VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií
Faculty of Information Technology

BAKALÁŘSKÁ PRÁCE
BACHELOR‘S THESIS

Brno, 2017 Tadeáš Kovář

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ZÍSKÁVÁNÍ INFORMACÍ O CENÁCH VOZIDEL PRODÁVANÝCH NAPŘÍČ ČR

OBTAINING PRICES OF VEHICLES SOLD IN CZECH REPUBLIC

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TADEÁŠ KOVÁŘ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PAVEL OČENÁŠEK , PH.D.

BRNO 2016/2017

Zadání bakalářské práce/19968/2016/xkovar69

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Kovář Tadeáš**
Obor: Informační technologie
Téma: **Získávání informací o cenách vozidel prodávaných napříč ČR**
Obtaining Prices of Vehicles Sold in Czech Republic
Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se s inzertními weby autobazarů. Vyberte vhodné servery pro efektivní získávání dat. Seznamte se s vhodnými webovými technologiemi pro toto téma.
2. Analyzujte požadavky na aplikaci, která bude provádět získávání dat o vozidlech.
3. Vytvořte návrh této aplikace dle instrukcí vedoucího práce.
4. Návrh aplikace implementujte.
5. Otestujte získávání dat nad jednotlivými servery a následně funkčnost celé aplikace včetně uživatelského rozhraní. Zhodnoťte možnosti dalšího rozšíření.

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

Hlavním cílem této bakalářské práce je vytvořit aplikaci pro získávání informací o vozidlech z různých autobazarů napříč ČR. Informace jsou získávány vyhledáváním klíčových slov z HTML a prostého textu. Aplikace obsahuje několik skriptů pro stahování informací a jednoduché uživatelské rozhrání. Nalezena data jsou ukládána do databáze a bakalářská práce Nikolase Kantora s nimi pracuje. Aplikace prozatím funguje pouze na lokálním zařízení. Celý projekt využívá tyto technologie: HTML, CSS, PHP a MySQL.

Abstract

The main aim of this bachelor thesis is to create an application for obtaining information about vehicles from different car dealers across the Czech Republic. Information is obtained by searching for keywords from HTML and plain text. The application contains several scripts for downloading information and simple user experience. Data found are stored in the database and Nikolais Kantor's bachelor thesis works with them. The app is currently running only working on your localhost. The entire project uses the following technologies: HTML, CSS, PHP and MySQL.

Klíčová slova

Získání dat, autobazary, ceny vozidel, webová aplikace, PHP, HTML, CSS, vyhledávání informací z textu, databáze

Keywords

Obtaining data, autobazar, price of vehicles, web application, PHP, HTML, CSS, find information from text, database

Citace

KOVÁŘ, Tadeáš. *Získávání informací o cenách vozidel prodávaných napříč ČR*. Brno, 2017. 34 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Očenášek Pavel , Ph.D.

Získávání informací o cenách vozidel prodávaných napříč ČR

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavla Očenáška , Ph.D.
Další informace mi poskytl Ing. Vladimír Bartík, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Tadeáš Kovář
17. května 2017

Poděkování

Chtěl bych poděkovat vedoucímu práce panu Ing. Pavlu Očenáškovi, Ph.D. za odborné konzultace a poskytnutí užitečných rad při implementaci. Dále bych chtěl poděkovat panu Ing. Vladimíru Bartíkovi, Ph.D. za odbornou konzultaci při implementaci vyhledávání informací z textu.

Obsah

1 Úvod

Tato bakalářská práce popisuje tvorbu aplikace pro stahování dat z autobazarů. Cílem je vytvořit aplikaci, která získá informace o osobních automobilech z jednotlivých inzertních serverů autobazarů. Tyto informace jsou ukládány do databáze. Na uložená data navazuje bakalářská práce Nikolase Kantora, která pracuje s těmito daty. Hlavním cílem těchto dvou bakalářských prací je vytvořit aplikaci, která umí oceňovat vozidla, porovnávat ceny vozidel na jednotlivých serverech a vyhledávat určitá vozidla napříč autobazary.

V České republice nejspíš žádná aplikace, která by uměla zpracovávat, oceňovat a vypisovat vozidla napříč autobazary není, tudíž by tato aplikace mohla být novinkou na internetu. Aplikace však prozatím nebude přístupná na internetu. Případné zpřístupnění celé aplikace na internetu se pouze uvažuje do budoucna.

V první části tohoto dokumentu se nejdříve seznámíme s technologiemi, které byly pro práci využity. V dalším kroku bude rozepsán seznam detailnějších požadavků práce na který navazuje analýza těchto požadavků. V analýze se dozvíme, jaké servery byly pro zisk dat vybrány, jaké informace o vozidle budou pro nás důležité a zjistíme, jaké bude mít aplikace chování z pohledu uživatele. Následná kapitola pojednává o návrhu aplikace, ve které jsou zejména umístěny návrhové modely. Předposlední dvě kapitoly se zabývají samotnou implementací a následným testováním. Implementace obsahuje popis vytvoření nejdůležitějších částí aplikace. U testování je uvedeno nejen jakým způsobem probíhalo během implementace, ale i konečné testování projektu. Na konci dokumentu nalezneme závěrečné shrnutí práce a možné návrhy vylepšení systému do budoucna.

2 Webové technologie

Zadavatel práce nám nijak nespecifikoval technologie, které bychom museli pro tvorbu použít. Samotné získávání dat je tedy tvořeno pomocí PHP skriptu. Získaná data jsou potřeba ukládat do databáze, využijeme tedy i MySQL. Uživatelské prostředí bude vytvořeno jakožto standardní webová aplikace pomocí HTML, CSS a PHP.

2.1 HTML (HyperText Markup Language)

HTML je značkovací jazyk využívaný pro tvorbu webových stránek. Tyto stránky bývají propojeny pomocí hypertextových odkazů. Jazyk obsahuje množinu značek (tagů). Značky bývají většinou párové a mezi ně se píše text, který díky značkám bude v určitém formátu. Názvy značek se píší do úhlových závorek např. Odstavec a ukončující značka obsahuje lomítko. Aktuální verze HTML je verze 5.0, která byla vydána po velmi dlouhé době. [1]

2.2 CSS (Cascading Style Sheet)

CSS je jazyk, který se používá pro definici vzhledu webových stránek napsaných pomoci jazyka HTML. Je to deklarativní jazyk, který je definován pravidly. Pomocí selektorů se určí množina HTML prvků, které budou mít určeny nějaké vlastnosti, pomocí kterých je tvořen vzhled stránky. Aktuální verze je CSS 3, která rozšiřuje verzi CSS 2 a pojí se s HTML 5.0. [2]

2.3 PHP (Hypertext Preprocessor)

PHP je široce používaný skriptovací jazyk, který je vhodný zejména pro tvorbu webových aplikací a může být vložen přímo do HTML kódu. PHP kód je spuštěn na serveru, tam se vygeneruje HTML kód, který je následně odeslán ke klientovi. Klient tedy získá pouze stránky s výsledným HTML kódem. Syntaxe jazyka je vypůjčena z jazyků C, Java a Perl. Je tedy snadné se syntaxi naučit.

Pro místní využití je potřeba nainstalovat PHP webový server např. Apache. Pro využití databáze se k PHP většinou instaluje MySQL. [3]

2.4 MySQL

MySQL patří k světově nejpopulárnějším databázovým serverům. Je velmi dobře využitelná pro webové aplikace. Kromě toho, že je to open source databáze, tak je také velmi výkonná, spolehlivá a velmi snadno použitelná. MySQL je založena na dotazovacím jazyku SQL. [4]

3 Seznam požadavků

Požadavky na výsledný program můžeme rozdělit do těchto bodů:

- Vybrat dva vhodné servery autobazarů pro zisk dat z HTML kódu.
- Vybrat jeden server autobazaru pro zisk dat z prostého textu.
- Získat důležité informace z těchto serverů.
- Uložit tyto informace do databáze.
- Rozpoznat konkrétní automobil.
- Vytvořit jednoduché uživatelské rozhraní.

Celý seznam požadavků byl vytvořen na základě konzultace se zadavatelem (vedoucím práce) a výsledný program by měl tyto požadavky splňovat.

Vybráním vhodných serverů pro zisk dat z HTML kódu je myšleno vybrat takové servery, ve kterých jsou data uložena v určitých HTML značkách (tabulky, seznamy, třídy atd.).

Vybráním serveru pro zisk dat z prostého textu se rozumí vybrat takový server, na kterém jsou detaily inzerátu psány prostým textem (ve větách) například: Prodám zachovalou Škodu Felicii vyrobenou v roce 1990 atd.

