

Sem vložte zadání Vaší práce.

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



Diplomová práce

Webová aplikace pro správu rodokmenů

Bc. Martina Plocová

Vedoucí práce: Ing. Jaroslav Kuchař

19. února 2016

Poděkování

Ráda bych poděkovala vedoucímu mé práce Ing. Jaroslavu Kuchařovi za vedení mé práce, za užitečné rady a připomínky k práci a v neposlední řadě za trpělivost, kterou se mnou při konzultacích měl. Také bych chtěla poděkovat celé své rodině za veškerou podporu, kterou mi během psaní a vývoje poskytovali. Především však děkuji mému příteli, který musel vynaložit nejvíce tolerance, trpělivosti a podpory ze všech.

Prohlášení

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

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

V Praze dne 19. února 2016

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2016 Martina Plocová. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Plocová, Martina. *Webová aplikace pro správu rodokmenů*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Obsahem práce je průzkum dosavadních aplikací zabývajících se tvorbou a správou rodokmenu. Byla provedena analýza požadavků budoucích uživatelů. Podle získaných poznatků byla realizována webová aplikace pro správu rodokmenů. Aplikace se skládá ze dvou částí, klientské a serverové. Jako součást serverové části bylo navrženo a implementováno REST API, umožňující přístup k datům aplikace i dalším aplikacím třetích stran. Po vytvoření aplikace proběhlo testování s vybranými uživateli, při kterém byla prověřována funkčnost a další použitelnost implementované aplikace.

Klíčová slova správa rodokmenu, tvorba rodokmenu, genealogie, webová aplikace, REST API, JavaEE

Abstract

The thesis includes research of existing applications with focus on familytree creation and management. The analysis was conducted and according to analysis of requests from future application users. Based on the results from user research the web application for management of family trees was implemented. Application consists of two parts, client and server. REST API enabling access to applications data and other third-parties applications was designed

and implemented as part of server side of application. After creating the application , the application was tested with selected users in which it was tested for functionality and usability of implemented application.

Keywords familytree management,familytree creation, genealogy, web application, REST API, JavaEE

Obsah

Úvod	1
1 Analýza	3
1.1 Rešerše	3
1.2 Externí služby	13
1.3 Formáty pro uložení dat	15
1.4 Průzkum	17
1.5 Shrnutí rešerše a průzkumu	18
1.6 Požadavky na systém	20
2 Návrh	25
2.1 Použité technologie	25
2.2 Databáze	30
2.3 Návrh REST API	31
2.4 Architektura aplikace	35
3 Implementace	41
3.1 Jazyk	41
3.2 REST API	42
3.3 Klientská strana webové aplikace	47
4 Testování	51
4.1 REST API	51
4.2 Heuristická analýza	51
4.3 Testování s uživateli	53
Závěr	57
Literatura	59

A Seznam použitých zkratek	61
B Obsah přiloženého CD	63

Seznam obrázků

2.1	Hrubý návrh databáze	30
2.2	Návrh REST API pomocí fasády	33
2.3	Klientská část aplikace - práce se soubory	36
2.4	Návrh architektury aplikace	37
3.1	Návrh webového rozhraní pro export/import dat	48

Seznam tabulek

1.1	Zhodnocení aplikace MyHeritage	5
1.2	Zhodnocení aplikace FamilyTreeBuilder	6
1.3	Zhodnocení aplikace Geni.com	7
1.4	Zhodnocení aplikace FamilySearch	8
1.5	Zhodnocení aplikace Ancestry.com	10
1.6	Zhodnocení aplikace Family Tree Maker	10
1.7	Zhodnocení aplikace Ancestry	11
1.8	Zhodnocení aplikace Rodokmen PRO	12
1.9	Zhodnocení aplikace Family Echo	13
1.10	Porovnání současných aplikací	20

Úvod

Rodokmen neboli strom rodu byl dříve zpracováván pro panovníky a šlechtické rody a zasahoval mnoho století zpět. V posledních několika letech se stále širší veřejnost začíná zajímat o své předky a prapředky. Zjišťují svůj původ jak ze společenského tak z profesního hlediska. Nemalý význam hraje i lokalita pobytu předešlých generací rodu. Všemi těmito aspekty se zabývá a studuje věda zvaná genealogie. Na základě zjištěných údajů o předcích a pomocí různých metod zpracovává rodokmeny pro výchozí osoby a poskytuje tím budoucím generacím dané osoby informace o původu jejich předků.

V současné době probíhá digitalizace matrik a informace o prapředcích jsou opět dostupnější. Stále rozsáhlejší zájem o zpracování rodokmenů a dostupnost informací prostřednictvím webových aplikací vyvolává tlak na vývoj a realizaci programů jak lokálních tak internetových, které se genealogií zabývají a pomáhají uživatelům rodokmen vytvořit.

Cílem této práce je návrh, implementace a otestování webové aplikace pro tvorbu a správu rodokmenů. Práce je rozdělena do čtyř kapitol.

První kapitola pojednává o analýze zpracovávané problematiky. Zabývá se zjišťováním dostupnosti a funkcionalit některých již realizovaných programů a aplikací pro zpracování rodokmenů, a to jak na území ČR tak ve světovém měřítku. Specifikuje klady i zápory jednotlivých produktů pro definování parametrů navrhované aplikace. Zkoumá vhodné formáty pro zpracování a ukládání dat, včetně identifikace a zabezpečení přístupu k nim. Neméně významnou oblastí analýzy je také průzkum možné platformy pro provoz budoucí aplikace. V neposlední řadě je analýza zaměřena také na budoucí uživatele, od kterých získala potřebné informace o požadovaném obsahu dat.

Druhá kapitola popisuje vlastní návrh webové aplikace pro správu rodokmenů. Jsou zde uvedeny použité technologie zpracování dat, jejich principy a komponenty. Dále je navržena databáze a její struktura. Architektura aplikace popisuje vazby mezi klientem a serverem, jejich vzájemnou komunikaci a jednotlivé vrstvy aplikace.

Třetí kapitola se zaměřuje na implementaci navrhované aplikace. Je zde

uveden rozbor použitého jazyka Java Platform Enterprise Edition (JEE), určený pro vývoj a provoz podnikových aplikací a informačních systémů. Součástí kapitoly jsou další použité operace a použité anotace z REST API.

Ve čtvrté kapitole je popsáno testování aplikace s uživateli s posouzením výsledků použitelnosti a funkčnosti navrhované aplikace z hlediska předložených požadavků.

Analýza

1.1 Rešerše

1.1.1 MyHeritage

Společnost MyHeritage byla založena roku 2003 dnešním výkonným ředitelem Giladem Japhetem a několika dalšími genealogickými nadšenci přímo v domě jednoho z nich. Jedná se o sociální síť, která umožňuje sestavení vlastního rodinného webu, tvorbu rodokmenu, sdílení fotografií a vyhledávání předků. Stránky obsahují 6 milionů rodokmenů a jsou dostupné v 34 jazycích. MyHeritage je dnes druhou největší genealogickou společností a nejrychleji rostoucí rodinnou sítí na internetu. Web Myheritage je zaměřen na rodiny, rodinné historiky a genealogy. Primární službou je tvorba rodokmenů, její členové mohou plánovat a organizovat různá setkání, oslavy výročí a jiné události.[1]

1.1.1.1 Hodnocení

Mezi hlavní výhody MyHeritage patří, že stránky jsou v češtině, což usnadňuje orientaci v aplikaci starším generacím, které se o stavbu rodokmenu zajímají více. Nalezneme tu i funkci přidání člena rodiny ke správě rodokmenu, takže přestože administrátorem rodokmenu na stránkách MyHeritage zůstává jeho zakladatel, tak přizvaní členové rodiny mohou údaje v rodokmenu doplňovat o chybějící a pomáhají tak doplnit rodokmen o důvěryhodné informace. Stejně jako mohou přizvaní uživatelé upravovat data, mohou navzájem i sdílet nahrané soubory od jiných členů rodokmenu. Toto obojí usnadňuje práci administrátora se schvalováním každé i malé změny. Aplikace umožňuje i správu více rodokmenů najednou, toto se může stát užitečným např. při adopci, kdy pořádně nelze zkombinovat dva různé rodokmeny. Při orientaci v aplikaci také pomáhá možnost umístit ji na vlastní poddoménu, a tím zajistit rychlý přístup k rodokmenu. Aplikace umožňuje import/export dat do/z formátu GEDCOM.

Co však orientaci vůbec nepomáhá, je organizace nastavení stránek, které je rozdělené do několika až zbytečných kategorií, pokaždé umístěných na jiné

stránce. Najít konkrétní nastavení se proto špatně hledá. Bylo by lepší, kdyby byla veškerá nastavení, která může uživatel nastavit, na jedné stránce. Také je zde klasicky mnoho placených funkcí a rozšíření, případně omezení. V placené verzi je například i možnost filtrace údajů podle osoby, rodokmenu nebo větve rodokmenu. Mezi omezení patří například také omezení počtu vložených osob do rodokmenu na 250 lidí, což se může zdát mnoho, avšak pro početné rodiny by to mohlo být opravdu málo. Tento počet osob se dá navýšit zakoupením Premium účtu, ale i tak zůstává omezený a časem nutí uživatele dokoupit ještě PremiumPlus účet.

V placené verzi je i možnost filtrace údajů podle osoby, rodokmenu nebo větve rodokmenu. Podobně je tomu s prostorem na nahrávání souborů, které se budou hromadit, a pravděpodobně rychleji než osoby, a zaberou tím pádem rychleji maximální velikost úložného prostoru. Avšak obě tato omezení jsou drobnostmi oproti problému, který nastává při automatickém vyhledávání podobných osob z cizích rodokmenů. V případě, že program nalezne možnou shodu v osobě, nabídne ji uživateli, který si může osoby porovnat. Toto bohužel již ale bez upgradu účtu nedovolí shodu přijmout nebo odmítnout. Jelikož se většinou nejedná o stoprocentní shodu, pouze se prodlužuje seznam nalezených shod, bez toho, aniž by tyto nějak ubývaly. Nalezená shoda se totiž nedá potvrdit, takže ani nelze sloučit uživatele ani rodokmeny, stejně jako je nejde ani odmítnout ani odebrat. Vzniká z toho tedy takový spamový seznam. Při úspěšném sloučení rodokmenu mohou oba rodokmeny spravovat oba správci. Tím se předává důvěryhodnost o správě údajů cizímu člověku bez nutnosti jakéhokoliv schválení změny správcem rodokmenu. Toto potvrzení, že změnu je možno provést, zde chybí.

Tisk diagramu vše v jednom je pro free verzi omezeno jen na 250 osob.

Novou funkcí je rozpoznávání obličejů osob na nových fotografiích s již vloženými fotografiemi známých osob. Poslední novou pokročilou funkcí je metavyhledávač, který vyhledává v interní databázi a dalších světových databázích. Uživatelé mohou najednou vyhledávat až 5 variant hláskování hledané osoby s využitím metod Soundex a Megadex. Megadex je unikátní funkcionalitou MyHeritage. Tato umožňuje zvolit skupinu nejčastěji používaných hláskování a znělosti a je na rozdíl od Soundex kompatibilní se všemi databázemi. Uživatelé mohou své výsledky skladovat a komentovat. Ve free verzi nelze publikovat na web video, audio ani dokumenty. Přehlednější zhodnocení v tabulce. 1.1

1.1.1.2 MyHeritage Family Graph API

The Family Graph API poskytuje jednoduchý přístup k základním funkcím a údajům na MyHeritage.com. Představuje jednoduchý, konzistentní pohled na data s uniformní reprezentací objektů, jako jsou uživatelé, místa, rodokmeny, osoby, fotografie a zdroje a propojení mezi nimi včetně rodinných vztahů, označení fotografií, apod.

Výhody	Nevýhody
v češtině	API jen pro čtení
tvorba více rodokmenů	špatné vykreslování rodokmenu
umí export i import	nešikovné předávání osob z jiných rodokmenů
práce více uživatelů najednou	většina pokročilých funkcí bez zaplacení omezena
práce ve 2 jazycích zároveň	stránka je někdy zbytečně rozdělena do několika kategorií
funkce rozpoznávání obličejů na fotografiích	
funkce hláskování podle znělosti	
rychlé a jednoduché základní, vkládání osob	

Tabulka 1.1: Zhodnocení aplikace MyHeritage

The family Graph API je založeno na REST architektuře a všechna volání API vrací objekty JSON. API momentálně podporuje čtení informací, avšak zatím nepodporuje přidávání, úpravu a mazání informací. MyHeritage Family Graph API používá OAuth 2.0 protokol pro autorizaci. Pro přístup k API určitého uživatele je tedy nutné nejprve získat povolení od tohoto uživatele tím, že odešle autorizační token. Po získání tohoto tokenu mohou být provedeny autorizované požadavky v zastoupení daného uživatele tím, že se autorizační token zahrne do Family Graph API požadavku.

1.1.2 Family Tree builder

Společnost MyHeritage vyvíjí po boku online služeb rovněž genealogický software Family Tree Builder, umožňující využití pokročilých genealogických funkcí i bez připojení k internetu. Family Tree Builder je freeware a je nyní přeložen do 33 jazyků, počet jazyků se s časem zvětšuje. Pro aktivaci programu Family Tree Builder je bezpodmínečně nutná bezplatná registrace na stránkách MyHeritage s připojením na internet nebo přihlášení se ke stávajícímu účtu na stránce MyHeritage. Jednotlivé rodokmeny ukládá do samostatných souborů s příponou ".ftb", kterou používá jen tento program. Při prvním spuštění programu je nabídnuta možnost stáhnout data ze stránek MyHeritage, jen bez nalezených shod, ty se musí hledat znova. Program nabízí synchronizaci s webovou aplikací MyHeritage a možnost publikace z aplikace přímo na stránky MyHeritage.

1. ANALÝZA

Výhody	Nevýhody
kromě pokročilých funkcí zdarma	omezený počet osob v mapě
čeština	bez internetu nefungují některé funkce
neomezený počet osob	musí se aktivovat přes internet
práce s více rodokmeny	
export i import	

Tabulka 1.2: Zhodnocení aplikace FamilyTreeBuilder

1.1.2.1 Hodnocení

Program umožňuje práci s více rodokmeny, ale vždy může být otevřený pouze jeden. Oproti webové verzi není v programu omezení na počet osob v rodokmenu, avšak při aktualizaci do webové aplikace vznikají chyby (duplicity) při synchronizaci v případě, že na MyHeritage již existuje stejný rodokmen. Uživatelé mohou přidávat údaje o jednotlivých osobách, data narození, sňatku či úmrtí, apod. To je usnadněno tím, že kromě seznamu osob po levé straně obrazovky je zobrazen strom nejbližších příbuzných vybrané osoby, ve kterém je možné vyvolat editaci údajů nejen aktivní osoby, ale všech zobrazených, a jednoduše k němu lze přidávat příslušné příbuzné (rodiče, děti, partnery). Není možné tisknout diagram vše v jednom, pouze ve spojení s účtem Premium na stránkách MyHeritage. Bez předplaceného účtu se dá na mapě zobrazit pouze 50 osob. Mezi ostatní údaje, které lze vložit k dané osobě, patří i takzvané DNA markery, které lze použít pro ověření genetické příbuznosti osob. Více v tabulce 1.2

1.1.3 Geni.com

Geni.com je web poskytující službu propojené sociální sítě s důrazem na genealogii, která byla roku 2012 zakoupena společností MyHeritage. Web umožňuje rodinám společně stavět jejich rodokmen. K tomu je nutná registrace, uživateli potom stačí zadat e-mailovou adresu člena rodiny, a tím mu pošle pozvánku ke vstupu. Pozvaní uživatelé pak mohou upravovat profily své rodiny a kdokoliv tak může vytvářet rodokmen. Cílem Geni.com je vytvářet sdílené stromy přes společné předky a sjednocovat je do větších stromů. Pro ochranu informací o rodině jsou údaje soukromé, mohou se tedy vyhledávat pouze členové rodiny na základě podobností v rodokmenu. Pokud se začnou dva různé stromy protínat ve svých členech, mohou se stromy spojit dohromady. Díky kombinaci do jediného stromu, na kterém mohou uživatelé pracovat společně, se uživatelé mohou soustředit místo hledání předků na ověřování informací nebo na výzkum jiné větve stromu. Velkou nevýhodou je nemožnost importovat nebo

Výhody	Nevýhody
umožňuje jednoduchou spolupráci na rodokmenu)	česky není vše (kombinace dvou jazyků na jedné stránce)
propojení na sociální sítě	nejsou popsány vazby na ostatní osoby
zčásti čeština	zápis v API povolen pouze členům s Pro účtem
má API s autorizací	neumožňuje export ani import dat
čtení i zápis v API	pouze pro jeden rodokmen
	rodokmen musí být založený na tom, kdo ho založil

Tabulka 1.3: Zhodnocení aplikace Geni.com

exportovat založený rodokmen, rodokmen se musí vždy zakládat znova a nedá se jakkoliv zálohovat nebo přenášet do jiných aplikací. [2]

1.1.3.1 Hodnocení

Stránky nabízí možnost prohlížet a vyhledávat osoby z rodokmenů i bez registrace a přihlášení. Tyto stránky sice obsahují český překlad, ale ne všude. Polovina jedné stránky může tedy být přeložena do češtiny a druhá půlka může být v angličtině. Žádná ze stránek tudíž nemá celý obsah v jednom jazyce. Další nevýhodou je, že rodokmen se vyobrazuje pouze od zakládajícího uživatele nahoru k předkům, ale rodokmen dolů k potomkům se již nezobrazuje. Nejsou dostatečně popsány typy vztahu, jako stav rozvedení, snoubenci, manželé apod. Tyto se rozlišují pouze typem spojovací čáry, což je nevýrazné. Většina funkcí je poskytována zdarma, s výjimkou prostoru pro vkládání multimediálního obsahu, který je omezen na 1GB, a s výjimkou pokročilého vyhledávání podobností nebo samotného porovnávání údajů o osobě v různých rodokmenech. Porovnání výhod a nevýhod bylo shrnuto do tabulky 1.3 .

