

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Individualizované hodnocení nemovitostí

Bakalářská práce

Pavel Šeps

Školitel: PhDr. Miloš Prokýšek, Ph.D.

České Budějovice 2018

Bibliografické údaje

Šeps, P., 2018: Individualizované hodnocení nemovitostí. [Personalized real estate evaluation. Bc. Thesis, In Czech.] – 37p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Cílem práce je vytvořit aplikaci, která bude schopna vyhledat nemovitosti v dané oblasti na základě požadavků uživatele na potřebu bodů zájmu v okolí nemovitosti. Aplikace porovná požadavky od uživatele se všemi nemovitostmi ve vybraném okolí a výstupem je doporučený seznam nemovitostí s procentuální shodou. Součástí práce je popis funkčnosti aplikace, dokumentace a rešerše stávajících řešení.

Annotation

The aim of this bachelor thesis is to develop an application which can search real estates in a selected area based on user requirements for points of interest in the area. The application compares user requirements with all real estates in the area and the output is list of recommended real estates with a percentage match. The thesis contains a description of the application, documentations and a research of existing solutions.

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne

.....

Pavel Šeps

Poděkování

Rád bych poděkoval panu PhDr. Miloši Prokýškovi PhD. za vedení mé bakalářské práce, cenné rady a odborný dohled. Také bych rád poděkoval svým blízkým za trpělivost a podporu při mých studiích.

Obsah

| | | |
|-------|--|----|
| 1 | Úvod..... | 1 |
| 2 | Popis problému a specifikace aplikace..... | 2 |
| 2.1 | Rešerše existujících portálů | 2 |
| 2.2 | Logický rámec | 3 |
| 2.3 | Funkční požadavky | 5 |
| 2.4 | Nefunkční požadavky | 5 |
| 2.5 | Scénáře použití aplikace | 6 |
| 3 | Realizace | 10 |
| 3.1 | Přehled vybraných technologií | 10 |
| 3.2 | Architektura systému | 11 |
| 3.3 | Zdroje dat..... | 12 |
| 3.3.1 | Sreality..... | 12 |
| 3.3.2 | TomTom..... | 12 |
| 3.3.3 | Google Maps | 13 |
| 3.3.4 | Openstreetmap..... | 14 |
| 3.3.5 | Databázové tabulky | 15 |
| 3.4 | Serverová část..... | 16 |
| 3.4.1 | Komponenty | 19 |
| 3.4.2 | Použité knihovny | 21 |
| 3.5 | Klientská část..... | 21 |
| 3.5.1 | Komponenty | 22 |
| 3.5.2 | Použité knihovny | 25 |
| 4 | Dokumentace..... | 27 |
| 5 | Testování..... | 28 |
| 6 | Závěr..... | 29 |
| 7 | Literatura | 30 |

| | | |
|-----------|------------------------------|----|
| Příloha A | Vizuální část aplikace | 31 |
| Příloha B | Testovací dokumentace | 35 |

1 Úvod

Tato bakalářská práce má za cíl vytvořit aplikaci, která bude schopna vyhledat nemovitosti v dané oblasti na základě požadavků uživatele na potřebu bodů zájmu v okolí nemovitosti – například potřeba pekárny, restaurace, obchodu, MHD zastávky, školy nebo parkoviště v blízkosti nemovitosti. Aplikace ohodnotí každou nemovitost ve vyhledávané oblasti a následně je porovná s požadavky od uživatele, zjistí procentuální shodu a zobrazí nejvíce shodné nemovitosti se seznamem shodných a neshodných parametrů. Aplikace je schopna získávat data z různých datových zdrojů a je navržena pro jednoduché přidání datových zdrojů nebo otázek.

2 Popis problému a specifikace aplikace

U většiny stávajících řešení nalezneme v detailu nemovitosti údaje o bodech zájmu, které se nacházejí v okolí, a ty jsou vypsané do seznamu nebo zobrazeny na mapě. Žádný z realitních portálů, popsaných v kapitole Rešerše existujících portálů, nenabízí možnost vyhledávání vůči okolí nebo zpracování dopravních dat.

Aplikace se odlišuje od stávajících řešení možností vyhledávání nemovitostí podle požadavků na okolí či možností vložení pozice na mapě, ke které se bude vyhodnocovat nejkratší vzdálenost i s ohledem na dopravu. Díky tomu, že aplikace získává data z více mapových a realitních portálů jsou data více různorodá a bohatší na informace než u stávajících řešení.

2.1 Rešerše existujících portálů

Sreality.cz [1] nabízejí vyhledávání nemovitostí podle okresu, města, adresy nebo poloměru vzdálenosti od zadané adresy. V detailu nemovitosti nalezneme seznam bodů zájmů, jako jsou zastávky MHD, vlak, bankomat, pošta, lékárna, sportoviště, restaurace, obchod a škola. Ke každé kategorii vypisují nejbližší bod zájmu a popisují ho názvem a vzdáleností.

Na realitním portálu RE/MAX Česká republika [2] se dají vyhledávat nemovitosti v okolí zadané adresy, ale další informace o okolí nezískáme.

Na německém realitním portálu Immowelt [3] můžeme vyhledat nemovitost pomocí města a městských částí, popřípadě i ulice. V detailu nemovitosti nalezneme mapu s její polohou, do které můžeme přidat pozice pekáren, dětských hřišť, řeznictví, lékaře, lékáren, škol, restaurací a obchodů. Dále můžeme vyhledat další body zájmu pomocí řetězce. Všechna data získávají z Gelbeseiten [4]. Realitní portál Immobilienscout24 [5], na rozdíl od ostatních realitních portálů, nabízí například ještě pokrytí mobilním signálem a rychlostí internetu od Telekom Deutschland [6], kriminalitu a místa s bezbariérovým přístupem.

2.2 Logický rámec

| Cíl projektu | Objektivně ověřené ukazatele | Prostředky ověření | |
|--|--|---|---|
| <p><i>Zjednodušení vyhledávání nemovitostí.</i></p> <p><i>Inovativní systém vyhledávání nemovitostí.</i></p> | <p><i>Systém je hodnocen lépe než stávající referenční systémy (...)</i></p> | <p><i>Kvalitativní / Kvantitativní studie</i></p> | |
| Účel projektu | Objektivně ověřené ukazatele | Prostředky ověření | Předpoklady |
| <p><i>Vytvoření modulárního nástroje pro hodnocení nemovitostí.</i></p> | <p><i>Aplikace bude schopna vyhodnocovat data z rozhraní aplikace.</i></p> | <p><i>Výsledky testování.</i></p> <p><i>Seřazení nemovitostí dle požadavků.</i></p> | <p><i>Jednoduché použití aplikace.</i></p> |
| Výstupy | Objektivně ověřené ukazatele | Prostředky ověření | Předpoklady |
| <p><i>Webová aplikace zobrazující nejvíce shodné nemovitosti se zadanými daty uživatele.</i></p> <p><i>Rozhraní pro získání dat z více webových portálů.</i></p> | <p><i>Aplikace je intuitivní</i></p> <p><i>Možnost použití aplikace do konce dubna 2018.</i></p> <p><i>Aplikace bude schopna pracovat s více datovými zdroji.</i></p> <p><i>Možnost obsluhování aplikace</i></p> | <p><i>Kvalitativní studie.</i></p> <p><i>Výsledky testování</i></p> | <p><i>Získání dostatku a rozmanitosti dat pro posuzování.</i></p> |

| | | | |
|---|---|--|--|
| | <i>z webového prohlížeče.</i> | | |
| Činnosti | Objektivně ověřené ukazatele | Prostředky ověření | Předpoklady |
| <i>Analýza možnosti získání dat.</i> <i>Analýza a zadávací dokumentace</i> <i>Dokumentace projektu.</i> <i>Vytvoření rozhraní pro příjem dat.</i> <i>Vytvoření webové aplikace.</i> <i>Testování.</i> <i>Uživatelský a administrační manuál</i> | <i>Odborné poradenství.</i> <i>Dodržování harmonogramu postupu vývoje.</i> | <i>Prosinec 2017 – Analýza, znalost vhodných technologií a Dokumentace.</i> <i>Únor 2018 - Vytvoření rozhraní.</i> <i>Březen 2018 – Vytvoření webové aplikace.</i> <i>Duben 2018 – Testování.</i> | <i>Výběr vhodných technologií.</i> <i>Znalost získání dat z webových portálů.</i> |

