



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**INTELIGENTNÍ DOPLŇOVÁNÍ SLOV V GUI PRO  
PŘEPISOVÁNÍ MATRIČNÍCH ZÁZNAMŮ**

INTELLIGENT WORD COMPLETING IN GUI FOR PARISH RECORDS REWRITING

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**ALEŠ POSTULKA**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. JAROSLAV ROZMAN, Ph.D.**

**BRNO 2019**

## Zadání bakalářské práce



21717

Student: **Postulka Aleš**  
Program: Informační technologie  
Název: **Inteligentní doplňování slov v GUI pro přepisování matričních záznamů**  
**Intelligent Word Completing in GUI for Parish Records Rewriting**  
Kategorie: Umělá inteligence  
Zadání:

1. Nastudujte problematiku tzv. našeptávačů slov. Seznamte se se strukturou matričních záznamů.
2. Navrhněte systém, který bude standardně doplňovat jednotlivá slova podle jazyku matriky a podle jejich četnosti. Dále navrhněte systém, který na základě co nejmenšího množství údajů (např. č.p. nebo jméno a příjmení otce) doplní celý záznam. Vycházejte opět podle četnosti, ale např. i podle data naposledy narozeného dítěte.
3. Navržený systém implementujte jako součást GUI pro přepisování matričních záznamů.
4. Výsledný systém nechte otestovat a zhodnoťte úsporu času pro jeden záznam. Navrhněte případná vylepšení.

### Literatura:

- Autocomplete, <https://jqueryui.com/autocomplete/>
- Lednická B., Sestavte si rodokmen, Grada, ISBN: 978-80-247-4069-0

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 1. listopadu 2018

## Abstrakt

Tato práce se zabývá vytvořením systému inteligentního doplňování slov pro přepis matričních záznamů. Systém je implementován jako součást grafického uživatelského rozhraní webové aplikace DEMOS. Hlavním úkolem systému je zobrazování návrhů slov pro jednotlivé položky záznamu, inteligentní doplňování částí matričního záznamu, zobrazování návrhů údajů k jednotlivým osobám záznamu a zobrazování návrhů obsahujících údaje vhodné k doplnění celého matričního záznamu.

## Abstract

This thesis is about creation of an intelligent word completion system for parish records rewriting. System is implemented as a part of graphical user interface of web application DEMOS. Main task of this system is displaying word suggestions for an individual record items, intelligent populating parts of parish record, displaying data suggestions for individual people of record and displaying suggestions which contains data suitable to populate the whole parish record.

## Klíčová slova

genealogie, matrika, přepis matričních záznamů, DEMOS, inteligentní doplňování slov, databáze, AJAX

## Keywords

genealogy, parish book, parish records rewriting, DEMOS, intelligent word completion, database, AJAX

## Citace

POSTULKA, Aleš. *Inteligentní doplňování slov v GUI pro přepisování matričních záznamů*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

# Intelligentní doplňování slov v GUI pro přepisování matričních záznamů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rozmana, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Aleš Postulka  
8. května 2019

## Poděkování

Tímto bych chtěl poděkovat panu Ing. Jaroslavu Rozmanovi, Ph.D. za ochotu, cenné rady a věcné připomínky při vypracování této práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Studium problematiky</b>	<b>4</b>
2.1	Genealogie . . . . .	4
2.2	Matriky . . . . .	5
2.3	Digitalizace matrik . . . . .	7
2.4	Projekt DEMoS . . . . .	8
2.5	Vstupní pole s automatickým dokončováním . . . . .	8
2.6	Inteligentní vyplňování formuláře . . . . .	9
2.7	Serverové webové technologie . . . . .	10
2.8	Klientské webové technologie . . . . .	11
2.9	Formát JSON . . . . .	13
2.10	Schéma databáze . . . . .	15
<b>3</b>	<b>Analýza požadavků a návrh řešení</b>	<b>17</b>
3.1	Cíl práce . . . . .	17
3.2	Výběr podle jazyka matriky . . . . .	17
3.3	Vstupní pole s automatickým dokončováním . . . . .	18
3.4	Inteligentní vyplňování záznamu . . . . .	18
<b>4</b>	<b>Implementace a testování</b>	<b>20</b>
4.1	Použité nástroje . . . . .	20
4.2	Komunikace mezi klientem a serverem . . . . .	21
4.3	Vstupní pole s automatickým dokončováním . . . . .	22
4.4	Inteligentní vyplňování záznamu . . . . .	24
4.5	Zobrazování návrhů osob . . . . .	26
4.6	Zobrazení návrhů na celý záznam . . . . .	27
4.7	Testování . . . . .	28
4.8	Návrhy na vylepšení . . . . .	30
<b>5</b>	<b>Závěr</b>	<b>31</b>
	<b>Literatura</b>	<b>32</b>
<b>A</b>	<b>Schéma databáze</b>	<b>34</b>
<b>B</b>	<b>Návod k použití</b>	<b>35</b>
<b>C</b>	<b>Testovací protokol</b>	<b>38</b>

<b>D Dotazník</b>	<b>40</b>
<b>E Obsah přiloženého paměťového média</b>	<b>42</b>

# Kapitola 1

## Úvod

Pátrání po rodinné historii v posledních letech mezi lidmi nabírá na popularitě. Lidé po svých předcích pátrají ze dvou důvodů. Někteří pátrají pouze kvůli poznání co nejvzdálenější historie svého rodu. Jiní mohou svou rodovou linii zkoumat za účelem dosažení zisku z dědictví a podobně. Velké množství veřejně přístupných zdigitalizovaných matrik toto bádání značně usnadňuje, protože umožňuje pátrat po předcích z pohodlí domova. I přesto může být zkoumání složité a náročné, kvůli špatné čitelnosti historických matrik, které byly psány ručně. Obecně lze říci, že čím je matrika starší, tím je hůře čitelná.

Za účelem zjednodušení genealogického výzkumu vzniká projekt *Možnosti tvorby komunitní genealogické databáze se sémantickou informací a zahrnutou neurčitostí* (dále označován názvem *DEMoS*), který vzniká ve spolupráci Vysokého učení technického v Brně a Masarykovy univerzity. Cílem projektu DEMoS je vytvoření webové aplikace sloužící pro přepis matrik a dalších historických knih a následnou automatickou tvorbu rodokmenů.

Přepis matričních záznamů je běh na dlouhou trať. Od poloviny 16. století, kdy matriky začaly vznikat, je matrik a záznamů v nich velké množství. S rostoucím množstvím informací v jednotlivých záznamech roste také čas, který je potřeba pro přepis jednoho záznamu. Za účelem snížení doby přepisu jednotlivých matričních záznamů vznikla tato práce. Jejím cílem je s co největší přesností navrhnout hodnoty pro jednotlivé položky záznamu, a také inteligentně doplňovat, či navrhnout údaje ke zbývajícím částem záznamu na základě již vyplněných hodnot.

První část práce je věnována studiu teorie, která byla potřeba k vytvoření implementace. Tato část obsahuje popis pojmu genealogie, a také získané informace o samotných matrikách. Část kapitoly je věnovaná digitalizaci matrik a již zmíněnému projektu DEMoS. Dále je v kapitole popsána problematika tzv. našeptávačů slov a inteligentního vyplňování formulářů. Poté je zde zmínka o vybraných webových technologiích na straně serveru i na straně klienta, a také o formátu JSON používaném k přenosu dat mezi klientskou a serverovou částí. Na konci kapitoly je popsána struktura databáze sloužící pro ukládání matričních záznamů narozených.

Na počátku druhé části je definován cíl práce. Dále je v této kapitole uveden návrh výběru podle jazyka matriky. Poté je zde popsán návrh systému automatického dokončování slov jednotlivých položek záznamu a návrh systému pro inteligentní doplňování částí záznamu na základě již vyplněných údajů.

Poslední část práce je zaměřena na popis implementace již zmíněných systémů a jejich testování. V kapitole je popsán také postup pro modifikaci těchto systémů na matriky sňatků a úmrtí. Poslední část této kapitoly je věnována návrhům na možná vylepšení implementovaných systémů.

## Kapitola 2

# Studium problematiky

V této kapitole jsou uvedeny teoretické informace, které bylo potřeba nastudovat pro další práci na projektu. Na začátku kapitoly je popsán pojem genealogie, včetně souvisejících pojmů, následován popisem pojmu matrika. Dále je popsáno vstupní pole s automatickým dokončováním a inteligentní vyplňování formuláře. Následuje popis serverových a klientských webových technologií a popis schématu databáze pro uchovávání matričních záznamů. V závěru kapitoly je popsán formát JSON, který je používán k přenosu dat mezi serverovou a klientskou částí webové aplikace.

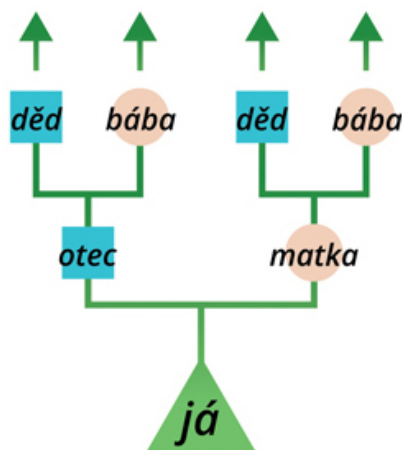
### 2.1 Genealogie

Název *Genealogie* [12] pochází ze dvou slov řeckého původu. Jedná se o slova *génos* – rod a *lógos* – věda. Genealogie je tedy vědou o rodu, rodině. Úkolem této vědy je zkoumání vztahů mezi lidmi, kteří mají společný původ. Zkoumány jsou především příbuzenské vztahy mezi lidmi v rámci společnosti.

Genealogii lze rozlišit na dvě skupiny, a to genealogii vědeckou a genealogii soukromou. Vědecká genealogie se zabývá rodinnými vazbami a zkoumá dlouhodobý vývoj těchto vazeb. Má přímě vazby na další vědecké obory, které se zabývají historií, jimž poskytuje materiály k dalšímu výzkumu. Soukromá genealogie, nazývaná také jako *rodopis*, je omezená na jednotlivce, kteří zkoumají historii svého rodu. Jedním z důvodů tohoto zkoumání může být například poznání historických kořenů vlastní rodiny.

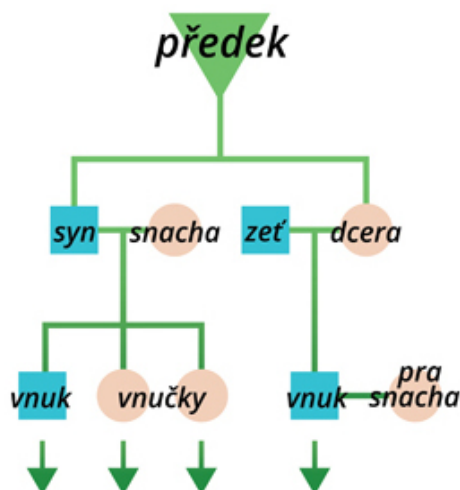
Při genealogickém pátrání existují dva způsoby postupu. Prvním způsobem je *vývod* (viz obrázek 2.1), kdy jsou hledáni všichni předci výchozí osoby. Při vývodu je tedy postupováno od současnosti zpět do minulosti po všech rodových liniích.





Obrázek 2.1: Schéma vývodu (převato z [14]).

Druhým, a také nejsložitějším, způsobem je *rozrod* (viz obrázek 2.2), kdy jsou zkoumáni všichni potomci nejstaršího dohledaného předka. Zkoumána je jak mužská, tak ženská linie potomků. V některých případech lze pozorovat postupné vymírání rodu, či naopak jeho neustálé rozrůstání.



Obrázek 2.2: Schéma rozrodu (převato z [14]).

## 2.2 Matriky

*Matriky* [12][15] jsou písemným genealogickým pramenem. Jedná se o úřední knihy určené k evidenci občanů. Protože obsahují všechny důležité životopisné údaje, jsou matriky hlavním zdrojem při genealogickém bádání. Do matrik je zapisováno postupně v průběhu času, a tak jsou záznamy řazeny chronologicky.

Matriky jsou rozděleny na tři typy. Prvním typem jsou matriky narozených (viz obrázek 2.3), které většinou obsahují datum a adresu narození, datum křtu a jméno dítěte, dále pak obsahuje jméno, místo narození a povolání rodičů. Matriky narozených mohou obsahovat také například údaje o pohlaví, nebo o legitimitě (informace, zda bylo dítě narozené ve svazku manželském), porodní bábě, křtiteli dítěte a jeho kmotrech.

