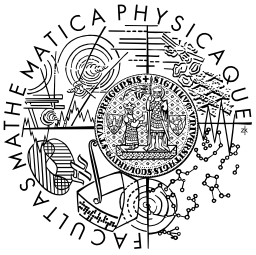


MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Kristýna Lhoťanová

Webová aplikace pro vyhledávání receptů

Katedra softwarového inženýrství

Vedoucí bakalářské práce: doc. Mgr. Martin Nečaský, Ph.D.
Studijní program: Informatika
Studijní obor: Programování a vývoj software

Praha 2022

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ....

Podpis autora

Poděkování.

Název práce: Webová aplikace pro vyhledávání receptů

Autor: Kristýna Lhoťanová

Katedra: Katedra softwarového inženýrství

Vedoucí bakalářské práce: doc. Mgr. Martin Nečaský, Ph.D., Katedra softwarového inženýrství

Abstrakt: Cílem této práce je vyvinout webovou aplikaci pro vyhledávání receptů založenou na agregaci datových sad z existujících webových stránek s recepty a jejich obohacení o data ze znalostních grafů. Znalostní grafy byly zastoupeny projekty DBpedia a Wikidata, z nichž byla získána data o ingrediencích a kategoriích jednotlivých receptů. Data byla extrahována s pomocí knihovny Apify a v dokumentovém modelu uložena do databázového systému Apache CouchDB. Aplikace uživateli poskytuje různé možnosti filtrování výsledků včetně fasetového vyhledávání, k čemuž využívá platformu Apache Solr. Zaměřuje se zejména na vyhledávání dle ingrediencí. Jedná se o tzv. single-page aplikaci implementovanou pomocí JavaScriptové knihovny React pro uživatelské rozhraní a frameworku Express.js na straně serveru. Obě části aplikace jsou psány staticky typovaným jazykem TypeScript a komunikují spolu prostřednictvím REST API.

Klíčová slova: webová aplikace, recept, znalostní graf, propojená data

Title: Web application for searching recipes

Author: Kristýna Lhoťanová

Department: Department of Software Engineering

Supervisor: doc. Mgr. Martin Nečaský, Ph.D., Department of Software Engineering

Abstract: Abstract.

Keywords: web app, recipe, knowledge graph, linked data

Obsah

Úvod **2**

1 Analýza **5**
1.1 Požadavky aplikace . . . 5
1.1.1 Funkční požadavky . . . 5
1.1.2 Nefunkční požadavky . . . 6
1.1.3 Obrazovky aplikace . . . 8
1.2 Dostupné datové sady . . . 8
1.2.1 Recipe1M+ . . . 11
1.2.2 Open Recipes . . . 11
1.2.3 FoodKG . . . 11
1.2.4 Food.com Recipes and Interactions . . . 12
1.2.5 DBpedia . . . 13
1.2.6 Wikidata . . . 14
1.2.7 Generování vlastního datasetu . . . 14

2 Architektura řešení **17**
2.1 Příprava dat . . . 17
2.2 Databázový model . . . 17
2.3 Backend . . . 17
2.4 Frontend . . . 17
2.5 Aplikační logika . . . 17

3 Implementace návrhu **18**
3.1 Zpracování vstupních dat . . . 18
3.2 Databáze Apache CouchDB . . . 18
3.3 Vyhledávání pomocí Apache Solr . . . 18
3.4 Rozhraní REST API . . . 18
3.5 Middleware . . . 18
3.6 Single-page aplikace . . . 18

4 Testování **19**

Závěr **20**

Seznam použité literatury **21**

Seznam obrázků **22**

Seznam tabulek **23**

A Přílohy **24**
A.1 První příloha . . . 24

Úvod

Vyhledávání relevantního obsahu je spolu s elektronickou komunikací jednou z klíčových funkcí internetu. S rostoucím množstvím dostupných informací se filtrování nalezených výsledků stává stále obtížnějším. Tvůrci webových stránek se často zaměřují spíše na uživatelsky přívětivé interaktivní rozhraní, zatímco optimalizace strojového vyhledávání jde stranou. Pro webové vyhledávače, jmenovitě např. Google, Bing nebo Yahoo, je pak náročné analyzovat obsah těchto stránek po sémantické stránce a tedy vyhodnotit, zda obsahují užitečné informace k zodpovězení dotazu uživatele.

V reakci na tuto problematiku vznikl tzv. *Sémantický Web* neboli Web dat jakožto rozšíření původního Webu dokumentů, tak jak jej známe z platformy *World Wide Web*. Sémantický Web lze vnímat jako globální databázi, nad kterou se lze pomocí speciálního jazyka *SPARQL* dotazovat podobně jako nad tradičními databázovými systémy. Data jsou poskytována v různých serializacích formátu RDF a mohou být přímo vnořena do HTML dokumentů nebo zpřístupněna v samostatných souborech. Tato strukturovaná data nazýváme *propojená* (v originále *Linked Data*). Umožňují snadnější hledání souvislostí mezi entitami z různých zdrojů na základě společných slovníků neboli ontologií [1].

V posledních letech termín Sémantický Web ustupuje do pozadí a často je místo něj zmiňován tzv. *znalostní graf* (anglicky *Knowledge Graph*). Začátky fenoménu znalostních grafů bychom mohli datovat do roku 2012, kdy společnost Google představila svůj znalostní graf pro vyhledávání obsahu na webu. K technologii znalostních grafů se brzy poté přihlásily další velké společnosti včetně firem Microsoft, IBM, Facebook, LinkedIn, Amazon, eBay, Airbnb nebo Uber. Grafový model totiž oproti tradičnímu relačnímu modelu nabízí flexibilnější správu dat z oblasti sociálních sítí, dopravních spojení, bibliografických citací a řady dalších odvětví. Výše zmíněné příklady znalostních grafů všechny spadají do kategorie komerčních znalostních grafů, které jsou určeny pro interní využití v rámci dané firmy. Protikladem jim jsou otevřené znalostní grafy poskytující data k volnému využití všem uživatelům internetu. Nejvýznamnějšími představiteli otevřených znalostních grafů jsou aktuálně DBpedia, Wikidata, Freebase a YAGO [2]. První dva zmíněné projekty si představíme v této práci a integrujeme je s aplikací na vyhledávání receptů.

