný může být i živý přenos a záznam z jednání. Existují i města, která mají na svých webových stránkách dostupné informace o jednotlivých hlasováních, např. Praha⁹ (viz [52]). Jednotky municipalit také publikují datové sady o hlasování svých zastupitelstev v NKOD (viz sekce 3.1), čímž umožňují další zpracování těchto dat libovolným zainteresovaným osobám. Dostupné informace o hlasování jsou však nejednotné z důvodu chybějící standardizace, jejich zveřejňování je také z velké části pouze důsledkem ochoty či dobré vůle dané municipality.

Cílem tohoto projektu je poskytnout systém, který jednotným způsobem vizualizuje data libovolné municipality a zároveň data poskytne ve formě datové sady se standardizovaným schématem. První cíl práce je tak zobecnit existující systém pro práci s více municipalitami a odstranit manuální definice konfiguračních souborů pro municipality v oddělených instancích systému. Dalším cílem je vytvořit rozšíření systému – administrační rozhraní, které nově umožní přímo v rámci systému zadávání a upravování hlasovacích dat, přidávání nových municipalit a jejich správu a správu uživatelů. Měla by se tak usnadnit práce s hlasovacími daty v systému a umožnit jeho použití pro více municipalit.

Zdrojem pro data v systému jsou data zveřejňovaná či poskytnutá municipalitami, které mohou být v různých formátech. Tato data se po publikování mění minimálně či vůbec. V případě strojově zpracovatelných formátů je možné data předzpracovat a vložit do systému ve formě obecného modelu. Podle použité úrovně detailnosti ve zdrojové datové sadě

⁹Výsledky hlasování Zastupitelstva hl. m. Prahy – https://www.praha.eu/jnp/cz/o_meste/primator_a_volene_organov/zastupitelstvo/vysledky_hlasovani/index.html

však nemusí být ve vložených datech obsaženy všechny relevantní informace, které mohou být dále doplňovány na základě informací dostupných např. ze zápisů či webových stránek municipality. Dále je třeba umožnit korekce případných chyb ve zdrojových datech, které mohou být odhaleny později. Systém proto musí umožnit editovat údaje o jednotlivých entitách.

Také je nutné zohlednit situace, kdy zdroj dat nelze nebo není praktické zpracovat strojově, např. jedná se o naskenované dokumenty nebo se liší použitý formát mezi jednotlivými zasedáními. V takovém případě je třeba umožnit data postupně vkládat ručně. Pro municipality s iniciálním prázdným stavem databáze musí být možné postupně vkládat jednotlivé entity jako jsou zastupitelé a politické subjekty a později je propojovat, např. vložením na zasedání.

Pro ruční vkládání dat se je primárně třeba zaměřit na efektivní zadávání individuálních hlasování, jelikož ta reprezentují naprostou většinu dat. Údaje týkající se celého zastupitelstva bude při běžném stavu nutné zadávat pouze každé 4 roky po volbách (až na případné změny, např. členů, nebo opravy). U hlasovacích dat lze předpokládat, že budou zadávána sekvenčně po proběhlých zasedáních. Editování by také mělo sloužit k případné opravě chyb po vložení jednotlivých zasedání.

Základní skupina uživatelů pro vizualizace zůstává stejná jako v případě systému pro město Brno, jedná se o občany, kteří mají zájem o komunální politiku. Nepřihlášeným uživatelům systém nabízí vizualizace hlasování pro ně relevantního zastupitelstva. Stále platí, že mohou mít různou úroveň technické zdatnosti, a měla by tak být zachována jednoduchost a přehlednost systému. Nově však tato skupina nezahrnuje pouze občany se zájmem o hlasování Zastupitelstva města Brna, ale i občany dalších municipalit, které mohou být potenciálně přidány.

Další skupinou jsou uživatelé, kteří využijí novou část systému pro vkládání a správu dat. Jedná se o užší skupinu uživatelů, kteří mají přímo zájem se podílet na doplňování či korekcích dat a případném vkládání zcela nových hlasovacích dat pro nové municipality. U těchto uživatelů se očekává vyšší úroveň technických znalostí a schopnost práce s webovým uživatelským rozhraním. Očekávaná platforma pro práci s administračním rozhraním je počítač. Zároveň vzhledem k povaze práce musí mít základní přehled o hlasování zastupitelstev potřebný pro vkládání a úpravu těchto dat. Do této kategorie uživatelů patří především další členové projektu, kterým bude umožněno pracovat s daty bez nutnosti přímých zásahů do databáze. Potenciálně by se také mohlo jednat např. o zástupce municipalit, případně i aktivní občany, kteří mají zájem o publikování dat ze své municipality.

Uživatelé vkládající data plní roli správců municipality, kteří mohou spravovat konkrétní municipality a její data. Zároveň by měla v systému existovat role centrálního správce, který má v systému všechna práva a může tak spravovat všechny municipality a jejich data, i ostatní uživatele.

Z předchozí analýzy vyplývají požadavky na existenci následujících rolí v systému: **nepřihlášený uživatel**, **uživatel** (přihlášený správce municipality) a **administrátor**. Nepřihlášený uživatel odpovídá uživateli v původním systému. Role uživatele a administrátora vyžadují autentizaci, jelikož mají v rámci systému editační práva.

5.3 Specifikace požadavků

Dále jsou uvedeny požadavky na úpravy systému, především na novou funkcionalitu administračního rozhraní. V následujícím výčtu se správou rozumí možnost přidávat, upravovat,

případně i mazat danou entitu. V uživatelském rozhraní administrační části by měly být dostupné následující akce:

- **správa municipalit**

Zahrnuje i možnost vytvořit novou municipalitu. Dále úpravu názvů a informačních textů specifických pro municipalitu zobrazovaných v rámci vizualizace, počtu zastupitelů, barvy atd.

- **správa zastupitelstev** (jednotlivých funkčních období)

- **správa politických subjektů**

- **správa členů** (zastupitelů)

- **správa zasedání**

- **správa hlasování** (hlasovacího bodu jako celku)

- **zadávaní a úprava hlasování jednotlivých členů**

Vzhledem k potenciálně velkému množství členů je třeba se zaměřit na efektivní zadávání jejich voleb, např. lze předpokládat, že zastupitelé za jeden politický subjekt budou hlasovat stejně, příp. některé body mohou být schváleny většinově, např. procedurální hlasování.

- **správa profilu uživatele**

- **správa uživatelů**

Uživatelům musí být umožněno získat možnost spravovat či přidat municipalitu, rozhodnutí o přiřazení práva náleží pouze administrátorovi.

Na systém jsou mimo uživatelských požadavků kladeny i další níže uvedené požadavky, a to především technické, které vyplývají z implementace původního systému, a dodatečné požadavky na celý projekt vizualizace hlasování municipalit:

- Jedná se o rozšíření existujícího systému, čímž jsou dány použité technologie.

- frontend: JavaScript a knihovna React

- backend: Python a framework Flask, relační databáze MariaDB

- Jednotlivé municipality by mělo být možné spravovat v rámci jednoho systému, tedy odstranit nutnost udržovat více instancí systému.

- Nová administrační část systému by neměla ovlivnit uživatele, kteří systém používají pouze v režimu pro čtení jako dosud.

- Uživatel by neměl být nutně vázán pouze na jednu municipalitu, jelikož může být žádoucí, aby jich spravoval více.

Uživatelské účty a autentizace

Systém bude nově umožňovat úpravu dat, tyto operace by však měly být přístupné jen oprávněným uživatelům. Přibývá proto požadavek na vytváření uživatelských účtů, přihlašování a autentizaci uživatelů a s tím spojená existence entity uživatele v rámci systému. V rámci uživatelského rozhraní by měly stránky či komponenty pro správu entit nebo odkazy na ně být dostupné pouze přihlášeným uživatelům s potřebnými právy. Koncové body REST API pro správu dat či přístup k údajům o uživatelích musí být dostupné pouze autentizovaným uživatelům s odpovídajícími právy.

Kapitola 6

Návrh

Obsahem této kapitoly je návrh rozšíření existujícího systému pro vizualizaci hlasování Zastupitelstva města Brna. Jsou zde popsány úpravy systému nutné pro realizaci správy municipalit a celková architektura rozšířeného systému s administračním rozhraním. Je představeno rozšíření současného datového modelu o nové entity a změny v práci s databází. Následně je popsáno rozšíření API o nové koncové body a rozšíření webového uživatelského rozhraní.

6.1 Architektura systému

Vychází se z architektury existujícího systému pro jednu municipalitu, který je implementován jako webová aplikace s dvouvrstvou architekturou (viz sekce 5.1). Systém bude rozšířen tak, aby správa municipalit byla řešena v rámci něj bez nutnosti vytváření zcela oddělených instancí systému pro každou individuální municipalitu.

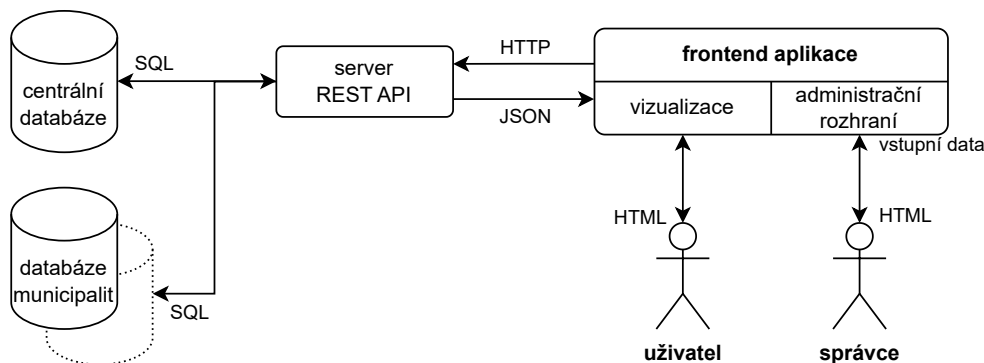
Pro účely rozšíření systému je nutné do systému přidat nové entity. Současný návrh databáze je sice obecný, ale slouží pouze pro uložení dat vztahujících se k jedné municipalitě (resp. zastupitelstvu municipality). Systém zároveň nepracuje s reprezentací municipality jako takové.

Rozšíření systému by mělo způsobit co nejmenší zásahy do existujícího systému, a tedy i databázového schématu, zároveň je vhodné zachovat oddělená data pro jednotlivé municipality. Databázový model proto bude rozšířen o novou **centrální databázi**, která bude obsahovat údaje o jednotlivých municipalitách. Schéma původní databáze zůstane zachováno, jeho použití ale bude zobecněno na **databáze municipality**, jednotlivé databáze obsahující údaje týkající se hlasování zastupitelstev konkrétních municipalit. Záznamy o každé municipalitě a její databázi jsou uloženy v centrální databázi. Architektura rozšířeného systému je zobrazena v diagramu 6.1.

Uživatelské rozhraní bude rozšířeno o nové administrační rozhraní přístupné pouze pro autentizované uživatele. Na rozdíl od vizualizační části systému budou tito uživatelé mít možnost použít systém nejenom pro čtení dat, ale nově i pro jejich vkládání a modifikaci. Se serverovou částí probíhá komunikace pomocí REST API, které bude rozšířeno a bude poskytovat novou funkcionalitu pro práci s municipalitami a pro účely administrace. Serverová část systému bude nově zohledňovat práci s více databázemi namísto jedné. Základní komponenty rozšířeného systému jsou tedy následující:

- **uživatelské rozhraní:** Implementováno jako jednostránková webová aplikace. Obsahuje existující vizualizace hlasování a oddělené administrační rozhraní.

- **serverová aplikace:** Poskytuje REST API sloužící pro komunikaci s klientskou částí systému. Na základě přijatých požadavků provádí interakce s databázemi a vrací odpovídající HTTP odpovědi a případně data ve formátu JSON.
- **centrální databáze a databáze municipalit:** Slouží pro uložení dat. Centrální databáze obsahuje údaje o municipalitách a uživateli systému. Záznamy o hlasování zastupitelstev jsou pro jednotlivé municipality uloženy v oddělených databázích.



Obrázek 6.1: Návrh architektury rozšířeného systému. Uživatelské rozhraní obsahuje oddělené administrační rozhraní pro správu dat. Na serverové straně je zobrazen databázový model pro práci s daty více municipalit.

6.2 Serverová část systému

Základní architektura serverové části a použité technologie zůstávají stejné jako v původním systému. Uživatelské rozhraní komunikuje s REST API, které je poskytováno aplikací implementovanou pomocí frameworku Flask, která zajišťuje práci s databází.

Pro práci s daty je ponechán stejný systém řízení báze dat jako v původním systému. Nově ale bude systém pracovat s několika databázemi namísto jedné. Schéma původní databáze bude s nutnými změnami dále používáno pro databáze municipalit. Navíc bude přidána centrální databáze s odlišným schématem.

centrální databáze Existuje pouze jedna instance centrální databáze v systému. Údaje pro připojení k této databázi jsou dané konfigurací systému, existence centrální databáze je nutná pro fungování systému.

databáze municipality Obsahuje data o hlasování jedné municipality. Údaje týkající se municipality samotné včetně identifikace této databáze jsou obsaženy v centrální databázi. Může jich existovat libovolné množství, nicméně pro běžný provoz systému by měla existovat alespoň jedna, tzn. systém je z hlediska vizualizace dat pro uživatele ekvivalentní původnímu systému. Název databáze je složen z prefixu **zastupitelstvo** odděleného od názvu municipality podtržítkem, používají se pouze malá písmena bez diakritiky, např. pro Brno je použit název **zastupitelstvo_brno**.

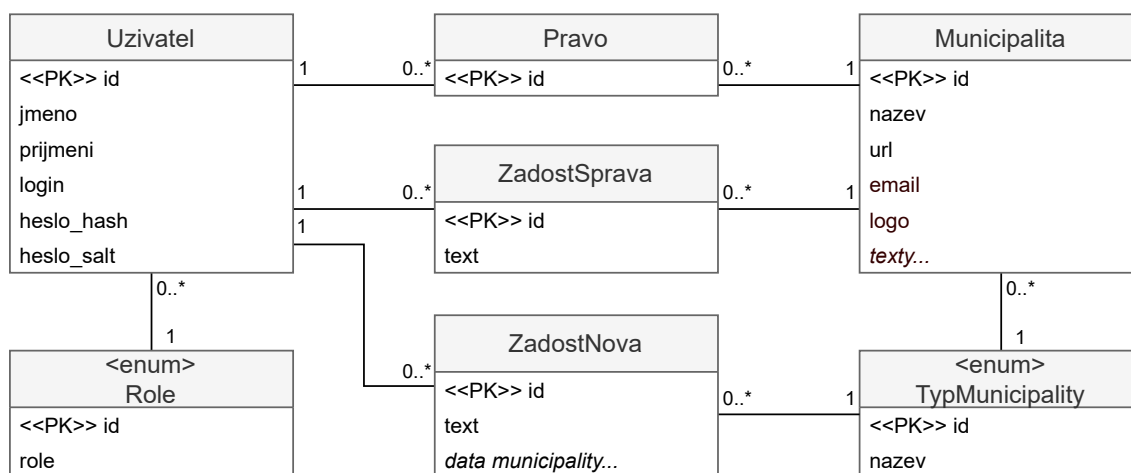
Při spuštění systému musí existovat pouze centrální databáze, aplikace se na základě konfigurace jako první připojí k této databázi. V případě přijetí požadavku na koncový

bod, který vyžaduje načtení dat pro konkrétní municipalitu, se na základě údajů v centrální databázi v dalším kroku připojí k databázi pro danou municipalitu. Vytvoření nové municipality v systému tedy odpovídá přidání záznamu s daty o municipalitě do centrální databáze a vytvoření nové databáze pro danou municipalitu.

6.2.1 Datový model

Dle výsledků analýzy je nutné systém rozšířit minimálně o entitu municipality a uživatele. Dále je třeba připojit entitu reprezentující právo uživatele na správu municipality a žádosti o možnost spravovat municipalitu. Jelikož se ve všech případech jedná o entity, které se vztahují k municipalitám nebo k systému jako celku, budou data všech nových entit uložena v centrální databázi. Konceptuální model pro nové entity je k vidění v diagramu 6.2. Navíc ke zmíněným entitám jsou připojeny číselníky pro role uživatelů a typy municipalit. Význam hlavních entit je následující:

- **Municipalita:** Údaje o jedné municipalitě, především název a typ dané municipality, kontaktní e-mail a všechny informační texty v uživatelském rozhraní týkající se municipality či jejího zastupitelstva.
- **Uzivatel:** Reprezentuje uživatele systému, jeho přihlašovací údaje a roli.
- **Pravo:** Právo uživatele, který není administrátorem, na správu konkrétní municipality.
- **ZadostSprava:** Žádost uživatele o právo spravovat municipalitu.
- **ZadostNova:** Žádost uživatele o přidání nové municipality.



Obrázek 6.2: Konceptuální model entit v centrální databázi. Pro zjednodušení nejsou vy-psány všechny textové atributy entity **Municipalita**, které jsou reprezentovány pouze atri-butem **texty**. U entity **ZadostNova** jsou obdobně reprezentovány atributy týkající se mu-nicipality.

Kromě přidání nových entit je třeba provést některé úpravy v existujícím datovém modelu (viz diagram 5.3), aby bylo možné reprezentovat nově vkládaná data a lépe s nimi pracovat. Jako první je třeba modifikovat entitu **členství** člena v politickém subjektu a funkčním období (entita **PrislusiK**). V původním systému se nepočítá s existencí členství bez vazby na zasedání, tzn. funkční období vždy obsahuje zasedání, a začátek a konec

členství je odvozovaný od uložených zasedání. Při postupném vkládání dat do systému je žádoucí vložit členy pro funkční období a až následně vkládat zasedání, na kterých může potenciálně docházet ke změnám členství v politických subjektech. Pro tyto účely bude upravena tabulka příslušností a budou povoleny hodnoty NULL pro datum začátku i konce. Význam těchto hodnot je, že členství není omezeno. Konkrétní datum je nastaveno pouze v případě, že člen získal, či ztratil mandát v průběhu funkčního období, nebo pokud změnil politickou stranu. Výchozím stavem je hodnota NULL pro začátek i konec, což odpovídá členství po celé funkční období.