Blansko										
Kniha narozených a pokřtěných. — Geburts- und Taufbuch.										
Annus: 1892										
tomus: X. A										
pag. 197										
Rak, měsíc a den narození; pak jméno křez- teř, který křtil	Číslo domu — Haus-Nr.	Jméno křtěnce	Rod Ge- schlecht	männlich weiblich unbekannt	männlich weiblich unbekannt	Rodičové — Eltern				Kmotrův — Páthen
						Otec:	Matka:	Mutter:		
Zeit der Geburt und Taufe, Monat, Tag; Hat getauft?	Name des Täuflings	Name des Täuflings				Tauf- und Familienname, Stand, und Wohnort, dann dessen Vaters Tauf- und Familienname, Stand und Wohnort, und der Mutter Tauf- und Familienname	Tauf- und Familienname, Stand und Wohnort, und der Mutter Tauf- und Familienname	Tauf- und Familienname, Stand und Wohnort, und der Mutter Tauf- und Familienname		Jméno, stav a bydliště
										Name, Stand und Wohnort
1892	236	Rudolf	1	1		Pelikán Rudolf	Magdalena, veve- řka Trávníčka, podsedníka ve Žďáře a ješoman želky Karoliny rozené Burg- heim			J. Frant. kř. biskup Rudolf obchodník v Brně v Blansku Otcovo jeho manželka

Obrázek 2.3: Ukázka záznamu v matrice narozených. Záznam pochází z Blanska z roku 1892 (převzato z [1]).

Dalším typem jsou matriky oddaných (viz obrázky 2.4), ve kterých je uvedeno datum uzavření sňatku, jména a místa narození oddáváných, údaje také o jejich rodičích, a také údaje o svědcích. Někdy lze v těchto matrikách nalézt také informace o povolání.

[illegible]

Obrázek 2.4: Ukázka záznamu v matrice oddaných. Záznam pochází z Blanska z roku 1936 (převzato z [1]).

Posledním typem jsou matriky zemřelých (viz obrázek 2.5). V těchto matrikách je uvedeno datum úmrtí, jméno a adresa zemřelého, příčina úmrtí, případně další údaje.

*Blansko.*  
**Kniha zemřelých.**

Annus: 1919.      tomus: IX.      pag: 1878

Číslo postupné	Doba úmrtí a pohřbu	Číslo domu	Jméno, příjmení, stav a místo narození zemřelého	Nábo- ženství			Stáří			Nemoc nebo způsob smrti	Zaopatřoval	Pochoval	Poznámka
				křesť.	nechr.	nechr.	let	měsíc	dne				
23	15/15 června	64	Škaroupková Marie, svobodná při psanosti v Blansku; manželka Martina Škaroupky, kotla- ře v Blansku, a Františky, rozn. Berdickové, manžel- ky 1895 v Blansku.	1	-	-	1	13	-	Tuberculosis pulmonum  období 1912-23.	Kuň byl rozn. Marekova pohřb.	František Zádla, krojo.	

Obrázek 2.5: Ukázka záznamu v matrice zemřelých. Záznam pochází z Blanska z roku 1919 (převzato z [1]).

Vzhled a náležitosti matrik se s časem postupně měnily. Záznamy v nejstarších matrikách jsou zapsány jako prostý text, v těch novějších jsou již použity předtištěné tabulky, do kterých jsou vepsány jednotlivé údaje. Množství údajů v záznamu je závislé jednak na době vzniku tohoto záznamu, ale také na osobě, která tento záznam zapisovala. Některé záznamy tak obsahují pouze základní informace, a některé záznamy naopak obsahují velké množství detailních informací.

Na území České republiky se v průběhu času měnil i jazyk, v němž byly matriky vedeny. První matriky jsou psány v latině, jež byla nahrazena němčinou a novější matriky jsou již vedeny v češtině.

První matriky na našem území pocházejí z poloviny 16. století. Od roku 1784 byly matriky úředním dokumentem a bylo přesně určeno, co mají jednotlivé záznamy v těchto matrikách obsahovat. Od roku 1870 jsou vedeny také civilní matriky vedené většinou u okresních soudů, které byly určeny osobám bez příslušnosti k církvi. V roce 1949 přešly všechny matriky plně do správy státu.

## 2.3 Digitalizace matrik

Digitalizace matrik [12][3] byla zahájena v roce 2007. Digitalizací je myšleno skenování originálních matričních listin. První digitalizované matriky byly zveřejněny SOA<sup>1</sup> v Třeboni roku 2008.

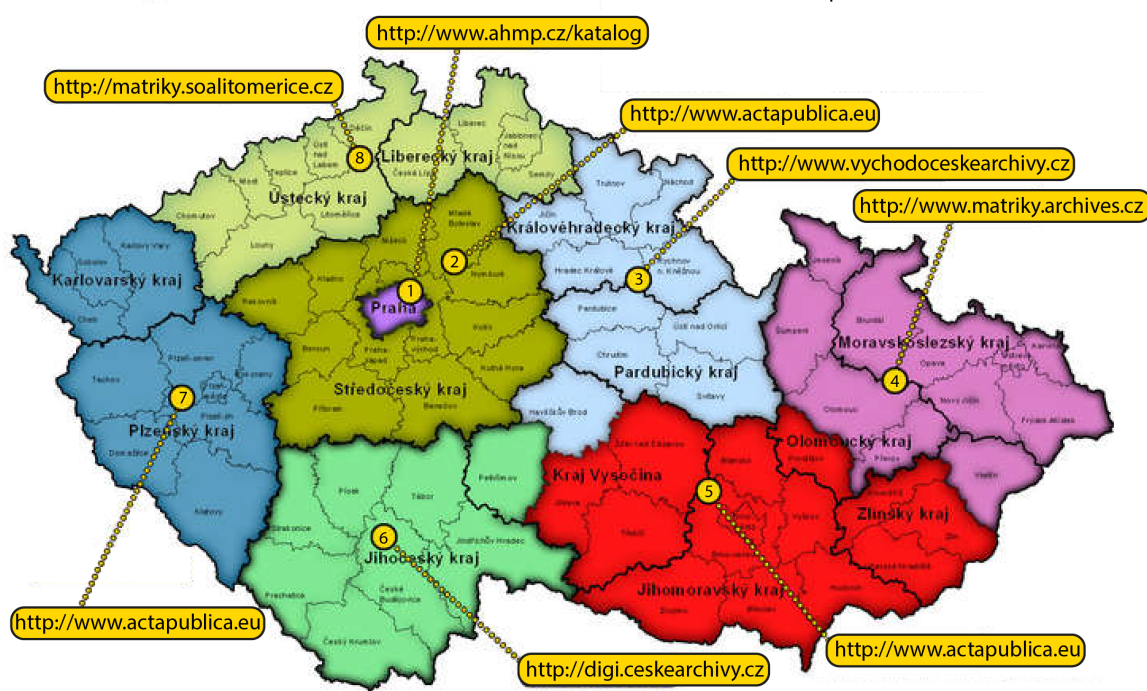
Na velkém množství dlouhodobých projektů digitalizace má podíl americká genealogická společnost *Genealogical Society of Utah*. Ostatní projekty jsou prováděny v jednotlivých oblastních archivech.

<sup>1</sup>SOA – Státní oblastní archiv

Všechny krajské archivy mají digitalizované matriky zveřejněny online, kde jsou dostupné veřejnosti, a tím usnadňují genealogický výzkum. Na obrázku 2.6 jsou zobrazeny odkazy na digitální archivy pro jednotlivé oblasti České republiky.

## Matriky na Internetu

- |                                  |   |
|----------------------------------|---|
| 1. Archiv hlavního města Prahy   | <a href="http://www.ahmp.cz">http://www.ahmp.cz</a>                       |
| 2. SOA v Praze                   | <a href="http://www.soapraha.cz">http://www.soapraha.cz</a>               |
| 3. SOA v Zámrsku                 | <a href="http://vychodoceskearchivy.cz">http://vychodoceskearchivy.cz</a> |
| 4. SOA v Opavě                   | <a href="http://archives.cz">http://archives.cz</a>                       |
| 5. Moravský zemský archiv v Brně | <a href="http://mza.cz">http://mza.cz</a>                                 |
| 6. SOA v Třeboni                 | <a href="http://ceskearchivy.cz">http://ceskearchivy.cz</a>               |
| 7. SOA v Plzni                   | <a href="http://soaplzen.cz">http://soaplzen.cz</a>                       |
| 8. SOA v Litoměřicích            | <a href="http://soalitomerice.cz">http://soalitomerice.cz</a>             |



Obrázek 2.6: Mapa s odkazy na digitální archivy (převzato z [1]).

## 2.4 Projekt DEMoS

Projekt *DEMoS* [5] vzniká ve spolupráci s Masarykovou univerzitou. Jeho cílem je vytvořit systém umožňující přepis, opravu a doplňování historických pramenů jako jsou matriky, lánové rejstříky, urbáře, atd. Systém má být zpracován tak, aby byl uživatelsky co nej-přívětivější a zjednodušil zadávání údajů. Dále má systém umožnit generování rodokmenů a dalších dat pro historicko-statistický a demografický výzkum. Doba řešení tohoto projektu je stanovena na roky 2018-2021.

## 2.5 Vstupní pole s automatickým dokončováním

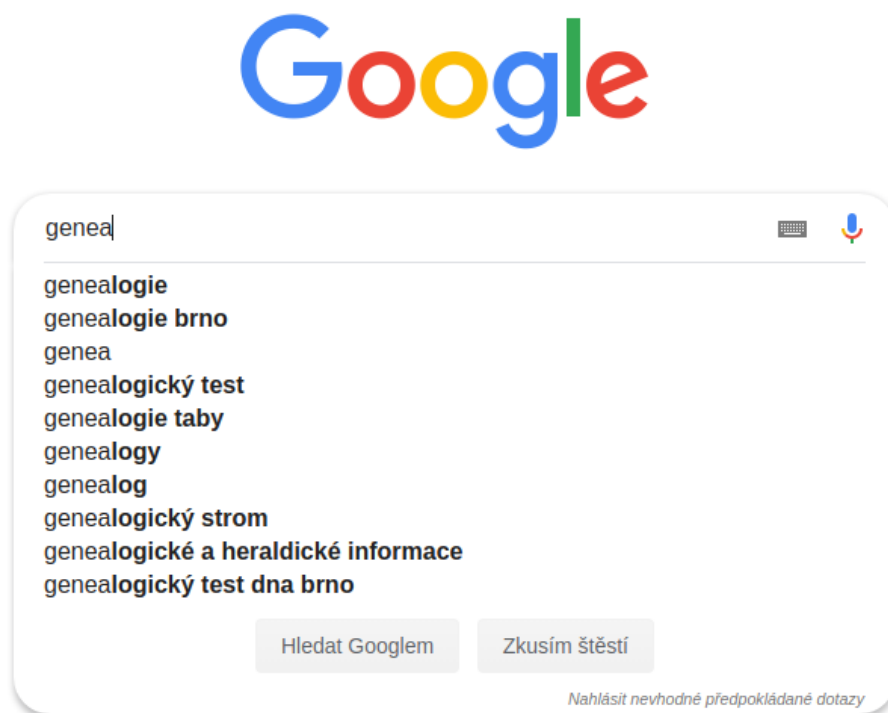
V současné době je nezbytnou součástí uživatelských rozhraní interaktivita. Často používaným interaktivním prvkem je *vstupní pole s automatickým dokončováním* [16][8]. Po zahájení psaní do takového pole začne podle aktuální hodnoty v poli vyhledávat vhodná slova, či fráze, k doplnění do tohoto pole. Jedná se tedy o schopnost automaticky předpovědět slovo či frázi, kterou chce uživatel vepsat do textového pole. Vyhledávání může probíhat



jak lokálně, tak vzdáleně pomocí zasílání požadavků na server. Komunikace se serverem je v tomto případě asynchronní a probíhá na pozadí.

Účelem automatického doplňování je vyhledávat co nejpřesnější návrhy k usnadnění a urychlení práce uživatele. Automatické doplňování má širokou škálu využití. Je používáno například ve webových prohlížečích, internetových vyhledávačích, e-mailových klientech, vývojových prostředích, nebo v operačních systémech. Ukázka vstupního pole s automatickým doplňováním je na obrázku 2.7.

Návrhy lze vyhledávat pouze podle počátečních písmen zadaných ve vstupním poli, ale také s využitím složitých algoritmů s umělou inteligencí, které zohledňují například aktuální polohu uživatele.



Obrázek 2.7: Ukázka vstupního pole s automatickým dokončováním ve vyhledávači Google.

## 2.6 Inteligentní vyplňování formuláře

*Intelligentní vyplňování formuláře* funguje na podobném principu jako vstupní pole s automatickým dokončováním, ovšem s tím rozdílem, že místo návrhů slov a frází pro jedno textové pole jsou vyplňována ostatní textová pole formuláře v závislosti na již vyplněných položkách.

