



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**TVORBA CENOVÝCH MÁP V OBLASTI REALITNÉHO TRHU**

GENERATION OF PRICE MAPS FOR REAL ESTATE MARKET

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MICHAEL HALINÁR**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. PAVEL OČENÁŠEK, Ph.D.**

BRNO 2017

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav informačních systémů

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Halinár Michael**

Obor: Informační technologie

Téma: **Tvorba cenových map v oblasti realitního trhu**  
**Generation of Price Maps for Real Estate Market**

Kategorie: Softwarové inženýrství

**Pokyny:**

1. Seznamte se s API různých projektů webových map.
2. Analyzujte požadavky na aplikaci využívající API vybraných map a informací o realitách získaných z jiných projektů.
3. Navrhněte aplikaci realizující cenové mapy nabídky realitního trhu v ČR.
4. Navrženou aplikaci implementujte a otestujte na reálných datech.
5. Diskutujte dosažené výsledky práce a možnosti dalšího pokračování.
6. Vytvořte plakát či kratší souhrnnou zprávu prezentující výsledky aplikace.

**Literatura:**

- Liu Bing, Totty Brian. Web data mining: exploring hyperlinks, contents, and usage data. Springer, 2007. ISBN 978-3-540-37881-5.
- Gourley David, Totty Brian. HTTP: the definitive guide. O'Reilly, 2002. ISBN 15-659-2509-2.
- Mitchell Ryan, Holmes James. Instant web scraping with Java, Packt Publishing, 2013. ISBN 978-184-9696-883.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Očenášek Pavel, Ing., Ph.D., UIFS FIT VUT**

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
612 66 Brno, Božetěchova 2



doc. Dr. Ing. Dušan Kolář  
vedoucí ústavu

## Abstrakt

Táto práca sa zaoberá implementáciou cenových máp pomocou mapového aplikačného rozhrania s využitím informácií o realitách na českom trhu. Ako mapové aplikačné rozhranie bolo vybrané Google maps API. Hlavnou časťou implementácie je skript, ktorý získava informácie o realitách pomocou regulárnych výrazov a ukladá ich do databáze. Aplikácia umožňuje zobrazenie ponúk na mape, aplikovanie filtrov a zobrazenie teplotnej mapy. Databáza si uchováva aj neaktuálne ponuky, vďaka tomu dokáže zobrazovať na teplotnej mape aj vývin cien. Ako primárny jazyk bol použitý PHP, interakciu s užívateľom zabezpečuje zase javascript.

## Abstract

This work deals with implementation of price maps through the use of map application interface using informations about real estates on Czech market. Google maps API was the chosen map application interface. Main part of implementation is script which gathers informations about real estates by the help of regular expressions and stores them in database. Application allows viewing offers on map, applying filters and displaying them on heat map. Database stores also outdated offers, this allows displaying growth of prices. PHP was used as main programming language for implementation, interaction with user was implemented in javascript.

## Klíčové slová

reality, cenové mapy, mapové API, teplotná mapa, webová aplikácia, databáza, PHP, javascript

## Keywords

real estate, price maps, maps API, heat map, web application, database, PHP, javascript

## Citácia

HALINÁR, Michael. *Tvorba cenových máp v oblasti realitného trhu*. Brno, 2017. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Očenášek Pavel.

# Tvorba cenových máp v oblasti realitného trhu

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Pavla Očenáška, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....  
Michael Halinár  
16. mája 2017

## Podakovanie

Ďakujem vedúcemu tejto bakalárskej práce pánovi Ing. Pavlovi Očenáškovi, Ph.D. za poskytnuté rady, pripomienky a odborné vedenie.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Cenové mapy . . . . .	3
<b>2</b>	<b>Základné pojmy</b>	<b>4</b>
2.1	Cenová mapa . . . . .	4
2.2	Webová aplikácia . . . . .	4
2.2.1	Tenký klient . . . . .	4
2.3	API . . . . .	5
2.4	UML . . . . .	5
2.4.1	Diagram prípadu použitia . . . . .	5
2.4.2	Konceptuálne modelovanie . . . . .	6
2.5	HTML . . . . .	7
<b>3</b>	<b>Vývoj aplikácie</b>	<b>8</b>
3.1	Životný cyklus aplikácie . . . . .	8
3.1.1	Analýza a špecifikácia požiadavkov . . . . .	8
3.1.2	Návrh . . . . .	9
3.1.3	Implementácia . . . . .	9
3.1.4	Integrácia a testovanie . . . . .	9
3.1.5	Prevádzka a údržba . . . . .	9
<b>4</b>	<b>Analýza</b>	<b>10</b>
4.1	Existujúce riešenia . . . . .	10
4.2	Mapové aplikačné rozhrania . . . . .	11
4.3	Požiadavky na aplikáciu . . . . .	12
4.4	Realitné portály . . . . .	12
4.5	Potrebné dáta . . . . .	13
4.6	Získavanie dát . . . . .	13
4.6.1	Regulárne výrazy . . . . .	13
4.6.2	Simple HTML DOM parser . . . . .	13
4.7	Nasadenie na server . . . . .	13
4.7.1	VPS . . . . .	13
4.7.2	Raspberry pi . . . . .	14
4.7.3	Dedikovaný server . . . . .	14
4.8	Cielová skupina . . . . .	14
4.9	Zabezpečenie webových stránok proti robotom . . . . .	14
4.10	Prevenca blokovania robotov na stránkach . . . . .	15

<b>5</b>	<b>Návrh</b>	<b>16</b>
5.1	Diagram prípadu použitia . . . . .	16
5.2	Entitno-vzťahový diagram . . . . .	16
<b>6</b>	<b>Použité technológie</b>	<b>18</b>
6.1	XAMPP . . . . .	18
6.2	Apache . . . . .	18
6.3	MariaDB . . . . .	18
6.4	PHP . . . . .	18
6.5	Google maps . . . . .	19
6.6	Geocoding . . . . .	21
6.7	Javascript . . . . .	21
6.8	PhantomJS . . . . .	21
6.9	Windows Task Scheduler . . . . .	21
6.10	PhpMyAdmin . . . . .	22
<b>7</b>	<b>Implementácia</b>	<b>23</b>
7.1	Aplikácia . . . . .	23
7.2	Server . . . . .	26
7.3	Databáza . . . . .	26
7.4	Dizajn . . . . .	27
7.5	Javascript . . . . .	27
	7.5.1 GoogleMap . . . . .	27
	7.5.2 HeatLayer . . . . .	28
	7.5.3 NavBar . . . . .	28
	7.5.4 Ošetrovanie vstupu . . . . .	28
<b>8</b>	<b>Testy</b>	<b>29</b>
8.1	Porovnanie s ostatnými aplikáciami . . . . .	29
8.2	Blokovanie automatického skriptu . . . . .	29
8.3	Užívateľské testovanie . . . . .	30
<b>9</b>	<b>Záver</b>	<b>31</b>
	<b>Literatúra</b>	<b>32</b>
	<b>Prílohy</b>	<b>34</b>
<b>A</b>	<b>Ukážky z aplikácií</b>	<b>35</b>
<b>B</b>	<b>Obsah priloženého pamäťového média</b>	<b>38</b>

# Kapitola 1

## Úvod

### 1.1 Cenové mapy

Táto bakalárska práca sa zameriava na spracovanie dát z realitných serverov. Cieľom je sprostredkovať a agregovať informácie takým štýlom aby ponúkli užívateľovi ďalšie možnosti rozhodovania sa pri kúpe nehnuteľnosti.

Cenové mapy by mali pomáhať s výberom toho najlepšieho bývania v rámci finančných možností. Veľké množstvo nehnuteľností na trhu môže byť neprehľadné pre potenciálneho kupcu. Preto je teplotné zobrazenie cien výhodné vďaka rýchlemu sa zorientovaniu na mape. Užívateľ môže porovnávať nehnuteľnosti vzhľadom na okolité ponuky a urobiť si tak predstavu o reálnej cene. Cieľom tejto práce je výsledná aplikácia ktorá bude zobrazovať nehnuteľnosti na mape.

Rozdelenie kapitol je nasledovné:

1. analýza,
2. návrh,
3. implementácia,
4. testy,
5. záver.

Analýza sa zameriava na existujúce riešenia hlavne na ich nedostatky ktorým je treba sa vyhnúť. Taktiež je nutné analyzovať dostupné mapové aplikačné rozhrania. Návrh obsahuje entitno-vzťahový diagram dôležitý pre vytvorenie databázy a detailný prípad použitia. V kapitole implementácie je popísaná výsledná aplikácia a jej časti. Nasleduje testovanie a záver, ktorý obsahuje aj ďalšie možné rozšírenia aplikácie.

## Kapitola 2

# Základné pojmy

### 2.1 Cenová mapa

Cenová mapa je grafické vyjadrenie reálnych nehnuteľností na určitom území. Ako podklady pre vytvorenie mapy slúžia ponuky z realitných portálov, ktoré sa zobrazujú na mape. Cenová mapa je nástroj pre rýchlu ale veľmi všeobecnú predstavu o ponukách na trhu. Môžeme sa stretnúť s rôznymi druhmi cenových máp ako napríklad cenová mapa pozemkov, bytov či nájomného.

### 2.2 Webová aplikácia

Webová aplikácia je taká aplikácia ktorá je prístupná užívateľom cez internet alebo intranet pomocou webového serveru. Tento druh aplikácie je populárny hlavne kvôli nízkym požiadavkám na klientskú aplikáciu. Schopnosť aktualizovať a spravovať webové aplikácie bez nutnosti inštalovania softvéru potencionálne neobmedzenému množstvu zákazníkov taktiež pridáva na ich obľúbenosti. Webové aplikácie môžu byť používané pre implementáciu informačných systémov, e-shopov, diskusných fór a mnoho iného. Dokážu pracovať bez ohľadu na použitú platformu alebo operačný systém. Podpora pre takýto druh aplikácie je jednoduchšia oproti klasickej klient-server aplikácie kedy pri opravovaní chyby nie je nutné cestovať za klientom, ale stačí chybu opraviť na serveri.

Webová aplikácia má však aj svoje nevýhody. Závislosť na internetovom pripojení obmedzuje používanie aplikácie. Bez internetu nie je možné používať takúto aplikáciu. Kvalita internetového pripojenia môže byť pri niektorých aplikáciách kľúčová keďže odozva serveru je na ňom závislá. Výpadok serveru taktiež znemožňuje používanie. Ako klientské aplikácie slúžia prehliadače ktoré sa líšia v svojej implementácii, na tieto rozdiely treba brať ohľad.[5]

#### 2.2.1 Tenký klient

Tenký klient je počítač alebo počítačový program, ktorý nepozná logiku aplikácie a silno závisí na serveri. So serverom sa spája pomocou HTTP protokolu a posiela mu svoje požiadavky. Všetky výpočty ktoré si klient vyžiada sa realizujú na strane servera. Medzi výhody takéhoto klienta patrí: nízke nároky na hardvér, jednoduché aktualizácie, prenositeľnosť. Medzi nevýhody patria zase: vysoké nároky na server, odozva. [7]



## 2.3 API

API z angl. Application Programming Interface je rozhranie pre programovanie aplikácií. Je to množina definícií, protokolov a funkcií pre vývin softvéru. Tieto nástroje majú uľahčovať prácu programátora poskytovaním potrebných nástrojov pre vyvíjanú aplikáciu. Veľké množstvo aplikácií poskytuje svoje API nechýba medzi nimi Google Maps API. [2]

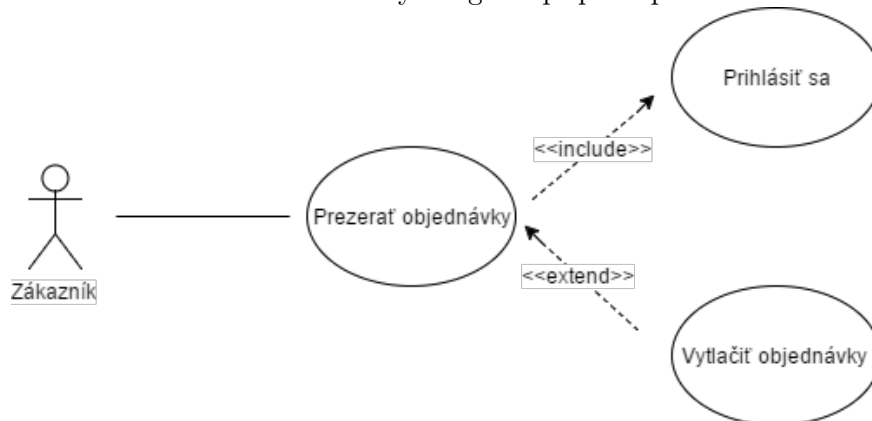
## 2.4 UML

Unified Modeling language je jazyk pre modelovanie softvéru. Používa sa pri analýze a návrhu aplikácií. Podporuje taktiež objektovo orientované programovanie.[8] UML diagramy sa delia na dva základné typy a to statické modely a dynamické modely. Medzi statické patria napríklad: diagram tried, diagram nasadenia. Medzi dynamické zase: objektový diagram, diagram spolupráce, sekvenčný diagram a iné. Programovanie v tíme je čoraz viac rozšírené je nutné aby všetci členovia tímu chápali projektu rovnako. Tu nastupujú UML diagramy ktoré jednoznačne určia model systému a jeho komponenty. Pomáha takisto aj pri komunikovaní so zákazníkom ktorý nie vždy vie správne vyložiť problém ktorý chce vyriešiť. Po schválení diagramu sa prechádza na ďalšie fáze vývoja softvéru[13].