Dále budou upraveny **aliasy**, které reprezentují změnu jména člena v systému. Tato entita vznikla jako korekce pro stav, kdy po změně jména člena při zpracování dat dojde k jeho identifikování a uložení do databáze jako nového člena, i když by měl být systémem chápán jako jediná osoba. Alias proto funguje jako propojení dvou členů, z nichž každý může mít vlastní členství, hlasování atd. Jeden z členů pro alias je původní a druhý nový, se kterým se dále pracuje. Toto řešení má za následek, že většina koncových bodů API musí kontrolovat, zda pro členy neexistuje alias a následně všechny relevantní záznamy spojit pod aktivního člena. Tato funkcionalita je navíc na několika místech omezena pouze pro jeden alias a tedy jednu změnu jména, což je stav, který odpovídá datům pro Brno.

Pro účely zjednodušení metod API a pro přidání více aliasů bude současné řešení nahrazeno novou entitou aliasu reprezentující nové jméno (křestní jméno i příjmení), která se bude vázat pouze k jednomu členovi. Tento alias bude mít platnost danou daty začátku a konce, na rozdíl od existujících aliasů, které se váží ke konkrétnímu funkčnímu období, kdy ke změně jména došlo. Bude tak možné pouze načíst platné aliasy pro zvolené datum, stejně jako seznam všech minulých aliasů s dobou platnosti. Data začátku a konce platnosti lze odvodit z dat zasedání, kdy došlo ke změně. Existující data se upraví a ponechá se pouze jedna entita aktivního člena, čímž budou odstraněny problémy se slučováním dat pro více členů reprezentujících jedinou fyzickou osobu člena zastupitelstva se změněným jménem.

Loga municipalit jsou v současném systému dostupná jako statické soubory přístupné přes webové uživatelské rozhraní, jedná se tak o součást výchozí konfigurace systému, kterou může změnit pouze správce nahrazením daného souboru. Pro účely přidání nových municipalit a jejich správy je třeba umožnit soubory přidávat a nahrazovat pomocí API. Cesta k logu či jeho identifikátor je atributem municipality.

REST API

Do existujících koncových bodů API pro vizualizace by měly být provedeny pouze nutné zásahy potřebné pro zobecnění systému pro práci s více municipalitami. Aplikace bude dále rozšířena o novou funkcionalitu potřebnou pro administrační rozhraní a nové koncové body pro práci s municipalitami a administraci dat.

Úpravy koncových bodů pro vizualizaci

Zobecnění systému vyžaduje úpravu všech koncových bodů API, které přistupují k datům konkrétní municipality. Vzhledem k tomu, že původní systém používal pouze jednu databázi municipality, se jedná o všechny koncové body přistupující do databáze, což jsou všechny kromě testovacího koncového bodu.

Do URL cesty pro každý koncový bod je třeba přidat další parametr identifikátoru municipality, pro kterou mají být vrácena data. Úpravu lze demonstrovat na dříve představeném koncovém bodu pro načtení zastupitelstev municipality:

```
/flask/councils  
/flask/<municipality_id>/councils
```

Na prvním řádku je původní cesta, na druhém upravená. Za prefixem cesty nově následuje identifikátor municipality, jinak zůstávají cesty nezměněny. Ve špičatých závorkách není uveden identifikátor typu, jelikož se používá výchozí datový typ řetězec. Textový identifikátor je zvolen, protože je vhodný i pro použití v cestách v rámci uživatelského rozhraní, ze kterých tak půjde snadno rozpoznat zvolenou municipalitu.

V rámci implementace koncového bodu API je nově nutné zvolit pro vykonávaný požadavek správnou databázi municipality. Pro tyto účely se na základě identifikátoru municipality dle údajů v centrální databázi zvolí správná databáze municipality, se kterou metoda dále pracuje. U jednotlivých metod tak dojde pouze ke změně v úvodním připojení k databázi, které nově vede k připojení k dynamicky zvolené databázi namísto jediné nakonfigurované při spuštění aplikace. Nicméně poté se s databází pracuje identicky jako dosud a do implementace tak není třeba více zasahovat.

Vizualizační API je dále třeba rozšířit o koncové body pro načtení seznamu municipalit a načtení konkrétní municipality. Formát cest pro nové koncové body by měl odpovídat již existujícím.

Administrační část

Administrační část bude implementována jako rozšiřující součást aplikace oddělená do vlastního modulu (*subpackage*). Pro účely práce s entitami v rámci administračního rozhraní, především jejich vkládání a úpravy, bude zaveden objektový přístup pro práci s databází. Namísto současného přístupu používajícího parametrizované SQL dotazy bude pro nové koncové body práce s daty probíhat pomocí objektově relačního mapování (ORM).

V rámci aplikace bude definováno schéma databáze pomocí metod poskytovaných ORM. Práce s databázovými entitami a interakce s databází bude probíhat skrze vytvořené třídy či jejich instance. Jména tříd reprezentujících jednotlivé databázové entity budou odpovídat použitým názvům v diagramu entit 5.3. Spolu s ORM by však mělo být stále možné použít i obyčejné parametrizované SQL dotazy, např. pro účely analytických dotazů, použití SQL také zůstává zachováno u původních koncových bodů.

Dále je třeba pro administrační rozhraní zavést uživatelské účty a autentizaci. Správa uživatelů je přímo součástí rozšíření systému, k uložení údajů o uživateli slouží výše představená entita uživatele. Pro přenos stavu autentizace mezi klientem a serverem lze použít JWT (*JSON Web Tokens*), standardizovaný přístup pro přenos sady deklarovaných vlastností (*claims*) ve formě řetězce znaků – tokenu, který rovněž obsahuje kryptografický podpis bránící podvržení. Token je uživateli vydán koncovým bodem API pro přihlášení po zadání správných přihlašovacích údajů a je následně posílán spolu s dalšími požadavky, které vyžadují autentizaci. Ukládání hesel do databáze a vytváření tokenů by mělo odpovídat aktuálním bezpečnostním doporučením, především ukládání hesel pouze ve formě kryptografického hashe a použití dostatečně silných algoritmů.

API musí nově poskytovat funkcionalitu potřebnou pro načtení municipalit a dalších nových entit, práci s uživateli a správou dat. Pro tyto účely musí být API rozšířeno o nové koncové body a metody. Jedná se převážně o operace typu CRUD (*Create, Read, Update, Delete* – Vytvořit, Číst, Editovat, Smazat) pro jednotlivé entity. Pro entity původního systému jsou již k dispozici koncové body pro jejich načtení, které však ve většině případů vracejí předzpracovaná data sloužící pro vizualizaci. Pro správu dat jsou tak nevhodné a bude třeba doplnit i metody pro načítání dat.

Pro administraci je třeba přidat zcela nové koncové body pro entity v systému, jejichž data je třeba spravovat. Na rozdíl od existujících koncových bodů pro načítání dat nové koncové body podporují několik metod odlišujících se metodou HTTP požadavku dle požadované operace. Jednotlivé operace budou implementovány pomocí odpovídajících metod tříd, které dědí od třídy `Resource` pro definici koncových bodů z balíčku `Flask-RESTful`.

Vzhledem k tomu, že se jedná o zcela nové koncové body, tak pro každý nový koncový bod bude vytvořena nová třída. Název tříd by měl popisovat entitu, se kterou se pracuje, a zda se jedná o seznam entit či konkrétní entitu (tzn. cesta obsahuje identifikátor entity). Koncové body je tak možné dělit na dva základní typy podle toho, s jakým počtem zdrojů interagují. Operace upravení dat se budou vztahovat pouze ke koncovým bodům pro jednu entitu.

Z hlediska dělení metod dle parametrů je naprostá většina parametrických, kvůli nutnosti zahrnout do cest identifikátor municipality. Významná část koncových bodů bude nově také pracovat s tělem požadavku (požadavky typu `POST`, `PUT` a `PATCH`). Formát pro těla požadavků i návratové hodnoty je `JSON`.

Požadavky, které mění data v systému nebo přistupují k datům uživatele, musí obsahovat autentizaci uživatele, který musí mít dostatečná práva pro danou operaci, jinak bude požadavek odmítnut. Při úpravě entit je rovněž třeba zohlednit existenci statistik (viz diagram 5.3), které je v případě změny relevantních dat potřeba přepočítat.

Nové koncové body API

Následuje přehled koncových bodů API, které je třeba přidat to systému. Seznam všech koncových bodů včetně cest je součástí přílohy A. Naprostá většina nových koncových bodů slouží pro účely správy dat. Tyto koncové body typicky podporují více HTTP metod podle potřebných operací. Základní metody pro práci s entitami jsou `GET` pro načtení dat, `POST` pro přidání nové entity a `PUT` pro upravení existující entity. Pro některé entity může být případně k dispozici i metoda `DELETE` pro smazání.

Parametry pro metody na přidání a upravení entit jsou dány existujícím schématem databáze municipality nebo navrženým schématem centrální databáze. Stejně tak povinnost parametru je dána na základě odpovídajícího sloupce ve schématu. Pro koncové body pro administraci část se musí uživatel autentizovat a mít potřebná přístupová práva, tj. roli administrátora nebo právo spravovat municipalitu.

Koncový bod pro přidání **municipality** je dostupný pouze pro administrátora. Vyžaduje zadání údajů o municipalitě i zastupitelstvu. Přidání municipality zahrnuje vytvoření záznamu o municipalitě v centrální databázi a vytvoření nové databáze pro zastupitelstvo municipality identifikované zvoleným textovým identifikátorem municipality. Úprava municipality a loga municipality je dostupná všem uživatelům s právem spravovat municipalitu.

Koncové body pro správu i načítání dat jednotlivých entit hlasování zastupitelstva vyžadují právo spravovat odpovídající municipalitu. Pro každou entitu je dostupné načtení seznamu, přidání nového záznamu a editování. V případě potřeby může být dostupné i načtení konkrétní entity (např. pro načtení dat při editování na samostatné stránce) a mazání. Smazat entitu je možné pouze, pokud nejsou ovlivněna hlasování, např. nelze smazat člena, pro kterého jsou uložena hlasování.

Součástí přidání **zasedání** je výběr členů pro dané zasedání. Pro každého člena je uveden politický subjekt, do kterého člen v době zasedání patřil. Seznam členů musí obsahovat všechny členy, kteří dle členství byli členy ke zvolenému datu konání zasedání. Pokud je třeba některého ze členů odebrat, musí být explicitně označen za odebraného. V takovém

případě se jeho členství ukončí k datu předchozího zasedání (identické chování jako při vkládání předzpracovaných dat). Také je možné přidat nové členy, opět se zvoleným politickým subjektem. Při vytvoření zasedání je třeba provést korekci členství, tzn. zkrácení, či vytvoření nových členství pro nové členy nebo při změně politického subjektu. Pro každého člena je také třeba uložit statistiku účasti na zasedání. Výběr členů by měl být možný také již při vytváření funkčního období. V tomto případě lze vybrat libovolné členy, kterým je přidáno členství v daném politickém subjektu po celé funkční období. Členství poté může být případně změněno při přidávání zasedání.

Při přidávání nebo úpravě **hlasování** se hlasování členů načítají podle data zasedání, ke kterému hlasování patří. Pro každého člena se zadává hlasovací možnost i příznak prezenního hlasování, který by však měl být ve výchozím stavu předvyplněn.

Registrace a přihlášení jsou dostupné všem uživatelům, ale noví uživatelé nemohou spravovat žádné municipality. Uživatelé mohou získat právo spravovat municipality podáním žádosti. Také mohou požádat o přidání zcela nové municipality, součástí této žádosti jsou všechny povinné parametry pro vytvoření municipality.

Administrátor má přístup k seznamu **žádostí**, které může schválit či zamítnout. Schválením žádosti se uživateli přidá právo spravovat danou municipality. Schválením žádosti o přidání municipality dojde k vytvoření municipality na základě údajů z požadavku stejným způsobem jako při použití koncového bodu pro vytvoření municipality.

6.3 Uživatelské rozhraní

Obdobně jako v případě API je v uživatelském rozhraní třeba provést úpravy související se zobecněním systému pro více municipalit. Dalším krokem je uživatelské rozhraní rozšířit o novou administrační část.

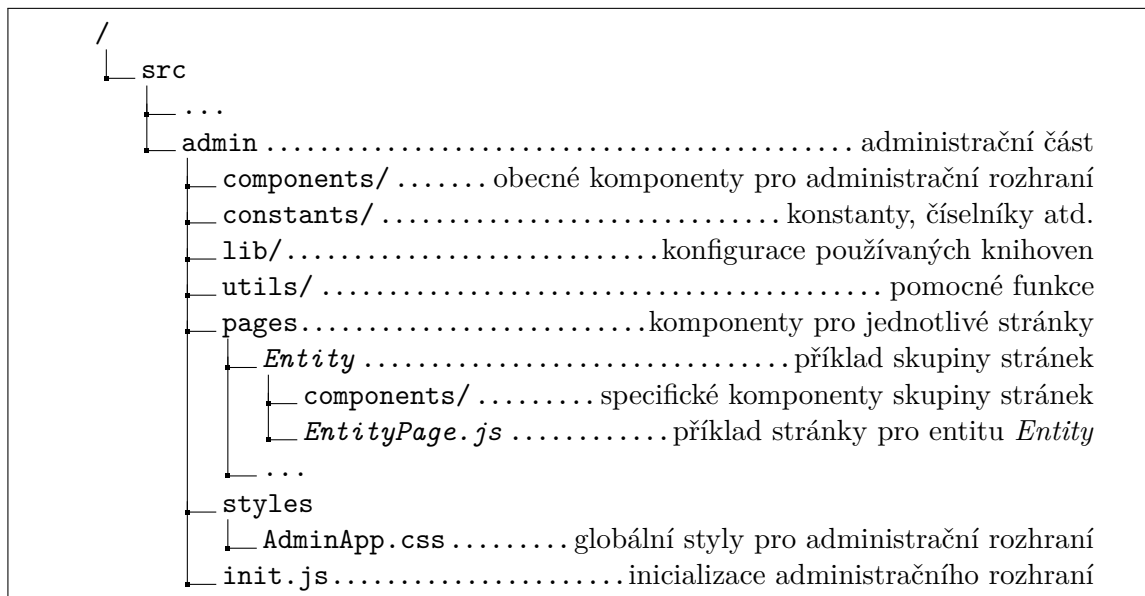
Do aplikace bude přidána nová úvodní stránka výběru municipality namísto současného výběru zastupitelstva. Použije se nyní oddělená vstupní stránka (*landing page*), která bude integrována do aplikace. Stránku bude třeba upravit nahrazením statických údajů o municipalitách načtením seznamu municipalit z API. Po výběru municipality se přejde na výběr zastupitelstva, který stále může sloužit jako vstupní bod do systému pro konkrétní municipality.

Parametr textového identifikátoru municipality se přidá do definice cest pro jednotlivé stránky ve směrovači aplikace, stejně jako je tomu u ostatních entit v systému, např. zastupitelstev. Následně bude třeba provést korekci odkazů a navigace v rámci aplikace. Nový parametr bude také použit pro aktualizaci cest u existujících požadavků na koncové body API, které nově vyžadují výběr municipality.

6.3.1 Administrační část

Stejně jako v případě serverové aplikace bude administrační část v rámci existující aplikace vytvořena jako oddělený modul ve vlastní podsložce. Stránky administračního rozhraní budou vycházet z existujícího stylu pro vizualizace, rovněž budou využívat závislosti z existujícího systému. Především knihovnu **React Router** pro navigaci a definice cest, **axios** pro požadavky na API a **PrimeReact** pro základní komponenty uživatelského rozhraní.

Adresářová struktura pro administrační část je k vidění na obrázku 6.3. Organizace kódu v rámci podsložky pro administrační rozhraní vychází ze členění zavedeného pro celou aplikaci.



Obrázek 6.3: Adresářová struktura administrační části uživatelského rozhraní.

Navíc je zavedeno členění do podsložek pro jednotlivé stránky příslušící k jedné funkcionalitě či entitě, které mohou mít komponenty specifické pouze pro tyto stránky. Pro účely správy dat bude třeba používat formuláře pro jejich zadávání, kterých bude větší množství a zároveň budou používány jen na jedné stránce nebo malém počtu stránek. Používání pouze globálních komponent by tak pro členění kódu bylo nepraktické a nepřehledné. Toto členění je inspirováno členěním dle **features** z projektu Bulletproof React¹, ze kterého je také převzata terminologie pro další složky jako např. **lib** pro konfigurace knihoven.

Je opět žádoucí, aby se omezily zásahy do existujícího systému, proto je nová administrační část oddělena. Vstupním bodem z hlediska kódu je soubor **init.js**, který rozšíří směřování v aplikaci pomocí **React Router** o cesty k novým stránkám. Zároveň může obsahovat další inicializaci specifickou pro administrační rozhraní.

Pro definici stylů aplikace se používá jediný soubor ve formátu **CSS** s globálními styly. Pro administrační část se obdobně použije jediný soubor se styly, které se však budou aplikovat pouze pro administrační část. V případě, že se nebude jednat o styl pro specifickou komponentu, je třeba styly omezit např. selektorem třídy aplikovaným pouze pro šablonu stránek administračního rozhraní.