Dalším požadavkem je uložit získané data do databáze. Struktura databáze je potřeba konzultovat s Nikolasem Kantorem, protože jeho práce bude z této databáze čerpat informace.

Rozpoznáním konkrétního automobilu se myslí situace, kdy je jeden automobil inzerován na více serverech a je tedy nutné rozpoznat, že se jedná pouze o jedno a to samé vozidlo.

Posledním požadavkem je vytvořit uživatelské rozhraní, ze kterého je snadno nastavitelné chovaní skriptu. Tohle nastavení je přístupné pouze pro administrátora.

4 Analýza

Tato kapitola je věnována zanalyzování vstupních požadavků práce. Můžete si zde přečíst detailnější informace o jednotlivých serverech, ze kterých se čerpají data, které parametry jsou stahovány a ve spodní části kapitoly také informace o uživatelském rozhraní.

4.1 Inzertní weby autobazarů

Na internetu existuje spousta webových serverů, které se zabývají prodejem automobilů. Téměř každý web autobazarů poskytuje základní informace o automobilu jako je například: značka, model, rok výroby, počet najetých kilometrů, cena, druh paliva, objem motoru, výkon motoru, druh převodovky, karosérie a barva. Tyto informace jsou většinou jednoduše vyhledatelné v samotném kódu webové stránky díky přiděleným třídám. Dále téměř všechny webové servery obsahují informace o počtu míst a dveří automobilu avšak tyto informace jsou na různých webech v odlišných sekcích. Nejvíce se weby autobazarů liší podrobným výpisem vybavení určitého automobilu. Tento výpis má téměř každý web vytvořený odlišně. Některé servery mají tyto informace uloženy formou tabulky, jiné zase prostým textem či využitím odrážek. Podrobný výpis výbavy obsahuje také odlišné informace u jednotlivých automobilů. U některých automobilů jsou do výbavy zahrnuty i banální věci jako je například otáčkoměr, který u jiných automobilu ve výbavě není zadán, ale automobil je otáčkoměrem vybaven. Tento fakt ovšem nebude pro výsledný projekt žádným problémem, protože informace o výbavě pro nás nejsou klíčové, slouží pouze jako doplňkové informace k vozidlům. Větším problémem je, že je jen malé množství autobazarů, které obsahují u vozidel jejich VIN[1] kód, díky kterému bychom mohli jednoznačně určit duplicitní vozidla.

Ve větších autobazarech je možné automobily vyhledávat pomoci filtrů nebo si vypsat všechny dostupné automobily. V menších autobazarech většinou žádné vyhledávací filtry nejsou, tudíž musíme procházet i značky, které nás nezajímají.

4.2 Vybrání vhodných webů pro zisk dat

Z nalezených inzertních webových serverů byly vybrány 3 servery:

- www.cars.cz[2],
- www.sbazar.cz[3],
- www.hyperinzerce.cz[4].

[1] Jednoznačné identifikační číslo vozidla
[2] Dostupný na adrese http://www.cars.cz/
[3] Dostupný na adrese https://www.sbazar.cz/170-osobni-auta
[4] Dostupný na adrese http://autobazar.hyperinzerce.cz/inzerce-auta-znacky/inzerce/nabidka/

Při výběru vhodných webových server byl kladen důraz nejen na dostupné informace o vozidlech, ale také na co největší množství inzerovaných vozidel. Vybrané servery disponují jedním s největším počtem inzerovaných vozidel v České republice.

4.2.1 Sbazar.cz

Server www.sbazar.cz se zabývá prodejem různých věcí jako je například: elektro, oblečení, obuv, knihy atd. Kromě osobních vozidel, které nás zajímají inzeruje také náhradní díly, bourané vozidla, motocykly a užitkové vozy. Server nabízí přibližně 34000[5] osobních vozidel. Obsahuje filtr vyhledávání pouze pomocí značky vozidla nebo výpis všech vozidel, kdy se na stránce zobrazí přibližně 34 inzerátů. I když má tento server vlastní odvětví pro prodej náhradních dílů atd., tak i přesto přibližně 10% inzerátů obsahuje prodej právě náhradních dílů, disků či pneumatik nebo vozidla na náhradní díly.

Při zobrazení konkrétního automobilu (část konkrétního inzerátu na obrázku Obrázek 4.1) můžeme na první pohled vidět cenu vozidla, fotografie, prodejce a informace o vozidle. Všechny informace o vozidle jsou vypsány formou prostého textu. Získání důležitých informací o vozidle bude muset tedy probíhat vyhledáváním v textu. Vyhledávání informací z textu bylo konzultováno s Ing. Vladimírem Bartíkem, Ph.D. Výsledkem konzultace bylo, že k vyhledávání využijeme slova, která by se měla nacházet v okolí vyhledávaných dat například: údaj o počtu najetých kilometrů by měl obsahovat 4 číslice a za nimi jednotku (km), výkon by měl obsahovat 2 nebo 3 číslice a za nimi jednotku (kw). Při takovém způsobu zisku dat není možno zaručit stoprocentní pravdivost získaných informací, protože informace mohou být zapsány v různých formátech nebo může být uvedeno více informací ve stejném formátu a tím pádem se může stát, že například místo celkového počtu najetých kilometrů získá aplikace informaci o tom, kolik vozidlo najelo kilometrů po výměně spojky. Zbylé informace, které nelze nalézt vyhledáváním klíčových slov v okolí potřebné informace, jsou vyhledávány na základě již existujících záznamů v databázi. Jedná se o informace značky, modelu a barvy vozidla.

Protože se některé informace vyhledávají na základě již existujících dat v databázi, stahování dat z tohoto serveru bude funkční pouze po předchozím stažení dat ze serverů cars.cz a hyperinzerce.cz nebo pokud databáze obsahuje alespoň 15000 záznamů o vozidle. Číslo 15000 záznamů bylo zvoleno náhodně a mělo by bát dostačující pro efektivní vyhledávání v textu.

Jelikož se z inzerátu nemusí vždy povést zjistit všechny informace, tak systém obsahuje kontrolu, zda se povedly nalézt alespoň čtyři informace (do informací se nepočítá značka a model vozu, protože ty musíme znát vždy), v případě nenalezení alespoň čtyř informací o vozidle se inzerát ignoruje a data nebudou ukládány do databáze, protože vozidlo, o kterém neznáme téměř žádné informace by v databázi spíše překáželo.

Ve většině inzerátech na tomto serveru není uveden VIN kód, proto je zbytečné, aby se ho systém snažil vyhledat.

[5] K datu 22.4.2017

Obrázek 4.1: Část inzerátu z www.sbazar.cz

4.2.2 Cars.cz

Server www.cars.cz obsahuje přibližně 32000[6] inzerátů. V těchto inzerátech jsou také zahrnuty náhradní díly jako jsou světlomety atd. a obsahují také prodej disků či pneumatik. V přibližně 5% inzerátů se jedná právě o tyto díly. U většiny vozidel je VIN kód dostupný, jen v některých případech je VIN kód neúplný, tudíž skrytý. Inzeráty lze zobrazit buďto všechny nebo pomocí filtru jen některé.

Při zobrazení konkrétního inzerátu vidíme v nadpisu inzerátu značku, model a většinou i objem motoru a bližší specifikaci modelu. Základní informace o vozidle jsou zobrazeny v přehledné tabulce. Tyto informace obsahují: rok výroby, počet najetých kilometrů, palivo, objem motoru, spotřebu (většinou nebývá uvedena), výkon, počet dveří a sedadel, barvu, karosérii, VIN kód a cenu v korunách i eurech. V ojedinělých případech není některá ze základních informací uvedena. Informace o rozšiřující výbavě jsou vypsány ve sloupcích. Rozšiřující výbava obsahuje například informaci o druhu převodovky, počet rychlostních stupňů, vyhřívaná zrcátka, mlhovky atd. U tohoto serveru jsou doplňující informace celkem zavádějící, protože například otáčkoměr má téměř každý automobil, avšak ve výbavě se zobrazuje jen zřídka. Také informace o tom, že vozidlo má pevnou střechu se zdá být celkem zbytečná. Tyto zavádějící informace obsahují zejména starší a levnější automobily jako je například Škoda Felicia. Pod nadpisem stav vozidla je prostým textem většinou vypsán původ vozidla. Další doplňující informace jsou vypsány formou prostého textu a obsahují například: upřesnění výbavy, možnosti platby atd., tyto informace v práci budou zanedbány, protože jsou pro nás nedůležité. Část konkrétního inzerátu je možné vidět na obrázku Obrázek 4.2.