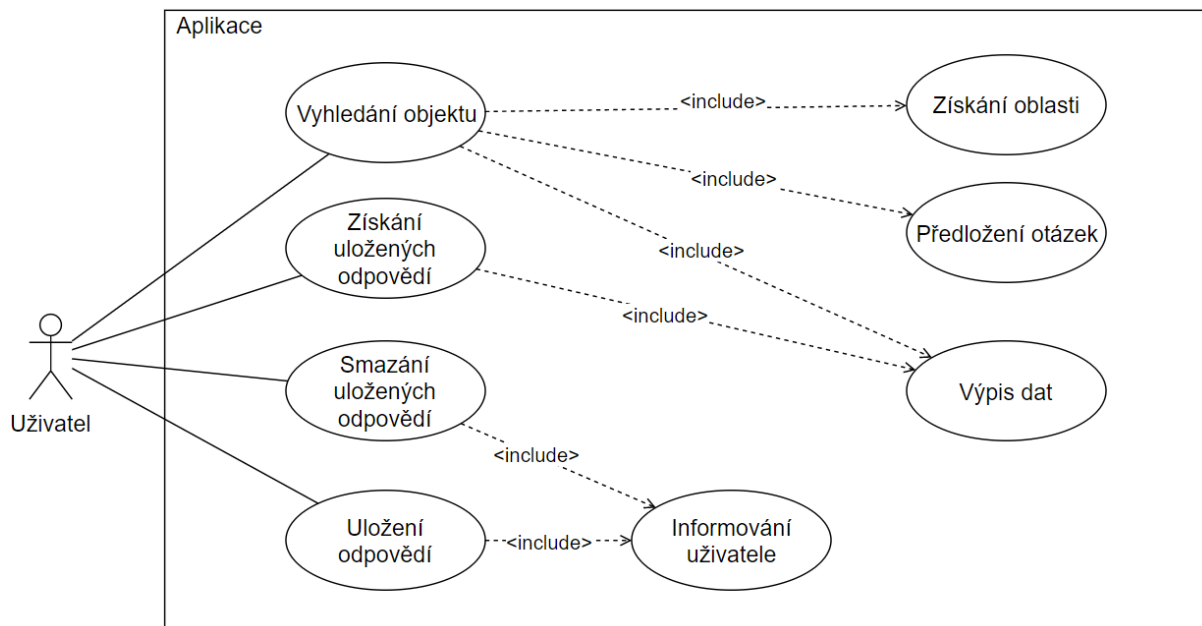
Tabulka 1: Logický rámec projektu

2.3 Funkční požadavky

1. Aplikace bude vyhledávat nejvhodnější nemovitosti na základě informací z otázek získaných od uživatele.
2. Aplikace bude získávat údaje o bodech zájmu z:
 - a. Google Maps
 - b. Openstreetmap
 - c. Možnost vlastního vložení
 - d. Databázové tabulky
3. Aplikace bude získávat údaje o nemovitostech z:
 - a. Možnost vlastního vložení
 - b. Databázové tabulky
4. Aplikace bude schopna zjišťovat hustotu dopravy v dané oblasti.
5. Aplikace bude zjišťovat dobu jízdy autem do jiného vybraného bodu na mapě.
6. Aplikace bude předkládat uživateli otázky a možnosti odpovědí, které budou definovány programátorem pro danou kategorii dat.
7. Aplikace bude předkládat otázky pouze pro data, které je schopna získat v dané oblasti.

2.4 Nefunkční požadavky

1. Aplikace bude ve formě webového UI.
2. Aplikace bude komunikovat pomocí REST API.
3. Webové rozhraní bude plně responzivní.
4. Aplikace bude přístupná veřejnosti.
5. Architektura aplikace bude modulární, dále rozšiřitelná.
6. Aplikace bude chráněna autorským právem.



Obrázek 1: Diagram užití

2.5 Scénáře použití aplikace

Vyhledání objektu

Ohodnocování všech objektu ze získaných kritérií.

Vstupní podmínky

Uživatel zodpověděl předložené otázky a vybral oblast.

Základní scénář

1. Aplikace zpracuje získaná data od uživatele a porovná s nemovitostmi v oblasti.

Alternativní scénář

1. Aplikace nenalezla žádné nemovitosti v dané oblasti a upozorní uživatele.

Výstupní podmínky

Aplikace má zpracovaná data pro vykreslení.

Získání oblasti

Zadání adresy a poloměr od adresy.

Základní scénář

1. Systém vygeneruje vstup a mapu.
2. Uživatel vybere pozici na mapě, nebo napíše do vstupu adresu, popřípadě souřadnice.
3. Aplikace ověří, zdali je schopna nalézt místo na mapě.
4. Systém vygeneruje možnosti na výběr poloměru

5. Možnosti:
 - a. 0,5 km
 - b. 1 km
 - c. 2 km
 - d. 5 km
 - e. 10 km
6. Uživatel si vybere jednu z možností.

Alternativní scénář

1. Uživatel zadal adresu, kterou není aplikace schopna nalézt na mapě.
2. Aplikace upozorní uživatele na neplatnou adresu a postup se vrací k bodu 2. základního scénáře.
 1. Uživatel nezadá adresu.
 2. Aplikace upozorní uživatele, aby zadal adresu

Podmínky pro dokončení

Uživatel zadal adresu, kterou aplikace byla schopna naleznout na mapě a vybere poloměr z možností.

Předložení otázek

Uživateli budou předkládány otázky pro danou oblast, které následně ohodnotí.

Vstupní podmínky

Uživatel zadal adresu a vybral poloměr.

Základní scénář

1. Aplikace předloží otázku uživateli pro danou kategorii s výběrem možností.
2. Uživatel vybere jednu z možností.
3. Aplikace se vrací k prvnímu bodu, dokud nevyčerpá otázky v dané oblasti.
4. Aplikace zobrazí seznam všech zodpovězených otázek.
5. Uživatel může vybrat až 3 otázky.
6. Aplikace bude vybraným otázkám přiřadit vyšší váhu.

Alternativní scénář

1. Aplikace nebude mít žádné otázky pro zobrazení a upozorní uživatele.

Podmínky ukončení

Uživatel vybere odpovědi pro všechny otázky v dané oblasti.

Získání uložených odpovědí

Získání uloženého seznamu z databáze.

Vstupní podmínky

Uživatel přistoupí pomocí vygenerovaného jedinečného odkazu pro zobrazení.

Základní scénář

1. Aplikace nalezne záznam v databázi.

Alternativní scénář

1. Záznam v databázi neexistuje.
2. Aplikace upozorní uživatele.

Výstupní podmínky

Aplikace má data z databáze pro vykreslení.

Výpis dat

Předložení dat uživateli.

Vstupní podmínky

Aplikace má data pro vykreslení.

Základní scénář

1. Aplikace vykreslí získaná data uživateli.

Smazání uložených odpovědí

Smazání uloženého seznamu z databáze

Vstupní podmínky

Uživatel přistoupí pomocí vygenerovaného jedinečného odkazu pro smazání.

Základní scénář

1. Aplikace se dotáže uživatele, jestli si opravdu přeje smazat tento záznam.
2. Uživatel potvrdí.
3. Aplikace vymaže daný záznam z databáze a informuje uživatele.

Alternativní scénář

1. Záznam v databázi neexistuje.
2. Aplikace upozorní uživatele.

Výstupní podmínky

Aplikace smazala záznam z databáze.

Uložení odpovědí

Uložení zobrazeného seznamu do databáze

Vstupní podmínky

Uživatel má zobrazený výsledný seznam nemovitostí.

Základní scénář

1. Aplikace předloží vstup pro zadání emailu.
2. Uživatel zadá platný email.
3. Aplikace uloží seznam do databáze a informuje uživatele s odkazem pro zobrazení a smazání.

Alternativní scénář

1. Uživatel zadal nevalidní email.
2. Aplikace upozorní uživatele a vrací se k bodu 1 v hlavním scénáři.

Výstupní podmínky

Aplikace uložila seznam do databáze.

Informování uživatele

Aplikace informuje uživatele pomocí emailu.

Vstupní podmínky

Aplikace má data pro informování.

Hlavní scénář

1. Aplikace odešle email na danou emailovou adresu s danými daty.

3 Realizace

Celá aplikace byla postupně vyvíjena po malých komponentách, které byli postupně rozšiřovány. Samotná aplikace byla zpočátku funkční jenom z konzole. Po naprogramování základní logiky, byla aplikace napojena na HTTP API¹, kde už byla potřeba i vizuální část. Vizuální část byla pouze pro rychle testování a vůbec se neshodovala s finálním vzhledem. Během pokročilého vývoje byla rozšiřovaná serverová část.

Při finální podobě serverové části se vývoj přesunul na tvorbu vizuální části. Autor pro lepší představu jako první nakreslil několik vzhledů a vybral ty, které byly nejvíce uživatelsky přívětivé. Poté následovalo samotné programování vizuální části.

Celý projekt je od počátku verzovaný² pomocí nástroje git³, kde je vidět historie vývoje aplikace nebo i samotný finální kód. V repozitáři jsou popsány kroky pro instalaci.

3.1 Přehled vybraných technologií

Celá logika aplikace a zpracování dat se odehrává na serverové straně, která je napsána v programovacím jazyku C#. Jedná se o kompilovaný objektově orientovaný programovací jazyk třetí generace. Disponuje širokou škálou možnosti použití od konzolové aplikace, až po webové aplikace s frameworkem .NET⁴, který je využitý v této práci. Webová aplikace běží na IIS serveru⁵, který je také produktem firmy Microsoft.

Aplikace komunikuje s relační databází Microsoft SQL. Pro komunikaci s databázovým serverem aplikace využívá objektově-relační mapování s použitím Entity frameworku⁶. Tento framework usnadňuje práci s databázovým serverem v C#.

Pro vizualizaci všech dat je využit javascriptový framework Vue.js⁷. Jedná se o jeden z nejvíce používaných javascriptových frameworků v dnešní době. Usnadňuje tvorbu webového frontendu, kdy sjednocuje HTML⁸, javascript a css⁹ do samostatně funkčních

¹ Definice komunikačního rozhraní na straně serveru s vizuální částí.

² https://bitbucket.org/pavel_seps/bakalarka

³ Systém správy verzí

⁴ <https://www.microsoft.com/net/>

⁵ Webový server od firmy Microsoft pro ASP.NET

⁶ [https://msdn.microsoft.com/en-us/library/aa937723\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx)

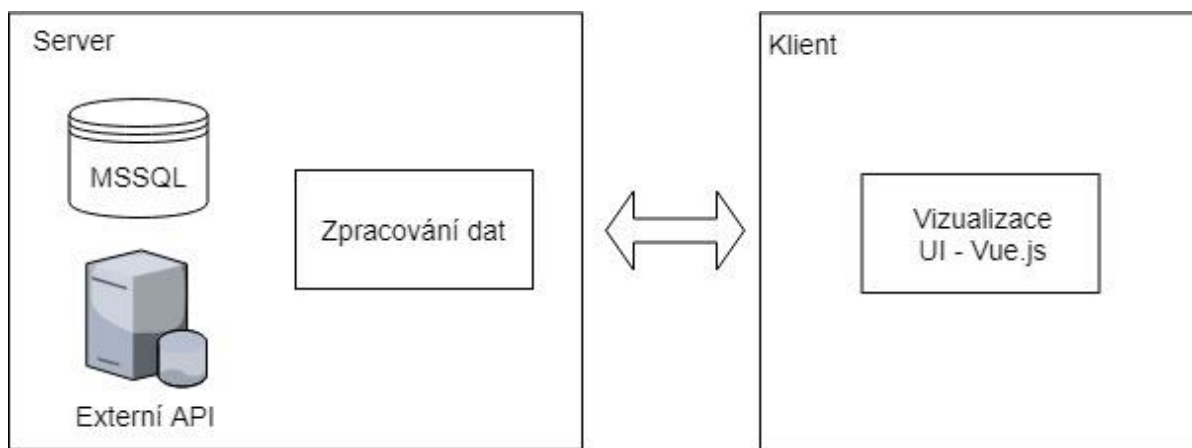
⁷ <https://vuejs.org/>

⁸ HyperText Markup Language

⁹ Cascading Style Sheets

komponent. Vue.js je vhodné pro zobrazení dat, protože jakékoliv změny v datech automaticky zobrazí.

