



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

# **VYKRESLOVÁNÍ RODOKMENŮ VE WEBOVÝCH STRÁNKÁCH**

FAMILY TREE DRAWING IN WEB PAGES

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MATĚJ ZEDNÍK**

**VEDOUcí PRÁCE**

SUPERVISOR

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

**BRNO 2023**

## Zadání bakalářské práce



144927

Ústav: Ústav inteligentních systémů (UITS)  
Student: **Zedník Matěj**  
Program: Informační technologie  
Specializace: Informační technologie  
Název: **Vykreslování rodokmenů ve webových stránkách**  
Kategorie: Webové aplikace  
Akademický rok: 2022/23

### Zadání:

1. Nastudujte si základy genealogie, především způsoby zobrazování rodokmenů. Dále nastudujte knihovny pro zobrazování dat ve stromové struktuře ve webových prohlížečích.
2. Na základě nastudovaných knihoven vyberte nejvhodnější knihovnu pro zakomponování do vybraných webových stránek. Knihovna musí umožňovat zobrazení vývodu (všechny předky) a rozrodu (všechny potomky) i kombinaci (všechny předky i potomky). Rodokmen také musí být interaktivní, tzn. po kliknutí na vybranou osobu se buď o ní zobrazí informace nebo se přejde na jinou stránku.
3. Vybranou knihovnu včetně potřebného rozšíření zakomponujte do požadovaných webových stránek a otestujte.

### Literatura:

- CZERNÍN, David. Uživatelské rozhraní pro přepis matrik. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. František Vídeňský

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání

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

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**  
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.  
Datum zadání: 1.11.2022  
Termín pro odevzdání: 17.5.2023  
Datum schválení: 3.11.2022

## Abstrakt

Tato práce pojednává o zobrazení interaktivních rodokmenů ve webovém prohlížeči pomocí vhodně zvolené knihovny. K řešení byla využita knihovna naprogramovaná v jazyce Javascript, která byla poupravena a vylepšena. Pro získání potřebných dat rodokmenu je využit soubor typu GEDCOM anebo načtení z grafové databáze Neo4j, který se po analýze převede na strukturu typu JSON. Tato struktura je následně předána modifikované knihovně pro zobrazení rodokmenu. Výsledkem je tedy možnost zobrazit rodokmen kteréhokoli člena rodiny včetně všech jeho předků anebo i potomků.

## Abstract

This thesis deals with displaying interactive family trees in a browser using a suitable library of choice. A library programmed in Javascript was used for the solution, which was modified and improved. To obtain the necessary family tree data, a Gedcom-type file is used, which after analysis is converted to a JSON-type structure. This structure is then passed to the modified library to display the family tree. It is therefore possible to display any family's family tree including all its ancestors or a member and descendants.

## Klíčová slova

Rodokmen, webová stránka, grafová databáze Neo4j, soubor Gedcom, probant, programovací jazyk Javascript, knihovna Family-chart, genealogie, předci, potomci

## Keywords

Family tree, webpage, Neo4j graphic database, Gedcom file, Javascript programming language, Family-chart library, genealogy, ancestry, progeny

## Citace

ZEDNÍK, Matěj. *Vykreslování rodokmenů ve webových stránkách*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

# Vykreslování rodokmenů ve webových stránkách

## Prohlášení

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

.....  
Matěj Zedník  
17. května 2023

## Poděkování

Tímto způsobem bych chtěl poděkovat panu Ing. Jaroslavu Rozmanovi, Ph.D. za pomoc, ochotu a trpělivost při vedení bakalářské práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
<b>2</b>	<b>Studium problematiky</b>	<b>6</b>
2.1	Genealogie . . . . .	6
2.2	Souborové formáty . . . . .	8
2.3	Grafové databáze . . . . .	14
<b>3</b>	<b>Analýza požadavků a návrh řešení</b>	<b>18</b>
3.1	Použité technologie . . . . .	18
3.2	Knihovny programovacího jazyka Javascript . . . . .	21
<b>4</b>	<b>Implementace a výsledné zobrazení rodokmenů</b>	<b>25</b>
4.1	Analýza a zpracování vstupních dat . . . . .	25
4.2	Základní nastavení zobrazení rodokmenu . . . . .	29
4.3	Změny knihovny Family-chart a vykreslení rodokmenu . . . . .	30
4.4	Výsledné zobrazení rodokmenu . . . . .	36
4.5	Možnosti rozšíření . . . . .	38
<b>5</b>	<b>Závěr</b>	<b>39</b>
	<b>Literatura</b>	<b>40</b>

# Seznam obrázků

2.1	Grafické znázornění rozrodu přes ženskou i mužskou linii s popisky . . . . .	7
2.2	Grafické znázornění vývodu přes ženskou i mužskou linii s popisky . . . . .	7
2.3	Grafické znázornění rodokmenu přes mužskou linii s ukázkou neprovdané a provdané dcery potomků . . . . .	8
2.4	Grafické znázornění rodiny Simpsonových . . . . .	9
2.5	Grafické znázornění rodokmenu rodiny Novákových. Karel Novák nahoře, pod ním jeho první a druhá manželka a potomci, které měl s jednotlivými manželkami. . . . .	14
2.6	Ukázka grafu uloženého v grafové databázi Neo4j. Kde fialové uzly představují jednotlivé osoby a šipky představují vztah mezi osobou odkud vedou k dané osobě na kterou ukazují. . . . .	15
2.7	Ukázka grafu uloženého v grafové databázi Neo4j. Kde fialové uzly představují osoby a modré uzly města. Šipky představují vztahy mezi jednotlivými uzly. . . . .	16
2.8	Grafické znázornění komunikace mezi klientem a webovým serverem pomocí HTTP protokolu přes metodu POST. . . . .	17
3.1	Grafické znázornění komunikace webového prohlížeče s webovým serverem. . . . .	19
3.2	Grafické znázornění komunikace webového prohlížeče s webovým serverem při zpracování dynamické webové stránky na straně serveru. . . . .	20
3.3	Grafické znázornění komunikace webového prohlížeče s webovým serverem při zpracování dynamické webové stránky na straně uživatele. . . . .	20
3.4	Grafické znázornění rodokmenu rodiny Simpsonových pomocí knihovny dTree. . . . .	21
3.5	Grafické znázornění rodokmenu rodiny Kennedyových pomocí knihovny yFiles for HTML. Převzato z [1]. . . . .	22
3.6	Ukázka grafu se stromovou strukturou vykreslený pomocí knihovny D3.js převzato z ukázkových grafů z oficiální stránky knihovny D3.js [2]. . . . .	23
3.7	Ukázka zobrazení rodokmenu Britské královské rodiny pomocí knihovny Family-chart [14]. . . . .	24
3.8	Ukázka, jak knihovna Family-chart ukazuje možnost zobrazit rodokmen dané osoby, když existují osoby, které nepatří k již vykreslenému rodokmenu, ale k rodokmenu dané osoby, u které jsou tyto objekty. . . . .	24
4.1	Grafické znázornění postupu mezi jednotlivými skripty od zpracování vstupních dat až po vykreslení výsledného rodokmenu. . . . .	25
4.2	Grafické znázornění posloupnosti volání a zpracování jednotlivých hlavních funkcí ve skriptu parse-neo4j.js včetně zobrazení předávání informací přes pole osob. . . . .	27

4.3	Grafické znázornění posloupnosti volání a zpracování jednotlivých hlavních funkcí ve skriptu parse-gedcom.js včetně zobrazení předávání jednotlivých parametrů. . . . .	28
4.4	Grafické znázornění barevného rozdílu pozadí kartiček probanta ženského pohlaví oproti kartičkám jeho předků. . . . .	33
4.5	Grafická ukázka části vykresleného rodokmenu v případě, že jsou v rodokmenu duplicitní osoby špatně spojené s jejich manželky či manžely. . . . .	34
4.6	Grafické znázornění vykreslení duplicitních osob v rodokmenu se správným napojením linek jejich manželek a manželů. . . . .	34
4.7	Grafická ukázka části výsledného vykreslení rodokmenu obsahující muže a jeho ženu. . . . .	36
4.8	Grafická ukázka části výsledného vykreslení rodokmenu obsahující muže a jeho ženu. . . . .	37
4.9	Porovnání rozepnutých a sepnutých sponek u stejného probanta. Nejprve všechny sponky rozepnuté, poté sepnutá sponka pro předky probanta a na- posled rozepnutá sponka předků, ale sepnutá sponka potomků. . . . .	38

# Kapitola 1

## Úvod

Z digitalizací matričních záznamů došlo ke snazšímu přístupu k datům o rodech. Tím se i usnadnilo pátrání po předcích a různých větví všech rodů. Tato data se dají načítat do různých systémů, a to jak do systémů fungujících pouze lokálně na počítačích, tak i do systémů fungujících na webových stránkách. Zrovna webové stránky a aplikace běžící na nich jsou dnes dostupné komukoliv s nějakým chytrým zařízením, ať už je to počítač, chytrý telefon, tablet anebo například i chytrá televize. Možností dostat se ke genealogickým datům odkudkoliv kde je dostupný internet přichází i touha lidí zobrazovat si tato data a například je i porovnávat se známými. Každý z těchto badatelů, kteří zkoumají genealogická data a chtějí znát historii své anebo případně i jiné rodiny, by ocenil mít možnost data jednoduše zobrazit a mít možnost toto zobrazení provést na jakémkoliv chytrém zařízení v podstatě kdykoliv bude potřebovat. Přehledné zobrazení a vykreslení základních informací o každém členovi a vztahů mezi členy jde pomocí rodokmenu dané rodiny.

Tato bakalářská práce se zabývá právě vykreslením takovýchto rodokmenů, které každému uživateli dokážou zobrazit ve webovém prohlížeči. Pro vykreslení rodokmenů na webových stránkách existuje spousta nástrojů, ale většina z nich zvládá vykreslit pouze menší rodokmeny a bez nějakých složitých vztahů mezi jednotlivými členy.

Mým osobním zájmem při výběru tohoto tématu bylo osvojení technologií pro vytváření webových aplikací. Většinu technologií jsem v předchozích projektech neměl šanci otestovat a zadání této bakalářské práce napovídalo, že je tím správným pro naplnění mých vzdělávacích cílů v dané oblasti.

Výsledkem této bakalářské práce má být vykreslení přehledného a interaktivního rodokmenu. Výsledný rodokmen by měl mít možnosti vykreslení, jak předků, tak i potomků. Pro vykreslení rodokmenu musí být předána data o členech a také data o vztazích mezi jednotlivými členy. Tato data se dají získat z různých databází anebo načíst ze souboru. Cílem této bakalářské práce je správné načtení těchto dat z grafové databáze Neo4j anebo import dat ze souboru Gedcom a následně tyto data analyzovat a správně zobrazit na webové stránce.

Text bakalářské práce je rozdělen na pět základních kapitol. První kapitolou je úvod. Druhá kapitola se věnuje teoretickým poznatkům ohledně problematiky genealogie, souborových formátů určených k předávání genealogických dat a grafových databází, které mohou sloužit pro ukládání a předávání genealogických dat. Následuje kapitola, která se věnuje analýze technologií, které povedou ke splnění požadavků na tuto bakalářskou práci. Jedná se spíše o technický rozbor a přípravu vhodných technologií k implementaci, což je obsah čtvrté kapitoly této bakalářské práce. Tato kapitola popisuje jednotlivé části imple-



mentace, ukazuje výsledné zobrazení rodokmenu a speciálních případů. Obsahem čtvrté kapitoly jsou také možnosti, jak lze případně rozšířit tuto bakalářskou práci.

## Kapitola 2

# Studium problematiky

Tato kapitola pojednává o souvislostech vykreslování rodokmenů. Objasní vědu nad rodokmeny, která se nazývá genealogie. Seznámí se souborovými formáty GEDCOM a JSON, které jsou základem pro uložení, zobrazení a přenos dat mezi systémy.

### 2.1 Genealogie

Slovo genealogie pochází z řeckých slov „genos“ a „logos“ tedy rod a věda. Jedná se o pomocnou historickou vědu, která se zabývá příbuzenskými vztahy osob v rámci jedné rodiny anebo jednoho rodu. V posledních letech se zvedá počet laických badatelů, kteří chtějí zjistit informace o svých předcích a místech, kde jejich předci pobývali. Tento v podstatě soukromý zájem o historii své rodiny a svých předků se v češtině nazývá rodopis.

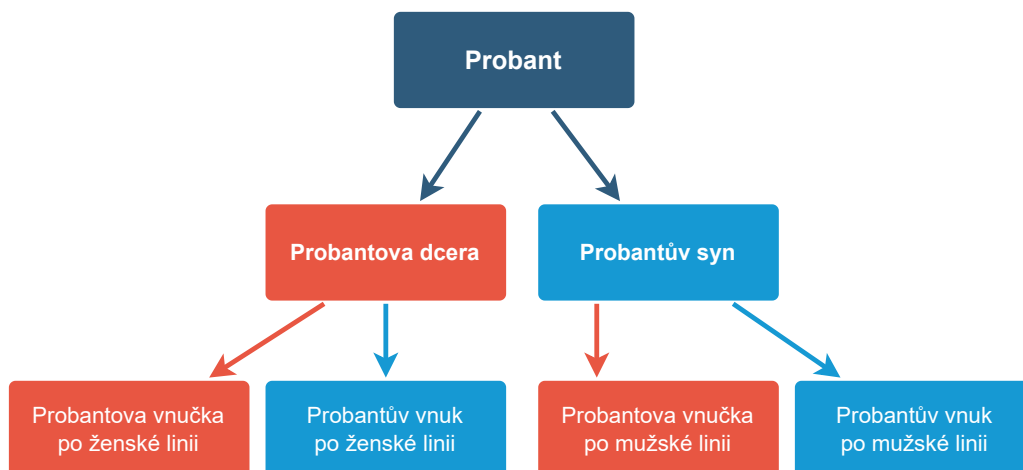
Využití genealogie můžeme najít i v ostatních pomocných historických vědách, zejména v heraldice a sfragistice. Dále je genealogie důležitá ve vědách demografie a sociologie, které se bez dat o rodech a příbuzenských vztazích nemohou obejít. Tato data se ovšem dají použít i pro získávání historických statistik typu porodnost, úmrtnost, kojenecká úmrtnost a jiné [10].

#### Rozrod

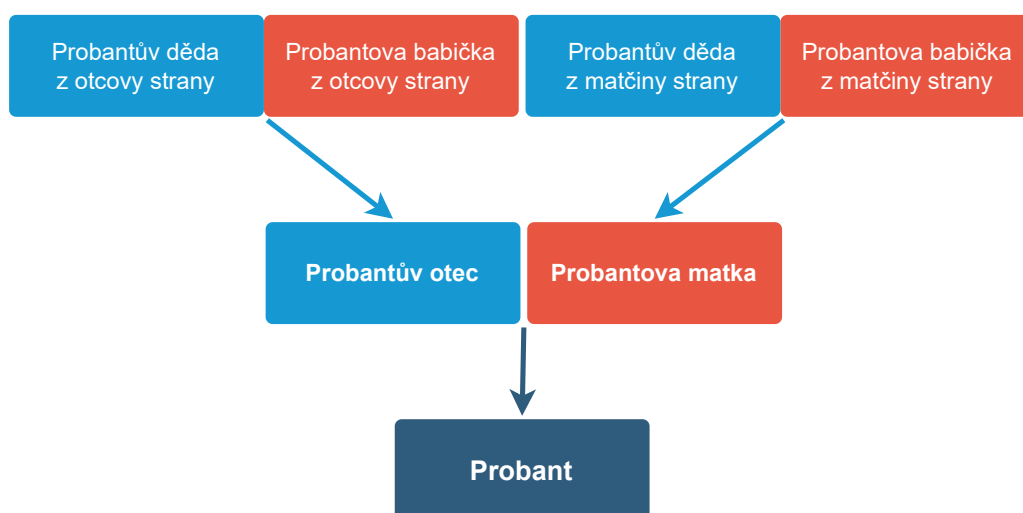
Jedná se o genealogické schéma, které sleduje všechny potomky vybrané osoby. Znázorněno na Obrázku 2.1. Takováto osoba, od které postupuje rozrod anebo také i vývod, se nazývá probant nebo také stěh. Pokud tedy genealogické schéma zobrazuje rozrod neboli descenci, tak se jedná o posloupnost jdoucí z minulosti do přítomnosti a to jak přes mužskou linii, tedy přes potomky "po meči", tak i přes ženskou linii, která byla v minulosti často vynechávána, protože byla právně, a ostatně i celou společností, brána jako méněcenná [15].