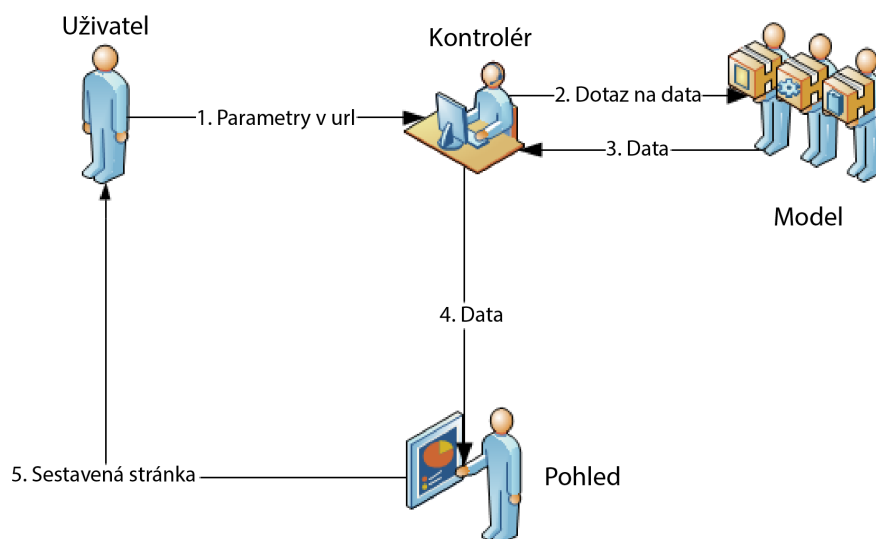
Při každé změně hodnoty libovolného vstupního pole formuláře, jsou aktuálně vyplněné údaje zaslány na server jako požadavek na vyhledání ostatních položek formuláře. Algoritmus vyhledávání návrhů je pak závislý na konkrétní aplikaci a formuláři. Po vrácení odpovědi mohou být nalezené výsledky automaticky doplněny do formuláře, nebo mohou být uživateli zobrazeny pouze návrhy a uživatel sám určí, které hodnoty mají být doplněny. Tyto možnosti mohou být kombinovány, nebo může být ponechána volba samotnému uživateli. V případech, kdy je často nalezeno více možností k doplnění je vhodnější volbou pouze zobrazování návrhů než přímé doplňování hodnot do formuláře.

## 2.7 Serverové webové technologie

V této části jsou popsány dva PHP frameworky určené k tvorbě serverových webových aplikací. Je zde zmíněn framework *Laravel*, který v současné době patří k nejpoužívanějším PHP frameworkům na světě. Dále pak *Nette Framework*, který je pro změnu nejpoužívanějším PHP frameworkem v České republice. Oba frameworky jsou založeny na architektuře MVC popsané v sekci 2.7.

### Architektura MVC

*MVC (Models-Views-Controllers)* [9][18] je softwarová architektura, která dělí aplikaci na 3 části. Modely (Models) slouží k reprezentaci a zpracování dat. Kontroléry (Controllers) se starají o práci uživatelského rozhraní a běh aplikace. Pohledy (Views) zajišťují grafické zobrazování dat a jejich prezentaci uživateli. Princip architektury je zobrazen na obrázku 2.8.



Obrázek 2.8: Princip architektury MVC (převzato z [18]).

### Laravel Framework

Jak již bylo zmíněno, *Laravel* [13] je jedním z nejvíce používaných PHP frameworků na světě. Jeho autorem je Taylor Otwell. Laravel je k dispozici jako open source s licencí MIT.

Mezi hlavní prvky tohoto frameworku patří *Eloquent ORM*<sup>2</sup> usnadňující práci s databází. Dále pak šablonovací systém *Blade*, možnost vytváření migrací a skriptů pro naplnění databáze. Jedním z mnoha dalších prvků je vestavěný HTTP server, který umožňuje vyvíjet webové aplikace lokálně bez nutnosti instalace vlastního serveru.

### Nette Framework

*Nette Framework* [6] je nejpoužívanějším frameworkem v České republice, kde je také vyvíjen. Jeho autorem je David Grudl, který je dodnes jedním z hlavních vývojářů. Nette je chráněno licencemi *New BSD* a *GNU GPL*.

<sup>2</sup>ORM – Object-relational mapping (Objektově relační mapování)

Nette klade hlavní důraz na produktivitu vývoje a bezpečnost aplikací. Je složeno z několika samostatně použitelných komponent. Z těch hlavních jsou to například šablonovací systém *Latte*, systém pro vytváření formulářů *Nette Forms*, ladící nástroj *Tracy*, nebo datábázovou vrstvu *Nette Database*. Za zmínku stojí také podpora AJAXu popsaného v sekci 2.8.

## 2.8 Klientské webové technologie

Pro práci s GUI<sup>3</sup> webové aplikace je potřeba používat také technologie, které vykonávají svou práci na straně klienta. V této části je popsána JavaScriptová knihovna jQuery, technologie AJAX a dvě knihovny pro tvorbu automatické predikce slov a zobrazování návrhů *jQueryUI Autocomplete widget* a *typeahead.js*.

### jQuery

*jQuery* [7] je knihovna založená na programovacím jazyku JavaScript. Jejím účelem je zjednodušení interakce mezi objektovým modelem HTML dokumentu a jazykem JavaScript. Hlavními vlastnostmi knihovny jQuery jsou zjednodušení manipulace s HTML dokumenty, reakce na události, animace a komunikace se serverem s využitím technologie AJAX.

Důraz je kladen především na postupné volání metod, neboli *řetězení*, kdy každá metoda vrací množinu elementů již lze použít k dalšímu zpracování. jQuery obsahuje funkce několika typů určených účelem, za kterým jsou tyto funkce využívány. Funkce jsou rozděleny na hlavní funkce, výběr, manipulace, procházení, kaskádové styly, atributy, události, efekty, AJAX a nástroje.

### AJAX

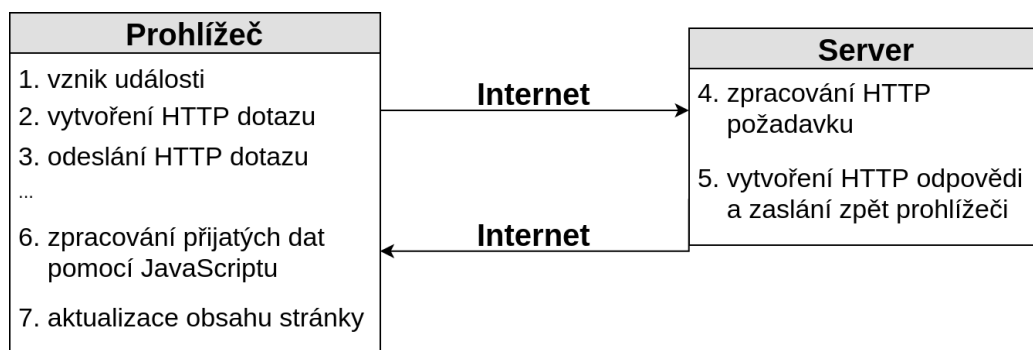
*AJAX*<sup>4</sup> [17][8] je technologie rozšiřující programovací jazyk JavaScript. Tato technologie umožňuje komunikaci klienta se serverem na pozadí. Požadavky jsou na server zasílány v případě potřeby získání dat ze serveru. Tímto je umožněno měnit jednotlivé části stránky bez nutnosti jejího obnovení. AJAX může být využíván při tvorbě interaktivních a inteligentních webových aplikací, které usnadňují práci jejich uživatelům. Pro použití této technologie není potřeba instalace žádných rozšíření, protože AJAX je implementován ve všech moderních webových prohlížečích.

Přestože písmeno *X* v názvu znamená XML, již tento formát není pro asynchronní komunikaci téměř využíván. XML bylo ve velké většině případů nahrazeno formátem *JSON* (popsaný v sekci 2.9), který je odlehčeným formátem určeným pro přenos dat. Pro přenos dat může být využit libovolný formát uložení. Data lze přenášet dokonce jako prostý text.

Po provedení předem určené události AJAX na straně klienta vytvoří HTTP požadavek a odešle jej na server. Požadavek je serverem zpracován a poté je vytvořena HTTP odpověď, která je zaslána zpět klientovi. Klient přijatou odpověď zpracuje a následně aktualizuje obsah stránky. V časovém intervalu mezi odesláním požadavku a přijetím odpovědi je klient plně funkční a neomezuje tak uživatele v jeho používání.

<sup>3</sup>GUI – Graphical User Interface (grafické uživatelské rozhraní)

<sup>4</sup>AJAX – Asynchronous JavaScript and XML (asynchronní JavaScript a XML)



Obrázek 2.9: Princip fungování AJAXu.

Knihovna jQuery obsahuje metodu `$.ajax()` [17][4], jejímž parametrem je asociativní pole s nastavením (viz výpis 2.1). Základní nastavení přijímaná metodou jsou následující:

- `method` – metoda HTTP požadavku (např. GET, POST, PUT); výchozí hodnotou je metoda GET,
- `url` – URL požadavku,
- `data` – data požadavku; může být použit libovolný formát,
- `success` – funkce, která je volána při úspěšném přijetí odpovědi,
- `error` – funkce, která je volána při výskytu chyby,
- kompletní seznam nastavení viz [4]

```
$.ajax({
  method: "POST",
  url: "/suggestName",
  data: {
    "name": "J",
    "surname": "Pakosta",
    "birth-date": "04-05-1880"
  },
  success: function (data) {
    alert(data);
  },
});
```

Výpis 2.1: Ukázka použití AJAXu s využitím knihovny jQuery.

## jQueryUI Autocomplete widget

*Autocomplete widget* [2] je funkce JavaScriptové knihovny *jQueryUI* určená k tvorbě seznamu návrhů pro automatické doplnění formulářového pole. Funkci autocomplete je možno použít pro elementy `<input>`, `<textarea>`, a také pro elementy, které obsahují atribut `contenteditable`.

Autocomplete widget umožňuje vyhledávat návrhy v lokálním zdroji (například pole hodnot), který je vhodný pro malé množství statických dat. Vyhledávání návrhů je však



také možné pomocí vzdáleného zdroje, tedy serveru a databáze, s využitím již zmíněné asynchronní komunikace AJAX.

Tento nástroj má také vestavěné ovládání pomocí klávesnice. Směrové klávesy nahoru a dolů jsou určeny k pohybu v seznamu návrhů, klávesa **Escape** slouží k uzavření seznamu, stisk klávesy **Enter** použije aktuálně zvolenou položku a uzavře seznam, klávesa **Tabulator** má stejné chování jako klávesa **Enter** pouze s rozdílem, že navíc zaměří další vstupní pole.

Je zde také možnost téměř libovolných úprav chování seznamu návrhů a automatického doplňování. Od vestavěných nastavení jako například `autoFocus`, `delay`, či `minLength` po vlastní nastavení zobrazování a reakcí na různé události.

## Knihovna `typeahead.js`

JavaScriptová knihovna `typeahead.js` [10] je vytvořena podle automatického doplňování při vyhledávání na stránkách [twitter.com](https://twitter.com).

Knihovna se skládá ze dvou částí, enginu pro vyhledávání návrhů *Bloodhound* a *Typeahead* pro zobrazování nalezených návrhů. Stejně jako předchozí knihovna může využívat lokální i vzdálený zdroj dat pro vyhledávání návrhů k doplnění. Obsahuje také vestavěné možnosti nastavení, ovšem vlastní úpravy chování již nejsou tak jednoduché jako u *Autocomplete widgetu*. Oproti němu je zde však možnost předvyplnění prvního návrhu rovnou do textového pole.

*Typeahead.js* má přednastavený vzhled, což znamená nutné změny vzhledu při použití ve vlastním projektu. Knihovna byla naposledy aktualizována v roce 2015 a v současné době již není vyvíjena což může způsobit nekompatibilitu s novějšími verzemi knihovny jQuery.

## 2.9 Formát JSON

*JSON* [17] je jednoduchý datový formát, který je založený na syntaxi polí a objektů v JavaScriptu. Ukázka uložení dat je ve výpisu 2.2. Vzhledem k použité syntaxi není potřeba žádný speciální postup ani při čtení dat z formátu JSON ani při jejich vytváření v tomto formátu. Takto uložená data jsou přístupná stejným způsobem jako data uložená v poli a lze k nim tedy přistupovat s využitím hranatých závorek.

```
{"classinfo":
  {
    "students": [
      {
        "name": "Michael Smith",
        "average": 99.5,
        "age": 17,
        "graduating": true
      },
      {
        "name": "Steve Johnson",
        "average": 34.87,
        "age": 17,
        "graduating": false
      },
      {
        "name": "Rebecca Young",
```

```

        "average": 89.6,
        "age": 18,
        "graduating": true
    }
]
}

```

Výpis 2.2: Uložení dat ve formátu JSON (převzato z [17]).

## Srovnání s XML

*XML* [17] je značkovací jazyk, to znamená, že každá informace je ohraničená počáteční a koncovou značkou nesoucí název dané informace. V praxi to znamená, že data přenášená ve formátu XML obsahují velké množství přebytečných znaků, které značně zvětšují velikost přenášených dat. Ve výpisech 2.2 a 2.3 jsou ukázky uložení shodných dat ve formátech XML a JSON. Data bez mezer ve formátu JSON mají velikost 224 bajtů a data ve formátu XML mají velikost 365 bajtů, tedy o 141 bajtů více než data uložená ve formátu JSON.

Snad jedinou výhodou XML oproti formátu JSON je pouze snadnější čitelnost dat pouhým pohledem.

```

<classinfo>
  <students>
    <student>
      <name>Michael Smith</name>
      <average>99.5</average>
      <age>17</age>
      <graduating>true</graduating>
    </student>
    <student>
      <name>Steve Johnson</name>
      <average>34.87</average>
      <age>17</age>
      <graduating>false</graduating>
    </student>
    <student>
      <name>Rebecca Young</name>
      <average>89.6</average>
      <age>18</age>
      <graduating>true</graduating>
    </student>
  </students>
</classinfo>

```

Výpis 2.3: Uložení dat ve formátu XML (převzato z [17]).