3.2 Architektura systému



Obrázek 2: Architektura systému

Autor se snažil navrhnout aplikaci podle populární softwarové architektury MVC¹⁰, kdy je snaha rozdělit aplikaci do 3 hlavních segmentů, ale v této situaci se nedá přímo aplikovat. Část view („V“) se nedá použít, protože vizuální část je interpretována ve frameworku Vue.js a nejedná se o typ aplikace, která by generovala celé HTML.

Na základě tohoto problému je lepší rozdělit aplikaci na serverovou a klientskou část. Serverová strana aplikace zajišťuje veškerou práci s daty a zpracování dat. Na straně klientské není pouze interakce s uživatelem a zobrazení dat, ale je sem i přesunuta základní logika aplikace, kterou je zde možné zajistit za pomoci javascriptového frameworku, snížit tak zátěž serveru a zrychlit celou aplikaci.

Tyto dvě části aplikace společně komunikují pomocí HTTP API, kde si předávají data ve formátu JSON¹¹.

¹⁰ Model-View-Controller

¹¹ Datový typ určený pro jednoduchou komunikaci mezi programy. (JavaScript Object Notation)

3.3 Zdroje dat

3.3.1 Sreality

Sreality.cz je jeden z největších realitních portálů v České republice, jak bylo zjištěno z rešeršního výzkumu. Obsahuje tisíce realit určených k prodeji nebo k pronájmu po celé České republice. Na základě analýzy funkčnosti portálu aplikace využívá stejných požadavků na API server jako dotazuje vizuální rozhraní Sreality.

V prvním kroku aplikace dotáže na získání nemovitostí v okolí. Tento dotaz provede pro prodej¹² a pronájem¹³ nemovitostí. Aplikace získá id daných nemovitostí a souřadnice. Po získání všech nemovitostí aplikace dotáže na detail¹⁴ dané nemovitosti, kde získá název, cenu, popis a další informace o nemovitosti.

Aplikace se chová jako uživatel na portálu, který si chce prohlédnout zadanou oblast na maximální přiblížení a následně si chce prohlédnout všechny detaily nemovitostí.

3.3.2 TomTom

Firma TomTom je výrobce navigačních systémů pro automobily. Na základě používání jejich zařízení firma poskytuje API¹⁵, které se dá využít například pro vyhledávání míst, tras, zobrazení map a zjištění aktuální dopravy.

Aplikace využívá data právě pro zjištění aktuální dopravy v daném okolí, aby byla schopná uživateli poskytnout důležité informace.

Dotazem¹⁶ na API server, kterému se předají souřadnice a API klíč¹⁷, se aplikace dozví, jaká je aktuální průměrná a maximální rychlost v okolí. Poměrem těchto dvou hodnot aplikace zjistí, jak hodně doprava dosahuje maximální rychlosti.

Tyto data se nezapočítávají do hodnocení, jsou pouze informativní.

¹²

http://www.sreality.cz/api/cs/v2/clusters?leftBottomBounding={leftBottomLng}%7C{leftBottomlat}&rightTopBounding={rightTopLng}%7C{rightToplat}&zoom=18&category_main_cb=1

¹³

http://www.sreality.cz/api/cs/v2/clusters?leftBottomBounding={leftBottomLng}%7C{leftBottomlat}&rightTopBounding={rightTopLng}%7C{rightToplat}&zoom=18&category_main_cb=2

¹⁴ <http://www.sreality.cz/api/cs/v2/estates/{id}>

¹⁵ <https://developer.tomtom.com/tomtom-maps-apis-developers>

¹⁶ <https://api.tomtom.com/traffic/services/4/flowSegmentData/absolute/10/json?key={key}&point={lat},{lng}>

¹⁷ Identifikační klíč

3.3.3 Google Maps

Google Maps je jako mapová aplikace celosvětově rozšířená. Obsahuje mapová data a informace o okolí z celého světa, na které se dá dotazovat pomocí HTTP API požadavků. Je to hlavní zdroj informací o bodech zájmu v okolí pro aplikaci.

Aplikace dotazuje¹⁸ pro informace z různých kategorií ve vybrané oblasti, které jsou následně přiřazeny k dané otázce, jak je popsáno v Tabulka 2: použití kategorií z Google maps pro otázky – vždy jeden požadavek na server za každou kategorií.

| Typ dotazu | Typ otázky |
|---------------|------------|
| bakery | Pekárna |
| restaurant | Restaurace |
| supermarket | Obchod |
| store | Obchod |
| supermarket | Obchod |
| shopping_mall | Obchod |
| bus_station | Zastávka |
| school | Škola |
| pharmacy | Lékárna |
| post_office | Pošta |
| parking | Parkoviště |
| park | Park |
| library | Knihovna |
| bar | Bar |
| cafe | Kavárna |
| church | Kostel |
| gym | Posilovna |

Tabulka 2: použití kategorií z Google maps pro otázky

¹⁸

<https://maps.googleapis.com/maps/api/place/nearbysearch/json?key={key}&language=cs&location={lat},{lng}&radius={radius}&type={type}>

3.3.4 Openstreetmap

OpenStreetMap je volně dostupná mapová databáze, která je spravována komunitou. Poskytuje velké využití díky otevřenosti, ale hlavně poskytuje možnost dotazovat na HTTP API server pro získání bodů zájmu z okolí.

Jedná se spíše o záložní zdroj dat, ale poskytuje také kategorie, které například Google Maps neposkytují. Díky spravování dat komunitou zde může být přidán nový bod zájmu dříve než na Google Maps.

Dotaz¹⁹ na API server se formuluje pomocí xml²⁰ formátu, který se následně připojí do parametru v url. Kategorie se skládají z klíče a hodnoty – kategorie se dá upřesnit vyplněním obou hodnot, nebo je možné dotázat pouze na klíč a data jsou více různorodá, ale podobná. Seznam použitých klíčů a hodnot navázaných na otázky je popsán v Tabulka 3: použití kategorií z OpenStreetMap pro otázky.

Ukázka xml dotazu na API server:

```
<osm-script output="json">
  <query type="node">
    <around lat="{lat}" lon="{lng}" radius="{radius}" />
    <has-kv k="{key}" v="{value}" />
  </query>
  <print e="" from="_" geometry="skeleton" limit=""
    mode="body" n="" order="id" s="" w="" />
</osm-script>
```

| Klíč | Hodnota | Typ otázky |
|---------|-------------|------------|
| shop | bakery | Pekárna |
| amenity | restaurant | Restaurace |
| shop | mall | Obchod |
| shop | supermarket | Obchod |
| highway | bus_stop | Zastávka |
| amenity | school | Škola |

¹⁹ <http://overpass-api.de/api/interpreter?data={xmlDotaz}>

²⁰ eXtensible Markup Language – obecný značkovací jazyk

| | | |
|---------|------------------|------------|
| amenity | pharmacy | Lékárna |
| amenity | post_office | Pošta |
| amenity | parking | Parkoviště |
| amenity | parking_entrance | Parkoviště |
| leisure | park | Park |
| leisure | playground | Park |
| amenity | library | Knihovna |
| amenity | bar | Bar |
| amenity | nightclub | Bar |
| amenity | pub | Bar |
| amenity | cafe | Kavárna |
| amenity | place_of_worship | Kostel |
| amenity | fitness_centre | Posilovna |

Tabulka 3: použití kategorií z OpenStreetMap pro otázky

3.3.5 Databázové tabulky

Aplikace je napojena na databázi MSSQL, do které je možné vkládat vlastní záznamy bodů zájmu a nemovitostí.

Tyto záznamy jsou následně přidány do zpracování. Tento způsob má spíše využití pro testovací účely. Aplikace je navržena tak, aby hlavním způsobem bylo získávání dat z externích datových zdrojů.

Pro práci s databází aplikace využívá Entity framework s použitím code first metody. Nejdříve se nadefinují třídy a vztahy mezi třídami. Následně Entity framework na základě tříd vytvoří strukturu relační databáze. Při potřebě změny struktury databáze framework nabízí změnu pomocí migrací, kdy následně framework všechny změny promítne do relační databáze. Entity framework zajišťuje také ORM²¹, které je programátorsky více přívětivé.