#### Vývod

Stejně jako u rozrodu se jedná také o genealogické schéma, akorát s tím rozdílem, že u vývodu neboli ascenci jde posloupnost od probanta směrem do minulosti. Jinými slovy, sledujeme předky probanta a to také přes mužskou (agnátní) i ženskou (kognátní) linii, tedy rodiče, rodiče rodičů a tak dále a dále do minulosti [11]. Ukázka vývodu k náhlednutí na obrázku 2.2.



Obrázek 2.1: Grafické znázornění rozrodu přes ženskou i mužskou linii s popisky

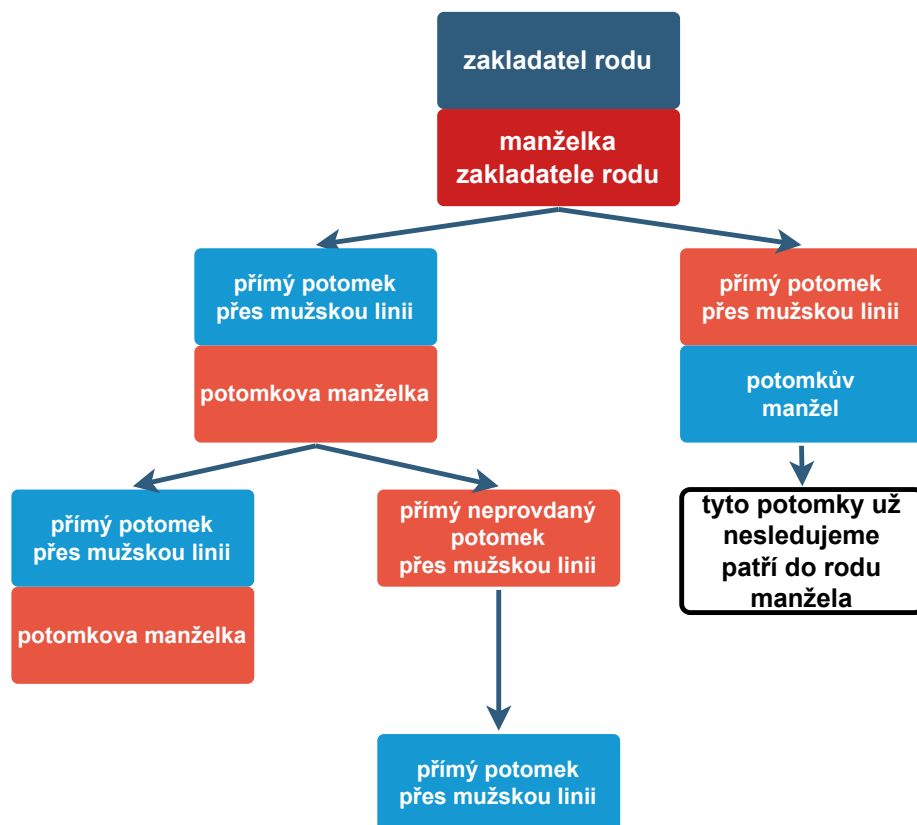


Obrázek 2.2: Grafické znázornění vývodu přes ženskou i mužskou linii s popisky

## Rodokmen

Rodokmen, narozdíl od rozrodu nebo vývodu, sleduje posloupnost všech potomků přes mužskou linii od nejstaršího zakladatele rodu (přímého předka). Do rodokmenu patří i potomci neprovdaných dcer nebo také muži, kteří nabyli jiného příjmení než zakladatel rodu, například sňatkem nebo povolenou změnou [3]. Ukázka rodokmenu na obrázku 2.3.

Definice rodokmenu je ale v této práci malinko jiná. Rodokmen je zde chápán spíše tak, jak jej chápe laická veřejnost. Takovýto rodokmen tedy zobrazuje i potomky přes ženskou linii a to i v případě provdaných dcer. Jde o to zobrazit všechny možné potomky a všechny možné předky, tak jak je to ve vývodu a rozrodu. Dále v práci je tedy pojem rodokmen myšlen jako spojení vývodu a rozrodu probanta.



Obrázek 2.3: Grafické znázornění rodokmenu přes mužskou linii s ukázkou neprovdané a provdané dcery potomků

## 2.2 Souborové formáty

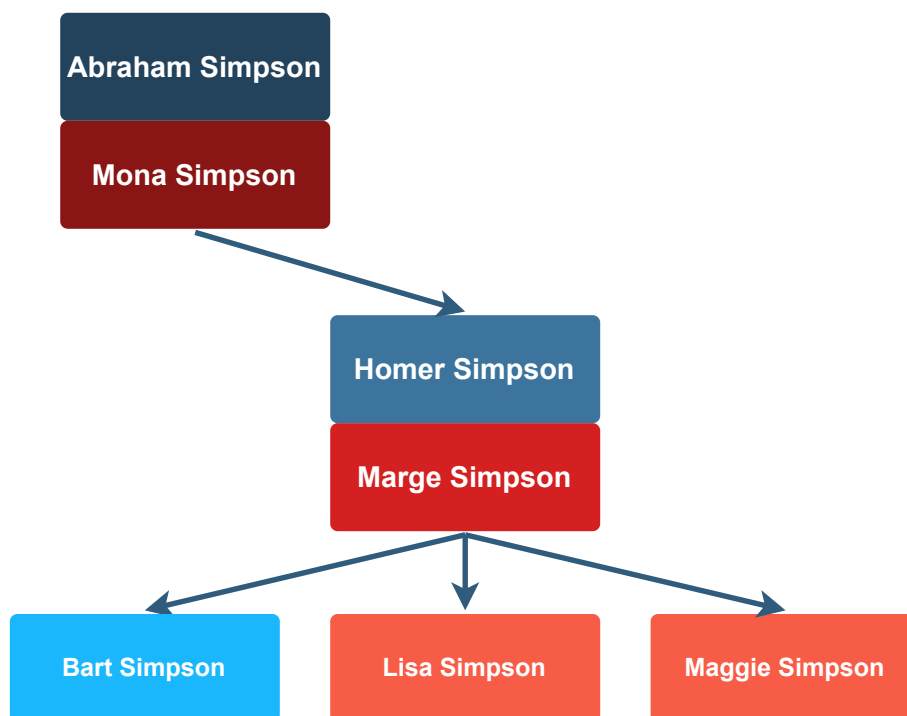
Jedná se o určení způsobu, jakým jsou data uchována v souborech anebo v databázích. Souborový formát určuje pravidla a pevně danou strukturu dat, čímž umožňuje programům správně načíst data a zpracovat je [13].

### GEDCOM souborový formát

Souborový formát GEDCOM je standard pro uchování a výměnu genealogických dat mezi různými systémy. Mezi informace, které se dají v souborovém formátu GEDCOM uchovat, patří podrobnosti o jednotlivých členech rodiny, jako je např. jméno, příjmení, datum narození, místo narození, datum úmrtí, místo úmrtí, do jakých rodin patří jako choť a do jakých jako potomek. Konkrétnější popis struktury uchování dat o osobě níže ve struktuře tohoto souborového formátu. Jako další informace se dají uchovat data o jednotlivých rodinách. Tedy, kdo v jedné konkrétní rodině figuruje jako choť a kdo jako potomek. Přitom se jedná o data pouze o jedné generaci rodu. V jedné této rodině najdeme otce, matku, syny a dcery. Jiní členové, jako jsou děda, vnuci, atd., uchovává souborový formát GEDCOM v dalších rodinách zase jako otce, matku, syny, dcery. Jinými slovy, v jedné rodině uchovává tento formát otce, ale jeho otce nalezneme v rodině jiné, kde je původní otec uchován jako syn.

## Struktura souborového formátu GEDCOM

Struktura je rozdělena do bloků, které jsou dále rozděleny na řádky a každý řádek je rozdělen na tři hodnoty mezerou. Začátek bloku se pozná pomocí hodnoty „0“ na prvním řádku v první hodnotě, následované druhou hodnotou, a to unikátním identifikátorem tohoto bloku. Poslední, tedy třetí hodnota, obsahuje označení typu bloku. Například typ **INDI** označuje, že v tomto bloku jsou data o jedné osobě, nebo typ **FAM**, který označuje, že v bloku jsou data o rodině. Konec bloku není označen a pozná se pouze začátkem nového bloku anebo koncovým řádkem, který má jako první hodnotu také „0“ a jako druhou hodnotu již nemá identifikátor bloku, ale označení konce souborového formátu GEDCOM tedy **TRLR**. Níže rozbor bloků s daty o osobách a také rozbor bloků s daty o rodinách z rodu [7]. Ukázkový rozbor souborového formátu GEDCOM rodiny Simpsonových zobrazených na obrázku 2.4.



Obrázek 2.4: Grafické znázornění rodiny Simpsonových

Ukázky z převzatého souboru [4].

```
0 @Homer_Simpson@ INDI
1 NAME Homer /Simpson/
2 GIVN Homer
2 SURN Simpson
1 SEX M
1 FAMC @F0002@
1 FAMS @F0000@
1 OBJE
2 FORM URL
2 FILE http://en.wikipedia.org/wiki/Homer_Simpson
1 OBJE
```

```

2 FORM URL
2 FILE safety.officer@springfieldnuclear.com
1 CHAN
2 DATE 11 FEB 2007
3 TIME 15:05:36
0 @Lisa_Simpson@ INDI
1 NAME Lisa /Simpson/
2 GIVN Lisa
2 SURN Simpson
1 SEX F
1 FAMC @F0000@
1 OBJE
2 FORM URL
2 FILE lisa@springfieldhigh.edu
1 CHAN
2 DATE 11 FEB 2007
3 TIME 15:04:14

```

Tento jeden blok výše obsahuje ukázková data o jedné osobě, konkrétněji o Homeru Simpsonovi, ze souborového formátu GEDCOM. Na prvním řádku, jak bylo vysvětleno výše 2.2, je na druhé hodnotě jednoznačný identifikátor tohoto bloku, tedy `@Homer_simpson@`, který je použit například v blocích s daty o rodině, jako takový odkaz na tento blok s daty o osobě.

Další řádky tohoto bloku jsou rozdělené podle druhé hodnoty:

**NAME** Většinou následuje hned po začínajícím řádku a obsahuje spojení dat z řádků obsahujících data o jménu, příjmení anebo také přezdívky. V ukázce `Homer /Simpson/`.

**GIVN** Řádek s touto hodnotou obsahuje od třetí hodnoty až po konec řádku jméno osoby, v ukázce pouze jedna hodnota `Homer`.

**SURN** Tento řádek obsahuje data o příjmení osoby daného bloku. V ukázce `Simpson`.

**NICK** Pokud je toto druhá hodnota na řádku, tak třetí až poslední hodnota na řádku určuje přezdívku osoby z bloku.

**SEX** Na tomto řádku je uloženo pohlaví osoby, velké „M“ pro mužské a velké „F“ pro ženské.

**FAMC** Třetí hodnota na tomto řádku obsahuje identifikátor rodiny, do které patří osoba jako syn či dcera.

**FAMS** Podobně jako FAMC, akorát třetí hodnota obsahuje identifikátor rodiny, do které patří osoba jako otec či matka nebo v případě bezdětného manželství jako manžel či manželka.

**OBJE** Zde jsou dvě možnosti. Pokud je zde třetí hodnota, tak nám určuje identifikátor a tedy odkaz na objekt, který je popsán jinde v souboru formátu GEDCOM. Pokud zde není třetí hodnota, je tento objekt popsán hned na dalších řádcích.

**CHAN** Tento typ druhé hodnoty určuje, že datum z následujícího řádku, tedy řádek s druhou hodnotou **DATE**, je datum poslední změny dat této osoby, a to stejné pro případný následující řádek s druhou hodnotou **TIME**, které určuje čas poslední změny.

**BIRT** Podobně jako typ **CHAN** akorát, že na následujících řádcích s první hodnotou 2 jsou data o narození dané osoby. **DATE** určuje datum narození, **TIME** čas narození, **PLAC** místo narození.

**DEAT** Vesměs stejné jako typ **BIRT** jen s tím rozdílem, že neobsahuje informace o narození, ale o úmrtí dané osoby. Takže následující řádky mohou obsahovat datum úmrtí, čas úmrtí anebo i místo úmrtí.

```
0 @F0000@ FAM
1 REFN 0
1 HUSB @Homer_Simpson@
1 WIFE @Marge_Simpson@
1 CHIL @Bart_Simpson@
1 CHIL @Maggie_Simpson@
1 CHIL @Lisa_Simpson@
1 CHAN
2 DATE 11 FEB 2007
3 TIME 15:04:14
0 @F0002@ FAM
1 REFN 2
1 HUSB @Abraham_Simpson@
1 WIFE @Mona_Simpson@
1 CHIL @Homer_Simpson@
1 CHAN
2 DATE 11 FEB 2007
3 TIME 15:05:36
0 TRLR
```

Tyto dva další bloky obsahují ukázková data o dvou rodinách. První rodina s identifikátorem **@F0000@** a druhá rodina s identifikátorem **@F0002@**. Jednotlivá data jsou rozdělená podle druhé hodnoty na každém řádku a to na tyto typy:

**HUSB** Na řádku s touto druhou hodnotou najdeme odkaz na blok s osobou, která je v této rodině jako otec a manžel.

**WIFE** Podobně jako **HUSB**, akorát v tomto případě se jedná o odkaz na blok s osobou, která je v této rodině jako matka a manželka.

**CHIL** V tomto případě se na třetí hodnotě nachází odkaz na osobu, která je v rodině jako syn či dcera. Synů a dcer může být v jedné rodině větší počet anebo také žádný počet.

**CHAN** Stejně jako u bloku s osobou se jedná o určení data a času poslední změny dat tohoto bloku.

**REFN** Jedná se o odkaz uložený k této rodině. Používá se v některých genealogických programech.

## Souborový formát JSON

Zkratka **JSON** znamená „Javascript object notation“. Vychází z programovacího jazyka Javascript a slouží pro zápis objektů. Jedná se o souborový formát s pevně danou strukturou

a pravidly, jak v něm mají být data zapsána. Je jednoduchý a dobře čitelný, a i proto se tento textový souborový formát velice hodí pro výměnu dat mezi systémy [5].