Administrační rozhraní primárně slouží ke správě dat, naprostá většina stránek proto obsahuje seznamy základních entit systému nebo formuláře pro jejich úpravu. Pro zobrazení dat budou použity tabulky se základními atributy jednotlivých entit. Výběrem entity v tabulce bude možné přejít na podřazené entity nebo provádět akce s entitou jako je úprava nebo mazání. Pro vkládání či úpravu dat budou k dispozici formuláře se vstupy odpovídajícími parametrům relevantního API požadavku a tedy atributům entity v databázi. Formuláře budou dle komplexnosti dostupné buď na oddělené stránce, nebo přímo v přehledu pro danou entitu.

Pro potřebné prvky uživatelského rozhraní budou využity komponenty z komponentové knihovny **PrimeReact**². Především tabulka **DataTable** pro zobrazování dat a jednotlivé

¹Projekt Bulletproof React – <https://github.com/alan2207/bulletproof-react>

²Knihovna PrimeReact – <https://www.primefaces.org/primereact-v8/>

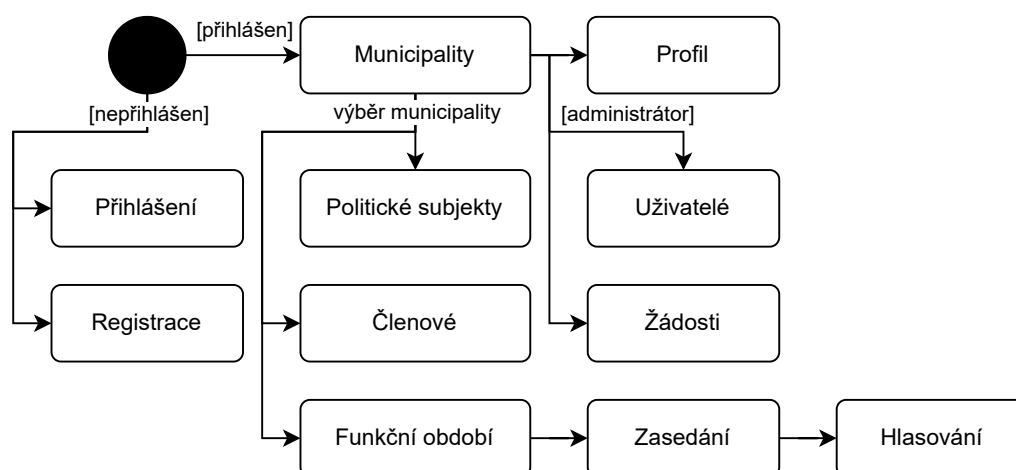
komponenty vstupů (např. textový vstup `InputText`) pro zadávání dat. Zobrazení formulářů přímo na přehledové stránce je možné pomocí komponenty dialogového okna `Dialog`. Knihovna umožňuje poměrně širokou konfiguraci komponent, které tak mohou být přizpůsobeny potřebám jednotlivých vstupů.

Při práci se systémem je třeba uživatele při provádění požadavků informovat, když systém provádí akci a je třeba vyčkat. Stejně tak při načítání dat, jako je tomu v existující aplikaci. V případě selhání požadavků je uživateli třeba v uživatelském rozhraní zobrazit chybu, použije se text chyby vrácený z API, pokud je k dispozici, jinak se zobrazí výchozí chybový text. Pro tyto účely je možné použít komponenty z knihovny `PrimeReact` pro zobrazování zpráv – `Messages` nebo `Toast`.

Formuláře zajistí validaci vstupních dat a zobrazení případné chyby. V případě selhání požadavku po odeslání formuláře je tato informace uživateli zobrazena a data ve formuláři by měla zůstat nezměněna, aby je uživatel nemusel zadávat znovu, dává-li to v daném případě smysl.

Stránky a struktura aplikace

Základní přehled stránek a navigace mezi nimi je zobrazen v diagramu 6.4. Nepřihlášený uživatel jako první při přístupu do administračního rozhraní musí provést přihlášení, případně registraci. Pro přihlášeného uživatele je vstupním bodem seznam municipalit, ze kterého může navigovat na další stránky. Běžné stránky budou mít stejné základní rozložení jako ve vizualizační části, tedy menu v záhlaví, zápatí a proměnlivý obsah stránky.



Obrázek 6.4: Návrh stránek administrační části systému. Pro zjednodušení jsou všechny stránky týkající se stejné entity reprezentovány pouze jednou stránkou, ve skutečnosti se může jednat o více stránek, např. přehled a detail, na který je možné z přehledu přejít. Také nejsou uvedeny možnosti navigace pomocí hlavního menu a je vynechána chybová stránka.

Obsah menu záleží na roli uživatele, kdy uživatel má k dispozici pouze práci s municipalitami, které může spravovat, a jednotlivé entity hlasování zastupitelstev těchto municipalit. Administrátor má navíc k dispozici přehled uživatelů a žádostí. Pro každého přihlášeného uživatele je v rámci menu také k dispozici správa profilu a odhlášení. Pro nepřihlášeného uživatele nebo uživatele bez potřebných práv by nemělo v UI být možné navigovat na stránky, na které nemají přístup.

Navigace v uživatelském rozhraní odpovídá hierarchii entit, které mohou být upravovány. Po výběru municipality jsou k dispozici entity příslušící přímo k municipalitě – členové, politické subjekty a funkční období. Poté je výběrem funkčního období možné přistoupit k zasedáním a ze zasedání k hlasováním. Úzce související entity, které nemají význam samostatně nebo je nevhodné je upravovat odděleně, jsou zobrazovány či upravovány spolu s „rodičovskou“ entitou, jedná se především o hlasy členů, které jsou součástí správy hlasování. Dále se jedná např. o aliasy členů nebo odkaz na záznam zasedání.

Jak již bylo popsáno, z hlediska základního dělení stránky obsahují buď tabulkové přehledy, nebo formuláře, případně kombinaci obojího, viz dále. Formuláře mohou sloužit k přidávání nebo editování entit, je-li to možné měl by pro oba případy být použit formulář se stejným rozložením a vzhledem, který pouze v případě editování bude předvyplněn daty dané entity.

Následuje popis jednotlivých stránek administračního rozhraní a dalších hlavních komponent na nich dostupných. Kromě stránek přihlášení a registrace všechny stránky používají rozložení s menu, které lze použít k navigaci mezi stránkami. Vzhledem k celkovému množství stránek jsou návrhy uživatelského rozhraní ukázány jen pro vybrané stránky, které slouží pro ilustraci nebo se jedná o složitější stránky.

registrace Formulář pro registraci pro nepřihlášené uživatele.

přihlášení Formulář pro přihlášení, vstupní bod pro přihlášené uživatele.

profil uživatele Zobrazuje údaje uživatele, umožňuje upravit jméno a změnit heslo.

přehled municipalit Vstupní bod do administrační části pro přihlášené uživatele. Obsahuje tabulku municipalit, která umožňuje výběr municipality nebo přechod na stránku pro upravení municipality, pokud má uživatel potřebná práva. Pro administrátora také umožňuje vytváření nových municipalit a jejich mazání. Návrh stránky je k vidění na obrázku 6.5. Pro uživatele, kteří nemají práva, umožňuje založit žádost o spravování vybrané municipality pomocí dialogového okna.

 Municipality
 Uživatelé
 Žádosti


Municipality

Přidat

Název	Typ	Textové ID	Email		
Brno	Statutární město	brno	email@brno.cz		
Most	Statutární město	most	email@most.cz		

Obrázek 6.5: Návrh přehledu municipalit. Ukazuje základní rozložení stránky s menu v záhlaví a hlavním obsahem stránky. Menu obsahuje odkazy na další stránky a ikonu pro zobrazení menu pro přihlášeného uživatele na pravé straně.

přidat municipalitu, upravit municipalitu Formulář pro vytvoření nové municipality nebo úpravu existující. Součástí je výběr loga municipality a zadání všech atributů pro zastupitelstvo.

přehled politických subjektů Tabulka politických subjektů zvolené municipality, přímo na stránce je možné politické subjekty přidávat, upravovat a mazat. Pro přidávání a úpravy slouží dialogové okno s formulářem.

přehled členů Tabulka členů zvolené municipality. Umožňuje přidání členů pomocí dialogového okna, dále umožňuje mazání a obsahuje odkazy na detail členů.

detail/upravit člena Obsahuje formulář pro upravení údajů člena a dále tabulku členství a tabulku aliasů, viz obrázek 6.6. Aliasy je v tabulce možné mazat a přidávat nové pomocí dialogového okna

BRNO
Municipality
Uživatelé
Žádosti
Subjekty
Členové
Funkční období

Upravit členku Jana Nová

Jméno

Jana

Příjmení

Nová

Pohlaví

☐ Muž
☒ Žena

Upravit

Příslušnosti

Pořadí zastupitelstva	Politický subjekt	Datum od	Datum do
8	Subjekt A	20. 11. 2018	27. 10. 2020
8	Subjekt B	28. 10. 2020	06. 09. 2022
9	Subjekt B	20. 10. 2022	12. 12. 2023

Alias

Přidat alias

Jméno	Příjmení	Datum od	Datum do	
Jana	Původní	-	5. 7. 2021	✗

Obrázek 6.6: Návrh stránky detailu člena/členky. Obsah stránky je složen ze 3 hlavních částí – formuláře pro úpravu údajů o členovi/členke, tabulky členství a tabulky aliasů.

přehled funkčních období Tabulka funkčních období zastupitelstva zvolené municipality s odkazem na vložení nového funkčního období a na detaily.

přidat funkční období Formulář pro vložení funkčního období. Součástí je výběr členů a tabulka zvolených členů, kteří mají být přidáni do funkčního období. Na obrázku 6.7 je ukázán návrh stránky se zvolenými členy.

detail/upravit funkční období Obsahuje formulář pro upravení údajů o funkčním období a tabulku všech členů s členstvími v daném funkčním období.

přehled zasedání Tabulka zasedání pro zvolené funkční období, obsahuje odkazy na přidání a detaily zasedání.

detail/upravit zasedání Obsahuje formulář pro úpravu údajů o zasedání a tabulku členů zasedání.

Přidat funkční období

Začátek 
Konec 
Koalice
Název funkce lídra
Lídr

Členové

Přidat člena

ID	Jméno	Příjmení	Politický subjekt	
10	Člen	První	Strana A	✗
45	Člen	Druhý	Strana A	✗
7	Člen	Třetí	Strana C	✗

Přidat

Obrázek 6.7: Návrh stránky pro přidání funkčního období. Součástí formuláře jsou vstupy pro údaje o funkčním období a tabulka členů. Je možné přidávat nové členy výběrem ze členů municipality, v tabulce je poté členům nastaven politický subjekt, jehož byli součástí.

přehled hlasování Tabulka hlasování pro zvolené zasedání. Obsahuje odkazy na přidání, upravení a smazání hlasování.

přidat hlasování, upravit hlasování Obě stránky používají stejný formulář, který obsahuje vstupy pro údaje o hlasování a další komponentu pro vkládání hlasů členů. Na obrázku 6.8 je příklad pro upravení hlasování, kde jsou pro vkládání hlasování členů k dispozici karty po jednotlivých stranách. Pro účely efektivnějšího vkládání hlasování je alternativně možné zadávat hlasy členů v tabulce, která obsahuje sloupce pro jednotlivé hlasovací možnosti.

přehled žádostí Přehled žádostí uživatelů dostupný pro administrátora, součástí jsou tlačítka či dialogy pro potvrzení či zamítnutí žádostí.

přehled uživatelů Přehled všech uživatelů systému a jejich rolí pro administrátora.

BRNO
Municipality
Uživatelé
Žádosti
Subjekty
Členové
Funkční období
Zasedání
Hlasování

Upravit hlasování č. 1

Zasedání

Číslo

Čas

Výsledek

Z8/01

1

08:38:30

přijato

Předmět

Validní

Procedurální

Tajné

Bod č. 3 - ověřovatelé zápisu Určení ověřovatelů zápisu a schválení programu ustavujícího zasedání Zastupitelstva města Brna

☒

☐

☐

Politický subjekt A (8)

Hromadná volba

Člen Zastupitelstva	ano	☒ prezenčně	Člen Zastupitelstva	ano	☒ prezenčně
Člen Zastupitelstva	ano	☒ prezenčně	Člen Zastupitelstva	ano	☒ prezenčně
	ne				
	zdržel se				
	...				

Politický su

Hromadná volba

Upravit

Obrázek 6.8: Návrh stránky pro úpravu hlasování. Ukázkové hlasování má členy ze dvou politických subjektů, karta pro jeden je z nich je minimalizována. Dále je zobrazeno menu pro výběr hlasovací možnosti jednoho ze členů.

Kapitola 7

Implementace

Implementace vychází z návrhu řešení představeného v kapitole 6. Je rozšiřován existující systém popsáný v sekci 5.1, čímž jsou dány základní použité technologie. Dále je popsána konkrétní realizace řešení, provedené změny a nově přidané technologie.

Součástí řešení jsou i úpravy existujícího systému, především pro práci s více municipalitami. Většinou se však jednalo o menší zásahy, které nebyly hlavní částí práce, tyto zdrojové soubory původního systému proto nejsou zahrnuty v odevzdaném archivu.

Pro udržení jednotného stylu zdrojového kódu v rámci celého systému bylo dále zavedeno používání automatických formátovacích nástrojů. Zdrojový kód uživatelského rozhraní se nově formátuje pomocí nástroje `prettier`¹ a zdrojový kód v jazyce Python v serverové části pomocí nástroje `autopep8`².

7.1 Serverová část systému

Serverová část systému pro komunikaci s uživatelským rozhraním poskytuje REST API implementované jako aplikace ve frameworku Flask. Implementace je umístěna v balíčku `api` a vstupním bodem aplikace je soubor `app.py`. Data jsou uložena v relačních databázích, se kterými aplikace komunikuje. Aplikace byla upravena pro umožnění práce s více municipalitami, bylo zavedeno používání více databází municipalit a přidána centrální databáze. Pro práci s databázemi byla přidána možnost použití ORM. Aplikace byla dále rozšířena o novou administrační část.

7.1.1 Generalizace pro více municipalit

Prvním krokem bylo provedení generalizace existujícího systému. Generalizací jsou míněny úpravy existujícího systému za účelem umožnění práce s více obecnými municipalitami. Pro uložení údajů o municipalitách slouží nově přidaná centrální databáze se schématem vytvořeným dle návrhu. Údaje o municipalitách byly přesunuty do odpovídajících tabulek databáze z konfiguračních souborů používaných v uživatelském rozhraní. K dispozici je inicializační skript pro vytvoření schématu centrální databáze a také skript pro inicializaci číselníků a vložení údajů o municipalitách (původně umístěných v konfiguračních souborech).

Konfigurace pro připojení k databázi je v existující aplikaci nastavena pomocí proměnných prostředí a dosud sloužila pro připojení ke specifické databázi municipality. Konfi-

¹Formátovač `prettier` – <https://prettier.io/>

²Formátovač `autopep8` – <https://pypi.org/project/autopep8/>

gurance byla převzata a nově slouží pro připojení k centrální databázi. Tyto úpravy byly provedeny v existující třídě `DatabaseHandler` sloužící pro komunikaci s databází. Připojení k databázi je vytvořeno přímo v konstruktoru, v původním řešení je proto připojení k jediné výchozí databázi a zahájení komunikace zajištěno vytvořením nové instance třídy. Vzhledem k existenci více databází byly přidány nové statické metody pro připojení k centrální databázi nebo k jedné z databází municipalit na základě parametru, které vytvoří instanci s odpovídající konfigurací.

Do definice cest existujících koncových bodů, které načítají data konkrétní municipality, byl přidán nový parametr identifikátoru municipality. V implementaci metody `GET` každého z koncových bodů bylo nahrazeno výchozí připojení k databázi pomocí vytvoření objektu `DatabaseHandler` za použití nové metody pro připojení k databázi municipality. Výběr konkrétní databáze je proveden na základě identifikátoru municipality získaného z parametru v cestě. Dále byla třída rozšířena o možnost použít kontextový manažer (*context manager*), který zajistí automatické ukončení spojení s databází při opuštění kontextu, což bylo dosud prováděno u každého koncového bodu explicitně.

Byly přidány nové koncové body pro načítání dat municipalit, které pracují přímo s centrální databází. Dále byly provedeny dílčí úpravy koncového bodu pro vytvoření datové sady, kde se nově název municipality použitý ve jménu souboru a informace o municipalitě v něm obsažené načítají z centrální databáze.

Chyby při připojování k databázi, typicky při výběru neexistujícího identifikátoru municipality, způsobí výjimku, která je zpracována centrálně a je vrácena chybová odpověď. Třída `Api` z balíčku `Flask-RESTful` používaná pro definici REST API umožňuje upravit zpracování chyb pouze nastavením textu chybové odpovědi pro HTTP chyby, resp. chybové kódy. Pro účely vracení chybových odpovědí ve vlastním formátu a použití libovolných výjimek byla vytvořena třída `ExtendedApi`, která umožňuje přidat vlastní funkce pro řešení výjimek (*error handlers*). Při zpracování výjimky se použije registrovaný chybový handler podle typu výjimky, pokud existuje, jinak se použije výchozí chování.

Vytvoření Flask aplikace a definice koncových bodů byla původně součástí souboru `app.py`, který slouží ke spuštění aplikace. Nově bylo vytvoření aplikace přesunuto do tovární funkce `create_app`, umístěné ve stejnojmenném modulu v balíčku `api`. Primárně toto řešení slouží pro jednodušší práci s aplikací při testování, viz sekce 8.1. Součástí refaktORIZACE bylo dále i vyčlenění inicializace a definice cest ke koncovým bodům do oddělených funkcí pro zvýšení přehlednosti kódu i vzhledem k rozšíření o vlastní zpracování výjimek.