Některé autobazary, které inzerují automobily právě na tomto serveru mají vytvořenou vlastní strukturu inzerátu, což je pro náš zisk dat problém, protože by muselo být stahování dat přizpůsobeno

[6] K datu 24.4.2017

právě těmto strukturám. Těchto inzerátů, které mají vlastní strukturu je však na serveru www.cars.cz jen nepatrné množství a tím pádem mohou být v práci zanedbány.

Škoda Superb Combi 2.0 TDi 110 kW Style — **799 000,- Kč**
2016, Diesel, 1 968 ccm, 110 kW — (cena při platbě v hotovosti)

V provozu od:	06.2016
Tachometr:	14 688 km
Palivo:	Diesel
Objem:	1 968 ccm
Výkon:	110 kW
Dveří / sedadel:	5 / 5
STK:	06.2020
Karoserie:	Kombi
Barva:	bílá metalíza
VIN: TMBJH9NP2G7020307	Prověřit vůz

Cena s DPH: 799 000,- Kč
Cena bez DPH: 660 331,- Kč
orientačně v EUR: 30 186,- €
Možnost odpočtu DPH

Upozornit na chyby v inzerátu

VÝBAVA VOZIDLA (podle skupin / bez skupin)

ABS	AirBag 10x	Alu kola
Asistent-parkovací	Centrální zam. dálkově	Elektrická okna
Imobilizér	Klimatizace dig. čtyřokruhová	Kontrola tlaku v pneu
Kůže	Multifunkční volant	Nastavitelný volant
Ostřikovač světel	Palubní počítač	Protiskluzový systém ASR
Převodovka manuální	Rychlostní stupně 6x	Rádio přijímač/CD
Satelitní navigace GPS	Sedadla el. nastavitelná	Sedadla nastavitelná výškově
Sedadla vyhřívaná	Sedadla zadní - dělená	Senzor deště
Senzor světel	Servo	Splňuje normu euro IV
Stabilizace podvozku ESP	Střecha pevná	Střešní nosič
Světlomety přídavné mlhovky	Tažné zařízení	Tempomat
Venkovní teploměr	Zrcátka elektricky nastavitelná	Zrcátka vyhřívaná

Obrázek 4.2: Část inzerátu u www.cars.cz

4.2.3 Hyperinzerce.cz

Server www.hyperinzerce.cz se zabývá inzerováním různých věcí jako jsou například: seznamka, bydlení, zvířata, sběratelství atd. Nás ovšem zajímá inzerování osobních vozidel. Hyperinzerce nabízí přibližně 41000[7] inzerátů osobních vozidel, což je pro náš zisk dat výhoda. Server nabízí zobrazení inzerátů pomocí filtru, kde je možno si nechat zobrazit vozidla jen určité značky, modelu atd. V našem případě však je využito zobrazení všech nabízených vozidel. Přibližně 40 vozidel je zobrazeno jako „TOP“ a tím pádem jsou tyto inzeráty zobrazeny jako první.

Při zobrazení konkrétního vozidla můžeme na první pohled vidět značku a model vozidla v nadpisu. Základní informace o vozidle jsou zobrazeny v přehledné tabulce. Tato tabulka mimo jiné obsahuje i znovu uvedenou značku a model vozidla. Následuje popis vozidla formou prostého textu,

[7] K datu 25.4.2017

který blíže upřesňuje vlastnosti vozidla, stav vozidla atd. Tuto část inzerátu v práci zanedbáváme, protože pro nás nejsou tyto údaje podstatné. Vedle popisu se nachází seznam výbavy vozidla, avšak tento seznam není uveden u každého inzerátu. Část konkrétního inzerátu je možno vidět na obrázku Obrázek 4.3.

Určité množství inzerátu, které zde vkládají některé autobazary, mají vlastní strukturu. V tomto případě je inzerátů s vlastní strukturou na serveru velké množství a tím pádem není přípustné tento fakt ignorovat. Proto se v práci zabýváme zjištěním, o jaký autobazar se jedná a získáváme data i z těchto upravených struktur. Jedná se zejména o větší autobazary jako je například: AAA Auto nebo AutoESA.

Značka vozu	Ford
Modelová řada	Ford Mondeo
Druh karoserie	Combi
Palivo	nafta
Převodovka	manuální převodovka
Barva vozu	modrá
Stav vozu	havarované auto
Rok výroby	2015
Stav tachometru	83804 km
Obsah motoru	1997 ccm
Výkon motoru	150 koní (cca 110 kW)
VIN kód	WF0FXXWPCFFL22990

Cena: **197 000 Kč**
Povinné ručení: spočítat povinné ručení

Obrázek 4.3: Část inzerátu u www.hyperinzerce.cz

4.3 Výběr podstatných parametrů

Výběr informací, které je skript schopný získat jsou zanalyzovány z konkrétních serverů, ze kterých se data stahují. Po konzultaci s Nikolasem Kantorem, který tyto informace využívá jsme usoudili, že čím víc informaci o vozidle bude, tím lépe. Výsledná aplikace tedy získává téměř všechny informace, které z daného inzerátu lze získat. Mimo jiné je nutné získat také odkaz na daný inzerát. Aplikace neumí získat dodatečné informace o vozidle jako je například: vozidlo na náhradní díly, má nějakou poruchu, vozidlo vlastnil pouze jeden majitel atd. Pokud některé informace nejsou možné z inzerátu získat, tak se pouze neuloží tyto informace do databáze. Skript získává všechny informace o rozšiřující výbavě vozidla, to znamená, že získává i banální výbavu jako je například: otáčkoměr, pevná střecha atd.

4.4 Rozpoznání konkrétního automobilu

Jeden automobil může být uložen na více inzertních webech, proto je nutné umět tuto situaci nějakým způsobem rozpoznat.

Většina inzerátů vozidel neobsahuje VIN kód nebo je tento kód skrytý, proto není možné jednoznačně určit identická vozidla. Je tedy potřeba zvolit jiné řešení, které by mohlo zjistit identitu vozidel na základě údajů již uložených v databázi a aktuálně stažených údajů z inzerátu. Jako řešení se nabízí porovnávat tyto informace:

- Značka vozidla,
- model vozidla,
- rok výroby,
- počet najetých kilometrů,
- objem motoru,
- výkon motoru,
- druh paliva,
- barva vozidla,
- druh karosérie.

Tato metoda bohužel nemůže zaručit stoprocentní shodu vozidel, ale je velmi pravděpodobné, že při shodě všech těchto parametrů půjde o identický vůz. Některé inzeráty neobsahují všechny tyto základní parametry vozidla, čímž klesá procento úspěšnosti nalezení identického vozidla. V případě nalezení identického vozu se vytvoří instance tohoto vozu a tím budeme vědět, že jedno určité vozidlo obsahuje několik instancí.

4.5 Ukládání dat

Veškerá získaná data je potřeba ukládat do databáze. Databáze by měla být zprovozněna na nějakém webovém serveru, protože tato aplikace musí data do ní vkládat a jiná aplikace bude tyto data z databáze čerpat. Za tímto účelem je databáze tvořena v předem domluveném formátu, tak aby vyhovovala oběma aplikacím. Při ukládání dat do databáze probíhá synchronizace dat v databázi. Tímto se urychlí celý chod skriptu, protože instance vozidla, která již v databázi existuje se nevkládá znovu (jsou vkládány pouze nové inzeráty). Tento fakt se porovnává při procházení seznamu vozidel pomoci adresy URL[8].

4.6 Uživatelské rozhraní

Uživatelské rozhraní patří k méně podstatným částem celé práce, proto je vytvořeno jako jednoduchá webová aplikace.

Toto rozhraní využívá pouze administrátor celé aplikace a může po přihlášení zadat základní nastavení chování skriptu. Toto nastavení obsahuje pouze čas prvního spuštění, periodu stahování a výběr serveru. Čas prvního spuštění bude pro systém znamenat informaci, v jakém čase se poprvé spustí

[8] Adresa informující o umístění webové stránky[5]

stahování dat. Perioda znamená, jak často se bude stahování dat realizovat (stahování se bude spouštět vždy ve stejný čas, podle zadaného času prvního stahování). Protože samotné stahování dat je časově velmi náročné, tak uživatelské rozhraní obsahuje výběr serverů, ze kterých se data mohou stahovat. Tím pádem se data nemusí stahovat vždy ze všech serverů. Pro případ, že by se administrátorovi celé aplikace zdály informace v databázi zastaralé, uživatelské rozhraní obsahuje možnost vymazání všech záznamů z databáze.