## Struktura souborového formátu JSON

Struktura tohoto textového souborového formátu se rozděluje na dvě další možné struktury. První je spojení názvu a hodnoty do páru:

```
{
  "název" : "hodnota",
  "jiný" : "jiná"
}
```

Složené závorky („{ }“) a sekce mezi nimi označují objekt. Takovýto objekt se dá dále vkládat do hodnoty. Poté může taková struktura vypadat třeba následovně:

```
{
  "název" : {
    "objekt" : "vložený",
    "místo" : "hodnoty"
  },
  "jiný" : "jiná"
}
```

Druhá možná struktura je takzvané pole, tedy několik objektů se stejnou strukturou, ve které mohou mít odlišné hodnoty. Takovéto pole se v souborovém formátu JSON pozná podle hranatých závorek („[ ]“). Předchozí ukázkou rozšíříme o pole se dvěma hodnotami:

```
{
  "název" : {
    "objekt" : "vložený",
    "místo" : "hodnoty"
  },
  "jiný" : "jiná",
  "pole" : [
    "hodnota", "další hodnota"
  ]
}
```

Stejně jako jde hodnota vyměnit za objekt a za pole, jde i do hodnot v poli dát objekt. Vzorová ukáзка:

```
{
  "název" : {
    "objekt" : "vložený",
    "místo" : "hodnoty"
  },
  "jiný" : "jiná",
  "pole" : [
    {"jméno": "příjmení"}, {"další jméno": "a příjmení"}
  ]
}
```



Anebo můžeme dát do pole další pole:

```
{
  "název" : {
    "objekt" : "vložený",
    "místo" : "hodnoty"
  },
  "jiný" : "jiná",
  "pole" : [
    [
      {"jméno":"příjmení"}, {"další jméno":"a příjmení"}
    ],
    [
      {"jméno":"příjmení"}, {"další jméno":"a příjmení"}
    ]
  ]
}
```

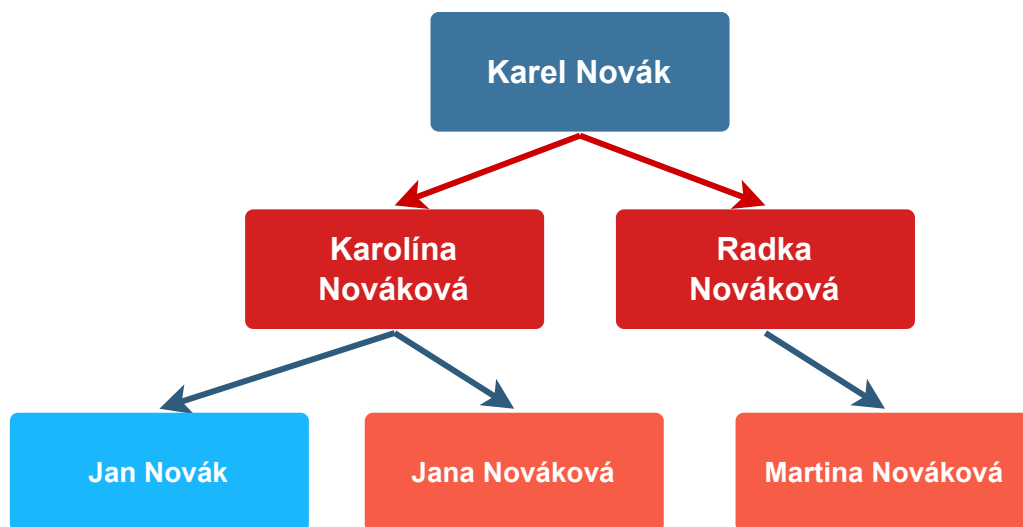
Objekty a pole se dají různě kombinovat, a dá se tak pomocí souborového formátu JSON popsat, uložit, anebo předat data v různých typech a možnostech. V případě genealogických dat se dá například udělat pole z osob a osoby se dají jako objekty se stejnou vnitřní strukturou. Například:

```
[
{
  "id" : "xnovak01"
  "jméno" : "Karel",
  "příjmení" : "Novák",
  "pohlaví" : "muž",
  "datum narození" : "21. 05. 2000",
  "místo narození" : "Nové Město na Moravě",
  "manželství" : [
    "xneuma00","xsklena03"
  ],
  "děti" : [
    "xnovak02", "xnovak03", "xnovak04"
  ]
},
{
  "id" : "xneuma00"
  "jméno" : "Karolína",
  "příjmení" : "Nováková",
  "pohlaví" : "žena",
  "datum narození" : "21. 05. 2000",
  "místo narození" : "Nové Město na Moravě",
  "manželství" : [
    "xnovak01"
  ],
}
```

```

    "děti" : [
        "xnovak02", "xnovak03"
    ]
}
]
```

V tomto příkladu je pole se dvěma osobami a každá osoba je jako objekt, který má nějaké své „id“ dále jméno, příjmení, pohlaví, datum a místo narození. Pak také pole svých manželství, protože se mohl rozvést a mít manželství více než jedno, ve kterém jsou uloženy „id“ odkazující se na osoby se kterými byl sezdán. A podobně je to s dětmi těchto osob, které jsou u každé osoby uloženy jako pole „id“ odkazující na jednotlivé děti této osoby. Pro lepší představu znázornění rodiny Novákových na obrázku 2.5.

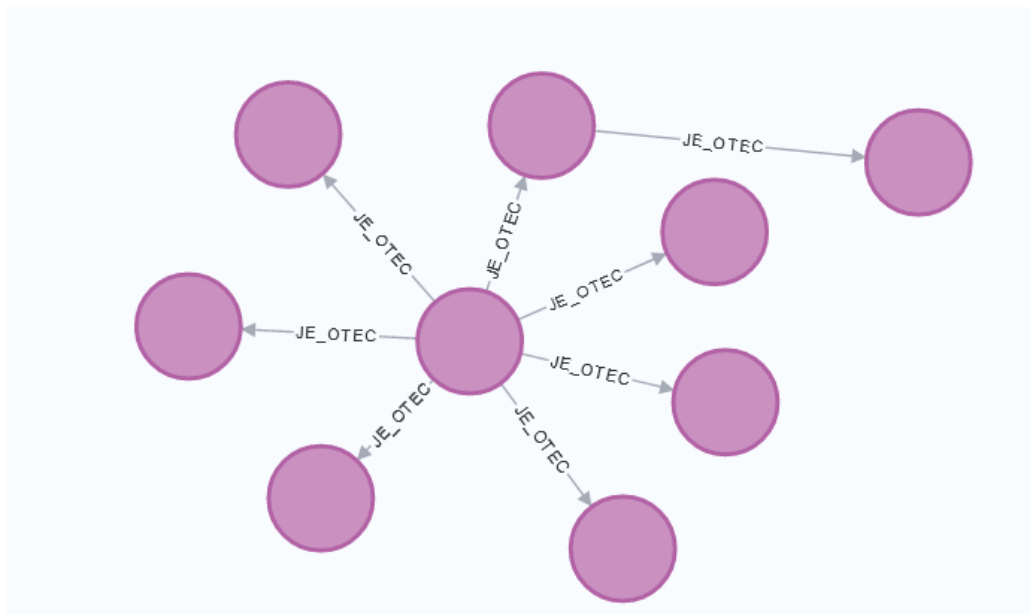


Obrázek 2.5: Grafické znázornění rodokmenu rodiny Novákových. Karel Novák nahoře, pod ním jeho první a druhá manželka a potomci, které měl s jednotlivými manželkami.

## 2.3 Grafové databáze

Jedná se o způsob uložení dat, jehož základem uložení je graf. Databáze si uchová jednotlivé uzly a vztahy mezi těmito uzly. Tyto uzly a vztahy mohou mít své označení, které je rozděluje do různých skupin. Uzly s označením „Osoba“ patří do skupiny osob. Vztahy s označením „JeOtec“ označují skupinu vztahů mezi osobami, které spolu mají vztah jako otec a syn. Ukázka grafu uloženého v grafové databázi na obrázku 2.6. Uzly kromě označení uchovávají data, která mají stejnou strukturu u všech uzlů z jedné skupiny neboli jednoho označení. Struktura je odlišná podle dat, která si chceme v jednotlivých skupinách uložit. Například u skupiny osob bude ve struktuře jméno, příjmení, datum narození, datum úmrtí, místo narození, místo úmrtí a případně i další. Ale naopak u skupiny třeba aut by mohlo být ve struktuře spíše značka, typ, motor, pohon, případně i další. U vztahů se ukládá pouze typ vztahu (je otec, vlastní, atd.), z jakého uzlu a do kterého uzlu vede v podobě jednoznačného identifikátoru daných uzlů. Jednoznačný identifikátor se ukládá i u jednotlivých vztahů.

Mezi výhody grafových databází oproti jiným databázím patří:



Obrázek 2.6: Ukázka grafu uloženého v grafové databázi Neo4j. Kde fialové uzly představují jednotlivé osoby a šipky představují vztah mezi osobou odkud vedou k dané osobě na kterou ukazují.

**Vyšší výkon** Zatímco u jiných databází (zejména u těch relačních) dochází ke snižování výkonu při větším počtu dat v databázi, a to kvůli prohledávání všech dat v tabulkách a sestavení výsledků z těchto dat. U grafových databází naopak dochází k nalezení správných dat v podstatě stejném čase jak u menšího tak i u většího počtu dat. Grafová databáze totiž lokalizuje dotazy a prochází tedy pouze část dat. Výkon je tedy závislý na velikosti zkoumané oblasti dat v grafu.

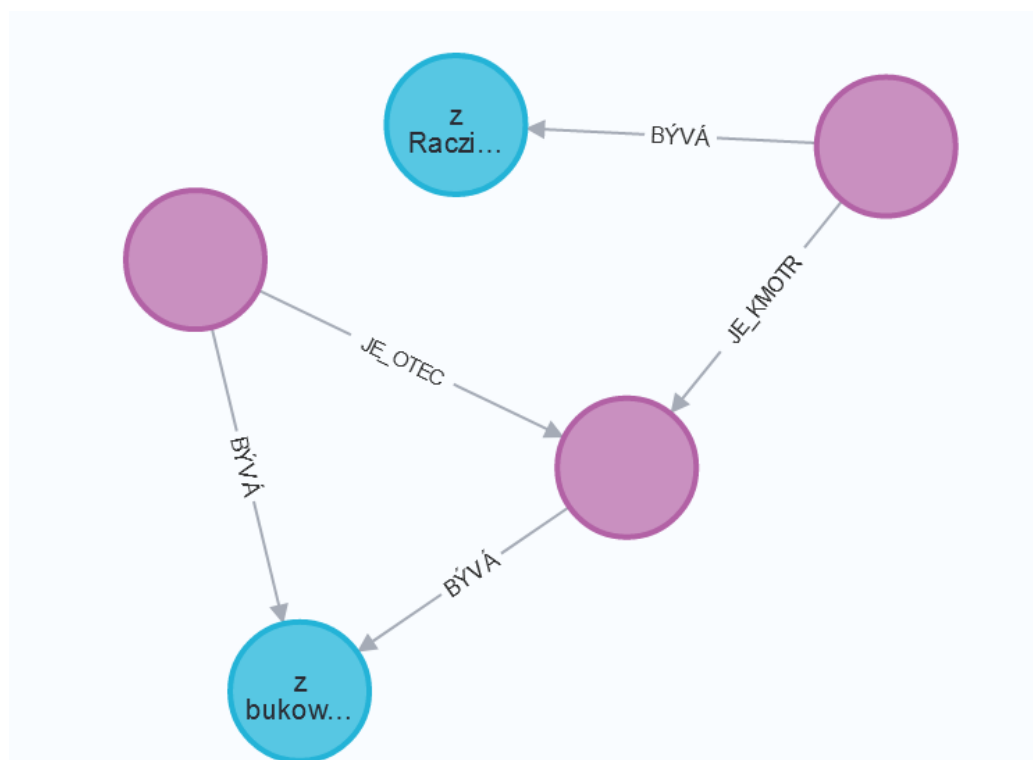
**Flexibilita** Grafové databáze mají flexibilní datový (grafový) model. Což dává jejich uživatelům z řad vývojářů možnost přidávat nové druhy uzlů, vztahů anebo nové podgrafy přímo do stávající běžící databáze bez toho aniž by museli měnit existující dotazy v již běžících aplikacích, které si vyčítají data z těchto databází. Díky flexibilitě také není nutné dopodrobna studovat a modelovat veškeré případy užití. Vše se může dělat postupně a za běhu a nenastane přitom žádný větší problém. Přidá se vztah, který spojí jednotlivé případy a pokračuje se. Díky těmto výhodám se snižují náklady na údržbu grafové databáze oproti ostatním typům databází.

Jelikož je základem grafových databází přímo graf, je možné pomocí přímého spojení procházet mezi sousedními uzly. Díky tomuto jevu lze tedy projít opravdovou spoustu uzlů v řádech sekund na rozdíl od ostatních databází, kde dochází ke spojování jednotlivých dat přes globální index, který je několikanásobně pomalejší. Grafové databáze jsou v posledních letech trend při vývoji různých sociálních sítí, systémů správy dat, geografických systémů, neurochirurgických systémů a spousty dalších. [12]

## Grafická databáze Neo4j

Typickým zástupcem grafických databází je Neo4j. Neo4j vyvinula švédská společnost „Neo Technology“, která ji zprvu nechala jako open source. Ovšem později rozšířila funkce této databáze a dnes je rozdělena do více produktů, kde některé jsou zdarma a některé s více funkcionalitami jsou již placené. Databáze je dostupná na většině operačních systémů, například na Windows, Linux, macOS. Java je programovací jazyk, na kterém je tato grafová databáze postavena. [8]

V grafové databázi Neo4j se posílají dotazy na data pomocí jazyku Cypher. Tento jazyk vychází z jazyka SQL akorát s tím rozdílem, že je cílený na grafové databáze, které prohledávají uzly a jejich vztahy v grafu. Díky jazyku Cypher se dají vyhledat data podle vlastností, na které se dotazujeme včetně vztahů, které dané uzly mají mít. Na vztahy se dá dotazovat ve třech možnostech, i když je vztah pouze z jednoho uzlu do druhého, tak je možné se dotazovat i na opačný směr anebo na oba směry zároveň.



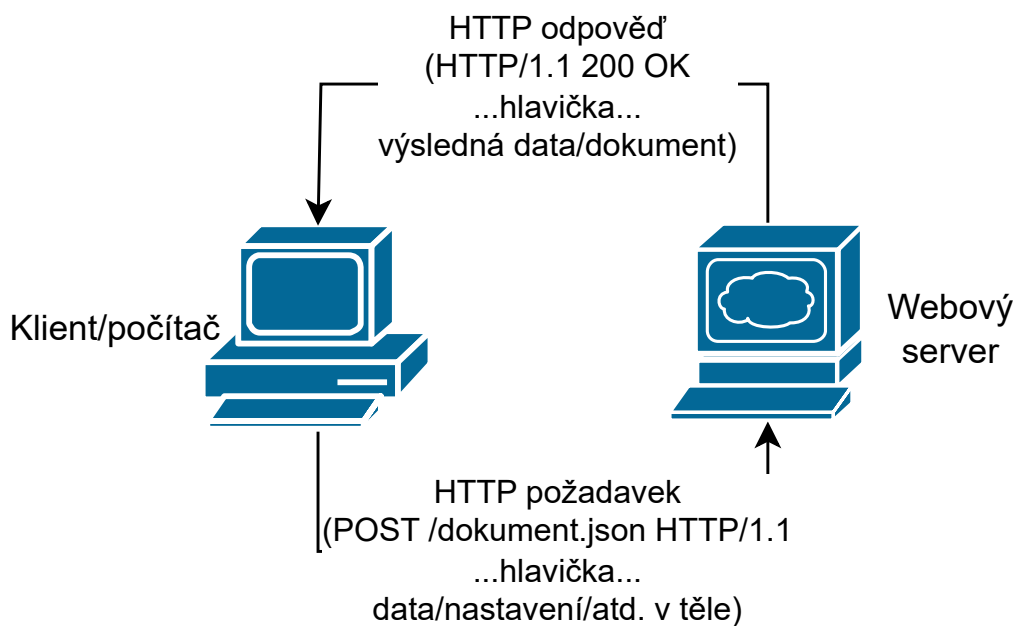
Obrázek 2.7: Ukázka grafu uloženého v grafové databázi Neo4j. Kde fialové uzly představují osoby a modré uzly města. Šipky představují vztahy mezi jednotlivými uzly.