Oblast gastronomie je rozvěž vhodným kandidátem k zapojení do sítě znalostních grafů a propojených dat. Pro tvůrce webových aplikací je poměrně jednoduché publikovat obsah svých stránek ve formátu strukturovaných dat. Vhodným způsobem je např. vložení RDF reprezentace daných entit (receptů, uživatelů, recenzí) ve formátu JSON-LD[1] přímo do hlavičky jednotlivých HTML dokumentů. V takovém případě je žádoucí použít existující ontologie raději než definovat vlastní, byť by mohly být lépe strukturované a uzpůsobené dané doméně. Využití standardizovaných slovníků usnadňuje webovým vyhledávačům interpretaci stránky a je větší šance, že se aplikace dostane na vyšší příčky vyhledávaných výsledků.

Cílem této bakalářské práce je prozkoumat možnosti využití otevřených dat v doméně receptů, propojit je s daty publikovanými na různých webových strán-

[1]Koncovka *LD* v názvu JSON-LD odkazuje na pojem Linked Data.

kách shromažďujících recepty a prezentovat tyto výsledky uživateli ve formě vlastní webové aplikace. Zároveň v rámci této aplikace poskytnout užitečné možnosti filtrování agregovaných výsledků včetně fasetového vyhledávání. Proces sběru, konverze a uložení dat by měl být co nejvíce automatizovaný a snadno zreprodukovatelný. Práce se nevěnuje přidávání nových receptů prostřednictvím uživatelského rozhraní. Existujících webové stránky totiž obsahují velké množství dat, které lze díky bohaté historii v podobě hodnocení a recenzí lépe filtrovat. Navíc by bylo potřeba se vypořádat s automatickou kalkulací nutričních hodnot receptu z obsažených surovin, přičemž ne všechny ingredience dokážeme automaticky identifikovat a získat jejich nutriční hodnoty. V budoucnu by funkce nahrávání nových receptů měla být přidána spolu s více lokalizacemi aplikace, registrací uživatelů a celkovou personalizací obsahu pro přihlášené uživatele. Dále se práce v této fázi nezabývá nasazením, neboť by vyžadovalo větší časové i finanční prostředky na získání dostatečně velkého množství dat a také poměrně robustní databázi pro uložení extrahovaných dat.

Volba tématu

Příprava jídla je tématem každodenního života a na webových stránkách, které se této oblasti věnují, má velmi silnou komunitu. Většina z nás se chystání domácích pokrmů z ekonomických důvodů nevyhne, takže se hodí mít po ruce sadu receptů pro inspiraci. Typicky máme na recepty různé požadavky — někdo preferuje rychlejší postup, jiný se dívá po ceně ingrediencí nebo nutričních hodnotách. Občas dostaneme chuť na recept z řecké nebo italské kuchyně a jindy zkrátka chceme experimentovat a najít recept kombinující našich 5 oblíbených surovin. Některé ingredience z receptu nám mohou být neznámé, nebo si podle samotného názvu nejsme jistí, zda máme na mysli tu správnou. V takovém případě musíme stránku s receptem opustit a dodatečné informace k ingredienci vyhledat jinde, pokud na ně aplikace přímo neodkazuje. Zde je příležitost zapojit otevřená data a namapovat názvy ingrediencí na jejich odpovídající entity ve znalostních grafech. Data pak můžeme začlenit do aplikace a nabídnout uživateli informace nad rámec samotného receptu, např. popisy a glykemické hodnoty surovin, ilustrační obrázky a podobně. Také můžeme identifikovat ingredience a tranzitivně recepty ze stejných kategorií. Oproti původní datové sadě tak vytvoříme nové vazby a poskytneme uživateli rozmanitější filtrování výsledků.

Doména receptů navíc poskytuje spoustu prostoru pro zajímavá rozšíření se zapojením moderních technologií. Uplatnění by zde našlo například počítačové vidění s rozpoznáváním obrázků. S dostatečně velkou databází bychom díky němu mohli analyzovat fotografii hotového pokrmu a nalézt příslušný recept. Usnadnili bychom tak uživateli práci v situacích jako je návštěva restaurace, při které návštěvníkovi zachutnalo servírované jídlo a chtěl by si jej později připravit v domácích podmínkách. Dalším uplatněním strojového učení by mohlo být vyhledávání na základě příkazů v přirozeném jazyce. Namísto zdlouhavého zadávání nejrůznějších filtrů by stačilo aplikaci položit dotaz: „Jaké recepty z italské kuchyně mohu vyrobit z kuřete, rajčat a parmazánu?“. Této problematice se věnuje například projekt FoodKG konstruující nad recepty a ingrediencemi znalostní graf [3]. Ruku v ruce s touto funkcionalitou jde hlasové zadávání, které by se hodilo zapojit nejen ve fázi vyhledávání receptů, ale také například pro hands-free ovládání aplikace.

Uživatel by měl možnost diktovat příkazy k přečtení části receptu, pokud zrovna pracuje na jeho přípravě a nemá volné ruce k listování obsahem. Využití by našlo i populární *full-text* vyhledávání, pomocí kterého lze snadno objevit recepty na základě klíčových slov v popisku receptu, postupu či recenzích. V komerční sféře by se nabízelo propojení s online supermarkety, konkrétně zrychlení nákupu pomocí vyhledávání surovin k vybranému receptu. S tímto konceptem již na svých stránkách pracuje firma rohlik.cz, nabídka receptů a možnosti filtrování jsou ale omezené. Nepochybně by se hodilo integrovat také doporučovací systém pro ještě snadnější nalezení relevantních výsledků. Aplikace má velký prostor pro škálování objemu dat, přičemž datasety mohou být následně použity jako podklad pro strojové učení.