## 2.10 Schéma databáze

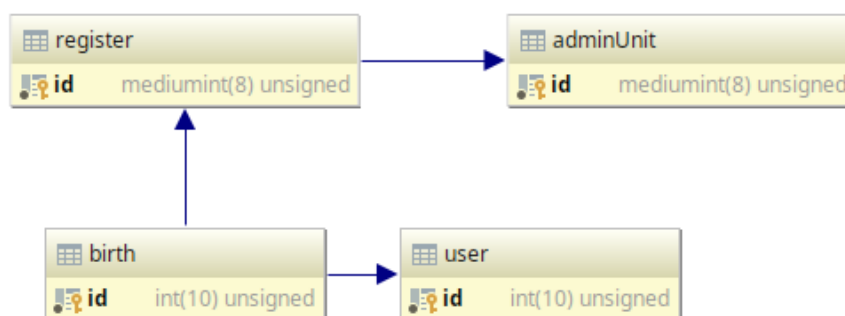
Součástí webové aplikace DEMoS je databáze jejíž autorem je Ing. Radek Kočí, Ph.D. Pochopení struktury této databáze bylo nezbytné pro další práci. Databáze obsahuje velké množství tabulek, a proto zde bude popsána po částech. Kompletní schéma databáze je obsaženo v příloze A.

### Matriky a záznamy

Schéma tabulek, do něhož jsou ukládány záznamy o matrikách a záznamech v nich obsažených je na obrázku 2.10. Tabulka **register** je určena pro uchovávání informací o jednotlivých matrikách. Záznamy o matrikách obsahují údaje o archívu (atribut **archive**), fondu (**fond**), signatuře matriky (**signature**), jazycích matriky (**lang1**, **lang2**, **lang3**) a počtu skenů (**scan\_count**). Každá matrika spadá do nějaké administrační jednotky, neboli původce. Údaje o administračních jednotkách jsou uchovávány v tabulce **adminUnit**.

Záznamy z jednotlivých matrik jsou ukládány do tabulky **birth**, kde je pomocí atributu **register\_id** odkazována matrika, z níž záznam pochází. V tabulce jsou dále atributy udávající číslo skenu (**scan**) a číslo pozice (**pos**) v tomto skenu. Atribut **lay** určuje rozložení záznamu na skenu, které může být *vlevo*, *vpravo* nebo *přes celé*. Mezi další atributy patří například **baptism\_date**, který uchovává datum křtu narozeného dítěte, nebo atribut **sex**, udávající pohlaví dítěte. Tabulka obsahuje několik dalších atributů, které se týkají úmrtí, bířmování, sňatku rodičů a dalších už spíše dodatečných údajů.

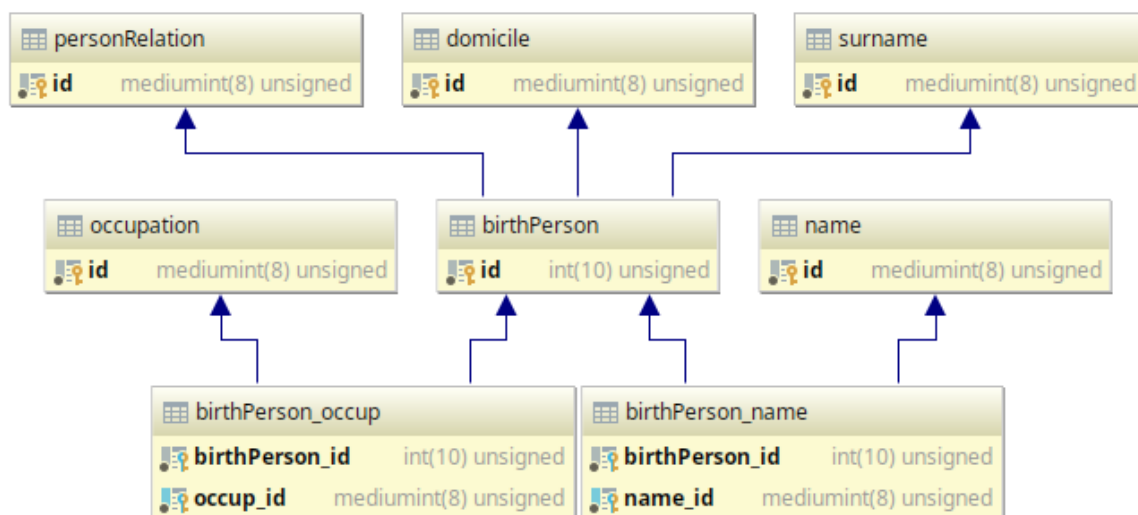
U každého záznamu je také uveden autor jeho přepisu. Ukládán je v atributu **owner**, který odkazuje na uživatele uloženého v tabulce **user**.



Obrázek 2.10: Schéma tabulek pro ukládání informací o matrikách a matričních záznamech.

### Osoby v záznamu

Ke každému matričnímu záznamu je vztaženo několik osob. Údaje o těchto osobách jsou ukládány do tabulky **birthPerson** jejíž schéma včetně okolních tabulek je znázorněno na obrázku 2.11. Záznam, ke kterému se daná osoba váže je identifikován atributem **birth\_id**.



Obrázek 2.11: Schéma tabulek pro ukládání informací o osobách vztažených k matričním záznamům.

Atribut **rel** udává roli, kterou daná osoba v matričním záznamu zastává. Narozené dítě je označováno hodnotou **main**. Hodnoty **m**, **f**, **f\_f**, **f\_m**, **f\_m\_f**, **f\_m\_m**, **m\_f**, **m\_m**, **m\_m\_f**, **m\_m\_m** identifikují předky dítěte v daném záznamu. Křtitel je určován hodnotou **granted** a porodní bába hodnotou **midwife**. Záznamy mohou obsahovat také údaje o kmotrech, hodnoty **gf\_1**, **gf\_2**, **gf\_3**, **gf\_4**, a také údaje o příbuzných těchto kmotrů. Příbuzní kmotrů jsou označováni hodnotami **gfrel\_1**, **gfrel\_2**, **gfrel\_3**, **gfrel\_4**.

Tabulka **birthPerson** dále obsahuje atribut **birth\_date**, který uchovává datum narození. Atributy **street** a **descr\_num** uchovávají informace o názvu ulice a číslu popisném. Názvy obcí jsou k dispozici v tabulce **domicile** a je na ně odkazováno ze stejnojmenného atributu v tabulce **birthPerson**.

Stejným způsobem jsou uchovávána i příjmení, která jsou uložena v tabulce **surname** a jsou odkazována z atributu **sname**. Křestní jména jsou uchovávána v tabulce **name**, která je s tabulkou **birthPerson** ve vztahu M:N. Tyto tabulky jsou propojeny vazební tabulkou **birthPerson\_name**.

Povolání jsou uchovávána v tabulce **occupation** a jsou, stejně jako křestní jména, ve vztahu M:N s tabulkou **birthPerson**. Tyto dvě tabulky jsou svázány vazební tabulkou **birthPerson\_occup**.

Dalšími atributy v tabulce **birthPerson** jsou **title** udávající titul osoby, **religion** uchovávající vyznání a atribut **dead**, který určuje zda byla daná osoba v době vzniku matričního záznamu po smrti.

## Kapitola 3

# Analýza požadavků a návrh řešení

Tato kapitola je zaměřena na analýzu požadavků a na návrh řešení problému. Na začátku kapitoly je definován cíl práce. Druhá část kapitoly je věnovaná návrhu automatického dokončování slov, včetně zobrazování seznamu návrhů, pro jednotlivá vstupní pole. Dále je popsán návrh automatického doplňování záznamu. Poslední část kapitoly obsahuje popis výběru podle jazyka matriky.

### 3.1 Cíl práce

Cílem práce je implementace inteligentního doplňování slov v GUI webové aplikace DEMoS, již popsané v sekci 2.4. Inteligentní doplňování slov má sloužit k usnadnění a urychlení přepisů matričních záznamů. Součástí implementace má být systém pro automatické dokončování slov se seznamem návrhů při vyplňování jednotlivých položek matričního záznamu. Dále je úkolem vytvořit systém, který bude schopný na základě co nejmenšího počtu zadaných údajů doplnit zbytek matričního záznamu, nebo alespoň co největší část tohoto záznamu. Systém má být také schopný rozpoznat jazyk matriky a podle něj pak vybírat jednotlivá slova, či části záznamu k doplnění.

Součástí systému by mělo být také nastavení, které bude umožňovat aktivaci/deaktivaci jednotlivých částí. Navíc bude obsahovat volbu, zda se mají jednotlivé položky záznamu automaticky doplňovat, nebo mají být zobrazeny pouze návrhy vhodné k doplnění do daných částí záznamu. Tyto možnosti bude možné libovolně kombinovat.

Zmíněné části aplikace mají být implementovány pro formulář určený k přepisu matrik narozených, které obsahují informace o dítěti, křtiteli, porodní bábě, kmotrech, rodičích a případně dalších předcích.

Hlavním úkolem systému bude co nejvíce zkrátit čas potřebný k přepsání jednoho matričního záznamu.

### 3.2 Výběr podle jazyka matriky

V sekci 2.2 bylo zmíněno, že jazyk matrik se v průběhu času měnil. Jedna matrika tak může být psána více jazyky. Všechny matriky mají uvedeno v jakých jazycích jsou vedeny.

Původním návrhem bylo zjišťování jazyka podle již vyplněných údajů záznamu. Poté měly být navrhována slova pouze v tomto jazyce. Z databáze ovšem nelze zjistit, kdy se jazyk matriky mění. Přejít z jazyka nemusel být striktní a slova původního jazyka mohla být ještě nějakou dobu používána i po změně.

Slova tedy nebudou vyhledávána podle jazyka matrice. Vyhledávání návrhů bude prováděno podle původce, a podle data narození, či podle data křtu, pokud bude alespoň jedno z těchto dat zadáno.

### 3.3 Vstupní pole s automatickým dokončováním

Systém vstupního pole s automatickým dokončováním slov bude zobrazovat návrhy vhodných slov pro jednotlivé textové položky formuláře pro přepis matričních záznamů narozených.

Při prázdném záznamu budou slova navrhovaná k doplnění filtrována pouze podle původce a řazena podle počtu výskytů v databázi. Filtrování podle původce znamená vyhledávání záznamů pouze v oblasti jejich vzniku, a tím i eliminaci velkého množství ostatních záznamů.

V průběhu vyplňování záznamu bude vyhledávání slov více specifikováno. Při zadaném datu narození dítěte budou výsledky filtrovány podle tohoto data (viz sekce 3.2). Starší matrice, tedy knihy přibližně z 18. století a starší, byly vedeny jako matrice křtů, a datum narození tedy neobsahují. V těchto případech bude pro filtrování místo data narození použito datum křtu.

Po zadání obce narození bude vyhledávání probíhat pouze pro danou obec. Názvy obcí ovšem mohou být stejné na více místech. V České republice patří v současné době mezi nejznámější případ obce s názvem *Lhota*. Vyhledávání záznamů týkajících se pouze obce v dané oblasti bude zajišťovat již zmíněný filtr podle původce, jelikož názvy obcí spadajících pod určitého původce by měly být jedinečné.

### 3.4 Inteligentní vyplňování záznamu

Systém pro inteligentní doplňování záznamu bude podle co nejmenšího počtu zadaných údajů prohledávat databázi a doplňovat zbývající části záznamu. Doplňovány však nebudou jednotlivé položky záznamu, ale vždy všechny dostupné údaje o osobách, které budou nalezeny jako vhodní kandidáti pro vložení do daného záznamu. Nalezené a vyfiltrované výsledky jsou seřazeny podle počtu výskytů v databázi. S rostoucím počtem výskytů dané osoby v databázi roste také pravděpodobnost správnosti návrhu.

Vyhledávání *křtitele* bude zahájeno ve chvíli, kdy bude zadána alespoň jedna z těchto položek: datum křtu, obec narození, příjmení křtitele. Kromě těchto položek může být výběr filtrován i podle křestního jména křtitele, pokud je zadáno. Při postupném přidávání těchto hodnot budou tyto použity při filtrování a údaje o křtiteli tak budou přesnější. Podle původního návrhu mělo hledání vhodného návrhu na křtitele probíhat pouze při zadaném datu křtu, které by určovalo přítomnost údajů o křtiteli v daném záznamu. Při průzkumu matričních záznamů ovšem byly nalezeny případy, kdy datum křtu uvedeno nebylo a údaje o křtiteli v záznamu obsaženy byly. Vyskytovaly se i záznamy, kdy záznam datum křtu obsahoval, ale informace o křtiteli zapsány nebyly.

Způsob vyhledávání informací o *porodní bábě* bude stejný jako u *křtitele*. Jediný rozdíl bude v tom, že filtrování nebude probíhat podle data křtu, ale podle data narození dítěte. Vyhledávání křtitele a porodní báby by mělo mít poměrně velkou přesnost vzhledem k faktu, že v určitém období měla každá obec pouze několik osob (často jednu nebo dvě), které zastávaly funkci porodní báby nebo křtitele.

Dále je potřeba doplnit *předky* dítěte, kterými jsou rodiče, prarodiče a případně praprarodiče, a také *kmotry*. Průzkumem matričních záznamů bylo zjištěno, že ze všech předků bývá vyplněno nejvíce informací o otci, u ostatních předků to často bývá například pouze křestní jméno. Pokud tedy budou zadány jméno, příjmení a povolání nebo obec otce, vyhledají se všechny záznamy odpovídající těmto údajům. Z těchto záznamů následně bude možné doplnit zbývající údaje o otci a data o dalších předcích a kmotrech.