7.1.2 Administrační část

Pro integraci nového rozšíření bylo potřeba provést dílčí úpravy existující aplikace, primárně týkající se již popsanych změn práce s databázemi, na kterých rozšíření dále staví. Samotné rozšíření je umístěno do podřazeného balíčku `api.admin`. Administrační část je možné aktivovat nastavením proměnné prostředí, na základě nastavení se při vytvoření aplikace volitelně provede aktivace. Samotná inicializace rozšíření a registrace nových koncových bodů je umístěna v balíčku `admin`. Implementace koncových bodů stejně jako u existujících využívá třídy `Resource` z balíčku `Flask-RESTful`. Dále jsou použity následující nové balíčky:

- `SQLAlchemy 1.4`: knihovna pro objektově relační mapování
- `marshmallow 3.19`: deserializace a validace požadavků

- `marshmallow_dataclass` 8.6: rozšíření `marshmallow` umožňující deserializaci do objektů
- `Flask-JWT-Extended` 4.6: rozšíření pro práci s JWT tokeny v rámci Flask aplikace, primárně pro jejich vytváření a validaci

Objektově-relační mapování

Pro komunikaci s databází pomocí ORM slouží třída `ORMHandler`, která byla vytvořena jako ekvivalent existující třídy `DatabaseHandler`. Využívá stejnou konfiguraci a rovněž poskytuje metody pro připojení k centrální databázi a ke zvolené databázi municipality. Na rozdíl od třídy `DatabaseHandler` neposkytuje pro interakci s databází kurzor ale vrací objekt *session*, který slouží pro ukládání provedených změn databázových objektů. Vytvoření odpovídajících SQL dotazů a synchronizace s databází je provedena automaticky při potvrzení transakce (operace *commit*). Objekt *session* zároveň podporuje kontextový manažer, základní struktura kódu je tak při použití ORM stejná.

Objektový přístup je zaveden primárně pro účely administračního rozhraní, je však k dispozici pro použití v libovolné části aplikace. Moduly pro práci s ORM proto nejsou umístěny v balíčku `admin`. Třída `ORMHandler` a další definice jsou obsaženy ve vlastním balíčku `api.orm`.

Definice databázového schématu používá deklarativní přístup. Jednotlivé databázové entity odpovídající tabulkám jsou definovány pomocí tříd dědicích od báze třídy (typicky nazývané `Base`). Báze třídy také udržuje metadata, především definice databázových tabulek pro vytvořené třídy. Vzhledem k existenci dvou různých databázových schémat jsou použity dvě báze třídy – `BaseCentral` a `BaseMunicipality`.

Definice schématu databází municipalit kopíruje existující definici schématu z inicializačního SQL skriptu, jsou zachovány používané názvy, omezení atd. Definice datových typů primárně používají obecné datové typy poskytované `SQLAlchemy`. Jedinou výjimkou je datový typ `YEAR`, který je specifický pro dialekt MySQL. Pokud to bylo možné, byl v dalších případech specifický typ nahrazen za obecnější, např. datový typ `TINYINT` pro malá čísla za obecný číselný typ `SmallInteger`. Kromě těchto dílčích změn je vytvořené schéma identické. Do ORM schématu byly rovněž zahrnuty i nepoužívané tabulky připravené pro budoucí účely.

Schéma centrální databáze je vytvořeno zcela nově a používá pouze generické datové typy. Tabulka municipalit obsahuje všechny konfigurační řetězce, které bylo třeba vyjmout z uživatelského rozhraní, v průběhu implementace byla modifikována povinnost těchto atributů na základě požadavků pro nově přidávané municipality a byly přidány další atributy. Názvy cizích klíčů v centrální databázi odpovídají notaci `fk__<zdrojová tabulka>__<cílová tabulka>`.

Dále jsou definovány číselníky (*enums*) obsahující identifikátory odpovídající číselníkům v databázi, které slouží např. pro vytváření dotazů nebo nastavení výchozích hodnot nových entit. Také jsou definovány konstanty pro šířky (počty znaků) jednotlivých textových polí ve schématech databází.

Hesla, autentizace a přístupová práva

Ukládání a ověřování hesel využívá funkcionality z modulu `werkzeug.security`, který je součástí knihovny Werkzeug dostupné jako jedna ze závislostí frameworku Flask. Pro ukládání hesel je použita funkce, která ze zadaného vstupu vygeneruje hash za použití krypto-

grafické soli (*salt*). Výstupem je řetězec obsahující použitý algoritmus, sůl a samotný hash, který se následně ukládá do databáze. Uložený řetězec je následně vstupem funkce pro kontrolu hesla použitou při přihlášení.

Podpora pro JWT tokeny je přidána pomocí rozšíření **Flask-JWT-Extended**. Inicializace a výchozí konfigurace rozšíření se provádí při vytvoření aplikace (jako součást inicializace administrační části). Jedná se především o nastavení použitých algoritmů, doby platnosti a vlastního řešení chyb, např. pro neplatný token. V rámci aplikace je poté možné použít funkce na vytvoření JWT tokenů na základě nastavené konfigurace a v kontextu zpracování požadavků je možné verifikovat přijaté tokeny a číst z nich data.

Používají se přístupové (*access*) i obnovovací (*refresh*) tokeny. Tokeny jsou vytvořeny po úspěšném přihlášení a vráceny v těle odpovědi. Token následně klient používá při odesílání požadavků na koncové body vyžadující autentizaci. Pro přenos tokenu se používá HTTP hlavička **Authorization** s hodnotou ve formátu **Bearer <token>** (*Bearer authentication*). Běžné koncové body používají přístupový token, koncový bod pro obnovení vyžaduje obnovovací token a vygeneruje nové tokeny s prodlouženou platností.

Pro kontrolu potřebných práv u jednotlivých metod API je vytvořen pomocný dekorátor **protected**. Dekorátor verifikuje existenci a platnost přístupového tokenu. Poté na základě identifikace uživatele v tokenu načte jeho roli a přístupová práva z centrální databáze a porovná je s parametry předanými dekorátoru. Pokud má uživatel dostatečná práva, pokračuje se vykonáváním dekorované funkce, které je předán jako parametr objekt uživatele. V opačném případě vyvolá výjimkou s odpovídajícím chybovým kódem.

Koncové body a zpracování požadavků

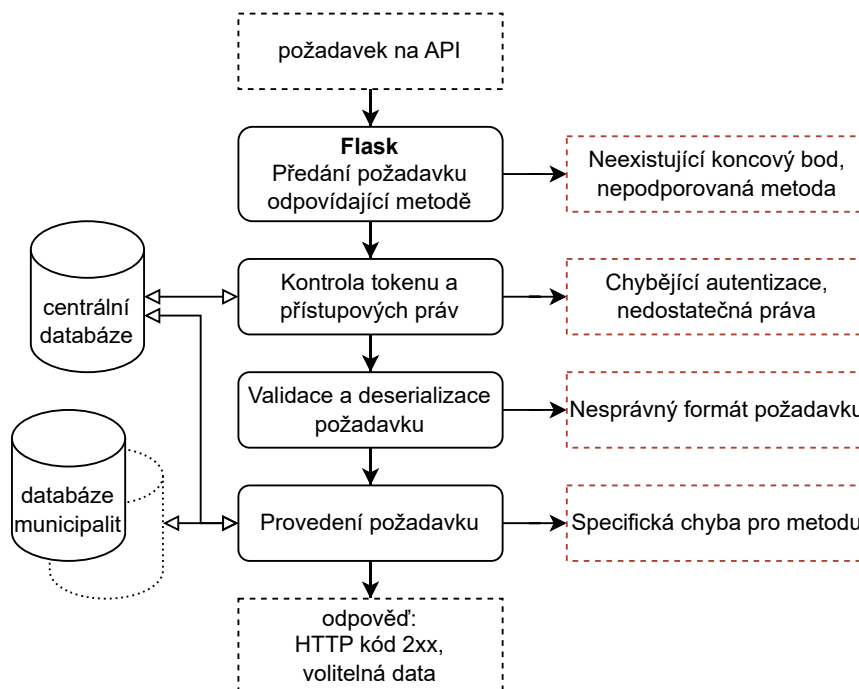
Implementace každého koncového bodu je obsažena v jednotlivých metodách třídy dědicí od třídy **Resource**. Pro každou entitu jsou typicky k dispozici dva koncové body, první reprezentující seznam entit, poskytující metody jako načtení všech entit nebo přidání nové entity, a druhý reprezentující konkrétní entitu, který umožňuje např. upravení entity. Třídy s implementací mají jednotný formát názvů složený z názvu entity a typu koncového bodu podle počtu entit (**List** nebo **One**), např. pro práci se členy existují třídy **MemberList** a **MemberOne**. Kromě koncových bodů entit existují i další se specifickou funkcionalitou, např. koncový bod přihlášení **Login**.

Značná část požadavků obsahuje data v těle požadavku, která jsou deserializována a validována pomocí balíčku **marshmallow**. Formát dat je JSON, jehož deserializace do datových typů programovacího jazyka Python (slovník nebo seznam) je poskytována již v rámci frameworku Flask. Balíček tak především zajišťuje validaci, kontrolu struktury přijatých dat, především existenci povinných položek, dále zajišťuje kontroly hodnot povinných položek, datových typů, doplnění výchozích hodnot atd. Další validaci je možné přidat pomocí předdefinovaných validátorů nebo vlastních validačních funkcí. Z existujících validátorů jsou použité např. kontroly délek u textových vstupů (podle konstanty odpovídající šířce sloupce v databázi) nebo ověření, že jsou zadávána pouze kladná čísla (např. u atributu počet zastupitelů). Vlastní validační funkce jsou použity např. pro kontrolu hexadecimálních kódů barev.

Schémata reprezentující jednotlivé typy požadavků jsou v balíčku **marshmallow** definována pomocí tříd. Data jsou však deserializována do obecných datových typů, tedy do stejné formy jako je poskytována na vstupu z frameworku Flask. Pro usnadnění práce s daty požadavků je dále použit balíček **marshmallow_dataclass**, který umožňuje definovat schémata

pomocí datových tříd³ a deserializovat validovaná data do instancí těchto tříd. Validace a deserializace pomocí `marshmallow` schémat je použita také pro parametry v cestě dotazů.

Průběh zpracování typického požadavku je zobrazen na schématu 7.1. Většina metod API v prvním kroku provede ověření uživatele a jeho práv pomocí dekorátoru `protected` a poté validaci přijatých dat. Následně s objektem uživatele a parametru požadavku začne provádět požadovanou operaci. Dle typu koncového bodu se připojí k odpovídající databázi, pro databáze municipality na základě parametru v cestě, a provede požadovanou operaci.



Obrázek 7.1: Schéma zpracování typického požadavku na administrační API. Na pravé straně jsou zobrazeny chyby, které mohou v daném kroku nastat. V případě chyby je zpracování požadavku ukončeno a je vrácen chybový HTTP kód a text chyby v těle odpovědi. Na levé straně je znázorněna interakce s databázemi.

Interakce s databází probíhá pomocí ORM. Komunikace přes připojení k databázi z knihovny `SQLAlchemy` je jednotně použita i pro případné SQL dotazy. Pro složitější požadavky nebo funkcionalitu používanou ve více metodách API se používají třídy `Service` obsahující sdílené metody pro práci s určitým typem entity, např. `MemberService` pro členy.

Na rozdíl od koncových bodů sloužících pro vizualizaci nemusí každý koncový bod vracet data. Odpověď je složena z odpovídajícího HTTP kódu podle typu a výsledku provedené operace a z volitelného těla ve formátu JSON. V případě chyb se vrací chybový HTTP kód a tělo obsahuje text chyby ve standardním formátu. Chybová odpověď může být vrácena přímo v metodě API nebo pomocí vyvolání výjimky. Vytvoření chybové odpovědi pro konkrétní výjimky je řešeno pomocí handlerů chyb, např. neúspěšná validace.

Vytvoření nové municipality

Každá municipalita v systému má uložena data zastupitelstva ve vlastní databázi municipality. Původní municipality v systému již databáze mají, pro nově přidané je vytvoření

³Python - modul `dataclasses` – <https://docs.python.org/3/library/dataclasses.html>

databáze součástí metody API na přidání municipality. Po přijetí validního požadavku je vytvoření municipality složeno z následujících základních kroků:

1. Validace textového identifikátoru z požadavku, kontrola duplicit.
2. Vytvoření nové databáze municipality `zastupitelstvo_<id_municipality>`.
3. Po vytvoření se nová databáze inicializuje vytvořením schématu pomocí metadat z báze třídy `BaseMunicipality` a následným vložením dat do tabulek číselníků.
4. Vložení údajů o municipalitě z přijatého požadavku do centrální databáze.

Po vytvoření je možné začít s novou municipalitou pracovat a ukládat data do databáze municipality. Vytvoření municipality na základě žádosti o přidání municipality používá stejný postup, pouze se údaje o municipalitě načtou z databáze, kde jsou uloženy jako součást žádosti.

Statistiky

Při provádění požadavků je třeba zohlednit používání statistik v systému představených v rámci analýzy, viz diagram 5.3. Statistika přítomnosti členů na zasedání (`BylPřítomen`) je vytvořena při přidání zasedání. Jako výchozí hodnota je použita *přítomnost* a výchozí účast na zasedání tak odpovídá počtu členů zasedání. Statistiky pro jednotlivé hlasovací možnosti jsou vytvořeny při přidání nebo upravení hlasování ve chvíli, kdy je daná hlasovací možnost poprvé použita.

Všechny statistiky jsou odvozeny od jednotlivých hlasování členů⁴. Při každé změně hlasování je proto provedena aktualizace. Pro každou tabulku statistik je vytvořen SQL dotaz, který načte aktuální hodnoty statistik z uložených hlasování v databázi. Po provedení změny v hlasovacích datech se pomocí dotazu načtou aktuální hodnoty relevantních statistik a uloží se. V případě smazání nebo úpravy hlasování může dojít k úplnému odstranění některé hlasovací možnosti, která tak již není v nově načtených hodnotách přítomna. Navíc je proto třeba smazat původní hodnoty těchto statistik.

Sumy hlasovacích možností pro hlasování se běžně vypočítají přímo z ukládaných hlasovacích možností při přidání nebo upravení hlasování. U tajných hlasování je však umožněno zadat explicitní hodnotu, jelikož sumární čísla mohou být známá a je žádoucí je reprezentovat. Jedná se o jedinou statistiku, která může být v systému nastavena přímo.

Členství

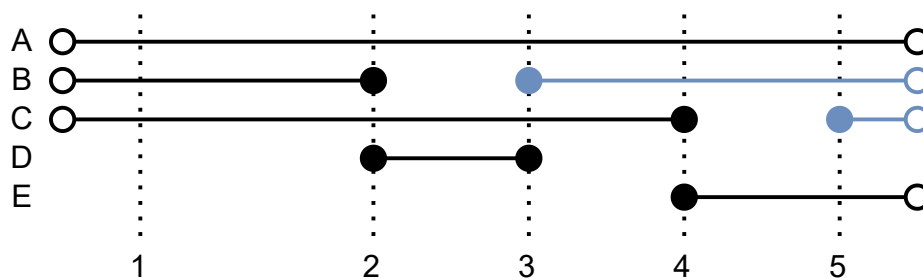
Příslušnosti členů k politickému subjektu a funkčnímu období udávají jejich členství na konkrétních zasedáních a účast na hlasování. Není možné je přímo editovat, jsou vytvářeny nebo upravovány nepřímo při vkládání či editování dalších entit.

Součástí požadavku přidání nového funkčního období může být seznam členů, pro které se vytvoří výchozí členství po celou dobu funkčního období. V požadavku přidání nového zasedání je třeba zohlednit existující členství, každý člen s platným členstvím k datu zasedání musí být explicitně zvolen nebo odebrán.

Příklady intervalů členství jsou ukázány na obrázku 7.2, dále je do obrázku odkazováno pomocí písmen v závorkách. Při ukončení členství (D) nebo změně politického subjektu (B, C) se existující členství ukončí k datu předchozího zasedání. Pokud je na zasedání

⁴Statistika přítomnosti členů na zasedání je odvozena z hlasovacích možností *nepřítomen* a *omluven*. Člen je nepřítomen, pokud jsou tyto hlasovací možnosti použity u všech hlasování. Atribut účast na zasedání odpovídá počtu členů, kteří nebyli nepřítomni.

přidán nový člen, je vytvořeno nové členství začínající od daného data (D, E). Zasedání také mohou být vkládána nechronologicky (tzn. před nebo mezi již existující zasedání). V takovém případě může dojít k rozdělení existujícího členství na dvě, je-li na nově přidaném zasedání člen členem jiného politického subjektu. Zasedání také může být vytvořeno s datem v mezeře mezi členstvími člena (pokud člen změnil politický subjekt v průběhu funkčního období), v takovém případě je prodlouženo existující členství nebo přidáno nové.



Obrázek 7.2: Schéma s příklady upravených intervalů členství. Písmena reprezentují jednotlivé členy a vodorovné čáry intervaly jejich členství. Barva čáry reprezentuje politický subjekt. Prázdné kolečko na konci intervalu značí, že interval není omezen – hodnota NULL, plné kolečko odpovídá konkrétnímu datu. Čísla a svislé přerušované čáry značí zasedání. Členové A, B a C mají členství po celé funkční období, D byl členem pouze pro zasedání 2 a 3, E je členem od zasedání 4 dále.

Loga municipalit

Existující loga municipalit byla přesunuta do serverové části a byl přidán nový koncový bod načtení loga municipality nahrazující původní statické soubory. Přidání nebo nahrazení loga je možné pomocí nového koncového bodu pro nahrání souboru s obrázkem loga, který se používá při přidávání nebo upravování municipalit.