1. Analýza

V této kapitole si zadefinujeme požadavky na funkcionalitu naší aplikace. Také se v kontextu požadavků podíváme na existující webové stránky s recepty a provedeme diskuzi nad jejich funkcemi, možnými vylepšeními a rozšířeními. Následně si rozebereme různé alternativy dostupných datových sad a srovnáme jejich výhody i nevýhody vzhledem k požadavkům aplikace.

1.1 Požadavky aplikace

Nyní si rozebereme požadavky na naši aplikaci, které můžeme rozdělit do skupin funkčních a nefunkčních požadavků. Funkční požadavky popisují konkrétní funkcionalitu systému, zabývají se vstupem od uživatele a prezentací výstupu. Díky tomu je lze poměrně snadno definovat a testovat jejich naplnění v hotové aplikaci. Nefunkční požadavky se naopak na konkrétní vstup nevážou a místo toho popisují vlastnosti a omezení, které by měl systém splňovat. Zjednodušeně lze říci, že funkční požadavky popisují, co má systém dělat, zatímco nefunkční požadavky specifikují, jaký má systém být [4].

1.1.1 Funkční požadavky

Následuje výčet funkcionalit, které by aplikace svým uživatelům měla nabídnout. Uživatelé mohou mít různé role od běžného návštěvníka stránky po administrátora nebo vývojáře integrujícího data do jiného systému.

Běžný uživatel

1. Aplikace poskytuje uživatelské rozhraní pro vyhledávání receptů na základě ingrediencí, klíčových slov, času přípravy, hodnocení a nutričních hodnot.
2. Aplikace umožňuje kombinovat libovolné množství vyhledávácích filtrů.
3. Aplikace podporuje zadávání vlastních i předdefinovaných ingrediencí prostřednictvím našeptávače.
4. Aplikace podporuje fasetové vyhledávání, tedy u nabízených možností zobrazuje počet receptů, které se po zvolení daného filtru zobrazí.
5. Aplikace poskytuje možnost smazání všech vyhledávacích filtrů jedním kliknutím, ale také mazání po jednom filtru.
6. Aplikace zobrazuje uživateli všechny nalezené výsledky bez omezení na maximální počet výsledků.
7. Aplikace při otevření vyhledávací obrazovky bez zadaných filtrů zobrazuje všechny recepty, které má v databázi.
8. Aplikace umožňuje zobrazení detailu receptu rozkliknutím nalezeného výsledku.

9. Aplikace zobrazuje pouze recepty s titulní fotografií.
10. Aplikace na vyhledávací stránce pro každý nalezený recept zobrazuje jeho název, popis, obrázek, čas přípravy, hodnocení a počet recenzí.
11. Aplikace nabízí náhledy všech ingrediencí u vyhledaných receptů a zvýrazňuje aktuálně vyhledávané ingredience.
12. Aplikace umožňuje listování nalezenými výsledky prostřednictvím systému stránkování, nikoli nekonečným posouváním stránky.
13. Aplikace plně podporuje navigaci v rámci historie prohlížeče včetně přidávání a odebírání filtrů i listování více stranami výsledků.
14. Aplikace na detailní stránce každého receptu zobrazuje název, hodnocení, počet recenzí, popis, čas přípravy, fotografii, ingredience, postup přípravy a nutriční hodnoty.
15. Aplikace zvýrazňuje ingredience na detailní stránce receptu, ke kterým má dodatečné informace.
16. Aplikace přesměrovává na obrazovku s detailem ingredience po kliknutí na zvýrazněnou ingredienci.
17. Aplikace zobrazuje na detailní stránce ingredience následující informace nebo jejich podmnožinu: název, popis, obrázek, nutriční hodnoty, náhrady, kategorie a níže recepty obsahující tuto ingredienci, které lze otevřít stejně jako z vyhledávací obrazovky.
18. Aplikace má nezávisle na otevřené stránce viditelný ovládací panel s možností navigace na vyhledávací obrazovku.

Externí systém

1. Aplikace poskytuje REST API endpointy pro získání dat k receptům a ingrediencím.
2. Aplikace zpřístupňuje JSON-LD reprezentaci dat v hlavičkách dokumentů s recepty a ingrediencemi.
3. Aplikace podporuje navigaci a vyhledávání receptů přes url adresy s query parametry.

1.1.2 Nefunkční požadavky

Požadavky z této kategorie lze dále dělit podle jejich zaměření. Některé se věnují výkonu aplikace, jiné spolehlivosti, přenositelnosti, bezpečnosti, využitým technologiím, vývojovému prostředí nebo platformě, testovatelnosti či rozšiřitelnosti. Oblastí je zde skutečně mnoho, uvedeme proto pouze výčet konkrétních požadavků na naši aplikaci.