1.1.3.2 The Geni API

Geni API stejně jako MyHeritage Family Tree API používá OAuth 2.0 protokol pro autorizaci a autentifikaci a rovněž je založeno na REST architektuře, volání API vrací objekty typu JSON.

Avšak oproti MyHeritage Family Tree API umožňuje informace kromě čtení i vkládat, upravovat a mazat.

1.1.4 FamilySearch.org

FamilySearch je genealogická organizace provozovaná společností Genealogy Society of Utah („GSU“). Organizace nabízí on-line volně přístup ke svým záznamům a službám na FamilySearch.org. Tyto webové stránky jsou jedny

1. ANALÝZA

Výhody	Nevýhody
největší databáze genealogických záznamů	nemá češtinu
při založení osoby hned hledá shodu v databázi	nepohodlná editace osob na stránce rodokmenu
prohledávání podle lokality nebo události	neumožňuje některé vazby mezi osobami (manželé stejného pohlaví,...)
má API	může vytvořit jen jeden strom

Tabulka 1.4: Zhodnocení aplikace FamilySearch

z nejvytíženějších genealogických sítí na internetu. Jde o největší sbírku genealogických a historických záznamů na světě a dají se zde nalézt i digitalizované matriky, které nejsou na aplikacích archivů v rámci České republiky k dispozici.[3]

1.1.4.1 Hodnocení

Jelikož jsou stránky vlastněny katolickou církví, nepovolují v rámci tvorby rodokmenu vkládání manželství stejného pohlaví nebo jiných spojení. Při tvorbě rodokmenu se špatně zaměřuje na workspace, musí se tedy na velké ploše hledat, kde vlastně se dá pracovat s rodokmenem. Špatně a pomalu se vkládají členové v rodokmenu, musí se vkládat po jednom na vlastní stránce osoby. Náročný způsob vkládání údajů k osobě. Pomalé a nepřehledné vkládání informací k osobě na stránce rodokmenu, jednodušší je vložit údaje přímo na stránce samotné osoby, tím se prodlužuje a komplikuje způsob vkládání více informací k více lidem. Při vložení nové osoby se zároveň prohledají záznamy kvůli podobnostem, což šetří čas zvláště při případném pozdějším dohledávání zvláště. Zajímavá je zde možnost vyhledávat zdroje pro vlastní porovnávání podle místa či tématu. Více v tabulce 1.4.

1.1.4.2 FamilySearch API

I FamilySearch API používá k autorizaci protokol OAuth 2.0 a postup autorizace je tak stejný. Také se jedná o RESTful webovou službu, která může být použita jak pro čtení a zápis, tak pro úpravu a mazání informací v různých zdrojích z FamilySearch. RESTful webová služba používá webové technologie HTTP, XML a JSON. Pro používání API je nutné obdržet unikátní App Key, který je určen pro aplikaci. Tento App Key musí být použit pro přístup k volání FamilySearch API.

1.1.5 Ancestry.com

Ancestry.com Inc., dříve The Generations Network je soukromá společnost se sídlem v Provo v USA. Jedná se o největší genealogickou společnost na světě, která provozuje síť genealogických a historických stránek, převážně v USA. Vyvíjí a prodává genealogický software a nabízí širokou škálu služeb souvisejících s genealogií. [4]

Ancestry.com jsou webové stránky na bázi předplatného s více než 5 miliardami on-line záznamů převážně z USA a poskytuje tvorbu jednoduchého rodokmenu se základními údaji. Tento rodokmen dále slouží jako základ pro vyhledávání podobností a shod v záznamech z poskytnutých zdrojů. Některé záznamy jsou zdarma přístupné pro každého, ale většina z nich je přístupných pouze v rámci předplatného. Společnost poskytuje i placený software Family Tree Maker, ten jako ostatní genealogické programy pomáhá spravovat rodokmen a informace získané během výzkumu rodokmenu a tvořit z těchto údajů grafy, zprávy apod.

1.1.5.1 Hodnocení

Stránky vyžadují registraci, bez které si není možné prohlédnout, co stránky přibližně poskytují. Tvorba rodokmenu je zde velmi jednoduchá a rychlá, avšak s velmi málo údaji v přehledu, pro další podrobnější údaje jako datum a místo svatby nebo rozvodu je třeba proklikat se na vlastní stránku osoby a také zde údaje zvláště přidat. Při tvorbě rodokmenu se nepočítá s možností, že manželčino příjmení se liší od manželova nebo se změnou jména během let, poskytuje pouze manželčino jméno za svobodna, což má negativní vliv na vyhledávání podobností v záznamech. Většina informací kromě struktury rodu se zobrazuje vždy vzhledem k jedné osobě, nelze zde tedy vidět údaje vzhledem k více členům rodiny najednou. Stránky slouží primárně pro vyhledávání záznamu v několika zdrojích z celého světa, avšak pro přímou tvorbu a správu rodokmenu jsou nevhodné. 1.5

1.1.6 Family Tree Maker

Je desktopová aplikace společnosti Ancestry pro vytváření rodokmenu. Program je pouze v angličtině, ale lze ho používat v počítačích PC s operačním systémem MS Windows a počítačích Mac. Lze ho používat i na iPodech a tabletech. Pro import a export dat používá formát GEDCOM. Může provádět synchronizaci dat s online verzí. Umí pomocí vlastního migračního nástroje sdílet soubory mezi verzí ve Windows a verzí Mac. Umožňuje pracovat s více databázemi najednou.

1. ANALÝZA

Výhody	Nevýhody
největší databáze	vyhledává záznamy v USA
také desktopová aplikace	bez předplatného je zakázána většina funkcí
rychlé založení, tvorba rodokmenu	špatný přístup k detailním údajům o osobě
umí vyhledávat v různých zdrojích z celého světa	špatný přístup k nahraným globálním souborům
	špatná orientace na stránkách
	chybí import dat
	nemá češtinu
	nemá API

Tabulka 1.5: Zhodnocení aplikace Ancestry.com

Výhody	Nevýhody
umí import a export	je celý placený
více rodokmenů	není demo verze
pro pc a mac	není v češtině

Tabulka 1.6: Zhodnocení aplikace Family Tree Maker

1.1.6.1 Hodnocení

Program obsahuje mnoho funkcí, ale je celý placený. Proto neumožňuje vyzkoušení před zaplacením. Zákazník se musí orientovat při výběru pouze podle popisu na webových stránkách. Prostřednictvím formátu GEDCOM umožňuje spolupracovat s ostatními genealogickými programy. 1.6

1.1.7 Ancestry

Výhodou je, že celý program je v češtině a případná technická podpora je také v češtině včetně nápovědy. Program umí v rámci jednoho rodokmenu členit osoby do skupin, což zjednodušuje práci v rozsáhlých rodokmenech. Dále umožňuje prostřednictvím formátu dat GEDCOM přenos údajů programu Ancestry a dalšími programy. Je možné vygenerovat si příbuzenské stromy, jako je např. rodový vývod, rozrod rodu apod., libovolně si je v programu upravovat a pak je uložit ve formátu JPG, PDF či SVG. Při zobrazení v programu však lze pohybovat plochou zobrazeného rodokmenu pouze posuvníky na okraji plochy. Jednotlivé rodokmeny lze rozdělovat a spojovat. Program dokáže uživatelem zadané údaje zpracovat formou statistik, výročí a tabulek. Do nich zahrne pouze osoby, které mají vyplněné veškeré potřebné údaje. Statistiku, přehled

Výhody	Nevýhody
Je v češtině	nešikovná práce se soubory
podpora v češtině	podobné funkce se volají různě
umí více rodokmenů	neumí vazby na další osoby (pěstouny, manželé stejného pohlaví)
import a export	
export do dalších formátů	

Tabulka 1.7: Zhodnocení aplikace Ancestry

výročí a kalendář výročí jsou hlavními plusy tohoto programu. Nevýhodou je, že každá funkce se volá v jiném místě programu. Statistiky se vyvolají kliknutím na záložku statistika formuláře dané osoby, jiné funkce se volají z menu, přestože v obou případech je výsledkem sestava z celého rodokmenu. Program je orientován téměř výhradně textově, to znamená, že všechny údaje se vkládají v textové podobě. Pouze sestavy zobrazující rodokmen mají grafickou podobu, kterou lze upravovat bez zpětné vazby do rodokmenu. Program neumožňuje jako partnery zadat osoby stejného pohlaví. Vazby na další osoby, jako pěstouny apod. lze pouze dokreslit do grafické podoby rodokmenu, při opětovném vygenerování se ale ztratí. Dále nelze vložit vazbu na sourozence, pokud nejsou vloženi rodiče. Program neobsahuje možnost zobrazení osob v mapě ani jiné geolokační funkci. Práce se soubory, které mají vazbu na rodokmen nebo osoby v rodokmenu, je komplikovaná, protože uživatel nemá přehled, zda se soubory pracuje globálně nebo na úrovni konkrétní osoby. Célmu programu v zásadě lze vytknout, že striktně neodděluje globální funkce pro celý rodokmen a lokální funkce pro konkrétní osoby. [5]

1.1.7.1 Hodnocení

Výhodou je, že celý program je v češtině a případná technická podpora je také v češtině včetně nápovědy. Umožňuje prostřednictvím formátu dat GEDCOM přenos údajů s jinými programy. Je možné vygenerovat si "příbuzenské stromy", jako je např. rodový vývod, rozrod rodu, apod., libovolně si je v programu upravovat a pak je uložit jako JPG, PDF či SVG. Program dokáže uživatelem zadané údaje zpracovat formou statistik a tabulek. Do statistik zahrne pouze osoby, které mají vyplněny veškeré potřebné údaje. Statistiky jsou hlavním plusem tohoto programu. Jejich zobrazení se vyvolá kliknutím na kolonku Statistika rodokmenu. Všechny statistiky se zobrazí v jednom sloupci. Statistiky jsou podle různých klíčových údajů. 1.7

Výhody	Nevýhody
je v češtině	je celý placený
demo verze na 15 dní	málo funkcí
import a export	málo funkcí a údajů

Tabulka 1.8: Zhodnocení aplikace Rodokmen PRO

1.1.8 Rodokmen Pro

Je dalším českým genealogickým programem, oproti ostatním je však relativně nový. Pro dlouhodobější práci je nutné zakoupit licenci, přesto je možné program vyzkoušet v rámci patnáctidenní free verze, která neskýtá žádná omezení. Po uplynutí patnáctidenní lhůty již nelze ukládat nová data na disk, i když možnost exportu již zadaných dat, tisku či zpřístupnění na internetu zůstává. Program ukládá vytvořené databáze do souborů RodokmenPro (.rp). Jako většina genealogických programů umožňuje export dat do souboru GEDCOM, a také import těchto souborů. [6]

1.1.8.1 Hodnocení

Přestože se jedná o placený program, obsahuje pouze ty nejzákladnější funkce, které by měl genealogický program mít. Mnohé neplacené verze jiných genealogických programů nabízejí mnohem více funkcí než RodokmenPro. Cena plné verze je však malá, takže odpovídá rozsahu poskytovaných funkcí. Práce s programem je jednoduchá. Z hlediska množství dostupných funkcí však není, čím se v něm inspirovat. Jeho jedinou výhodou je, že je celý v češtině. Dostupné zhodnocení najdete v tabulce 1.8.

1.1.9 Family Echo

Jednoduchá webová aplikace pro tvorbu rodokmenu. Soustředěná především na vizualizaci rodokmenu. Rodokmen je možné sdílet a zasílat přes e-mail členům rodiny.[7]

1.1.9.1 Hodnocení

Aplikace umožňuje vkládání minimálního množství údajů k osobě. Také má omezený počet generací od základní osoby, aby rodokmen nekonečně nerostl. Pokud je dosaženo tohoto limitu, musí se vytvořit nový rodokmen, počínajíc od osoby, která se do původního rodokmenu nevešla. Aplikace je zajímavá z hlediska možnosti exportu dat do formátu CSV, FamilyScript, GEDCOM nebo do Dynamického HTML, díky čemuž se vytvořená data dají použít jako základ do jiných aplikací. 1.9

Výhody	Nevýhody
export dat do více formátů	poskytované formy dat umí zpracovat pouze tato aplikace
export do HTML	vložit se dá jen minimální počet údajů
má API	zobrazuje omezený počet generací
	nestandardní API

Tabulka 1.9: Zhodnocení aplikace Family Echo

1.2 Externí služby

1.2.1 Google Maps API

Mapy zobrazují většinu světa s přibližně stejnou přesností. Google Maps je světově nejrozšířenější online mapová služba na světě a použitím pouze pro nekomerční použití. Obsahuje několik podřazených API. Mezi ty nejzajímavější patří Google Maps JavaScript API, Google Static Maps API, Google Maps Embedded API, Google Maps Directions API a Google Maps Geocoding API.

1.2.1.1 Google Maps Geocoding API

Geocoding je proces, který umožňuje získat z adresy zeměpisné souřadnice, které se mohou využít pro umístění uzlu na mapu nebo k umístění mapy. API umožňuje i reverzní geocoding, který naopak získá ze zeměpisných souřadnic adresu. Díky reverse geocoding je možné získat adresu i ze zadaného ID místa. Google Maps Geocoding API poskytuje přímý způsob získání těchto služeb pomocí HTTP požadavku. API se využívá převážně pro umístování obsahu na mapu. [8]

1.2.2 Yandex.Maps API

API patří jedné z největších internetových společností v Evropě, která provozuje největší ruský vyhledávač a jeho nejnavštěvovanější webové stránky. API poskytuje nezbytné nástroje pro práci s Yandex mapami pro webové aplikace nebo na web. Ačkoliv jsou všechny nástroje a API přínosné, nás nejvíce zajímají JavaScript API, Geocoder, YMapsML a Places Http API.

Jelikož je s API poskytováno i rozhraní Build Map, které kombinuje rozhraní obsažená v Yandex.Maps pro tvorbu vlastní mapy, dá se funkčnost i odzkoušet. S API se pracuje dobře, je snadno pochopitelné a obsahuje mnoho zajímavých funkcí, včetně možnosti obarvení tras a uzlů. Nevýhodou je nemožnost vložení více událostí/ popisu k jedné lokaci, respektive uzlu, je nutné vložit několik uzlů do blízkého okolí, což způsobuje nepřehlednost.

1.2.2.1 JavaScript API

Je sada komponentů pro vkládání interaktivních Yandex map na webové stránky nebo do webových aplikací. API umožňuje zobrazení mapy s různorodými předměty, vyhledávání adres, plánování a vytváření tras a další. Avšak rozhraní vyžaduje podporu JavaScriptu a při konstrukci mapy se zadanými prvky vzniká poměrně dlouhý JavaScript kód, což by mohlo způsobovat pomalou odezvu na dotaz. Také je zde problém s odhadem škálování mapy při velké vzdálenosti jednotlivých bodů od sebe. Přesto se jedná o šikovné a jednoduché rozhraní.

1.2.2.2 Geocoder

Komponenta pomáhá určit zeměpisné souřadnice a další informace o objektu na základě jeho jména nebo adresy i naopak, pak se funkce nazývá reverzní geocoding. Ke Geocoderu se dá přistupovat buď přes protokol HTTP nebo pomocí zmíněného JavaScript API. Pokud se k rozhraní přistupuje přes HTTP, odpověď se může vrátit jako XML document v YMapsML nebo jako object format JSON.