### 2.4.1 Diagram prípadu použitia

Diagram prípadu použitia detailne popisuje chovanie systému. Patrí medzi dynamické modely. Zahŕňa typy užívateľov a spôsob interakcie so systémom. Tento model opisuje systém tak že ukazuje to čo sa dá docieľiť ale nie spôsob akým to systém docieľi.[4]

Obr. 2.1: Jednoduchý Diagram prípadu použitia



#### Aktér

Aktér predstavuje užívateľa, systém alebo jeho časť. Môže ním byť teda admin, zákazník alebo časť systému ako napríklad aplikácia na pravidelné odosielanie newsletteru. Názov aktéra určuje jeho postavenie v systéme.

## Prípád použitia

Prípád použitia predstavuje úkon ktorý môže aktér vykonávať ako napríklad: vyhľadať produkt, vložiť do košíka, prezeráť objednávky a podobne. Prípád použitia je z veľkej časti abstraktný keďže proces ktorý sa za jednotlivými úkonmi skrýva nie je v diagrame zahrnutý.

### Príklad diagramu prípadu použitia

Tento diagram obsahuje 1 aktéra a to zákazníka, ktorý môže prezeráť svoje objednávky. Ďalej tu je väzba «*include*» ktorá značí že pred tým ako si zákazník bude môcť prezeráť objednávky, musí byť prihlásený. Väzba «*extend*» určuje ďalšie možné funkcie ktoré môže aktér vykonávať. V tomto prípade po zobrazení objednávok môže zákazník svoje objednávky vytlačiť.

## 2.4.2 Konceptuálne modelovanie

Konceptuálne modelovanie je zahrnuté v množstve metód vývoja softvéru. Používa sa pri analýze požiadavkov a jeho cieľom je vytvorenie relačného modelu databáze s ktorým potom bude daný softvér pracovať. Pri tomto modelovaní nie je dôležité akým spôsobom budú dáta uložené.[13]

### Entitno-vzťahový diagram

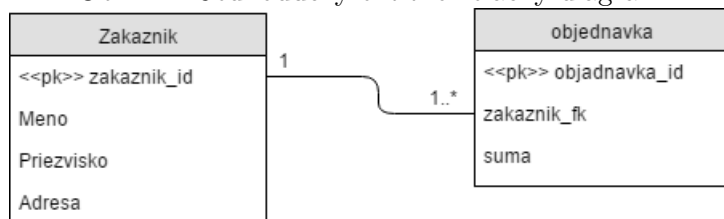
Najpoužívanejšou modelovacou metódou je entitno-vzťahové modelovanie. Teda vytváranie entitno-vzťahového diagramu. Obsahuje tieto dve základné časti:

1. Entita : Predstavuje objekt o ktorom budeme uchovávať v databáze informácie. Pri objektovom programovaní sa entita dá chápať aj ako trieda. Entita obsahuje svoje atribúty ktoré budú uložené v databáze. Atribúty môžu byť jednoduché a zložené, pričom jednoduchý sa skladá zo skalárnych hodnôt a zložitý obsahuje niekoľko iných atribútov.
2. Vzťah: Entity môžu byť v určitých vzťahoch. Tieto vzťahy sa v databáze implementujú pomocou cudzích kľúčov. Počet možných vzťahoch u entít určuje kardinalita vzťahu.

### Príklad entitno-vzťahového diagramu

V nasledujúcom príklade nebudú uvedené všetky možnosti entitno-vzťahového diagramu ale iba tie, ktoré sú potrebné pre túto bakalársku prácu. Obrázok 2.2 obsahuje 2 tabuľky: ZAKAZNIK a OBJEDNAVKA. Každá tabuľka obsahuje primárny kľúč ktorý slúži na jednoznačnú identifikáciu položky v konkrétnej tabuľke. Tabuľka ZAKAZNIK ďalej obsahuje atribúty: MENO PRIEZVISKO a ADRESA. Tabuľka OBJEDNAVKA obsahuje atribúty: SUMA a ZAKAZNIK\_FK, ktorý bude slúžiť ako cudzí kľúč do tabuľky ZAKAZNIK pomocou neho sa budú spájať záznamy medzi tabuľkami. Vzťah medzi týmito tabuľkami má kardinalitu 1..\* to znamená že zákazník môže mať viacero objednávok ale jedna objednávka bude patriť práve jednému zákazníkovi.

Obr. 2.2: Jednoduchý entitno relačný diagram



## 2.5 HTML

HTML - Hypertext Markup Language je značkovac jazyk ktorý slúži na popis webových stránok a aplikácií [6]. Spolu s CSS(Cascading Style Sheets) a Javascript-om umožňujú vytváranie interaktívnych webových stránok. Webový prehliadač dostáva od servera HTML dokument ktorý vykreslí. Popisuje štruktúru dokumentu nie jeho výzor. V minulosti sa používal aj na definovanie štýlu v dnešnej dobe už tomu tak nie je. Vznikol ako podmnožina metajazyka SGML5 v ktorom je možné definovať značkovacie jazyky pre dokumenty. Takýto dokument sa skladá z tagov. Jednotlivé tagy definujú rozličné časti štruktúry. Ďalej môže obsahovať hypertextové<sup>1</sup> odkazy, multimediálny obsah, skripty a metainformácie pre webový prehliadač. V jazyku HTML existujú 4 druhy prvkov:

1. štruktúrové prvky - označujú zmysel textu (<h1>Web</h1> - slovo Web je nadpisom),
2. prezentačné prvky - popisujú výzor textu (<b>aplikácia</b> - zobrazí slovo aplikácia tučným písmom),
3. ovládacie prvky - môžu reprezentovať tlačidlá, zaškrťavacie polia, formuláre a pod.,
4. hypertextové odkazy - odkazy (<a href="www.google.com/">google</a> - slovo google je odkazom na stránku).

Každý dokument začína deklaráciou typu dokumentu. To znamená že dokument v najnovšej verzii HTML5 na prvom riadku bude obsahovať: <!DOCTYPE HTML>.

---

<sup>1</sup>Nelineárne usporiadanie textu v ktorom su jednotlivé časti prepojené

## Kapitola 3

# Vývoj aplikácie

### 3.1 Životný cyklus aplikácie

Existuje veľa prístupov k vývoju softvéru. Jednotlivé prístupy sú vyjadrené v modeloch životných cyklov softvéru, ktoré sa líšia hlavne v poradí vývojových etáp. Model životného cyklu softvéru je teda zložený z etáp, pre každú etapu definuje potrebné činnosti ďalej jej vstupy a výstupy [3]. Etapy popisujúce činnosti vývoju softvéru sú nasledujúce:

1. analýza a špecifikácia požiadavkov - 8%,
2. návrh - 7%,
3. implementácia - 12%,
4. integrácia a testovanie - 6%
5. údržba a prevádzka- 67%.

Každú etapu definuje aj približné úsilie ktoré je potrebné na jej uskutočnenie. Najviac úsilia je spotrebovaného na údržbu a prevádzku. Preto je treba dbať na dokonalé splnenie prvotných etáp aby sa zamedzilo zbytočnému vynakladaniu úsilia v poslednej etape pri opravovaní chýb.

#### 3.1.1 Analýza a špecifikácia požiadavkov

Prvou etapou pri vývoji softvéru je analýza. V tejto etape prebieha konzultácia so zákazníkom o požadovanom softvére. Kladie sa dôraz na úplné definovanie funkcií ktoré by mal finálny produkt spĺňať bez ohľadu na to akým spôsobom to bude dosiahnuté. Špecifikácia od zákazníka je prevedená do jasnej a štruktúrovanej podoby. Častou tejto analýzy by mala byť aj štúdia vhodnosti, ktorá rozhodne o tom či je softvér možné vytvoriť a udržať v prevádzke. Takisto môže pomôcť aj pri analýze rizík. Výstupom tejto analýzy je plán akceptačného testovania na základe ktorého potom zákazník akceptuje projekt, je potom aj zárukou pre vývojárov kedy chýbajúca špecifikácia môže byť problémom pri preberaní aplikácie.

### **3.1.2 Návrh**

Návrh nadväzuje na analýzu požiadavkov. Slúži na dekompozíciu projektu a ujasnenia si systému ako celku. V tejto fáze je už vhodné naplánovanie testovania celého systému, rovnako ako zaškoľovanie užívateľov. Sústreďuje sa na stanovenie logickej a fyzickej štruktúry uloženia dát, výber funkcií a algoritmov ktoré s nimi budú pracovať. Výsledkom sú aj diagramy ako napríklad entitno-vzťahový diagram, ktorý špecifikuje databázu či diagram prípadu použitia.

### **3.1.3 Implementácia**

Predstavuje vytvorenie softvéru v podobe programu. Súčasťou implementácie je aj vytvorenie dokumentácie a otestovanie implementovaných častí.

### **3.1.4 Integrácia a testovanie**

Poslednou fázou je testovanie a inštalácia. Táto fáza zahŕňa spojenie všetkých častí systému do jedného celku a jeho otestovanie. Môžu sa pritom objaviť nové chyby ktoré sa nedali odhaliť pri testovaní samostatných modulov. Po dôkladnom otestovaní výsledného programu nasleduje akceptačné testovanie na základe ktorého sa zákazník rozhodne či produkt prevezme. Po akceptovaní zákazníkom je na rade inštalácia a zaškoľovanie užívateľov.

### **3.1.5 Prevádzka a údržba**

Reálnym nasadením u zákazníka môžu nastať ďalšie problémy vzniknuté dlhodobým používaním. Údržba teda zahŕňa opravy týchto problémov či rozširovanie funkcionality.

# Kapitola 4

## Analýza

### 4.1 Existujúce riešenia

Existuje veľmi málo cenových máp ktoré ponúkajú aj teplotné zobrazenie. Jednou z nich je *cenovamapa.cz*<sup>1</sup>. Pre zobrazenie mapy je nutné sa zaregistrovať. Disponuje pekným užívateľským rozhraním pomocou ktorého sa jednoducho nastaví aplikácia podľa požiadavkov užívateľa. Pre úplných laikov je dostupné aj video ktoré podrobne vysvetlí všetko potrebné ohľadom používania. Pri vstupe do mapy je potrebné zadať špecifikácie hľadaných objektov. Na výber je možné vyhľadávať byty, domy či pozemky. Pre vyhľadávanie bytov je nutné zadať mesto rozlohu a dispozíciu bytu. Rozšírená ponuka nástrojov je poskytnutá tým čo sú ochotný zaplatiť si za ich služby. Patrí medzi ne napríklad kreslenie oblasti z ktorej sa má urobiť výber ponúk alebo ďalšie rôzne agregácie údajov . Je možné kúpiť jednorazový prístup spolu s dvomi reportmi alebo rovno ročný prístup pre viacero používateľov. Vykresľovanie teplotnej vrstvy trvá pomerne dlho a pri malom rozlíšení zaberá užívateľské rozhranie veľkú časť obrazovky. Tieto faktory môžu pre užívateľa pôsobiť rušivo a bude treba sa ich vyvarovať. Na stránke neudávajú aktuálnosť dát ani nie sú ochotný tieto údaje sprístupniť verejnosti. Využívané mapové rozhranie - OpenStreetMap.