Systém bude informace k doplnění do záznamu hledat také podle adresy narození. V databázi již může existovat záznam o narození se shodnou adresou jako je adresa aktuálně vyplňovaného záznamu. V takovém případě je zde šance, že nalezen záznam, či jeden z nalezených záznamů se týká narození sourozence aktuálně přepisovaného záznamu, a bude tedy obsahovat většinu požadovaných údajů pro aktuálně přepisovaný záznam.

## Kapitola 4

# Implementace a testování

Obsahem této kapitoly je popis implementace systému podle výše zmíněného návrhu, a také popis testování tohoto systému. První část kapitoly je věnovaná technologiím použitým při implementaci navrženého systému. Dále je popsána struktura požadavků, které jsou zasílány na server pro získání požadovaných dat. Poté je v kapitole popis implementace zobrazování návrhů slov pro jednotlivé položky záznamu a popis inteligentního vyplňování záznamu. Poslední část věnována implementaci obsahuje popis zobrazování návrhů na vyplnění celého záznamu.

### 4.1 Použité nástroje

Tato práce je implementována jako součást webové aplikace *DEMoS*. Serverová část této aplikace je implementována s využitím PHP frameworku *Nette* (viz sekce 2.7). Z tohoto důvodu je framework *Nette* využit i k implementaci systémů pro automatické vyplňování matričních záznamů.

K uložení dat používá aplikace *DEMoS* MySQL databázi jejíž struktura je popsána v sekci 2.10.

Při implementaci klientské části práce byl použit zejména jazyk JavaScript a knihovna jQuery, která je na tomto jazyce postavena. K tvorbě modálních oken pro nastavení a zobrazování návrhů byl použit značkovací jazyk HTML a knihovna *Bootstrap*<sup>1</sup>. K zobrazování ikon byla použita knihovna *Font Awesome*<sup>2</sup>.

Pro vývoj na lokálním počítači bylo potřeba mít nainstalovaný interpret jazyka PHP, a také server s aktivním PHP modulem. Konkrétně byl použit server *Apache2*<sup>3</sup>. Dále byl potřeba *MySQL server*<sup>4</sup> pro možnost vytvoření databáze na lokálním PC.

Práce byla implementována s využitím vývojového prostředí *PhpStorm*<sup>5</sup>, které je dle mého názoru nejlepším vývojovým prostředím pro webové aplikace. Toto IDE<sup>6</sup> má kromě jazyka PHP podporu spousty dalších programovacích jazyků, jako jsou například jazyky HTML, CSS, JavaScript nebo SQL, které byly během vývoje využity. *PhpStorm* umožňuje také přístup a práci s databází bez nutnosti využití dalšího programu.

---

<sup>1</sup><https://getbootstrap.com>

<sup>2</sup><https://fontawesome.com>

<sup>3</sup><https://httpd.apache.org>

<sup>4</sup><https://dev.mysql.com>

<sup>5</sup><https://www.jetbrains.com/phpstorm>

<sup>6</sup>IDE – Integrated Development Environment (vývojové prostředí)



Během vývoje byl využíván také systém pro správu verzí *Git*<sup>7</sup>. Protože na projektu DEMoS pracovalo více lidí, byl vytvořen také webový Git repozitář pomocí nástroje *GitLab*<sup>8</sup>. Tento repozitář byl zpřístupněn na serveru [perun.fit.vutbr.cz](http://perun.fit.vutbr.cz).

## 4.2 Komunikace mezi klientem a serverem

Každá webová aplikace je založena na komunikaci mezi klientem a serverem. K tomu je využíváno HTTP požadavků a HTTP odpovědí. V této práci je potřeba zasílat serveru hodnoty již vyplněných položek záznamu. Server tato data zpracuje a nalezená data zašle zpět klientovi. Těla požadavků a odpovědí musejí mít pevně danou strukturu, aby jim klient se serverem rozuměli a uměli je zpracovat.

V práci byla ke komunikaci mezi serverem a klientem použita technologie AJAX, již popsaná v sekci 2.1, která umožňuje asynchronní komunikaci mezi klientskou a serverovou částí aplikace bez nutnosti obnovování stránky. Asynchronní komunikace je vytvářena pomocí funkce `$.ajax()`, která je součástí knihovny jQuery. Vzhledem k tomu, že bude klient zasílat serveru data, byla zvolena metoda POST (atribut *method* v nastavení metody), která je k tomu určená a umožňuje zasílat neomezené množství dat.

K přenosu dat byl zvolen formát JSON (viz sekce 2.9), jehož struktura vychází z jazyka JavaScript a na jeho zpracování tak nejsou potřeba žádné další nástroje či knihovny. O vytváření těla požadavků (atribut *data* v nastavení metody `$.ajax()`) v tomto formátu se stará metoda `createJSON()`. Do každého požadavku je nejprve vložen atribut *focused-input-name*, jehož hodnota je název aktuálně vyplňované položky záznamu. Díky tomuto atributu server rozpozná, ve kterých částech databáze má vyhledávat návrhy slov. Dalšími atributy, které jsou vždy obsaženy v těle HTTP požadavku, jsou *archive*, *fond* a *signature*, podle nichž lze jednoznačně identifikovat matriku a zjistit informace s ní související. Nakonec jsou do požadavku vloženy hodnoty ručně vyplněných vstupních položek, tedy těch, které mají v HTML přidělenou třídu *man-populated*. Jednotlivé hodnoty jsou uloženy v attributech pojmenovaných podle názvů těchto položek ve formuláři. Ukázka výsledné struktury dat se nachází na obrázku 4.1.

```
{
  "focused-input-name": "__granted_names",
  "archive": "MZA",
  "fond": "E67",
  "signature": "1",
  "__child_baptism_date": "1860-06-04",
  "__child_birth_date": "1860-06-03",
  "__child_descr_num": "10",
  "__child_village": "Adamsthal",
  "__granted_names": "A"
}
```

Výpis 4.1: Ukázka těla požadavku zaslaného při vyplňování jména křtitele.

Po přijetí odpovědi ze serveru je odpověď zpracována (atribut *success* v nastavení metody `$.ajax()`) v závislosti na typu aktuálního požadavku, kdy se může jednat o automatické doplňování záznamů či získání návrhů pro části záznamu, získání návrhů na celý

<sup>7</sup><https://git-scm.com>

<sup>8</sup><https://about.gitlab.com>

záznam nebo získání návrhů slov pro aktuálně vyplňovanou položku matričního záznamu. Jednotlivé způsoby zpracování odpovědí jsou popsány v sekcích 4.3-4.6.

```
$.ajax({
  method: "POST",
  url: path + "autocomplete/autocomplete",
  data: createJSON(),
  success: function (data) {
    response(data);
  },
});
```

Výpis 4.2: Ukázka nastavení metody `$.ajax()` pro zasílání požadavků na server.

### 4.3 Vstupní pole s automatickým dokončováním

K implementaci vstupního pole s automatickým doplněním byl zvolen *Autocomplete widget* (viz sekce 2.8), který je součástí knihovny jQueryUI, jenž se na straně klienta stará o komunikaci se serverem a o zobrazování návrhů slov. O zpracování požadavků klienta se na straně serveru stará třída `AutocompletePresenter`.

#### Klientská část

Již zmíněný Autocomplete widget je aktivován po načtení stránky s formulářem pro přidání matričního záznamu. Tato aktivace je provedena za použití metody `autocomplete()`, která je aplikována na všechna textová vstupní pole, tedy HTML elementy `input` s atributem `type=text`. Již zmíněná metoda `autocomplete()` přijímá jediný parametr a tím je asociativní pole s nastavením chování. Nastavení použita v této práci:

- Atribut `delay` určuje dobu prodlevy v milisekundách mezi posledním vložením znaku a zasláním požadavku pro získání návrhů. Pokud tento atribut není uveden, je výchozí hodnota prodlevy nastavena na 300 ms. V této práci je explicitně nastavena hodnota 100 ms, aby se návrhy zobrazovaly i při rychlejším vyplňování. Hodnota by při používání vzdálených zdrojů dat neměla být nižší, aby nedocházelo k příliš velkému množství zasílaných požadavků.
- Atribut `autoFocus` určuje, zda má být při zobrazení seznamu návrhů automaticky označena jeho první položka. `autoFocus` může nabývat hodnot `true`, nebo `false` (výchozí hodnota). V této práci je tento atribut nastaven na hodnotu `true`.
- Atribut `select` obsahuje funkci, která definuje činnost při volbě jednoho návrhu ze seznamu. Výchozí akcí je pouze vyplnění vstupního pole zvolenou hodnotou. V této implementaci je navíc danému vstupnímu poli přidána třída `man-populated` a zavolána metoda `autopopulate()` popsaná v sekci 4.4.
- Jako poslední byl použit atribut `source` definující zdroj požadovaných dat. Má přiřazenou funkci, která umožňuje vlastní nastavení volání AJAXu včetně těla požadavku a reakce na přijatou odpověď. Url je zde nastavena na `"autocomplete/autocomplete"` a tělo požadavku (atribut `data`) je vytvořeno pomocí již zmíněné funkce `createJSON()`.

Po tomto nastavení již zasílání dat a zobrazování seznamu návrhů funguje zcela automaticky. Poté co klient přijme odpověď ze serveru, jsou přijatá data vložena do seznamu návrhů a zobrazena uživateli tak jako na obrázku 4.1.



Obrázek 4.1: Zobrazení seznamu návrhů pro doplnění položky matričního záznamu.

## Serverová část

Zpracování požadavku na získání návrhů pro jednu položku matričního záznamu provádí třída `AutocompletePresenter`. Zpracování začíná metodou `actionAutocomplete()`. Nejprve je s využitím názvu archivu a signatury matrice zjištěn identifikátor původce, pod kterého daná matrice spadá. K tomu je využívána metoda `searchByArchiveSignature()` ze třídy `RegisterManager`. Získaný identifikátor původce je vložen k datům požadavku. Poté je z atributu `focused-input-name` získán název vstupního pole, pro které jsou požadovány návrhy. Podle něj je pak zavolána příslušná metoda. Tato metoda v daném pořadí a s různými parametry invokes metody ze třídy `BirthManager`, které prohledávají databázi, dokud není nalezeno pět vhodných návrhů. Nakonec jsou nalezené návrhy zaslány zpět klientovi.

Metody třídy `BirthManager`, které jsou určeny pro vyhledávání návrhů jednotlivých položek, mají vždy mezi parametry data obsažená v požadavku, podle nichž bude možné vyhledávat v databázi. Dalším parametrem nazývaným `$exclude` je pole již nalezených návrhů, které budou z dané části vyhledávání vyloučeny, aby nebyly nacházeny duplicitní návrhy. Některé metody mají také parametr `$birthPerson`, který nese název osoby (father, mother, granted, midwife, atd.), pro níž jsou hledány návrhy. Poslední z možných parametrů je `$relation`, který udává název vztahu, tak jak je uvedena v databázi (m, f, m\_m, m\_f, atd.). Hodnota tohoto parametru může být zadána i obecněji, a to pro vyhledávání mezi muži (hodnota *men*) nebo mezi ženami (hodnota *women*).

Každá z těchto metod na začátku provede filtrování podle původce za účelem redukce záznamů pouze na záznamy týkající se dané oblasti. Dále je vždy prováděno filtrování podle již zadaného začátku slova aktuálně vyplňované položky záznamu.

K vyhledání *křestních jmen* slouží metody `suggestNames()` a `suggestNameNames()`. První zmíněná metoda vybere křestní jména ze všech záznamů týkající se dané oblasti. Dále pak vybere pouze osoby, jejichž vztah v záznamech odpovídá vztahu uvedenému v parametru `$relation`. Následně je výběr zúžen na jména začínající písmeny již vepsanými v dané položce záznamu. Poté, pokud je v požadavku uvedena, jsou návrhy filtrovány podle obce narození dané osoby, tedy u dítěte podle obce dítěte, u otce podle obce otce atd. V poslední fázi je výběr ještě zúžen pouze na období kolem data narození, případně podle data křtu. Nalezené návrhy jsou seřazeny podle počtu výskytů v záznamech a přidány do seznamu návrhů. Druhá zmíněná metoda `suggestNameNames()` vyhledává jména v tabulce `name` bez závislosti na záznamech. Vyhledávání probíhá pouze podle počátečních písmen jména a případně podle zadaného pohlaví.

Návrhy na *příjmení* jsou vyhledávány metodou `suggestSurnames()`, která pracuje na téměř stejném principu jako již popsaná metoda `suggestNames()`. Návrhy příjmení jsou však navíc ještě filtrovány podle již zadaného křestního jména dané osoby záznamu.

Metoda `suggestBirthPersonOccupation()` má na starosti vyhledávání návrhů *povolání* a *titulu křtitele*. Metoda nejprve vyhledá matriční záznamy pro danou oblast, které případně profiltruje pole zadaného vztahu. Z těchto záznamů jsou následně vybrány ty, u kterých je obec dané osoby shodná s obcí vyplněnou u příslušné osoby ve formuláři. Výsledná množina příjmení je stejně jako u předchozích metod seřazena podle počtu výsledků. K doplnění návrhů povolání je určena metoda `suggestOccupation()`, kde jsou povolání hledána pouze na základě zadaných počátečních písmen.