1.2.2.3 YMapsML (Yandex Maps Markup Language)

YMapsML je jazyk vyvinutý společností Yandex pro popis zeměpisných údajů, tím jsou myšleny všechny údaje, které jsou geograficky spojeny s místem. Tento formát se používá pro přenos dat mezi Yandex.Maps a softwarem třetích stran. Geocoder používá tento formát dat pro vrácení informací o umístění objektu.

1.2.2.4 Places HTTP API

Pomáhá zjistit souřadnice přírodních útvarů, ale také podporuje vyhledávání informací a polohy podle jména společnosti, její adresy, telefonu nebo třeba i podle poskytovaných služeb. Pro používání rozhraní je zapotřebí API key a data jsou vrácena ve formátu JSON nebo JSONP.

1.2.3 Yandex.Translator API

API poskytuje přístup k online službě automatického překladu. Podporuje více než 60 jazyků včetně češtiny a dokáže přeložit jak jednotlivá slova, tak celé texty. Díky API je možné, aby je překladatel vložil do mobilní aplikace nebo webové služby. Dokáže tedy přeložit i velké množství textu, jako např. technické dokumentace. Rozhraní vyžaduje použití API Key a pro přístup k Yandex.Translator API pomocí HTTPS jsou dostupná rozhraní XML a JSON či JSONP. Všechna rozhraní poskytují stejnou funkcionalitu a používají stejné vstupní parametry.

Yandex automatický překlad je založen na statistickém přístupu. Pro naučení jazyka se porovnávají tisíce paralelních textů a vzájemně se po větách překládají. Má dvě hlavní komponenty, překladový model a jazykový model. Překladový model vytváří graf obsahující všechny možné cesty, jak větu přeložit. Jazykový model poté z těchto možností vybere nejlepší překlad z hlediska optimálního výběru slov v překládaném jazyce. Jazykový model je sestaven z velkého jednojazyčného souboru textů a obsahuje nejčastější n-slovné kombinace v jazyce.

1.2.4 Bing Translator API

Bing Translator API je rozhraní vytvořené firmou Microsoft a jedná se pravděpodobně o nejlepší překladové API, které se dá najít i v neplacené verzi. Bing Translator Control povoluje Windows Store App běžet na systému Windows pro překlad textu z jednoho jazyka do jiného. Bing Translator Control vezme přijatý text a předá ho webové službě Bing Translator k překladu a pak pošle překlad vstupu zpět. Aplikace může přeložený text zpracovat, jak bude potřebovat.

Avšak pro použití Bing Translator API musí mít uživatel účet u Microsoftu, dále je nutné registrovat aplikaci, zakoupit členství, které je do jistého limitu zdarma, a nastavení údajů pro autorizaci. Navíc celkový přístup k rozhraní Bing Translator API je nepřehledný.

1.3 Formáty pro uložení dat

Většina genealogických programů používá pro export a import datový soubor ve formátu GEDCOM. Některé programy ale umožňují pouze export, nebo jen import tohoto typu souboru a najdou se i takové, které formát GEDCOM nepodporují vůbec. Výjimečně se objeví i aplikace jako Family Echo, která používá vlastní specifický formát dat (FamilyScript) nebo obecný formát dat (např. CSV).

1.3.1 Formát GEDCOM

GEDCOM je zkratka zastupující GENEalogy Data COMMunication. GEDCOM je tedy jazyk, který umožňuje různým genealogickým programům komunikovat mezi sebou. Cílem je možnost výměny dat mezi rozdílnými programy bez nutnosti opakovaně manuálně zadávat data. Jedná se o standard, nikoliv program, proto musí genealogický program být schopen zpracovat tento formát. Chceme-li provést výměnu dat mezi dvěma programy, musí oba programy formát GEDCOM podporovat. [9]

Formát GEDCOM je hierarchický a založený na odkazech, podobně jako HTML, ale kóduje strukturu rodiny jako genealogický graf. V zásadě se jedná o soubor záznamů o jednotlivcích (Individual - IND), propojených pomocí

uzlu rodiny (Family - FAM). Uzel rodiny obsahuje maximálně jednoho manžela (Husband - HUSB), maximálně jednu ženu (wife - WIFE) a několik dětí (child - CHILD), ostatní vztahy jako bratři, sestry, apod. se řeší spojením jednotlivých rodinných uzlů. Uzel může dale obsahovat doplňující informace o vztahu partnerů, jakými jsou informace o svatbě či rozvodu. Partneri však nemusí být nutně manželé, avšak jakékoliv dítě může mít vždy maximálně jen 2 rodiče (biologické). V případě adopce či pěstounství se dá pouze uvést poznámka k jednotlivci, že uvedená rodina není biologická, jinak je dítě přiděleno případně biologické rodině.

GEDCOM 5.5.1 byl vydán v roce 1999, představuje devět nových tagů (WWW, EMAIL a FACT) a přidává UTF-8 jako schválené kódování. Tento návrh nebyl formálně uznán, ale jeho možnosti jsou částečně použity v některých genealogických programech. 6. prosince 2002 byla uvolněna beta verze specifikace GEDCOM 6.0 pro programátory k seznámení a k implementaci do programů. GEDCOM 6.0 je první verzí s formátem dat v XML. Má podporovat zapisování znaků textu v kódu Unicode, který umožňuje například zápis východoasijských jmen originálními znaky, bez nejednoznačností, což by mohlo být využitelné pro celosvětový genealogický výzkum.

1.3.2 Formát FamilyScript

Je formát podporovaný pouze programem Family Echo. FamilyScript je optimalizován pro efektivní komunikaci a rychlou interpretaci dat v programu. Každý soubor popisuje osoby pouze v jedné rodině. Jednotlivým osobám je přiděleno unikátní alfanumerické ID. Soubor používá kódování UTF-8 Unicode. Je rozdělen do řádků, které oddělují LF a CR. Obsah a význam dat na příslušném řádku je indikován prvním znakem v daném řádku. Soubor může obsahovat i nevýznamné komentáře.

1.3.3 Formát CSV

CSV (Comma-separated values, hodnoty oddělené čárkami) je jednoduchý souborový formát. Soubor ve formátu CSV obsahuje řádky, ve kterých jsou jednotlivé položky odděleny čárkou. Hodnoty položek mohou být uzavřeny do uvozovek, což umožňuje, aby text položky obsahoval čárku. Uvozovky se v textu zdvojují. Jelikož se v některých jazycích včetně češtiny čárka používá v číslech jako oddělovač desetinných míst, existují varianty, které používají jiný znak pro oddělování položek než čárku, nejčastěji středník, případně tabulátor (taková varianta se pak někdy označuje jako TSV, Tab-separated values). Variantu se středníkem (ale stále pod názvem CSV) používá např. Microsoft Excel v české verzi Microsoft.

1.4 Průzkum

1.4.1 Příprava

Pro průzkum bylo vybráno 15 účastníků s věkovým rozsahem 22 – 58 let. Účastníkům budou na základě odpovědi na první otázku podány různé následující otázky. Cílem je roztrždit účastníky na základě jejich zkušeností s tvorbou rodokmenu do dvou skupin.

První skupina bude obsahovat jedince, kteří se doposavad nepokusili nebo nezajímali o tvorbu jakéhokoliv rodokmenu. A nemají tedy žádné povědomí o současných aplikacích či problémech při tvorbě rodokmenu. Vzhledem k nezkušenosti s jinými programy bude skupina nápomocná převážně pro určení základních funkcí aplikace, které by od každé aplikace byly pravděpodobně vyžadovány. Předpokládáme, že v této skupině budou i jedinci, kteří se někdy zajímali o sestavení rodokmenu a u nich předpokládáme, že nad problematikou tvorby rodokmenu více přemýšleli a mohou tak poskytnout i další zajímavé funkce.

Do druhé skupiny budou patřit zkušenější účastníci, kteří odpověděli, že mají nějaké zkušenosti s tvorbou rodokmenu, a to ať v nějaké aplikaci, tak i díky vlastnímu průzkumu nebo jinému intenzivnějšímu projevení zájmu o své předky. Vzhledem k jejich pokročilejší znalostem s prací s rodokmeny jsou tito jedinci obeznámeni s problematikou tvorby rodokmenů, znají konkrétní typy dat, které se mohou vkládat do rodokmenu a které prakticky v rodokmenu opravdu využijí. Mohou tak poskytnout ucelenější náhled na klady a zápory současných aplikací včetně svého názoru. Díky tomu mohou z vlastní zkušenosti zhodnotit, které části i jiných aplikací jsou užitečné, které nepodstatné a které jsou problémové z pohledu funkčního nebo logického. Zkušenost této skupiny může pomoci při úpravě funkcionality požadovaných funkcí aplikace a také může přinést návrhy na nové funkce.

V rámci průzkumu byli účastníci postupně dotazováni na následující otázky. První otázka byla pro všechny účastníky společná, ostatní otázky se lišily v závislosti na odpovědi na první otázku.

1. Máte nějaké zkušenosti s tvorbou vlastního či cizího rodokmenu?
Pokud ano, jaké?

Skupina účastníků bez zkušeností s tvorbou rodokmenu:

2. Zajímali vás někdy vlastní rodokmen? Přemýšlel/a jste někdy nad jeho sestavením?
3. Znáte nějaké programy či aplikace pro tvorbu rodokmenu?
4. Co byste od aplikace pro tvorbu a správu rodokmenu očekávali?
Jaké funkce, obsah?

Skupina účastníků se zkušenostmi s tvorbou rodokmenu:

2. Využíval/a jste někdy aplikaci pro tvorbu rodokmenu? Pokud ano, jakou?
3. Byl/a jste s touto aplikací spokojena? Popište, co se Vám na ní líbilo/nelíbilo. Proč jste si vybral/a tuto aplikaci?
4. Znáte nějaké další programy pro tvorbu rodokmenu? Jaké?
5. Co by podle Vás měla ideální aplikace pro tvorbu rodokmenu obsahovat? Pokuste se jednotlivým funkcím přiřadit prioritu.
1- velmi důležité (nutné), 3-užitečné, 5-není nutné

1.4.2 Výsledky průzkumu

Většina dotazovaných spadá do skupiny účastníků bez zkušeností s tvorbou rodokmenu. Tato skupina je složena převážně z mladších jedinců průzkumu, což může naznačovat, že mladší generace se o tvorbu rodokmenu prozatím příliš nezajímá. Na základě výsledků průzkumu můžeme říci, že předpoklad, že jedinci, kteří se někdy zamysleli nebo přemýšleli nad tvorbou rodokmenu, se potvrdil, tito účastníci se během průzkumu více angažovali a lépe spolupracovali.

Na základě porovnání požadavků účastníků průzkumu na funkce aplikace jsme jako základní vybírali funkce, které se u účastníků často opakovaly, a byla jim přiřazována vyšší priorita. Vysoké požadavky byly kladeny hlavně na snadnost práce s aplikací a její přehlednost. Dále pak možnost tvorby genealogického stromu (rodokmenu) grafickou podobou, výpis všech osob ve stromech, úprava dat osoby ve stromu, možnost uložení grafické podoby genealogického stromu do souboru, vkládání a úprava dokumentů, fotografií, prolínání uživatelů skrze společné dokumenty a zobrazení geografických údajů na mapě.

Vybrali jsme i další funkce, které byly během průzkumu zmíněny, i když s nižší prioritou nebo bez častého opakování. Tyto funkce jsme vybrali kvůli jejich originalitě, praktičnosti nebo kvůli nedostatečnosti současných řešení. Mezi tyto funkce patří zobrazení pohybu osoby na mapě chronologicky, možnost přiřazení jednoho souboru více osobám z rodokmenu, ukládání vytvořených rodokmenů do souboru jako způsob zálohy, možnost přidání vlastní definované poznámky a import dat z aplikace do jiných aplikací.

Mezi funkce, které nebyly pro samotnou aplikaci vybrány patří například lékařské statistiky, chorobopisy, napojení na matriky, přepis znaků ze staré češtiny, automatický převod dokumentů z psané formy na digitální, apod.

1.5 Shrnutí rešerše a průzkumu

Provedli jsme rešerši stávajících aplikací a jejich API, externích geolokačních a rovněž slovníkových webových služeb. Dále jsme provedli průzkum funkcí

pro aplikace pro rodokmen a na základě poznatků z této řešerše jsme zjistili, že mezi nejoblíbenější ze současných aplikací pro správu rodokmenů patří aplikace MyHeritage. Je to zejména z důvodu podpory češtiny a poměrně přehledného vzhledu aplikace. Vezmeme tedy v úvahu její strukturu při tvorbě nové aplikace a pokusíme se odstranit nedostatky, které aplikace obsahuje, například při vkládání osob do rodokmenu, při vykreslování rodokmenu nebo vkládání adres. Díky průzkumu můžeme aplikaci také doplnit o další zvolené funkce, které pomohou usnadnit a zpříjemnit práci s aplikací a které budou uživatelům poskytovat možnosti, chybějící ve stávajících aplikacích. Stejně tak nebudeme do aplikace zahrnovat takové funkce, které přidělávají uživatelům práci a pouze dělají aplikaci nepřehlednou.

I když některé aplikace pro tvorbu rodokmenů obsahují vlastní API, ne vždy toto API je plně dostupné. V případě MyHeritage aplikace API sice poskytuje, avšak je nutná aktivace ze strany podpory aplikace. Tato, i když aplikaci nejspíš někdo spravuje, každým dotazem spíše uživatele nenásilnou cestou nutí zakoupit si placené předplatné aplikace. Každý zasláný e-mail, i vygenerovaný aplikací na podporu, končí oznámením, že požadavek je odeslán a že se podpora co nejdříve ozve. Také připomíná, že pokud chce uživatel lepší a rychlejší komunikaci, může zaplatit prémiové členství. I přes ujištění, že se podpora ozve co nejdříve, nebyla žádost o aktivaci API klíče pro přístup k MyHeritage Family Graph API vyřízena ani během 3 měsíců, a to i přes opakované zasílání žádostí o aktivaci. Podobným případem je API aplikace FamilySearch.org, která opět vyžaduje autorizaci aplikace a API klíče ze strany administrátora aplikace po vyplnění registrace a údajů o aplikaci, která bude API používat. V tomto případě však není podbízeno žádné zakoupení předplatného, naopak není zde žádná komunikace ze strany administrátora nebo technické podpory. Opět tedy ani během 3 měsíců aplikace nebyla certifikována a ani se její stav nijak nezměnil. U API aplikace Geni problém s autorizací aplikace není, proto jsme si její API zvolili pro import mezi aplikacemi. Ani toto API ovšem není bez chybičky. Je zde omezení, že data lze z API pouze číst a minimálně upravovat, vkládání dat je povoleno pouze uživatelům s aktivním PRO účtem, proto není možné provádět export z aplikace do Geni.

Z dalších externích služeb jsme se rozhodli použít geolokační a mapové služby od Google Maps API, které má detailní a přehlednou dokumentaci a poskytuje značné množství různých API pro práci s mapami. Po řešerši a otestování slovníkových služeb jsme zjistili, že dostupné služby bohužel neposkytují žádnou nebo minimální podporu českého jazyka a překlady nejsou vždy přesné, pokud vůbec jsou. Tyto služby jsou spíše zaměřené na světové jazyky, jakými jsou angličtina, němčina, francouzština a některé další. Ideální služba pro překlad by byla Google Translate API, tato češtinu podporuje a velmi dobře překládá, avšak je licencovaná a placená, a to i pro minimální počet překladů.

1. ANALÝZA

Aplikace	Čeština	Import dat	Export dat	Práce s více rodokmeny	Přístupné API	Neomezený počet osob	Placené funkce	Typ aplikace	Přívětivost rozhraní
My Heritage	X	X	X	X	X*		X	web	2
Family Tree Builder	X	X	X	X		X	X	desktop	2
Geni.com	částečně				X		X	web	3
FamilySearch.org					X*		X	web	3
Ancestry.com		X	X	X			X	web	4
Family Tree Maker		X	X	X			X	desktop	-
Ancestry	X	X	X	X		X		desktop	2
Rodokmen Pro	X	X	X				X	desktop	-
Family Echo		X	X	X		X		web	2

Vysvětlivky k tabulce 1.10: *) API sice existuje, ale není dostupné

Hodnocení v posledním sloupci je jako ve škole 1 - 5

Tabulka 1.10: Porovnání současných aplikací

1.6 Požadavky na systém

1.6.1 Popis produktu

Aplikace je určena pro uživatele, které zajímá historie rodiny a chtějí si sestavit svůj rodokmen. Má poskytovat uživatelům na základě požadavku grafické i textové výstupy, které usnadňují správu a orientaci ve vložených datech. Aplikace je rozdělena na klientskou a serverovou část.