Ďalším existujúcim riešením je *cenovamapa.eu*<sup>2</sup>. Užívateľské rozhranie ponúka 3 zobrazenia cenových vrstiev a to byty na prenájom, byty na predaj a domy na predaj. Vykresľovanie je pomerne rýchle a farebná schéma v odtieňoch modrej a fialovej dobre vynikne na mape pričom nesplýva s cenovou vrstvou. Pri väčšom priblížení je možné zobrazíť aj popisné čísla domov, definičné čísla parciel a mapu katastru. Po kliknutí na miesto na mape je automaticky vypočítaná priemerná cena za meter štvorcový. Pri malom rozlíšení obrazovky tu nastáva rovnaký problém ako pri predchádzajúcom portáli, užívateľské rozhranie príliš zasahuje do mapy a zaberá tak až 40% z celkového okna. Stručná ale jasná nápoveda vysvetlí všetko potrebné. Informácie o cenách sú zbierané priebežne a aktualizované raz za mesiac vzhľadom na náročnosť všetkých výpočtov. Využívané mapové rozhranie - Google maps.

Portál *www.realitymon.cz*<sup>3</sup> taktiež poskytuje zobrazenie cien nehnuteľností na mape. Užívateľ si môže vybrať z veľkého množstva ponúk medzi ktoré sa triedia do troch skupín a to byty, domy alebo komerčné nehnuteľnosti. Pri každej skupine je možné ďalej špecifikovať požiadavky, pre byty to je dispozícia, pre domy je to rozdelenie medzi chalupy, rodinné domy či vily. Komerčné nehnuteľnosti je možné vyberať z reštaurácií, skladov a pod. Tieto ponuky

<sup>1</sup>Dostupné na <http://www.cenovamapa.cz/>

<sup>2</sup>Dostupné na <http://www.cenovamapa.eu/>

<sup>3</sup>Dostupné na <http://www.realitymon.cz/>

môže ďalej filtrovať podľa rozlohy, ceny a ceny za meter štvorcový alebo zadať konkrétne realitné kancelárie z ktorej sa budú zobrazovať dáta. Pri kliknutí na jednotlivé ponuky na mape sa otvorí užívateľovi náhľad na podrobnosti ponuky a prepojenie na pôvodný inzerát. Nie vždy sú v náhlade korektné údaje, pri niektorých realitách sa v cene za meter štvorcový nachádza údaj -1 prípadne 0 Kč za meter štvorcový. Rovnaký problém s rozhraním zasahujúcim príliš veľa do mapy nastáva aj tu, navyše náhodné ponuky zobrazované na spodnej lište pôsobia ako nepríjemná reklama.

*Cenovamapa.org*<sup>4</sup> poskytuje základné rozdelenie medzi bytové domy, rodinné domy a ostatné developerské projekty. Nehnutelnosti sú na mape prezentované ako značky pričom každá poskytuje náhľad a popis. Po výbere reality má užívateľ možnosť si objednať za poplatok report z okolia do 500 alebo 1000 metrov. V tomto reporte budú vypočítané priemerné ceny realít vo vybranom okruhu alebo v konkrétnej budove. Teplotné zobrazovanie cien nie je podporované.

## 4.2 Mapové aplikačné rozhrania

Výber rozhrania je veľmi dôležitý z hľadiska finančného ale aj programátorského. Niektoré API sú spoplatnené iné zase neposkytujú príliš mnoho funkcií. Na trhu existujú desiatky rozhraní. Tu je analýza niektorých z nich:

1. Microsoft Bing Maps<sup>5</sup> populárne rozhranie Microsoftu ktorý sa stále snaží o pridávanie nových možností, pri prechode na HTML5 pridali už aj teplotné zobrazenie,
2. OpenLayers<sup>6</sup> open source, poskytuje veľa možností, dokáže pracovať s GeoJSON, TopoJSON, KML, GML a ďalšími geografickými dátovými formátmi,
3. Foursquare<sup>7</sup> jeho hlavnou výhodou je spojenie tohoto API s aplikáciou Foursquare a jej užívateľmi, poskytuje zasielanie POST dotazov na server na ktorom aplikácia beží keď si niektorý užívateľ zapne aplikáciu, zanechá značku na mieste či pridá fotku, toto umožní pracovať s mapou v reálnom čase,
4. OpenStreetMap<sup>8</sup> je vyvíjané veľkou skupinou dobrovoľníkov ktorý vytvárajú a poskytujú geografické dáta zadarmo, avšak toto rozhranie nedisponuje možnosťou teplotného zobrazenia dát preto nemôže byť použitá v tomto projekte,
5. HERE Maps<sup>9</sup> pôvodne vyvíjané spoločnosťou NOKIA, poskytujú javascriptové rozhranie aj pre nekomerčné používanie do 15 000 prístupov za mesiac, podporuje aj teplotné zobrazenie dát,
6. Google Maps<sup>10</sup> toto rozhranie je na trhu už cez 10 rokov len preto sa dá považovať za najpokročilejšie, poskytuje hneď niekoľko verzií rozhrania pre rôzne účely ako napríklad Street View, Google Earth, Google Roads a mnoho ďalších,

---

<sup>4</sup>Dostupné na <https://www.cenovamapa.org/>

<sup>5</sup>Dostupné na <https://www.bing.com/maps>

<sup>6</sup>Dostupné na <https://openlayers.org/>

<sup>7</sup>Dostupné na <https://foursquare.com/>

<sup>8</sup>Dostupné na <https://www.openstreetmap.org/>

<sup>9</sup>Dostupné na <https://wego.here.com/>

<sup>10</sup>Dostupné na <https://www.google.com/maps>

7. Mapbox<sup>11</sup> poskytuje sadu nástrojov ktoré pomocou ktorých sa dajú jednoducho vytvárať aplikácie na mieru, patria medzi ne textúry, ilustrácie, značky, geocoding a iné. Online mapbox editor umožňuje rýchlo pracovať s rozhraním. Napriek tomu že Mapbox spustil svoje služby v roku 2010, jeho popularita stále rastie.

### 4.3 Požiadavky na aplikáciu

Aplikácia by mala byť voľne dostupná všetkým bez nutnosti zdlhovej a zbytočnej registrácie. Prednastavená konfigurácia filtrov realít umožní užívateľovi používať mapu hneď ako zadá do vyhľadávачa jej adresu. Jednoduchý minimalistický panel na boku mapy bude prekryvať len veľmi malú časť okna. Ak bude chcieť užívateľ zmeniť konfiguráciu mapy, tento panel si rozvinie a zadá požadované filtre. Priehľadným panelom môžeme docieľiť ešte menšieho prekryvania mapy. Treba si dať ale pozor na prekryvanie loga spoločnosti ktorá mapové rozhranie poskytuje. V prípade že bude zakryté môže nastať rozpor s pravidlami používania.

Panel bude ponúkať niekoľko zobrazení prvým bude zobrazenie realít na mape pomocou značiek. Farba značky bude odvíjať cenu nehnuteľnosti. Po kliknutí na značku sa zobrazia podrobnosti ako adresa, rozloha, cena, krátky popis, fotka a prepojenie na originálny inzerát. Tieto ponuky bude možné filtrovať na základe ceny, rozlohy, dispozície a portálu na ktorom sa nachádza. Pri vstupoch od užívateľa bude treba dbať na bezpečnosť a ošetriť SQL injections<sup>12</sup>.

Ďalšou možnosťou zobrazenia bude teplotná mapa. Konfigurácie bude podobná ako u zobrazení ponúk značkami. tzn zadanie ceny, rozlohy, dispozície a portálu vykreslí požadované nehnuteľnosti v podobe teplotného rozloženia. Dôležitý bude výber farby a rádiusu teplôt tak aby teplotná vrstva nesplývala s mapou a zafarbované oblasti príliš nezasahovali do seba ale zároveň odrážali ceny svojho okolia.

V ponuke bude možné prezeráť si aj časovú os vývoja cien na trhu. Užívateľ si zadá požadované rozpätie dátumov a vykreslí sa mu odpovedajúca teplotná mapa z ponúk ktoré boli v danom časovom úseku platné.

### 4.4 Realitné portály

Existuje veľké množstvo realitných portálov. Dôležitý je však výber tých portálov ktoré budú na získavanie dát vhodné, to znamená že z inzerátu bude možné extrahovať dáta. Zároveň je potrebné aby realitný portál bol dostatočne veľký a jeho spracovanie sa oplatí vzhľadom na čas potrebný pre prípravu skriptu.

Medzi najväčšie portály patria:

1. [www.topreality.sk](http://www.topreality.sk) - 250 000 ponúk,
2. [www.sreality.cz](http://www.sreality.cz) - 90 000 ponúk,
3. [www.reality.idnes.cz](http://www.reality.idnes.cz) - 70 000 ponúk,
4. [www.ceskereality.cz](http://www.ceskereality.cz) - 70 000 ponúk.

<sup>11</sup>Dostupné na <https://www.mapbox.com/>

<sup>12</sup>Chyba založená na možnosti manipulovať s dátami v databáze bez vlastníctva prístupových údajov



## 4.5 Potrebné dáta

Pre implementáciu mapy je treba mať k dispozícii údaje o realitách. Z vybraných realitných serverov sa získajú potrebné dáta. Medzi ne patrí adresa, rozloha, dispozícia, cena, odkaz na inzerát, odkaz na foto ponuky, krátky popis, dátum pridania, dátum odstránenia inzerátu pre časovú os a nakoniec informácie o presnej polohe čiže súradnice rovnobežky a poludníka. Väčšina údajov sa dá zistiť priamo z inzerátu no súradnice bude treba získavať prevodom z adresy. Dátum odstránenia inzerátu nebýva súčasťou inzerátu avšak dá sa zistiť v prípade že skript ktorý bude denne prehľadávať portály zistí že daná ponuka už neexistuje.

V prípade že dáta budú získavané z viacerých portálov bude potrebné rozlišovať duplicity inzerátov. Ak sa na danej adrese bude nachádzať viacero ponúk z rozličných serverov ale cena, rozloha alebo dispozícia bude iná tak sa táto ponuka môže považovať za inú. Tým pádom sa vyradí z výpočtov do teplotnej mapy pretože by sa javila ako jedna oblasť s veľkou - nepravdivou cenou. Naopak značka zobrazená bude ale bude musieť byť o niečo posunutá keďže súradnice budú pre tieto dve ponuky budú rovnaké.

## 4.6 Získavanie dát

Získavane dát môže byť implementované rôznymi spôsobmi. Líšia sa v rýchlosti a spôsobe implementácie. Vo všetkých prípadoch ide o získanie konkrétneho div<sup>13</sup> elementu.

### 4.6.1 Regulárne výrazy

Regulárny výraz je sekvencia znakov ktorá opisuje množinu reťazcov podľa určitých syntaktických pravidiel.[20] Umožňuje tak v texte vyhľadávať tieto reťazce. Mnohé jazyky podporujú regulárne výrazy už v štandardných knižniciach. Pomocou regulárneho výrazu je možné lokalizovať element na stránke a získať jeho obsah. Po orezaní niektorých nechcených medzier a ďalších znakov môžu byť dáta ďalej spracované.

### 4.6.2 Simple HTML DOM parser

Tento open source parser<sup>14</sup> bol vyvinutý pre urýchlenie práce s DOM<sup>15</sup> modelom. Je založený na získavaní jednotlivých elementov. Pre použitie je nutné si ho stiahnuť a pomocou vložiť do projektu.

## 4.7 Nasadenie na server

Aby aplikácia mohla bežať je treba aby mala svoj server. Naskytuje sa niekoľko možností kam sa dá aplikácia nasadiť, od fyzického serveru až po raspberry pi. Nasledujúca podkapitola sa bude zaoberať analýzou niektorých z nich.

### 4.7.1 VPS

Na trhu je veľké množstvo spoločností ktoré poskytujú privátny virtuálny server. Zákazník má na výber z množstva operačných systémov a má nad ním úplné práva, môže si tak na

---

<sup>13</sup>Definuje časť alebo sekciu v HTML kóde

<sup>14</sup>Program ktorý má na starosti syntaktickú analýzu

<sup>15</sup>Rozhranie ktoré umožňuje dynamicky pristupovať a meniť obsah, štruktúru a štýl dokumentu

server nainštalovať všetky potrebné programy. Vďaka tomu že sú virtuálne ich vytváranie a správa je o mnoho jednoduchšia ako u dedikovaných fyzických serverov. Takýto server je veľmi flexibilný, dokáže sa prispôbiť zákazníkovi na mieru výberom počtu jadier, veľkosťou operačnej pamäti či kapacitou disku. Medzi ďalšie služby patrí aj vytváranie snapshotov<sup>16</sup> či migrácia serveru za behu v prípade problémov. Za nižšiu cenu ale poskytujú aj menší výkon oproti fyzickým, kvôli hardvéru ktorý zdieľajú s ostatnými VPS.