1. Backend aplikace je postaven na frameworku Express.js pro Node.js prostředí.

2. Frontend aplikace je implementován pomocí knihovny React.
3. Backend i frontend aplikace jsou psány staticky typovaným jazykem TypeScript.
4. Aplikace využívá dokumentovou databázi Apache CouchDB pro uložení dat o receptech a ingrediencích.
5. Aplikace využívá systém Apache Solr pro implementaci vyhledávání receptů.
6. Aplikace využívá program Silk Workbench pro objevování linků mezi dvěma entitami.
7. Aplikace je implementovaná jako single-page aplikace s podporou routingu mezi více obrazovkami.
8. Aplikace integruje data z aspoň 2 různých veřejných znalostních grafů.
9. Aplikace pro komunikaci mezi klientem a serverem používá REST API v kombinaci s asynchronními requesty.
10. Uživatelské rozhraní aplikace je založeno na knihovně Material UI poskytující sadu univerzálních komponent pro React aplikace.
11. Uživatelské rozhraní aplikace je responzivní pro desktopová i mobilní zařízení.
12. Databáze obsahuje v prvotní fázi přes 50 000 receptů z aspoň 2 různých zdrojů.
13. Aplikace je škálovatelná co do množství poskytovaných dat.
14. Aplikace je škálovatelná z pohledu nových lokalizací a jejich distribuce.
15. Aplikace je připravena pro implementaci nových rozšíření bez nutnosti výrazné změny stávajícího kódu.
16. Vyhledávání receptů je pro nového uživatele přímočaré a filtrování zvládne nastavit v řádu vteřin až minut v závislosti na počtu požadovaných filtrů.
17. Zdrojový kód aplikace je open-source a verzovaný na platformě GitHub.
18. Zdrojový kód aplikace je přehledný a snadno rozšiřitelný dalšími vývojáři.
19. Komponenty aplikace jsou znovupoužitelné v rámci projektu i mimo něj.
20. Získání receptů pro jednu stránku výsledků trvá méně než 500 ms (200 ms dotaz na server, 200 ms doručení odpovědi klientovi a 100 ms rezerva).
21. Rendering jedné stránky vyhledaných receptů trvá méně než 1 000 ms od načtení dat do paměti.
22. Nově extrahovaná data se uživatelům aplikace zobrazí nejpozději do druhého dne.
23. Aplikace je kompatibilní s webovými prohlížeči Google Chrome, Mozilla Firefox a Microsoft Edge.

1.1.3 Obrazovky aplikace

Přestože vyvíjíme single-page aplikaci, počítáme s více obrazovkami pro pohodlnější navigaci. S využitím knihovny React Router dokážeme simulovat existenci libovolného množství obrazovek a zároveň zůstat na jedné stránce bez potřeby opětovného načítání. Tím se odlišíme od tradičních statických aplikací, kterým při každé změně url včetně query parametrů musí server v odpovědi poslat odpovídající HTML obsah. Náš přístup má ovšem nevýhodu z pohledu strojového zpracování, neboť pro vygenerování obsahu stránky potřebujeme v prohlížeči spustit JavaScript kód. Tím znemožníme zpracování naší aplikace prostřednictvím pouhých HTTP requestů, což je podstatně jednodušší a ekonomičtější varianta ve srovnání s automatizací celého webového prohlížeče. Tento nedostatek ale kompenzujeme transparentním REST API, přes které si lze vyžádat strukturovaná data přímo přes HTTP requesty. Zároveň usnadníme automatické zpracování vyhledávačům, které automatizaci prohlížeče využívají, neboť v detailech receptů a ingrediencí zahrneme jejich JSON-LD reprezentaci.

Aplikaci složíme ze 3 základních uživatelských obrazovek: vyhledávání receptů, detail receptu a detail ingredience. Všechny obrazovky musí být responzivní a poradit si s proměnlivou velikostí obrazovky.