Klientská část umožňuje uživatelům registraci do webové aplikace, úpravu účtu, založení rodokmenů, jejich správu, ukládání osoba do rodokmenů, úpravu jejich informací, vkládání dat pomocí exportu/importu a komunikaci s externími aplikacemi. Klientská část na základě požadavku uživatele komunikuje se serverovou částí aplikace.

Serverová část aplikace by měla provozovat a využívat vlastní REST API, které bude poskytovat uživatelům a případným vývojářům přístup k datům i

mimo webovou aplikaci. Realizuje autorizaci uživatele pro manipulaci s daty. Zprostředkovává a zpracovává data z databáze a předává je klientské části. Server předává data v nezpracované formě nebo ve formě předpřipravených webových stránek.

1.6.2 Požadavky

1.6.2.1 Funkční požadavky

- Webové rozhraní aplikace pro správu a tvorbu rodokmenu
- Serverová strana s REST API pro manipulaci s daty
- Webová aplikace bude využívat REST API
- Registrace uživatele
- Správa účtu uživatele
- Tvorba rodokmenu a jeho zobrazení v grafické formě
- Úprava informací o osobě v rodokmenu
- Osobní stránka osoby se všemi údaji
- Výpis všech vložených osob uživatele
- Zobrazení údajů z rodokmenu do mapy
- Správa vložených souborů
- Přidání osoby k souboru
- Export/import dat do/ze souboru
- Export/import dat mezi jinými aplikacemi

1.6.2.2 Nefunkční požadavky

- **Vyhledávání osob** – aplikace bude vyhledávat osoby prostřednictvím jiných API pro tvorbu rodokmenu
- **Statistiky** – aplikace bude zobrazovat statistiky o vložených osobách

1.6.3 Případy užití

1.6.3.1 Seznam účastníků

Neregistrovaný uživatel

- Do aplikace nemá přístup. Aby přístup získal, musí se zaregistrovat

Registrovaný uživatel

- Upravuje své přihlašovací údaje
- Tvoří a spravuje své rodokmeny
- Vkládá a odstraňuje osoby ve svých rodokmenech
- Přidává a upravuje informace o osobách v rodokmenech
- Zobrazuje seznam všech osob ve všech rodokmenech a jejich osobní stránky se všemi jejich vloženými informacemi
- Ukládá ke svému profilu dokumenty, fotografie či jiné soubory. Tyto soubory může přiřazovat k osobám ve svých rodokmenech a přidávat k nim doplňující data
- Importuje a exportuje své rodokmeny do různých datových formátů a do jiných webových aplikací
- Může si zobrazit geografické údaje uvedené u osob ze zvoleného stromu v mapě a také zobrazuje chronologický pohyb osob na mapě

1.6.3.2 Popisy případů užití

Správa a vytvoření uživatelského účtu

- **Registrace uživatelského účtu** – uživatel zadá nové přihlašovací údaje pro účet do formuláře.
- **Změnit údaje o uživatelském účtu** – Uživatel zadá nové údaje do formuláře. Může měnit pouze údaje u svého účtu, k jiným účtům nemá přístup.
- **Změnit hlavní rodokmen** - uživatel zobrazí všechny jeho vložené rodokmeny zadané v systému. Vybere si rodokmen, který chce nastavit jako hlavní. Tento rodokmen se mu bude zobrazovat při vykreslování grafu rodokmenu a v rámci tohoto rodokmenu bude moci přidávat, odebírat a upravovat jeho osoby. Uživatel může zvolit vždy jen jeden hlavní rodokmen, a to jen takový rodokmen, který sám vytvořil.
- **Smazání uživatelského účtu** - uživatel smaže trvale svůj registrovaný uživatelský účet, se všemi daty o vložených rodokmenech a osobách v rodokmenech. Uživatel může smazat pouze svůj účet, k jiným účtům nemá přístup.
- **Přihlášení uživatele** – uživatel se přihlásí do webové aplikace
- **Odhlášení uživatele** – uživatel se odhlásí z webové aplikace

Správa rodokmenů

- **Zobrazit seznam rodokmenů** – uživatel si zobrazí všechny své uložené rodokmeny v systému
- **Vytvořit nový rodokmen** – uživatel zobrazí všechny své vložené rodokmeny zadané v systému. Uživatel přidá nový rodokmen ke svému účtu a do systému zadáním jména nového rodokmenu do formuláře. Uživatel může přidávat rodokmeny pouze ke svému účtu.
- **Změnit název rodokmenu** – uživatel zobrazí všechny své vložené rodokmeny zadané v systému. Vybere si rodokmen, u kterého chce změnit název a nový název vyplní do zobrazeného formuláře. Uživatel může měnit nastavení a údaje pouze u svých rodokmenů.
- **Změnit viditelnost rodokmenu** - uživatel zobrazí všechny své vložené rodokmeny zadané v systému. Vybere si rodokmen, u kterého chce změnit viditelnost. Viditelnost se volí mezi veřejnou a soukromou. Osoby v soukromém rodokmenu vidí pouze uživatel, který rodokmen vytvořil. Osoby ve veřejném rodokmenu mohou vidět všichni registrovaní uživatelé. Uživatel může měnit viditelnost pouze svých stromů.
- **Smazat rodokmen** - uživatel zobrazí všechny své vložené rodokmeny zadané v systému. Vybere si rodokmen, který chce odebrat ze systému. Rodokmen bude trvale odstraněn i se všemi osobami a jejich informacemi v něm. Uživatel může mazat pouze své rodokmeny.

Následující případy užití již kopírují scénář průběhu již popsanych případů užití, proto je popíšeme velmi stručně.

Správa osob

- **Zobrazit seznam všech osob** - uživatel si zobrazí seznam všech osob v rookmenech
- **Vložit novou osobu do rodokmenu** - uživatel vloží novou osobu do rodokmenu/systemu
- **Přidat, zobrazit a upravit údaje o osobě** - uživatel přidá, zobrazí nebo upraví informace o osobě do systému
- **Odstranit osobu** - uživatel odstraní osobu ze systému

Správa souborů

- **Vložit, zobrazit a upravit soubor** - uživatel vloží, zobrazí nebo upraví soubor
- **Stáhnout soubor** - uživatel si stáhne soubor ze serveru
- **Odstranit soubor** - uživatel odstraní soubor ze serveru i ze systému
- **Přidat/odebrat spojení osoby se souborem** - uživatel odstraní nebo přidá spojení mezi osobou a souborem

Zálohování a převod dat mezi formáty

- **Importovat/Exportovat rodokmeny do souborů GEDCOM, XML** - uživatel exportuje nebo importuje genealogická data z/do aplikace
- **Import dat z aplikace Geni** - uživatel se přihlásí ke svému účtu Geni a systém importuje data z aplikace do systému

Práce s mapou

- **zobrazení údajů z rodokmenu do mapy** - uživatel si zobrazí všechna místa u svých osob v rodokmenu na mapě
- **zobrazit a uložit graf rodokmenu** - uživatel si chronologicky zobrazí trasu pohybu všechny svých osob z rodokmenu na mapě

Návrh

2.1 Použité technologie

2.1.1 REST (Representational state transfer)

REST je architektura navržená pro distribuovaná prostředí. Jednotlivé části programu mohou tedy běžet na různých strojích a přitom spolu vzájemně komunikovat přes síť. Komunikace probíhá voláním standardních HTTP metod (GET, POST, PUT, DELETE). Architektura je navržená pro jednoduchý a uniformní přístup ke zdrojům. REST je datově orientován, proto mohou těmito zdroji být jak konkrétní data ze serveru, tak i stavy aplikace, pokud mohou být popsány konkrétními daty. REST nabývá na významu a stává se spolu s JSON defacto standardem pro API webových služeb. Jeho rozšíření napomáhá i technika AJAX, které REST vychází vstříc. Moderní frameworky pro vývoj serverové strany aplikace pomáhají vytváření REST rozhraní tím, že dokáží nadefinovat patřičné procedury pro všechny potřebné metody, a vytvoření RESTové služby se tak stává snadnějším.

Základní principy REST:

- Jednotlivé zdroje mají unikátní identifikátor, kterým jsou v požadavku jednoznačně identifikovány
- HATEOAS (Hypermedia as the Engine of Application State), Hypermedia jako stav aplikace) – stav aplikace je určen pomocí URL. Klient provádí přechody mezi stavy pouze pomocí akcí, které jsou serverem dynamicky rozpoznány jako hypermedia. S výjimkou jednoduchých vstupních bodů do aplikace klient nepředpokládá, že je k dispozici konkrétní operace pro konkrétní zdroj mimo těch, které jsou popsány v odpovědi od serveru.
- Definuje jednotný přístup pro získání a manipulaci se zdrojem v podobě čtyř operací CRUD (Create, Read, Update, Delete).

- Zdroje mohou mít různé reprezentace (XML, HTML, JSON, SVG, PDF), klient nepracuje přímo se zdrojem, ale s jeho reprezentací. Každá zpráva obsahuje informace o tom, jak má být zpráva zpracována.

2.1.1.1 Způsoby autorizace

- **OAuth 1.0:** OAuth je otevřený standard pro registraci. Tento standard je používán jako způsob přihlášení uživatele internetu na webové stránky třetích stran pomocí svého účtu (např. Microsoft, Google, Facebook nebo Twitter), aniž by odhalil své heslo. Obecně platí, že OAuth poskytuje klientům bezpečný přístup ke zdrojům serveru jménem vlastníka zdrojů. Stanovuje postup pro vlastníky zdrojů k autorizaci přístupu třetích stran do svých serverů, aniž by sdílel své pověření. Je navržen speciálně pro práci s Hypertext Transfer Protocol (HTTP). OAuth umožňuje přístup s pomocí tokenů, které jsou vydány pro klienty třetích stran prostřednictvím autorizačního serveru, se souhlasem vlastníka zdrojů. Třetí strana poté použije autorizační token pro přístup k chráněným prostředkům. Specifikace OAuth core 1.0 (označovaná také jako RFC 5849) byla dokončena v dubnu 2010. Poskytuje řešení pro zabezpečení webových API, aniž by uživatelé sdíleli svá uživatelská jména a hesla. [10]
- **OAuth 2.0:** OAuth 2.0 (RFC 6749) byl zveřejněn v říjnu 2012. Je to autorizační protokol, který se stal v podstatě standardem pro zabezpečení RESTových webových služeb. Umožňuje podrobně vymezit pravomoci jednotlivých klientských aplikací a podrobně sledovat využívání přidělených privilegií. Klientská aplikace se může autentizovat jak sama za sebe, tak i identitou jejího uživatele, který k tomu dá výslovný souhlas. Díky tomu lze, bez rizika narušení ochrany osobních údajů, umožnit přístup k citlivým datům uživatelů. Zároveň je principiálně jednoduchý, takže se snadno integruje. OAuth 2.0 není zpětně kompatibilní s OAuth 1.0. [11]

2.1.2 Jersey

Jersey RESTful Web Services framework je jedna z nejpoužívanějších referenčních implementací pro vývoj i produkci. Jedná se o open-source framework pro jednoduchý vývoj RESTových webových služeb v Javě. Poskytuje podporu pro JAX-RS API a slouží jako referenční implementace JSX-RS. Jersey také poskytuje své vlastní API, které rozšiřuje balíček JAX-RS o doplňující funkce a utility pro další zjednodušení RESTových služeb a vývoj klienta. Také poskytuje několik rozhraní pro poskytovatele služeb, takže vývojáři mohou rozšířit Jersey tak, aby vyhovovalo jejich potřebám. Poskytuje knihovnu pro klienty, která má pomoci s komunikací se serverem a k testování RESTových služeb.

Jelikož je tato knihovna generická, může spolupracovat s jakoukoliv webovou službou se základem ať HTTP nebo HTTPS.

Jersey obsahuje 3 hlavní komponenty:

- Jádru serveru: poskytováním anotací a API se standardem JSP 311, můžeme RESTové webové služby vytvářet velmi jednoduchým a intuitivním způsobem.
- Jádru klienta: Jersey Client API pomáhá jednoduše komunikovat s REST službami
- Integrovaný modul: Díky obsaženým knihovnám je možná snadná integrace dalších technologií jako Spring nebo Guice.

Cíle projektu Jersey se dají shrnout do několika základních bodů:

- Sledovat JAX-RS API a poskytovat pravidelné vydávání nových verzí projektu pro produkci kvalitních referenčních implementací, dodávaných s projektem GlassFish
- Poskytnout API k rozšíření Jersey a budování komunity uživatelů a vývojářů
- Pomoc jednoduše vytvořit RESTové webové služby využívající JVM (Java Virtual Machine)

2.1.3 JAX-RS (Java API for RESTful Web Services)

Java API pro REST webové služby (JAX-RS) je Java programovací jazyk API, který poskytuje podporu při vytváření webových služeb v souladu s Representational State Transfer (REST) architektonickým vzorem. JAX-RS používá anotace, které byly zavedeny v Java SE 5, zjednodušují vývoj a nasazení klientů webových služeb a koncových bodů. V lednu 2011 JCP tvořil JSR 339 expertní skupinu pro práci na JAX-RS 2.0. Hlavní cíle jsou (mimo jiné) společné rozhraní API klienta a podpora pro HyperMedia pro HATEOAS principy z REST. V květnu 2013 se dosáhlo konečné fáze Release.

2.1.4 JAXB (Java Architecture for XML Binding)

XML a Java technologie jsou uznávány jako ideální stavební bloky pro vývoj webových služeb a aplikací, které přistupují k webovým službám. Často se používají pro komunikaci mezi aplikacemi na různých operačních systémech. JAXB představuje výhodný framework pro zpracování XML dokumentů, poskytující značné výhody oproti starším metodám jako je například pomocí

DOM (Document Object Model). Při použití DOM vytvoří parser strom objektů, které reprezentují obsah a organizaci dat v dokumentu. Aplikace se pak může navigovat skrz strom, aby dostala požadovaná data. U tohoto modelu jsou však objekty pouze jediného typu s jednotlivými uzly objektu obsahující element, atribut, atd. Jejich hodnoty jsou zde vždy poskytovány jako řetězce. Oproti tomu zpracování XML dokumentu vhodnou JAXB metodou také vede ke stromu objektů, avšak s významným rozdílem, že uzly tohoto stromu odpovídají XML prvkům, které obsahují atributy a obsah jako proměnné a odkazují se na podřízené prvky pomocí referencí na objekt. JAXB tedy usnadňuje přístup k XML dokumentům v Javě a umožňuje vývojářům mapovat Java třídy na XML. Poskytuje dvě hlavní funkce, vytvoření XML formátu z Java objektu a naopak. JAXB povoluje ukládání a načítání dat v paměti v jakémkoliv XML formátu, bez nutnosti implementace specifické sady pro převod konkrétního XML.

2.1.5 RMI (Remote Method Invocation)

Java remote method invocation (Java RMI) je technologie programovacího jazyka Java, která umožňuje z jednoho virtuálního stroje (JVM) volat metody objektů na jiném virtuálním stroji. Tento stroj zpravidla běží na jiném počítači (hostu). Výhodou RMI je, že programátor používá vzdálený objekt, jako by byl místní. RMI se skládá ze tří vrstev, Stub na straně klienta, vrstvy Remote reference a vrstvy Transport na straně serveru. Stub je vrstvou mezi aplikací klienta a vzdálenou referencí. Je to něco jako zástupce vzdáleného objektu. Když je z klienta zavolána metoda vzdáleného objektu, tak Stub vytváří blok informací. Tento blok obsahuje identifikátor vzdáleného objektu, číslo metody, která se má zavolat, parametry zabalené do marshall streamu, tj zabalené tak, aby se daly posílat po síti. RMI podporuje dynamické stahování tříd. Pokud klient dostane jako návratovou hodnotu objekt třídy, pro kterou nemá potřebnou definici, umí si ji stáhnout ze vzdáleného FTP nebo HTTP serveru. Nezbytné URL odkazující na FTP/HTTP server je posláno společně s daným objektem.

2.1.6 HTML5 (HyperText Markup Language 5)

HTML5 je značkovací jazyk používaný pro strukturování a prezentaci obsahu webových stránek. Jedná se o pátý stupeň standardu HTML od počátku webové sítě. Cílem, kterého se snaží HTML5 dosáhnout, je zlepšení podpory jazyka pro multimédia a zároveň zachování jednoduché struktury a konzistence dat pro bezproblémové zpracování na různých zařízeních. Zaměřilo se na sémantiku webových stránek a převážně na zvýšení přehlednosti zdrojového kódu. Většina stránek je dnes tvořena obvyklými částmi, jako je záhlaví, různé sloupce a zápatí. Také se jedná o pokus definovat jediný značkovací jazyk, který může být sepsán jak podle standardu HTML, tak podle standardu

XHTML. Pro mnoho navržených funkcí HTML5 byly brány v úvahu méně výkonná zařízení, jakými jsou třeba smartphony nebo tablety.