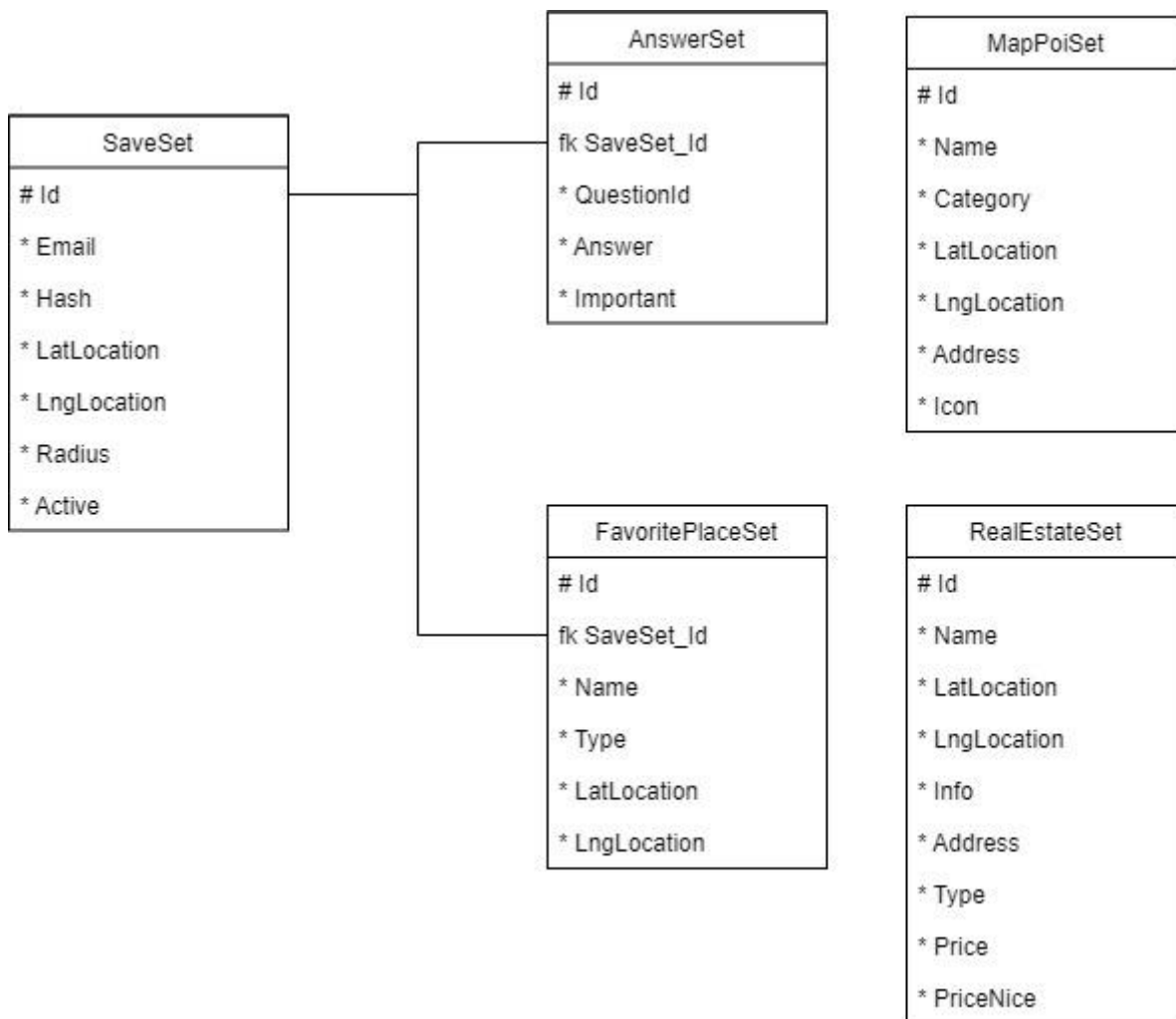
²¹ Objektově-relační mapování

3.4 Serverová část

Serverová část aplikace zajišťuje veškerou logiku pro ohodnocení nemovitostí, tudíž je to nejdůležitější část aplikace, na kterou je tato práce zaměřena. Server komunikuje s okolím pomocí HTTP API požadavků, které jsou popsány v kapitole Dokumentace HTTP API požadavků na server. Aplikace je nezávislá na tomto API. Celý chod aplikace řídí jedna třída, která může být implementována do jiných aplikací nebo využít jiná komunikační rozhraní.

Díky návrhu aplikace, kde je logika výpočtu aplikace naprosto nezávislá na způsobu použití je jednoduché i tento kód upravit nebo rozšířit pro nějaké specifitější použití.

Aplikace využívá napojení na databázi MSSQL za pomoci Entity frameworku. Do databáze může uživatel přidat bod zájmu nebo nemovitost, které jsou přidány do ohodnocování. Tento případ zadání dat pro ohodnocování je spíše pro testovací účely, protože hlavním zdrojem dat pro aplikaci jsou externí data, případné použití může být pro místa nezmapovaná externími zdroji.



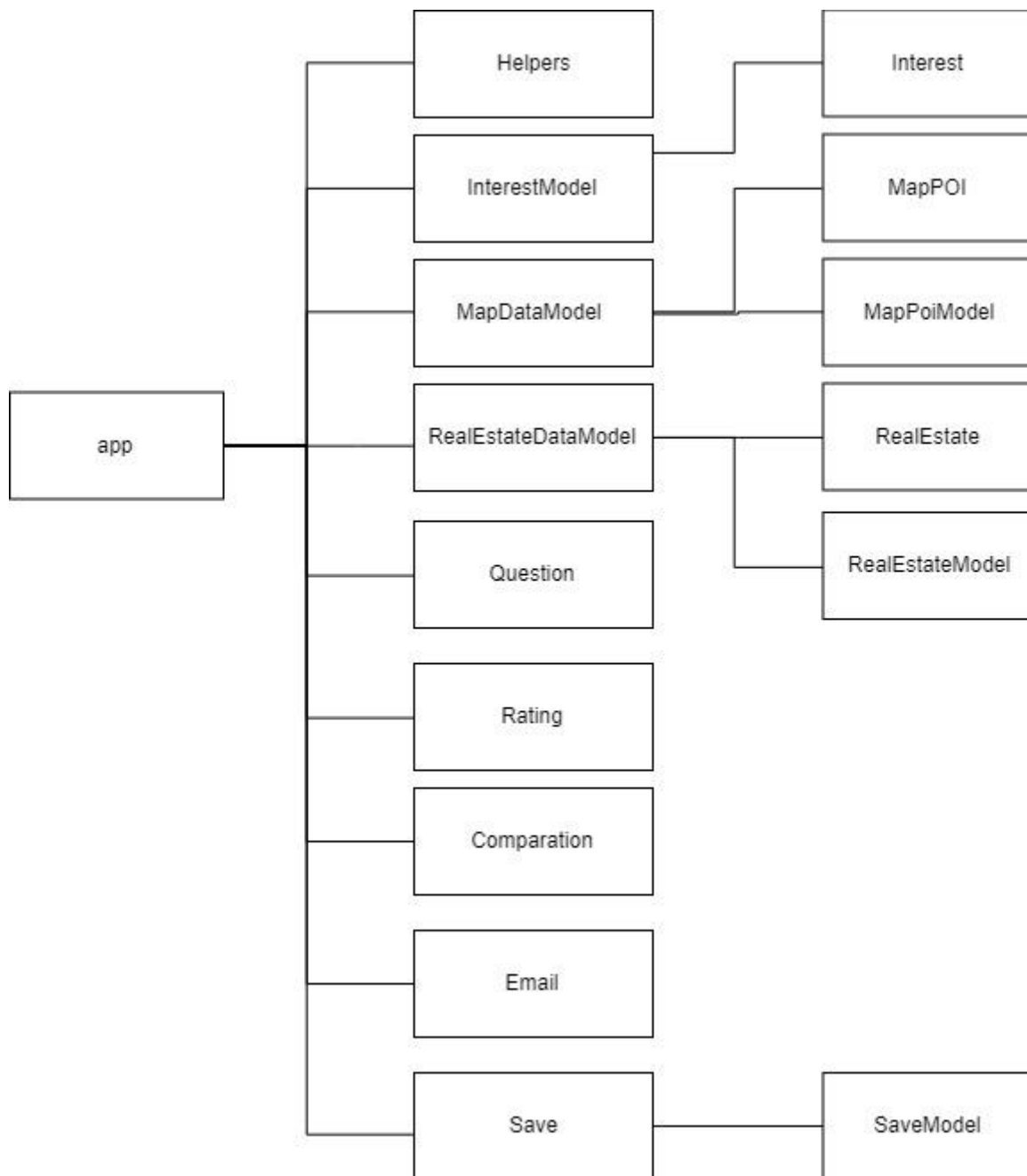
Obrázek 3: Databázový návrh

Do databáze si může uživatel uložit svůj seznam vyhledávání, kde uživateli přijde na email jedinečná url pro znovunačtení uloženého seznamu. Aplikace do databáze uloží seznam odpovědí, oblíbená místa a lokaci s poloměrem, jak je popsáno v Obrázek 3: Databázový návrh. Díky těmto datům se jedná spíše o uložení profilu vyhledávání než výsledku, protože při načtení aplikace vše znovu vyhledává.

Aplikace je připravena na velké množství dat pro ohodnocení. Díky tomu je celý chod aplikace asynchronní pro co nejrychlejší dosažení výsledku. Aktuálně aplikace dotazuje na 3 API servery²² a do vlastní databázové tabulky. Během ohodnocování jsou požadavků na externí datové zdroje desítky a data pro zpracování se pohybují ve stovkách až tisících záznamů. Díky více datovým zdrojům je aplikace schopna pracovat, i kdyby jeden ze zdrojů nebyl funkční. Aplikace je navržena tak, aby přidání nového datového zdroje bylo co

²² Google maps, Openstreetmap, Sreality.cz

nejpohodlnější a díky tomu má aplikace v budoucnu potenciál na rozšíření o další datové zdroje, popřípadě i specifická data jen pro jednu část oblasti. Typ datového zdroje je nepodstatný, je na programátorovi, jak bude zdroj dat implementovat.



Obrázek 4: Struktura aplikace

3.4.1 Komponenty

App

Hlavní třída celé aplikace, která zajišťuje celý chod aplikace. Získává a udržuje potřebná data a vytváří instance dalších tříd potřebných pro zpracování.

Helpers

Třída, která implementuje pomocné metody, které se využívají po celém projektu. Obsahuje metodu `GetLocationByDistance`, která na základě souřadnic, vzdálenosti a úhlu určí souřadnice [7].

InterestModel

Jedná se o třídu, která vytvoří instance všech tříd, které jsou v daném jmenném prostoru a implementují správný interface. Všechny tyto instance představují datový zdroj pro získání zajímavostí v daném místě. Výstupem je seznam instancí `Interest`, které se aplikují do výsledku.

Interest

Tato třída může obsahovat zajímavosti o daném místě. Nezapočítává se do ohodnocení nemovitosti. Například v aplikaci je využita pro popsání dopravy v okolí, ale je možné ji rozšířit například o kriminalitu, sílu mobilního signálu, rychlost internetu a další.

MapDataModel

Model, který vytváří seznam instancí otázek v daném jmenném prostoru a rovnou získává všechny body zájmu pro danou otázku v oblasti.

MapPOI

Datová struktura pro body zájmu na mapě, která obsahuje souřadnice, název a další potřebné informace.

RealEstateDataModel

Datový model, který vytváří instance tříd v daném jmenném prostoru a získává seznam nemovitostí z dané instance.