Úkolem metod `suggestVillages()` a `suggestDomicileVillages()` je získávání návrhů názvu obcí. První z těchto metod vyhledává obce v již vytvořených záznamech z nichž vybere pouze záznamy týkající se dané oblasti. V kombinaci s filtrováním podle zadaných počátečních písmen se množství možných obcí značně sníží. Nalezené záznamy mohou být dále filtrovány podle vztahu osoby k dítěti. Vzhledem k faktu, že názvy obcí byly v průběhu času upravovány, je potřeba vyhledávání zaměřit pouze na určité období. Z nalezených výsledků jsou tedy ponechány pouze ty, které se týkají daného období. Toho je dosaženo s využitím data narození či data křtu. Nalezené obce jsou nakonec stejně jako v předchozích metodách seřazeny podle počtu výskytů v záznamech. Druhá zmíněná funkce vyhledává obce v tabulce `domicile` bez ohledu na tabulku se záznamy a pouze podle počátečních písmen.

Poslední metodou pro vstupní pole s automatickým dokončováním je `suggestStreets()`, jejímž cílem je vyhledání návrhů názvů ulic. Vyhledávání probíhá stejným způsobem jako při hledání obcí v metodě `suggestVillages()`. Pokud je známa obec, jsou ulice vyhledávány pouze v rámci této obce.

## 4.4 Inteligentní vyplňování záznamu

V průběhu vyplňování formuláře na vytvoření matričního záznamu jsou průběžně (při každém vyplnění položky záznamu) na server zasílány požadavky. Na serveru je vždy požadavek zpracován a v případě vhodné kombinace zadaných údajů je zahájeno hledání vhodných dat k doplnění do formuláře matričního záznamu. Systém inteligentního vyplňování záznamu pracuje stejně jako systém vstupního pole s automatickým dokončováním s identifikátorem původce za účelem specifikace oblasti vyhledávání. Hledání údajů v databázi zajišťují opět metody ze třídy `BirthManager`. Vyhledávání osob je vždy zahájeno nalezením záznamů, které by mohly obsahovat údaje o osobách vhodných k doplnění do nového záznamu. Tyto záznamy jsou vyhledány podle oblasti, ze které pochází matrika, do níž je přidáván nový

záznam. Filtrovány jsou také podle data narození či data křtu, pokud je alespoň jedno z těchto dat zadáno.

Pokud je v záznamu zadán alespoň jeden z údajů datum křtu, obec narození nebo příjmení křtitele, jsou podle těchto údajů vyhledáni křtitelé. Vyhledávání křtitelů zajišťuje metoda `suggestGranted()`. V záznamech z dané oblasti jsou vyhledáni křtitelé, kteří odpovídají zadaným údajům. Kromě data křtu a oblasti může být vyhledávání křtitele blíže specifikováno jeho jménem a příjmením, pokud jsou zadána. Všichni nalezení křtitelé jsou vloženi do odpovědi. Do formuláře jsou vyplněny údaje toho křtitele, který byl v prohledávaných záznamech nalezen nejčastěji.

*Porodní báby* jsou vyhledávány metodou `suggestMidwife()`. Algoritmus je stejný jako u vyhledávání křtitelů. Jediný rozdíl je použití data narození místo data křtu pro vymezení období prohledávání.

Doplňování křtitelů a porodních bab by mělo mít vysokou přesnost, protože v obcích ve velkém množství případů působilo v jednu dobu pouze několik osob zastávajících pozici porodní báby nebo křtitele. Velmi často se jednalo pouze o jednu či dvě osoby, které zastávaly jednu z těchto pozic.

Starší matriky neobsahují mnoho údajů o předcích. Detailnější informace jsou často uvedeny pouze u otce. U ostatních předků je často zapsáno pouze jméno, případně je uvedeno ještě příjmení. Z tohoto důvodu je z předků jako první vyhledáván otec a až podle záznamu, ve kterém je otec nalezen, jsou vyhledáni ostatní předkové. Aby mohl být otec vyhledán s vysokou přesností, je potřeba znát minimálně jeho jméno a příjmení. O vyhledání se poté stará metoda `suggestFather()`. Prohledávané záznamy jsou opět omezeny pouze na oblast, do které spadá daná matrika, dále na datum narození, případně datum křtu, a pokud je zadána obec narození dítěte, je vyhledávání omezeno pouze na tuto obec. K vyhledání otce v matričních záznamech mohou být využity všechny údaje o otci, které je možné do záznamu zapsat. Čím více těchto údajů bude zapsáno, tím přesnější bude vyhledávání.

Po nalezení otce jsou údaje o něm vloženy do odpovědi a je získáno ID narození, ke kterému jsou tyto údaje vázány. Na základě tohoto identifikátoru záznamu narození jsou vyhledáni také ostatní předkové a kmotři dítěte. Jejich údaje jsou opět vloženy do odpovědi, a ta je zaslána klientovi, který všechny přijaté údaje doplní do příslušných polí formuláře pro přepis matričního záznamu.

Při vyhledávání může nastat situace, kdy bude nalezeno více možných variant k doplnění. V takovém případě nelze určit, která varianta je správná, a proto je varianta vyplněná do formuláře vybrána náhodně.

Zcela automatické vyplňování formuláře však může vlivem nepřesností s kombinací nepozornosti a nedůsledné kontroly uživatele zanášet do matričních záznamů chyby, které by mohly negativně ovlivňovat také vytváření rodokmenů, které je plánovanou součástí aplikace DEMoS. Tyto problémy by mělo řešit pouze zobrazování návrhů možných k doplnění, které je popsáno v sekci 4.5, namísto automatického vyplňování formuláře bez kontroly uživatele.

```
{
  "midwife": [
    {
      "__midwife__names": "Mariana",
      "__midwife__surname": "Blahuš",
      "__midwife__domicile": "z Ostré Lhoty",
      "__midwife__street": "",
      "__midwife__descr_num": "206"
    }
  ]
}
```

```

    },
    {
      "__midwife__names": "Mariana",
      "__midwife__surname": "Maluš",
      "__midwife__domicile": "z Ostré Lhoty",
      "__midwife__street": "",
      "__midwife__descr_num": "155"
    }
  ]
}

```

Výpis 4.3: Ukázka odpovědi s návrhy osob.

## 4.5 Zobrazování návrhů osob

Zobrazování návrhu osob poskytuje oproti automatickému doplňování uživateli větší kontrolu nad údaji vyplňovanými do formuláře s matričním záznamem. Místo automatického doplnění dat jsou pro jednotlivé osoby záznamu zobrazeny pouze nalezené návrhy s údaji o těchto osobách, které je možno doplnit do záznamu. Uživateli je tímto také udělena možnost volby, zda a kterou z navržených možností nechá do záznamu zapsat.

Algoritmus vyhledávání vhodných údajů na straně serveru je shodný s algoritmem popsaným v sekci 4.4. Rozdílné je ovšem zpracování odpovědi na straně klienta.

U každé osoby, pro kterou je v odpovědi uveden alespoň jeden návrh k doplnění, je zobrazeno tlačítko s ikonou osoby (viz výpis 4.2) a přijatá odpověď je uložena. Po kliknutí na toto tlačítko u některé z osob, je zobrazeno předem připravené modální okno, do jehož záhlaví je vyplněn název dané osoby (křtíel, porodní bába, otec atd.) a v těle modálního okna je zobrazen seznam návrhů pro danou osobu. Pokud nějaký ze zobrazených návrhů odpovídá požadavku, je možné na něj kliknout a tím zvolené údaje vyplnit do formuláře. Modální okno je poté automaticky uzavřeno.

Obrázek 4.2: Tlačítko umožňující zobrazení návrhů pro porodní bábu.

Vytváření jednotlivých položek seznamu návrhů je implementováno pomocí metody `createListItem()`, která podle předaných parametrů vytvoří položku seznamu ve formátu `jméno[mezera]příjmení, povolání, obec`. Tyto položky jsou vytvořeny pro všechny návrhy na danou osobu a poté zapsány do těla modálního okna.

## Porodní bába

- Mariana Blahuš, z Ostré Lhoty 206
- Mariana Maluš, z Ostré Lhoty 155

Zavřít

Obrázek 4.3: Modální okno s návrhy porodních bab.

## 4.6 Zobrazení návrhů na celý záznam

Návrhy na celý záznam jsou vyhledávány na základě zadané adresy narození dítěte. Z nalezených záznamů jsou všechny údaje, s výjimkou informací o dítěti, zaslány klientovi, který jej zobrazí formou tabulky v modálním okně tak jako na obrázku 4.4. Po kliknutí na zvolený řádek se záznamem dojde k automatickému vyplnění tohoto záznamu do formuláře.

Zasílání požadavků na server je implementováno v metodě `getRecordSuggestions()`, která využívá již zmiňovanou funkci `$.ajax()`. Zobrazování nalezených záznamů je implementováno metodou `showRecordSuggestionsModal()`, jenž do modálního okna vloží data o nalezených záznamech a toto modální okno následně zobrazí.

### Nalezené záznamy

Kliknutím na řádek se záznamem jej vyplníte do formuláře.

Křestitel	Porodní bába	Otec Otcův otec Otcova matka Otcův děda (po matce) Otcova bába (po matce)	Matka Matčín otec Matčina matka Matčín děda (po matce) Matčina bába (po matce)	Kmotři
Anton Dostal lokální kurát	Josefa Blažek von Babitz	- Anton Hřebíček, Schmelzmeister; in Josefsthäl - Josef Hřebíček, Müllermeisters; in Blansko - Franciska Kučera - -	- Filipine - Martin Touchin, Förmers; in Josefsthäl - Maria Opletal - -	- Josef Hřebíček, Müllermeister; in Klein Kinitz - Anna - -
Anton Dostal lokální kurát	Josefa Blažek von Babitz	- Mathias Kostínek, Bahnwächter; in Adamsthäl - Thomas Kostínek, Schneider; in Drasow - Catharina Doliška - -	- Theresia - Josef Navrátil, Schneiders; in Klein Věstin - Rosalia Pokorný - -	- Georg Klepárník, Häusler; in Adamsthäl - Franciska - -

Zavřít

Obrázek 4.4: Ukázka zobrazování návrhů na celé záznamy.



## 4.7 Testování

Hlavním cílem implementovaného systému je redukce času potřebného pro přepis jednoho záznamu. Výsledek bylo potřeba otestovat za účelem ověření funkčnosti a splnění deklarovaného cíle.

Aplikace DEMoS bude veřejně přístupná a bude určena primárně pro amatérské genealogy, kteří se studiu rodové historie věnují ve svém volném čase a kteří nemají velké zkušenosti se čtením matričních záznamů. Z tohoto důvodu byl systém testován na lidech bez zkušeností se zkoumáním matrik.

Podle testovacího protokolu z přílohy C bylo úkolem respondentů přepsat záznam z digitalizované matriky s různými kombinacemi nastavení. Uživatelé měli nejprve za úkol si přečíst návod k použití testovaného systému a seznámit se s uživatelským rozhraním. Návod k použití, který měli respondenti k dispozici po celou dobu testování, je uveden v příloze B. Dalším krokem byl přepis záznamu, kdy byly vypnuty všechny systémy pro hledání a zobrazování návrhů. Tento krok byl určen ke zjištění referenčního času potřebného na přepis jednoho záznamu. Při druhém vyplňování měl uživatel k dispozici systém zobrazování návrhů pro jednotlivé položky záznamu. V další fázi pak byl navíc zapnut systém pro zobrazování návrhů k jednotlivým osobám záznamu. Při posledním přepisu bylo aktivní také zobrazování návrhů na doplnění celého záznamu.

Pro testovací účely byl zvolen záznam digitalizované matriky z MZA<sup>9</sup> Brno se signaturou 11765. Testovací záznam je obsažen ve skenu číslo 28. S pořadovým číslem 8. Záznam je zobrazen na obrázku 4.5. Pochází z Ostré Lhoty z roku 1887.

Kniha a strana	Rok, měsíc a den		Křestěnce							Otce				
	narození	křstu	Číslo dom.	J m e n o	Nábož.		Pohlavi		Rod		Mrtvorození	Jmeno, stav a rod	Nábož.	
					katolické	nekatol.	mužské	ženské	manželský	nemanželský			katolické	nekatol.
III 28.	1887 listopad 6. 7.		211.	Martin Děs.	/		/		/			Martin Děs, domkář v Ostré Lhotě, syn Josefa Děs, domkáře v Ostré Lhotě a jeho ženy Anny, dcery Domáše Pavlaše z Ostré Lhoty.	/	
vstoupil zde o Marii Radochovnu 3. července 1912.														

M a t k y			K ř e s t i t e l		K m o t r ů v	
Jmeno, stav a rod	Nábož.		Jmeno a stav jeho Zkouš. bába		Jmeno	Stav a bydliště
	katolické	nekatol.				
Kateřina, dcera Josefa Turčína, domkáře v Ostré Lhotě a jeho manželky Marianny z rodu Martina Zeleneho z Ostré Lhoty.	/		Francišká Murat, domkářka v Ostré Lhotě č. 157.		Kateřina, jeho manželka.	
			Marianna Malá z Ostré Lhoty č. 157.			