Mezi hlavní změny oproti předchozím standardům patří:

- Nové datové typy a možnosti polí ve formulářích: dates, times, email, url, search, number, range, tel, color, required, pattern, min, max, minlength, maxlength
- Globální atributy, které mohou být použity na všechny prvky: id, tabindex, hidden, data-*
- Nové elementy, hlavně: canvas, header, footer, nav, section, data, data-list, article
- Možnost použít SVG a MathML v text/-html

2.1.7 Bootstrap

Bootstrap je v současnosti pravděpodobně nejpopulárnější framework pro tvorbu responsivního webového obsahu na světě. Jedná se o elegantní a intuitivní framework pro rychlý a snadný vývoj webových projektů. Obsahuje HTML a CSS šablony pro snadnou tvorbu formulářů, tlačítek, tabulek, navigačních prvků a mnoho dalších, stejně jako množství volitelných JavaScript pluginů.

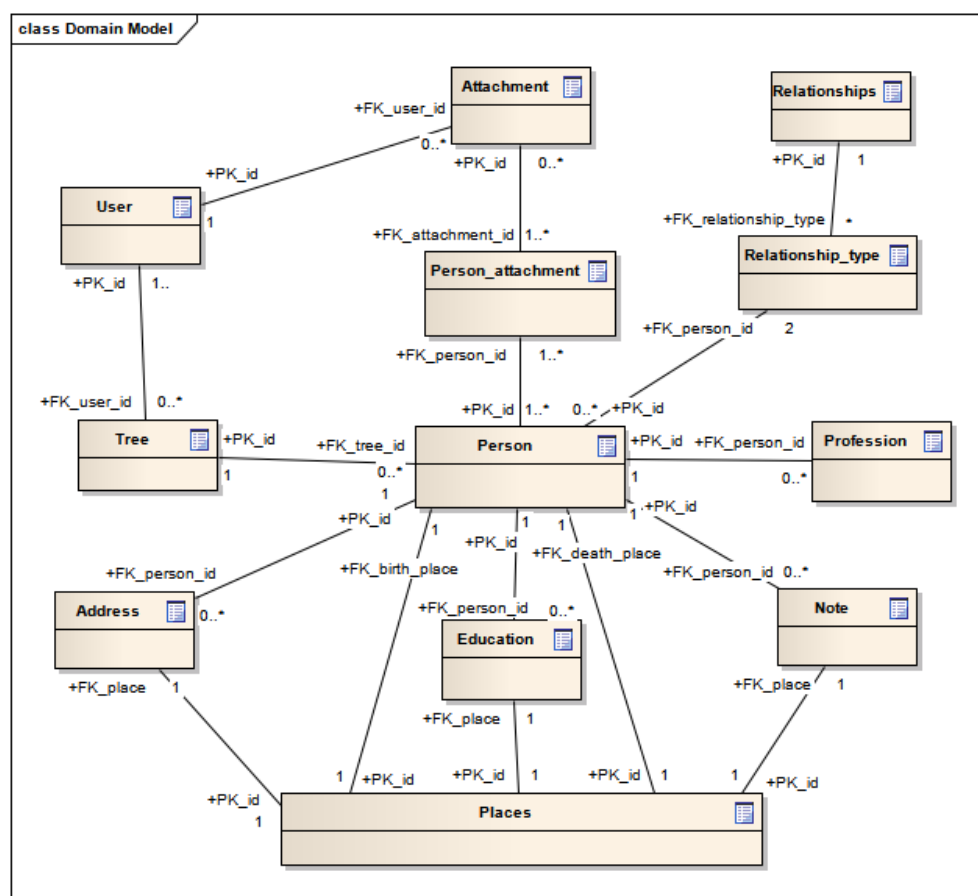
2.1.8 Raphaël

Raphaël je malá JavaScriptová knihovna, usnadňující práci s vektorovou grafikou na webu. Pomocí knihovny se dá snadno dosáhnout ořezávání obrázků, vytváření grafů i dalších pokročilejších možností zobrazení jako je tvorba animací. Jako základ pro vytváření grafických prvků používá SVG doporučené W3C a VML. Každý grafický objekt, který knihovna vytvoří, je DOM object, takže se na něj mohou nadále vázat další JavaScriptové funkce a manipulátory. Cílem knihovny je poskytnout adaptér, který bude vykreslovat vektorovou grafiku napříč různými prohlížeči. Knihovna umožňuje vykreslovat základní tvary a objekty jakými jsou třeba obdélník, kruh, obrázek nebo text. Tyto objekty podporují značné množství vlastností, které pomáhají formovat například barvu, otočení, obsah objektu. Objekty se vykreslují na papír, který je reprezentován HTML elementem canvas. Všechny objekty na tomto papíře pak “žijí”. [12]

2.1.9 MySQL

MySQL je relační databáze typu DBMS založená na jazyce SQL. Používá dialekt jazyka SQL s některými rozšířeními. Pracuje na principu klient-server.

2. NÁVRH



Obrázek 2.1: Hrubý návrh databáze

Je v zásadě jedno, jestli klientská i serverová aplikace běží na stejném počítači anebo na různých strojích. Je multiplatformní, to znamená, že podporuje instalaci na různé operační systémy (MS Window, Linux a další). MySQL je nabízeno ve dvojnásobném licencování. V bezplatné licenci GPL (open source) a pod komerční placenou licenci.

2.2 Databáze

Databáze je navržena na základě poznatků získaných při řešení současných řešení, průzkumu a požadavků na systém. Hrubý návrh schématu databáze a vztahy mezi jednotlivými entitami je přiblížen na obrázku 2.1.

2.2.1 Popis entit

User

Obsahuje informace o uživateli webové aplikace včetně nastavení, který strom se má nastavovat jako hlavní, to je strom, který se bude primárně načítat pro zobrazení. Dále ukládá identifikátor hlavní osoby stromu, tato osoba pak bude pro rodokmen primární bod, od kterého se bude rodokmen vykreslovat. Pro každou osobu v rodokmenu existuje jiná relativní verze rodokmenu, vztahená k této osobě.

Tree Každý uživatel může vlastnit několik rodokmenů. Informace o těchto rodokmenech jsou uloženy v entitě Tree. Ta zároveň obsahuje informaci o tom, zda je rodokmen soukromý. Pokud soukromý není, mohou osoby v tomto rodokmenu být zobrazovány ostatním uživatelům. Každý rodokmen může patřit jednému uživateli.

Person Základní entita v rodokmenu se základními a jedinečnými informacemi o osobě. Stejně jako může každý rodokmen patřit k jednomu uživateli, tak každá osoba může patřit do jednoho rodokmenu.

Address, Education, Notes a Profession

Každá z entit obsahuje informace o Bydlišti, Vzdělání, Poznámkách a Zaměstnání osoby. Těchto entit může mít každá osoba několik.

Place Uchovává geolokační informace o místě. To se váže k bydlišti, vzdělání a poznámce osoby. Dále také k osobě samotné, konkrétně k místu narození a místu úmrtí.

Relationship obsahuje seznam konkrétních typů vztahů, které mezi sebou mohou mít dvě osoby, a to jak vztahy přímé (rodiče, dítě, partner, ...), tak nepřímé (kmotr, svědek, ...)

Relationship-type Má uložený seznam vztahů mezi osobami. Jedná se o informaci, které konkrétní dvě osoby mezi sebou mají vztah a jaký je konkrétní druh vztahu mezi těmito osobami. Druh vztahu musí být jeden z definovaných v entitě Relationship.

Attachment uchovává odkazy na uložené soubory. Tyto soubory mohou být vázány pouze k uživateli nebo mohou mít vazbu také ke konkrétním osobám. Tato vazba je pak uložena v Attachment_person.

2.3 Návrh REST API

Při návrhu REST API musíme mít stále na vědomí, že jednotlivé zdroje jsou charakterizovány unikátním globálním identifikátorem. A také skutečnost, že se zdroje mohou vracet v různých formátech a v těchto formátech s nimi může být také manipulováno. Globálním identifikátorem je v našem případě URL zdroje, které je navrženo podle modelu databáze. REST API může klientovi vracet data ze zdroje jako seznam záznamů nebo jako jeden konkrétní záznam identifikovaný dle jeho id v dané tabulce v databázi.

V aplikaci pracujeme s několika různými třídami. To může být často nepřehledné, zejména pokud jsou třídy z pohledu klienta složité. V implementaci REST API budeme proto používat návrhový vzor Facade, který nám umožní zjednodušit komunikaci mezi klientem a REST API. Návrhový vzor Facade spadá pod strukturální vzory, protože přidává rozhraní, aby skryl složitost systému ve vztahu ke klientovi. Struktura REST API a jeho jednotlivé třídy jsou potom skryté za fasádu, což je jediná třída se kterou komunikuje klient, aniž by věděl jak složitý systém obsluhuje jeho požadavky. Facade potom podle typu požadavku klienta tyto přeposílá na příslušné třídy uvnitř REST API, které umí daný požadavek zpracovat. Výhodou vzoru je také řešení jednotlivých komponent v REST API, aniž by to mělo vliv na komunikaci s klientem. Pokud mezi nimi předáváme nějakou instanci, řeší se to za fasádou. Toto řešení nám dále umožní výměnu či změnu komponent za fasádou beze změny jejího vnějšího rozhraní, a vznikne možnost k funkcím jednotlivých komponent přidat i jednoduchou řídicí logiku. [13]

Klient předá požadavek na fasádu REST API, ta na základě URL vybere třídu Session bean a zároveň podle případných dalších částí URL a metody HTTP, vybere metodu této třídy, která zajistí zpracování daného požadavku.

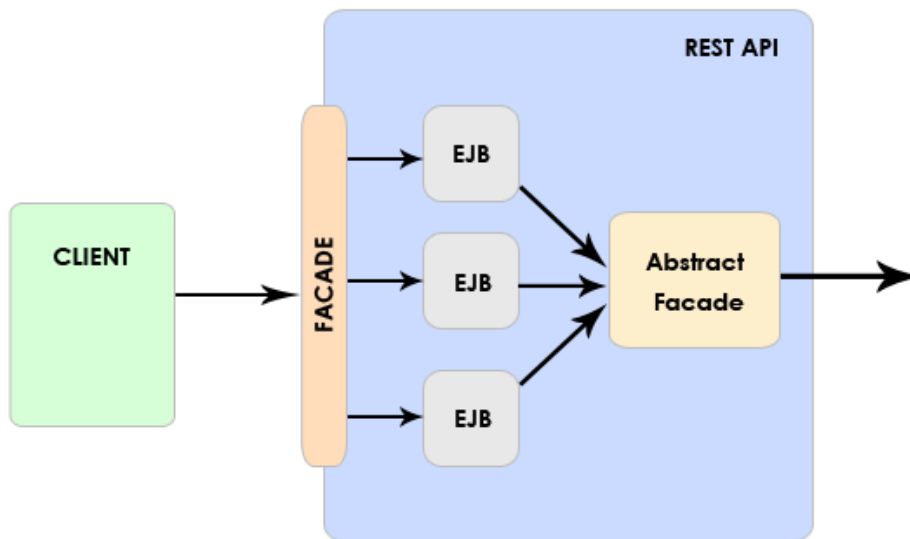
Požadavek od klienta má strukturu :

Req_Method Req_URI HTTP-Version , kde:

- HTTP-Version představuje verzi HTTP protokolu
- Req_Method je metoda HTTP požadavku, typu GET, POST, PUT nebo DELETE
- Req_URI obsahuje další údaje, podle nichž fasáda (Facade) určuje, která metoda z které Session bean (EJB) bude požadavek vyřizovat a jak.

HTTP požadavek může obsahovat URI s jedním nebo více parametry, a případně i data určená k zápisu do zdroje dat. Metoda přijímá parametr pouze v případě, že ho implementuje. Pokud fasáda metodu definovanou požadavkem nenajde, vrátí klientovi odpověď se stavovým kódem o neúspěšném zpracování požadavku ze strany klienta. V tomto případě vrátí kód "405 Method Not Allowed". Fasáda zajistí konverzi dat z formátu, ve kterém je klient zaslal do formátu Java Object podle volané Session Bean. Konverze mezi formátem, který klient zaslal a Java Object musí být implementován. V našem případě je implementovaná pomocí Jax-SB v entity třídě. Formáty dat, ve kterých může klient svá data posílat, jsou pouze XML a JSON.

Session Bean je tedy rozšířením generické abstraktní třídy Abstract Facade. Pro každou Session Bean musí být entity třída. Tím vzniká opakování



Obrázek 2.2: Návrh REST API pomocí fasády

CRUD logiky v jednotlivých Session Bean. Abychom se tomuto opakování vyhnuli, vkládá se CRUD logika do generické abstraktní fasády (Abstract Facade). To znamená že, všechny metody, které jsou společné pro všechny Session Bean jsou implementovány v abstraktní fasádě. Příslušná metoda Session Bean obdržený požadavek vyhodnotí. Pokud se jedná o jednoduchý HTTP požadavek, Session Bean předá požadavek včetně parametrů a dat abstraktní fasádě. Abstraktní fasáda podle metody volání, a podle toho která Session Bean jí požadavek předala, pozná s jakým zdrojem má pracovat, a z metody volání ví, kterou CRUD operaci má požadovat. Na základě těchto poznatků pošle na JPA příkaz na požadovanou operaci s konkrétními daty a s jednoznačně definovaným zdrojem dat. JPA zajistí vytvoření příkazu v jazyce, kterému datový zdroj rozumí, a přepoše tento příkaz k provedení.

Pokud bylo provedení dotazu na datový zdroj úspěšné, obdrží JPA data, pokud byla příkazem požadována. Data jsou ve formátu datového zdroje. JPA provede pomocí příslušné anotované entity třídy konverzi na odpovídající formát Java Object. Ten je prostřednictvím abstraktní třídy předán volající Session Bean s návratovým kódem. Ten podle požadavku klienta provede konverzi dat do formátu XML nebo JSON. V případě že, provedení dotazu je neúspěšné, nastane výjimka, kterou JPA zachytí na základě chybového hlášení ze zdroje dat. Tato výjimka se postupně přenese až do volající Session Bean. Metoda Session Bean obsahuje strukturu pro ošetření výjimek. Pokud je výjimka zachycena, Session Bean vybere odpovídající stavový kód pro konkrétní výjimku. Session Bean předá prostřednictvím fasády klientovi odpověď obsa-

hující stavový kód reprezentující úspěšné nebo neúspěšné provedení požadavku a případná data, o která klient žádal. Pokud se jedná o složitější HTTP požadavek (např. vyžaduje data na základě více parametrů), vytvoří v tomto případě Session Bean vlastní dotaz podle Named Query. Named Query je statický předem definovaný a neměnitelný řetězec reprezentující dotaz. Tento řetězec reprezentuje strukturu dotazu, která umožňuje dynamicky vkládat do předem definovaných míst různé parametry podle potřeby Session Bean. Všechny Named Query jsou definovány v odpovídající entity třídě. Pak Session Bean přepośle úplný dotaz přímo na JPA. JPA již tento sestavený dotaz dále nemodifikuje a rovnou ho přeposílá na datový zdroj.

2.3.1 Autorizace požadavku

Pro úspěšné používání REST API je nutná autorizace. To vyžaduje mít zaregistrovaný účet v databázi. Sama autorizace probíhá na REST API. Uživatel musí pomocí POST požadavku zaslat své přihlašovací údaje. Pokud uživatel má založený účet a vyplnil správné heslo, zašle se mu zpátky autorizační token, který pak musí posílat v hlavičce každého požadavku, který vyžaduje autorizovaný přístup. Při zaslání požadavku na zdroj, který vyžaduje autorizace, autorizační filtr zjišťuje, jestli hlavička obsahuje token. Pokud ano, zjistí, komu token patří a jestli je platný. Pokud je platný, dovolí uživateli přístup na REST API. Zde se dále kontroluje, jestli tento konkrétní uživatel má právo k datům přistupovat nebo je opravovat.

2.3.2 Stavové kódy

Během komunikace mezi serverem (REST API) a klientem vrací REST API stavové kódy indikující úspěch či neúspěch zaslání požadavku. Obecně stavové kódy formátu 2xx značí úspěšně zpracované požadavky, kódy 4xx neúspěšně zpracované požadavky ze strany klienta a kódy 5xx neúspěšně zpracované požadavky ze strany serveru. Níže popíšeme, jaké kódy a za jakých situací navržené REST API vrací odpovědi.