## HTTP rozhraní pro Neo4j

S příchodem Neo4j verze 3.5 přišla i možnost dotazování na data v databázi pomocí takzvaných HTTP požadavků. HTTP je aplikační protokol určený pro přenos dokumentů, textu a dalších libovolných binárních dat mezi klientem a webovým serverem. HTTP požadavky se dají posílat pomocí dvou základních metod, a to buď pomocí metody GET anebo POST. V případě HTTP rozhraní pro databázi Neo4j je lepší metoda POST, která naopak od metody GET nabízí možnost předat data i od klienta směrem k webovému serveru, který je

může zpracovat a poslat odpověď zpátky. Většina programovacích jazyků nabízí možnost sestavení a odeslání HTTP požadavků na webový server, který se určí pomocí url adresy. Ukázka HTTP komunikace na obrázku 2.8.

HTTP rozhraní databáze Neo4j naslouchá příchozím HTTP požadavkům a zpracovává dotazy, které přijdou v jednotlivých HTTP požadavcích. Následně zpracuje dotaz, který je ve správné syntaxi jazyka Cypher a odešle klientovi zpátky řešení v odpovědi, která chodí také pomocí protokolu HTTP. Databáze Neo4j může chtít autentizaci klienta při příchozím HTTP požadavku. Možností autentizace je více.



Obrázek 2.8: Grafické znázornění komunikace mezi klientem a webovým serverem pomocí HTTP protokolu přes metodu POST.

## Kapitola 3

# Analýza požadavků a návrh řešení

V této kapitole se projdou všechny požadavky, které mají být výsledným systémem splněny. Každý požadavek je podrobně rozepsán a vysvětlen v jednotlivých sekcích. Analýza požadavků vedla i k návrhu řešení, který obsahuje technologie, nástroje a popis každé z nich.

### Funkční požadavky

Cílem je zobrazit uživateli rodokmen ve webovém prohlížeči, tak aby umožňoval zobrazit pouze vývod, rozrod anebo kombinaci rozrodu a vývodu, tedy celý rodokmen. Takovýto rodokmen musí být interaktivní, což znamená, že musí reagovat na uživatelovy pohyby nebo klikání myši. Zobrazit více informací po kliknutí na danou osobu nebo zobrazit její rodokmen.

Dále by měl rodokmen zobrazovat také rozvedené a znovu sezdané osoby, i když se znovu sezdají s jinou než původní osobou a také přiřadit děti těchto manželství ke správným rodičům.

Zobrazení rodokmenu by mělo mít možnost zabalit nějaké menší části rodiny, které uživatele třeba nezajímají, a zabalit tak třeba i celou větev. To všechno pro zmenšení a zpřehlednění zobrazení celého rodokmenu.

Také by měl rodokmen zobrazovat u všech potomků, předků anebo přímo probanta jejich manžele nebo manželky pod těmito osobami, tak aby se rodokmen rozrůstal spíše do výšky než do šířky.

### 3.1 Použité technologie

Tato sekce seznamuje s technologiemi, které jsou používány pro splnění funkčních požadavků této bakalářské práce. Vysvětluje pojmy okolo webových stránek, programovacích jazyků a jejich knihoven. U každé technologie vysvětluje, proč byla zvolena a jaké využití nabízí.

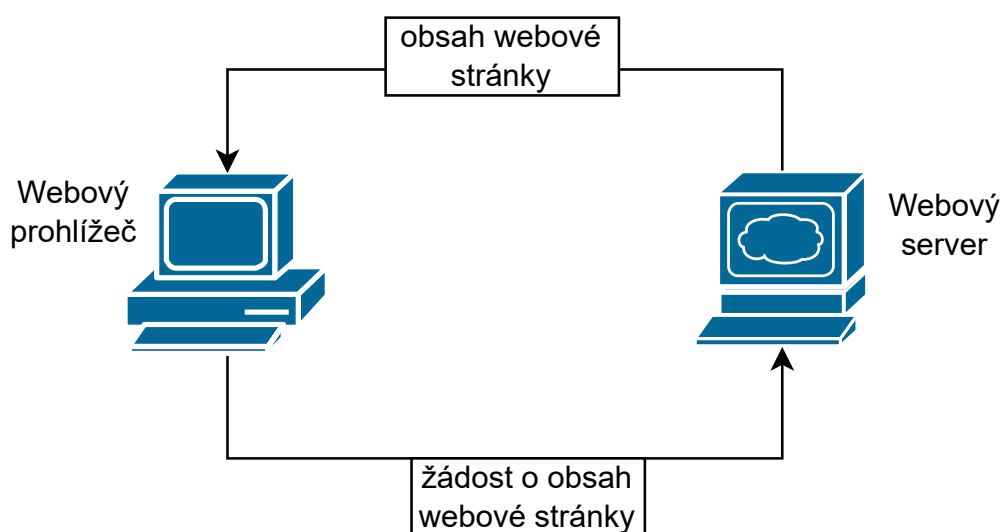
#### Webová stránka

Webová stránka je jenom jedna konkrétní stránka ve webové aplikaci. Webové stránky jsou tedy všechny jednotlivé stránky ve webové aplikaci. Konkrétně například školní web má několik webových stránek jako třeba stránku s informacemi o škole, stránku s informacemi pro uchazeče, stránku s jídelníčkem anebo spoustu jiných možných stránek. Podle článku Barbory Kodouškové si web lze představit jako knihu, která se skládá z jednotlivých listů – webových stránek [9].

Takovéto webové stránky se dále dají rozdělit na statické a dynamické. Rozdíl mezi těmito webovými stránkami popsán níže v sekcích 3.1 a 3.1.

### Statická webová stránka

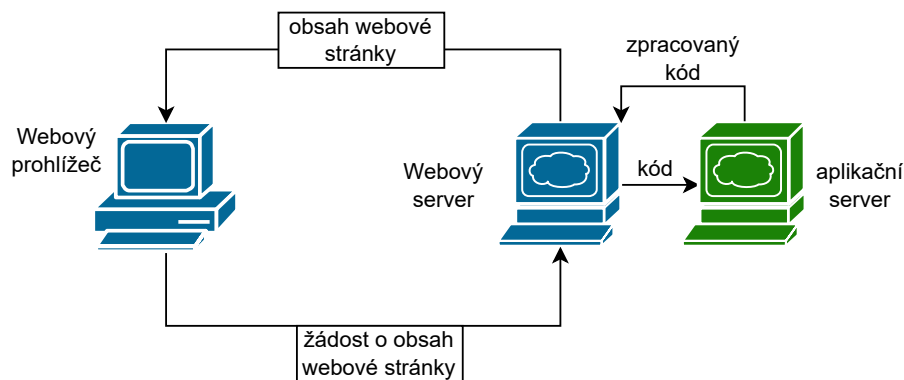
Statická webová stránka funguje tak, že poté co uživatel přistoupí na danou webovou stránku, tak jeho webový prohlížeč pošle žádost o obsah této webové stránky na webový server. Webový server pošle uživateli obsah webové stránky, o kterou zažádal jeho webový prohlížeč a webový prohlížeč obsah zobrazí. Pokud uživatel přejde na jinou webovou stránku daného webu, tak se proces znovu opakuje, akorát webový prohlížeč zažádá o danou jinou webovou stránku, kterou uživatel chce zobrazit. Takováto statická webová stránka nedokáže reagovat na různé akce od uživatele kromě načtení jiné stránky. Není tedy interaktivní a každá změna obsahu webové stránky jde přes webový server. Diagram zobrazující komunikaci webového prohlížeče uživatele a webového serveru na obrázku 3.1.



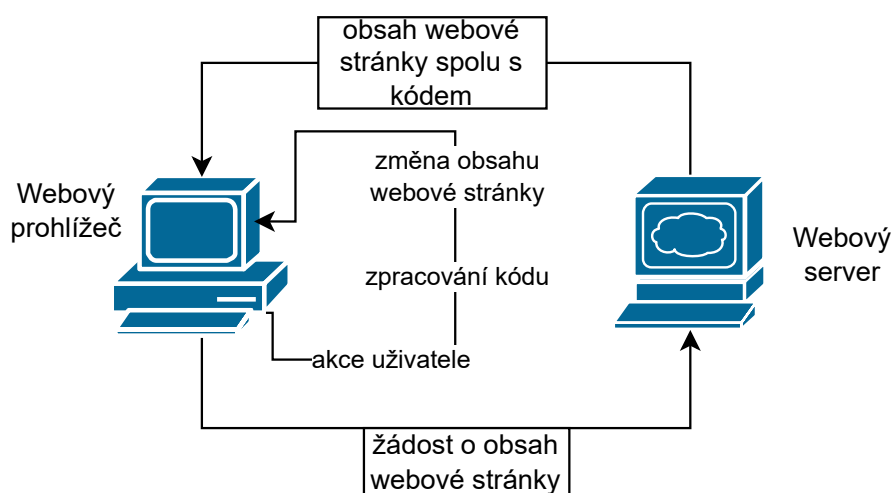
Obrázek 3.1: Grafické znázornění komunikace webového prohlížeče s webovým serverem.

### Dynamická webová stránka

Princip načtení obsahu je stejný jako u statické webové stránky popsané v sekci 3.1. Liší se tím, že webový server předá webovému prohlížeči ještě kód, který se má provést při určitých akcích uživatele. Tento kód se ovšem nemusí předávat vždy webovému prohlížeči, jelikož jsou dva způsoby, kde se tento kód provede. První je na straně uživatele a to tedy ve webovém prohlížeči a druhý je na aplikačním serveru, kdy webový server předá kód aplikačnímu serveru, ten jej provede a výsledek vrátí webovému serveru, který vrátí zase statickou webovou stránku webovému prohlížeči, který ji zobrazí uživateli. Znázornění takové komunikace na obrázku 3.2. V případě předávání kódu webovému prohlížeči dojde k provedení kódu předaného webovým serverem a zobrazením následné webové stránky. Po akci od uživatele (např. pohyb kurzorem myši) může dojít k provedení jiného kódu, aniž by se musela posílat žádost na webový server. Ukázka takové komunikace na obrázku 3.3. Pro splnění cílů této bakalářské práce byla vybrána možnost s předáváním kódu webovému prohlížeči.



Obrázek 3.2: Grafické znázornění komunikace webového prohlížeče s webovým serverem při zpracování dynamické webové stránky na straně serveru.



Obrázek 3.3: Grafické znázornění komunikace webového prohlížeče s webovým serverem při zpracování dynamické webové stránky na straně uživatele.

## Programovací jazyk Javascript

Rozdíl mezi statickou a dynamickou webovou stránkou popsaný výše nám zajistí právě programovací jazyk Javascript. Tento programovací jazyk je totiž standardem a v podstatě každý webový prohlížeč jej umí přeložit a zpracovat. Takže, když webový server pošle webovému prohlížeči uživatele webovou stránku doplněnou o kód psaný v programovacím jazyce Javascript, tak prohlížeč dokáže zpracovat kód a poté zajistit dynamiku dané webové stránky, aniž by musel posílat žádosti na webový server.

Programovací jazyk Javascript podporuje různé styly programování, a to objektově orientované, imperativní anebo deklarativní. Typy proměnných se přiřazují dynamicky, takže se nemusí při deklaraci označit typ proměnné a poté s tou proměnnou tak pracovat, ale dá se typ měnit v průběhu kódu. Další specifikace tohoto programovacího jazyka je funkce první třídy. Tedy, že se s funkcemi může nakládat stejně jako s proměnnými. Funkce se dají předávat v parametrech nebo se funkce může vracet v jiné funkci.

Ve webových stránkách, kde je zajištěna nějaká interakce nebo dynamika bez načtení dat z webového serveru, je s největší pravděpodobností použit právě tento programovací jazyk.



V průzkumu z roku 2010 společnosti Yahoo vyšlo, že v průměru pouhých 1,3 % uživatelů má na webovém prohlížeči zakázáno používat programovací jazyk Javascript [6]. Jedná se o 13 let starý průzkum a využívanost programovacího jazyka Javascript v následujících letech spíše stoupala. Co se týče využívání programovacího jazyka přímo vývojáři, tak zde podle statistiky vedené komunitou Stack Overflow v roce 2022 je tento programovací jazyk nejvíce používaný.

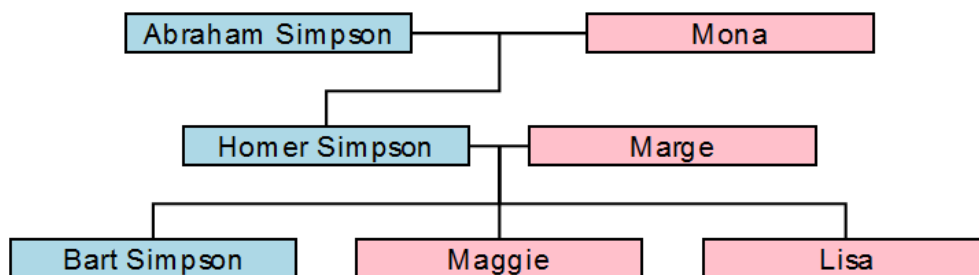
## 3.2 Knihovny programovacího jazyka Javascript

### Průzkum knihoven

Pro vykreslení rodokmenu se dá využít nějaký již existující nástroj a ten poté vylepšit. Takový nástroj, který se dá použít v implementaci tak, že se načtou jeho funkce a poté použijí se nazývá knihovna<sup>1</sup>. Mezi potencionální knihovny, které by splňovaly cíle této bakalářské práce patřily:

#### Knihovna dTree

Jedná se o knihovnu<sup>2</sup> pro programovací jazyk Javascript. Má licenci s otevřeným zdrojovým kódem. Jejím autorem je Erik Gartner. Pro vykreslení stačí předat soubor typu JSON. Ukázka vykreslení na obrázku 3.4. Vykreslení rodokmenu proběhne rychle a zobrazený rodokmen je přehledný a vizuálně vypadá dobře. Problém této knihovny, ale spočívá v zobrazení předků, zatímco potomky zobrazí správně, tak předky dokáže jen jednou linií. Jinými slovy zobrazí se osoba a její děti, také se správně zobrazí rodiče dané osoby, ale u jeho prarodičů se zobrazí pouze rodiče jeho otce anebo pouze rodiče jeho matky, nejde tedy současně zobrazit rozvětvení na všechny prarodiče. Tento problém se po analýze projevil jako těžko řešitelný a knihovna tedy byla označena za nevyhovující.