Na základě jména přijatého souboru je vygenerováno UUID. Soubor je následně uložen do složky s logy se jménem složeným z UUID a původní přípony. Použití UUID namísto původního jména slouží jako ochrana před škodlivými názvy⁵ (např. obsahujícími navigaci po souborovém systému) a brání vzniku duplicit při ukládání souborů, jelikož jména nahrávaných souborů se mohou opakovat. Výsledný název souboru je uložen v centrální databázi u municipalit, ke které logo přísluší.

7.1.3 Změny schématu databáze municipalit

Schéma databáze municipalit oproti původní verzi prošlo několika změnami, především za účelem lepší reprezentace dat pro administrační část systému, jak bylo popsáno v podsekcí 6.2.1 návrhu. Původní inicializační SQL skript byl upraven, aby odpovídal upravenému schématu. Definice pomocí ORM entit rovněž odpovídá novému schématu, které je tak automaticky použito pro nově vytvořené databáze municipalit. Pro úpravy schématu existujících databází a případné úpravy dat byly vytvořeny migrační skripty.

⁵Ošetření jména souboru je provedeno ještě před vygenerováním UUID pomocí funkce `secure_filename` z knihovny Werkzeug, viz https://werkzeug.palletsprojects.com/en/3.0.x/utils/#werkzeug.utils.secure_filename.

Tabulka **členství** (**Prislušík**) nově umožňuje pro data začátku a konce použít hodnotu **NULL**. Migrace upravuje uložená členství pro použití nové sémantiky. Data začátku odpovídající datu prvního zasedání a data konce odpovídající poslednímu zasedání jsou nahrazena za **NULL**.

Provedená změna vyžadovala změny v **SQL** dotazech použitých v koncových bodech pro vizualizaci. Do podmínek, které filtrují data podle intervalu členství, byla přidána kontrola pro **NULL** se sémantikou, že daný konec intervalu není omezen⁶. Při vytváření datové sady se pro členství datum začátku a konce opět doplní podle zasedání, aby zůstal zachován původní formát. Dále byly provedeny úpravy zobrazení členství v uživatelském rozhraní, kde je doplněn text pro reprezentaci intervalů neomezených konkrétním datem.

Pro **aliasy** členů byla přidána zcela nová entita **AliasJmeno**, která reprezentuje původní (neaktivní) jméno člena s intervalem platnosti. Administrační část systému pracuje pouze s těmito novými aliasy. Převodní původních aliasů (**Alias**) bylo provedeno pomocí migračního skriptu.

Skript načte původní aliasy, které definují vazbu mezi původním členem (staré jméno člena) a novým členem (aktivní člen s novým jménem). Podle začátku a konce členství původního člena se vypočítá interval platnosti aliasu a je vytvořen nový alias se starým jménem a vypočítanou platností. Poté se pro všechny entity, které mají vazbu na původního člena, provede úprava nahrazením hodnoty cizího klíče za **ID** nového člena. Navazující členství jsou sloučena a jsou vypočítány nové hodnoty statistik hlasování pro nového člena. Nakonec dojde ke smazání původního člena i entity původního aliasu.

Existující koncové body, které v odpovědi vracejí členy, byly doplněny o načítání nových aliasů. Uživatelské rozhraní bylo upraveno pro zobrazování seznamu aliasů s intervalem platnosti. Nově jsou aliasy zobrazeny jako celé jméno, je tak možné upravit nejen příjmení, ale i křestní jméno. Tabulka původních aliasů je kvůli zpětné kompatibilitě ponechána. Sloučení dvou členů není v systému možné, případné provedení takové korekce by vyžadovalo manuální přidání původního aliasu a spuštění migračního skriptu.

Dále byly provedeny drobnější změny definic sloupců, většinou pro účely reprezentace nově vkládaných dat. Např. byly některé původně povinné atributy změněny na volitelné a byl zvýšen limit pro délku některých textových atributů. Do systému byla také přidána hlasovací možnost *omluven*.

7.1.4 Další rozšíření

V průběhu implementace vznikl požadavek umožnit v rámci systému vizualizovat i hlasovací data **rady municipality**⁷. Data rady jsou uložena v databázi se stejným schématem jako existující databáze municipality, odděleně od databáze pro zastupitelstvo. Název databáze je ve formátu **rada_<municipalita>**.

Pro reprezentaci více politických orgánů jedné municipality byla upravena centrální databáze a rozšířena o novou entitu **Organ**. Existující atributy municipality týkající se zastupitelstva byly přesunuty do této nové entity. Dále byl přidán číselník typů orgánů, který obsahuje dva používané orgány – zastupitelstvo a radu. Pro typy orgánů je stejně jako v případě municipalit zaveden textový identifikátor.

⁶Cílem úpravy podmínek existujících dotazů je zohlednit možné hodnoty **NULL** v intervalu členství. Nedochozí ke změně významu dotazů, řádky načtené upraveným dotazem odpovídají výstupu před úpravou členství.

⁷Pozn.: zpracování dat a vytvoření databáze hlasování rady nebylo součástí této práce, pouze integrace databáze do vizualizační části zobecněného systému.

Každá municipalita nově může mít více svých databází pro jednotlivé orgány. Jméno databáze je složeno z identifikátorů orgánu a municipality ve tvaru `<id_organu>_<id_municipality>`. Do cesty pro každý koncový bod pracující s daty municipality byl doplněn nový parametr identifikátoru orgánu. Dále byly upraveny metody pro připojení k databázi municipality, které nově pracují i s identifikátorem orgánu, jako výchozí hodnota slouží zastupitelstvo.

Další změnou bylo vyjmutí některých informací o aplikaci (např. název) z uživatelského rozhraní. Vybrané **texty aplikace** byly přesunuty do databáze odkud se načítají podobně jako je tomu u jednotlivých municipalit. V centrální databázi jsou reprezentovány další přidanou entitou **AppInfo** a byl přidán nový koncový bod pro jejich načtení. Výsledné schéma centrální databáze je k vidění v příloze v diagramu **B.1**.

7.2 Uživatelské rozhraní

Implementace uživatelského rozhraní rozšiřuje existující aplikaci vytvořenou v knihovně **React**. Obdobně jako u serverové části je řešení složeno ze dvou základních částí – generalizace existující aplikace pro práci s více municipalitami a přidání nové administrační části.

Prvním krokem bylo přidání výběru municipality jako nové úvodní stránky, dostupné municipality se načítají z nového koncového bodu seznamu municipalit. Po výběru municipalit byla přidána další stránka výběru politického orgánu. Tato stránka se však zobrazí pouze, pokud má municipalita více politických orgánů, jinak se automaticky zvolí jediný existující.

Identifikátory zvolené municipality a orgánu jsou uloženy jako parametry v cestě. Kvůli přidání nových parametrů a stránek bylo třeba upravit existující odkazy v rámci aplikace. Přidané parametry byly dále použity k aktualizaci cest u existujících požadavků na API. Použití konfiguračních souborů s údaji o municipalitě bylo nahrazeno načítáním údajů z koncového bodu konkrétní municipality podle identifikátoru zvolené municipality. Dále byly provedeny dílčí úpravy zobrazení nově nepovinných hodnot, např. kontaktního e-mailu. Loga municipalit a informace o aplikaci byly rovněž upraveny pro načítání dat z nových koncových bodů.

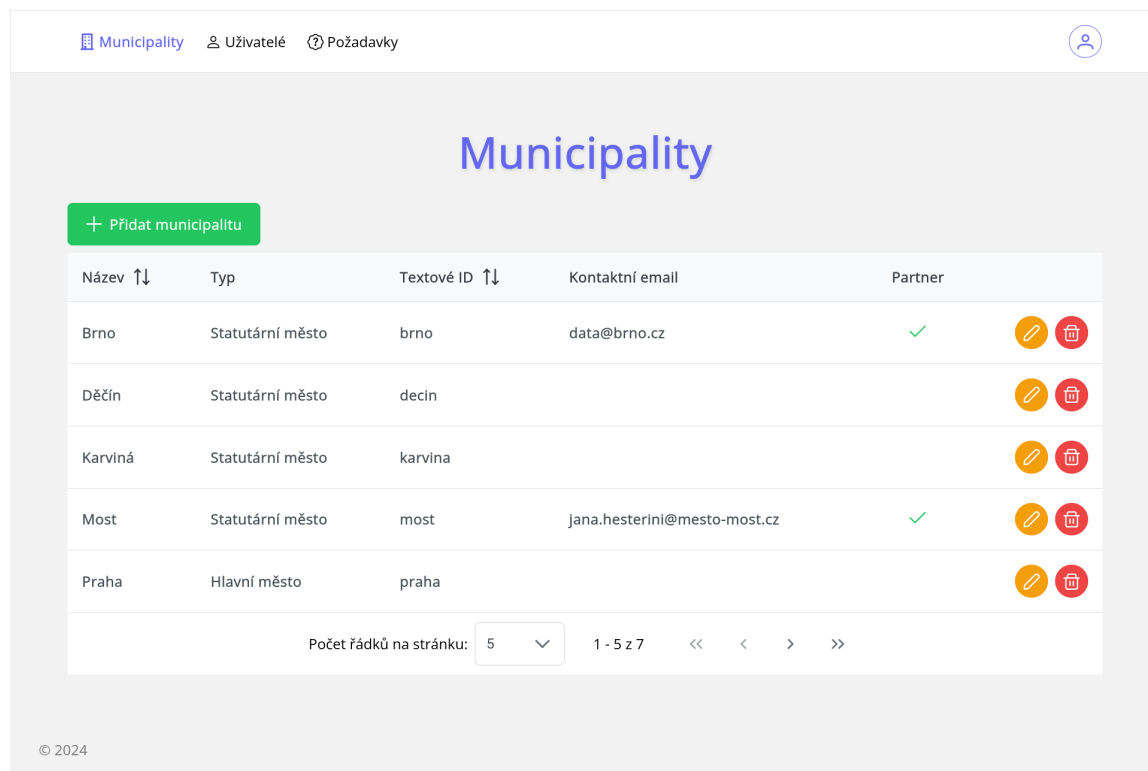
Soubor s šablonou HTML dokumentu pro **React** aplikaci původně v hlavičce (element **head**) obsahoval informace o municipalitě, a to např. v titulku stránky nebo meta značkách s názvem a popisem. Tento přístup není dále možný, jelikož systém pracuje s více municipalitami a údaje o municipalitách se načítají dynamicky, nelze tak použít konstanty v šabloně. Informace o municipalitě byly proto z hlavičky odstraněny a nahrazeny generickými texty.

Pro dynamické změny údajů v hlavičce stránky byla přidána knihovna **React Helmet**. Knihovna poskytuje komponentu **Helmet**, které se jako podřazené elementy předávají elementy hlavičky s novými hodnotami. Hodnoty jsou následně dynamicky změněny v dokumentu HTML stránky aplikace. Pomocí komponenty **Helmet** jsou doplněny údaje o municipalitě na všech stránkách s identifikátorem municipality v cestě.

7.2.1 Administrační část

Administrační část byla přidána jako nový modul v aplikaci dle struktury představené v návrhu, viz obrázek **6.3**. Inicializace administrační části přidá cesty pro nové stránky do směrovače aplikace z knihovny **React Router**. Na nejvyšší úrovni jsou dostupné stránky přihlášení, registrace a šablona ostatních stránek dostupných po přihlášení. Součástí šab-

lony je záhlaví s menu obsahujícím odkazy na dostupné přehledové stránky administrační části, dále zápatí stránky a komponenta `Outlet`, sloužící pro vykreslení těla stránky. Obsah stránky je dán komponentou níže v hierarchii cest směrovače, která odpovídá kompletní cestě. Ukázka stránky přehledu municipalit je k vidění na obrázku 7.3. Snímky obrazovky dalších stránek představených v návrhu jsou umístěny v příloze C.



Obrázek 7.3: Stránka přehledu municipalit. Ukazuje záhlaví stránky s menu se zvýrazněnou aktuální stránkou a ikonou pro zobrazení menu uživatele na pravé straně, dále hlavní obsah stránky s tabulkou municipalit a zápatí.

Přístup do administrační části je povolen pouze přihlášeným uživatelům, přihlášení je kontrolováno na základě existence platného přístupového tokenu na úrovni šablony stránek administrační části. Pro kontrolu přístupu na základě práv je vytvořena komponenta `ProtectedRoute`, která umožní přístup na stránku jen, pokud má aktuální uživatel požadovanou roli či právo spravovat municipalitu. Role a přístupová práva přihlášeného uživatele se načítají z API. Kontrola práv je dále použita v rámci jednotlivých komponent pro úpravy zobrazení na základě povolených akcí uživatele, např. odkaz na editaci v tabulce municipalit dostupný jen při existenci práva spravovat municipalitu nebo zobrazení odkazů na stránky pro administrátora v menu (odkazy Uživatelé a Požadavky na obrázku 7.3).

Komunikace se serverem probíhá v aplikaci pomocí HTTP klienta z knihovny `axios`. Pro požadavky vyžadující autentizaci byla vytvořena další instance klienta `authAxios`. Konfigurace klienta obsahuje funkci, která při každém odeslání požadavku vloží do odpovídající hlavičky aktuální přístupový token.

Pro tvorbu uživatelského rozhraní byly použity komponenty z knihovny `PrimeReact`. Hlavní komponentou použitou k zobrazování dat je datová tabulka `DataTable`, která také slouží pro stránkování a řazení zobrazovaných dat. Dialogová okna s formuláři na přehle-

dových stránkách a potvrzovací dialogy jsou implementovány pomocí komponenty `Dialog`. Zobrazování informací o úspěšně provedeném požadavku nebo případných chyb je na jednotlivých stránkách prováděno skrze komponentu `Messages`, která poskytuje metody pro zobrazování zpráv. Mimo zmíněných hlavních komponent jsou použity i další, z obecných např. tlačítko `Button`, ze specifických např. komponenta `Menubar` pro hlavní menu (stejně jako ve vizualizační části aplikace). Navíc jsou používány ikony z knihoven `primeicons` a `bootstrap-icons`.

Formuláře

Vytváření formulářů a práce s nimi využívá knihovny `React Hook Form` (RHF), která poskytuje správu stavu a hodnot formuláře, validaci vstupů, zobrazování chyb a další pomocné funkce. Každý formulář je definován ve vlastní formulářové komponentě. Integrace s RHF probíhá pomocí použití *hook* funkce `useForm`, které se na začátku parametru předá inicializace formuláře objektem výchozích hodnot a další konfigurace.

Jména klíčů výchozích hodnot odpovídají jménům jednotlivých vstupů (resp. hodnot formuláře, polí). Vstupní elementy jsou poté propojeny s RHF, čímž je jim nastavena výchozí hodnota a chování, včetně validace. U každého vstupu je také zobrazena případná chyba příslušející danému poli, dostupná ze stavu poskytovaného `useForm`. Každé formulářové komponentě je jako parametr předána funkce pro spuštění po odeslání formuláře (`onSubmit`), která je pomocí RHF vykonána pouze při úspěšné validaci všech polí.

Použité komponenty vstupů z knihovny `PrimeReact` jsou kontrolované (*controlled*), tzn. jejich stav se řídí hodnotou stavové proměnné vytvořené pomocí *hook* funkce `useState`. Integrace kontrolovaných komponent s RHF se provádí pomocí komponenty `Controller`, která umožňuje vytvořit kontrolovaný vstup či vstupy odpovídající jednomu poli formuláře. Pomocí parametrů (*props*) se nastaví základní parametry vstupu, především název pole a případná validace, a definuje se funkce `render`, která provádí vykreslení kontrolovaného vstupu. Funkce jako parametr dostane sadu parametrů pro vstupní element obsahující především funkci `onChange`, která provádí aktualizaci stavu pole, a dále aktuální stav pole, např. příznak neplatného vstupu.

Většina komponent je použita opakovaně a ve více formulářích. Pro usnadnění znovupoužití a kvůli snaze o omezení duplikace kódu byly pro většinu používaných vstupů z knihovny `PrimeReact` vytvořeny generické RHF komponenty, které zajišťují integraci vstupu s RHF pomocí komponenty `Controller`, např. pro textový vstup `InputText` je vytvořena komponenta `RHFInputText`. Komponenty zároveň obsahují případnou výchozí konfiguraci komponent vstupů.

Většina použitých vstupů je z knihovny `PrimeReact`. Pro některé účely se však ukázaly poskytované komponenty jako nevhodné, a jsou proto použity i komponenty z dalších knihoven. Pro zadávání barev slouží komponenta `SketchPicker` z knihovny `react-color`, která umožňuje vybírat barvu z palety, ale také pomocí textových polí zadávat jednotlivé segmenty RGB nebo celý hexadecimální kód⁸. Pro zadávání času se používá komponenta `TimeInput`⁹ z knihovny `NextUI`, která lépe podporuje vyplňování pomocí klávesnice. Obě

⁸Zadání konkrétní barvy je potřebné u municipalit, které mají přesně definovaný vizuální styl, např. město Brno – <https://www.brno.cz/w/logo>

⁹NextUI – TimeInput – <https://nextui.org/docs/components/time-input>. Komponenta `TimeInput` je postavená na komponentě `TimeField` z projektu React Aria, viz <https://react-spectrum.adobe.com/react-aria/TimeField.html>. Díky rozšířeným možnostem konfigurace je pro použití v aplikaci vhodnější komponenta `TimeInput`.

komponenty jsou také používány jako kontrolované skrze generické RHF komponenty s výchozí konfigurací.

Příkladem formulářové komponenty je **formulář hlasování**. Komponenta byla vytvořena na základě návrhu a umožňuje vkládání nových hlasování i úpravy existujících. Ukázka formuláře ve verzi pro úpravu hlasování je na obrázku 7.4. Formulář v obou verzích podporuje zadávání hlasování členů pomocí tabulky i pomocí karet politických subjektů. Mezi oběma komponentami lze přepínat pomocí záložek (využívá komponenty `TabView` a `TabPanel` z knihovny `PrimeReact`).