Informace o času prvního spuštění a periodě stahování nejsou v této práci využity (jedná se spíše o rozšíření aplikace do budoucna), protože prozatím aplikace není vložena na žádný webový server, je spouštěna pouze na lokálním zařízení. Za tímto účelem uživatelské rozhraní obsahuje tlačítko spustit stahování, čímž se ihned spustí stahování dat z vybraných serverů.

5 Návrh

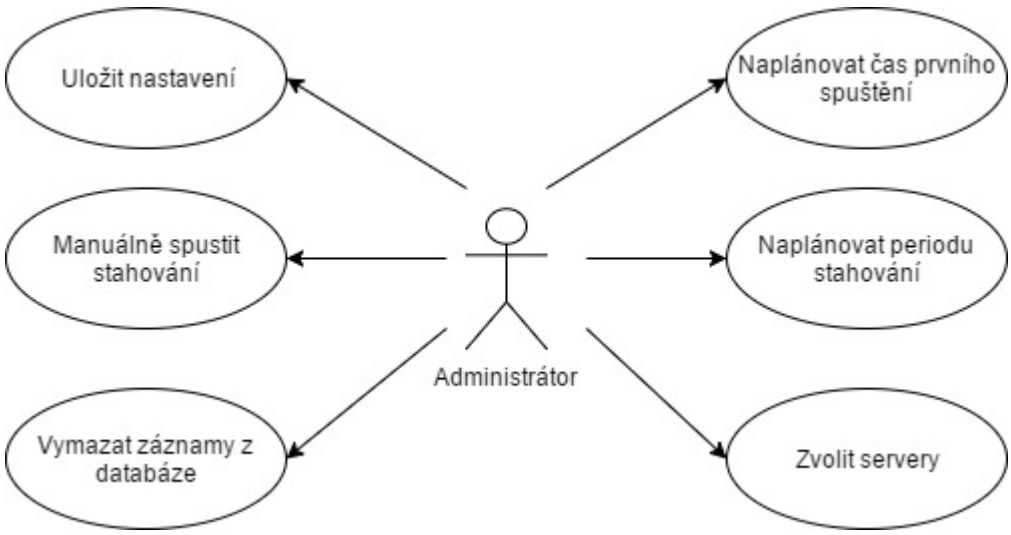
5.1 Diagram případů užití (Use case diagram)

Pro zachycení požadavků na systém se využívají diagramy případů užití, který je součástí jazyka UML[9]. Diagram případů užití můžeme chápat jako funkce systému vykonávané z pohledu jednotlivých účastníků nebo v jejich prospěch. Specifikuje tedy akce, které mohou jednotliví účastníci v systému vykonávat a určuje hranice mezi systémem a účastníky.[6]

Tento model byl vytvořen na základě požadavků na výsledný systém a zobrazuje akce, které lze z pohledu uživatele vykonávat. Tato bakalářská práce obsahuje pouze jednoduché uživatelské rozhraní, proto výsledný model případů užití je velmi strohý.

Model obsahuje pouze jednoho účastníka, kterým je administrátor, který může vykonávat tyto jednotlivé akce:

- Naplánovat čas prvního spuštění stahování dat.
- Naplánovat periodu stahování, která se bude počítat od každého spuštění stahování.
- Vybrat servery, ze kterých chce data stahovat (urychlení stahování).
- Uložit nastavení o stahování.
- Ihned manuálně spustit stahování některého ze serverů bez jakéhokoliv plánování.
- Vyprázdnit tabulky databáze vymazáním všech záznamů.



Obrázek 5.1: Diagram případů užití

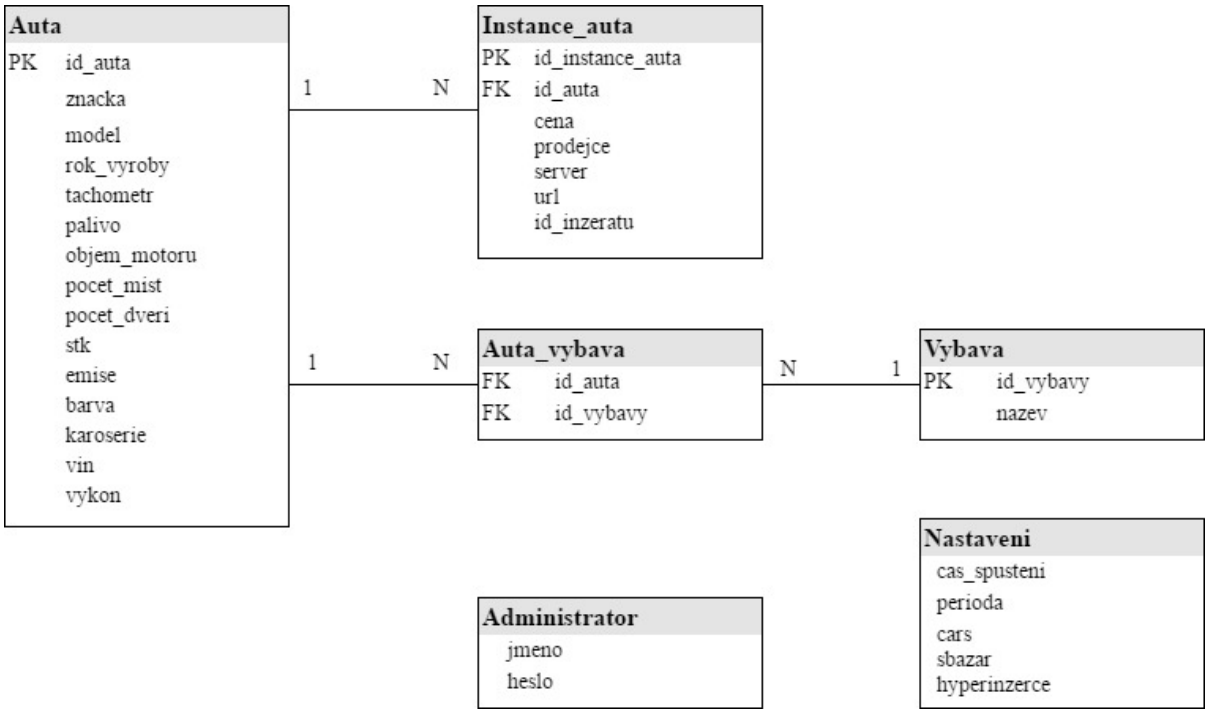
[9] Unified Modelling Language – grafický jazyk pro navrhování programových systémů

5.2 Entitně relační diagram (ERD)

ERD slouží jako návrh uložených statických dat v databázi. Modeluje data, která potřebuje v systému uchovat a vztahy mezi těmito daty. Na základě tohoto modelu se tvoří návrh samotná databáze. Model tvoří entity a vztahy mezi nimi. Každá entita musí být jedinečná a znázorňuje tabulku v databázi. Každá entitní množina obsahuje atributy. Atribut je vlastnost entity, která nás v kontextu daného problému zajímá. Výsledný diagram tvoří neorientovaný graf, kde entity představují uzly a hrany vztahy mezi entitami. [6]

Model byl vytvořen pro uložení základních informací a obsahuje 6 entit:

- **Auta** – tato entitní množina představuje jedno konkrétní vozidlo. Primárním klíčem této entity je atribut `id_auta`. Dalšími atributy jsou základní informace o vozidle.
- **Instance_auta** - tato tabulka představuje konkrétní inzerát. Ze vztahu je patrné, že jeden automobil může být inzerován vícekrát. Primárním klíčem je atribut `id_instance_auta`. Tabulka obsahuje také jeden cizí klíč `id_auta`.
- **Vybava** – primárním klíčem tabulky je `id_vybavy`. Tato entita představuje výbavu vozidel díky atributu `nazev`.
- **Auta_vybava** – této tabulce se říká vazební tabulka, protože tvoří pomoci cizích klíčů vztah N:M. V tomto případě se jedná o cizí klíže z tabulek `Auta` a `Vybava`.
- **Administrator** – tabulka reprezentující administrátora stahování. Atributy obsahují přihlašovací údaje.
- **Nastaveni** – tato entita reprezentuje nastavení plánování stahování.



Obrázek 5.2: Entitně relační diagram

6 Implementace

Tato část bakalářské práce se zabírá samotnou implementací aplikace. Jsou zde popsány využité knihovny, pořizování dat z jednotlivých webových serverů a uživatelské rozhraní.