Obrázek 4.5: Matriční záznam z Ostré Lhoty z roku 1887 (převzato z [1]).

<sup>9</sup>MZA – Moravský zemský archiv



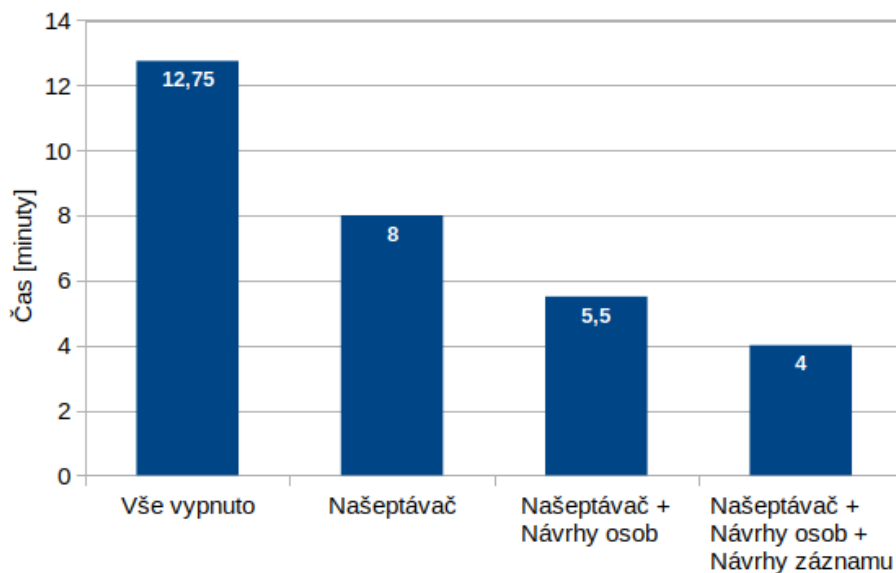
U jednotlivých přepisů byl měřen čas, který byl potřeba pro přepis záznamu. Každému z respondentů byly tedy naměřeny čtyři časy. Kromě doby vyplňování záznamů byla testována také přesnost navrhovaných dat. Dále byla zkoumána také orientace uživatele v systému a schopnost použití vytvořeného rozhraní.

Všichni respondenti měli k dispozici dotazník (viz příloha D), do kterého průběžně vyplňovali časy jednotlivých přepisů. Dále odpovídali na otázky zaměřené na přesnost zobrazování návrhů a na otázky ohledně přívětivosti uživatelského rozhraní pro inteligentní vyplňování záznamu. Uživatelé dostali v dotazníku také možnost vyjádřit své vlastní návrhy na případná rozšíření či úpravy testovaných systémů. V závěru dotazníku měli respondenti testované rozhraní zhodnotit a uvést pozitiva případně negativa tohoto rozhraní.

## Výsledky testování

Hlavním cílem testování bylo zjištění úspory času při použití implementovaných systémů během přepisu matričního záznamu. Z grafu na obrázku 4.6 je možné vidět narůstající úsporu času u jednotlivých přepisů. V případě kdy byly vypnuty všechny systémy byl průměrný čas přepisu téměř třináct minut. Již po aktivaci zobrazování návrhů pro jednotlivé položky klesl průměrný čas téměř o pět minut. Při použití také systému pro zobrazování návrhů jednotlivých osob byla průměrná doba přepisu zkrácena o další dvě a půl minuty. Během posledního přepisu byl zobrazen návrh na celý záznam. Zde byl průměrný čas přepisu čtyři minuty.

Největší úspora času byla zjištěna pro systém zobrazování návrhů k jednotlivým položkám záznamu. Při aktivaci tohoto systému společně se zobrazováním návrhů pro jednotlivé osoby a se zobrazováním návrhu na celý záznam bylo dosaženo zkrácení doby přepisu až o deset minut, tedy přibližně o 70 %.



Obrázek 4.6: Graf vyznačující úsporu času při použití implementovaných systémů během přepisu jednoho záznamu.

Z odpovědí respondentů v dotazníku bylo zjištěno, že všechny tři testované systémy nabízely uživateli přesné návrhy a jejich použití bylo intuitivní. Nejméně přesný byl návrh na celý záznam, kde bylo potřeba opravit údaje o porodní bábě a údaje o kmotrech. Tuto

situaci ovšem nelze ovlivnit, protože již existující záznamy mohou mít oproti aktuálnímu záznamu rozdílné údaje.

Při používání vstupního pole s automatickým dokončováním se respondenti rozdělili na dvě poloviny. Jedna polovina vyplnila formulářové pole návrhem až ve chvíli, kdy se vyskytl na první pozici. Podle odpovědí z dotazníku k tomu ve většině případů dostačovalo zadat pouze 1-2 písmena. Druhá polovina respondentů hledala požadovanou hodnotu v seznamu návrhů již po zadání prvního písmene. Z odpovědí v dotazníku vyplývá, že požadovaný návrh byl po zadání prvního písmene vždy zobrazen.

Jako pozitivum bylo respondenty hodnoceno značné urychlení doby přepisu. Naopak negativní ohlasy byly na nutnou kontrolu a případnou manuální opravu automaticky vyplněných údajů.

Na základě návrhů na úpravy vyjádřených respondenty bylo u vyhledávání návrhů pro jednotlivé položky odstraněno rozlišování velkých a malých písmen v průběhu vyhledávání a zobrazování návrhů.

## 4.8 Návrhy na vylepšení

Pro přesnější vyhledávání návrhů by mohly být použity výsledky práce [11], která je stejně jako tato práce součástí projektu DEMoS. Práce se zabývá normalizací slov obsažených v matričních záznamech. Normalizovány jsou křestní jména, příjmení, povolání, obce a názvy vztahů mezi osobami. Tyto normalizované názvy mohou být využity k nalezení všech variant zápisu dané položky a následné prohledávání záznamů bude prováděno s využitím všech těchto variant.

Během přepisu bude například zadáno křestní jméno Antonia. Za současného stavu by bylo vyhledávání prováděno pouze nad záznamy, které jsou navázány na jméno přesně v tomto tvaru. S využitím normalizací by však byla nalezena normalizovaná podoba tohoto jména a k ní všechny přiřazené způsoby zápisu tohoto jména. Tedy například tvary Antonia, Antonie, Antonije, Antonija, Antona, atd. Vyhledávání poté bude prováděno podle všech těchto tvarů křestních jmen.

Díky tomuto rozšíření by byly prohledávány také záznamy, které jsou odlišné ve způsobu zápisu některých položek a v současné verzi jsou takové záznamy vždy odfiltrovány.

## Kapitola 5

# Závěr

Tato bakalářská práce se zabývá realizací inteligentního doplňování slov v grafickém uživatelském rozhraní pro přepisování matričních záznamů. Hlavním cílem práce bylo co nejvíce zkrátit dobu přepisu matričních záznamů.

Na začátku bylo potřeba nastudovat základní informace o genealogii a matrikách. Dále bylo nutné seznámení se se strukturou matričních záznamů. Následně bylo potřeba nastudovat problematiku tzv. našeptávačů slov neboli vstupních polí s automatickým dokončováním a inteligentního vyplňování formuláře. Poté byly nastudovány některé serverové a klient-ské webové technologie. Před samotnou implementací bylo ještě potřeba pochopit schéma databáze určené pro ukládání matričních záznamů.

Výsledkem implementace je systém pro zobrazování návrhů k jednotlivým položkám matričního záznamu. Dále pak systém automatického vyplňování záznamu, systém pro zobrazování návrhů údajů o jednotlivých osobách a systém pro zobrazování návrhů s údaji k vyplnění celého matričního záznamu. Všechny tyto systémy jsou implementovány jako součást formuláře pro přepis matričních záznamů ve webové aplikaci DEMoS,

Na základě testování implementovaných systémů na uživateli bylo zjištěno, že při použití těchto systémů lze v ideálním případě zkrátit dobu přepisu až o 70 %. Z odpovědí uživatelů v dotazníku vyplynulo, že systémy jsou intuitivní a značně usnadňují a urychlují přepis. Dále byla podle výsledků testování odstraněna rozlišování malých a velkých písmen při vyhledávání návrhů k jednotlivým položkám záznamu.

V budoucnu by tyto systémy mohly být rozšířeny o vyhledávání pomocí normalizovaných variant jednotlivých slov. Byly by tak prohledávány i záznamy, které v prohledávaných položkách obsahují stejné údaje lišící se pouze jejich zápisem. Tímto by ve výsledku mohla být zvýšena přesnost vyhledávání návrhů.

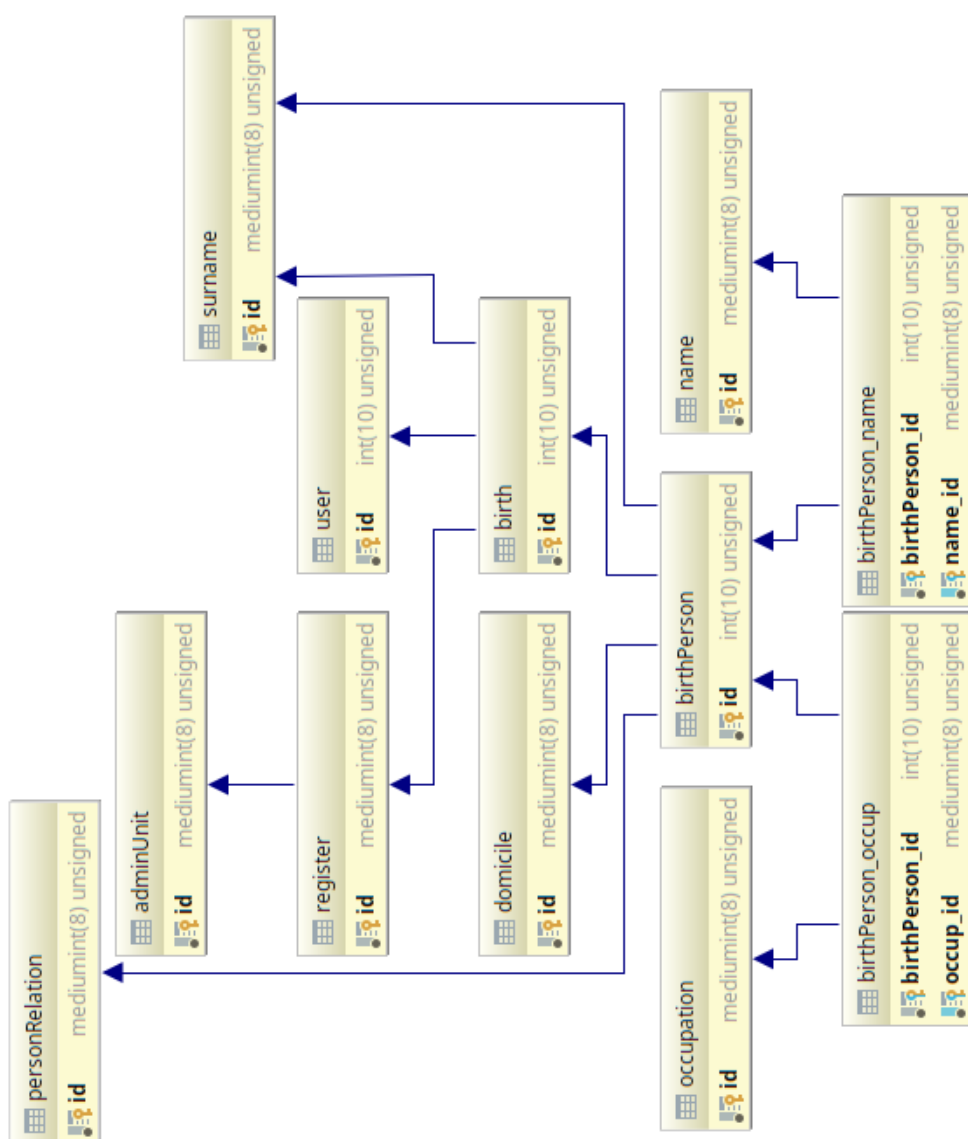
# Literatura

- [1] *Acta Publica*. [Online; navštíveno 26.3.2019].  
URL <http://actapublica.eu>
- [2] *Autocomplete*. [Online; navštíveno 25.10.2018].  
URL <http://jqueryui.com/autocomplete>
- [3] *Digitalizace v archivech*. [Online; navštíveno 27.3.2019].  
URL <http://www.genealogie.cz/aktivity/digitalizace>
- [4] *jQuery.ajax()*. [Online; navštíveno 27.3.2019].  
URL <http://api.jquery.com/jquery.ajax/>
- [5] *Možnosti tvorby komunitní genealogické databáze se sémantickou informací a zahrnutou neurčitostí (DEMOS)*. [Online; navštíveno 31.3.2019].  
URL <https://www.muni.cz/vyzkum/projekty/40566>
- [6] *Nette Framework*. [Online; navštíveno 23.1.2019].  
URL <https://nette.org/cs>
- [7] jQuery Community Experts: *jQuery: kuchařka programátora*. Brno: Computer Press, první vydání, 2010, ISBN 978-80-251-3152-7.
- [8] Darie, C.; aj.: *AJAX a PHP: tvoříme interaktivní webové aplikace profesionálně*. Brno: Zoner Press, první vydání, 2006, ISBN 80-86815-47-1.
- [9] Goodrich, G.: *Understanding the Model-View-Controller (MVC) Architecture in Rails*. [Online; navštíveno 12.11.2018].  
URL <https://www.sitepoint.com/model-view-controller-mvc-architecture-rails/>
- [10] Harding, J.; Skarich, V.; Trueman, T.: *typeahead.js*. [Online; navštíveno 22.10.2018].  
URL <https://github.com/twitter/typeahead.js>
- [11] Hříbek, D.: *Poloautomatická normalizace slov z matričních záznamů*. Brno, 2019. bakalářská práce, Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D. V přípravě.
- [12] Lednická, B.: *Sestavte si rodokmen*. Praha: Grada, 2012, ISBN 978-80-247-4069-0.
- [13] Otwell, T.: *Laravel*. [Online; navštíveno 4.12.2018].  
URL <https://laravel.com/>