#### 4.7.2 Raspberry pi

Raspberry pi je jedno-doskový počítač výviný firmou Raspberry pi foundation. Najnovší model pi 3 disponuje 4 jadrovým procesorom o takte 1.2. Pomerne jednoduchým pretaktovaním sa dá toto číslo zvýšiť. Podporuje niekoľko operačných systémov od Arch linux až po Ubuntu Mate [19]. Vďaka nízkej cene je jednou z možných variant pre aplikáciu cenovej mapy.

#### 4.7.3 Dedikovaný server

Ďalšou možnosťou je dedikovaný fyzický server. K prenájatým serverom poskytuje väčšina spoločností štandard v podobe zálohovaného napájania a kvalitného internetu.

### 4.8 Cieľová skupina

Cenová mapa má byť pomôckou pre laickú verejnosť ale aj pre realitných maklérov v orientovaní sa na trhu s nehnuteľnosťami. Z cieľovej skupiny vyplývajú aj nároky na aplikáciu. Poskytované dáta by mali byť čo najčastejšie aktualizované a prehľadné.

### 4.9 Zabezpečenie webových stránok proti robotom

Intertnetový robot je program ktorý opakovane vykonáva nejakú rutinnú činnosť na internete, zvyčajne zbiera dáta a odosiela ich vzdialenému serveru. Jedným zo známych robotov je Google bot ktorý získava informácie o stránkach a používa ich v Google vyhľadávači. Obmedzenie týchto robotov je možné pomocou súboru `robots.txt` ktorý sa vkladá do koreňového adresáru webu. V tomto súbore sú uvedené časti webu ktoré nemajú byť prístupné robotom.[21] Využíva štandard pre zakázanie prístupu robotom, čo je protokol obsahujúci niekoľko málo príkazov, pomocou ktorých je možné obmedziť prístup ke jednotlivým sekciam webu podľa druhu prehliadača. Využíva sa tu možnosť rozlíšiť napríklad mobilný a webový prehliadač. Pomocou tohto súboru je možné obmedzovať prístup pre nasledovné druhy súborov:

1. zdrojové súbory - súbor `robots.txt` môže byť použitý na blokovanie prístupu k zdrojovým súborom ako sú napríklad nepodstatné obrázky, skripty alebo súbory štýlov(.css), google neodporúča blokovanie zdrojových súborov, pri jeho prezeraní stránky nemusí vyhodnotiť analýzu správne a je možné že sa do vyhľadávača stránka nedostane,
2. obrázkové súbory - súbor `robots.txt` môže zabrániť zobrazovaniu obrázkom vo vyhľadávačoch,

---

<sup>16</sup>Záloha stavu systému v určitom čase

- iné ako obrázkové súbory - je možné zabrániť prístupu aj k iným súborom. Týmto sa odporúča hlavne riadenie prevádzky na stránke.

Riadenie prevádzky môže byť užitočné v prípade kedy stránku navštevuje veľké množstvo robotov a zbytočne tak vyťažujú procesorový čas serveru a prenášajú veľké množstvo dát. Tento súbor môže byť však aj jedným zo SEO<sup>17</sup> nástrojov. Jeho nesprávne použitie však môže vyústiť v stratu pozície vo webových prehliadačoch. Roboty dokážu rozoznávať duplicitný obsah stránok, na základe toho môžu umiestniť stránku na nižšej pozícii vo vyhľadávaní. Umiestnením tohoto duplicitného obsahu do súboru `robots.txt` je možné tomuto zabrániť. Niektoré prehliadače môžu tento súbor ignorovať alebo ho inak interpretovať aj na toto je treba brať ohľad.

Ďalšou možnosťou ako zabrániť robotom v prehľadávaní obsahu stránky je nastavenie špeciálneho meta tagu v HTML dokumente. Nasledujúca ukážka obsahuje meta tag ktorý zabraňuje vyhľadávačom indexovať stránku.

```
<meta name="robots" content="noindex" />
```

Obr. 4.1: Meta tag pre obmedzenie indexovania

## 4.10 Prevencia blokovania robotov na stránkach

Pri získavaní dát z webových stránok je treba dbať na určité pravidlá ktorými sa predíde zablokovaniu robota[22]. Väčšina stránok ochranu proti robotom nemá, pretože to môže v určitom zmysle obmedziť obyčajného užívateľa. Existuje niekoľko metód ktorých dodržovanie je zásadné pre zníženie šance zablokovania robota:

- pomalší skript - pri predĺžení času medzi jednotlivými požiadavkami sa skript bude javiť ako človek, dôležité je taktiež aby časové intervaly neboli stále rovnaké,
- obmieňanie ip adresy - server dokáže jednoducho vystopovať požiadavky robota z jednej ip adresy, preto obmena ip adresy môže zásadne ovplyvniť detekciu robota,
- vyhýbanie sa honeypotom - honeypot v preklade "hrniec medu" je mechanizmus pre detekciu malwaru<sup>18</sup>, v tomto prípade ide o zvyčajne o odkazy ku ktorým obyčajný užívateľ nemá prístup,
- rešpektovanie súboru `robots.txt` - pri porušení pravidiel uvedených v tomto súbore môže byť robot ľahko detekovaný,
- obmena spôsobu získavania dát - pri použití rovnakých techník pre získavanie dát, je rozpoznanie robota pre server jednoduché, obmieňanie prístupu akým sa dáta získavajú môže prispieť k ťažšej detekcii robota.

Zistiť obsah súboru `robots.txt` je jednoduché keďže ide o verejný súbor a zároveň sa nachádza v koreňovom adresári. Zadaním názvu súboru za názvom webovej stránky sa zobrazí jeho obsah. Napríklad súbor `robots.txt` pre server `facebook.com` sa dá zobraziť pomocou adresy `www.facebook.com/robots.txt`. Môžeme si tu všimnúť že obsahuje aj obmedzenia pre českého robota *seznambot*<sup>19</sup>.

<sup>17</sup>Optimalizácia pre webové prehliadače

<sup>18</sup>Program určený ku vniknutiu do systému alebo poškodeniu systému

<sup>19</sup>Dostupné na: <https://napoveda.seznam.cz/cz/fulltext-hledani-v-internetu/seznambot/>

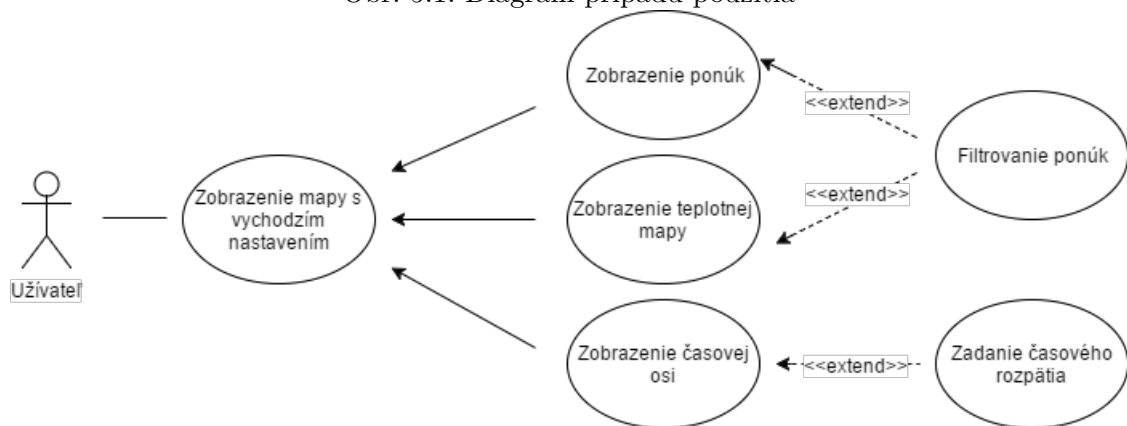
# Kapitola 5

## Návrh

### 5.1 Diagram prípadu použitia

Diagram prípadu použitia popisuje možnosti pri práci s mapou. Pri spustení aplikácie v jeho prehliadači sa automaticky zobrazí mapa so značkami ponúk bez nutnosti registrácie či počiatočného nastavenia aplikácie. Ďalej môže užívateľ vybrať požadované zobrazenie ponúk a to v podobe značiek, teplotnej mapy alebo časovej osi. Na časovej osi sa zadaním rozsahu zobrazí teplotná mapa z daného obdobia. Pri značkách a teplotnej mape bude možné filtrovať ponuky podľa ceny, rozlohy, portálu a dispozície. V tomto prípade sa budú zobrazovať iba aktuálne ponuky.

Obr. 5.1: Diagram prípadu použitia



### 5.2 Entitno-vzťahový diagram

Entitno vzťahový diagram zobrazuje usporiadanie databáze. Budú sa v nej nachádzať 3 tabuľky a to PROPERTY, OFFER a LOCATION. Vzhľadom k tomu že daná nehnuteľnosť sa môže nachádzať na viacerých serveroch je potrebné aby boli tabuľka PROPERTY a OFFER vo vzťahu 1:N<sup>1</sup>. V prípade že nastane tento stav nebude nutné pridávať ďalšiu položku do tabuľky s nehnuteľnosťou ale pridá sa iba ďalší záznam do tabuľky s ponukami. To isté

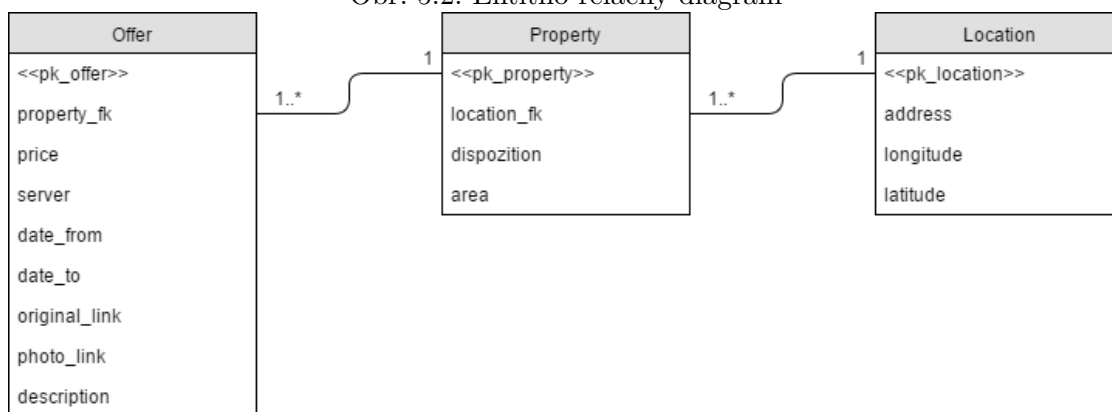
<sup>1</sup>Priradenie jednému záznamu viacero záznamov z druhej tabuľky

bude platiť aj pre prípad kedy tú istú adresu budú mať dve nehnuteľnosti, záznam sa potom pridá iba do tabuľky PROPERTY s cudzím kľúčom rovnakej adresy.

Tabuľka PROPERTY bude obsahovať identifikačné číslo ponuky, cudzí kľúč do tabuľky s adresou, dispozíciu a rozlohu. Tabuľka OFFER bude obsahovať identifikačné číslo ponuky, cenu, server na ktorom sa nachádza, dátum zaznamenania do databáze, dátum kedy bol inzerát naposledy platný ,odkaz na inzerát, odkaz na fotku ,krátky popis a cudzí kľuč do tabuľky PROPERTY.

Posledná tabuľka LOCATION bude uchovávať dáta o adresách a ich presných súradniciach tj. poludníka a rovnobežky. Pomocou primárneho kľúča bude možné spájať všetky ponuky.