RealEstate

Datová struktura pro nemovitosti, která obsahuje souřadnice, název, cenu, adresu a další potřebné informace.

Question

Abstraktní třída, která obsahuje všechny potřebné informace pro otázky, jako je text otázky, odpovědi, ale třeba i body zájmu dané otázky.

Rating

Na této třídě závisí celé ohodnocení nemovitosti. Metoda, která ohodnocuje nemovitosti vůči bodu zájmu, zjistí pro danou nemovitost a otázku nejbližší bod zájmu. Poté pro danou otázku se všechny nemovitosti seřadí od nejkratší vzdálenosti po nejdelší. Následně se tento seznam rozdělí do kategorií. Počet kategorií se rovná počtu odpovědí pro danou otázku. Algoritmus postupuje pro všechny otázky.

Součástí této třídy je také ohodnocení oblíbených míst na mapě. Tato metoda funguje na podobném principu jako ohodnocení bodů zájmu, ale kategorie jsou pevně nastaveny na tři. Není použita vzdušná vzdálenost, ale reálná vypočtená trasa podle Google maps. Pokud je vybrána možnost cesty automobilem, tak je započítána i doprava.

Comparison

Comparison třída porovnává ohodnocenou nemovitost s odpověďmi od uživatele. Porovnávací algoritmus prochází všechny body ohodnocení pro každou nemovitost. Vždy porovná, zdali se dané ohodnocení shoduje s odpovědí od uživatele, pokud ano, jsou za tento bod připočítána procenta. Pokud byl bod označený jako důležitý, tak je připočítáno 4/3 procent bodu. Oblíbené místo na mapě má stejnou hodnotu jako bod zájmu.

Procentuální hodnota bodu se určí poměrem 100 ku celkovému počtu otázek a oblíbených míst.

Email

Třída pro odeslání emailu, která se využívá pro ukládání.

Save

Třída, která zajišťuje zpracování dat pro uložení záznamu do databáze a také následné načtení dat z databáze do hlavní třídy ke zpracování.

SaveModel, MapPoiModel a RealEstateModel

Databázové modely pro práci s databází.

3.4.2 Použité knihovny

Entity framework

Framework pro pohodlnější práci s databází, který poskytuje ORM a možnost práce s databází pomocí code first metody, kdy se nejdříve nadefinují třídy pro data, která chceme ukládat do databáze a Entity framework následně vygeneruje databázovou strukturu.

Microsoft.AspNet.WebApi

Balíček, který je potřeba pro jednoduchou práci s HTTP API požadavky na IIS serveru.

Newtonsoft.Json

Populární knihovna pro zpracování datového typu JSON, která ho převede z textové podoby na daný objekt a obráceně.

3.5 Klientská část

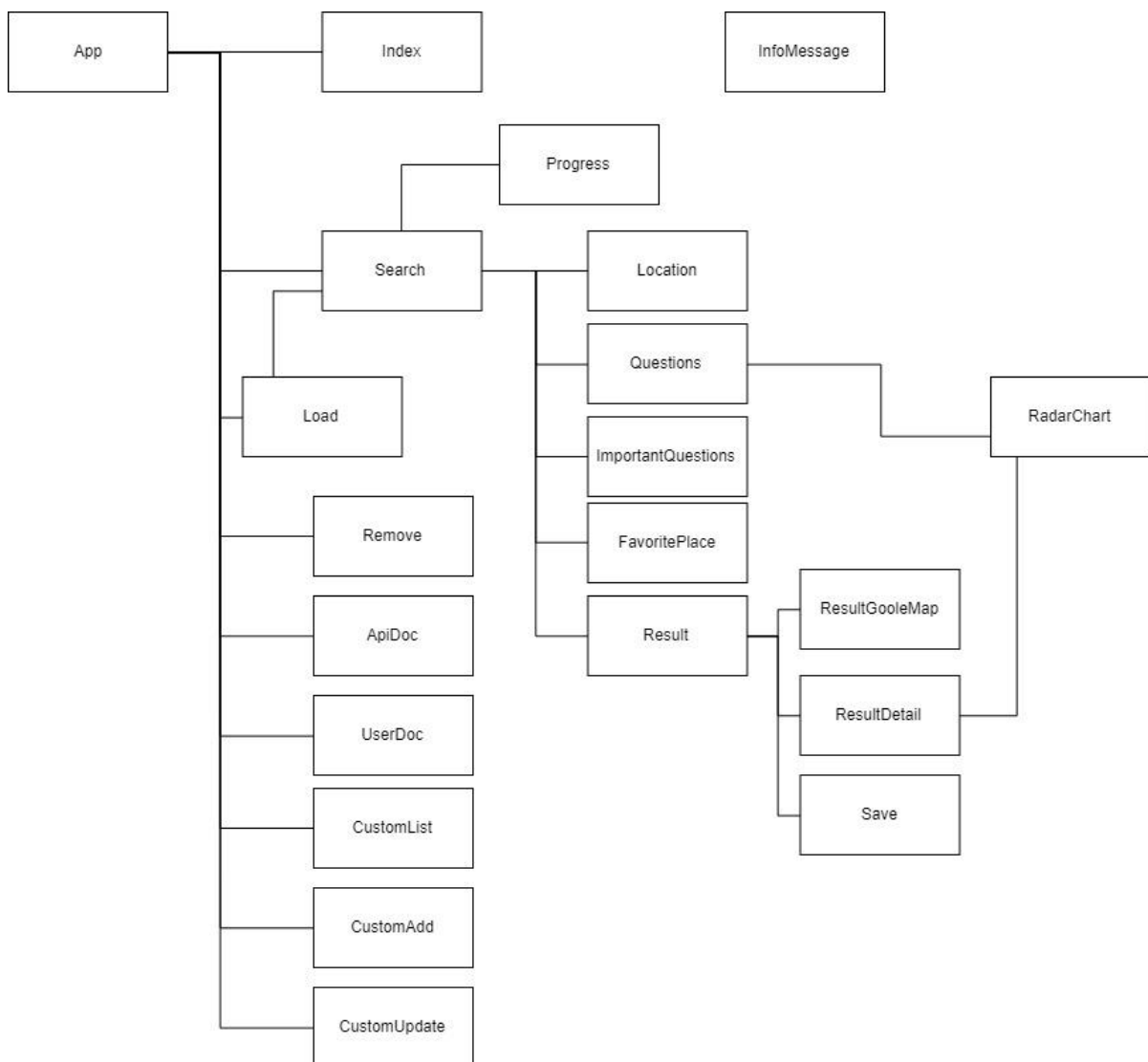
Vizuální rozhraní aplikace je tvořeno za pomoci javascriptového frameworku Vue.js. Vue.js nabízí možnost tvorbu aplikace za pomoci komponent pro větší přehlednost kódu a lepší použitelnost. Pro každou komponentu je nadefinované HTML a javascript. HTML zajišťuje strukturu dané komponenty a pomocí javascriptu se do HTML promítají potřebná data k zobrazení. Zároveň zde může být prováděna jednoduchá logika.

Cílem použití těchto javascriptových frameworku je zrychlit uživatelské stránky a to tím, že část logiky aplikace se přesune do view a nemusí se čekat na odpovědi serveru. Další veliké zrychlení je díky tomu, že se stránka načte pouze jednou při příchodu. Při jakékoliv změně se upraví pouze daná komponenta a nemusí se čekat na server na novou stránku.

Komponenty se dají do sebe vnořovat a díky tomu vzniká stromová struktura, která je k vidění na obrázku Obrázek 5: strom Vue.js komponent.

Pro vizuální ostylevání aplikace je použit jazyk scss (sass), kde se dá psát css s možností použití proměnných, funkcí a vnořování stylů. Scss soubory se následně musí kompilovat do typu css pomocí kompilátoru. Pro usnadnění tvorby stylů je použit Bootstrap framework²³, který nabízí předdefinované styly, jenž jsou poté upraveny do finální podoby aplikace.

²³ <http://getbootstrap.com/>



Obrázek 5: strom Vue.js komponent

3.5.1 Komponenty

App

Hlavní komponenta, která zajišťuje spuštění aplikace, jako je třeba načtení knihoven a inicializaci Vue.js routeru.

Index

První textová stránka, která je uživateli zobrazena, kde nalezne základní informace o aplikaci a může přejít k vyhledávání.

Search

Tato komponenta zajišťuje libovolnou komunikaci se serverem pro vyhledávání nemovitosti. Následně rozhoduje, jaká komponenta bude zobrazena během vyhledávání, a poskytuje získaná data vnořeným komponentám pro zobrazení.

Load

Komponenta Load je využita, pokud si uživatel přeje zobrazit uložený seznam. Při načtení se dotáže serveru, jestli daný záznam existuje. Pokud ano, stáhne všechna potřebná data a ty následně předá komponentě Search.

Remove

Je to samostatná stránka, která je použita, pokud chce uživatel smazat uložený záznam.

ApiDoc a UserDoc

Stránky, ve kterých je popsána dokumentace. V ApiDoc jsou zdokumentované API požadavky na server a v UserDoc je popsána uživatelská dokumentace, jak aplikaci správně použít.

CustomList, CustomAdd a CustomUpdate

Stránky, které jsou použity pro práci s body zájmu a nemovitostmi uloženými v databázi. Pomocí těchto stránek je možné nechat si vypsat záznamy, přidat nový nebo upravit.

InfoMessage

Tato komponenta je použita v každé vnořené komponentě Search, ale není na ničem závislá, tudíž se dá použít kdekoli jinde. Jedná se o šipky, které se zobrazí po kliknutí na „i“ pro nápovědu uživateli.

Progress

Komponenta zobrazená na vrchu stránky během vyhledávání. Znázorňuje uživateli jeho postup ve vyhledávání.

Location

První komponenta ve vyhledávání, která zobrazí uživateli mapu a místo pro zadání adresy, popřípadě souřadnic a poloměru. Komponenta je určena pro získání informace od uživatele, v jaké oblasti si přeje vyhledávat.

Questions

V této komponentě jsou uživateli postupně předkládány otázky, pro kterou může vybrat odpověď, nebo nemusí odpovídat.