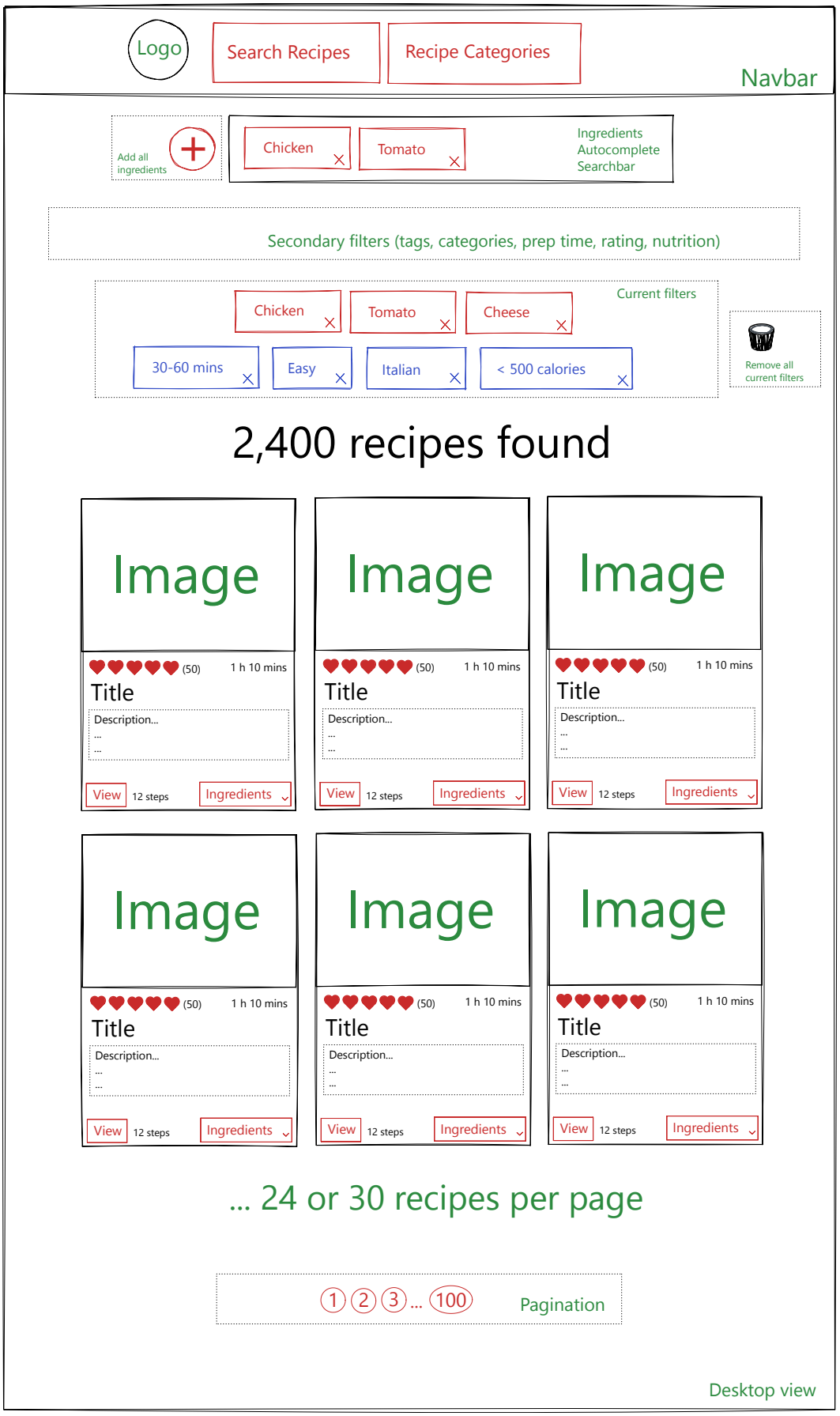
Vyhledávání receptů

Domovskou stránku bude tvořit vyhledávání receptů na základě různých kritérií. Primárně bude k dispozici výběr požadovaných ingrediencí, sekundárně filtry klíčových slov, kategorií, času přípravy, hodnocení a nutričních hodnot. Všechny filtry bude možné odstranit samostatně i najednou pomocí společného tlačítka pro smazání. Pro získání přesnějších výsledků bude při vyplňování filtrů k dispozici našeptávač, který zobrazí známé možnosti a spolu s nimi počty receptů, které jsou při výběru tohoto nastavení k dispozici. Výsledky budou zobrazovány na stránkách s 24 nebo 30 kartami receptů. Přepínání stránek bude umístěno standardně ve spodní části stránky a zároveň bude aktuální stránka figurovat v query parametrech pro přímočarou podporu navigace v historii prohlížeče. Karta receptu bude obsahovat název, popis, obrázek, hodnocení, počet recenzí, čas přípravy a počet instrukcí. Navíc bude možné rozbalit seznam ingrediencí, ve kterém budou zvýrazněny aktuálně vyhledávané ingredience. Uživatel bude přesměrován na obrazovku s detailem receptu při stisknutí tlačítka `View` nebo při kliknutí na obrázek receptu.

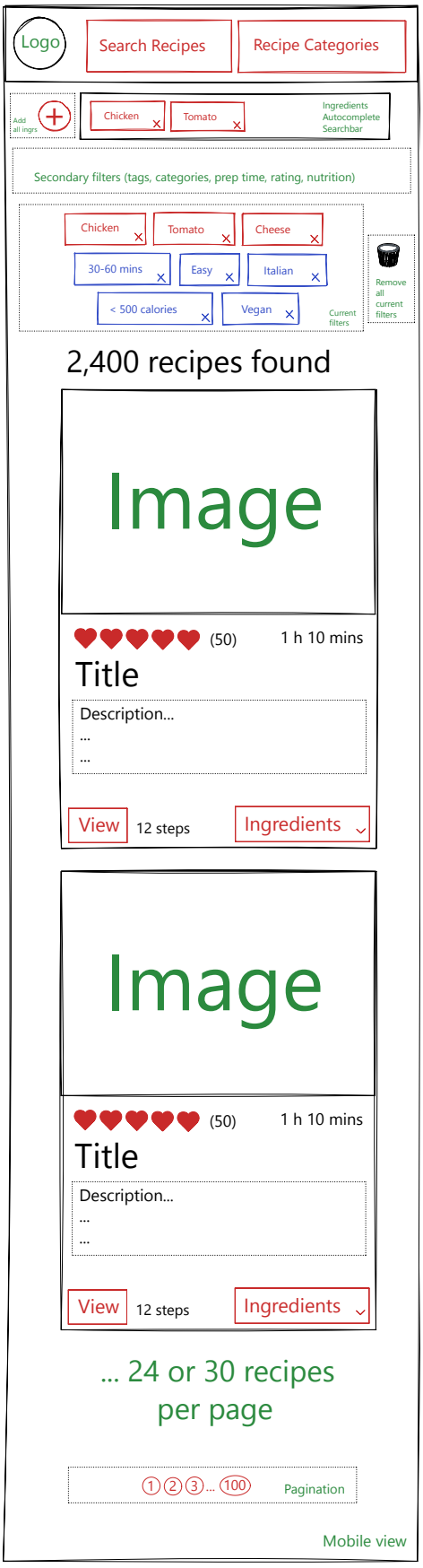
Konkrétní rozložení obrazovek viz obrázek 1.1 pro desktopová zařízení a 1.2 pro mobilní zařízení.

1.2 Dostupné datové sady

V této sekci je vyhrazen prostor pro analýzu různých veřejně dostupných datasetů z domény receptů. Nejedná se ani zdaleka o kompletní výčet, měly by ale být představeny nejznámější alternativy, které by mohly být vybrány jako podklad pro obsah aplikace.



Obrázek 1.1: Obrazovka vyhledávání receptů pro desktopová zařízení



Obrázek 1.2: Obrazovka vyhledávání receptů pro mobilní zařízení

1.2.1 Recipe1M+

Jedním z nejdůležitějších projektů v této oblasti je *Recipe1M+*, strukturovaný korpus obsahující přes 1 milion receptů a 13 milionů souvisejících obrázků jídla. Aktuálně se jedná o největší veřejně dostupnou sadu receptů. Dataset je dostupný pouze přihlášeným uživatelům z ověřené organizace a je povoleno jej využívat výhradně pro účely studia a výzkumu. Pro registraci lze využít univerzitní email. Z celkového počtu 1 milionu receptů obsahuje 50 000 receptů s nutričními informacemi [5]. V naší aplikaci preferujeme nutriční hodnoty zahrnout, pokud jsou dostupné na zdrojové stránce receptu. Měli bychom tedy k dispozici 50 000 dokumentů s touto informací. Ostatní data jsou určena přednostně pro strojové zpracování prostřednictvím trénování modelů.