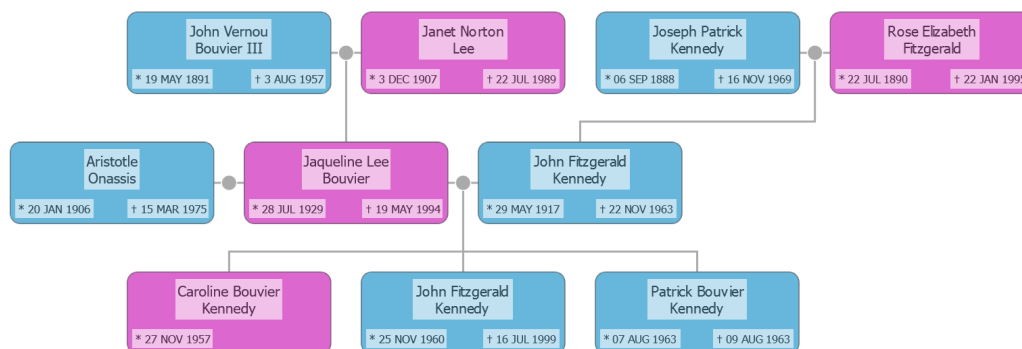
Obrázek 3.4: Grafické znázornění rodokmenu rodiny Simpsonových pomocí knihovny dTree.

<sup>1</sup>Přesnou definici knihovny v odvětví informačních technologií najdete zde <https://it-slovník.cz/pojem/knihovna>

<sup>2</sup>Knihovna pro zobrazení rodokmenu ve webové aplikaci <https://github.com/ErikGartner/dTree>

## yFiles for HTML

Tato knihovna<sup>3</sup> je určená pro vytváření diagramů a grafů ve webových aplikacích. Rodokmen se pomocí této knihovny dá zobrazit jako graf, který dokáže reagovat na uživatelské vstupy jako je třeba kliknutí na vybrané osoby. Zobrazí správně všechny předky i potomky, a to jak přes mužskou, tak i přes ženskou linii. Tato knihovna je, ale placená a startovací cena pro jednoho vývojáře a nasazení pro jednu webovou doménu je 300 000 korun českých, čímž se knihovna dostala na seznam nevyhovujících. Ukázka zobrazení pomocí této knihovny na obrázku 3.5.



Obrázek 3.5: Grafické znázornění rodokmenu rodiny Kennedyových pomocí knihovny yFiles for HTML. Převzato z [1].

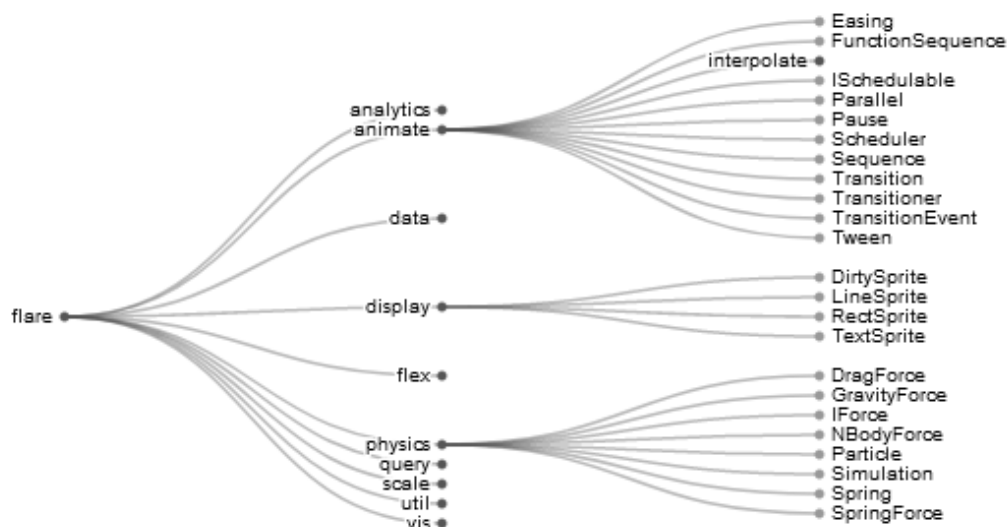
Nakonec jako nejvhodnější knihovna pro zobrazení rodokmenu a splnění stanovených cílů této bakalářské práce byla vybrána knihovna Family-chart 3.2, která staví na knihovně D3.js 3.2.

## Knihovna D3.js

Jedná se o knihovnu v programovacím jazyce Javascript. Pomocí této knihovny lze oživit data na webové stránce spolu se značkovacím jazykem HTML, SVG a kaskádových stylů, které slouží pro popis způsobu vykreslení jednotlivých elementů na webové stránce. Pomocí knihovny D3.js lze tedy provázat data s objektově orientovanou reprezentací XML nebo HTML dokumentu a poté na dokument aplikovat transformace řízené daty. Například se dá knihovna D3.js použít ke generování HTML tabulky z pole čísel nebo pomocí této knihovny se také dá vzít stejná data k vytvoření interaktivního pruhového grafu v značkovacím jazyce SVG s hladkými přechody a interakcí. Knihovna D3.js zajišťuje efektivní manipulaci s dokumenty na základě dat. Tím se vyhýbá proprietární reprezentaci a poskytuje mimořádnou flexibilitu, odhalující plné možnosti webových standardů, jako jsou značkovací jazyky HTML, SVG a kaskádové styly. S minimální režií je D3.js extrémně rychlý, podporuje velké datové sady a dynamické chování pro interakci a animaci. Funkční styl D3.js umožňuje opětovné použití kódu prostřednictvím rozmanité kolekce oficiálních a komunitou vyvinutých modulů [2]. Na obrázku 3.6 ukázka grafu se stromovou strukturou pomocí knihovny D3.js.

<sup>3</sup>Knihovna pro vytváření a editaci diagramů ve webových aplikacích od společnosti yWorks <https://www.yworks.com/yfiles-overview>

Tato knihovna je pod licencí ISC, a řadí se tedy do knihoven s otevřeným zdrojovým kódem. Je tedy zdarma k použití a i díky tomuto je tato knihovna jedna z nejpoužívanějších pro oživení dat na webových stránkách.



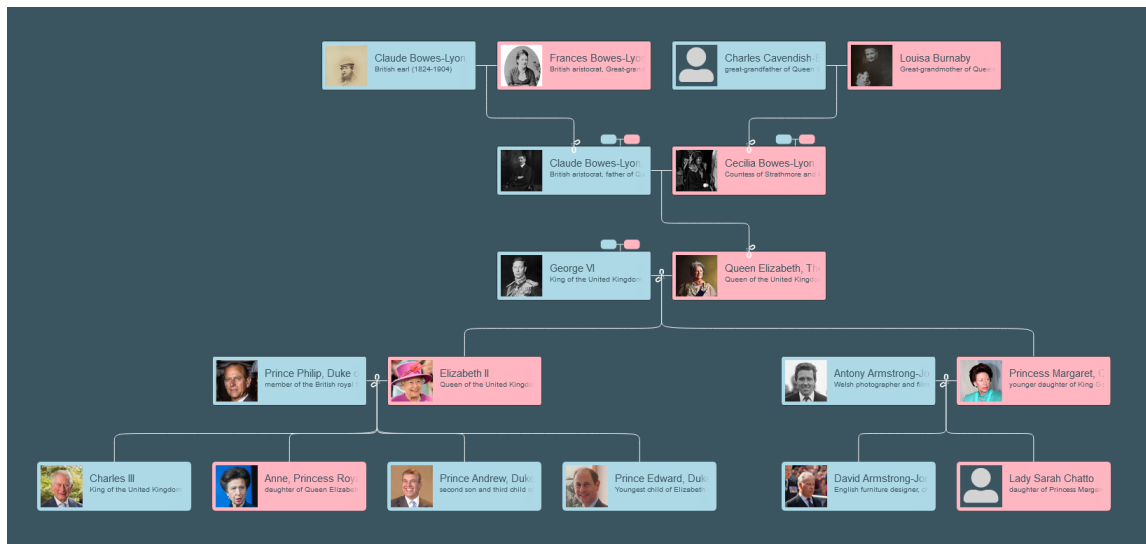
Obrázek 3.6: Ukázka grafu se stromovou strukturou vykreslený pomocí knihovny D3.js převzato z ukázkových grafů z oficiální stránky knihovny D3.js [2].

## Knihovna Family-chart

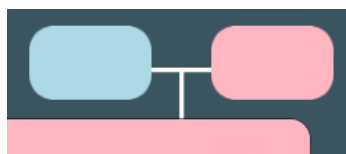
Autorem této knihovny je Donat Soric. Knihovna je pod licencí MIT a patří tedy mezi knihovny s otevřeným zdrojovým kódem. Mezi účely této knihovny patří vytváření a vykreslování rodokmenů. Výsledný rodokmen se dá nastylovat pomocí kaskádových stylů a také je možné při inicializaci určit data, která mají být zobrazena u každé osoby, jak velké mají být bloky s jednotlivými osobami, jak velké mají být mezery mezi těmito bloky nebo jestli může uživatel sbalit jednotlivé větve. Jako obrovská výhoda oproti dříve zmiňovaným knihovnám se jeví zobrazení všech předků i potomků probanta včetně předků nebo potomků přes mužskou i ženskou linii. Ukázka zobrazení rodokmenu pomocí této knihovny na obrázku 3.7.

Knihovna takřka vyhovuje funkčním požadavkům stanoveným pro tuto bakalářskou práci. Mezi požadavky, které zobrazení pomocí této knihovny nesplňuje, patří zobrazení manželek či manželů pod jejich partnery nikoliv vedle těchto osob. Dále má knihovna problém se zobrazením osob, které jsou v rodokmenu více než jednou. Tento jev může nastat například, když si osoba vezme svého bratrance či stříně (bratrance z druhého kolene) nebo pokud si muž vezme ženu a poté se rozvedou a on si vezme její sestru a nebo jiné případy, které život přináší. Dále je trochu matoucí, jak knihovna zobrazuje u osob, ke kterým má data o jejich rodokmenu, že se dá zobrazit jejich rodokmen, a to ve formě dvou malých objektů nad osobami. Ukázka těchto dvou objektů na obrázku 3.8. Dále není poznat, která ze zobrazených osob je probant. Probant tedy není nijak odlišen od ostatních osob v rodokmenu.

Všechny tyto nedostatky, které tato knihovna má se, ale dají odladit a tak byla vybrána jako nejvhodnější. Použití této knihovny, odstranění jejích nedostatků a zakomponování výsledné implementované webové stránky popsáno níže v kapitole 4.



Obrázek 3.7: Ukázka zobrazení rodokmenu Britské královské rodiny pomocí knihovny Family-chart [14].

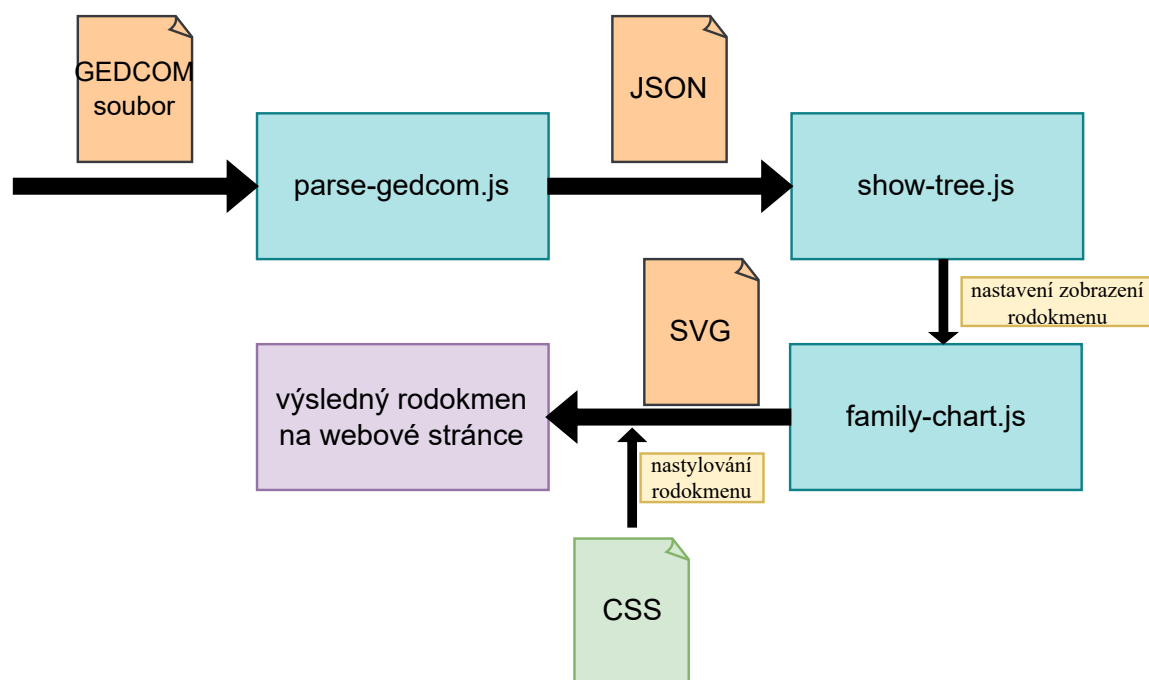


Obrázek 3.8: Ukázka, jak knihovna Family-chart ukazuje možnost zobrazit rodokmen dané osoby, když existují osoby, které nepatří k již vykreslenému rodokmenu, ale k rodokmenu dané osoby, u které jsou tyto objekty.

## Kapitola 4

# Implementace a výsledné zobrazení rodokmenů

Pro implementaci vykreslení rodokmenu se nejdříve musejí načíst a zpracovat vstupní data. Poté je ve správné struktuře předat knihovně pro vykreslení. Jelikož knihovna nesplňuje některé požadavky na výsledné vykreslení je zapotřebí tyto nedostatky opravit. Popis jednotlivých částí implementace níže v jednotlivých sekcích.



Obrázek 4.1: Grafické znázornění postupu mezi jednotlivými skripty od zpracování vstupních dat až po vykreslení výsledného rodokmenu.

### 4.1 Analýza a zpracování vstupních dat

V této sekci se rozebere, jak probíhala implementace zpracování vstupních dat. Vstupní data do softwaru mohou přijít ve dvou formátech. První možností je nahrání vstupních dat

v souborovém formátu GEDCOM, jehož definice a struktura je popsána v sekci 2.2. Jako druhá možnost je nahrání dat z databáze typu Neo4j popsané v sekci 2.3.

### Načtení a zpracování vstupních dat z grafové databáze Neo4j

Pro získání dat o osobě je nejprve důležité sestavit správný dotaz. Výsledkem dotazu by měla být data o osobě, vztahy dané osoby, předci, potomci a data o všech osobách patřících do rodokmenu dané osoby. Výsledný dotaz, jehož výsledkem jsou data o probantovi, jeho předcích a vztahy mezi nimi je sestavený v jazyce Cypher pomocí příkazu „MATCH“. Pomocí tohoto příkazu se najde osoba podle jejího identifikátoru a přes vztahy `JE_OTEC` anebo `JE_MATKA` se dohledají předci dané osoby a informace o každém z nich. Aby dotaz vrátil i rodiče rodičů a dále, tak se nastaví takzvaný „HOP“, který udává přes kolik osob může být výsledná odpověď, a to jak minimální množství, tak i maximální, které je v tomto případě nastaveno na nekonečno. Dotaz:

MATCH

```
p=(probant {id: $props.id})<-[re:JE_OTEC|JE_MATKA*1..]-(person: Osoba)
return p
```

Co se týče dotazu pro získání dat a vztahů o potomcích probanta se výše zmiňovanému dotazu pouze přehodí směr vztahů. Šipka vedoucí ze závorky s označením `person` do závorky s označením `probant` se otočí aby vedla naopak. Výsledný dotaz zajistí data o potomcích probanta, ale pouze děti dětí a jejich děti atd., ale už nezajistí data o manželství jednotlivých potomků a probanta. Pro získání těchto dat přibyl další dotaz, který vrací potomky a jejich druhého rodiče, který nebyl získán předešlým dotazem. Výsledný dotaz:

MATCH

```
p=(probant:Osoba)-[re:JE_OTEC|JE_MATKA*1..]->
(child:Osoba)
<-[re2:JE_OTEC|JE_MATKA]-(parents: Osoba) where probant.id = $props.id
return child,parents
```

Po sestavení dotazů, které získají potřebná data se jako další krok objevuje zaslání těchto dotazů na databázový server, který dotazy zpracuje a výsledek vrátí zpět programu. Na tento krok se použije HTTP rozhraní databáze Neo4j popisované výše v sekci 2.3. Tyto popisované funkcionality jsou implementované ve skriptu `load-file.js` ve funkci `loadNeo4jData()`. Tato funkce pošle HTTP požadavek, který obsahuje dotazy na data a čeká na odpověď. V momentě kdy přijde dojde k zavolání funkce `parseNeo4j` ze skriptu `parse-neo4j.js`, které se jako parametr předá odpověď a identifikátor probanta.