ImportantQuestions

Zde je zobrazený seznam zodpovězených otázek, kde uživatel může vybrat maximálně 3 otázky, které se označí jako důležité.

FavoritePlace

Komponenta, která zobrazí uživateli mapu, kde si může vybrat pro něj důležitá místa na mapě. U těchto míst může specifikovat, jestli se do nich dopravuje automobilem nebo pěšky.

Result

Poslední komponenta v procesu vyhledávání, která vizuálně zpracovává výsledek. Výsledek se dá filtrovat dle ceny a procentuální shody. Po levé straně je seznam všech nemovitostí s procentuální shodou. Následně tato komponenta předává data vnořeným komponentám, pro další zpracování.

ResultGoogleMap

Tato komponenta zajišťuje zobrazení informací na mapě. Jedná se o polohu nemovitostí, bodů zájmu s informacemi a důležitých míst.

ResultDetail

Detail se zobrazí, pokud uživatel klikne na nemovitost, kterou si chce prohlédnout. V detailu jsou zobrazeny informace o nemovitosti jako je popis, cena, adresa atd. Také v detailu nalezneme výsledky vyhledávání pro danou nemovitost, například shodu s otázkami, body zájmu, které byli použity, nebo informace o okolí.

Save

Komponenta pro zadání emailu a uložení seznamu. Tuto komponentu nalezneme ve výsledku pod seznamem nemovitostí.

RadarChart

Grafická komponenta, která zobrazuje zodpovězené otázky vykreslené na paprskovém grafu. Graf je zobrazený při odpovídání otázek, kdy se průběžně doplňuje, a také je vidět v detailu nemovitosti, kde jsou na grafu zobrazeny odpovědi od uživatele a vyhodnocená data pro nemovitost.

3.5.2 Použité knihovny

vue-fontawesome

Knihovna, jenž usnadňuje použití ve Vue.js oblíbené knihovny Fontawesome²⁴, která obsahuje sadu vektorových ikon. Knihovna byla použita ve vizuálním rozhraní pro zlepšení uživatelské přívětivosti.

axios

Tato knihovna umožňuje dotazovat na HTTP API server a následně zpracovat získaná data od serveru.

bootstrap

Frontendový CSS a javascriptový framework, který obsahuje sadu nadefinovaných komponent se styly. Pomocí těchto komponent je možné rychlé sestavení vizuálního rozhraní.

vue-chartjs

Knihovna, která obaluje knihovnu Chart.js²⁵ pro pohodlné použití ve Vue.js. Knihovna Chart.js obsahuje sadu různých grafů pro zobrazení dat. Tato knihovna je použita pro vykreslení hvězdicovitého grafu s odpověďmi na otázky.

node-sass a sass-loader

Knihovna pro zpracování stylových souborů sass a zkompilování do souboru CSS pomocí node.js.

vue-router

Rozšíření pro Vue.js, které přidává možnost odkazovat na různé komponenty aplikace za pomoci URL.

vue-scrollto

Za pomoci této knihovny je možná plynule posunout stránku na určitou pozici.

vue-slider-component

Vue.js komponenta, která umožňuje pohybem posuvníku vybrat určitou hodnotu z rozsahu, nebo vybrat rozsah hodnot.

²⁴ <https://fontawesome.com/>

²⁵ <http://www.chartjs.org/>

vue2-google-maps

Knihovna pro použití Google Maps ve Vue.js.

vue2-scrollbar

Komponenta do Vue.js, která umožní stylovat za pomoci css scrollbar, který je poté na všech zařízeních stejný.

vue2-touch-events

Knihovna, která na dané komponentě dokáže poslouchat pohyby a dotyky prstem na dotykových zařízeních. Knihovna je použita pro lepší uživatelskou přívětivost na těchto zařízeních.

4 Dokumentace

Uživatelská dokumentace

Uživatelská dokumentace je součástí aplikace na adrese „/user-documentation“. V této dokumentaci je popsáno, jak by měl uživatel správně postupovat v aplikaci, aby dosáhl správného výsledku.

Dokumentace kódu

Kód aplikace je dokumentovaný formou komentářů v kódu. Celý souhrn dokumentace kódu je k nalezení ve složce projektu „code-documentation“, popřípadě na adrese <http://rate.pavelseps.cz/code-documentation/>. Dokumentace kódu byla vygenerována za pomoci nástroje Doxygen²⁶, který z komentářů kódu aplikace vygeneruje html dokumentaci.

Dokumentace HTTP API požadavků na server

Formát dotazů na HTTP API server je popsán v aplikaci na adrese „/api-documentation“. Zde jsou popsány všechny požadavky s povinnými parametry, možnostmi odpovědi a chybovými hláškami.

Testovací dokumentace

Testovací dokumentace, která obsahuje scénáře pro testování je k nalezení v příloze Testovací dokumentace.

²⁶ www.doxygen.org

5 Testování

Testování aplikace probíhalo od samého začátku vývoje aplikace. Při vývoji aplikace byla vždy testována daná komponenta, na které se pracovalo. Komponenta byla testována manuálně za pomoci konzolové aplikace, kde byl nasimulovaný její chod a následně se kontroloval výstup v konzoli, zdali komponenta pracuje správně vůči vstupním datům.

V pozdější fázi vývoje, kdy byla otestovaná komponenta napojena do aplikace na API controller, se funkčnost manuálně otestovala za pomoci aplikace Postman²⁷.

Po napojení na vizuální část byla aplikace celkově testována několika manuálními průchody s důrazem na vyvíjenou komponentu a sledováním komunikace mezi serverovou částí a vizuální částí.

Vizuální část byla manuálně testována na v prohlížečích Chrome²⁸, Firefox²⁹ a Edge³⁰. Responzivita byla testována na různých zařízeních do minimální šířky 320px³¹, primárně na Apple iPad³² a Samsung A5 2016, zdali je aplikace stále uživatelsky přívětivá.

Pro testování funkčnosti odesílání emailu bylo využita aplikace Mailtrap.io, která funguje jako SMTP³³ server, ale emaily se nikam neodesílají a zobrazí se pouze v rozhraní zmíněné aplikace.

²⁷ Aplikace pro pohodlné otestování API požadavků na server - <https://www.getpostman.com/>

²⁸ Google Chrome 65.0.3325.181

²⁹ Mozilla Firefox 58.0.2

³⁰ Microsoft Edge 41.16299.15.0

³¹ Často používané nejmenší rozlišení. Starší telefon, nebo Apple Iphone 6 při zvětšení zobrazení v nastavení.

³² Často používaný tablet s rozlišením 1024px : 768px a 768px : 1024px.

³³ Simple Mail Transfer Protocol – internetový protokol určený pro přenos elektronické pošty

6 Závěr

Všechny cíle práce byly splněny, tedy vytvoření aplikace, sepsání dokumentací a její otestování. Byla vytvořena aplikace, která používá nový postup vyhledávání oproti stávajícím řešením, které byly zhodnoceny v rešeršním výzkumu. Využívá ohodnocování vůči osobním preferencím uživatelů, ze kterých získá doporučený seznam nemovitostí v okolí. Aplikace splňuje všechny analýzy, které byly provedeny před vývojem.

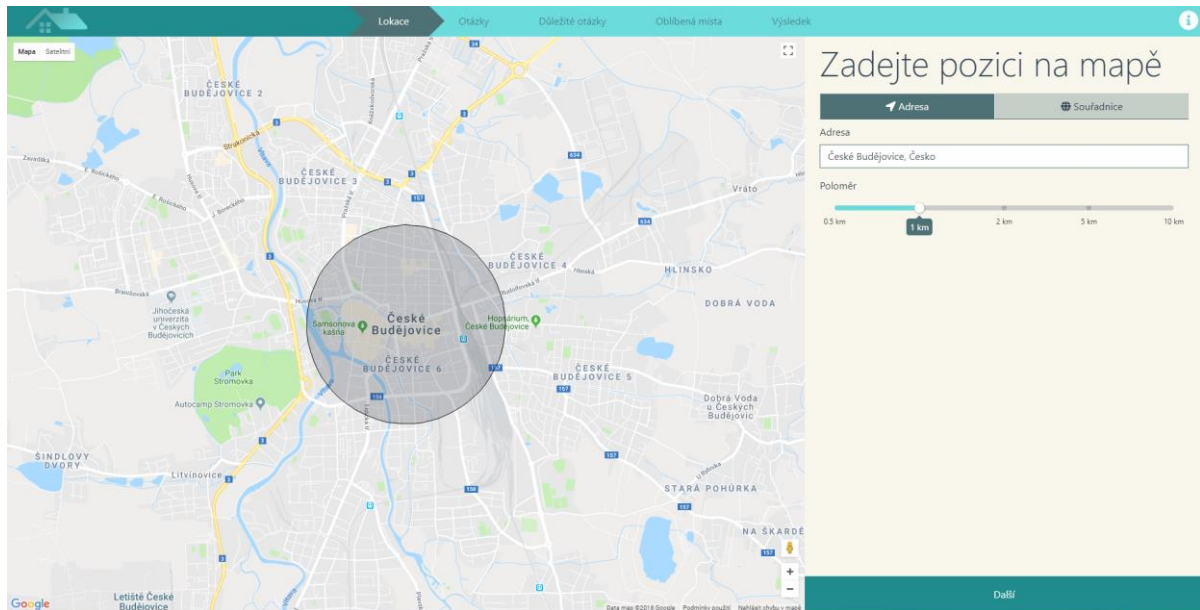
Aplikace má velký potenciál na rozšíření o další datové zdroje, díky kterým by se mohla rozšířit i do ciziny. Díky modulárnosti existuje možnost jednoduchého napojení do stávajících řešení, nebo úprava aplikace pro podobné použití, třeba pro nalezení podobných nemovitosti vůči nemovitosti jiné.

Aplikace je přístupná na adrese <http://rate.pavelseps.cz/>.