Celková velikost datové sady se pohybuje v řádu stovek gigabytů, samotné JSON dokumenty se strukturovanými recepty z adresáře `layers` se ale vejdou do 2 GiB, tudíž by byly vhodné pro potřeby této práce limitované omezenou výpočetní kapacitou. Lze odtud využít 1 029 720 receptů obsahujících název, url, ingredience a postup přípravy. Odkazy na ilustrační fotografie jsou u 402 760 z těchto receptů. Pro příjemnější uživatelský zážitek se omezujeme pouze na recepty s obrázky, takže jsme z datasetu Recipe1M+ schopni použít přibližně 400 000 receptů, pokud akceptujeme absenci nutričních hodnot. Bylo by spíše obtížnější z tohoto datasetu identifikovat názvy ingrediencí, neboť jsou suroviny uloženy včetně jejich množství a jednotek měření v rozmanitém formátu.

1.2.2 Open Recipes

Dalším významným aktérem na poli volně dostupných receptů je iniciativa *Open Recipes*. Autoři Finkler, Shiflett a Birkebæk projekt představují jako otevřenou databázi záložek s recepty. Pojem záložky je použit z důvodu absence instrukcí k přípravě receptu. Dataset má sloužit pouze k vyhledání receptu a pro detailní informace má být uživatel přesměrován na zdroj s kompletním receptem [6]. Tohoto přístupu úspěšně využívají některé z vyhledávačů receptů, např. populární aplikace *SuperCook*. Naše aplikace si ale klade za cíl zpracovat i stránky s detaily receptů, ze kterých lze dále pokračovat na detaily ingrediencí s informacemi ze znalostních grafů. Projekt Open Recipes tedy pro náš scénář nebude vhodnou volbou.

1.2.3 FoodKG

Přímo v oblasti znalostních grafů figuruje projekt *FoodKG*, který je postaven nad sadou receptů z již zmíněného datasetu Recipe1M+. Recepty doplňuje o podrobnější data k ingrediencím ze stránky The Cook's Thesaurus a definuje vlastní ontologii. Model ontologie je navržen pro zodpovídání dotazů na recepty dle ingrediencí s přihlédnutím k individuálním potřebám uživatele, jako jsou alergie a intolerance na určité složky potravin.

Vývojáři projektu FoodKG zpřístupňují skripty k extrakci dat z encyklopedie The Cook's Thesaurus a k vytvoření znalostního grafu. Neposkytují ale žádné nové recepty nad rámec datové sady Recipe1M+, naší horní hranicí by tedy bylo 50 000 receptů s nutričními hodnotami (viz sekce *Recipe1M+*). Ontologie publikovaná na webových stránkách projektu obsahuje 75 entit ingrediencí, které

kromě obecného popisu poskytují informace o glykemickém indexu, obsahu lepku a možných náhradách dané ingredience. Výhodou je připravený RDF formát, nad kterým se lze snadno dotazovat pomocí jazyka SPARQL. Autoři Chen a kol. uvádějí ukázky dotazů, vyberme například dotaz vracející recepty, které obsahují banán a zároveň neobsahují vlašské ořechy [3]:

```
@PREFIX food: <http://purl.org/heals/food/>
@PREFIX ingredient: <http://purl.org/heals/ingredient/>
SELECT DISTINCT ?recipe
WHERE {
    ?recipe food:hasIngredient ingredient:Banana .
    FILTER NOT EXISTS {
        ?recipe food:hasIngredient ingredient:Walnut .
    }
}
```

1.2.4 Food.com Recipes and Interactions

Rozsáhlý dataset *Food.com Recipes and Interactions* s téměř 200 000 recepty extrahovanými z webové stránky Food.com (původního GeniusKitchen) je publikován na portálu *Kaggle*, který shromažďuje podklady pro strojové učení. Datová sada pokrývá 18 let interakce uživatelů včetně hodnocení, počtu recenzí i konkrétních reakcí [7]. Kromě základních informací obsahuje také nutriční hodnoty receptů, datum publikování a rovněž normalizovaná jména ingrediencí. Ta byla získána parsováním originálního textu surovin, kvůli čemuž nejsou vždy zcela spolehlivě přesná (např. ve jménech často zůstala jednotka měření z původního textu). Unikátních ingrediencí je k dispozici kolem 8 000, což by měl být dostačující základ pro hledání linků s entitami otevřených znalostních grafů. Zároveň ve srovnání s předchozími projekty nabízí nejbohatší informace k jednotlivým receptům.

Nevýhodou datasetu je jeho primární určení pro strojové zpracování. Byl vytvořen jako podklad pro generování personalizovaných receptů na základě dřívějších preferencí uživatele [8]. Syrová data nejsou zamýšlena pro přímou prezentaci, což se negativně odráží na jejich přesnosti a estetice. Slova jsou občas zařazena do špatných kategorií a problematický je zejména plně *lowercase* formát textu, ze kterého nejsme schopni zpětně zrekonstruovat originální text receptu. Dataset bychom tedy nemohli použít samostatně, ale pouze v kombinaci s vlastní extrakcí dat, která by respektovala velikost písma a lépe se vypořádala s parsováním jednotlivých kategorií.

Tento problém je poměrně snadno řešitelný díky struktuře stránky Food.com. Z id receptu lze jednoduše složit url ve formátu `www.food.com/recipe/id` a navíc aplikace podporuje koncept propojených dat, tedy poskytuje recepty ve strukturovaném RDF formátu. Do HTML hlaviček všech dokumentů s recepty vkládá JSON-LD serializaci dle ontologie *Schema.org*. Z připraveného datasetu bychom tedy mohli využít identifikátory receptů a normalizované ingredience, pro každý recept extrahovat jeho JSON-LD a spojit informace dohromady. Zároveň bychom si ušetřili práci s převáděním receptů do JSON-LD formátu a připravené soubory rovnou vložili do hlaviček dokumentů. Nevytvářeli bychom nové entity receptů,

pouze bychom změnili prezentační vrstvu RDF dat. Identifikátory entit v podobě IRI by tedy zůstaly nezměněné.