Obr. 5.2: Entitno relačný diagram



## Kapitola 6

# Použité technológie

### 6.1 XAMPP

XAMPP je sada open source softvéru pozostávajúca z Apache serveru, MariaDB databáze, php a perl interpretu. Tento multiplatformový softvér slúži programátorom pri vývoji a testovaní lokálneho web serveru. Vďaka tomu že väčšina fyzických web serverov používa rovnaké služby ako XAMPP, prenos lokálneho serveru na iný server je potom veľmi jednoduchý. Okrem iného poskytuje aj lokálny mail server a FTP<sup>1</sup> server. XAMPP bol vybraný vďaka jeho multiplatformnosti. Pri nasadení na iný server teda nebude nutné dbať na odlišnosti v použítom softvéri ako tomu je u LAMP, WAMP alebo MAMP sady kde každá z nich je určená na iný operačný systém.[24]

### 6.2 Apache

Apache patrí medzi najpoužívanéjšie webové servery. Tento server podporuje niekoľko skriptovacích jazykov medzi ktoré patria: BeanShell, Javascript, Groovy, Python, PHP a Ruby. Umožňuje prístup k webovej aplikácii a vybavuje jednotlivé požiadavky klientov.[1]

### 6.3 MariaDB

MariaDB je voľne dostupný viacúčítateľský, viacvláknový databázový server slúži pre uchovanie perzistentných dát. Je podporovaný viacerými platformami ako napríklad Linux, Windows či Solaris. Databáze vytvorené na tomto serveri obsahujú jednu alebo viac tabuliek kde stĺpec tabuľky určuje dátový typ a v riadku sa nachádza záznam. Komunikuje s relačnou databázou pomocou jazyka SQL.[15]

### 6.4 PHP

PHP je skriptovací jazyk používaný hlavne pre programovanie serverových aplikácií a na vývoj webových stránok ale aj pre programovanie všeobecne. PHP prebralo veľa vlastností od jazyka C. Od verzie 5 podporuje taktiež objektovo orientované programovanie. Patrí medzi interpretované jazyky a je voľne dostupný. Je jednoduchý a flexibilný hlavne pre začiatočného programátora.[9]

---

<sup>1</sup>Protol pre prenos súborov po sieti

Medzi jeho výhody[17] patrí:

1. hybridný jazyk - funkcionálny aj objektovo orientovaný,
2. open source,
3. veľké množstvo frameworkov,
4. rozsiahla kvalitná dokumentácia.

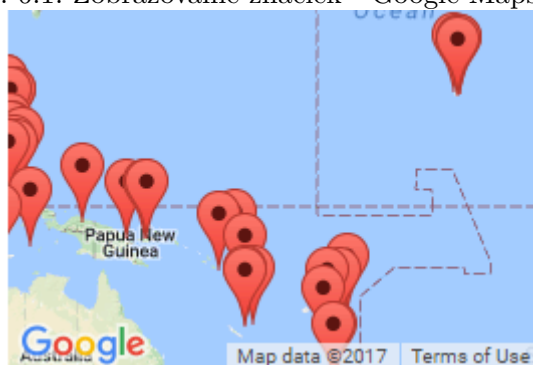
Nevýhody:

1. interpretovaný jazyk,
2. slabo typovaný,
3. jeho jednoduchosť môže viesť k zlým návykom.

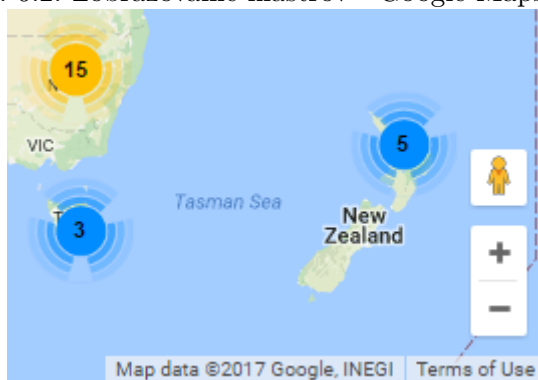
## 6.5 Google maps

Toto aplikačné rozhranie bolo vybrané na základe predchádzajúcej analýzy mapových rozhraní. Jeho prehľadná dokumentácia spolu s ukážkami kódu ktoré je možné meniť a sledovať v reálnom čase rozhodla pri výbere mapového aplikačného rozhrania. Možností ako zobrazovať dáta ponúka hneď niekoľko.[11]

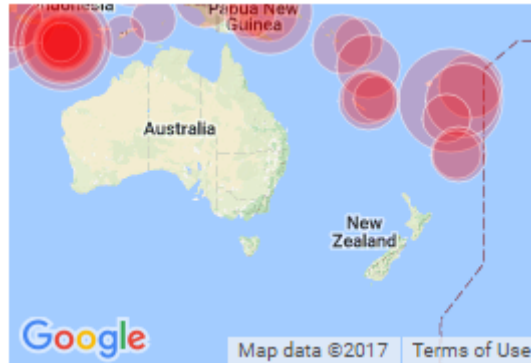
Obr. 6.1: Zobrazovanie značiek - Google Maps API



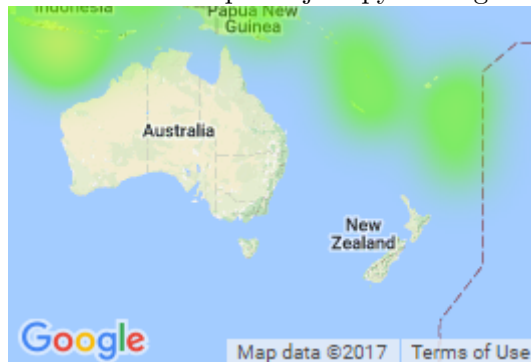
Obr. 6.2: Zobrazovanie klastrov - Google Maps API



Obr. 6.3: Zobrazovanie kruhmi - Google Maps API



Obr. 6.4: Zobrazovanie teplotnej mapy - Google Maps API



Nás bude zaujímať teplotné zobrazenie. Pomocou presných súradníc rovnobežiek a poludníkov dokáže toto API vykresľovať teplotné mapy na základe počtu súradníc v okolí ale aj na základe váhy súradníc týmto sa bude simulovať cena reality kedy drahšia realita dostane väčšiu váhu a lacnejšia menšiu. Ďalej je možné upraviť rádius jednotlivých vykresľovaných bodov, meniť farby teplôt či priehľadnosť.

Každá realita bude mať svoju značku na mape. Farba tejto značky bude určovať cenové rozpätie do ktorého patrí. Užívateľ si bude môcť vybrať či si zobrazí iba teplotnú vrstvu alebo značky.

Pre používanie Google aplikačného rozhrania je potrebné si vyžiadať API kľúč ktorý bude identifikovať aplikáciu. Pokiaľ aplikácia neprekročí 25 000 prístupov za deň[12] tak je toto rozhranie zadarmo, so zvyšujúcimi prístupmi sa zvyšuje aj cena používania. Google ďalej poskytuje pomocou tohto kľúča sledovanie rôznych štatistík ako napríklad sledovanie prístupov, priemernú odozvu či chybovosť aplikácie. Google si vyhradzuje právo zablokovania aplikácie v prípade porušenia licenčných pravidiel. Medzi ktoré patrí aj nutnosť zobrazenia Google loga presne na takom mieste na akom mieste akom je. Ďalej aj modifikovanie lišty so zmluvnými podmienkami a názvom spoločnosti je zakázané. Rozsiahla dokumentácia detailne popisuje jednotlivé časti rozhrania. Každý ukázkový kus kódu je dobre vysvetlený s možnosťou si ho vyskúšať prípadne meniť a sledovať zmeny v nástroji JSFiddle.



## 6.6 Geocoding

Geocoding je proces prevodu adresy na geografické súradnice (rovnobežky a poludníky), ktoré budú ďalej použité pre túto aplikáciu. Geocoding je potrebný pre túto aplikáciu pretože väčšina realitných serverov poskytuje iba adresu bez konkrétnych geografických súradníc. Opakom geocodingu je reverzný geocoding ktorý dokáže zo súradníc získať adresu. Google taktiež poskytuje rozhranie pre geocoding alebo je možné ho využívať pomocou HTTP požiadavky. Pre toto javascriptové API existuje spoplatnenie podobne ako pri mapovom rozhraní, povolených je 2 500 požiadaviek/deň, pri prekročení tohoto limitu je služba nedostupná.[10]

Ďalším obmedzením je počet dotazov za sekundu. Pri vykonávaní geocodingu hneď za sebou je povolených iba 12 dotazov, preto bude nutné urobiť skript, ktorý sa bude neustále dopytovať na nové položky. Medzi jednotlivými dotazmi bude potrebné stanoviť taký interval pri ktorom dotazovanie nebude obmedzované. Získané údaje sa potom budú ukladať do databázy takže nebude potrebné pri každom načítaní stránky sa znova dožadovať súradníc od Geocoderu. Týmto sa zabezpečí aj rýchlejšie vykresľovanie mapy a možnosť pracovať aj pri výpadku týchto služieb.

## 6.7 Javascript

O vytváranie mapy, aplikovanie filtrov, interakciu s užívateľom sa bude starať javascript. Dokáže automaticky spravovať pripojenie do Google Maps Service, sťahovať mapy a ovládať ich. Jeho vstupom bude súbor s objektom ktorý bude obsahovať všetky potrebné údaje pre aplikáciu. Výstupom potom bude mapa zobrazená užívateľovi podľa jeho nastavenia[14].

## 6.8 PhantomJS

Nie všetky realitné portály obsahujú vo svojom zdrojovom kóde potrebné údaje pre aplikáciu, niektoré sa zobrazia až po spustení javascriptu. Preto bude potrebné javascript na stránkach spustiť a až potom získavať dáta. Pre tento účel posluží program PhantomJS ktorý mnoho iných vecí okrem získavania zdrojového kódu. PhantomJS dokáže meniť proxy server<sup>2</sup> pomocou ktorého získava obsah stránky. Detekcia robota používajúceho tento PhantomJS je teda podstatne sťažená[16]. Bol vybraný kvôli jeho multiplatformovosti a kvalitnej dokumentácii.

## 6.9 Windows Task Scheduler

Aby bolo spúšťanie skriptu automatizované, je treba použiť plánovač úloh. Windows Task Scheduler je aplikácia ktorá dokáže automaticky spúšťať príkazy v presný čas. V závislosti na serveri na ktorom bude aplikácia bežať sa nastaví také odstupky medzi úlohami aby sa zaznamenali ponuky zo všetkých portálov. Bol vybraný vzhľadom k tomu že je súčasťou operačného systému windows 10.[23]

<sup>2</sup>Server ktorý umožňuje klientom nepriame pripojenie k inému serveru

## 6.10 PhpMyAdmin

PhpMyAdmin je bezplatný softvér ktorý umožňuje správu MariaDB databázy cez webové rozhranie. Bol napísaný v PHP a má rozsiahlu dokumentáciu. Pomocou užívateľského rozhrania je možné spravovať celú databázu, prezerať štatistiky, importovať/exportovať dáta, prezerať tabuľky v dizajnéri.[18]

## Kapitola 7

# Implementácia

### 7.1 Aplikácia

Php skripty budú hlavnou časťou pri získavaní a spracovávaní dát, ktoré bude prebiehať v niekoľkých fázach. Prvou bude získanie počtu všetkých inzerátov na danom portáli. Následne sa podľa tohoto počtu prejdú všetky stránky s ponukami a vyhľadajú sa na nich pomocou regulárnych výrazov všetky odkazy inzeráty. Zapišu sa do súboru a prejde sa na ich spracovanie. Každý jeden odkaz na inzerát prejde skriptom ktorý v ňom vyhledá potrebné dáta pomocou regulárnych výrazov. Po pripojení sa na databázu sa uložia získané dáta. Tento proces sa postupne vykoná na niekoľkých portáloch . Vznikne tak naplnená databáza. Ďalej bude potrebné získať presné súradnice adries. Skript ktorý sa o toto bude starať bude využívať *geocoder* ktorý bude popísaný nižšie. Prejde všetky adresy a získa k nim presné súradnice rovnobežky a poludníka. Posledný skript bude vytvárať súbor ktorý bude obsahovať všetky výsledné ponuky s potrebnými dátami.

Spúšťaním týchto skriptov v určitých intervaloch získame neustále aktualizované dáta. Aby sa nemuseli prechádzať inzeráty ktoré už sú v databáze, získajú sa iba odkazy na všetky inzeráty a urobí sa rozdiel týchto nových odkazov a starých odkazov z predošlého cyklu. Nové odkazy sa ďalej spracujú a uložia do databáze. Staré odkazy ktoré už na portáli nie sú sa budú považovať za vymazané ponuky. Nastaví sa im dátum expirácie ktorý, bude využívaný pri zobrazovaní ponúk na časovej osi, kedy si užívateľ vyberie len požadovaný časový úsek a zobrazia sa mu len ponuky z daného časového obdobia.