Na začátku celé implementace bylo vyzkoušeno stažení pár základních informací z jednotlivých serverů a bylo zjištěno, že server www.tipcars.cz není možné pro zisk dat využít, protože obsahuje ochranu proti takzvaným „internetovým botům“. Dále bylo zjištěno u serveru www.sauto.cz, že HTML kód výpisu jednotlivých inzerátů je skrytý pomoci javascriptu to znamená, že je potřeba použít speciální program pro získání tohoto HTML kódu. Pro tuto situaci slouží program PhantomJS[10]. Avšak server www.sauto.cz není možné pro zisk dat využít, protože obsahuje také ochranu proti použití programu PhantomJS.

Celá aplikace obsahuje tři základní skripty, pomocí kterých probíhá samotné získávání dat z webových serverů autobazarů a několik dalších souborů tvořící uživatelské rozhraní.

6.1 Postup zisku dat z webových serverů

Základní postup zisku dat je pro všechny skripty stejný a je rozdělen do několika částí. V této podkapitole si uvedeme pouze základní postup, kterým se řídí všechny skripty. Detailnější popis implementace jednotlivých skriptů bude uveden v dalších podkapitolách.

Na úplném začátku implementace bylo potřeba vybrat URL odkaz, ze kterého začne vyhledávání potřebných informací. Pro všechny skripty byl vybrán odkaz, který vedl k první straně výpisu všech inzerátů bez použití jakýchkoliv vyhledávacích filtrů. Seřazení inzerátů (podle ceny, data vložení atd.) není pro nás podstatné, protože toto seřazení nemá na funkci systému žádný vliv, pouze v tomto pořadí budou data ukládána do databáze. Jedna stránka výpisu obsahuje přibližně 20 – 35 inzerátů.

Skript tedy po načtení první strany výpisu inzerátů vyhledá odkaz na konkrétní inzerát (v některých případech vyhledá i cenu vozidla, které je inzerováno). Provede kontrolu, zda se tento inzerát nachází v databázi, pokud ano, přechází k dalšímu inzerátů, pokud ne, načte odkaz příslušného inzerátu a pokusí se na stránce detailu inzerovaného vozidla získat co nejvíce možných informací o vozidle. Jestliže se povedlo získat dostatečný počet informací o vozidle, přechází se k zápisu získaných informací do databáze.

Před samotným zápisem do databáze se zjišťuje, zda se nejedná o duplicitní vozidlo (jedno vozidlo může být inzerováno na více webových serverech). Při duplicitním vozidle se v databázi vytvoří pouze nová instance vozidla. Při novém vozidle se do databáze vloží nejen jeho instance, ale také jeho informace.

[10] Dostupný na http://phantomjs.org/

Po projití všech inzerátů z první strany výpisu se pomocí odkazu na další stránku přejde na další stranu výpisu a cyklus procházení jednotlivých inzerátů se opakuje. V případě, že jsme již došli na poslední stránku výpisu (neexistuje odkaz na další stránku), aplikace vypíše počet nově stažených inzerátů a čas stahování. V případě, kdy si vybereme v uživatelském rozhraní stažení dat z více webových serverů, tak po ukončení stahování z jednoho webového serveru se spustí celý cyklus stahování z dalšího serveru.

6.2 Načtení a vyhledávání v HTML

Základem všech tří skriptů pro zisk dat je načtení požadovaného odkazu, stažení jeho HTML struktury do proměnné a následné vyhledávání informací z této proměnné.

Projekt pro co nejjednodušší práci s HTML strukturou využívá knihovnu `simple_html_dom.php`[11]. Tato knihovna slouží pro vyhledávání informací v HTML kódu. Aplikace z knihovny využívá funkci `file_get_html` pro získání HTML struktury z webové stránky a funkci `find` pro vyhledávání informací z HTML struktury. Funkce pro stažení HTML struktury a uložení do proměnné je naznačená na ukázce Ukázka 6.1.

```
$nacteny_odkaz = file_get_html($adresa, false, $GLOBALS["context"]);
```

Ukázka 6.1: Stažení HTML stránky do proměnné

Proměnná `$adresa` obsahuje URL adresu požadované stránky, globální proměnná `$GLOBALS["context"]` určuje, jakým způsobem má být URL adresa načtena. Do proměnné `$nacteny_odkaz` je uložena stažená HTML struktura. Způsob, jakým lze vyhledávat potřebné informace z HTML struktury je vidět na ukázce Ukázka 6.2.

```
$seznam_aut->find('div.ftr strong', 7)->plaintext;
```

Ukázka 6.2: Vyhledání informací z HTML

V proměnné `$seznam_aut` je uložena HTML struktura ze které probíhá vyhledávání. Vyhledává se značka `div`, která má třídu s názvem `ftr`. V této nalezené značce se vyhledá další takzvaná vnořená značka `strong`. Číslice 7 určuje kolikátý výskyt hledané značky nás zajímá. Slovo `plaintext` nám říká, že chceme získat pouze text, který je zapsán mezi značky.

[11] Dokumentace dostupná na http://simplehtmldom.sourceforge.net/manual.htm

6.2.1 Kontrola načtení adresy

Stahování dat z jednoho webové serveru může trvat až 40 hodin. Během tohoto stahování může dojít například ke krátkodobému výpadku internetu a tím by se celé stahování přerušilo. Aby nebylo přerušeno stahování při krátkodobém výpadku, obsahuje aplikace funkci, která kontroluje, zda se povedla adresa načíst a získat HTML strukturu či nikoliv. Obsah této funkce je uveden v ukázce Ukázka 6.3.

```
$nacteny_odkaz = file_get_html($adresa, false, $GLOBALS["context"]);
$citac_zdrzeni = 0;
while ($nacteny_odkaz == false) {
      if (chyba_404($adresa) == true){
            return false;
      }
      if ($citac_zdrzeni == 5) {
            return false;
      }
      sleep(60);
      $citac_zdrzeni++;
      $nacteny_odkaz = file_get_html($adresa, false, $GLOBALS["context"]);
      }
return $nacteny_odkaz;
```

Ukázka 6.3: Kontrola načtení adresy

V případě, kdy se nepovede odkaz z jakéhokoliv důvodu načíst, tak program 60 sekund počká a následně zkusí znovu načíst tu stejnou adresu. Tento postup se děje po dobu 5 minut a v případě, že se ani po této době nepovede odkaz načíst, tak se stahování buď přeruší (načítaná adresa je nepostradatelná pro další chod stahování) nebo se odkaz přeskočí a jde se na další (jedná se o odkaz na detail vozidla, což neovlivňuje další chod stahování). Čekání 60 sekund před opětovným načítáním bylo zvoleno z toho důvodu, že se většinou dříve, než za tuto dobu, nepovede výpadek vyřešit. Horní hranice čekání 5 minut byla zvolena dle vlastní úvahy, protože ve většině případů, pokud se nepovede problém vyřešit do 5 minut, tak se jedná o delší výpadek, na který nemá smysl čekat.

Při testování kontroly načtení jednotlivých adres bylo odhaleno, že některé odkazy na inzeráty jsou nefunkční, protože jejich stránka nebyla nalezena a tím pádem se celková doba stahování ještě prodlouží, protože se aplikace pokouší tuto stránku načítat po dobu 5 minut. Za účelem zkrácení tohoto zbytečného načítání byla vytvořena funkce `chyba_404`, jejíž zavolání můžeme vidět na ukázce Ukázka 6.3 a tělo této funkce na ukázce Ukázka 6.4.

```
      $ch = curl_init();
      curl_setopt($ch, CURLOPT_URL, $adresa);
      curl_setopt($ch, CURLOPT_NOBODY, 1);
      curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
```

```
            curl_exec($ch);
            if   (curl_getinfo($ch,   CURLINFO_HTTP_CODE)   ==   404   ||
curl_getinfo($ch, CURLINFO_HTTP_CODE) == 301){
                  return true;
            }
            curl_close($ch);
            return false;
```

Ukázka 6.4: Odhalení chyby nefunkční adresy

Funkce se pokusí do proměnné `$ch` načíst zadanou adresu. Pokud načítaná adresa hlásí stavový kód 404 (stránka nenalezena) nebo 301 (trvalé přesměrování), tak funkce vrací hodnotu `true` a tím se přeskočí doba opakovaného načítání adresy, což bylo možno vidět na ukázce Ukázka 6.3. Stavový kód 301 byl do funkce přidán na základě testování, kdy byl odhalen velmi častý výskyt tohoto stavového kódu při načítání adresy.