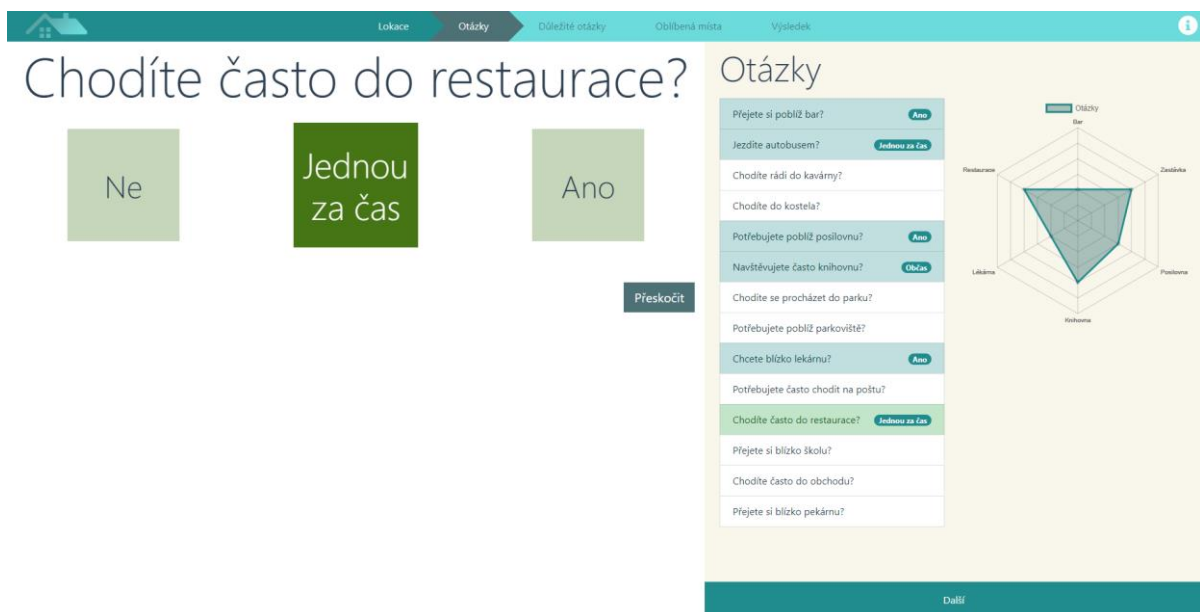
7 Literatura

- [1] Sreality.cz • reality a nemovitosti z celé ČR. *Sreality*. [Online] [Citace: 23. 1 2018.] <https://www.sreality.cz/>.
- [2] Váš partner na trhu s nemovitostmi | Realitní kanceláře RE/MAX. *RE/MAX Česká republika*. [Online] [Citace: 23. 1 2018.] <https://www.remax-czech.cz/>.
- [3] Immobilien - Häuser - Wohnungen - finden bei immowelt.de. *Immowelt*. [Online] [Citace: 23. 1 2018.] <https://www.immowelt.de/>.
- [4] Gelbe Seiten. *Gelbe Seiten*. [Online] [Citace: 23. 1 2018.] <https://www.gelbeseiten.de/>.
- [5] Immobilien, Wohnungen und Häuser bei ImmobilienScout24. *Immobilien Scout GmbH*. [Online] [Citace: 23. 1 2018.] <https://www.immobilienscout24.de/>.
- [6] Telekom | Mobilfunk, LTE, Festnetz und DSL Angebote. *Telekom Deutschland GmbH*. [Online] [Citace: 23. 1 2018.] <https://www.telekom.de/>.
- [7] Calculate distance, bearing and more between Latitude/Longitude points. *Movable Type Scripts*. [Online] <https://www.movable-type.co.uk/scripts/latlong.html>.

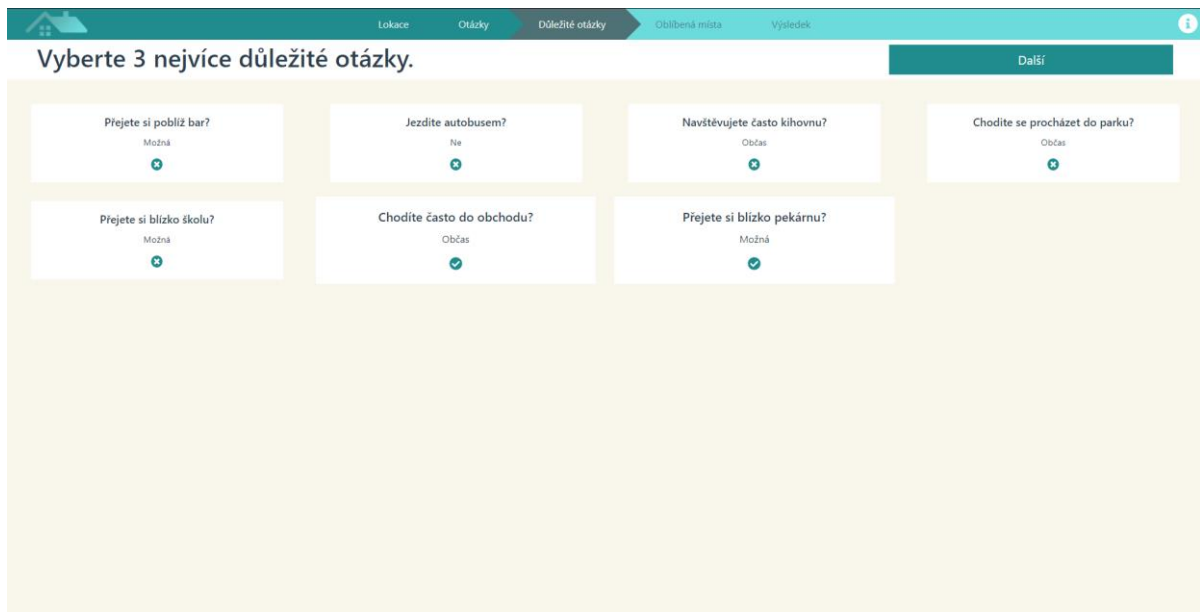
Příloha A Vizuální část aplikace



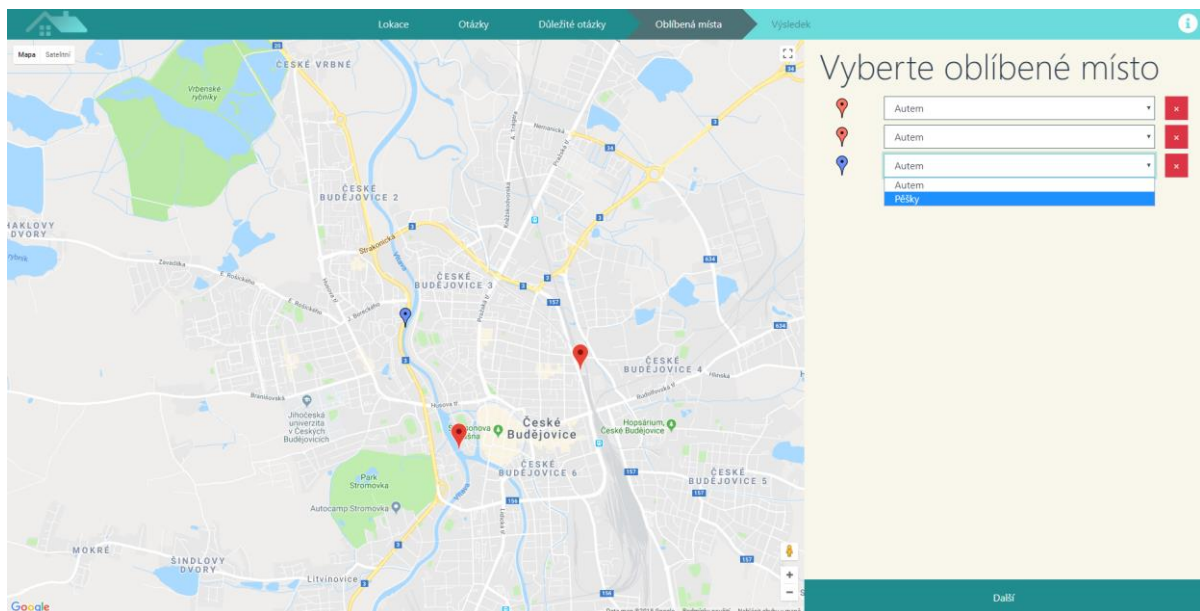
Obrázek 6: pohled výběru lokace pro vyhledávání