Hlavný skript ktorý sa spúšťa pravidelne v určitých intervaloch obsahuje 5 častí: spracovanie portálu *sreality.cz*, *topreality.cz*, *realityidnes.cz*, geocoder a vytvorenie súboru s dátami. Po poslaní hlavičky sa spustí sťahovanie dát z prvého portálu. Skript *sreality.php* najprv vymaže súbor so starými odkazmi na ponuky v súbore *OldLinks.txt* a premenuje súbor s novozískanými odkazmi v predchádzajúcom cykle *NewLinks.txt* na *OldLinks.txt* . S týmito odkazmi sa budú porovnávať nové odkazy podľa ich rozdielu sa zistí ktoré ponuky boli deaktivované a ktoré ponuky boli pridané na portál. Pomocou skriptu *getPagesCount.js* sa zistí aktuálny počet ponúk na portáli. Tento skript pomocou regulárneho výrazu získa počet ponúk a túto hodnotu pošle do pôvodného skriptu. Nasledujúce ukážky kódu slúžia pre získanie informácií z portálu *www.sreality.cz*. Počet všetkých ponúk sa vydelením počtom ponúk ktoré sa nachádzajú na jednej stránke (v prípade *sreality.cz* je to 20) a získa sa tak počet stránok na ktorých sú ponuky. Keďže na niektorých stránkach je kód skrytý až do spustenia javascriptu je nutné použiť PhantomJS, ktorý získa celý obsah stránky. Následne pomocou regulárneho výrazu vyhledá všetky odkazy na ponuky a vráti ich v poli pôvodnému skriptu. Ten si všetky odkazy uloží a zapíše ich do súboru *NewLinks.php*. Po získaní

```
var myRegex=/(\d[^\<]*)</span>&nbsp;nalezených/g;
match = myRegex.exec(page.content);
console.log(match[1]);
```

Obr. 7.1: Zisťovanie počtu ponúk

```
/\detail\prodej\byt\[^\]*\[^\]*\[0-9]*g;
```

Obr. 7.2: Regulárny výraz pre zistenie všetkých odkazov ponúk na jednej stránke

nových odkazov sa prechádza na vytvorenie rozdielu starých a nových odkazov. O toto sa stará skript `LinksDiff.php`. Súbor s odkazmi sa načítajú do polí a vstavaná php funkcia `array_diff()` vráti novo pridané a staré zrušené odkazy. Zrušeným odkazom sa nastaví dátum vymazania a novo pridané odkazy sa ďalej spracujú. Nasleduje spracovanie jednotlivých ponúk. Nové ponuky sa po jednom odkaze načítavajú zo súboru a posielajú funkcií `getSrealityData()` ktorá získa potrebné dáta. Najskôr sa spustí na stránke z odkazu javascript pomocou už zmieneného PhantomJS. Celý obsah stránky potom prejde niekoľkými regulárnymi výrazmi tak aby sa získala adresa, cena, dispozícia, rozloha, popis a odkaz na fotku. Nie vždy sa podarí získať kód stránky, preto má PhantomJS stanovený čas po ktorom vracia hodnotu `false` a prechádza sa na druhú ponuku. Môže nastať aj situácia kedy medzi zapísaním odkazu a spracovaním sa ponuka deaktivovala, preto ako prvý regulárny výraz hľadá na stránke kľúčové slová ktoré určia neaktívny inzerát. 7.3 Predíde sa tak zbytočnému zisťovaniu informácií o neexistujúcej ponuke. Ďalší regulárny výraz vyhľadáva v HTML

```
preg_match("/Je mi líto, inzerát neexistuje/", $output, $match)
```

Obr. 7.3: Zisťovanie aktuálnosti inzerátu

dokumente adresu ponuky. Adresa sa portáli *sreality.cz* sa nachádza v elemente s názvom `location-text`. Pre portál *topreality* je to zase element *Lokalita* a pre portál *reality.idnes* je to element `realAddress`. Nasleduje regulárny výraz pre zistenie ceny. Element v ktorom sa

```
preg_match("/location-text ng-binding">([^\<]+)/", $output, $matches);
```

Obr. 7.4: Zisťovanie adresy ponuky

nachádza cena obsahuje slovo *price*. Podľa toho kľúčového slova je možné element lokalizovať a získať tak cenu ponuky. Regulárny výraz pre zistenie rozlohy hľadá v texte sekvenciu

```
preg_match("/itemprop=\"price\" class=\"ng-binding\">(\d+)/", $output, $matches);
```

Obr. 7.5: Zisťovanie ceny ponuky

znakov  $m^2$ . Podľa nich dokáže zistiť pozíciu rozlohy v dokumente. Následne si uloží 1 až 3 číslice nachádzajúce sa pred touto sekvenciou, ktoré predstavujú rozlohu. Krátky popis

```
preg_match("/(\d{1,3}&nbsp;)+m²/", $output, $matches)
```

Obr. 7.6: Zisťovanie rozlohy nehnuteľnosti

ponuky sa nachádza medzi tagmi `<p>` a `</p>`. Regulárny výraz vyhľadá tieto tagy a uloží si text medzi nimi. Následne je tento text skrátený na 100 znakov aby nezaberal príliš

```
preg_match_all("/<\\s*p[>]*(<^<*)<\\s*\\|\\s*p\\s*>/", $output, $matches);
```

Obr. 7.7: Zisťovanie popisu ponuky

veľa miesta v databázi. Odkaz na fotku bude použitý v náhlade ponuky pri zobrazovaní na mape. Ďalší regulárny výraz vyhledá na stránke odkaz na fotku. Po získaní všetkých dát

```
preg_match_all("%src=\"([^\"]*res[^\"]*)%", $output, $matches);
```

Obr. 7.8: Získavanie odkazu na fotku

sa prechádza na vkladanie do databázy. Tu môžu nastať 4 situácie:

1. ponuka už existuje v databáze,
2. ponuka už existuje v databáze ale na inom serveri,
3. ponuka ešte neexistuje, ale adresa áno,
4. ponuka a adresa ešte v databáze neexistuje,

Selekty<sup>1</sup> do databáze zistia ktorý prípad nastal, a podľa toho sa vloží záznam buď do všetkých tabuliek alebo iba do niektorých. Nasledujúci dotaz do databáze zisťuje duplicitu ponuky. Pole s názvom \$data uchováva získane údaje o ponuke z realitného serveru. Ak

```
SELECT * FROM location,property,offer
WHERE property_id=property_fk AND
address=$data[0] AND
location_id=location_fk AND
price=$data[1] AND
disposition=$data[2] AND
area=$data[3] AND
server=$server AND
original_link=$data[6]
```

Obr. 7.9: Zisťovanie duplicity ponúk

tento selekt vráti neprázdnu množinu záznamov, daná ponuka už existuje a do databáze sa nepridáva. Ďalší dotaz zistí či už daná ponuka existuje v databáze ale na inom serveri, kľúčová tu je nerovnosť na položku server ktorá určuje výsledok selektu. Posledný select zistí či daná adresa existuje v databáze. V prípade absencie adresy v databáze sa vkladá táto adresa do príslušnej tabuľky.

Objekt ktorý spracoval vloženie do databáze nahlási počet chýb ktoré nastali a zapíše ich do súboru. Tento proces sa opakuje na ďalších portáloch s menšími zmenami v regulárnych výrazoch.

Po dokončení spracovania ponúk sa prechádza na Geocoding. V tabuľke location sa nájdu všetky adresy u ktorých chýbajú presné súradnice. Táto adresa sa pošle pomocou http requestu<sup>2</sup> Google Geocodéru. Následne sa skontroluje status requestu, ak prebehol v poriadku, presné súradnice sa pripíšu do záznamu v tabuľke. V prípade že status requestu obsahuje "ZERO\_RESULTS", presné súradnice neboli nájdené a geocodéru sa posielajú

<sup>1</sup>Vracia množinu záznamov z jednej alebo viacerých tabuliek.

<sup>2</sup>Požiadavka klienta pre server pomocou HTTP protokolu

```
SELECT * FROM location,property,offer
WHERE property_id=property_fk AND
address=$data[0] AND
location_id=location_fk AND
price=$data[1] AND
dispozition=$data[2] AND
area=$data[3] AND
server!=$server
```

Obr. 7.10: Zistovanie serveru ponuky

```
SELECT * FROM location WHERE
address=$data[0]
```

Obr. 7.11: Zistovanie presencie adresy

iba časť adresy tak aby sa našla aspoň približná lokácia. Adresy ktorým neboli nájdené súradnice sa zapisujú do logu<sup>3</sup> pre neskoršie riešenie problému.

Poslednou fázou je vytvorenie skriptu s dátami. Všetky ponuky v databáze sa zapíšu do súboru. Ak je ponuka neúplná tj. chýba jej adresa, lokácia alebo iná položka do súboru sa potom nezapisuje ale ukladá sa do logu pre neskoršie ladenie chýb. Ďalej môže nastať situácia kedy dve ponuky majú rovnaké súradnice, na mape by sa tak prekrývali. Preto sa pred zápisom zisťuje duplicita adresy a v prípade zhody sa druhá značka posunie o niekoľko metrov.

## 7.2 Server

Pre testovacie účely bude aplikácia bežať serveri s Windows 10. Po nainštalovaní XAMPP sady bude mať server všetko potrebné. Toto riešenie má určité nedostatky nakoľko môže nastať výpadok internetu či prísunu elektriny. Zotavovanie z výpadku nemusí byť veľký problém. Aplikácia sa spustí znova pričom môže nastať situácia že sa bude pokúšať vkladať do tabuliek dáta ktoré tam už sú. Nastane zbytočné spomalenie aplikácie a plytvanie procesorového času. Tomuto plytvaniu sa dá zabrániť zálohovaním súboru ktorý uchováva odkazy na ponuky.

## 7.3 Databáza

Databáza MariaDB sa bude starať o správu dát. Cez poskytované rozhranie phpMyAdmin sa dá databáza jednoducho spravovať. Na výber poskytuje niekoľko typov databáz. Vo východnom nastavení sú záznamy ukladané do tabuliek typu INNODB, ktorý ponúka kompromis medzi výkonom a spoľahlivosťou. Jeho kľúčovými vlastnosťami sú: využívanie ACID modelu<sup>4</sup>, možnosť využívania viacerými užívateľmi naraz, podpora cudzích kľúčov pre zachovanie konzistencie medzi tabuľkami, ukladanie dát také aby sa minimalizovalo vyhľadávanie na základe primárnych kľúčov.

<sup>3</sup>Záznam z činnosti programu

<sup>4</sup>Sada vlastností ktoré musí spĺňať databázová transakcia(atomickosť, konzistencia, izolovanosť, trvanlivosť)

Po vytvorení databáze v kolácii<sup>5</sup> pre český jazyk, sú postupne vytvárané tabuľky `location`, `offer` a `property`. Ďalej je potrebné ich spojiť cudzími kľúčmi. Nasledujúce riadky ukazujú prepojenie tabuľky `offer` a `property`.

```
ALTER TABLE offer ADD CONSTRAINT propertyFk FOREIGN KEY
(property_fk) REFERENCES property (property_id)
```

Obr. 7.12: Vytváranie cudzieho kľúča

## 7.4 Dizajn

Najväčšiu časť stránky zaberá element `map` ktorý obsahuje mapu. Ďalej sú tu 4 panely, ktoré ale nie sú v počítačom stave viditeľné. Po rozkliknutí základného menu sa rozvinie panel s možnosťou zobrazenia ponúk, časovej osi alebo teplotnej mapy.

Užívateľ má možnosť interagovať s aplikáciou pomocou `checkboxov`<sup>6</sup>, tie odfiltrujú ponuky na základe požadovaných portálov a dispozície. Ďalej panel obsahuje vstupy pre zadávanie ceny, rozlohy a dátumu.

## 7.5 Javascript

Interakciu s užívateľom a vytváranie máp zabezpečuje javascript. Načítaním stránky sa vytvoria 3 inštancie objektov : `googleMap`, `heatLayer` a `navBar`.

### 7.5.1 GoogleMap

Tento objekt obsahuje metódy pre vytváranie mapy a aplikovanie filtrov. Prvou metódou je `initMap()`, ktorá inicializuje mapu. Nastaví počiatočný zoom mapy, vycentruje mapu na zadané súradnice a vypne východzie užívateľské rozhranie. Akonáhle si pripojí do DOM modelu potrebné dáta, metóda `eqfeed_callback()` spracuje tieto dáta. Pre každý jeden záznam vytvorí na mape značku. Podľa ceny zadanej v zázname sa rozhodne o farbe značky. Z databáze sa zistí ponuka s najvyššou cenou a bude mať tmavo červenú farbu. Farba značiek ďalších ponúk bude určená pomerom k najvyššej cene. Každéj značke sa pridá listener<sup>7</sup>, tak aby po kliknutí zobrazil náhľad fotky, cenu a odkaz na ponuku. Všetky tieto značky sa uložia do pomocného poľa pre ten prípad keď sa bude zobrazovať teplotná mapa a značky bude treba vymazať. Do ďalšieho pomocného poľa sa uložia informácie o súradniciach a príslušnej cene. Metóda `showOffers()` sa volá pri každom aktualizovaní nejakého filtru alebo pri prechode z teplotnej mapy na mapu so značkami. Ako prve sa vymaže z mapy teplotná vrstva ak tam nejaká bola a zobrazia sa iba tie značky ktoré vyhovujú všetkým zadaným filtrom. `VisualiseHeatMap()` Slúži na vykreslenie teplotnej vrstvy. Zároveň sa jej nastaví požadované farebné rozloženie a rádius jednotlivých ponúk. `AllCoords_callback()` Vykreslí teplotné vrstvy v zadanom časovom úseku.