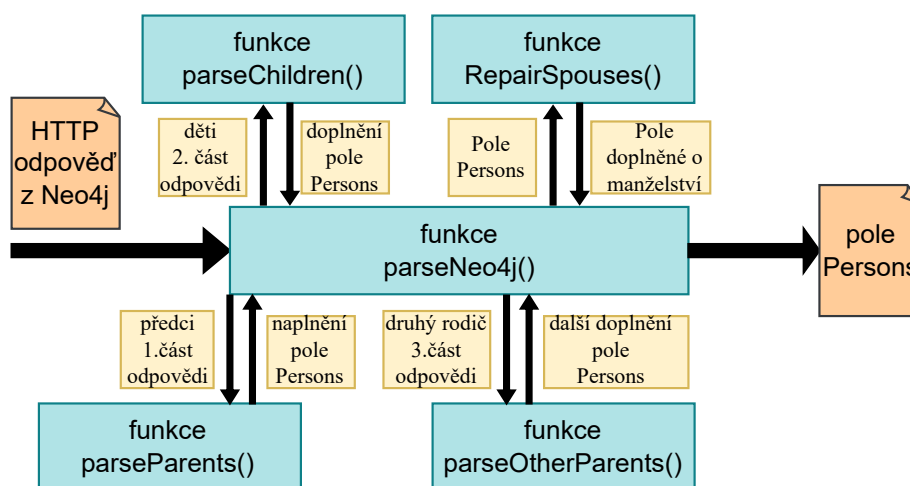
Funkce si rozdělí odpověď na tři, což jsou dílčí odpovědi na tři předešlé dotazy. Odpověď na první dotaz se zpracuje ve funkci `parseParents(parents)`, která projde celou odpověď na dotaz pro získání předků probanta. U každé osoby analyzuje informace a podle pořadí osob v odpovědi přidělí i vztahy daných osob mezi sebou. Konkrétně kdo je otec, matka a naopak kdo je jejich dítě. Pro zpracování informací pro každou novou osobu v odpovědi dojde k zavolání funkce `setPerson(persondata)`, která projde vlastnosti dané osoby a přeformuluje je na strukturu určenou pro knihovnu Family-chart. Po doplnění vztahů a vlastností ke každé osobě se doplní tato struktura osoby do pole `Persons`.

Druhá část odpovědi obsahující data o dětech probanta a dětí jejich dětí se zpracuje ve funkci `parseChildren(children)`. Tato funkce projde danou odpověď a stejně jako funkce pro zpracování předků, doplní vlastnosti a vztahy daných osob. Nové vlastnosti a vztahy

aktualizuje v konkrétních osobách v poli `Persons`, do kterého případně i doplní nové osoby, které zatím v poli chybí.

Nakonec proběhne zpracování třetí části odpovědi, která obsahuje druhé rodiče od předků probanta, které chybí v předchozí části odpovědi. Tato část se zpracuje ve funkci `ParseOtherParents(OtherPar)`. Nové osoby, resp. druhé rodiče, funkce zpracuje podobně jako předchozí funkce na zpracování předků a dětí. Načte jejich vlastnosti a doplní jejich vztahy podle pozice v této části odpovědi. Mezi vztahy patří otec, matka a dítě. Jelikož ve schématu Neo4j databáze chybí vztah manžel a manželka, tak musí dojít pouze k doplnění tohoto vztahu přes jejich děti. Doplnění tohoto vztahu se provede ve funkci `RepairSpouses()`, která projde všechny osoby v poli `Persons`. U každé osoby, která má oba rodiče dojde k doplnění vztahu mezi těmito rodiči.

Po zpracování všech odpovědí a doplnění manželství vrátí funkce `parseNeo4j()` pole `Persons` obsahující struktury osob, které uchovávají vlastnosti a vztahy. Grafické znázornění analýzy a zpracování dat z Neo4j na obrázku 4.2.

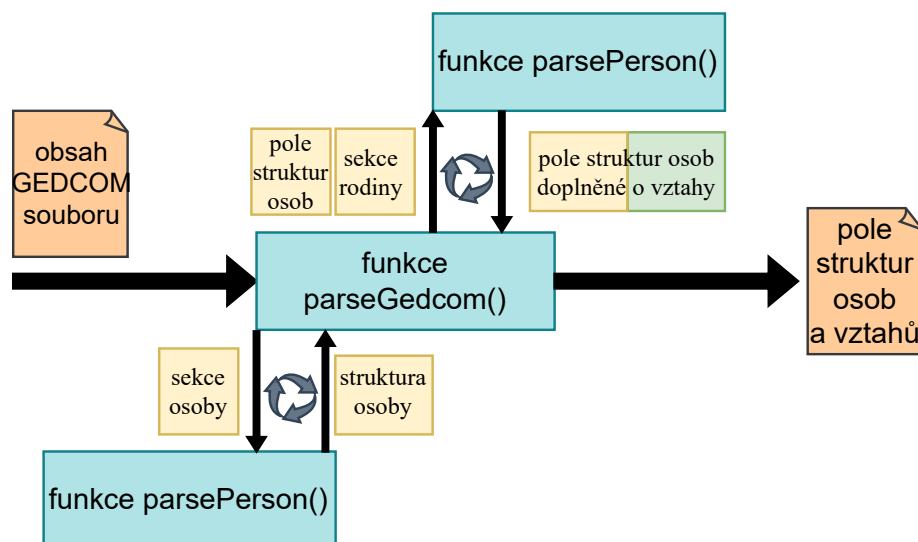


Obrázek 4.2: Grafické znázornění posloupnosti volání a zpracování jednotlivých hlavních funkcí ve skriptu `parse-neo4j.js` včetně zobrazení předávání informací přes pole osob.

## Zpracování vstupních dat ze souborového formátu GEDCOM

Co se týče analýzy a zpracování dat z tohoto souborového formátu, tak je celá tato část implementace ve skriptu s názvem `parse-gedcom.js`. Grafické zobrazení průběhu a volání mezi funkcemi tohoto skriptu na obrázku 4.3. Jak už přípona tohoto souboru napovídá, je napsán v programovacím jazyce Javascript.

Při načtení tohoto souboru do webové stránky dojde pouze k načtení jeho funkcí. Základem je funkce s názvem `parseGedcom(gedcom={})`, které se jako parametr předá obsah ze souboru ve formátu GEDCOM. Tato funkce rozdělí obsah podle řádků, které pozná podle koncového znaku. Jelikož koncové znaky nejsou vždy stejné, ale existuje více možností označení konce řádku. Mezi první je styl označení podle operačního systému Windows, který je podle speciálních znaků `\r\n`. Druhý je styl pro operační systémy nad Unixem, který má pouze jeden speciální znak na konci řádku a to `\n` a nakonec třetí způsob podle operačního systému určeného pro zařízení značky Apple a to pouze jeden speciální znak `\r`. Na rozeznání, který způsob je zrovna použit v obsahu souboru ve formátu GEDCOM, slouží funkce



Obrázek 4.3: Grafické znázornění posloupnosti volání a zpracování jednotlivých hlavních funkcí ve skriptu parse-gedcom.js včetně zobrazení předávání jednotlivých parametrů.

`GetLinesEnd(source)`, která zkusí najít pozici znaku `\n`, pokud jej nenajde, tak vrátí jako označení konce řádků znak `\r`. Pokud jej najde, tak ověří, jestli před tímto znakem leží znak `\r` a jestli ano, tak vrátí jako označení konce řádků spojení znaků `\r\n`. Ovšem zdali znak `\r` se nenachází před znakem `\n`, tak vrátí jen samotný znak `\n`, který označuje konce řádků v daném obsahu souboru ve formátu GEDCOM.

Rozdělené řádky se projdou jeden po druhém v cyklu. Každý řádek se ještě rozdělí na hodnoty podle mezer mezi nimi. Pokud řádek začíná hodnotou „0“, tak se každý následující řádek přidá do sekce a končí v momentě, kdy se najde další řádek s hodnotou „0“. Jakmile toto nastane dojde ke zpracování celé sekce. Pokud jde o sekci s informacemi o osobě tedy sekce, která měla jako první řádek s první hodnotou „0“ a třetí hodnotou `INDI`, tak dojde k zavolání funkce `ParsePerson(section)`, kde se jako parametr předá celá daná sekce.

V této funkci dojde k vyčtení základních informací o osobě, které jsou obsaženy v jednotlivých řádcích v celé sekci. Mezi informace patří identifikátor dané osoby, který se nachází na prvním řádku na druhé hodnotě mezi hodnotou „0“ a třetí hodnotou, což je označení typu dané sekce, v tomto případě `INDI`. Tyto informace jsou uloženy do struktury a tato struktura se vrací do funkce, odkud byla volána funkce `ParsePerson(section)`. Zpátky ve funkci `GetLinesEnd(source)` dojde k doplnění struktury s informacemi o dané osobě do pole osob. Tento postup je stejný pro všechny osoby uložené v obsahu souboru formátu GEDCOM. Když cyklus projde a vloží všechny osoby do pole, tak se začnou zpracovávat sekce s daty o jednotlivých rodinách.

Na konci každé sekce dojde k volání funkce `parseFamily(section, persons)`, které se jako parametr předá daná sekce s informacemi o rodině a také pole s informacemi o všech osobách, které se zpracovaly dříve. V této funkci následně dojde ke zpracování informací o rodině z dané sekce. Dojde k uložení informace o tom, kdo je v této rodině uložen jako manžel a otec, kdo jako manželka a matka a kdo jako dítě. Po projití celé sekce s daty o jedné rodině se informace propíší do struktury jednotlivých osob. Vezme se identifikátor na dané osoby a ten se zapíše do vztahů k daným osobám. Respektive u dětí se do vztahů zapíše, kdo je jejich otec a matka a u rodičů se napíše, kdo je jejich manželem či manželkou



a také kdo jsou jejich děti. Všechny tyto informace se doplní k jednotlivým osobám v poli a poté, co cyklus dokončí zpracování poslední sekce s informacemi o rodině, tak se výsledné pole s osobami, které mají data se správnou strukturou, vrátí do místa odkud byla volána funkce `parseGedcom(gedcom={})`.

## 4.2 Základní nastavení zobrazení rodokmenu

Implementace základního nastavení zobrazení se nachází ve funkci `store(data)` v souboru `show-tree.js` (také v programovacím jazyce Javascript), které se jako parametr předá pole struktur osob s odkazy na osoby, ke kterým mají jednotlivé osoby nějaký vztah. Struktura tohoto pole je pevně daná a dá se popsat pomocí struktury typu JSON popsaného v sekci 2.2. Konkrétní struktura jednotlivých osob v poli by měla vypadat následovně:

```
{
  id : "identifikátor osoby",
  data : {
    birthday : "datum narození",
    birthplace : "místo narození",
    "first name" : "křestní jméno",
    gender : "pohlaví (M nebo F)",
    "last name" : "příjmení",
    deathday : "datum úmrtí",
    deathplace : "místo úmrtí"
  },
  rels : {
    father : "identifikátor otce",
    mother : "identifikátor matky",
    spouses : [
      "identifikátor manželky/manžela",
      "identifikátor další manželky/manžela"
    ],
    children : [
      "identifikátor dítěte",
      "identifikátor dalšího dítěte"
    ]
  }
}
```

Každá osoba musí mít jednoznačný identifikátor zapsaný ve struktuře. Struktura dále obsahuje dvě menší struktury. Jednou z nich jsou „data“, kde jsou uloženy informace o dané osobě. A druhou strukturou je „rels“ což představuje vztahy k jiným osobám. Tyto jednotlivé informace jsou podle daných osob vyplněné anebo mohou úplně chybět. Osoby, u kterých není známo např. datum narození, místo narození, datum úmrtí, místo úmrtí anebo třeba pohlaví, tak se předají a vykreslí podle jejich vztahu k jiným osobám. Takováto struktura jedné osoby je pak spolu s ostatními strukturami s daty o jiných osobách v poli, které se pozná pomocí znaků [ a ].

Mezi další možnosti základního nastavení patří:

**Vzdálenosti** Vzdálenost mezi osobou a jeho manželky anebo manžely určuje proměnná `spouse_separation`. Dále vzdálenost mezi jednotlivými generacemi (otec jedna generace a jeho synové a dcery jako druhá generace) se dá nastavit pomocí proměnné `level_separation`. Co se týče vzdálenosti mezi jednotlivými manželkami či manžely a vzdálenosti mezi jednotlivými dětmi (potomky), tak tato se určuje pomocí proměnné `node_separation`. Zmíněné vzdálenosti jsou pevně nastavené a zaručují tak přehlednost spolu se správnou vizualizací a vyvarují se tedy zbytečnému zvětšování jednotlivých rodokmenů.

**Rotace rodokmenu** Rodokmen lze vykreslit horizontálně anebo vertikálně. Pro tuto bakalářskou práci byl zvolen způsob vykreslení rodokmenu vertikálně. Tedy předky směrem nahoru a potomky směrem dolů od probanta.

**Velikost kartiček** Jednotlivé osoby mají každá svou kartičku, na které se nachází informace o ní. Všem těmto kartičkám se dá nastavit šířka a výška. Vše dostatečně velké, aby se na kartičkách objevily veškeré informace.

**Pozice textu** Na každé výše zmíněné kartičce jsou tedy informace o jednotlivých osobách a v základních nastaveních lze určit, na jakých souřadnicích na kartičce mají být jednotlivé informace. Finální nastavení je, že se na prvním řádku větším písmem zobrazí jméno a příjmení dané osoby. Druhý řádek obsahuje datum narození a případně i datum úmrtí. Na třetím řádku se zobrazuje místo narození a případně i místo úmrtí.

**Zobrazení sponky** Dále lze nastavit, jestli se má zobrazovat sponka u jednotlivých větví rodokmenu. Takováto sponka je ve výchozím stavu rozepnutá a jsou tak všechny jednotlivé větve rodokmenu zobrazené a vykreslené. Pokud ale na nějakou sponku klikneme, dojde k jejímu sepnutí a tím pádem se celá větev, která patří k dané sponce, schová, a nebude vykreslená, dokud na sponku znovu neklikneme a nerozepneme ji.

Po sestavení těchto základních nastavení vykreslování rodokmenu se předá toto nastavení upravené knihovně, která je v souboru `family-chart.js`. Tato knihovna vykreslí finální rodokmen na webovou stránku.

## 4.3 Změny knihovny Family-chart a vykreslení rodokmenu

Tato sekce popisuje změny knihovny Family-chart popisovanou výše v sekci 3.2. Každý jednotlivý nedostatek knihovny je nejprve dostatečně popsán a následně jsou popsány změny v kódu, které vedly k opravě těchto nedostatků.