1.2.5 Generování vlastního datasetu

Pokud se nespokojíme s žádnou z dostupných datových sad, případně potřebujeme data rozšířit a posbírat je přímo ze zdroje, využijeme metodu zvanou *web scraping*. V rámci tohoto procesu musíme analyzovat cílovou stránku z pohledu získávání a prezentace dat. S využitím vývojářským nástrojů ve webovém prohlížeči můžeme přes panel `Network` sledovat požadavky, které aplikace odesílá na svůj server a v mnoha případech se na toto interní API dokážeme napojit a získat data ve strukturované podobě. Aplikace typicky pracují s REST API, GraphQL API nebo jejich kombinací a standardně data poskytují ve formátu JSON. Pokud žádný fetch request pro získávání potřebných dat neobjevíme, musíme informace extrahovat přímo z HTML dokumentu prostřednictvím CSS selektorů. V obou případech budeme aplikaci posílat GET requesty, ať už na její backend pro strukturovaná data nebo na frontend pro HTML dokumenty k následnému parsování.

Problematická je kategorie aplikací, které data nezískávají s využitím transparentních fetch requestů a zároveň potřebují spouštět JavaScript kód pro vygenerování obsahu. Zde nestačí pouhé poslání GET requestu přes HTTP, neboť odpověď neobsahuje žádná relevantní data uvnitř HTML. Pro zvládnutí tohoto typu stránek potřebujeme zapojit automatizaci webového prohlížeče. Nejznámějšími projekty, které se této automatizaci věnují, jsou Selenium[1], Puppeteer[2], Playwright[3] a Cypress[4] pro automatizaci testování [9]. Všechny ze zmíněných projektů jsou open-source.

Během posílání requestů můžeme rovněž narazit na různé formy blokování, od limitu maximálního počtu requestů z jedné IP adresy přes povinné autorizační tokeny až po captcha testy řešitelné pouze s využitím umělé inteligence. Některé aplikace navíc kontrolují tzv. otisk webového prohlížeče. Jedná se o sadu informací k zařízení uživatele, jmenovitě data o konkrétním hardwaru, operačním systému a webovém prohlížeči včetně konfigurace [10]. Také se při neopatrnosti může stát, že server aplikace zahltíme příliš velkým množstvím paralelních requestů, čímž prodloužíme dobu odezvy nebo zpracování dalších requestů dočasně zcela znemožníme.

Stejně jako v jiných oblastech se hodí využít nástroj, který co nejvíce běžných problémů vyřeší za nás. Na poli open-source nástrojů pro extrakci dat si vedoucí pozici drží knihovna Scrapy[5] psaná v jazyce Python, která nabízí celou řadu pokročilých funkcí proti blokování requestů. Pro potřeby této práce by ale vzhledem k rozsáhlejší osobní zkušenosti byla vhodnou volbou knihovna Apify[6] pro Node.js. V arzenálu má zpracování HTTP requestů s následným parsováním HTML pomocí knihovny Cheerio[7], ale také automatizaci webového prohlížeče s využitím knihoven Puppeteer nebo Playwright, včetně generování otisků webo-

[1] https://github.com/SeleniumHQ/selenium
[2] https://github.com/puppeteer/puppeteer
[3] https://github.com/microsoft/playwright
[4] https://github.com/cypress-io/cypress
[5] https://github.com/scrapy/scrapy
[6] https://github.com/apify/apify-js
[7] https://github.com/cheeriojs/cheerio

vého prohlížeče. Navíc zajišťuje rotaci IP adres, čímž snižuje množství zablokovaných requestů. IP adresy lze v rámci placeného účtu získat přímo od firmy Apify, nebo na vstupu poskytnout seznam vlastních. Obecně preferujeme program nespouštět z osobní IP adresy, neboť riskujeme, že nás stránka někdy i natrvalo zablokuje, případně se naše IP adresa dostane na veřejný seznam adres doporučených k blokování.

S dostatkem času, výpočetních prostředků, IP adres pro rotování a s velikou kapacitou úložiště bychom byli schopni zpracovat většinu vybraných aplikací s recepty. Pro každou stránku bychom napsali dedikovaný program a postupně extrahovali data z celé stránky. Recepty z různých aplikací bychom uložili ve sjednoceném formátu a výsledkem by byl kvalitní dataset s maximálním množstvím dat, které lze od zdrojových stránek získat. Práce ovšem necílí na datovou sadu takovéto velikosti. Místo toho se zaměřuje na vytvoření infrastruktury nad podmnožinou receptů, kterou bude možné libovolně škálovat dle možností dalšího vývoje. V případě vlastní extrakce dat bychom si tedy vybrali dva až tři zástupce aplikací, navrhli pro ně jednoduché řešení extrakce dat a omezili počet sesbíraných výsledků na rozumnou hodnotu. Dle požadavků aplikace musíme zároveň splnit spodní limit více než 50 000 receptů. Vhodným kandidátem by jednoznačně byla zmíněná stránka Food.com, která v době psaní této práce obsahuje přes 500 000 receptů a pro cca 200 000 z nich máme k dispozici unikátní identifikátory skrze dataset z platformy Kaggle. Navíc dokumenty s recepty obsahují JSON-LD reprezentaci v hlavičce HTML. Pro každý recept se známým id by tedy stačilo vytvořit url, poslat na něj GET request a z HTML odpovědi extrahovat JSON-LD data. Podobně bychom mohli zpracovat recepty ze stránky Allrecipes, kde jsou v detailech receptů rovněž publikována JSON-LD data. Url receptů by mohl objevit přímo náš program během procházení stránky nebo bychom mohli využít nasbírané url adresy z datasetu Recipe1M+.