<sup>5</sup>Spôsob zoradenia znakov v danom kódovaní

<sup>6</sup>HTML element vo forme zaškrťavacieho políčka

<sup>7</sup>Procedúra alebo funkcia ktorá sa spúšťa v prípade nastatia určitej udalosti

### 7.5.2 HeatLayer

`HeatMapRange()` Táto metóda najskôr vymaže značky z mapy ak tam nejaké boli, následne pripojí do DOM modelu dáta s ponukami ktoré už nie sú aktuálne ale poslúžia na vykresľovanie časovej osi. `HeatMapRange()` metóda je volaná pri zmene niektorej hodnoty zo vstupov či `checkbox`ov alebo pri prvom zobrazení teplotnej mapy. Vstup od užívateľa je prevedený do formátu v akom sa ukladá do databáze aby sa mohol porovnávať. V cykle sa prechádzajú postupne všetky pripojené záznamy, ak rozsah platnosti inzerátu spadá do rozsahu zadaného užívateľom daná ponuka sa vloží do pola ktoré sa na konci cyklu pošle funkcii na vykresľovanie(`VisualiseHeatMap()`).

### 7.5.3 NavBar

Objekt ktorý sa stará o užívateľské rozhranie pozostáva z 6 metód. Po kliknutí na menu, sa zavolá metóda `openNav()` a otvorí bočný panel s možnosťou zobrazenia ponúk, teplotnej mapy alebo časovej osi. Po kliknutí na zobrazenie ponúk metóda `openOffers()` vysunie panel s filtrami. To isté platí pre metódy `openTimeLine()` a `openHeatMap()`. Ďalej ma užívateľ možnosť návratu na prvý panel metódou `backNav()` alebo zavretie panelu pomocou metódy `closeNav()`

### 7.5.4 Ošetrovanie vstupu

Keďže sa v aplikácii nachádzajú vstupy pre užívateľa, je nutné ich ošetriť aby nedochádzalo k nekonzistencii pri vykonávaní výpočtov. Pri vykresľovaní ponúk funkciou `showOffers()` sa zavolá funkcia `checkNumber()` ktorá rozhodne či sa jedná o validné číslo. Jednoduchou podmienkou na číslo väčšie rovné ako 0 sa rozhodne o vykreslení. V prípade že bol zadaný zlý vstup upozorní aplikácia užívateľa s vyskakovacím oknom a vyzve ho k zadaniu správneho čísla. Pri vykresľovaní teplotnej mapy v časovom rozpätí je kontrola vstupu o niečo zložitejšia. Daný dátum sa skontroluje regulárnym výrazom, ktorý musí mať formát `dd.mm.rrrr`, v prípade že je v poriadku rozdelí sa na 3 časti a kontrolujú sa zvlášť dni, mesiace a rok tak aby vyhovovali správne dátumu. Ak dátum neprešiel cez regulárny výraz alebo bol zadaný nesprávny dátum aplikácia zase upozorní užívateľa vyskakovacím oknom.



# Kapitola 8

## Testy

### 8.1 Porovnanie s ostatnými aplikáciami

Táto kapitola bude obsahovať otestovanie aplikácie na reálnych dátach. Prvé spustenie skriptu na získavanie dát trvalo celkovo okolo 50 hodín. Celkovo sa do databáze uložilo 49 tisíc záznamov do troch tabuliek. Z toho zhruba 19 tisíc predstavujú ponuky. Z niektorých ponúk sa však nepodarilo extrahovať dáta pomocou regulárnych výrazov. Nie všetky inzeráty sú totižto v jednotnej podobe. Skript ktorý vytvára súbor s dátami pre aplikáciu najskôr kontroluje prezenciu všetkých potrebných údajov. V prípade že niektorý chýba do výsledného súboru nie je zapísaný. Obrázky v prílohách porovnávajú cenové rozloženie ponúk s výslednou aplikáciou [A.1](#) a s portálom *realitymon.cz* [A.2](#). Ako je možné vidieť výsledná aplikácia obsahuje aj ponuky zo slovenska čo je spôsobené tým že portál *topreality.cz* nie vždy ponúka reality len z Českej republiky. Portál *realitymon.cz* neponúka zobrazenie teplotnej mapy. Preto pre porovnanie posluží portál *cenovamapa.eu* Tento portál poskytuje informácie o tom koľko ponúk zobrazuje. Preto sa bude porovnávať iba približné teplotné rozloženie. Ako je vidieť na obrázkoch [A.4](#) a [A.3](#) toto teplotné rozloženie sa zhruba zhoduje. V oblasti Prahy sa nachádzajú najdrahšie nehnuteľnosti, ďalej je to okolie Brna, Ostravy a Karlových Varov. Výsledná aplikácia poskytuje aj možnosť sledovania vývinu cien. Táto funkcia je však použiteľná iba po dlhodobom nasadení na serveri. Kedy sa inzeráty obmieňajú tak že je možné reálne sledovať vývin cien. Obrázok [A.5](#) ukazuje časovú os.

### 8.2 Blokovanie automatického skriptu

Počas sťahovania dát z portálu *sreality.cz* nastal problém s automatickým skriptom, proti ktorým je tento portál zabezpečený. Od 60. stránky s nehnuteľnosťami znemožňuje zobraziť obsah stránky pre automatické skripty, roboty a pod. Nanešťastie ani zmena proxy serveru pomocou programu PhantomJS nepomohla. Po nahliadnutí do súboru `robots.txt` bolo zrejmé prečo sa obsah stránky nezobrazoval. Súbor obsahuje pravidlo `Disallow: /hledani/*`, znemožňuje tak robotom zobrazovať obsahy stránok. Kvôli tejto strate asi 17 tisíc ponúk výsledná aplikácia zaostáva za existujúcim riešením *realitymon.cz*. Ak by k tejto strate nedošlo, počet ponúk by bol približne rovnaký. Aktualizovanie databáze, čiže spustenie hlavného skriptu druhý krát, trvalo niekoľko hodín. Aktualizovanie databáze raz za deň je dosiahnuteľné.

### 8.3 Užívateľské testovanie

Orezaná verzia ktorá sa neaktualizuje každý deň ale obsahuje iba statické dáta bola nasaďená na školský server Eva<sup>1</sup>. Celkovo 30 respondentov otestovalo túto verziu aplikácie. Bolo im povedané aby si aplikáciu vyskúšali a následne vyplnili dotazník. Medzi zaujímavé časti z výsledkov dotazníka patrili tieto: spokojnosť s grafickou stránkou aplikácie, porovnanie grafického rozhrania s aplikáciou na *www.realitymon.cz*, porovnanie teplotnej mapy s mapou na *www.cenovamapa.cz*. Viac ako polovici (55,2%) zúčastnených vyhovovala grafická stránka aplikácie. Pri porovnaní s grafickým rozhraním aplikácie *www.realitymon.cz* by však 69% uprednostnilo aplikáciu na *www.realitymon.cz*. Ďalším zaujímavým výsledkom bolo porovnanie teplotnej mapy výslednej aplikácie a aplikácie na *www.cenovamapa.cz*, kde by 65,9% uprednostnilo výslednú aplikáciu. Okrem tohoto dostali respondenti aj voľný priestor pre vyjadrenie svojich myšlienok o aplikácii. Celkovo sa tak získalo 14 zaujímavých pripomienok ako filtrovanie ponúk podľa kraja, zväčšenie legendy či pridanie nápovedy.

---

<sup>1</sup>Dostupné na: <https://www.stud.fit.vutbr.cz/~xhalin01/>

## Kapitola 9

# Záver

Cieľom tejto bakalárskej práce bolo implementovať aplikáciu cenovej mapy. Aplikáciu sa podarilo implementovať s reálnymi dátami získanými z portálov *sreality.cz*, *topreality.cz* a *reality.idnes.cz*. Dokáže zobrazovať ponuky na mape a filtrovať ich podľa požiadavok užívateľa. Ďalšou funkciou tejto aplikácie je aj zobrazenie teplotnej mapy. Aplikácia je jedinečná vzhľadom na možnosť zobrazenia ponúk na časovej osi. Verzia s ukázkovými dátami je dostupná na školskom serveri Eva. Na trhu neexistuje aplikácia ktorá by toto dokázala a to ju robí unikátnou. Jednoduchou konfiguráciou skriptu je možné získavať informácie o nehnuteľnostiach aj zo Slovenskej Republiky a to vďaka portálu *topreality.cz* ktorý disponuje databázou zahrňujúcou aj nehnuteľnosti z iných krajín. Možným rozšírením tejto aplikácie by bolo zobrazovanie ponúk domov, pozemkov či iných objektov. Vzhľadom k náročnosti výpočtov by však server na ktorom by aplikácia bežala musel byť dostatočne výkonný aby stíhal aktualizovať všetky ponuky každý deň. Tieto výpočty by mohli byť urýchlené použitím iných prostriedkov na získavanie dát. Regulárne výrazy ktoré boli použité, nie sú tou najoptimálnejšou voľbou. Dáta ktoré boli získané touto aplikáciou poslúžili aj pre bakalársku prácu na oceňovanie nehnuteľností<sup>1</sup>.

---

<sup>1</sup>Dostupné na: <https://wis.fit.vutbr.cz/FIT/st/bci-prj-z.php?id=19192&status=&pptype=&vedid=7215&typ=BP1>

# Literatúra

- [1] Apache ver. 2.4.17. [Online; navštívené 14. 05. 2017].  
URL <https://httpd.apache.org/>
- [2] Beal, V.: API - application program interface. [Online; navštívené 28. 12. 2010].  
URL <http://www.webopedia.com/TERM/A/API.html>
- [3] Bohuslav Křena, R. K.: Úvod do softwarového inženýrství IUS. 2010, [Online; navštívené 14. 05. 2017].  
URL [https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IUS-IT/texts/IUS\\_opora.pdf?cid=8697](https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IUS-IT/texts/IUS_opora.pdf?cid=8697)
- [4] Cockburn, A.: *Writing Effective Use Cases*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., první vydání, 2000, ISBN 0201702258.
- [5] Daconta, M. C.; Smith, K. T.; Obrst, L. J.: *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. New York, NY, USA: John Wiley & Sons, Inc., první vydání, 2003, ISBN 0471432571, 9780471432579.
- [6] Duckett, J.: *Accessible XHTML and CSS Web sites problem, design, solution*. Foster City, United States: Wiley, 2005, ISBN 9780764583063.
- [7] Čermák, M.: Vícevrstvá architektura: tenký, tlustý a chytrý klient. 2010, [Online; navštívené 14. 05. 2017].  
URL <http://www.cleverandsmart.cz/vicestrstva-architektura-tenky-tlusty-a-chytry-klient/>
- [8] Fowler, M.: *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., třetí vydání, 2003, ISBN 0321193687.
- [9] PHP ver. 5.6.19. [Online; navštívené 14. 05. 2017].  
URL <http://php.net/manual/en/>
- [10] Google geocoding, usage limits. [Online; navštívené 14. 05. 2017].  
URL <https://developers.google.com/maps/documentation/javascript/geocoding>
- [11] Google maps API ver. 3. [Online; navštívené 14. 05. 2017].  
URL <https://developers.google.com/maps/documentation/javascript/>
- [12] Google maps API, usage limits. [Online; navštívené 14. 05. 2017].  
URL <https://developers.google.com/maps/documentation/javascript/usage>