- **200 OK:** Požadavek proběhl v pořádku, REST API vrátí požadovaná data.
- **201 Created:** Požadavek POST proběhl v pořádku, do systému byla vložena nová data, zároveň REST API v odpovědi vrátí odkaz na tato nově vytvořená data.
- **204 No Content:** Požadavek proběhl v pořádku, ale v odpovědi se nevrací žádná data, tento stav REST API vrací při úspěšné úpravě existujících dat nebo při smazání existujících dat.
- **400 Bad Request:** Vrací REST API, pokud jsou data zaslána v požadavku ve špatném formátu.

- **401 Unauthorized:** REST API vrátí, pokud klient není ověřen. Požadavek vyžaduje pro vrácení dat nebo pro jejich modifikaci autentifikaci uživatele. Hlavička požadavku by měla obsahovat autentifikační údaje pro přístup k datům.
- **403 Forbidden:** Klient je ověřen, avšak nemá povolený přístup k daným datům.
- **404 Not Found:** Požadovaná data nebyla nalezena.
- **405 Method Not Allowed:** Zdroj dat není pro tuto metodu povolen.
- **409 Conflict:** Vrací REST API pouze při vkládání nového uživatele se stejnou přezdívkou nebo e-mailem jako má již jiný uživatel v systému.
- **415 Unsupported Media Type:** Klient zaslal požadavek na data ve formátu, který REST API nepodporuje.
- **500 Internal Server Error:** Požadavek byl vykonán správně, ale nastala chyba při běhu serveru.

2.4 Architektura aplikace

2.4.1 Klientská strana

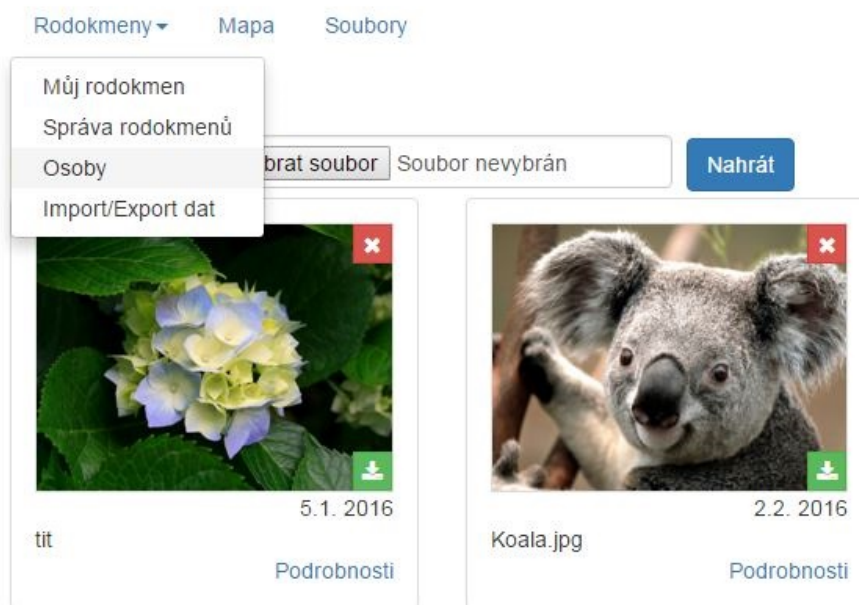
Klientská strana je reprezentována webovou stránkou zobrazenou uživateli v prohlížeči. Zobrazuje uživateli data získaná od serveru v textové nebo grafické podobě. Tato stránka obsahuje statickou část HTML a dynamickou část JavaScript. JavaScript využívá technologii jQuery pro jednodušší přístup a lepší manipulaci DOM objektů. Na základě podnětu od uživatele JavaScript obsluhuje požadavky na změnu elementů na stránce. Ke komunikaci mezi klientskou a serverovou stranou aplikace používá JavaScript technologii AJAX.2.3

2.4.2 Serverová strana

Serverovou stranu aplikace tvoří 2 spolupracující moduly. Oba tyto moduly umí komunikovat s klientskou stranou. Avšak každý z nich obsluhuje jiný požadavek ze strany klienta buď na změnu dat na stránce, nebo změnu celé webové stránky. Server obsahuje 2 servlety. První obsluhuje přihlášení do webové aplikace. Druhý servlet zjišťuje požadavky klientské strany na změnu zobrazení celé webové stránky nebo její znovunačtení. Tento servlet také vyřizuje složitější požadavky od klienta, které REST API neumí zpracovat. Tyto požadavky servlet obdrží pomocí metody POST. Mezi ně patří například stažení souboru nebo vložení souboru na server.

REST API je druhým modulem serverové strany, který reaguje na požadavky servletu nebo klientské strany. Tento modul běží na serveru i samostatně

FamilyTree

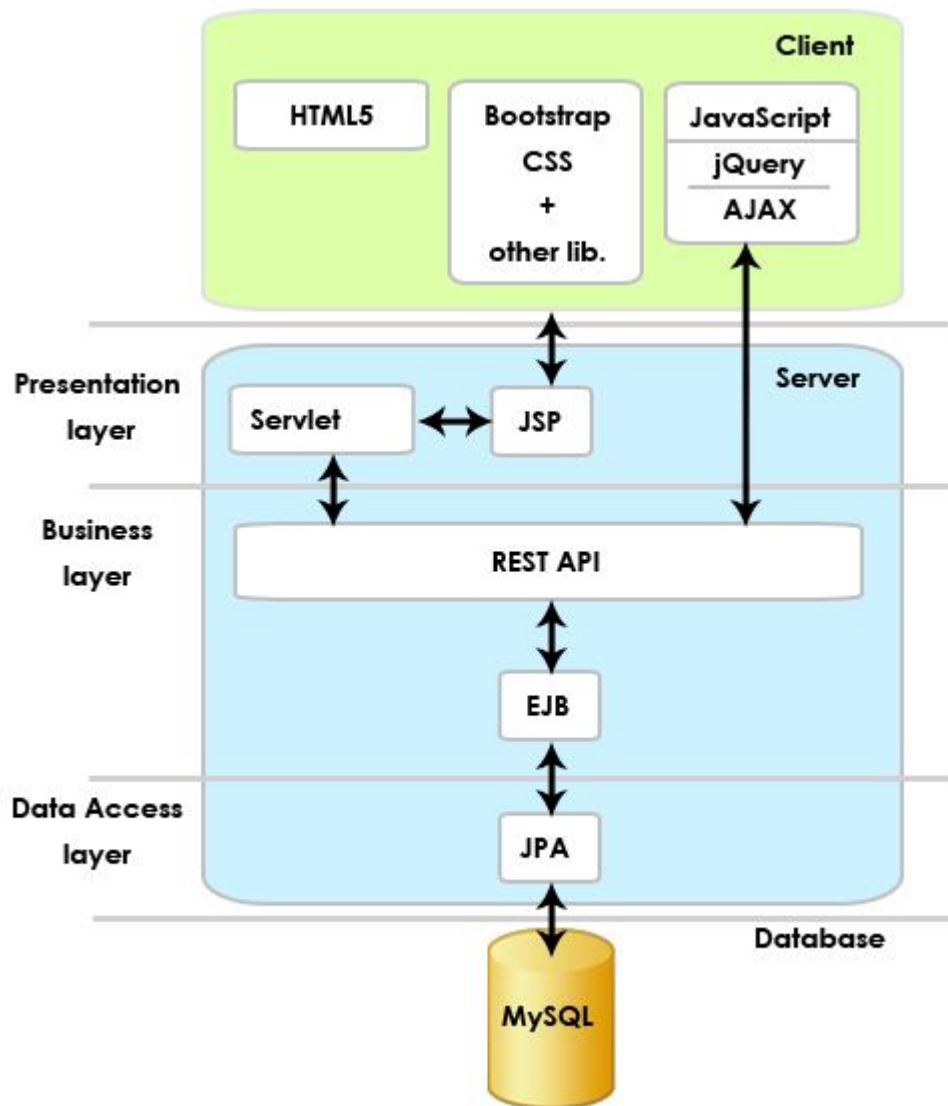


Obrázek 2.3: Klientská část aplikace - práce se soubory

a umožňuje tedy přistupovat k datům z databáze i mimo prostředí webové aplikace. Stejně tak mohou REST API využívat i aplikace třetích stran. Na základě vytvořeného požadavku zpracovává dotazy na datové úložiště a vrací konkrétní sadu dat. Tuto sadu dat vrací servletu nebo klientské straně, podle toho, která strana požadavek inicializovala. V případě servletu odešle servlet na klientskou stranu data přijatá od REST API na adresu požadované webové stránky. Při přijetí dat klientskou stranou jsou data zpracována JavaScriptem na straně klienta. [14]

2.4.3 Komunikace klient-server

Jak bylo naznačeno, komunikace v aplikaci probíhá při načítání stránky nebo při komplikovanější funkcionalitě přes Java Servlet a následně přes REST API. V jiných případech klient komunikuje přímo s REST API a modifikuje tak data v databázi. Komunikace probíhá na několika vrstvách podle obrázku 2.4.



Obrázek 2.4: Návrh architektury aplikace

2.4.4 Prezentační vrstva

JEE specifikace nebo technologie v této skupině obdrží požadavek z webového serveru a vrátí zpět odpověď, typicky v HTML formátu. Toto však není nutnost, v odpovědi se mohou vrátit i jen data z prezentační vrstvy, například ve formátu JSON nebo XML, které pomocí volání AJAXu obnoví pouze část webové stránky místo překreslování celé HTML struktury webové stránky. Třídy v prezentační vrstvě jsou vykonávány převážně ve webovém kontejneru. V této vrstvě se nachází JSP a Servlety.

2.4.4.1 Java Servlet

Servlety jsou moduly na straně serveru, typicky zodpovědné za zpracování požadavku a odeslání odpovědi zpět webové aplikaci. Servlety jsou užitečné při zacházení s požadavky, které negenerují odpověď jako velké HTML. Typicky se používají jako controllery ve frameworku MVC (Model View Controller) pro přeposílání nebo přesměrování requestu na jinou adresu nebo pro generování odpovědi v jiném formátu než je HTML, například PDF. K vygenerování HTML odpovědi ze servletu je potřeba vložit kód HTML jako řetězec v Javě. Z tohoto důvodu není Servlet vhodný pro generování velkých HTML odpovědí.

2.4.4.2 JSP(Java Servlet Pages)

Stejně jako servlety, JSP jsou také moduly na serverové straně používané pro zpracování webových požadavků. JSP stránky jsou vhodné pro zpracování odpovědí, které generují odpovědi jako velké HTML kódy. V případě JSP může být Java kód nebo JSP tag vepsán do HTML kódu jako HTML tagy, JavaScript a CSS. Jelikož Java kód je vložen ve velkém HTML kódu, je jednodušší generovat HTML odpověď z JSP stránky.

2.4.5 Bussiness vrstva

Do bussines vrstvy se typicky zapisuje kód pro zpracování aplikační logiky webové aplikace. Požadavek na tuto vrstvu může přijít jak z prezentační vrstvy serveru, tak přímo z klientské strany aplikace nebo z jiné webové služby, třeba i z jiné aplikace. Třídy v této vrstvě jsou spouštěny v aplikační části kontejneru serveru J2EE. Najdeme zde REST API a třídy EJB.

2.4.5.1 EJB

Enterprise Java Beans jsou Java třídy, do kterých můžeme zapisovat aplikační logiku. Přestože není použití EJB doporučováno k zápisu aplikační logiky, poskytují mnoho služeb, které jsou zásadní při psaní enterprise aplikace. Těmito jsou bezpečnost, správa transakcí, vyhledávání komponentů, atd. EJB mohou být distribuovány skrze několik serverů a nechávají tak aplikační kontejner

(EJB kontejner), aby se postaral o vyhledávání a sdružování komponentů. Toto může pomoci se škálováním aplikace.

EJB rozdělujeme do dvou typů:

- **Session Bean:** jsou volány přímo klientem nebo třetími aplikacemi
- **Message driven beans:** jsou volány v odpovědi na Java Messaging Service(JMS) události. JMS a Message driven beans mohou být použity pro práci s asynchronními dotazy. V typickém scénáři pro zpracování asynchronního požadavku klient vloží požadavek do fronty zpráv nebo do tématu a nečeká na okamžitou odpověď. Server aplikace obdrží zprávu v požadavku buď přímo pomocí JMS API nebo použitím MDB. Tím se zpracuje požadavek a může se vložit odpověď do jiné fronty nebo tématu, kterému bude klient naslouchat a získá tak odpověď.

2.4.6 Vrstva pro přístup k datům

API v této vrstvě se používají pro interakci s externími systémy v enterprise architektuře. Většina aplikací bude potřebovat přístup k databázi nebo jinému datovému úložišti. API, které tyto činnosti zajišťují, se nacházejí právě v této vrstvě.

2.4.6.1 JPA(Java Persistence API)

Jedním z problémů používání JDBC API přímo je, že je nutné neustále mapovat data mezi Java objekty a daty ve sloupcích a řádcích v relační databázi. Frameworky jako Spring a Hibernate udělaly tento proces mnohem jednodušší použitím konceptu známého jako Object Relationship Mapping(ORM). ORM je začleněno v J2EE ve formě JPA(Java Persistence API). JPA poskytuje flexibilitu při mapování objektů na tabulku v relační databázi a vykonává dotazy s nebo bez použití SQL(Structured query language). Při použití těchto dotazů v JPA se jazyk nazývá Java Persistence Query Language.

Implementace

3.1 Jazyk

Java Platform, Enterprise Edition (neboli Java EE, dříve označovaná jako Java 2 Enterprise Edition nebo J2EE) je součástí platformy Java určená pro vývoj a provoz podnikových aplikací a informačních systémů. Základem pro platformu Java EE je platforma Java SE, nad ní jsou definovány součásti tvořící Java EE. J2EE je standard pro vývoj robustních, škálovatelných a bezpečných serverových systémů v Javě. Poskytuje business component framework a komponenty pro tvorbu webových aplikací a služeb, které lze snadno integrovat do stávající infrastruktury. Celosvětovým trendem vývoje informačních systémů je orientace na prostředí Java EE (Java Enterprise Edition), které spojuje výhody internetových technologií a technologií používaných v rozsáhlých podnikových systémech.

Jednotlivé součásti platformy Java EE jsou definovány pomocí dílčích specifikací, které jsou vytvářeny ve spolupráci více firem v rámci tzv. Java Community Process (JCP). Vlastní Java EE je poté definována zastřešující specifikací opět vyvíjenou v rámci JCP. Tato specifikace především fixuje konkrétní verze jednotlivých dílčích specifikací patřících do dané verze Java EE. Základem Java Enterprise Edition (Java EE) je Java Standard Edition (Java SE). Java EE však navíc poskytuje knihovny, které by měly usnadňovat vývoj komplexnějších enterprise aplikací. Java EE poskytuje navíc oproti Java SE okolo dvou desítek API, z nichž každé usnadňuje vývoj v určité oblasti. Jako příklad můžeme uvést Java Transaction API (JTA), které usnadňuje správu transakcí, Java API for XML Web Services (JAX-WS), které poskytuje podporu pro webové služby, či JavaServer Faces.

Velkou výhodou je rovněž akceptace společného standardu ze strany většiny velkých softwarových společností, a tudíž možnost přecházet mezi prostředími od různých dodavatelů bez nutnosti významných změn v aplikaci. Pro běh aplikací je pak možné použít takový aplikační server, který nejlépe vyhovuje poměru cena : požadovaný výkon. Pro méně rozsáhlé aplikace lze do-

3. IMPLEMENTACE

konce použít kombinaci operačního systému, databáze a aplikačního serveru, která je zcela zdarma.

Součástí platformy Java EE jsou především specifikace pro :

- vývoj webových aplikací - Java Servlets, Java Server Pages (JSP), Java-Server Faces (JSF)
- Contexts and Dependency Injection - vkládání závislostí
- přístup k relačním databázím - Java Persistence API
- vývoj sdílené business logiky - Enterprise Java Beans (EJB)
- přístup k legacy systémům - Java Connector Architecture (JCA)
- přístup ke zprávovému middleware - Java Messaging Services (JMS)
- komponenty zajišťující integraci webových aplikací a portálů - Portlety
- podpora technologií Web services

Hlavní přínosy J2EE:

- nezávislost na operačním systému klientů i serverů
- databázová nezávislost
- nezávislost na aplikačních serverech
- automatická podpora bezpečnostních služeb
- možnost vzdálené správy aplikací přes internet
- možnost dynamického vyvažování zátěže serverů
- snadná výměna částí aplikace bez přerušení provozu atd.