### Změna pozice manželství

Jak bylo popsáno v sekci 3.2, knihovna Family-chart vykresluje manželky anebo manžele doprava od jednotlivých osob. Cílem této práce ale je, aby jednotlivá manželství byla hned pod danými osobami, ke kterým patří. Po provedení analýzy jednotlivých funkcí této knihovny jsem došel k závěru, že je potřeba předělat implementaci funkce:

```
setupSpouses({tree, spouse_separation, node_separation})
```

Která sestavuje souřadnice manželkám či manželům probanta, ale také manželky a manžele jeho potomků. Pokaždé projde všechna manželství dané osoby. Manželkám či manželům nastaví stejnou výšku v rodokmenu pomocí „y souřadnice“. Pozice na šířku určená pomocí „x souřadnice“ se počítá jako pozice dané osoby, od které se odečte násobek předem určené mezery mezi osobami a čísla, které představuje, kolikátá manželka nebo kolikátý manžel dané osoby se určuje.

```
spouse.x = d.x-(node_separation*(i+1))*side;
```

Tento způsob je tedy původní a vykresluje manželství do šířky místo do hloubky. Pro vykreslení do hloubky bylo zapotřebí počítat jinak obě souřadnice. V případě „x souřadnice“ je tedy nový výpočet, kdy u prvního manželství se vezme z osoby, pro kterou se manželství sestavují a všechna další manželství se rozdělí podle čísla, které určuje kolikáté manželství ze všech se zrovna vypočítává. Pokud je číslo sudé, dojde k posunu směrem doprava a u lichého čísla doleva. Zároveň se tento posun násobí tímto číslem, čímž se zajistí větší posun u dalších a dalších manželství. Následující kód ukazuje implementaci tohoto výpočtu souřadnic jednotlivých manželek či manželů. Objekt `spouse` představuje danou manželku či manžela. Objekt `d` představuje osobu, pro kterou dochází k sestavování manželství. Proměnná `i` určuje číslo, které říká, kolikáté manželství se zrovna prochází v cyklu. Proměnná `node_separation` a `spouse_separation` jsou mezery mezi jednotlivými osobami v rodokmenu, které se určovaly v základním nastavení popsaném v sekci 4.2. Co se týče výpočtu „y souřadnice“ daných manželství, tak ty se u každého manželství vypočítají stejně, a to tak, že od pozice osoby (pro kterou sestavujeme manželství) odečteme mezeru určenou v základním nastavení pro určení, jak velký má být posun manželek či manželů od osob, pro které se manželství sestavují, tedy pro probanta a jeho potomky.

```
if(i==0){
    spouse.x = d.x;
}
else {
    i%2==0 ? spouse.x =
d.x-((node_separation)*(-(i-1)))
:
spouse.x = d.x-((node_separation)*(i));
}
spouse.y = d.y-(spouse_separation)*side;
```

Touto změnou dojde ke správnému vykreslení daných manželství přímo pod dané osoby. Na druhou stranu ale nedojde ke správnému vykreslení jednotlivých vztahů pomocí linek. Tyto linky dále směřují na staré místo, které bylo mezi manželem a manželkou a pokračovalo dále na jejich společné děti. V implementaci tedy muselo dojít ke změně daných linek, aby se vykreslovaly až pod danou manželku či manžela se kterou má daná osoba dítě či děti. Určení souřadnic těchto linek se dělá ve funkci `handleProgenySide({d})`. V této funkci dojde ke změně „y souřadnic“ u linek vedoucích od manželek či manželů probanta nebo jeho potomků směrem k jejich dětem. Začátek linky bude zespodu uprostřed kartičky manželky či manžela směrem k jednotlivým dětem.

Mezi další funkce, které byly potřeba odladit patří také `handleAncestrySide({d})`. Tato funkce sestavuje linky vedoucí k předkům probanta, a to přes mužskou i ženskou linii. Pro výpočet „y souřadnice“ se zde použije „y souřadnice“ rodičů probanta, případně

rodičů jeho předků, a to od rodiče vykresleného níže oproti tomu druhému rodiči. Jelikož je výpočet linky potřeba jiný než v případě sestavování potomků, tak byla doplněna funkce `LinkParents(d, p)`, která počítá průběžnou cestu linky pomocí odečtení výšky kartičky nastavené v základním nastavení a „y souřadnice“ osoby.

V případě, že se v základním nastavení nastaví použití takzvaných sponek, tak po předešlých změnách dojde ke špatnému vykreslení těchto sponek. Respektive sponky jsou v rodokmenu jinde, než kudy vedou jednotlivé linky, ke kterým sponky patří. Pro odladění tohoto problému se po analýze označila funkce `LinkBreakIconWrapper({d, card_dim})`, která vykresluje dané sponky k jednotlivým větvím. Sponka má více možností, kam se u dané osoby má vykreslit. Tyto možnosti jsou podle vztahů daných osob. Jestli jde o předky probanta anebo potomky probanta. Došlo tedy k úpravě souřadnic sponek podle daných možností a podle předešlých úprav, aby byly na správném místě.

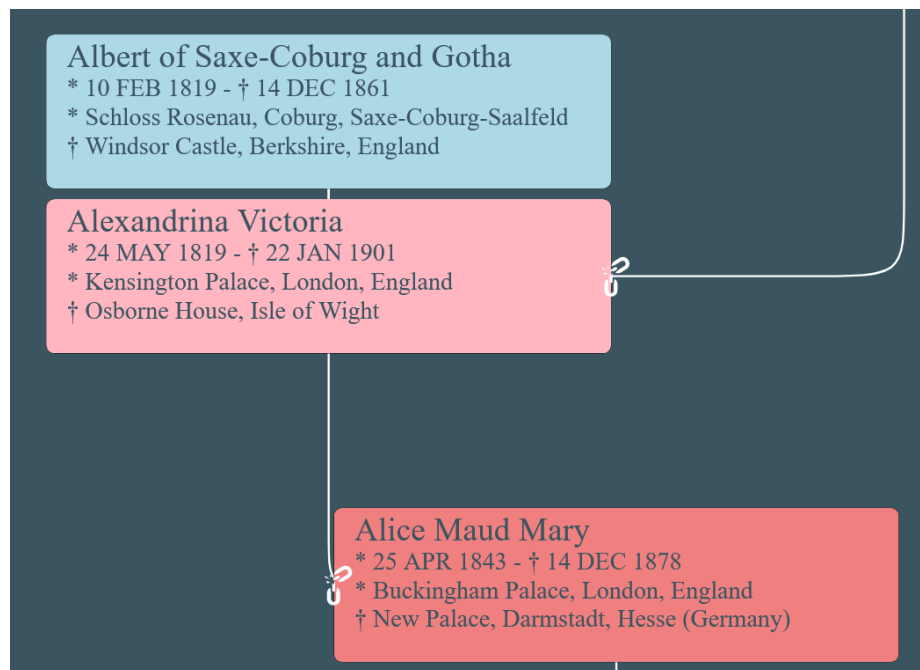
## Zvýraznění probanta

Pokud vykreslíme rodokmen pomocí knihovny Family-chart, tak dojde k rozdělení kartiček jednotlivých osob barevně pouze na tři druhy, a to růžová pro ženské pohlaví, modrá pro mužské a šedá pro osoby bez určeného pohlaví. Tímto způsobem může být matoucí, kde v rodokmenu je vykreslený probant. Abychom vykreslení barevně odlišilo probanta od jeho předků či potomků, tak musí probantova kartička mít jinou třídu kaskádových stylů. Tato třída se všem jednotlivým kartičkám osob určuje ve funkci `CardBody({d,w,h})`. Do této funkce tedy přibyly další možnosti tříd pro kartičky. Pokud kartička, které se zrovna určuje třída, je kartička probanta, tak podle pohlaví dojde k určení třídy z těchto možností: `card-male-main`, `card-female-main` a nebo `card-genderless-main`. Pokud se jedná o probantova předka, potomka anebo zkrátka jinou osobu z rodokmenu, dojde k určení třídy podle pohlaví a výběr je z těchto možností: `card-male`, `card-female` a nebo `card-genderless`.

Ukázka kódu z funkce `CardBody({d,w,h})` určujícího třídu podle vlastností dané osoby, která se uloží do konstanty `color_class`, a ta se následně vloží přímo do SVG elementu kartičky.

```
const color_class = !d.data.main ?(d.data.data.gender === 'M' ?
'card-male' : d.data.data.gender === 'F' ?
'card-female' : 'card-genderless')
:
(d.data.data.gender === 'M' ?
'card-male-main' : d.data.data.gender === 'F' ?
'card-female-main' : 'card-genderless-main');
```

Pro správnou funkčnost se do kaskádových stylů nastavených v souboru family-chart.css museli doplnit třídy určené probantovi rodokmenu. V případě, že je probant žena vykreslí se jeho kartička s tmavějším růžovým pozadím, které se dá rozeznat od růžové, která se vykresluje na pozadí kartiček ostatních žen v rodokmenu. Znázornění probanta ženského pohlaví v jiné barvě oproti jeho předkům na obrázku 4.4. Na pozadí mužského probanta se vykreslí zelená barva, která se oproti světle modré barvě na pozadí ostatních mužů v rodokmenu dost liší.



Obrázek 4.4: Grafické znázornění barevného rozdílu pozadí kartiček probanta ženského pohlaví oproti kartičkám jeho předků.

### Schování možného rozšíření rodokmenu

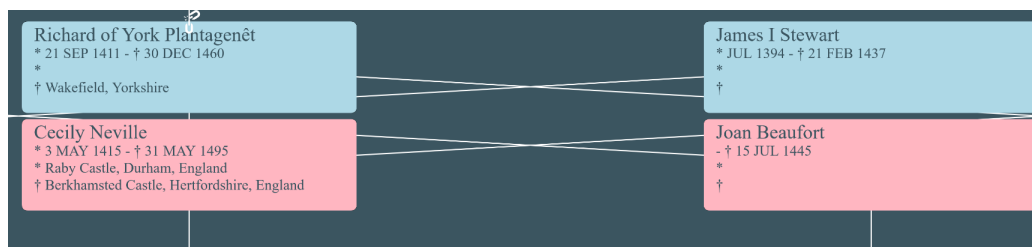
Jak bylo vysvětleno v sekci 3.2, tak knihovna Family-chart vykresluje možnost zobrazení rodokmenu jiných osob než probanta, když k nim má data. Toto vykreslení je vyobrazeno na obrázku 3.8. Takovéto vykreslení je zbytečné, a proto bylo potřeba jej nedělat anebo jej schovat. Po drobné analýze jsem se rozhodl vykreslení schovat pomocí kaskádových stylů. Bylo to nejlehčí řešení, které by vedlo sice k vykreslení tohoto elementu, ale ve výsledném zobrazení by nebyl vidět a nebylo by na něj možné ani kliknout. Takže se v souboru `family-chart.css` přidalo ke stylům tohoto elementu viditelnost na schováno. Výsledný kaskádový styl elementu:

```
.card_family_tree{
  visibility: hidden;
}
```

### Vykreslení duplicitních osob a vztahů

Ve vykreslení rodokmenu určitých rodin může dojít k situaci, že se jedna osoba vykreslí vícekrát. Samotné duplicitní vykreslení pro splnění cílů bakalářské práce nevadí. Bohužel knihovna Family-chart s tím ale neumí správně pracovat. Osobu vykreslí kolikrát je potřeba a dokonce na správných souřadnicích, ale v momentě, kdy spojuje jednotlivé osoby podle jejich vztahů, tak dojde ke špatnému spojení mezi osobami. S tímto špatným vykreslením dojde k tomu, že jsou linky jednotlivých vztahů přes celý rodokmen různě poházené a křížují se klidně přes spousty osob a dalších vztahových linek. Takovéto duplicitní vykreslení osob nastane například v případě, že si jedna osoba z rodu vezme svého bratrance či sestřenici a tím pádem, když se mapují jejich předci, dojde ke zdvojení těchto větví, to stejné i u

potomků v případě vykreslení rodokmenu, kde bude probant jeden předků z rozdvojených větví. Ukázka špatného spojení duplicitních osob na obrázku 4.5.



Obrázek 4.5: Grafická ukázka části vykresleného rodokmenu v případě, že jsou v rodokmenu duplicitní osoby špatně spojené s jejich manželky či manžely.

Aby došlo ke správnému vykreslení linek vztahů mezi osobami, musí dojít k úpravě funkce `handleSpouse({d})`. V této funkci se mapují manželství daných osob přes prohledávání pole se všemi osobami. Konkrétně pomocí tohoto řádku:

```
const spouse = tree.find(d0 => d0.data.id === sp_id);
```

Funkce najde první manželku či manžela dané osoby podle jeho identifikátoru. Jenže tento identifikátor je stejný i pro duplicitní osobu, respektive se tato osoba nachází v poli osob vícekrát a hledání pouze podle identifikátoru je nedostatečné. Pro nalezení správné osoby na správných souřadnicích došlo k rozšíření tohoto hledání ještě přes „x souřadnice“. Jelikož se při zpracování manželství ke každé osobě ukládá i „x souřadnice“, na které se nachází jeho manželka či manžel, tak se vyhledávání rozšíří i na podmínku zdali sedí tyto „x souřadnice“. Ukládání této souřadnice se provádí ve funkci:

```
setupSpouses({tree, spouse_separation, node_separation})
```

Finální úprava tohoto řádku:

```
const spouse =
tree.find(d0 => (d0.data.id === sp_id)&&(d0.spouse!==undefined ?
(d0.spouse.data.id==d.data.id&&d0.spouse)
:true)&&(d0.sx==d.x | d0.x==d.x));
```

Ukázka vykreslení duplicitních osob a správného vykreslení linek jejich vztahů na obrázku 4.6.



Obrázek 4.6: Grafické znázornění vykreslení duplicitních osob v rodokmenu se správným napojením linek jejich manželek a manželů.

Problém s vykreslováním duplicitních osob se projeví i na vykreslování linek vztahů u dětí těchto osob, a to v případě duplicit v potomcích probanta. Linky od těchto dětí vedou

přes celý rodokmen zleva doprava anebo zprava doleva a mohou křížovat spoustu osob a jiných linek. Funkce, která mapuje děti a jejich rodiče nese název `handleProgenySide` a jako parametr se jí předává struktura dané osoby (`{d}`), pro kterou se sestavují její potomci. V této funkci se projdou všechny děti dané osoby a nastaví se linky vedoucí od těchto dětí k jejich rodičům. Souřadnice pro linky se načítají přes druhého rodiče, než který se předal jako parametr. Tento druhý rodič každého jednoho dítěte se hledá pomocí funkce `otherParent(child, d, tree)`, které se jako parametry předávají dané dítě, daná osoba a pole se všemi osobami rodokmenu. Funkce vrací jiného rodiče, než který se předá jako parametr. Tady nastává problém, protože nepočítá s možností duplicitních rodičů a vrací prvního nalezeného, který splňuje podmínky, že jeho identifikátor není stejný jako rodiče předaného parametrem a také, že je rodičem dítěte, které se také předalo parametrem.