Jelikož se v systému snažíme načítat HTML strukturu co možná nejméně (abychom server nevytížili natolik, že by došlo k jeho výpadku), tak se nemůžeme vyhnout varování (warningu), který nás upozorňuje právě na to, že zadaný odkaz nelze načíst. Abychom se tomuto varování vyhnuli, tak bychom museli nejdříve otestovat, zda lze odkaz načíst a následně stáhnout HTML strukturu. Tímto způsobem bychom načítali adresu dvakrát a zatěžovali tak více webový server, ze kterého se právě stahují data.

```
Warning: 
file_get_contents(http://autobazar.hyperinzerce.cz/ford/inzerat/11765857-
ford-focus-1-8-nafta-r-v-2010-nabidka-praha/): failed to open stream: HTTP
request failed! in C:\xampp\htdocs\bakalarka\simple_html_dom.php on line 76
```

Ukázka 6.5: Varování

6.3 Průchod inzeráty

Na začátku této části se načte první strana výpisu všech inzerátů. Tato první strana byla zvolena z toho důvodu, že je to nejjednodušší způsob, jakým se lze dostat ke všem inzerátům, které daný server nabízí. Pokud se nepovede načíst již tato úvodní strana, tak se skript ukončí s chybovou hláškou.

Z úvodní strany se zjistí celkový počet inzerátů, které server obsahuje. Z celkového počtu se následně počítají procenta, podle kterých administrátor zjistí, kolik procent dat již bylo staženo. Tyto procentuální údaje však nemusí být úplně přesné, protože stahování dat trvá dlouhou dobu a během té doby může přibýt nebo být vymazáno spoustu inzerátů. Tato nepřesnost by se dala eliminovat častějším získáváním celkového počtu inzerátů, avšak tato drobná nepřesnost není pro aplikaci zásadní a proto systém nezatěžujeme častějším získáváním celkového počtu inzerátů.

Následuje smyčka, ve které se zjistí adresa detailu následujícího inzerátu. Adresa u všech tří webových serverů obsahuje identifikační číslo inzerátu. Pomoci identifikačního čísla inzerátu lze jednoznačně určit, o jaký inzerát se jedná. Nejprve se zjistí, zda již identifikační číslo inzerátu je v databázi uloženo, pokud ano, znamená to, že inzerát již je v databázi uložen a není potřeba znova získávat jeho informace z webového serveru (přestupuje se na další inzerát). V případě, že identifikační číslo inzerátu ještě v databázi není, uloží se toto číslo do databáze, načte se adresa inzerátu a přistoupí se k zisku dat. Jakmile se takto projdou všechny inzeráty na stránce, přistoupí se na další stránku pomoci odkazu na spodu stránky. Pokud se odkaz nepovede načíst, skript vypíše chybovou hlášku a ukončí se. Pokud odkaz na další stranu není k dispozici, znamená to že se jedná o poslední stranu, vypíší se informace o počtu nově stažených inzerátů, čas stahování a skript se ukončí.

```
$dalsi_strana = $seznam_aut->find('span.nextLink a', 0)->href;
$seznam_aut = nacteni_odkazu_cars('http://www.cars.cz/inzerce/' .
$dalsi_strana);
```

Ukázka 6.6: Zisk adresy další stránky

Na ukázce Ukázka 6.6 můžeme vidět zisk a načtení adresy další strany výpisu inzerátů. Je třeba si dát pozor, že některé adresy neobsahují kompletní adresu, ale pouze její část. Proto je potřeba doplnit kořenovou část adresy ručně což je možné vidět taktéž na ukázce.

6.4 Zisk dat

Zisk dat je nejdůležitější část celé aplikace a je na něj kladen největší důraz. U každého serveru jsou data udávaná v jiném formátu, proto je potřeba je sjednotit do jednotného tvaru. Téměř všechny informace jsou získávány vyhledáváním značek z HTML struktury pomocí funkce `find`. Pouze ze serveru www.sbazar.cz není možné získat všechny informace vyhledáváním v HTML struktuře a proto jsou informace vyhledávány pomocí hledání klíčových slov v prostém textu. Každá informace je ukládaná do speciální proměnné, která nese název podle typu informace. Pokud se některá data nepovede získat nebo nejsou uvedeny, příslušná proměnná je nastavená na hodnotu `NULL`. Na každém serveru jsou data ukládána odlišným způsobem, proto je potřeba mít pro každý server speciální skript, který data ze serveru získá. Jednotlivé skripty pro zisk dat ze serverů budou popsány v podkapitolách níže.

6.4.1 Úprava dat do jednotného tvaru

Každý server má svůj formát, ve kterém uvádí data. Některé formáty se shodují, ale některé ne. Každý typ informace je potřeba mít v jednotném formátu, protože se podle těchto informací vyhledávají

duplicitní vozidla a také navazující aplikace, kterou naprogramoval Nikolas Kantor, informace využívá k vyhledávání určitých vozidel.

U značky vozu se problém různých zápisů týká pouze značky Citroën, protože na některých serverech se uvádí tato značka jako Citroen (bez přehláskovaného e).

U modelu vozu byl zjištěn problém u modelu Cee´d. Na některých serverech byl tento model uváděn bez apostrofu jako Ceed.

Další úpravou byl rok výroby. V tomto případě nás zajímá pouze rok. Někdy se objevuje rok výroby zadán jako například: 1.6.2010, 1/6/2010. Proto byl rok výroby sjednocen na čtyřčíslí například: 2010.

V případě počtu najetých kilometrů byla potřeba udělat větší úprava pouze u serveru www.sbazar.cz protože zde může být uvedeno například: 91tis., 91t., 91 xxx. Tyto hodnoty byly změněny do podoby 91000. U ostatních serverů byly odmazány jednotky a ponecháno pouze číslo.

Problémovým druhem informací bylo palivo. Zde můžeme setkat s různými názvy a to zejména u vozidel poháněných plynem. Jelikož některé servery nerozlišují plyn na LPG a CNG, tak i tato aplikace nerozlišuje nijak toto palivo. Proto bylo zvoleno prosté slovo plyn. U paliva byla ještě potřeba sjednotit nafta, jelikož občas bývá uvedena nafta synonymem diesel.

Objem může být zadán ve dvou tvarech. Jeden tvar je udáván v litrech například: 1.9, 1.8. Druhý tvar je udáván v krychlových centimetrech například: 1896ccm. Nejčastěji je objem udáván v centimetrech krychlových, proto je to pro aplikaci výchozí formát. Objem udávaný v litrech je tedy převáděn na centimetry krychlové. Tento převod však není v aplikaci úplně přesný, protože jen doplní za objem v litrech dvě nuly.

U výkonu není potřeba kromě odmazání jednotek dělat žádné úpravy, protože je formát na všech serverech stejný. Jediné na co bylo potřeba si dát pozor, bylo to, abychom stahovali výkon v kilowatech a ne v koních (některé servery udávají výkon v kilowatech i v koních).

V případě počtu dveří, sedadel a STK[12] vozidla není potřeba dělat žádné úpravy, protože tyto informace jsou u inzerátu zadány jen zřídka a vždy ve stejném tvaru.

Systém získává také barvu vozidla. U barvy bývá často udáno, že se jedná o metalízu. Tato informace je z údaje o barvě vymazána, protože tento údaj někdy uveden je a někdy není a tím pádem by mohl vzniknout problém s vyhledáváním vozidel. Občas bývá u barvy dodána informace, zda se jedná o tmavý či světlý odstín. Tato informace je stejně jako metalíza odmazána. Systém tedy získá pouze základní informaci o barvě.

Největší úpravou musí projít karosérie vozidel. Typ karosérie je téměř na každém serveru uváděn trochu jinak. Některé typy karosérií jsou sloučeny pod jeden typ, protože jinak by nebylo možné dosáhnout jednotného formátu.

[12] Udává datum příští návštěvy technické prohlídky vozu

Obrázek 6.1: Karoserie u www.cars.cz

Obrázek 6.2: Karoserie u www.hyperinzerce.cz

Na obrázcích 6.**Chyba! Nenalezen zdroj odkazů.**1 a 6.2 můžeme vidět rozdíly v karosériích na serverech www.cars.cz a www.hyperinzerce.cz. Výsledné názvy typu karosérií, jsou tedy:

- Hatchback,
- Sedan / Limuzína,
- Kombi,
- SUV / MPV,
- Van (do tohoto názvu spadá i Minibus),
- Pick-up,
- Kupé (do tohoto názvu patří i Sportovní),
- Terénní (zde patří i Offroad),
- Kabriolet.