- [14] Pecháček, J.: *Genealogické diagramy aneb co je to vlastně rodokmen? Část II.* [Online; navštíveno 26.3.2019].  
URL <https://www.odkudjsme.cz/blog/genealogicke-diagramy-aneb-co-je-to-vlastne-rodokmen>
- [15] Pecháček, J.: *Jak vypadají matriční záznamy.* [Online; navštíveno 4.11.2018].  
URL <https://www.odkudjsme.cz/blog/jak-vypadaji-matricni-zaznamy>
- [16] Resig, J.: *JavaScript a Ajax: moderní programování webových aplikací.* Brno: Computer Press, první vydání, 2007, ISBN 978-80-251-1824-5.
- [17] Zakas, N. C.; McPeak, J.; Fawcett, J.: *Ajax Profesionálně.* Brno: Zoner Press, 2007, ISBN 978-80-86815-77-0.
- [18] Čápka, D.: *MVC architektura.* [Online; navštíveno 12.11.2018].  
URL <https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor>

## Příloha A

### Schéma databáze



Obrázek A.1: Schéma tabulek databáze (Autor: Ing. Radek Kočí, Ph.D.).

## Příloha B

# Návod k použití

Systémy inteligentního doplňování matričního záznamu jsou implementovány ve webové aplikaci DEMoS jako součást formuláře pro přidání matričního záznamu.

V pravé dolní části stránky je k dispozici nastavení uživatelského rozhraní pro přidávání matričních záznamů (viz obrázek B.1).

The image shows a web form titled "Matriky" with a question mark icon. It is divided into three main sections: "Pozice", "Datum a adresa", and "Křestitel".

- Pozice:** Contains fields for "Archiv" (MZA), "Fond" (E67), "Signatura" (1), "Pořadí scanu", "Rozložení na scanu" (Přes celé), and "Pořadí záznamu".
- Datum a adresa:** Contains fields for "Datum narození" (dd.mm.rrrr), "Datum křtu" (dd.mm.rrrr), "Obec", "Ulice", and "Číslo popisné".
- Křestitel:** Contains fields for "Jméno", "Příjmení", and "Titul".

A red arrow points to a circular icon with a gear and a cross, located at the bottom right of the form, indicating the settings for the intelligent completion.

Obrázek B.1: Přístup k nástrojům uživatelského rozhraní pro přepis matričních záznamů.

Výběrem poslední položky bude otevřeno okno s *Nastavením inteligentního doplňování* (viz obrázek B.2).

The image shows a "Nástroje:" (Tools) menu with a close button (X) in the top right corner. It contains four options:

- Režim celé obrazovky
- Přidat pole
- Zobrazit matriku
- Nastavení inteligentního doplňování** (highlighted with a red border)

Obrázek B.2: Nastavení inteligentního doplňování.

V tomto nastavení, které je zobrazeno na obrázku B.3, je možno aktivovat čtyři položky.

### Nastavení inteligentního doplňování

☒ **Nášeptávač**  
☐ **Automatické doplňování**  
☒ **Návrhy jednotlivých osob**  
☒ **Návrh celého záznamu**

Obrázek B.3: Možnosti nastavení inteligentního doplňování.

- *Nášeptávač* – budou zobrazovány seznamy návrhů při vyplňování jednotlivých položek záznamu. Ovládání nášeptávače je popsáno v části B.
- *Automatické doplňování* – nalezené informace budou automaticky vyplňovány do formuláře.
- *Návrhy jednotlivých osob* – všechny aktuálně nalezené návrhy budou v jednotlivých blocích formuláře zpřístupněny pomocí tlačítka . Po kliknutí na jeden z návrhů bude tento návrh vyplněn do příslušného bloku formuláře. Vyhledávání návrhů osob je více popsáno v části B.
- *Návrh celého záznamu* – jsou vyhledávány záznamy se shodnou adresou, které jsou následně zobrazeny jako seznam návrhů s informacemi o jednotlivých osobách (viz obrázek B.4). Po kliknutí na návrh bude daný záznam kromě informací o dítěti vyplněn do formuláře.

Nalezené záznamy				
Kliknutím na řádek se záznamem jej vyplníte do formuláře.				
Křestitel	Porodní bába	Otec Otcův otec Otcova matka Otcův děda (po matce) Otcova bába (po matce)	Matka Matčín otec Matčína matka Matčín děda (po matce) Matčína bába (po matce)	Kmotři
Anton Dostal lokální kurát	Josefa Blažek von Babitz	- Anton Hřebíček, Schmeltzmeister; in Josefsthal - Josef Hřebíček, Müllermeisters; in Blansko - Franciska Kučera - -	- Filipine - Martin Touchin, Förmers; in Josefsthal - Maria Opletal - -	- Josef Hřebíček, Müllermeister; in Klein Kinitz - Anna - -
Anton Dostal lokální kurát	Josefa Blažek von Babitz	- Mathias Kostínek, Bahnwächter; in Adamsthal - Thomas Kostínek, Schneider; in Drasow - Catharina Doliška - -	- Theresia - Josef Navratil, Schneiders; in Klein Věstin - Rosalia Pokorný - -	- Georg Klepárník, Häusler; in Adamsthal - Franciska - -
				Zavřít

Obrázek B.4: Návrhy celých záznamů.




## Našeptávač

Seznam návrhů je zobrazen u textového pole automaticky po zahájení psaní. Návrhy jsou zobrazeny pod tímto textovým polem. První položka ze seznamu návrhů je automaticky zvýrazněna a stačí pouze potvrdit její volbu. Našeptávač je možné ovládat pouze s využitím klávesnice:

- Šipka **nahoru/dolů** – pohyb v seznamu s návrhy směrem nahoru/dolů. Při zavřeném seznamu návrhů bude tento seznam otevřen.
- **Escape** – uzavře seznam návrhů.
- **Enter** – vyplní textové pole aktuálně zvýrazněnou položkou.
- **Tab** – vyplní textové pole aktuálně zvýrazněnou položkou a zvýrazní další textové pole.


## Vyhledávání návrhů osob

Návrhy k jednotlivým osobám jsou vyhledávány pouze pokud jsou vyplněny všechny požadované údaje. Pokud jsou k nějaké osobě nalezeny návrhy, objeví se v daném bloku tlačítko , pomocí kterého jsou návrhy údajů k dané osobě přístupné.

Aby mohly být návrhy k jednotlivým osobám vyhledány, je potřeba, aby byly zadány potřebné údaje. K vyhledání návrhů na *křtitele* je potřeba zadat alespoň jednu z hodnot *datum křtu*, *obec narození* nebo *příjmení křtitele*. Čím více z těchto údajů bude zadáno, tím přesnější budou zobrazované návrhy. Vyhledání návrhů na *porodní bábu* má stejné podmínky. Jediným rozdílem oproti křtiteli je použití *data narození*.

Návrhy na *otce* jsou vyhledávány na základě jeho *jména*, *příjmení*. Opět zde platí, že čím více údajů bude zadáno, tím přesnější budou návrhy. Informace o *ostatních předcích* a *kmotrech* jsou vyhledávány na základě nalezených návrhů s informacemi o otci.

## Odstranění údajů

Vyplněné údaje lze u jednotlivých osob odstranit tlačítkem , které je k dispozici v jednotlivých blocích formuláře.

## Upozornění

System vyhledávání návrhů pracuje s již existujícími záznamy, jejichž údaje mohou být zapsány odlišnými způsoby nebo mohou obsahovat rozdílné údaje, a tak nemusejí přesně odpovídat potřebným údajům aktuálně přepisovaného matričního záznamu. Z tohoto důvodu by měly být všechny údaje zapsané inteligentním doplňováním pečlivě zkontrolovány, aby bylo zamezeno zanášení chyb do přepisů.

## Příloha C

# Testovací protokol

### Příprava

1. Prostudujte návod k použití inteligentního doplňování matričního záznamu.
2. Seznamte se s uživatelským rozhraním pro přepis matričních záznamů ve webové aplikaci DEMoS.
3. V prohlížeči si otevřete stránku, na které naleznete digitalizovanou stránku matriky.  
<http://actapublica.eu/matriky/brno/prohlizec/10255/?strana=28>

### Přepis č. 1

1. Otevřete nastavení inteligentního doplňování a deaktivujte všechny možnosti.
2. Začněte měřit čas.
3. Přepište matriční záznam.
4. Přepsaný záznam pečlivě překontrolujte.
5. Zastavte měření času. Výsledný čas запиšte do dotazníku.
6. Přepsaný záznam neukládejte a obnovte stránku pro vyčištění formuláře.

### Přepis č. 2

1. Otevřete nastavení inteligentního doplňování a aktivujte pouze možnost *Našeptávač*.
2. Začněte měřit čas.
3. Přepište matriční záznam.
4. Přepsaný záznam pečlivě překontrolujte.
5. Zastavte měření času. Výsledný čas запиšte do dotazníku.
6. Přepsaný záznam neukládejte a obnovte stránku pro vyčištění formuláře.

### **Přepis č. 3**

1. Otevřete nastavení inteligentního doplňování a aktivujte pouze možnosti *Našeptávač* a *Návrhy jednotlivých osob*.
2. Začněte měřit čas.
3. Přepište matriční záznam.
4. Přepsaný záznam pečlivě překontrolujte.
5. Zastavte měření času. Výsledný čas zapište do dotazníku.
6. Přepsaný záznam neukládejte a obnovte stránku pro vyčištění formuláře.

### **Přepis č. 4**

1. Otevřete nastavení inteligentního doplňování a aktivujte pouze možnosti *Našeptávač*, *Návrhy jednotlivých osob* a *Návrh celého záznamu*.
2. Začněte měřit čas.
3. Přepište matriční záznam.
4. Přepsaný záznam pečlivě překontrolujte.
5. Zastavte měření času. Výsledný čas zapište do dotazníku.
6. Přepsaný záznam neukládejte a obnovte stránku pro vyčištění formuláře.

## Příloha D

# Dotazník

### Příprava

1. Byl návod k použití dobře srozumitelný a pochopitelný?
  - (a) Ano, návod je zpracován kvalitně
  - (b) Ano s výhradami
  - (c) Ne, s pochopením jsem měl(a) problémy
  - (d) Ne, návod jsem nepochopil(a)

### Přepis č. 1

1. Čas přepisu:

### Přepis č. 2

1. Čas přepisu:
2. Usnadnilo Vám *našeptávač* přepis záznamu?
  - (a) ANO
  - (b) NE
3. Zobrazovala se požadovaná slova v seznamu návrhů po zadání prvního písmene?
  - (a) Ano vždy
  - (b) Téměř vždy
  - (c) Někdy
  - (d) Ne nikdy
4. Kolik písmen bylo potřeba přibližně zadat, než se požadované slovo zobrazilo na první pozici seznamu návrhů?

### Přepis č. 3

1. Čas přepisu:
2. Usnadnilo Vám *zobrazování návrhů pro jednotlivé osoby* přepis záznamu?
  - (a) ANO
  - (b) NE
3. Věděli jste, jak zobrazit návrhy pro jednotlivé osoby?
  - (a) ANO
  - (b) NE
4. Byly návrhy osob přesné?
  - (a) Ano
  - (b) Ano s občasnými odchylkami ve způsobu zápisu
  - (c) Ano, některé údaje byly chybné
  - (d) Ne, navrhované osoby neodpovídaly

### Přepis č. 4

1. Čas přepisu:
2. Usnadnilo Vám *zobrazení návrhů na celý záznam* přepis záznamu?
  - (a) ANO
  - (b) NE
3. Odpovídal některý z navrhovaných záznamů požadovaným údajům?
  - (a) Ano přesně
  - (b) Ano, bylo třeba upravit zápis některých údajů
  - (c) Částečně, některé osoby bylo potřeba přepsat
  - (d) Ne, návrh neodpovídal

### Závěr

1. Jaké úpravy či vylepšení byste doporučili?
2. Jaké **výhody** mělo inteligentní doplňování?
3. Jaké **nevýhody** mělo inteligentní doplňování?

## Příloha E

# Obsah přiloženého paměťového média

/src – Zdrojové soubory

/doc/PDF – Technická zpráva ve formátu PDF

/doc/latex – Zdrojové soubory technické zprávy

/readme.md – Návod k instalaci