1.2.6 DBpedia

Znalostní graf DBpedia by se nám hodil pro extrakci rozšiřujících informací k ingrediencím nasbíraným z jednotlivých receptů. Prvním krokem by byla identifikace názvů ingrediencí z datasetu s recepty. Ideální by bylo mít k dispozici již extrahované ingredience, což není samozřejmostí, neboť recepty jsou často poskytovány bez strukturovaného textu surovin. Ten kromě názvu ingredience může obsahovat také její množství a jednotku měření. Není pak snadné spolehlivě určit, která část textu není součástí jména ingredience, zejména kvůli rozmanitým názvům jednotek měření. Nicméně i jednotek je jen konečné množství a s dostatečným úsilím bychom je měli být schopni identifikovat. Pro každou novou lokalizaci bychom ale problém řešili znova. Strukturovaným ingrediencím s odděleným názvem, množstvím a jednotkou měření bohužel nenahrává standardizovaný formát receptu dle ontologie Schema.org[8]. Ten definuje typ vlastnosti `recipeIngredient` jako prostý text, tedy včetně množství a jednotky.

Jakmile by se nám podařilo získat určitou skupinu jmen surovin, vytvořili bychom jednoduché entity ingrediencí v RDF formátu. Stačilo by každé surovině přiřadit unikátní IRI a vlastnost typu `rdfs:label` odpovídající názvu dané ingredience. Z těchto informací bychom sestavili RDF dataset a nahráli jej do aplikace

[8]https://schema.org/Recipe

Silk Workbench. Následně bychom provedli konfiguraci DBpedia SPARQL endpointu a nalezli shody s hodnotami `rdfs:label` ve znalostním grafu DBpedia. Abychom se vyhnuli prohledávání celého grafu, potřebujeme nastavit omezení na povolený typ entit. Po zběžné analýze konkrétních instancí surovin na DBpedia můžeme využít např. následující jednoduché omezení, kde proměnná `?a` znázorňuje dle konvence Silk Workbench hledanou entitu:

```
{
    {
        ?recipe <http://dbpedia.org/ontology/ingredient> ?a
    } UNION {
        ?a <http://dbpedia.org/ontology/ingredient> ?anotherIngredient
    }
}
```

Výše uvedený fragment dotazu cílí na všechny entity, které vystupují jako ingredience jiných entit a zároveň z opačného směru hledá všechny entity, které obsahují nějakou ingredienci. Z dat na DBpedia totiž můžeme vypozorovat, že obsahuje nejen základní ingredience, ale také suroviny složené z více přísad. Takové ingredience by bylo možné považovat za recept, nicméně i ony mohou být dále použity v rámci komplexnějšího postupu přípravy pokrmu. Dobrým příkladem složené ingredience je guacamole, které bývá často uváděno jako přísada (např. u hamburgerů), ale samo je produktem z avokáda, rajčat, cibule, česneku a limetky.

Dále si potřebujeme zvolit podmnožinu informací, které chceme extrahovat a uložit do vlastní databáze. Opět provedeme pozorování konkrétních instancí surovin a identifikujeme názvy vlastností pro jméno, popis, obrázek, nutriční hodnoty, kategorie a místo původu. Je třeba mít na paměti, že množství dat pro jednotlivé ingredience bude velmi proměnlivé a i jména vlastností nebo typ hodnot mohou být do určité míry odlišné. Do budoucna je zde prostor pro získání kvalitních dat pro rozdílné lokalizace aplikace, neboť ve znalostním grafu lze snadno filtrovat literály v požadovaném jazyce.

1.2.7 Wikidata

Práce se znalostním grafem Wikidata by probíhala velmi podobně jako u výše popsaného projektu DBpedia. Pomocí Silk Workbench bychom vytvořili vazby mezi entitami ingrediencí a následně extrahovali data k nalezeným přísadám. Ve srovnání s obsahem grafu DBpedia je zde většinou k dispozici menší množství textu a spíše je kladen důraz na odkazy do jiných zdrojů. Dle zběžného pozorování ale graf Wikidata často poskytuje relevantnější obrázky, kategorie a místo původu. Také mnohdy uvádí obsažené složky (např. smetanu a mléko u másla) a u vybraných ingrediencí zobrazuje barvu nebo dokonce Unicode znak.

Pokusíme se získat z DBpedia i Wikidata co nejvíce ze zmíněných informací a sloučit je dohromady. Tím bychom měli vylepšit poměr ingrediencí, ke kterým najdeme větší množství zajímavých informací. Dotaz na stejnou vlastnost dané entity může totiž na DBpedia i Wikidata dopadnout úplně jinak, přestože oba čerpají z projektu Wikipedia.

2. Architektura řešení

2.1 Příprava dat

2.2 Databázový model

2.3 Backend

2.4 Frontend

2.5 Aplikační logika

3. Implementace návrhu

3.1 Zpracování vstupních dat

3.2 Databáze Apache CouchDB

3.3 Vyhledávání pomocí Apache Solr

3.4 Rozhraní REST API

3.5 Middleware

3.6 Single-page aplikace

4. Testování

Závěr

Seznam použité literatury

[1] The World Wide Web Consortium. Semantic Web, 2015.

[2] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. *Knowledge Graphs*. Number 22 in Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool, 2021.

[3] Ching-Hua Chen, Deborah L. McGuinness, Nidhi Rastogi, Oshani Seneviratne, Sola Shirai, Ananya Subburathinam, and Mohammed J. Zaki. Foodkg: A semantics-driven knowledge graph for food recommendation.

[4] James David Moody. Categorizing Non-Functional Requirements Using a Hierarchy in UML. mathesis, East Tennessee State University, May 2003.

[5] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[6] Ed Finkler, Chris Shiflett, and Andreas Birkebæk. Open Recipes.

[7] Shuyang Li. Food.com Recipes and Interactions, 2019.

[8] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. Generating Personalized Recipes from Historical User Preferences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5976–5982, Hong Kong, China, November 2019. Association for Computational Linguistics.

[9] Boni García, Micael Gallego, Francisco Gortázar, and Mario Organero. A survey of the selenium ecosystem. *Electronics*, 9:1067, 06 2020.

[10] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. Browser fingerprinting: A survey. 05 2019.

Seznam obrázků

1.1 Obrazovka vyhledávání receptů pro desktopová zařízení 9
1.2 Obrazovka vyhledávání receptů pro mobilní zařízení 10

Seznam tabulek

A. Přílohy

A.1 První příloha