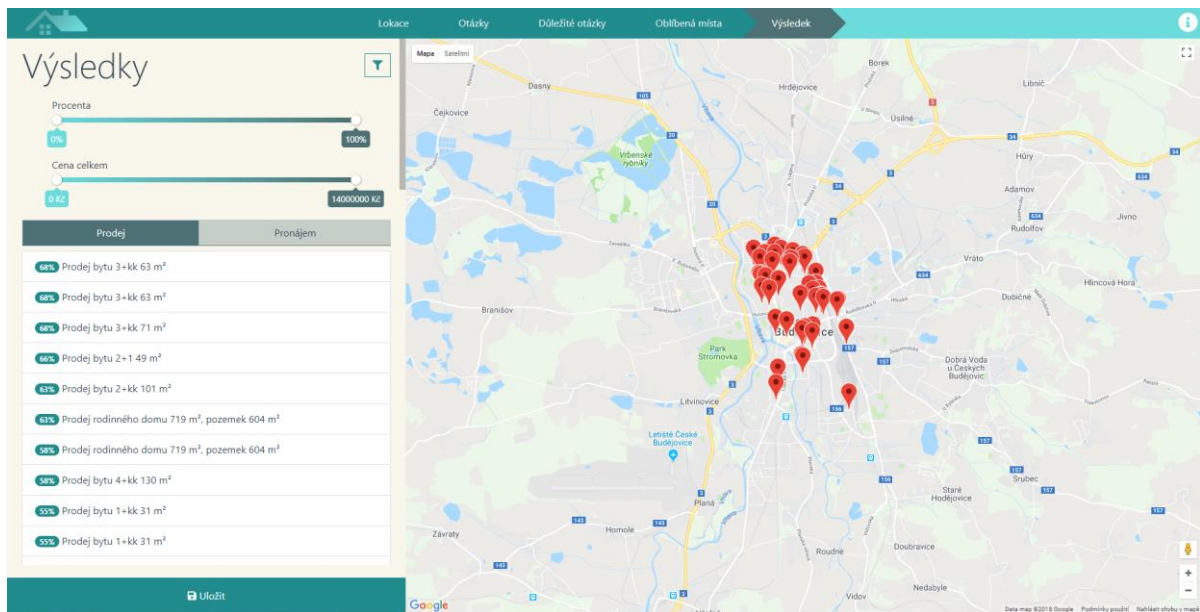
Obrázek 7: pohled odpovídání otázek



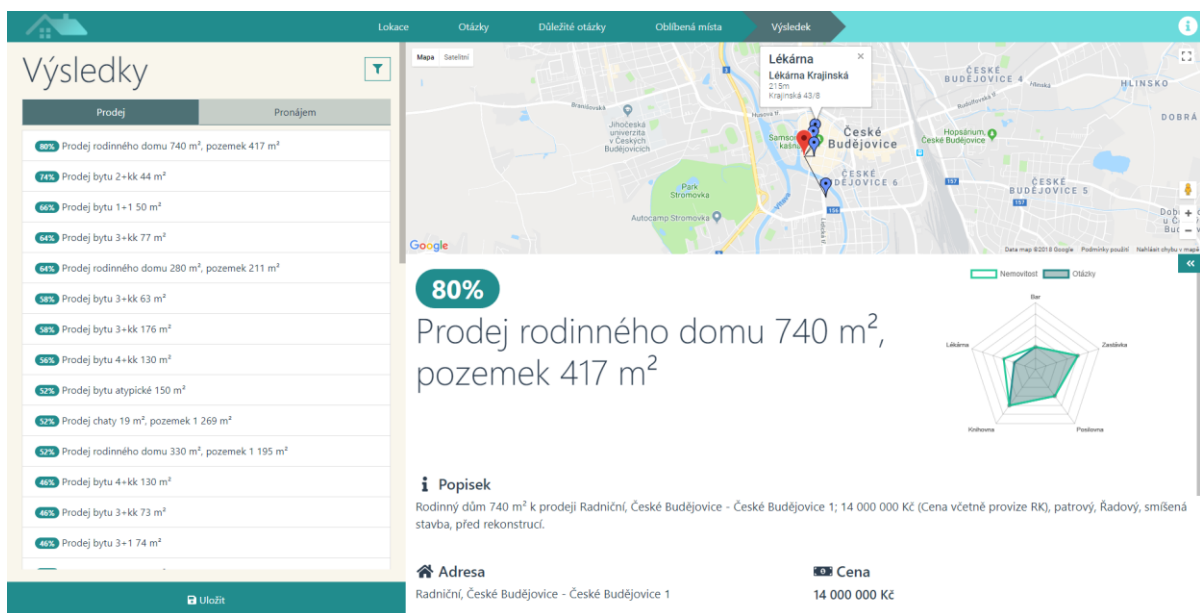
Obrázek 8: pohled výběru důležitých otázek



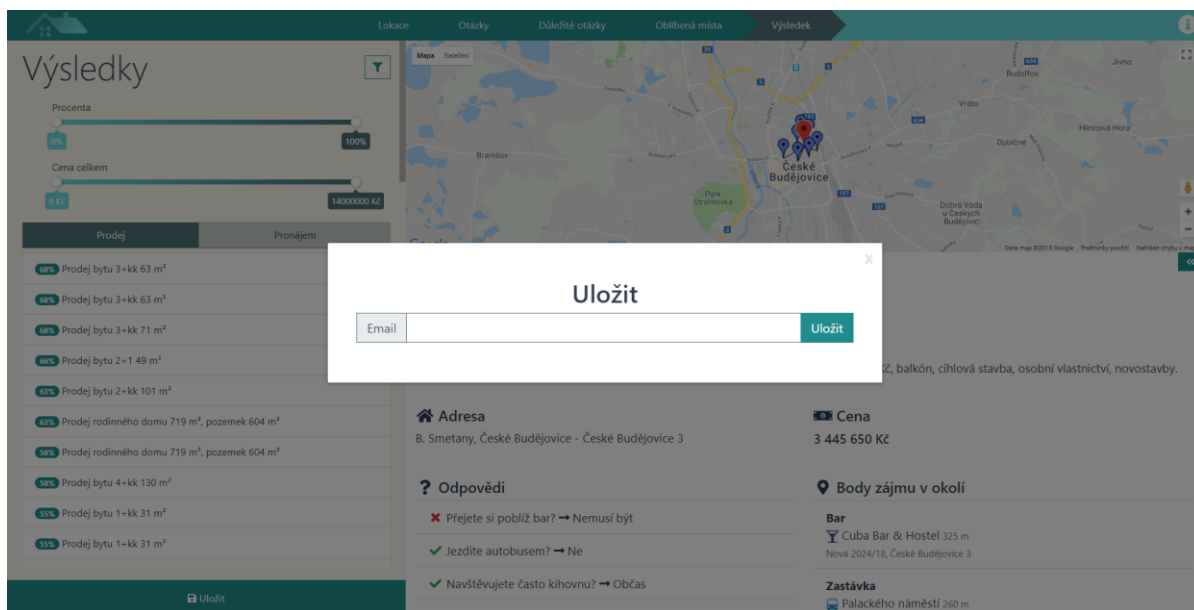
Obrázek 9: pohled výběru oblíbených míst



Obrázek 10: pohled výsledku



Obrázek 11: pohled výsledku s detailem



Obrázek 12: pohled pro uložení výsledku



Smazání uloženého listu

Prosím zadejte email pro potvrzení

Email Smazat

Obrázek 13: pohled pro smazání uloženého výsledku

Příloha B Testovací dokumentace

| | |
|--------------------|---|
| Název | Vybrání lokace |
| Účel | Otestování, zdali je možné vybrat lokaci z mapy, souřadnice anebo adresy. |
| Podmínky | Uživatel je na začátku vyhledávání. |
| Kroky | <ol style="list-style-type: none"> 1. Uživatel klikne na mapu. 2. Do pole „Adresa“ zadá „Most“, vybere. 3. Klikne na „Souřadnice“. 4. Upraví souřadnice. 5. Upraví velikost poloměru tažením myši. 6. Klikne na tlačítko „další“. |
| Očekávaný výsledek | <ol style="list-style-type: none"> 1. Při všech krocích se změní pozice na mapě až na 3. krok. 2. Během 5. kroku se změní pouze poloměr na mapě. 3. Při 6. kroku se načte další pohled. |

| | |
|--------------------|--|
| Název | Zodpovězení otázek |
| Účel | Ověření, zdali je možné zodpovědět otázky. |
| Podmínky | Získaná lokace vyhledávání. |
| Kroky | <ol style="list-style-type: none"> 1. Postupně projde všechny otázky. 2. Odpoví otázky, ale také i přeskočí. 3. Vráť se k některé otázce a změní odpověď. 4. Odpoví poslední otázku nebo klikne na tlačítko „další“. |
| Očekávaný výsledek | <ol style="list-style-type: none"> 1. Seznam otázek se opravuje dle odpovědí. 2. Při 3 a více zodpovězených otázkách se zobrazí graf. 3. Při 4. kroku se načte další pohled. 4. Aplikace povolí další postup, pokud je zodpovězena alespoň jedna otázka. |

| | |
|--------------------|---|
| Název | Vybrání důležitých otázek |
| Účel | Ověření vybírání otázek. |
| Podmínky | Získaná lokace a zodpovězené otázky. |
| Kroky | <ol style="list-style-type: none"> 1. Uživatel klikne na zobrazené otázky. 2. Klikne na tlačítko „další“ |
| Očekávaný výsledek | <ol style="list-style-type: none"> 1. Uživatel může vybrat maximální počet co je napsán. 2. Výběr se graficky projeví 3. Při 2. kroku se načte další pohled. |

| | |
|--------------------|---|
| Název | Vybrání oblíbeného místa |
| Účel | Otestování možnosti přidání oblíbených míst. |
| Podmínky | |
| Kroky | <ol style="list-style-type: none"> 1. Uživatel klikne víckrát na mapu. 2. Změní typ pohybu. 3. Odebere bod tlačítkem „X“. 4. Klikne na tlačítko „další“. |
| Očekávaný výsledek | <ol style="list-style-type: none"> 1. Při 1. kroku se přidá bod do seznamu. 2. Při 2. kroku se změní název pohybu. 3. Při 3. kroku se bod odebere. 4. Při 4. kroku se načte další pohled. |

| | |
|--------------------|--|
| Název | Zobrazení detailu výsledku |
| Účel | Otestování načtení detailu výsledku |
| Podmínky | Získaný výsledek. Uživatel je na pohledu výsledku. |
| Kroky | <ol style="list-style-type: none"> 1. Uživatel klikne na jednu z nemovitostí v seznamu nebo na bod na mapě. |
| Očekávaný výsledek | Zobrazí se detail dané nemovitosti. |

| | |
|--------------------|---|
| Název | Uložení seznamu |
| Účel | Ověření, zdali je možné uložit výsledek. |
| Podmínky | Získaný výsledek. Uživatel je na pohledu výsledku. |
| Kroky | <ol style="list-style-type: none"> 1. Klikne tlačítku „Uložit“. 2. Uživatel zadá emailovou adresu. 3. Klikne na „uložit“ nebo stiskne enter. |
| Očekávaný výsledek | <ol style="list-style-type: none"> 1. Při 1. kroku zobrazí modal. 2. Po 3. kroku aplikace zobrazí pozitivní zprávu o uložení. 3. Email přišel na zadanou emailovou adresu. |

| | |
|--------------------|--|
| Název | Smazání uloženého seznamu |
| Účel | Ověření smazání uloženého seznamu. |
| Podmínky | Uživateli přišel email ohledně uložení. |
| Kroky | <ol style="list-style-type: none"> 1. Klikne na url pro smazání, co přišla emailem. 2. Zadá mail, na který si nechal seznam uložit. 3. Klikne na „smazat“ nebo stiskne enter. 4. Klikne na url v emailu pro zobrazení seznamu. |
| Očekávaný výsledek | <ol style="list-style-type: none"> 1. Po 3. kroku se zobrazí pozitivní zpráva o smazání. 2. Při 4. kroku se zobrazí chyba. |