Formulář kvůli efektivnějšímu zadávání hlasování při výběru výsledku automaticky přednastaví hlasy členů na možnost *ano* nebo *ne*. Nastavení tajného hlasování obdobně přednastaví hlasovací možnost *tajná*, a navíc umožní editovat sumy hlasovacích možností. Jinak se sumy hlasovacích možností počítají automaticky podle vyplněných hlasů a mohou tak sloužit pro kontrolu vloženého hlasování podle sumárních čísel, která často bývají v hlasovacím protokolu či zápisu dostupná. Hromadný výběr hlasovací možnosti lze rovněž provést pomocí výběru hlasovací možnosti pro celý politický subjekt nebo kliknutím na název sloupce v záhlaví tabulky.

Číslo hlasování*

4

Čas

08:29:36

×

Výsledek* ⓘ

přijato

▼

Platné

✓

Procedurální

✓

Tajné ⓘ

□

Předmět hlasování

1. Technický bod (zapisovatelé, právní asistence, ověřovatelé zápisu, schválení programu zasedání ZMB) - program

Odkaz na protokol hlasování

<https://apl.brno.cz/protokoly-zmb/zmb-z8-40/20220906-4-30575.h>

Sumy hlasovacích možností

Ano

36

Ne

0

Zdržel se

0

Nehlasoval

8

Nepřítomen

11

Omluven

0

Tabulka

Politické subjekty

Jméno ↑↓	Příjmení ↑↓	Subjekt ↑↓	Ano	Ne	Nehlasoval/a	Nepřítomnost	Omluven/a	Tajná	Zdržel/a se	Prezenčně
Markéta	Vaňková	ODS	☒	☐	☐	☐	☐	☐	☐	☒
David	Aleš	Nezávislí pro Brno	☒	☐	☐	☐	☐	☐	☐	☒
Petr	Bořecký	ANO 2011	☐	☐	☒	☐	☐	☐	☐	☒
René	Černý	ANO 2011	☐	☐	☐	☒	☐	☐	☐	☒
Pavel	Dvořák	Nezávislí pro Brno	☐	☐	☐	☒	☐	☐	☐	☒
Jiří	Faltýnek	Nezávislí pro Brno	☐	☐	☐	☒	☐	☐	☐	☒
Marek	Janiček	ANO 2011	☐	☐	☐	☒	☐	☐	☐	☒
Kateřina	Jarošová	ANO 2011	☒	☐	☐	☐	☐	☐	☐	☐
Karel	Kalivoda	ANO 2011	☐	☐	☒	☐	☐	☐	☐	☒

Upravit hlasování

Obrázek 7.4: Formulář pro upravení hlasování. Je zobrazena výchozí záložka pro výběr hlasovacích možností v tabulce. Kvůli velikosti je zobrazen pouze samotný formulář bez nadpisu a dalších částí stránky, tabulka byla zkrácena a nejsou zobrazeni všichni členové.

Styly

Styly stránek a komponent administrační části jsou umístěny centralizovaně v jediném souboru ve formátu **CSS**. Po vzoru globálních stylů aplikace se pro názvy tříd používají konvence z metodiky BEM (*Block Element Modifier*¹⁰). Vytvořené třídy jsou aplikovány pouze na komponenty administrační části. Modifikace globálních stylů pro účely administrační části je provedena použitím selektoru se třídou, která je nastavena tělu stránek v šabloně. Konvence BEM nejsou použity u tříd rozšiřujících styly komponent z knihovny **PrimeReact** a u pomocných tříd. Komponenty vstupů z dalších knihoven mají kvůli jednotnosti definován styl vizuálně podobný komponentám z knihovny **PrimeReact**.

¹⁰Metodika BEM – <https://getbem.com/>

Kapitola 8

Testování

Tato kapitola se zabývá testováním rozšířeného systému pro ověření jeho správné funkčnosti a použitelnosti. Prováděné změny a přidaná rozšíření byly průběžně testovány v průběhu implementace. Ověření správné funkčnosti API bylo také prováděno nově vytvořenými automatickými testy. Nové verze systému byly postupně nasazovány pro interní použití a pro verzi splňující všechny zadané požadavky proběhlo uživatelské testování.

8.1 Testy API

Původní systém neobsahoval žádné automatizované testy. Vzhledem k tomu, že v rámci implementace bylo třeba provést zásahy do většiny existujících koncových bodů systému, bylo nutné ověřovat správnou funkčnost jednotlivých koncových bodů a zabránit vzniku chyb při úpravách.

Byly vytvořeny testy API v testovacím frameworku `pytest` sloužící pro testování aplikace poskytující REST API v balíčku `api`. Testy jsou prováděny na úrovni jednotlivých koncových bodů. Pro komunikaci s API je použit testovací klient frameworku Flask¹. Testovací klient simuluje komunikaci s aplikací vytvořenou pomocí funkce `create_app` bez nutnosti spuštění serveru. Odesílání HTTP požadavků na jednotlivé koncové body je dostupné pomocí metod klienta.

Ještě před zahájením implementace podpory pro více municipalit v systému byly jako první vytvořeny testy všech existujících koncových bodů. Jedná se o koncové body pouze pro čtení, které slouží pro načítání dat pro vizualizace. Každý test odešle požadavek na koncový bod, který načte údaje konkrétní známé entity. Součástí testů jsou soubory ve formátu JSON, které obsahují uložené odpovědi na odpovídající API požadavek získané z nasazeného systému v původní verzi. Ověření je následně provedeno porovnáním odpovědi vrácené z API s obsahem odpovídajícího souboru.

V některých případech není možné ověřit odpověď pouhým porovnáním. U koncových bodů vracejících seznam entit je testováno, jestli odpověď obsahuje všechny entity uložené v testovacím souboru. Samotný seznam v odpovědi ale může obsahovat více entit, např. při přidání nového funkčního období. Ověření datové sady také používá uložený soubor s očekávanou odpovědí. Vzhledem k velikosti souboru však v případě, že vrácená datová sada neodpovídá očekávané, není provedena identifikace konkrétních odlišností. Pouze se zkont-

¹Flask – testovací klient – <https://flask.palletsprojects.com/en/2.1.x/testing/#sending-requests-with-the-test-client>

roluje, jestli vrácená odpověď odpovídá očekávanému schématu. Pokud je rozdíl v datech, je třeba konkrétní problém identifikovat manuálně.

Jako testovací data byla zvolena data z 8. Zastupitelstvo města Brna, jelikož se jedná o již ukončené funkční období a správnost dat byla manuálně ověřena [52]. V očekávaných datech, a tedy i v testech, by proto mělo docházet k minimálním změnám. Tato data jsou použita pro naprostou většinu testů, ověření koncového bodu načtení funkčních období však dále používá i údaje o 9. Zastupitelstvu.

Po provedení generalizace systému pro více municipalit byly testy aktualizovány pouze přidáním identifikátoru municipality `brno` do cest a bylo ověřeno, že provedenými zásahy nedošlo ke změně chování existujících koncových bodů. Následně byly přidány testy nově vytvořených koncových bodů pro načítání dat municipalit. Jako testovací data byly použity údaje o městech Brno a Most pro ověření načtení údajů o municipalitách. Pro ověření správného výběru municipality podle identifikátoru byla také použita hlasovací data města Most.

Použitý způsob ověření správnosti znamená, že změna schématu odpovědí nebo dat způsobí selhání testů. Historická data by se neměla měnit vůbec, při úpravě chování koncového bodu je třeba aktualizovat i test a ověřit novou funkcionalitu. V průběhu implementace byly soubory s očekávanými daty upraveny na základě nového očekávaného chování, např. pro členství s hodnotami `NULL`. Také byly přidány testy dalších koncových bodů přidávaných do vizualizační části systému.

Většina vytvořených testů však cílí na nové koncové body pro administrační část systému. Testy byly přidávány postupně v průběhu implementace spolu s vytvářenými koncovými body. Na rozdíl od testů pro vizualizační část dochází k úpravě dat v databázi, byla proto zavedena možnost konfigurace testů a používání oddělených testovacích databází. Pro inicializaci a resetování testovacích databází jsou k dispozici `SQL` skripty a funkce, které potřebnou inicializaci provedou před spuštěním testů automaticky (*fixtures*).

Testy jsou rozděleny do jednotlivých modulů podle skupiny koncových bodů pro entitu nebo funkcionalitu, kterou testují. Základní testy ověřují správné vytváření a upravování entit, načítání dat ve správném formátu atd. Pro jednotlivé koncové body se také ověřují korektní chybové reakce na nesprávné vstupy či nepovolené požadavky (např. smazání člena, pro kterého jsou uložena hlasování). Specifické testy slouží k ověření automatických akcí při provádění požadavků, jejichž výsledek není přímo viditelný v uživatelském rozhraní, např. vytváření statistik nebo práv uživatelů. U statistik testy také ověřují správný výpočet hodnot po přidání nebo úpravě hlasování nebo naopak jejich odstranění po smazání hlasování.

Na rozdíl od testů pro vizualizační API se používají primárně uměle vytvořená data, která slouží k otestování konkrétní funkcionality. Část testů však také používá kopii existujících dat pro Brno, např. pro ověření korektního odmítnutí smazat členy s uloženými hlasováními nebo na ověření korektního přepočtu statistik hlasování na reálných datech.

Ověření správnosti uložených dat po provedení akce probíhá především pomocí načtení dat z nových koncových bodů administračního rozhraní. Ověření komplikovanějších operací využívá i původní koncové body pro vizualizaci, např. členové zasedání zobrazení ve vizualizaci musí odpovídat členům zvoleným při vytvoření zasedání. V některých případech jsou také data přímo načtena z databáze, typicky pro ověření hodnoty konkrétní statistiky nebo ověření jejího smazání.

Testy administrační části sloužily především pro ověření očekávaného chování koncových bodů před jejich integrací s uživatelským rozhraním a dále pro ověření akcí, které v uživatelském rozhraní nejsou možné, např. provedení akce bez dostatečných práv, ode-

slání požadavků s daty v nesprávném formátu atd. Stejně jako testy vizualizačního API také slouží k zamezení vzniku chyb při úpravách, čehož bylo využito např. při přechodu na validaci a deserializaci pomocí balíčku `marshmallow_dataclass`.

8.2 Uživatelské testování

Vytvářené řešení bylo při implementaci průběžně testováno pomocí testů API a manuální kontrolou uživatelského rozhraní v prohlížečích Mozilla Firefox a Chromium. Nové verze systému byly také průběžně nasazovány pro interní použití.

Generalizovaný systém rozšířený o podporu vizualizací pro více municipalit byl interně nasazen a používán pro vizualizaci dat ze všech municipalit, pro které byla dostupná data. Na začátku se jednalo o Brno a Most, postupně byly přidávány další. Byla tak ověřena správná funkčnost systému pro více municipalit. Na základě zpětné vazby a požadavků daných novými municipalitami docházelo k dílčím úpravám, např. změně atributů, jejichž hodnoty nebyly pro některé municipality dostupné, na volitelné. Stejně byla nasazena i verze s rozšířením o rady municipalit, která byla ověřena na datech pro město Most dostupných pro zastupitelstvo i radu. Další vývoj systému již probíhal primárně na této generalizované verzi systému.

Systém rozšířený o administrační rozhraní byl nasazen odděleně. Poprvé byl systém nasazen v základní verzi umožňující pouze zobrazování a vkládání dat a sloužil především pro prezentaci nové funkcionality a průběžnou zpětnou vazbu. Postupně byly nasazovány nové verze rozšířené o další funkcionalitu.

Ve verzi s kompletní funkcionalitou (finální verze) začal systém být interně dalšími členy projektu používán pro uložení dat hlasování zastupitelstev krajů, jejich správu i vkládání nových dat. Průběžně tak bylo ověřováno správné fungování systému, reportovány chyby či nevhodné chování a další zpětná vazba týkající se především použitelnosti systému a efektivního vkládání dat.

Finální verze systému byla rovněž podrobena uživatelskému testování s externími uživateli, kteří neměli se systémem předchozí zkušenost. Testování mělo několik cílů, především ověřit správnou funkčnost systému a jeho použitelnost a pochopitelnost pro nového uživatele. Dalším cílem bylo otestovat vkládání dat různých municipalit, a tedy v různých zdrojových formátech. Bylo žádoucí, aby test pokryl co největší část funkcionality systému. Obsahem testování proto bylo přidání hlasovacích dat zcela nové municipality.

Testování se zúčastnili celkem 4 uživatelé, 3 muži a 1 žena ve věkové kategorii 20–30 let. Každý vybraný uživatel byl dostatečně technicky zdatný pro práci s webovým uživatelským rozhraním. Každý také měl alespoň základní přehled o hlasování zastupitelstev. Testování probíhalo na vlastním zařízení s velkou obrazovkou, tzn. počítači nebo notebooku, a v libovolném prohlížeči. Byly použity prohlížeče Mozilla Firefox, Microsoft Edge a Google Chrome (dvakrát).

Uživatelé před zahájením testu dostali základní informace o systému a seznam pokynů pro testovací úlohu. Každý uživatel měl také pro municipalitu, jejíž data měl vložit, k dispozici základní údaje a odkazy na data. Nebylo žádoucí, aby uživatelé trávili v průběhu testování čas dohledáváním informací. Měli proto k dispozici odkazy např. na seznamy členů pro vkládaná zasedání, údaje o politických stranách a především odkazy na hlasovací protokoly nebo zápisy ze zasedání sloužící jako zdroj hlasovacích dat. Základní postup úlohy byl následující:

1. Registrace.

2. Podání žádosti o přidání nové municipality.
3. Vložení všech politických subjektů a členů pro funkční období.
4. Vytvoření funkčního období.
5. Vytvoření 2 vybraných zasedání dle pokynů, a následně vkládání hlasování ze zdrojových dokumentů. Minimálně jedno hlasování musí být vloženo pro každé zasedání.

Každý uživatel měl přiřazenou jinou municipalitu. Jednou z nich bylo jako referenční Brno², dále městské části Brno-střed, Brno-Nový Lískovec a město Slavkov u Brna. Hlasování prvních dvou municipalit byla dostupná z hlasovacích protokolů, pro ostatní dvě ze zápisů zasedání.

Uživatelé vykonávali testovací úlohu v časovém rozmezí zhruba 2 hodin. Všem uživatelům se ji podařilo úspěšně splnit. Počet vložených hlasování se mezi uživateli lišil, nejméně hlasování bylo vloženo pro Brno – 6 a nejvíce pro Brno-Nový Lískovec – 75. Cílem však není počty porovnávat, což ani není možné, jelikož se jednotlivé municipality liší počtem zastupitelů, a tedy i počtem vkládaných hlasování členů pro každé hlasování, a také formátem zdrojových dokumentů obsahujících hlasování. Úspěšným splněním testovací úlohy byla ověřena funkčnost systému a především použitelnost i pro uživatele bez předchozí zkušenosti se systémem. Také bylo ověřeno použití systému pro vkládání hlasovacích dat v různých vstupních formátech s odlišnou detailností.

Na základě zpětné vazby a podnětů z průběžného interního používání i uživatelského testování byly provedeny další úpravy systému za účelem zlepšení použitelnosti a zvýšení efektivity práce. Mezi provedené změny patří např.:

rozšíření řazení Možnost řazení byla doplněna do většiny sloupců tabulek `DataTable`, kde dává smysl. Pro konkrétní tabulky členů bylo dále umožněno řazení podle několika sloupců sloužící především pro řazení podle politického subjektu a příjmení. Navíc byla doplněna i možnost řazení podle zvolených hodnot ve sloupci obsahujícím vstup pro výběr politického subjektu.

výběr členů pomocí tabulky Původní výběr nových členů pro přidání do funkčního období nebo na zasedání používal komponentu `Dropdown` (rozbalovací seznam), což bylo pro vložení více členů neefektivní. Byl proto nahrazen výběrem z tabulky pomocí zaskrtávacích polí. Také je umožněn hromadný výběr všech, což je vhodné pro vložení prvního funkčního období.

řazení členů Řazení členů v hlasovacích protokolech nemusí odpovídat příjmení ani politickému subjektu, např. při řazení podle čísla hlasovací karty. Odlišné zobrazení členů v tabulce pro vkládání hlasování než v hlasovacím protokolu však vede ke zpomalení vkládání dat a zvyšuje šanci na vložení chybných hodnot. Členům byl proto přidán atribut pořadí a bylo umožněno nastavit explicitní pořadí členů. Tabulka hlasování členů je ve výchozím stavu seřazena právě podle pořadí členů.

drobečková navigace Pro lepší přehled uživatele o umístění v systému, navigaci mezi stránkami a usnadnění návratu ze stránky detailů na přehled byla do horní části stránky přidána drobečková navigace (*breadcrumb menu*), navigaci je možné vidět na snímcích obrazovky v příloze C.

²V rámci testovací úlohy byla vytvořena nová oddělená municipalita, žádná existující data nebyla použita. Význam je v tom, že data pro Brno již byla zpracována a je tak jisté, že hlasovací protokoly obsahují kompletní informace.

tabulka hlasování členů Tabulka prošla několika úpravami pro zefektivnění vkládání hlasování a snížení chybovosti. Zobrazení záhlaví stránky s nadpisy sloupců bylo upraveno na zafixované (*sticky*), aby při posunu níže na stránce zůstalo viditelné v horní části stránky. Název hlasovací možnosti je zároveň dostupný jako popis každého vstupního elementu.

Dále bylo upraveno zobrazení tabulky. Kvůli zobrazení většího množství členů na obrazovce a omezení nevyužitého místa je použito kompaktnější zobrazení s nižšími řádky. Pružované (*striped*) zobrazení tabulky slouží pro lepší vizuální oddělení řádků. Při výběru vstupu je aktivní řádek navíc zvýrazněn.