6.4.2 Cars.cz

Skript pro zisk dat ze serveru www.cars.cz je ze všech tří skriptů nejjednodušší. Kromě sjednocení typu dat je potřeba také ignorovat inzeráty, které nabízí náhradní díly. Jestliže z inzerátu nebyly zjištěny informace o počtu najetých kilometrů, objemu a výkonu nebo tyto informace nabývaly hodnoty 0, tak se většinou jedná o náhradní díl a tyto inzeráty jsou ignorovány. Bohužel u pár inzerátů jsou informace zadány ve stylu, jakoby se jednalo o vozidlo a tím pádem není možné rozpoznat, že se jedná právě o náhradní díl.

Dalšíma zároveň posledním problémem u tohoto serveru byla speciální struktura HTML u některých prodejců. Inzerátů s vlastní strukturou HTML je jen nepatrné množství. Pokud nějaký inzerát obsahuje vlastní strukturu HTML, tak není možné použít vyhledávání, které je vytvořeno pro stahování z tohoto serveru. Proto bylo potřeba u příslušného inzerátu zjistit, že se nejedná o originál strukturu serveru www.cars.cz a tento inzerát ignorovat.

```
$je_cars = $info_o_aute->find('body#carDetail',0);
if ($je_cars != NULL) {
 //zisk dat
}
```

Ukázka 6.7: Originál HTML struktura www.cars.cz

Na ukázce Ukázka 6.7 je možné vidět kontrolu, zda se jedná o originální strukturu z www.cars.cz nebo má prodejce svou speciální strukturu. Originální strukturu aplikace rozpoznává pomoci těla, které nese identifikátor `carDetail`. Pokud tělo nemá tento identifikátor, nejedná se o originální stránku serveru www.cars.cz a tento inzerát je tím pádem ignorován.

6.4.3 Hyperinzerce.cz

U tohoto serveru je poměrně velké množství inzerátů, které mají vlastní strukturu HTML, proto není možné tyto inzeráty ignorovat. Na začátku zisku dat je tedy potřeba zjistit, o jakou HTML strukturu se jedná. Toto zjištění můžeme vidět na ukázce Ukázka 6.8.

```
$hyperinzerce = $info_o_aute->find('h1.f30',0);

        if ($hyperinzerce != NULL) {
              hyperinzerce($info_o_aute, $adresa, $cena, $id_inzeratu);
        }
        elseif ($info_o_aute->find('iframe',0) != NULL) {

              $info_o_aute  =  nacteni_odkazu_hyperinzerce($info_o_aute-
>find('iframe',0)->src);
              if ($info_o_aute == false) {
                    return 1;
              }

              if ($info_o_aute->find('h2.not-found',0) != NULL) {
                    return 1;
              }
              elseif ($info_o_aute->find('div.car-card',0) != NULL) {
                    aaaauto($info_o_aute, $adresa, $cena, $id_inzeratu);
              }
              elseif ($info_o_aute->find('body.esa',0) != NULL) {
                    autoesa($info_o_aute, $adresa, $cena, $id_inzeratu);
              }
        }
```

Ukázka 6.8: Zjištění zdroje HTML struktury

Nejprve je tedy zjištěno, pomoci nadpisu s třídou `f30,` zda se jedná o originál strukturu z hyperinzerce. Pokud ano, zavolá se funkce pro zisk dat z hyperinzerce. Pokud ne, zjišťuje se, jestli je do HTML vložená externí HTML struktura pomoci značky `iframe` a načte se jako nová struktura, ve

které se následně vyhledává. V případě, že se v nové HTML struktuře najde `div` označený třídou `car-card`, jedná se o strukturu prodejce AAA Auto. Pokud se v HTML struktuře najde tělo s třídou `esa`, jedná se o strukturu prodejce AutoESA. Následně systém zavolá funkci pro zisk dat od příslušného prodejce.

U struktury hyperinzerce nejsou žádné další problémy, stačí pouze pohlídat jednotný tvar informací.

U prodejce AutoESA je situace podobná, jako u hyperinzerce s tím rozdílem, že jsou data zapsána v jednom textu oddělená lomítkem. Pomoci tohoto lomítka jsou data rozdělena do několika sekcí, u kterých víme, že například za pátým lomítkem je informace o karoserii vozidla. Výbava u tohoto prodejce není jednoduše vyhledatelná pomoci jednoznačného identifikátoru, proto musí vyhledávat nejen pomoci značky, ale také pomoci nadpisu, který by měl být Interiér, Exteriér nebo Bezpečnost. V případě, že systém narazí na některý z těchto nadpisů, víme, že v této části HTML struktury se vyskytují informace o výbavě.

Nejvíce problémů nastává u struktury, kterou jsme brali jako AAA Auto. Bylo zjištěno, že stejnou strukturu využívají také prodejci Mototechna a Auto Diskont. To by však nebyl zásadní problém, kdyby se ze struktury daly získat všechny pro nás důležité informace. Ze struktury se dají získat všechny důležité informace kromě barvy. Kvůli zisku barvy je nutné se dostat na stránku konkrétního prodejce.

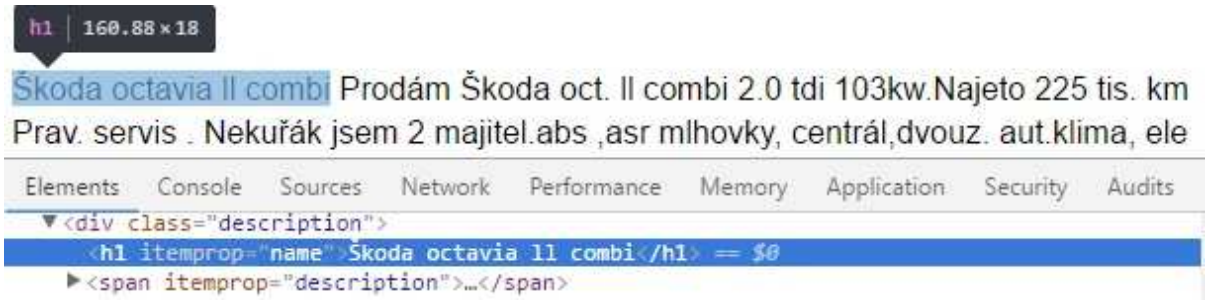
Obrázek 6.3: Odkaz na prodejce

Na obrázku 6.3 je vidět odkaz s názvem „Nezávazná rezervace vozu“ přes který se dostaneme na stránku konkrétního prodejce vozu, kterým může být AAA Auto, Mototechna nebo Auto-Diskont.

Pomoci adresy zjistíme, o kterého prodejce se jedná a získáme informace o barvě z jeho stránek. Tím, že je nutné načíst pro zisk barvy tři různé HTML struktury se celkové stahování dat razantně zpomalí. Pro co nejpřesnější vyhledávání duplicitních vozidel je však barva celkem podstatná vlastnost, proto je získávaná i na úkor rychlosti celého stahování.

6.4.4 Sbazar.cz

Na serveru www.sbazar.cz jsou informace zapsány formou prostého textu, proto je samotný zisk dat, oproti zisku dat z předchozích serverů, naprosto odlišný. Pro zisk dat jsou použity regulární výrazy a existující záznamy z databáze.



Obrázek 6.4: Nadpis v textu inzerátu

Na obrázku 6.4 je označený nadpis, který pouhým okem (bez náhledu do HTML) není viditelný. V tomto nadpise se většinou uvádí značka, model a občas i objem a karosérie. Protože vyhledání značky a modelu z delšího textu může vést k nalezení chybných informací, tak se nejprve značka a model vyhledává v tomto nadpise a až v případě, že se značku a model nepovede nalézt, se vyhledává ve zbytku textu. Všechny regulární výrazy byly vytvořeny po analýze a zjišťování v jakém různém formátu může být hledaná informace zadána.

Informace o ceně a prodejci vozidla jsou jako jediné získány vyhledáváním v HTML struktuře.

```
/((\d{2}|\d{3})(\s|)kw)/i
```

Ukázka 6.9: Regulární výraz pro zisk výkonu

Na ukázce Ukázka 6.9 můžeme vidět regulární výraz pro získání výkonu vozidla. Regulární výraz vyhledává dvě nebo tři čísla, za kterými následuje mezera nebo jakýkoliv jiný znak a za tímto znakem jednotka kw. Písmeno `i` na konci regulárního výrazu značí case insensitive, to znamená, že v celém regulárním výrazu nezáleží na velikosti písmen.