Pro odlišení těchto duplicit je jako jednou z možností sestavit pole potencionálních jiných rodičů a vybrat z nich jednoho, který je podle „x souřadnice“ nejbližší k danému dítěti. Po nalezení takovéto osoby se vrací zpátky do funkce `handleProgenySide`, která vykreslí linky vztahů těchto dětí ke správným rodičům. Jediná změna tedy byla potřeba provést ve funkci `otherParent(child, d, tree)`. Výsledná úprava:

```
var arr = data.filter(
d0 => (d0.data.id !== p1.data.id)
&&
(
(d0.data.id === d.data.rels.mother)
||
(d0.data.id === d.data.rels.father))
);

var old_x=d.psx!==undefined ? d.psx : d.x;

if(arr.length>1){
  var old_xc,new_x;
  old_x=undefined;
  for(i=0;i<arr.length;i++){
    new_x = arr[i].x;
    var vypocet = d.x-new_x;
    if(vypocet<0)vypocet*=-1;
    if(old_xc===undefined)
    {
      old_xc=vypocet;
      old_x=new_x;
    }
    if(vypocet<old_xc){
      old_xc=vypocet;
      old_x=new_x;
    }
  }
}
else if(arr.length==1){
```

```

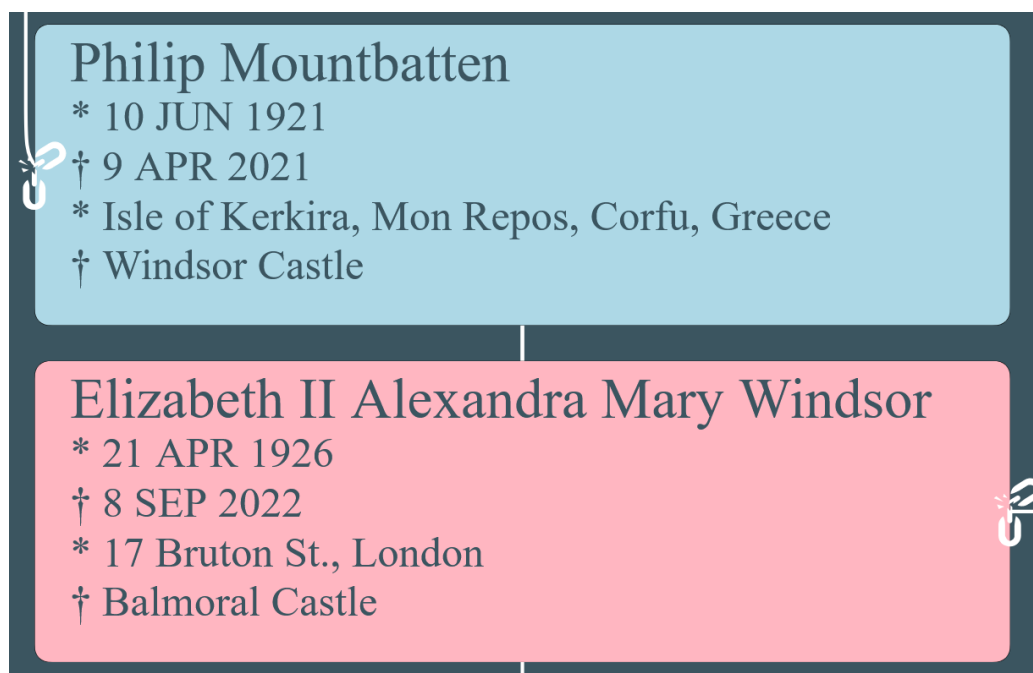
        old_x = arr[0].x;
    }

    return data.find(
    d0 => (d0.data.id !== p1.data.id)
    &&
    (
    (d0.data.id === d.data.rels.mother)
    ||
    (d0.data.id === d.data.rels.father)
    )
    &&
    (d0.x==old_x));

```

## 4.4 Výsledné zobrazení rodokmenu

Po dokončení implementace a odstranění nedostatků knihovny Family-chart, dojde k vykreslení přehledného rodokmenu, který obsahuje osoby z rodu probanta včetně všech jeho předků a potomků. U každé osoby jsou zobrazeny informace o ní a u každé osoby jsou správně zobrazeny její vztahy k ostatním osobám z rodokmenu. Na obrázku 4.7 se nachází ukázka části rodokmenu s mužem a jeho manželkou a u obou je možné vidět základní informace o nich.

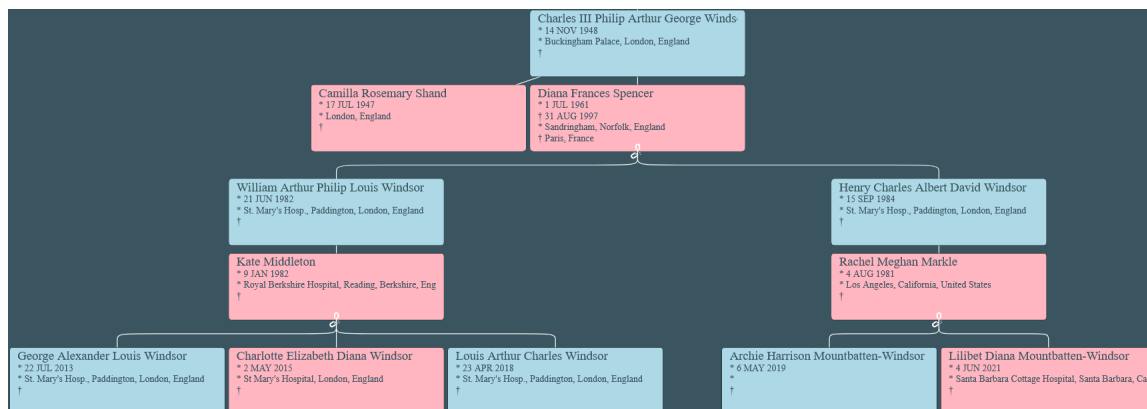


Obrázek 4.7: Grafická ukázka části výsledného vykreslení rodokmenu obsahující muže a jeho ženu.

Výsledné zobrazení obsahuje správné vykreslení jednotlivých vztahů mezi osobami ať už jsou jakkoliv složité. Pokud je nějaký člen rodokmenu rozvedený a má děti s různými

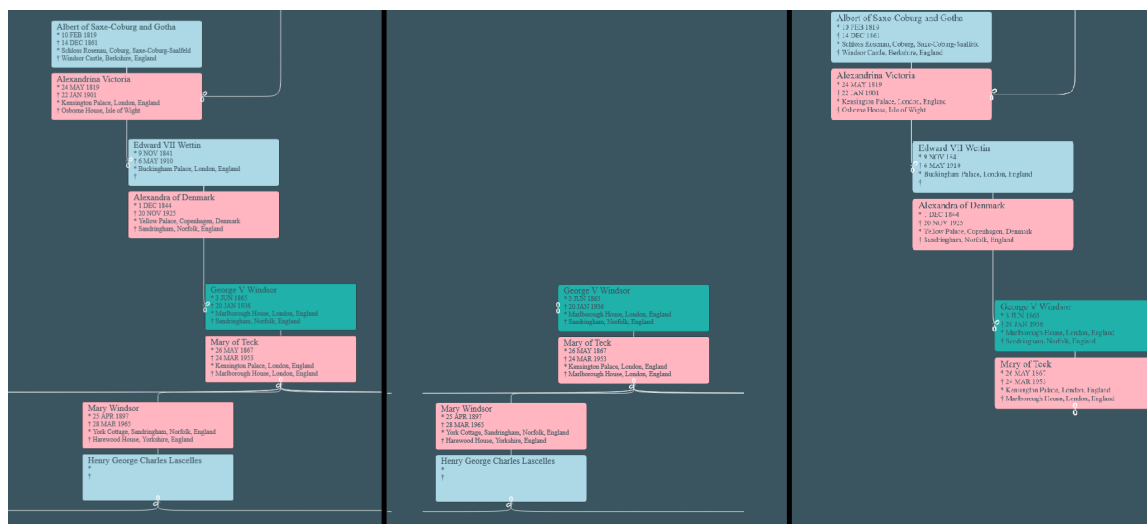


ženami je ve výsledném rodokmenu každá jednotlivá manželka pod daným mužem a jejich děti vedou ke správné matce. Ukázka vykreslení takové části rodokmenu na obrázku 4.8.



Obrázek 4.8: Grafická ukázka části výsledného vykreslení rodokmenu obsahující muže a jeho ženu.

Každá větev rodokmenu se dá sbalit a schovat, čímž zaručuje větší přehlednost mezi rodinami v rodokmenu. Sbalení se provádí kliknutím na sponku. Porovnání rozepnuté sponky pro předky i potomky a sepnuté sponky pro předky a sepnuté sponky pro potomky na obrázku 4.9. Vykreslení rodokmenů na webové stránce je tedy splněné a zároveň zvládá i složitější vztahy a větší rodokmeny.



Obrázek 4.9: Porovnání rozepnutých a sepnutých sponek u stejného probanta. Nejprve všechny sponky rozepnuté, poté sepnutá sponka pro předky probanta a naposled rozepnutá sponka předků, ale sepnutá sponka potomků.

## 4.5 Možnosti rozšíření

Jako jednu z možností rozšíření se dá s jistotou určit doplnění vztahu manžel či manželka do grafové databáze Neo4j. V této bakalářské práci se ve zpracování vstupních dat z této grafové databáze musí pouze předpovídat manžel a manželka podle vztahů matka a otec. Dále do vlastností jednotlivých osob doplnit informace o bydlišti případně i místa narození, úmrtí, atd.

Další možností rozšíření je možnost vyexportovat výsledné zobrazení jako vektorový obrázek, ať už v souborovém formátu PNG anebo SVG. Případně export i do souborového formátu PDF. Pro exportování se dá využít knihovna programovacího jazyka Javascript, ale je zapotřebí využít takovou, která umí přidat kaskádové styly do výsledného exportovaného obrázku. Tedy aby obrázek vypadal stejně jako rodokmen na webové stránce.

Také by nebylo špatné dát možnost základního nastavení zobrazení rodokmenu popisované v sekci 4.2 přímo uživateli na webové stránce. Tím by uživatel získal možnost nastavit velikost kartiček jednotlivých osob, mezery mezi jednotlivými osobami a také jaké informace mají být na kartičce vykresleny.

## Kapitola 5

# Závěr

Cílem této práce bylo načtení a zpracování dat o rodu z grafové databáze Neo4j nebo ze souboru Gedcom a následné vykreslení těchto dat jako rodokmen na webové stránce. Po průzkumu knihoven, které by mohli být vhodným nástrojem pro splnění tohoto cíle byla nejprve vybrána knihovna dTree, u které se nakonec ukázalo, že nezobrazuje všechny předky, tak jak by měla, a tak proběhlo druhé kolo průzkumu ve kterém zvítězila knihovna Family-chart, která už vedla ke správnému vykreslení rodokmenu včetně všech jeho předků a potomků. Tato knihovna měla také nějaké nedostatky, ale ty nakonec byly odladěny. Po výběru knihovny došlo implementaci načtení vstupních dat a předělání těchto dat do správné struktury, tak aby bylo možné tyto data správně předat knihovně Family-chart. Výsledkem je tedy dynamická webová stránka postavená na Javascriptu, HTML, CSS a SVG, která načte vstupní data z grafické databáze Neo4j anebo souboru Gedcom a tyto data předělá do struktury pro upravenou knihovnu Family-chart, která tyto data vykreslí jako rodokmen v SVG na webovou stránku. Výsledný rodokmen obsahuje všechny předky a všechny potomky osoby pro kterou se daný rodokmen vykresluje a také nabízí možnost schovat jednotlivé větve rodokmenu, ať už třeba všechny předky, všechny potomky anebo jen větve dětí jeho vnuků a tak dále. Mezi možnostmi rozšíření, které bych určitě zvažil patří export do souboru PDF. Výsledný export by měl dobrou kvalitu a dal se lépe sdílet a používat. A také v grafové databázi Neo4j doplnit vztah manželka a manžel, protože a jelikož tato bakalářská práce odvozuje tento vztah ze vztahu otec či matka.

# Literatura

- [1] BAUR, M. *Family Tree Demo* [online]. yworks, únor 2023. Dostupné z: <https://live.yworks.com/demos/layout/familytree/>.
- [2] BOSTOCK, M. *Data-Driven Documents* [online]. D3.js, 2021. Dostupné z: <https://d3js.org/>.
- [3] BÖHM, M. *Co je to rodokmen* [online]. 1999 [cit. 2023-04-26]. Dostupné z: <http://www.tady.cz/bohm/schemata.html>.
- [4] FEKETE, J.-D. *Geneaquilt* [online]. GitHub, 2014 [cit. 2023-04-26]. Dostupné z: <https://github.com/jdfekete/geneaquilt/blob/master/src/data/Simpsons.ged>.
- [5] FINDLAY, T. *Introducing JSON* [online]. Telerik, 30. prosince 2020 [cit. 2023-04-26]. Dostupné z: <https://www.telerik.com/blogs/what-is-json-how-to-handle-unexpected-token-error>.
- [6] HEIN, R. *How Many Users Have JavaScript Disabled* [online]. Search Engine People, 14. prosince 2010. Dostupné z: <https://www.searchenginepeople.com/blog/stats-no-javascript.html>.
- [7] JONES, T. *The GEDCOM 5.5.5 Specification with Annotations* [online]. GEDCOM, 2019 [cit. 2023-04-26]. Dostupné z: <https://www.gedcom.org/>.
- [8] KONOSHENKO, E. *Možnosti využití databáze Neo4j*. 2016 [cit. 2023-05-11]. Bakalářská práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních technologií. Dostupné z: [https://vskp.vse.cz/49576\\_moznosti-vyuziti-databaze-neo4j](https://vskp.vse.cz/49576_moznosti-vyuziti-databaze-neo4j).
- [9] KOĐOUSKOVÁ, B. *Web, webová stránka a webová aplikace, v čem je rozdíl?* [online]. Rascasone, 22. června 2021. Dostupné z: <https://www.rascasone.com/cs/blog/web-webova-aplikace-rozdil>.
- [10] MLČOCHOVÁ, K. *Genealogické prameny ve fondu okresních úřadů (s přihlédnutím k okresnímu úřadu Český Těšín)* [online]. Praha, 2022. [cit. 2023-04-25]. 95 s. Diplomová práce. Univerzita Karlova, Filozofická fakulta. Vedoucí práce DOC. PHDR. IVANA EBELOVÁ, C. Dostupné z: <https://dspace.cuni.cz/bitstream/handle/20.500.11956/176186/120430093.pdf?sequence=1>.
- [11] PECHÁČEK, J. *Genealogické diagramy II. - Odkud jsme: Genealogické diagramy aneb co je to vlastně rodokmen? Část II.* [online]. 2015 [cit. 2023-04-26]. Dostupné z: <https://www.odkudjsme.cz/blog/genealogicke-diagramy-aneb-co-je-to-vlastne-rodokmen/>.

- [12] PŘENOSIL, V. *Komparace výkonu Oracle Database a grafové databáze Neo4j při grafových úlohách [online]*. 2014 [cit. 2023-05-10]. Diplomová práce. Univerzita Pardubice, Fakulta elektrotechniky a informatiky Pardubice. SUPERVISOR: Ing. Tomáš Váňa. Dostupné z: <https://theses.cz/id/jks5kn/>.
- [13] ROUSE, M. *File Format [online]*. 2017 [cit. 2023-04-26]. Dostupné z: <https://www.techopedia.com/definition/952/file-format>.
- [14] SORIC, D. *Family-chart: donatso [online]*. GitHub, 2022 [cit. 2023-04-28]. Dostupné z: <https://github.com/donatso/family-chart/tree/master#readme>.
- [15] VLK, P. *Zobrazovanie genealogických dát [online]*. Brno, 2018. [cit. 2023-04-25]. 47 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce ING. JAROSLAV ROZMAN, P. Dostupné z: [https://www.vut.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=181027](https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=181027).