Výslednou verzi tabulky lze vidět na obrázku 7.4. Navíc byla vylepšena responzivita tabulky pro zobrazení na menších obrazovkách nebo při rozdělení obrazovky pro zobrazení systému i zdrojového dokumentu pro vkládání dat. Pro efektivnější vkládání nových hlasování byla také přidána možnost zapamatovat po vložení hlasování hlasovací možnosti *nepřítomnost* a *omluven* a hodnotu prezence, které se typicky v rámci jednoho zasedání nemění.

podpora vyplňování pomocí klávesnice Na vyplňování formulářů pomocí klávesnice bylo myšleno již při implementaci, např. u zmíněných vstupů pro zadávání barev a času nebo přidáním možnosti zvolit zaměřený (*focused*) přepínač (*radio button*) pomocí klávesnice. Uživatelské testování však odhalilo další problémy s výchozím chováním některých vstupních komponent.

Vstup pro počet členů zastupitelstva používal validaci čísel přímo pomocí komponenty `InputNumber`. Minimální počet členů je 5, validace však probíhala pro každý individuální vstup a nebylo tak možné vložit např. číslo 35, protože hodnota 3 je menší než zadané minimum 5. Problém byl vyřešen přesunutím validace do RHF.

Výběr roků začátku a konce funkčního období používá komponentu `Calendar`. Komponenta však ve verzi pro výběr roků na rozdíl od výchozího výběru celých dat nepodporuje automatické zpracování hodnoty zadané pomocí klávesnice. Vstupní element tak efektivně sloužil jen pro zobrazení hodnot vybraných pomocí menu. Bylo doplněno vlastní zpracování vstupního textu a nastavení hodnoty roku pro validní vstup.

Kapitola 9

Závěr

Cílem práce bylo rozšířit existující systém pro vizualizaci hlasování Zastupitelstva města Brna o podporu více municipalit, jejich správu a také vkládání a správu dat z hlasování jejich zastupitelstev. Účelem bylo poskytnout vizualizace z libovolného počtu zastupitelstev a umožnit uživatelům vkládat a spravovat relevantní data přímo v systému.

Teoretická část práce představila obce a kraje v České republice, zasazené do kontextu územního členění, samosprávy a relevantní legislativy, především se zaměřením na jejich zastupitelstva. Byla blíže prozkoumána data, která ze své činnosti zastupitelstva produkuje, a také systémy na zaznamenávání hlasování. Navíc byl proveden průzkum dostupných datových sad z hlasování zastupitelstev a pro vybraná města i dostupných hlasovacích dat v dokumentových formátech. Průzkum se zabýval hlasování zastupitelstev v České republice i zahraničí. Dále byla představena problematika informačních systémů, webových aplikací a použitelnosti uživatelských rozhraní.

Následně byla provedena analýza existujícího systému pro vizualizaci hlasování, byli identifikováni cíloví uživatelé a specifikovány požadavky na správu municipalit a dat v rozšířeném systému. Podle získaných informací byl vytvořen návrh úprav existujícího systému pro podporu více municipalit a nového rozšíření pro správu dat. Navržené úpravy a rozšíření systému byly úspěšně implementovány. Funkčnost zobecněného systému byla ověřena jeho použitím s reálnými daty více municipalit. Navíc byla přidána možnost vizualizovat hlasování z dalšího orgánu municipality, konkrétně rady. Pro ověření správné funkčnosti původních i nově přidaných koncových bodů byly vytvořeny testy API. Nově přidaná administrativní část pro správu dat byla nasazena pro interní použití a posloužila pro správu zastupitelstev krajů i vkládání dat. Také bylo provedeno uživatelské testování administrativní části. Dle zpětné vazby byly provedeny další úpravy systému pro zvýšení jeho použitelnosti. Cíle práce tak byly úspěšně naplněny.

Budoucí práce na projektu může systém rozšiřovat v několika oblastech. Návrhy pro další usnadnění a zefektivnění vkládání dat zahrnují např. umožnit vkládání dat do formulářů v předzpracovaných serializovaných formátech jako je CSV nebo začít používat uložená data v systému pro nastavení výchozích hodnot formulářů, např. statistiku přítomnosti. U datového modelu se nabízí přidání dalších atributů, mezi navrhovanými byly tituly členů nebo další detaily o hlasováních. Některé atributy by naopak měly být zřejmě nepovinné, např. číslo funkčního období mnoho municipalit vůbec nepoužívá. Z technických návrhů je možné uvést zavedení nástrojů pro databázové migrace nebo diskutovaný přechod na používání jazyka TypeScript pro tvorbu uživatelského rozhraní.

Literatura

- [1] AMAZON. *What is a Web Application?* [online]. 2024 [cit. 24. 1. 2024]. Dostupné z: <https://aws.amazon.com/what-is/web-application/>.
- [2] A.S.PARTNER. *Uživatelská příručka H.E.R. Systém* [online]. 2020 [cit. 3. 1. 2024]. Dostupné z: <https://www.bitest.cz/wp-content/uploads/2020/01/U%C5%BEivatelsk%C3%A1-p%C5%99%C3%ADtu%C4%8Dka.pdf>.
- [3] BITEST. *Webové stránky společnosti BitEST* [online]. 2024 [cit. 3. 1. 2024]. Dostupné z: <https://www.bitest.cz/>.
- [4] BIČÁK, O. *Územní samospráva České republiky, komparace s vybranými státy Evropy*. Praha, CZ, 2019. Bakalářská práce. Vysoká škola ekonomická v Praze, Národohospodářská fakulta. Vedoucí práce: Tomáš Louda. Dostupné z: <https://vskp.vse.cz/77726>.
- [5] BOSCH SECURITY SYSTEMS. *Bosch DCN Conference Software Suite Software Manual* [online]. Version 4.6, 2016. únor 2017 [cit. 4. 1. 2024]. Dostupné z: https://resources-boschsecurity-cdn.azureedge.net/public/documents/DCN_SW_SWM_Configuration_Manual_enUS_1220433323.pdf.
- [6] BOSCH SECURITY SYSTEMS. *DCN-SWSMV-E Synoptic microphone & voting license* [online]. V8. únor 2019 [cit. 4. 1. 2024]. Dostupné z: https://resources-boschsecurity-cdn.azureedge.net/public/documents/DCN_SWSMV_E_Data_sheet_enUS_68071792651.pdf.
- [7] BOSCH SECURITY SYSTEMS. *References* [online]. Září 2023 [cit. 4. 1. 2024]. Dostupné z: https://media.boschsecurity.com/fs/media/pb/media/news_2/customer_stories_1/Global_ST_references.pdf.
- [8] BROWN, E. *Web Development with Node and Express: Leveraging the JavaScript Stack*. 2. vyd. O'Reilly Media, 2019. ISBN 978-1-492-05351-4.
- [9] CENKOVÁ, D. *Vztahy obcí a krajů*. Brno, CZ, 2016. Bakalářská práce. Masarykova univerzita, Právnická fakulta. Vedoucí práce: Petr Kolman. Dostupné z: <https://is.muni.cz/th/wss8m/>.
- [10] ČESKO. Hlava 7 ústavního zákona č. 1/1993 Sb., ústava České republiky – znění od 1. 6. 2013. In: *Sbírka zákonů České republiky*. 1993, částka 1, s. 15. ISSN 1211-1244.
- [11] ČESKO. Část 1 ústavního zákona č. 347/1997 Sb., o vytvoření vyšších územních samosprávných celků a o změně ústavního zákona České národní rady č. 1/1993 Sb. In: *Sbírka zákonů České republiky*. 1997, částka 114, s. 7018. ISSN 1211-1244.

- [12] ČESKO. Zákon č. 106/1999 Sb., o svobodném přístupu k informacím – znění od 1. 1. 2024. In: *Sbírka zákonů České republiky*. 1999, částka 39, s. 2578–2582. ISSN 1211-1244.
- [13] ČESKO. Zákon č. 128/2000 Sb., o obcích (obecní zřízení) – znění od 1. 7. 2023. In: *Sbírka zákonů České republiky*. 2000, částka 38, s. 1737–1764. ISSN 1211-1244.
- [14] ČESKO. Zákon č. 129/2000 Sb., o krajích (krajské zřízení) – znění od 1. 2. 2022. In: *Sbírka zákonů České republiky*. 2000, částka 38, s. 1765–1782. ISSN 1211-1244.
- [15] ČESKO. Zákon č. 130/2000 Sb., o volbách do zastupitelstev krajů a o změně některých zákonů – znění od 2. 3. 2019. In: *Sbírka zákonů České republiky*. 2000, částka 38, s. 1783–1799. ISSN 1211-1244.
- [16] ČESKO. Zákon č. 131/2000 Sb., o hlavním městě Praze – znění od 1. 7. 2023. In: *Sbírka zákonů České republiky*. 2000, částka 39, s. 1802–1834. ISSN 1211-1244.
- [17] ČESKO. Zákon č. 491/2001 Sb., o volbách do zastupitelstev obcí a o změně některých zákonů – znění od 2. 8. 2021. In: *Sbírka zákonů České republiky*. 2001, částka 178, s. 11002–11024. ISSN 1211-1244.
- [18] ČESKÝ STATISTICKÝ ÚŘAD. *Malý lexikon obcí České republiky – 2023: Metodické vysvětlivky*. 320199-23. Prosinec 2023. Verze 1.0. Dostupné z: <https://www.czso.cz/csu/czso/maly-lexikon-obci-ceske-republiky-2023>.
- [19] DANES JE NOV DAN. *Webové stránky projektu Parlameter* [online]. 2024 [cit. 19. 1. 2024]. Dostupné z: <https://parlameter.org/>.
- [20] FELKE-MORRIS, T. A. *Web Development and Design Foundations with HTML5*. 8. vyd. Pearson, 2016. ISBN 978-0-13-432275-9.
- [21] FIŠER, D., LJUBEŠIĆ, N. a ERJAVEC, T. Parlameter – a Corpus of Contemporary Slovene Parliamentary Proceedings. *Contributions to Contemporary History*. Červen 2019, sv. 59, č. 1, s. 70–98. ISSN 2463-7807. Dostupné z: <https://doi.org/10.51663/pnz.59.1.04>.
- [22] GUPTILL, S. C. Metadata and data catalogues. In: LONGLEY, P. A., GOODCHILD, M. F., MAGUIRE, D. J. a RHIND, D. W., ed. *Geographical Information Systems: Principles, Techniques, Management and Applications*. 2. vyd. Wiley, 2005, sv. 2, s. 677–692. ISBN 978-0-471-73545-8.
- [23] HAVLÍKOVÁ, H. *Působnost obcí a krajů*. Praha, CZ, 2014. Diplomová práce. Univerzita Karlova, Právnická fakulta. Vedoucí práce: Martin Kopecký. Dostupné z: <https://dspace.cuni.cz/handle/20.500.11956/69034>.
- [24] HUG, S. Selection Effects in Roll Call Votes. *British Journal of Political Science*. Cambridge University Press. 2010, sv. 40, č. 1, s. 225–235. DOI: 10.1017/S0007123409990160. ISSN 1469-2112. Dostupné z: <https://doi.org/10.1017/S0007123409990160>.
- [25] HUG, S. Parliamentary Voting. In: MÜLLER, W. C. a NARUD, H. M., ed. *Party Governance and Party Democracy*. New York, NY: Springer, 2013, s. 137–157. ISBN 978-1-4614-6588-1. Dostupné z: https://doi.org/10.1007/978-1-4614-6588-1_8.

- [26] HÁSEK, R. *Metodické doporučení k činnosti územních samosprávných celků: č. 1.2 – Jednací řády zastupitelstev obcí*. Ministerstvo vnitra České republiky, odbor veřejné správy, dozoru a kontroly, 2022. Podle právního stavu k 1. 5. 2022. Dostupné z: <https://www.mvcr.cz/odk2/clanek/metodicke-materialy-k-zakonnym-zmocnenim.aspx>.
- [27] JOHNSON, J. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. 1. vyd. Morgan Kaufmann, 2010. ISBN 978-0-12-375030-3. Dostupné z: <https://doi.org/10.1016/C2009-0-20318-7>.
- [28] KARVONEN, K. The Beauty of Simplicity. In: Association for Computing Machinery. *Proceedings on the 2000 Conference on Universal Usability*. 2000, s. 85–90. CUU '00. ISBN 978-1-58113-314-1. Dostupné z: <https://doi.org/10.1145/355460.355478>.
- [29] KONSTANT, J. *Proces zveřejňování otevřených dat v Brně*. Brno, CZ, 2021. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce: Barbora Bührenová. Dostupné z: <https://is.muni.cz/th/l6ux7/>.
- [30] KRUGER, R., BROSENS, J. a HATTINGH, M. A Methodology to Compare the Usability of Information Systems. In: HATTINGH, M., MATTHEE, M., SMUTS, H., PAPPAS, I., DWIVEDI, Y. K. et al., ed. *Responsible Design, Implementation and Use of Information and Communication Technology*. Cham: Springer, 2020, s. 452–463. Conference on e-Business, e-Services and e-Society. ISBN 978-3-030-45002-1. Dostupné z: https://doi.org/10.1007/978-3-030-45002-1_39.
- [31] KRÁL, J. *Informační systémy: Specifikace, realizace, provoz*. 1. vyd. Science, 1998. ISBN 978-80-86083-00-1.
- [32] LAUDON., K. C. a LAUDON, J. P. *Management Information Systems: Managing the Digital Firm*. 12. vyd. Prentice Hall, 2012. ISBN 978-0-13-214285-4.
- [33] LNĚNÍČKA, M. a NIKIFOROVA, A. Transparency-by-design: What is the role of open data portals? *Telematics and Informatics*. Srpen 2021, sv. 61, č. 101605. ISSN 0736-5853. Dostupné z: <https://doi.org/10.1016/j.tele.2021.101605>.
- [34] LONG, E. *Election Data Visualisation*. Plymouth, UK, 2013. Disertační práce. University of Plymouth, Faculty of Arts, Humanities and Business. Dostupné z: <https://dx.doi.org/10.24382/4332>.
- [35] MAALI, F., CYGANIAK, R. a PERISTERAS, V. Enabling Interoperability of Government Data Catalogues. In: WIMMER, M. A., CHAPPELET, J.-L., JANSSEN, M. a SCHOLL, H. J., ed. *Electronic Government: 9th International Conference*. Berlin, Heidelberg: Springer, 2010, s. 339–350. EGOV. ISBN 978-3-642-14799-9. Dostupné z: https://doi.org/10.1007/978-3-642-14799-9_29.
- [36] MARGHESCU, D. Usability Evaluation of Information Systems: A Review of Five International Standards. In: WOJTKOWSKI, W., WOJTKOWSKI, G., LANG, M., CONBOY, K. a BARRY, C., ed. *Information Systems Development: Challenges in Practice, Theory, and Education*. Boston, MA: Springer, 2009, sv. 1, s. 131–142. ISBN 978-0-387-68772-8. Dostupné z: https://doi.org/10.1007/978-0-387-68772-8_11.

- [37] MEZEI, R. A. *Introduction to the Development of Web Applications Using ASP .Net (Core) MVC*. 1. vyd. Cham: Springer, 2023. Synthesis Lectures on Computer Science. ISBN 978-3-031-30626-6. Dostupné z: <https://doi.org/10.1007/978-3-031-30626-6>.
- [38] MINISTR SYSTEMS. *Webové stránky společnosti Ministr Systems* [online]. 2024 [cit. 3. 1. 2024]. Dostupné z: <https://www.ministr.cz/>.
- [39] MÍŠEK, J. *Nové povinnosti pro obce, kraje a orgány státní správy v oblasti otevřených dat* [online]. 3. srpna 2021 [cit. 4. 1. 2024]. Dostupné z: <https://data.gov.cz/%C4%8D1%C3%A1nky/nov%C3%A9-povinnosti-pro-obce-kraje-a-org%C3%A1ny-st%C3%A1tn%C3%AD-spr%C3%A1vy-v-oblasti-otev%C5%99en%C3%BDch-dat>.
- [40] NIELSEN, J. *Usability Engineering*. 1. vyd. AP Professional, 1993. ISBN 978-0-12-518406-9. Dostupné z: <https://doi.org/10.1016/C2009-0-21512-1>.
- [41] PEREIRA, G. V., MACADAR, M. A., LUCIANO, E. M. a TESTA, M. G. Delivering public value through open government data initiatives in a Smart City context. *Information Systems Frontiers*. Duben 2017, sv. 19, č. 2, s. 213–229. DOI: 10.1007/s10796-016-9673-7. Dostupné z: <https://link.springer.com/article/10.1007/s10796-016-9673-7>.
- [42] POLÁKOVÁ, N. *Otevřená data ve veřejné správě v České republice*. Liberec, CZ, 2016. Diplomová práce. Technická univerzita v Liberci, Ekonomická fakulta. Vedoucí práce: Šárka Laboutková. Dostupné z: <https://dspace.tul.cz/handle/15240/47118>.
- [43] PUBLICATIONS OFFICE OF THE EUROPEAN UNION, PAGE, M., HAJDUK, E., LINCKLAEN ARRIËNS, E. N., CECCONI, G. et al. *Open data maturity report 2023*. Publications Office of the European Union, 2023. Dostupné z: <https://data.europa.eu/doi/10.2830/384422>.
- [44] RADA, D. *Obec a její orgány v českém právním systému*. Brno, CZ, 2016. Rigorózní práce. Masarykova univerzita, Právnická fakulta. Dostupné z: <https://is.muni.cz/th/lio8t/>.
- [45] RANGRAZ JEDDI, F., NABOVATI, E., BIGHAM, R. a KHAJOU EI, R. Usability evaluation of a comprehensive national health information system: relationship of quality components to users' characteristics. *International Journal of Medical Informatics*. Elsevier. Leden 2020, sv. 133, č. 104026. ISSN 1386-5056. Dostupné z: <https://doi.org/10.1016/j.ijmedinf.2019.104026>.
- [46] RICHARDSON, L. a RUBY, S. *RESTful Web Services*. 1. vyd. O'Reilly Media, 2007. ISBN 978-0-596-52926-0.
- [47] ŠIMKOVÁ, P., SLAVÍČEK, P., KADLECOVÁ, K., URBÁNKOVÁ, A., KRUMLOVÁ, J. et al. *Metodické doporučení k činnosti územních samosprávných celků: č. 9 – Nejčastější nedostatky při výkonu samostatné působnosti obcí*. Ministerstvo vnitra České republiky, odbor dozoru a kontroly veřejné správy, 2013. Podle právního stavu k 1. 1. 2013. Dostupné z: <https://www.mvcr.cz/odk2/clanek/metodicke-materialy-k-zakonnym-zmocnenim.aspx>.
- [48] STAIR, R. a REYNOLDS, G. *Principles of Information Systems*. 9. vyd. Cengage Learning, 2009. ISBN 978-0-324-66528-4.