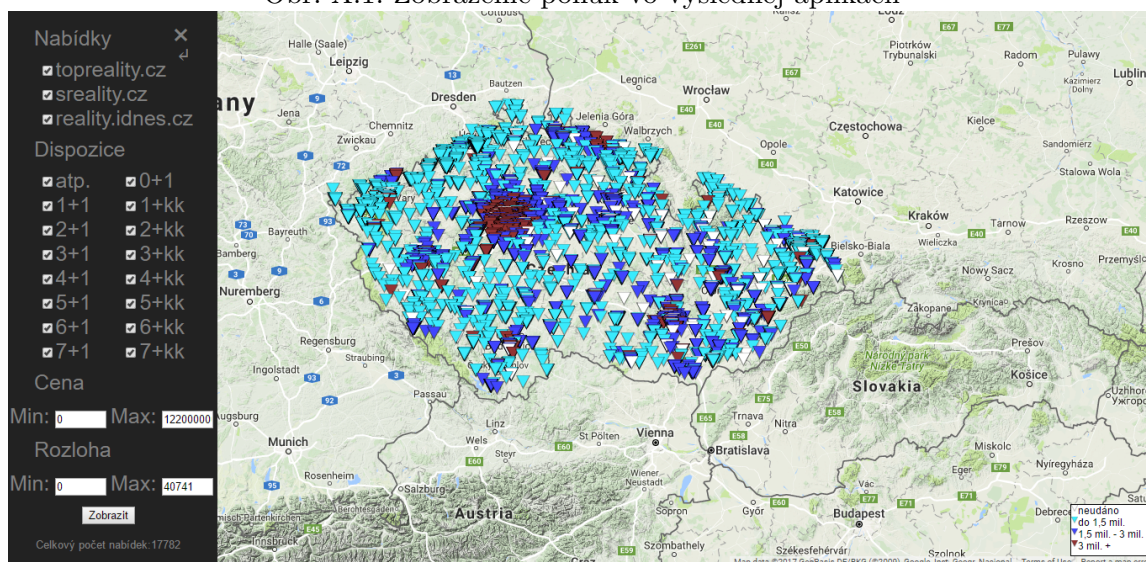
- [13] Jaroslav Zendulka, I. R.: Databázové systémy IDS. 2006, [Online; navštívené 14. 05. 2017].  
URL [https://wis.fit.vutbr.cz/FIT/st/course-files-st.php.cs?file=%2Fcourse%2FIDS-IT%2Ftexts%2FIDS\\_predn.pdf&cid=10302](https://wis.fit.vutbr.cz/FIT/st/course-files-st.php.cs?file=%2Fcourse%2FIDS-IT%2Ftexts%2FIDS_predn.pdf&cid=10302)
- [14] Javascript. [Online; navštívené 14. 05. 2017].  
URL <https://www.javascript.com/>
- [15] MariaDB ver. 10.1.10. [Online; navštívené 14. 05. 2017].  
URL <https://mariadb.org/>
- [16] PhantomJS 2.2.1. [Online; navštívené 14. 05. 2017].  
URL <http://phantomjs.org/>
- [17] What are the pros and cons of using PHP? [Online; navštívené 14. 05. 2017].  
URL <https://medium.com/@smartgamma/what-are-the-pros-and-cons-of-using-php-490553ed8ff2>
- [18] PhpMyAdmin 4.5.1. [Online; navštívené 14. 05. 2017].  
URL <https://www.phpmyadmin.net/>
- [19] Raspberry PI 3 Model B. [Online; navštívené 14. 05. 2017].  
URL <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [20] What Is a Regular Expression, Regexp, or Regex? 2016, [Online; navštívené 14. 05. 2017].  
URL <https://www.regexpbuddy.com/regex.html>
- [21] Learn about robots.txt files. 2017, [Online; navštívené 14. 05. 2017].  
URL <https://support.google.com/webmasters/answer/6062608?hl=en>
- [22] Scrapehero: How to prevent getting blacklisted while scraping. 2014, [Online; navštívené 14. 05. 2017].  
URL <https://www.scrapehero.com/how-to-prevent-getting-blacklisted-while-scraping/>
- [23] Windows Task Scheduler ver. 1.0. [Online; navštívené 14. 05. 2017].  
URL [https://technet.microsoft.com/en-us/library/cc721871\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc721871(v=ws.11).aspx)
- [24] XAMPP ver. 3.2.2. [Online; navštívené 14. 05. 2017].  
URL <https://www.apachefriends.org/index.html>

# Prílohy

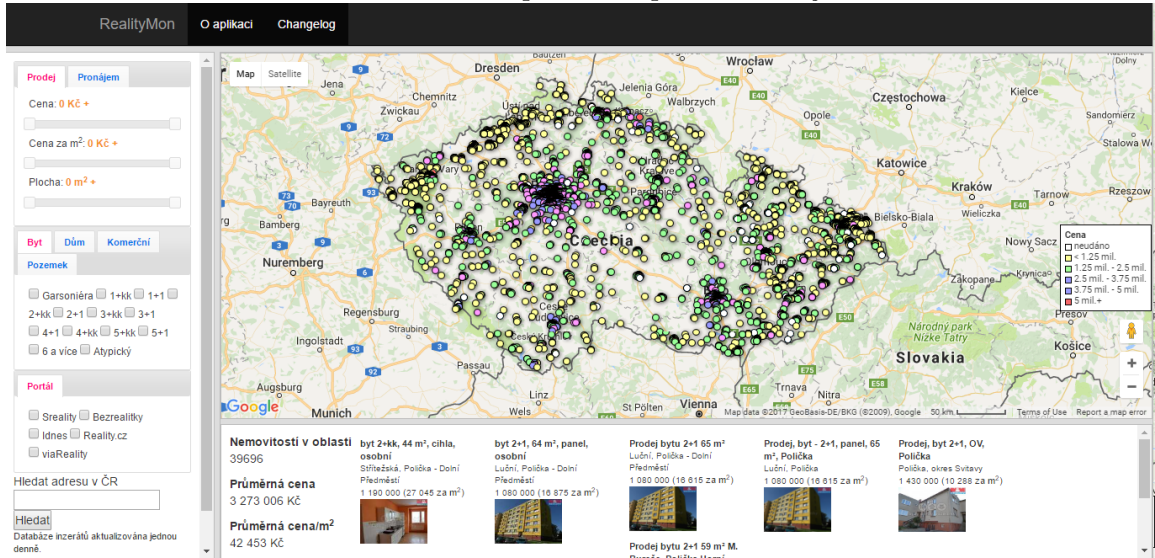
# Príloha A

## Ukážky z aplikácií

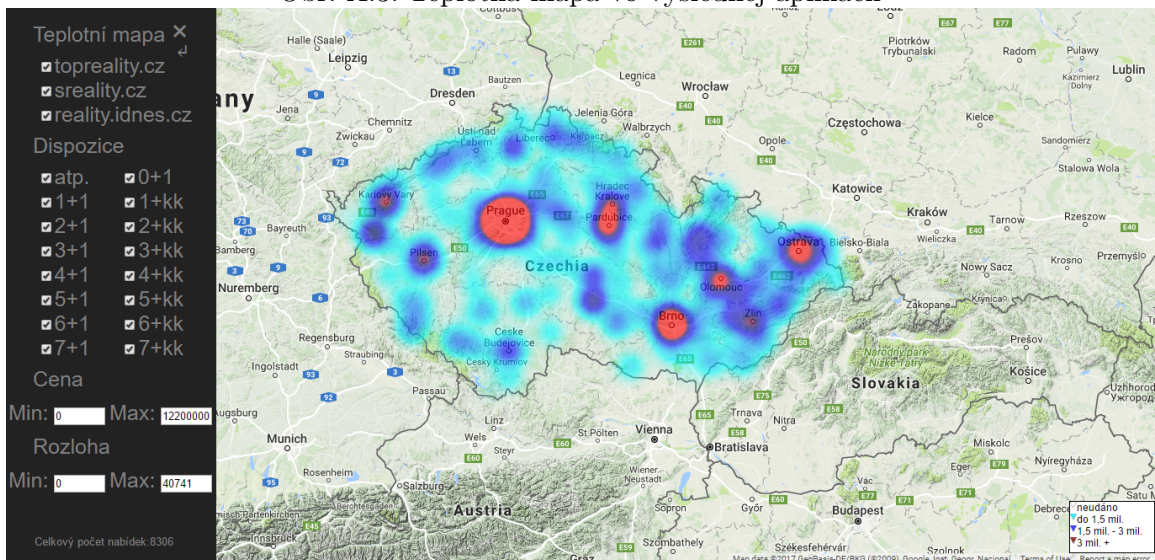
Obr. A.1: Zobrazenie ponúk vo výslednej aplikácii



Obr. A.2: Zobrazení ponúk na portáli *realitymon.cz*

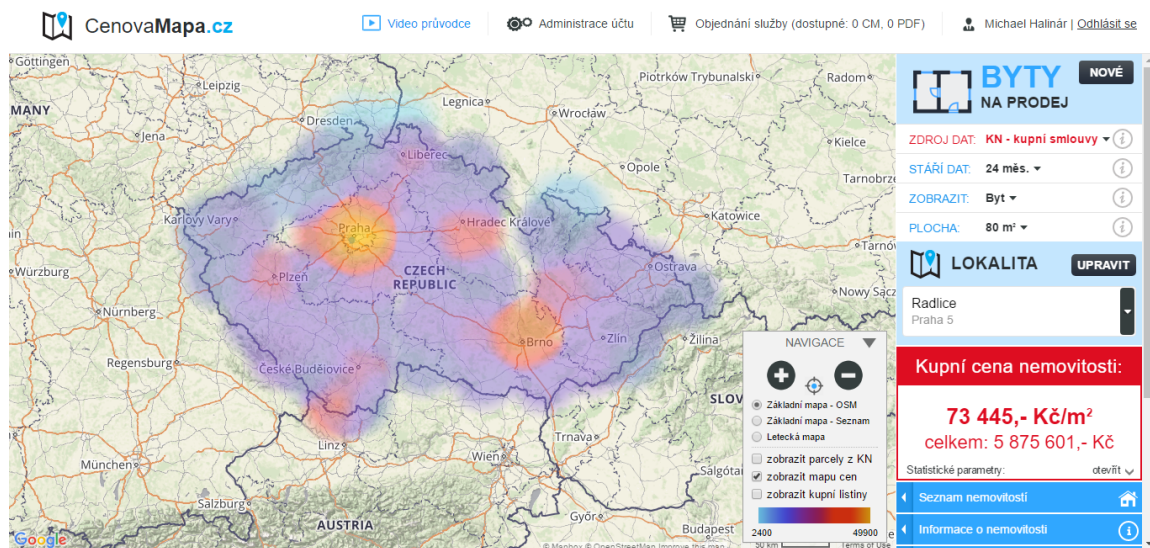


Obr. A.3: Teplotná mapa vo výslednej aplikácii

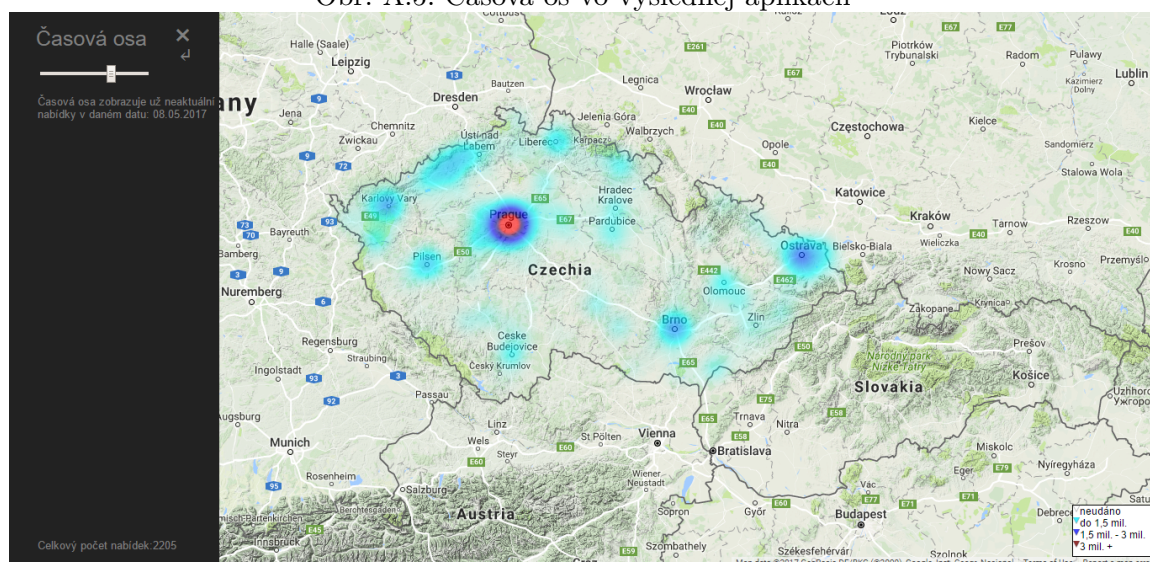




Obr. A.4: Teplotná mapa na portáli www.cenovamapa.cz



Obr. A.5: Časová osa vo výslednej aplikácii



## Príloha B

# Obsah priloženého pamäťového média

Pamäťové médium obsahuje všetky potrebné skripty pre aplikáciu vrátane pokynov, databáze so záznamami ponúk a plagát prezentujúci aplikáciu.