3.2 REST API

V implementaci REST API využíváme návrhový vzor Facade, který skrývá složitost systému a poskytuje uživateli rozhraní, pomocí kterého uživatel může přistupovat do systému. Tento návrhový vzor spadá pod strukturální vzory, protože přidává rozhraní systému, aby skryl jeho náročnost. V balíčku `src.rest.service` najdeme třídy které využívají tento návrhový vzor, třídu `ApplicationConfig.java`, třídu `AbstractFacade.java` a obecně třídy `*FacadeREST.java`. V balíčku `src.rest` ukádáme třídy entit reprezentující strukturu databázových tabulek.

3.2.1 Autorizace

3.2.1.1 Získání tokenu

Třídy pro autorizaci a autorizační filtr najdeme v balíčku `src.auth`. REST API point nalezneme v balíčku `src.res.service`. Pro autorizaci jsme se rozhodli zvolit se vlastní způsob autorizace. Vytvořili jsme si nový REST API identifikátor `/authenticate`, který obsahuje metodu pro přijetí POST požadavku s přihlašovacími údaji uživatele. Metoda tyto údaje ověří a pokud jsou platné, vygeneruje token.

```
public Response authenticateUser(Credentials credentials) {
    try {
        String username = credentials.getUsername();
        String password = credentials.getPassword();

        // Authenticate the user using the credentials provided
        authenticate(username, password);

        // Issue a token for the user
        String token = issueToken(username);

        // Return the token on the response
        return Response.ok(token).build();

    } catch (WebApplicationException e) {
        throw new WebApplicationException
(Response.status(Response.Status.NOT_FOUND).build());
    } catch (Exception e) {
        throw new WebApplicationException
(Response.status(Response.Status.UNAUTHORIZED).build());
    }
}
```

3.2.1.2 Vygenerování tokenu

Token se generuje na základě řetězce složeného z přihlašovacího jména uživatele a data expirace. Tento řetězec se zašifruje tajným kódem aplikace. Pro zašifrování a dešifrování tokenu používáme třídu `Encryptor` v balíčku `src.ext`.

```
public static String encrypt(String valueToEnc) {
    try {
        Key key = generateKey();
        Cipher c = Cipher.getInstance("DESede/CBC/PKCS5Padding");
        c.init(Cipher.ENCRYPT_MODE, key, iv);
```

```
        byte[] encValue = c.doFinal(valueToEnc.getBytes());
        String encryptedValue = new BASE64Encoder().encode(encValue);
        return encryptedValue;
    } catch (Exception e) {
        System.out.println(e);
        return null;
    }
}
```

3.2.1.3 Ověření tokenu

Token se pomocí třídy `Encryptor` dešifruje a ověří se, zda je stále platný. Pokud jeho platnost vypršela, musí s klient zažádat o nový token. Když nastane během ověření tokenu jakákoliv chyba, vrátí se klientovi odpověď 401 `Unauthorized`.

3.2.2 Třída `ApplicationConfig`

Je alternativou konfiguračního souboru `web.xml`. Definuje pomocí anotace `@ApplicationPath` na které relativní URI cestě k aplikaci se má REST API spouštět. Tato třída pomáhá při integraci JAX-RS s frameworky jako EJB3 nebo Spring a rozšiřuje třídu `javax.ws.rs.core.Application`. Obsahuje dvě základní konfigurační metody:

```
public Set<Class<?>> getClasses();
public void addRestResourceClasses(Set<Class<?>> resources);
```

Tato třída se využívá při vytvoření nové JAX-RS třídy. Ačkoliv JAX-RS třída obsahuje povinnou anotaci, systém ji nerozpozná jako JAX-RS, dokud není zaregistrovaná v `ApplicationConfig.java`. Tato instalace probíhá přidáním třídy jako zdroje druhou metodou uvedenou výše. Pokud je JAX-RS správně sestavena, probíhá tato instalace automaticky, proto se nedoporučuje vkládat nové zdroje ručně.

3.2.3 `AbstractFacade.java`

Protože máme více entit tříd, kde se opakují stejné CRUD operace, využíváme třídu `AbstractFacade.java`, která obsahuje metody pro CRUD operace, definované generickou třídou. Tak se vyhneme opakování stejného postupu CRUD operací u `FacadeREST` tříd a pouze očekáváme různá data.

3.2.4 Třídy `*FacadeREST`

Obsahuje session bean třídy opatřené anotacemi JAX-RS. Session bean obstarávají bussines logiku a přístup k datům systému. Všechna klientova volání

probíhají přes tyto třídy a vytváří tak funkčnost protokolu HTTP. Metody ve třídách přijímají požadavky od klienta a delegují proces pro splnění těchto požadavků. Na základě požadavku mohou zpět klientovi odeslat požadovaná data. Třídy k tomu využívají anotované metody a očekávají odpovídající funkcionálníitu.

Každá FacadeREST třída zastupuje odpovídající tabulku databáze a všechny úkony provádí relativně k této tabulce a jejím záznamům. Těchto tříd je v aplikaci mnoho a všechny mají společné základní metody. Tyto společné metody stručně popíšeme. Jedná se o metody, které využívají stejné CRUD operace a využívají tedy výše zmíněnou třídu AbstractFacade.java. Také zmíníme některé další zajímavé metody z těchto FacadeREST tříd.

Společné metody

Konkrétně budeme popisovat případ metod třídy PersonFacadeREST.java, ostatní třídy přistupují k datům stejně jen na jiném zdroji

```
public Response create(Person entity);
```

- volaná pomocí @GET, vytvoří v rodokmenu novou osobu a v odpovědi klientovi vrátí odkaz na zdroj dat této osoby

```
@Path("/{id}")
```

```
public void edit(@PathParam("id") Integer id, Person entity);
```

- volaná pomocí @PUT, upraví informace o osobě nahrazením celého objektu, vrací kód 204

```
@Path("/{id}")
```

```
public void remove(@PathParam("id") Integer id);
```

- volaná pomocí @DELETE, odebere osobu z rodokmenu včetně všech zdrojů, které jsou k ní vázány vrátí kód 204

```
@Path("/{id}")
```

```
public Person find(@PathParam("id") Integer id);
```

- volaná pomocí @GET, najde osobu podle ID a vrátí její informace klientovi

```
public List<Person> findAll();
```

- volaná pomocí @GET, vytvoří v rodokmenu novou osobu a v odpovědi klientovi vrátí odkaz na zdroj dat této osoby

3.2.5 Popis anotací JAX-RS

Součástí implementace REST API je JAX-RS zdroj, což je anotovaná třída POJO(Plain Old Java Object). POJO je tedy třída, která nutně nedodrží pravidla stavby objektu nebo frameworku. V souvislosti s JAX-RS se jedná o třídu, která poskytuje takzvané zdrojové metody, které jsou schopné zpracovat HTTP požadavky pro dané URI cesty, ke kterým se zdroj váže. Tyto třídy musí mít anotaci `@Path` a alespoň jednu metodu anotovanou také buď `@Path` nebo zdrojovou metodu označenou anotací `@GET`, `@POST`, `@PUT` nebo `@DELETE`.

Hodnota v `@Path` udává relativní cestu URI, ze které se má tato třída volat. JAX-RS podporuje práci s URI šablonami cesty, kde je proměnná vložena do URI pomocí URI syntaxe. Tyto proměnné jsou za běhu nahrazeny, aby zdroj reagoval na požadavek na základě nahrazené URI. Proměnné z URI se metodě předávají anotací `@PathParam` se jménem proměnné z cesty. Označení metod zdroje anotací `@GET`, `@PUT`, `@POST` a `@DELETE` odpovídá stejně pojmenovaným HTTP metodám. Reakce zdroje závisí na tom, na které HTTP má zdroj odpovědět.

Další důležitou anotací je `@Produce`, které se používá k určení formátu dat, v jakém má zdroj odpověď zpracovat a zaslat zpět klientovi. Pokud je třída schopna převést zdroj do více formátů, pak zvolí nejpříjemnější formát podle požadavku klienta, který v zaslaném požadavku určil, který formát je nejpříjemnější. Všechny formáty dat, do kterých může třída data zpracovat, se definují v jednom `@Produce`. Anotace, která má opačnou funkci a definuje, jaký formát dat může zdroj přijmout, je `@Consume`. Stejně jako `@Produce`, v sobě může obsahovat několik formátů, avšak oproti `@Produce` se tato anotace může vázat jak k celé třídě, tak i k jednotlivým metodám. Nakonec anotace `@Context` načte Java typy týkající se požadavku nebo odpovědi.

3.2.6 Třídy entit

Třídy entit jsou Java třídy(objekty) reprezentující tabulky z databáze. Každé tabulce v databázi odpovídá jedna třída. Tyto třídy obsahují JPA a JAXB anotace zároveň. Tyto obě anotace pohromadě zabírají mnoho místa a tvoří velké shluky anotací, avšak i tak nám přijde tento přístup lepší než definovat dvě velké části duplicitní třídy jen kvůli různým anotacím. Jak bylo zmíněno výše JAXB dokáže pomáhat generovat XML strukturu ze třídy a díky Jersey může JAXB generovat pomocí stejných anotací i JSON objekty. Díky JPA anotacím umí třída komunikovat s databází a business vrstvou a kontroluje validitu dat vzhledem k datovým typům v databázi. Díky JAXB anotacím komunikuje s facade třídama a business vrstvou. Třídy také obsahují základní metody(gettery a settery).

3.2.7 Popis použitých anotací JPA

Označením POJO třídy `@Entity` se z ní stává JPA entita. Důležitou anotací je i `@Id`, které určuje která proměnná je identifikátor tohoto objektu. Ostatní hodnoty jsou také namapovány podobným způsobem. Třídy jsou mapovány na příslušnou tabulku v databázi podle identifikátoru na primární klíč. `@Table` je definována na úrovni třídy a určuje ke které tabulce v databázi se Entita váže. Podobně se označuje `@Column`, které specifikuje kterému sloupci v tabulce proměnná ve třídě odpovídá. Pokud chceme aby se id entity generovalo samo použijeme `@GeneratedValue`. Jak název napovídá `@NotNull` značí že hodnota proměnné by neměla být nula. Chceme-li explicitně neomezená pole použijeme `@Basic`. `@Temporal` se používá pro anotaci Datumu a Času. Hojně používanou anotací je `@NamedQueries`, ta specifikuje seznam dotazů do databáze. A v neposlední řadě je tu `@Size`, které validují vstupy vsupy na délku dat.

3.3 Klientská strana webové aplikace

Webová stránka je uživateli reprezentována pomocí HTML5 a CSS. Při práci na stránce se hojně využívá JavaScript s rozšířením jQuery a AJAX, které mohou dynamicky měnit stránku podle požadavků uživatele. Pokud chce uživatel přistupovat k datům v databázi, je mu toto poskytnuto asynchroně pomocí AJAXu, bez jakéhokoliv opakovaného načítání stránky. Existuje zde však pár funkcí. Stránky jsou tedy v převážné části dynamické a na požadavky klienta reagují okamžitě. Vyjíměčně se zde volá serverová strana skrze Java Servlet, která obstarává vyřízení složitějších požadavků od klienta.

3.3.1 Práce s mapovou službou

Pro zobrazení údajů na mapě webová služba Google Maps API využíváme variantu pro JavaScript. Využíváme knihovny JavaScript API a Geocoding pro určení geolokačních údajů jako jsou latitude a longitude. Mapu využíváme jak pro zobrazení údajů, tak pro ukládání. Funkce umožňuje zadat geolokační údaje ručně do formuláře a podle toho je vyhledat v mapě, nebo kliknutím do mapy se vyberou příslušné geolokační údaje, které se vypíšou do formuláře. A uživatel těmito údaji může přiřadit individuální název. Dále využíváme službu pro zobrazení údajů. Uživatel si může místa, která přiřadil svým osobám v rodokmenu, zobrazit na mapě s jeho popisem místa a jménem a aktivním odkazem na konkrétní osobu z rodokmenu, ke které se tyto údaje vztahují. Tyto údaje se mohou zobrazit ve 2 variantách, a to buď jako výpis všech zadaných míst všech osob ze zadaného rodokmenu. V druhém způsobu si uživatel zobrazí na mapě všechna místa všech uživatelů daného rodokmenu. Tyto body jsou propojeny spojnicemi a označeny pořadovým číslem, tyto spojnice zobrazují pohyb osoby v průběhu času. V obou způsobech zobrazení má

FamilyTree

Rodokmeny ▾ Mapa Soubory

Import/Export dat

Zvolte operaci: ▾ Zvolte formát: ▾

Zvolte strom: ▾

Obrázek 3.1: Návrh webového rozhraní pro export/import dat

každá osoba všechny své body a spojnice jednou barvou. Pro zobrazení každé osoby je vybrána jiná barva.

3.3.2 Přenos dat mezi aplikacemi

Přenos dat s jinými aplikacemi se dá realizovat přímo prostřednictvím API anebo nepřímo uložením a exportem a importem souboru ve formátu GEDCOM. Formát GEDCOM je univerzální formát pro ukládání genealogických dat. Aplikace používá svůj vlastní formát pro ukládání a nahrávání dat. 3.1

3.3.2.1 Spolupráce aplikace s jinými programy prostřednictvím jejich API

Pro zjednodušení přenosu dat mezi aplikacemi by bylo užitečné, kdyby spolu různé genealogické aplikace mohly komunikovat přímo. Pro tuto komunikaci by bylo vhodné využívat API daných aplikací. API programu zpravidla umožňují čtení, zápis, nebo obojí. Ne však všechny aplikace provozují a poskytují vlastní API. Přenos dat mezi aplikacemi je dán tím, jaké služby dané API poskytuje. Některá API vyžadují autorizaci uživatele, to znamená, že musí mít zřízený přístup k dané API, a to formou předplatného nebo bezplatnou registrací. Proto jsou některé služby pro uživatele prakticky nedostupné.

3.3.2.2 Export z Geni API

Při volbě exportu z Geni.com je uživatel vyzván k přihlášení k aplikaci Geni. Tím se provede autentifikace uživatele a ten může svá data z aplikace Geni exportovat do této aplikace. Export dat probíhá pomocí JavaScriptu a poskytované knihovny od Geni.com pro komunikaci s API. Veškerý export dat z Geni probíhá prostřednictvím knihovny určené pro zprostředkování komunikace mezi webovým klientem a webovou službou API aplikace Geni.com. JavaScript na straně klienta po autorizaci zašle požadavek na data. Webová služba mu vrátí požadovaná data, která JavaScript klienta zpracuje do formátu, kterému server rozumí. Server tato data přijme a uloží na požadovaná místa.

3.3.2.3 Export dat GEDCOM

Import a export formátu GEDCOM je realizován pomocí knihovny gedcom4j. Knihovna je navržena pro použití v jazyce Java. Uživatel aplikace zvolením volby "export" zašle na server požadavek na export vybraného rodokmenu do formátu GEDCOM. Server vybere data uživatele, která chce exportovat a předá je pro zpracování knihovně gedcom4j. Knihovna vytváří pro data vkládaná serverem objekty podle typu dat. Těmto objektům předává postupně jednotlivé parametry. Po vložení všech údajů server zavolá příkaz pro vygenerování souboru ve formátu GEDCOM. Tento soubor odešle server na webové rozhraní klienta jako odpověď na požadavek uživatele. Webový prohlížeč uživatele tento soubor uloží do výchozí složky pro stahování souboru prostřednictvím prohlížeče. [15]

3.3.2.4 Import dat

Uživatel zvolí ve webové aplikaci možnost "import" a typ souboru pro import. Na základě zvolené možnosti "import" mu bude zobrazeno pole pro výběr souboru pro import. Potvrzením formuláře přepoše soubor s požadavkem na import serveru. Server pro import formátu GEDCOM opět využije knihovnu gedcom4j, které předá ke zpracování přijatý soubor. Knihovna převede data ze souboru do objektů. Server vytvoří pro uživatele, který data importoval, nový rodokmen a k tomuto rodokmenu uloží data z objektů vytvořených knihovnou.

3.3.2.5 Export a import XML

Požadavky probíhají podobným způsobem jako export a import formátu GEDCOM. Avšak oproti GEDCOM se používá pro zpracování formátu JAXB, které usnadňuje konverzi mezi Java Objektem a XML a naopak.

Testování

4.1 REST API

Prostředí REST API bylo otestováno sérií dotazů na API s požadavky POST, GET, PUT a DELETE v tomto pořadí pro kontrolu správného vložení dat, úpravy dat a jejich smazání. Testování bylo provedeno pomocí nástroje poskytnutého od Google Chrome, REST Console, kde je možné na konkrétní URL zaslat požadavek typu GET, POST, PUT, DELETE, a to i s daty ve zvoleném formátu a také s tokenem v hlavičce požadavku.