- [49] ÚŘAD PRO OCHRANU OSOBNÍCH ÚDAJŮ. *Stanovisko č. 6/2006 - Nahlížení do kandidátních listin a poskytování informací o kandidátech voleb do obecních zastupitelstev*. 2006. Poslední aktualizace únor 2014. Dostupné z: https://old.uoou.cz/assets/File.ashx?id_org=200144&id_dokumenty=22288.
- [50] WANG, V. a SHEPHERD, D. Exploring the extent of openness of open government data – A critique of open government datasets in the UK. *Government Information Quarterly*. Leden 2020, sv. 37, č. 101405. ISSN 1872-9517. Dostupné z: <https://doi.org/10.1016/j.giq.2019.101405>.
- [51] ZAKLOVÁ, K., HYNEK, J. a HRUŠKA, T. Towards Transparent Governance: Unifying City Councils Decision-Making Data Processing and Visualization. In: ROCHA, Á., ADELI, H., DZEMYDA, G., MOREIRA, F. a PONISZEWSKA MARAÑDA, A., ed. *Good Practices and New Perspectives in Information Systems and Technologies*. Cham: Springer, 2024, s. 402–411. ISBN 978-3-031-60221-4.
- [52] ZAKLOVÁ, K. *Analýza a vizualizace dat z hlasování Zastupitelstva města Brna*. Brno, CZ, 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce: Jiří Hynek. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/146480>.
- [53] ZBOŽÍNKOVÁ, Š. *Tvorba otevřených formálních norem: případová studie*. Brno, CZ, 2022. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce: Leonard Walletzký. Dostupné z: <https://is.muni.cz/th/c84we/>.

Příloha A

Koncové body REST API

Seznam koncových bodů API přidaných do systému. Odpovídá stavu finální verze systému včetně úprav v průběhu implementace a zpětné vazby z testování. Pro některé koncové body na přidání entit jsou uvedeny i parametry. Nejedná se však o kompletní specifikaci každého koncového bodu v systému, pouze o výčet dostupné funkcionality.

Pro každý koncový bod je uvedena cesta. Pro reprezentaci parametrů v cestě je použita stejná notace jako pro definici cest pomocí balíčku `Flask-RESTful`. Ve špičatých závorkách je uvedeno jméno parametru a před znakem dvojtečky případný typ, pokud není uveden, jedná se o řetězec. Na začátku cest je vynechán prefix `/flask`, který je stejný pro všechny koncové body.

Vizualizace

Koncové body sloužící pro vizualizační část systému.

Aplikace

- `/appInfo`
 - GET: Údaje týkající se celé aplikace, např. název a popis.

Municipalita

- `/municipalities`
 - GET: Seznam všech municipalit, pro výběr municipality v rámci vizualizace.
- `/municipality/<municipality_id>`
 - GET: Údaje o jedné municipalitě, nahrazuje původní konstanty pro město Brno.
- `/municipality/<municipality_id>/logo`
 - GET: Načtení loga municipality.

Administrační rozhraní

Koncové body pro novou administrační část systému.

Municipalita

- `/admin/municipality`
 - GET: Seznam všech municipalit.
 - POST: Přidat municipalitu včetně zastupitelstva. Přidá záznam o municipalitě do centrální databáze a vytvoří a inicializuje novou databázi pro zastupitelstvo municipality. Dostupné pouze pro administrátora.
 - * Parametry: textový identifikátor, typ municipality, název, počet zastupitelů, kontaktní email, URL, barva, logo, další textové atributy (např. popis aplikace).
- `/admin/municipality/<municipality_id>`
 - PUT: Upravit municipalitu. Používá stejné parametry jako přidat kromě textového identifikátoru.
 - DELETE: Smazat municipalitu.
- `/admin/municipality/<municipality_id>/logo`
 - POST: Vložit/nahradit logo municipality.

Typ municipality

- `/admin/municipality-type`
 - GET: Seznam všech typů municipalit pro výběr při vytváření municipality.

Politický subjekt

- `/admin/<municipality_id>/<body>/political-entity`
 - GET: Seznam politických subjektů.
 - POST: Přidat politický subjekt.
 - * Parametry: zkratka, název, barva.
- `/admin/<municipality_id>/<body>/political-entity/_<int:political_entity_id>`
 - PUT: Editovat politický subjekt, stejné parametry jako přidat.
- `/admin/<municipality_id>/<body>/political-entity/_<int:political_entity_id>/delete`
 - POST: Smazat politický subjekt. Možné pouze, pokud politický subjekt nemá žádné členy.

Člen

- `/admin/<municipality_id>/<body>/member`
 - GET: Seznam členů.
 - POST: Přidat člena.
 - * Parametry: jméno, příjmení, pohlaví.
- `/admin/<municipality_id>/<body>/member?type=meeting`
 - GET: Seznam členů zasedání a dostupných členů pro konkrétní datum. Slouží pro načtení členů při vytváření nového zasedání.
- `/admin/<municipality_id>/<body>/member/<int:member_id>`
 - GET: Načíst data člena.
 - PUT: Editovat člena, stejné parametry jako přidat.
 - DELETE: Smazat člena. Možné pouze, pokud nemá členství v žádném funkčním období (a nemohl se tak účastnit žádných hlasování).
- `/admin/<municipality_id>/<body>/member/order`
 - POST: Nastavit výchozí pořadí členů.
- `/admin/<municipality_id>/<body>/member/<int:member_id>/alias`
 - GET: Seznam aliasů člena.
 - POST: Přidat nový alias člena.
 - * Parametry: jméno, příjmení, datum začátku a konce platnosti.
- `/admin/<municipality_id>/<body>/alias/<int:alias_id>`
 - DELETE: Smazat alias člena.
- `/admin/<municipality_id>/<body>/member/<int:member_id>/membership`
 - GET: Seznam členství člena.

Funkční období

- `/admin/<municipality_id>/<body>/term`
 - GET: Seznam funkčních období.
 - POST: Přidat funkční období. Volitelně může být zadán i seznam členů se zvoleným politickým subjektem pro vytvoření členství pro funkční období.
 - * Parametry: pořadí, rok začátku, rok konce, název funkce lídra, lídr, seznam koaličních stran, seznam členů pro funkční období.
- `/admin/<municipality_id>/<body>/term/<int:term_id>`
 - GET: Načíst data funkčního období a členy příslušící k funkčnímu období.
 - PUT: Editovat funkční období, slouží pouze pro úpravu údajů o funkčním období, nemění členství členů.
 - * Parametry: rok začátku, rok konce, název funkce lídra, lídr, seznam koaličních stran

Zasedání

- `/admin/<municipality_id>/<body>/term/<int:term_id>/meeting`
 - GET: Seznam zasedání pro funkční období.
 - POST: Přidat nové zasedání. Součástí je seznam členů a jejich členství pro dané zasedání. Musí obsahovat výběr každého člena, který má členství ke zvolenému datu zasedání.
 - * Parametry: číslo, datum, URL záznamu zasedání, seznam členů.
- `/admin/<municipality_id>/<body>/meeting/<int:meeting_id>`
 - GET: Načíst data zasedání a seznam členů zasedání.
 - PUT: Editovat existující zasedání. Upravuje pouze entitu zasedání, kromě seznamu členů používá stejné parametry jako přidat. Změna data je možná pouze, pokud by nedošlo ke změně členů (tzn. nové datum stále patří do stejného intervalu členství členů).
- `/admin/<municipality_id>/<body>/meeting/<int:meeting_id>/member`
 - POST: Přidat členy na zasedání. Možné pouze, pokud zasedání neobsahuje hlasování.
- `/admin/<municipality_id>/<body>/meeting/<int:meeting_id>/member/_<int:member_id>`
 - DELETE: Odebrat člena ze zasedání. Možné pouze, pokud zasedání neobsahuje hlasování.
 - PATCH: Změnit členství člena v politickém subjektu k datu zasedání.

Hlasování

- `/admin/<municipality_id>/<body>/meeting/<int:meeting_id>/vote`
 - GET: Načíst seznam hlasování pro zasedání.
 - POST: Přidat nové hlasování na zasedání. Součástí je seznam hlasování členů se zvolenou hlasovací možností a příznak prezenčního hlasování.
 - * Parametry: číslo, čas, výsledek, předmět, validní, procedurální, tajné, URL hlasovacího protokolu, seznam hlasů a prezence členů.
- `/admin/<municipality_id>/<body>/vote/<int:vote_id>`
 - GET: Načíst data hlasování včetně hlasování členů.
 - PUT: Editovat hlasování, používá stejné parametry jako přidat.
 - DELETE: Smazat hlasování.

Uživatel

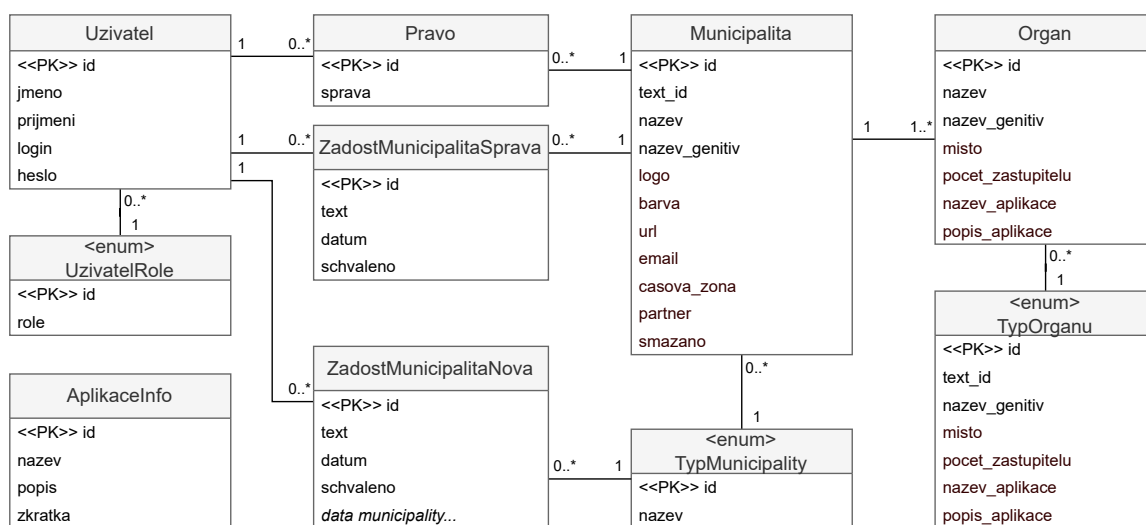
- /login
 - POST: Přihlášení uživatele.
- /refresh
 - POST: Obnovení JWT tokenů.
- /register
 - POST: Registrace nového uživatele.
 - * Parametry: login, heslo, jméno, příjmení.
- /admin/user
 - GET: Seznam uživatelů systému pro administrátora.
- /admin/user-permissions
 - GET: Načtení přístupových práv přihlášeného uživatele.
- /admin/profile
 - GET: Načíst profil přihlášeného uživatele.
 - PATCH: Upravit profil, změna jména nebo hesla.

Žádost

- /admin/request/manage-municipality
 - GET: Seznam uživateli podaných žádostí o správu municipality pro administrátora.
 - POST: Podat novou žádost o správu zvolené municipality.
- /admin/request/manage-municipality/<int:request_id>
 - PATCH: Vyřídít žádost administrátorem. V případě schválení je uživateli přidáno právo spravovat zvolenou municipalitu.
- /admin/request/add-municipality
 - GET: Seznam uživateli podaných žádostí o vytvoření nové municipality pro administrátora.
 - POST: Podat žádost o přidání nové municipality. Součástí požadavku je text žádosti a všechny povinné parametry pro vytvoření municipality.
- /admin/request/add-municipality/<int:request_id>
 - GET: Načíst žádost o přidání nové municipality. Pro zobrazení detailu případné nové municipality pro administrátora.
 - PATCH: Vyřídít žádost administrátorem. V případě schválení se provede přidání nové municipality a uživateli je nastaveno právo ji spravovat.

Příloha B

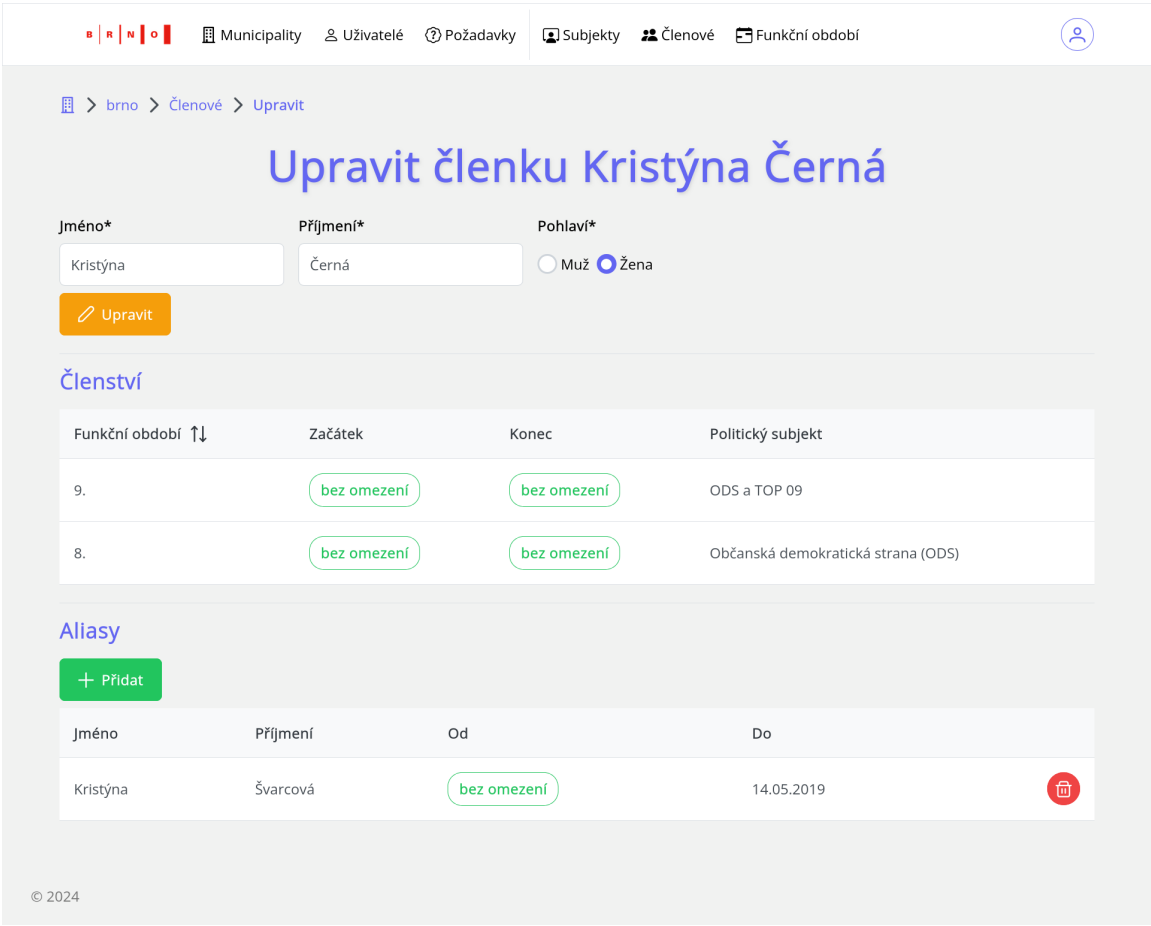
Schéma centrální databáze



Obrázek B.1: Schéma entit v centrální databázi, odpovídá stavu ve finální verzi systému včetně aktualizovaných názvů a změn provedených v průběhu implementace.

Příloha C

Snímky obrazovky uživatelského rozhraní



Obrázek C.1: Stránka detailu člena/členky s možností úpravy a tabulkami členství a aliasů.

Příloha D

Obsah přiloženého paměťového média

/	
src	zdrojové kódy systému
├── zmb-hlasovani/src	zdrojové kódy uživatelského rozhraní
│ ├── admin/	administrační část
├── api	zdrojové kódy Flask aplikace (API)
│ ├── admin/	balíček administrační části
│ ├── orm/	balíček se třídami a konstantami ORM
│ └── create_app.py	modul pro vytvoření aplikace
├── api_test/	testy API
├── migrations/	migrační skripty pro databáze municipalit
├── db-centralni.sql	skript pro vytvoření schématu centrální databáze
├── db-centralni-enums.sql	skript pro vložení číselníků do centrální databáze
├── text/	zdrojové soubory pro text práce a výsledný text ve formátu PDF
└── LICENCE.txt	soubor s licencí k vytvořenému zdrojovému kódu