4.2 Heuristická analýza

Tento způsob testování se provádí bez uživatelů. Provádí ho experti, v našem případě tedy sám programátor. Metoda spočívá v tom, že experti procházejí jednotlivé stránky a kontrolují, zda splňují jednotlivá doporučení ohledně použitelnosti. Výsledkem takové analýzy je seznam objevených problémů. Obsahuje 10 bodů, a tyto jsou:

1. **Viditelnost stavu systému** – systém by měl vždy dát uživateli vědět, co se právě odehrává
2. **Spojení mezi systémem a reálným světem** – komunikace systému s uživatelem by se měla odehrávat uživatelsky přijatelným způsobem (srozumitelnost jazyků bez odborných termínů)
3. **Uživatelská kontrola a svoboda** – uživatelé při práci se systémem dělají chyby a potřebují proto únikový východ pro návrat do předchozího stavu.
4. **Konzistence a standardizace** – uživatelé by neměli přemýšlet, jestli různé termíny znamenají to stejné, proto se doporučuje udržovat obecné zásady.

4. TESTOVÁNÍ

5. **Prevence chyb** - vyvarovat se chybovým hlášením bezpečným designem, který bude preventivně působit proti problémům.
6. **Rozpoznání místo vzpomínání** – uživatel by neměl být nucen vzpomínat si na provádění operací v systému, instrukce by měly být v systému vždy viditelně umístěny
7. **Flexibilní a efektivní použití** – umožnění zrychlení práce se systémem pro pokročilé uživatele
8. **Estetický a minimalistický design** – bez nepotřebných informací
9. **Pomoc uživatelů poznat, pochopit a vzpamatovat se z chyb** – chybové hlášky by měly být uváděny v přirozeném jazyce a měly by navrhnout řešení
10. **Nápověda a návody** – informace se musí vždy dát lehce vyhledat, nápověda by měla obsahovat postupy v krocích.

4.2.1 Výsledky analýzy

1. Pokud je načítáno do mapy mnoho poloh, může se doba, kdy se objekty zobrazují, protáhnout a uživatel v té chvíli neví, zda se něco děje. Podobný problém je u stahování rodokmenu do souboru, toto se může v důsledku velikosti rodokmenu opozdit se začátkem stahování a uživatel neví, zda se stahuje či ne. Některým netextovým elementům chybí popis pomocí title. Při přidávání údajů o osobě je způsob oznámení, že se údaj opravdu přidal, pouze jeho vypsáním v tabulce u příslušné skupiny údajů. Jinak se nezobrazuje žádný informační text, zda se informace nebo osoba opravdu přidala.
2. Na stránce se správou rodokmenů znamená popis změnit jméno rodokmenu, avšak toto není dostačující označení. Tlačítko uložit na stránce s grafem rodokmenu má ukládat obrázek rodokmenu, avšak popis není vhodný, evokuje spíše profil a plátno pro další použití. Dále by byl lepší výstižnější název pro uložení obrázku rodokmenu, který je nyní Uložit a tento název neevokuje pocit, že lze uložit rodokmen do obrázku.
3. Žádnou chybu jsme neobjevili.
4. Tlačítka i tabulky pro vkládání informací jsou konzistentní.
5. Vkládané údaje se validují pomocí HTML 5 a uživatelé jsou tedy před špatně vloženými daty upozorněni na špatně vyplněné informace.
6. Uživatelé se zobrazují ve formulářích již někdy dříve vyplněné údaje. Formulářová okna jsou předepsána typem hodnoty, který se má do nich vložit.

7. Do mapy je načítáno mnoho poloh, doba, kdy se objekty zobrazují, se může protáhnout. Protáhnout se může i export/import z rodokmenu, pokud rodokmen obsahuje mnoho dat.
8. V pořádku – nebyly nalezeny žádné nepotřebné prvky.
9. Chybové hlášky se zobrazují především při odesílání formuláře, kde validitu dat kontroluje HTML5
10. Na stránkách chybí nápověda a dokumentace k REST API

4.2.2 Závěr

1. Měly by se přidat informační texty o načítání údajů nebo o počátku stahování. Netextové prvky bez popisu by měly mít nastavenou vlastnost title, aby uživatel věděl, co se stane, když na ně klikne.
2. Popis "změnit" by se měl změnit na "změnit název". Popis "ulož" by se měl přepsat na "uložit rodokmen".
7. K mapě a importu a exportu by se měl přidat popis typu "Prosíme vyčkejte, data se načítají", nebo "soubor se stahuje".
10. Na stránky by se měla přidat nápověda a dokumentace pro REST API.

4.3 Testování s uživateli

REST API je určeno spíše pro cílovou skupinu vývojářů, zatímco my potřebujeme otestovat především výslednou aplikaci.

4.3.1 Příprava testu

Byly sepsány nejčastější scénáře, které klasický uživatel v aplikaci provede, všechny ostatní se od těchto liší pouze v maličkostech. Testování uživatelé byli obeznámeni s prostředím, ve kterém bude test probíhat. Průběh testu si pozorovatel zapisoval a akce uživatele pasivně sledoval. Zároveň žádal uživatele, aby každou svou akci popisoval, co bude dělat, proč ho to napadlo, atp. Průběh testu byl také nahráván na diktafon pro pozdější použití při hodnocení výsledků. Testování uživatelé jsou anonymní, avšak jedná se o skupinu 3 žen a 3 mužů ve věkovém rozpětí 23-60 let.

4.3.2 Scénář

1. Zaregistruj se do aplikace a přihlaš se.
2. Založ rodokmen a pojmenuj ho Novákovi.

4. TESTOVÁNÍ

3. Do vytvořeného rodokmenu vlož pana Jana Nováka.
4. Panu Novákovi přidej ženu Evu Novákovou.
5. Partnerům Evě a Janovi přidej společné dítě Davida Nováka.
6. K paní Novákové přidej fotografii na ploše svatba.
7. Přiřaď tento soubor i k manželovi panu Novákovi.
8. Ulož si svůj rodokmen do souboru.

4.3.3 Zhodnocení testování

1. Registrace proběhla u všech testovaných uživatelů v pořádku, pouze někteří z nich se museli zkusit registrovat se změněnými údaji, protože jejich heslo nebylo dostatečně dlouhé. Na toto účastníky aplikace upozornila.
2. Další bod scénáře také všichni splnili bez problémů. Ty nastaly, až když měli uživatelé ve vytvořeném stromě vytvořit osobu. Čtyři uživatelé tak trochu bezmyšlenkovitě zkusili klikat v různém pořadí na vše kolem na stránce se seznamem rodokmenů, než se dostali na stránku s hlavním rodokmenem, a to jen díky zvolení tohoto nově vytvořeného stromu jako hlavního. Ostatní uživatelé si stránky prohlédli, včetně položek menu a na zobrazení rodokmenu se dostali správně přes menu.
3. V dalším úkolu měli uživatelé do rodokmenu přidat novou osobu, to se jim všem povedlo bez problémů, avšak 5 uživatelů si nebylo jisto, že se osoba do rodokmenu přidala. Osoba je sice vidět na pozadí, avšak protože je pozadí zatemnělé, nevěnují mu uživatelé tolik pozornosti. Hlavním cílem toho, že okno pro úpravu osoby zůstává otevřené, bylo, aby uživatelé nemuseli informace o osobě opakovaně otevírat. Toto se ukázalo při přidávání osoby jako kontraproduktivní. Avšak když si po prvním vložení na toto uživatelé zvykli, již to pro ně problém nebyl.
4. Další bod scénáře opět všichni uživatelé v pořádku zvládli. Popis je jasný a jednoznačný.
5. Další komplikace nastaly až při přidání společného dítěte, kde většina uživatelů chvíli bádala, jak přidat druhého rodiče, u prvního to nebyl problém. Nebyl to závažný problém, poněvadž po proklikání všech záložek se všichni dovtípili, jak osobě vložit i druhého rodiče.
6. Úkol v 6.tém bodě lze provést třemi způsoby a všechny 3 způsoby byly také odzkoušeny. Nejrychlejším z nich bylo přidání souboru osobě přímo na stránce rodokmenu. Další dva způsoby jsou časově přibližně ekvivalentní, protože vyžadují pouze jedno překliknutí stránky. V jednom

případě byl uživatel rychlejší než prohlížeč, a tak se mu povedlo dostat se na dvě stránky ze tří. Zároveň se tento uživatel pokoušel zkopírovat obrázek z jedné html adresy, kde obrázek vložil, avšak ne k osobě, a nakopírovat ho do druhé html adresy. Nakonec se vrátil na správnou cestu a podařilo se mu obrázek k osobě přiřadit.

7. Další úkol závisel na tom, jak uživatel vložil obrázek k osobě. Pokud ho vložil před dokumenty, měl práci nejjednodušší a okamžitě věděl, jak obrázek přiřadit. 3 uživatelé vložili stejný obrázek na server znovu, aniž by přemýšleli o znění otázky nebo o tom, jestli neexistuje lepší způsob.
8. Co se týče posledního bodu, až na jednoho uživatele všichni uložili rodokmen pomocí formátu XML nebo GEDCOM a nikoho z nich nenapadlo hledat, zda se nedá uložit jako obrázek. Pouze jednoho zmíněného uživatele z praxe napadlo, že by mohl hledat uložení jako obrázek a opravdu tento způsob našel, ačkoliv ze začátku také trochu tápal kvůli pouhému označení "uložit" bez jakéhokoliv dalšího popisu.

4.3.4 Závěr testování

Na základě testování jsme zjistili, že by bylo vhodné udělat dále uvedené změny.

První změnou by bylo přidání odkazu na rodokmen přímo do seznamu rodokmenů ("Přidat rodokmen"). Toto však zapříčiní duplicitu prvku "Přejít do rodokmenu", který bude stejně jako "vybrat hlavní strom" nastavovat zvolený strom na hlavní.

Dále by byl vhodný nějaký úvodní tutoriál, který by uživatele provedl přidáním prvních dvou osob, protože se zdá, že po prvotním vyzkoušení si uživatelé na aplikaci rychle zvykli a jsou schopni s ní nyní efektivně pracovat. Toto by vyřešilo většinu problémů, které uživatelé měli při hledání požadovaných funkcí. Pokud na sebe funkce navazovaly, se splněním úkolu uživatelé problém neměli.

Závěr

Vzhledem ke stále dostupnějším informacím o předcích vzniká větší zájem o zpracování rodokmenu a jsou vytvářeny nové aplikace pro dané téma.

Také předložená práce se zabývá vytvořením webové aplikace pro správu rodokmenů. Před vlastní realizací byla provedena rešerše známých aplikací pro tvorbu rodokmenů, dostupných veřejných API a formátů pro přenos a ukládání dat. Rešerše a získané informace byly analyzovány se zaměřením na získání podkladů pro další postup a návrh aplikace. Jednotlivé aplikace prostředí a formáty dat byly porovnány s cílem zjistit přednosti a nedostatky již funkčních aplikací, z uvedených výsledků vychází návrh této aplikace.

Dále byl proveden průzkum mezi zkušenými uživateli některých genealogických programů a laiky z pohledu očekávaných služeb. Budoucím uživatelům byly položeny otázky na cílové požadavky, ke kterým bylo přihlédnuto při tvorbě funkcí aplikace.

V návrhu jsou popsány technologie použité při tvorbě programu jako například REST a jeho základní principy. Byla vytvořena databáze a definovány její entity. Aplikace byla navržena se strukturou klient-server a popsány jednotlivé vrstvy struktury aplikace. V serverové části je implementována vlastní REST API umožňující přístup k datům i jiným aplikacím. Program využívá externí služby pro zpracování dat. Serverová aplikace je realizována v jazyce Java. Klientská část je navržena pro použití ve webovém prohlížeči na straně uživatele a využívá realizované REST API na serveru. Při implementaci byl využit jazyk Java Platform, Enterprise Edition (J2EE).

Po realizaci byla aplikace otestována několika uživateli z hlediska použitelnosti a funkčnosti. Testováním proběhlo zároveň i ověření funkčnosti programu a jeho využití z hlediska definovaných požadavků na základě analýzy.

Literatura

- [1] MyHeritage Ltd: *Aplikace pro tvorbu rodokmenu MyHeritage [online]*. [cit. 2015-10-15]. Dostupné z: <http://www.myheritage.cz/>
- [2] Geni.Com: *Aplikace pro tvorbu rodokmenu Geni [online]*. [cit. 2015-10-15]. Dostupné z: <https://www.geni.com/>
- [3] The Church of Jesus Christ of Latter-day Saints: *Aplikace pro tvorbu rodokmenu FamilySearch.org [online]*. [cit. 2015-10-15]. Dostupné z: <https://familysearch.org/>
- [4] Ancestry: *Aplikace pro tvorbu rodokmenu Ancestry.com [online]*. [cit. 2016-01-10]. Dostupné z: <http://home.ancestry.com/>
- [5] Doležal, M.: *Desktopová aplikace pro tvorbu rodokmenu Ancestry*. [cit. 2016-01-10]. Dostupné z: <http://ancestry.nethar.cz/>
- [6] Veškrna, M.: *Desktopová aplikace pro tvorbu rodokmenu Rodokmen PRO*. [cit. 2015-10-15]. Dostupné z: <http://www.rodokmenpro.cz/index.php>
- [7] Google Inc.: *Webová aplikace Family Echo [online]*. [cit. 2015-10-15]. Dostupné z: <http://www.familyecho.com/>
- [8] Google Inc.: *Google Maps API [online]*. [cit. 2015-10-15]. Dostupné z: <https://developers.google.com/maps/>
- [9] The Church of Jesus Christ of Latter-day Saints: *The GEDCOM Standard Release 5.5 [online]*. [cit. 2016-01-10]. Dostupné z: <http://homepages.rootsweb.ancestry.com/~pmcbride/gedcom/55gctoc.htm>
- [10] E. Hammer-Lahav, E.: *The OAuth 1.0 Protocol*. 2010, [cit. 2016-01-10]. Dostupné z: <http://tools.ietf.org/html/rfc5849>
- [11] D. Hardt, E.: *The OAuth 2.0 Authorization Framework*. Microsoft, 2012, [cit. 2016-01-10]. Dostupné z: <http://tools.ietf.org/html/rfc6749>

LITERATURA

- [12] Raphael: *Raphael knihovna pro Javascript [online]*. [cit. 2016-01-10]. Dostupné z: <http://code.tutsplus.com/tutorials/an-introduction-to-the-raphael-js-library--net-7186>
- [13] Pecinský, R.: *Návrhové vzory*. Computer Press, a.s., první vydání vydání, 2007, ISBN 978-80-251-1582-4.
- [14] Kulkarni, R.: *Java EE Development with Eclipse*. Packt Publishing Ltd, second edition vydání, 2015, ISBN 978-1-78528-534-9.
- [15] Gedcom4j: *Gedcom4j Library*. [cit. 2016-01-10]. Dostupné z: <http://gedcom4j.org/main/>

Seznam použitých zkratek

- AJAX** Asynchronous JavaScript and XML
- API** Application Programming Interface
- CRUD** Create, Read, Update, Delete
- CSS** Cascading Style Sheets
- CSV** Comma-separated Values
- DOM** Document Object Model
- EJB** Enterprise Java Beans
- FTP** File Transfer Protocol
- GPL** General Public License
- GUI** Graphical User Interface
- HATEOAS** Hypermedia as the Engine of Application State
- HTML** Hypertext Markup Language
- HTML5** Hypertext Markup Language 5
- HTTP** Hypertext Transfer Protocol
- HTTPS** Hypertext Transfer Protocol Secure
- J2EE** Java 2 Enterprise Edition
- JAXB** Java Architecture for XML Binding
- JAX-RS** Java API for RESTful Web Services
- JAX-WS** Java API for XML Web Services

A. SEZNAM POUŽITÝCH ZKRATEK

JCA Java Connector Architecture

JPA Java Persistence API

JSF Java Server Faces

JSM Java Messaging Service

JSON JavaScript Object Notation

JSONP JSON with Padding

JSP JavaServer Pages

JTA Java Transaction API

JVM Java Virtual Machine

MathML Mathematical Markup Language

MDB Message Driven Bean

MVC Model View Controller

MySQL Relační databáze typu DBMS založená na jazyce SQL

ORM Objektově relační mapování

PDF Portable Document Format

POJO Plain Old Java Object

REST Representational State Transfer

RMI Remote Method Invocation

SQL Structured Query Language

SVG Scalable Vector Graphics

TSV Tab-separated Values

URI Uniform Resource Identifier

URL Uniform Resource Locator

VML Vector Markup Language

W3C World Wide Web Consortium

XHTML Extensible Hypertext Markup Language

XML Extensible Markup Language

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	exe	adresář se spustitelnou formou implementace
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu \LaTeX
	text	text práce
	thesis.pdf	text práce ve formátu PDF
	thesis.ps	text práce ve formátu PS
	appendix.....	přílohy
	instalation_guid.txt	instalační příručka
	mysql_queries.sql	příkazy do databáze MySQL