O poznání složitější je regulární výraz pro vyhledávání roku výroby vozidla, který můžeme vidět na ukázce Ukázka 6.10. Rok výroby může být zadán velmi mnoha způsoby, takže je možné, že ani tento regulární výraz nezvládne nalézt rok výroby z každého inzerátu.

```
/(reg|registrace|rok
v(ý|y)roby|r.{1,2}v(ý|y)roby|rok|r\..v\.|r\.v\.|rv|vyrobeno|v.rob.|provoz.
).{0,6}(\d{1,2}(\/|.)\d{4}|\d{4})/i
```

Ukázka 6.10: Regulární výraz pro zisk roku výroby

Na začátku hledá regulární výraz některé ze slov: reg, registrace, rok výroby, r. výroby, rok, rv, vyrobeno, výroba, provoz. Za tímto slovem se může nacházet nula až šest jakýchkoliv znaků. Tato tolerance je uvedena, aby regulární výraz nalezl správný výsledek, protože rok výroby může být uveden například: 30/10/2015 nebo jen 2015 a také nemusí být uveden hnedka za klíčovým slovem. Na konci regulárního výrazu se vyhledává samotné datum, které může být buďto ve tvaru jedna až dvě číslice, následuje jakýkoliv znak a další čtyři číslice nebo pouze čtyři číslice. Regulární výraz nalezne rok i například z tohoto textu: 5.2015, 3/2015, 10/2003, 2000.

```
/((tachometr|najeto|ujeto|tach.|naj.).{0,5}\d{0,3}(.{0,1}tis.|.{0,1}xxx|.{
0,1}\d{3}|\s{0,1}km)|\d{0,3}(.{0,1}tis.|.{0,1}xxx|.{0,1}\d{3})\s{0,1}km)/i
```

Ukázka 6.11: Regulární výraz pro zisk počtu najetých kilometrů

Na ukázce Ukázka 6.11 je regulární výraz pro zisk počtu najetých kilometrů. Tento regulární výraz je nejsložitější regulární výraz, který je v systému použit a je celkem složité se v něm vyznat, proto bude popsán zjednodušeně a co nejsrozumitelněji. Regulární výraz hledá číslo s údajem o počtu najetých kilometrů které obsahuje nula až tři číslice, za nimi může následovat buď slovo tis, xxx nebo další tři číslice. Údaj o počtu najetých kilometrů tedy může být například: 21tis, 21 xxx, 21923, 256. Abychom jistě věděli, že se jedná o počet najetých kilometrů, musí být buď před číslem jedno ze slov: tachometr, najeto, ujeto, tach., naj. nebo se musí za číslem objevit jednotka km.

Dalším informací, kterou aplikace ze serveru www.sbazar.cz získává je palivo. Velmi často inzeráty neobsahují informaci o tom, jakým palivem je vozidlo poháněno, protože tuto informaci nese zkratka označení motoru. Ke zjištění, co která zkratka znamená nám byl nápomocen server www.autolexikon.net[13]. Regulární výraz pro nalezení naftových motorů je na ukázce Ukázka 6.12 a regulární výraz pro nalezení benzínových motorů na ukázce Ukázka 6.13.

```
/(naft.{0,3}|diesel.{0,3}|dci|tdi|hdi|d-4d|cdti|cdi|tddi|tdci)/i
```

Ukázka 6.12: Nalezení naftových vozidel

```
/(benzín.{0,3}|tsi|hpi|fsi|mpi|\d(\.|,)\di)/i
```

Ukázka 6.13: Nalezení benzínových vozidel

[13] Označení motoru dostupné na http://www.autolexicon.net/cs/articles/tag/oznaceni-motoru/

Na těchto dvou ukázkách jsou zajímavé pouze nula až tři různé znaky, pomocí kterých je aplikace schopná vyhledat i slova jako je například: benzínové, dieselové atd. Benzínové motory lze rozpoznat i na základě objemu, kdy za číslem (udávané v litrech) je písmeno i (například: 1.8i). U plynových motorů se vyhledávají pouze slovo plyn nebo zkratky LPG a CNG.

Objem může mít dva způsoby zápisu, buďto se udává v litrech (1,8, 1.9 atd.) nebo v centimetrech krychlových (1938ccm, 1600ccm atd.). Z tohoto uvážení vyplývá regulární výraz, který je možno vidět na ukázce Ukázka 6.14.

```
/(\s\d(\.|,)\d{1,2}|\d{3,4}.{0,1}ccm)/i
```

Ukázka 6.14: Regulární výraz pro zisk objemu motoru

Poslední sadu regulárních výrazů tvoří zisk karoserie. Tyto regulární výrazy vyhledávají v podstatě jen názvy karoserií v textu.

Zjištění barvy vozidla je vytvořeno za použití již existujících záznamů databáze. Z databáze se tedy zjistí všechny možné názvy barev a ty jsou následně vyhledávány v textu. Pro ještě větší šanci nalezení barvy v textu je potřeba barvu vyskloňovat do druhého a čtvrtého pádu. Za tímto účelem byla použita knihovna Czech inflection[14], pomocí které lze skloňovat česká slova. Na ukázce Ukázka 6.15 je možné vidět využití skloňování v systému.

```
$inflected = $inflection->inflect($radek["barva"]);

if (mb_stripos($info_o_aute, $radek["barva"]) !== false ||
mb_stripos($info_o_aute, $inflected[2]) !== false ||
mb_stripos($info_o_aute, $inflected[4]) !== false){
      // telo funkce
}
```

Ukázka 6.15: Použití skloňování

Vyskloňovaná barva se uloží do pole `$inflected` a následně si stačí vybrat, podle pozice v poli, o jaký pád slova máme zájem. V toto případě chceme druhý a čtvrtý pád.

Značka a model jsou vyhledávány stejně jako barva podle již existujících záznamů v databázi. Bohužel se pro značku a model nedá využít výše zmíněné skloňování, protože to knihovna neumí (skloňuje názvy značek a modelů špatně). Proto je vytvořeno provizorní skloňování, které funguje jen u značek, které obsahují více než čtyři znaky. Poslední písmeno značky je vymazáno a hledá se pouze část slova, tím pádem jsme v textu schopni najít například značku Škoda, ale i Škodu. Omezení na více než 4 musí být kontrolováno, protože kratší slova vedou ke špatně nalezeným informacím. U vyhledávání modelu nemůže být tento způsob použit, protože většina modelů má krátký název.

[14] Dostupná na https://packagist.org/packages/heureka/inflection

V případě nalezení značky se vyhledávání modelu omezí pouze na modely nalezené značky. Značka Volkswagen je v textu často zapisována zkratkou vw, proto je systém připraven i na tuto variantu zápisu.

6.5 Uložení dat do databáze

Pro uložení dat do databáze jsou v každém skriptu dvě funkce. Jedna pro uložení informací o vozidle a jeho instanci `databaze_auta_instance()` a druhá pro uložení výbavy a propojení výbavy s `vozidlem databaze_vybava()`.

Ve funkci `databaze_vybava()` se nejdříve zjistí, zda se daná výbava již nachází v tabulce. Pokud má název výbavy třeba jen jedno písmeno jiné, než je v databázi, bere se to jako nová výbava a ukládá se do databáze. Následně jsou vloženy cizí klíče do vazební tabulky `auta_vybava`, která zajišťuje propojení vozidla s výbavou.

Do funkce `databaze_auta_instance()` přicházejí už ujednocené data. Na začátku se zjistí, jestli získaný VIN kód nebo parametry, popsané v kapitole 4.4 již jsou v databázi. Pokud ano, vkládá se do databáze pouze instance vozidla, v opačném případě se kromě instance vkládají také parametry vozu. V případě, že se nám nepovedl nějaký parametr vozu zjistit, tak se do databáze vkládá hodnota `NULL`.

6.6 Uživatelské rozhraní

Uživatelské rozhraní nepatří mezi důležité části aplikace, proto bylo vytvořeno jen velmi jednoduše. Je vytvořeno jako jednoduchá webová aplikace a proto se spouští přes prohlížeč.

Na začátku se musí administrátor aplikace přihlásit pomoci jmena (admin) a hesla (admin). Jednoduchý formulář pro přihlášení můžeme vidět na obrázku 6.5.

Obrázek 6.5: Přihlášení

Heslo je v tabulce zašifrováno pomocí hašovací funkce. Přihlášení je vytvořeno ve skriptu index.html. Po stisku tlačítka `Přihlásit` se chod programu přestěhuje do skriptu prihlaseni.php. Zde se zkontroluje správné zadání jména a hesla. V případě špatně vyplněných přihlašovacích údajů aplikace vypíše hlášku a vrátí se zpátky k přihlášení. V opačném případě jsou pomocí funkce `echo` vykresleny formuláře, které můžeme vidět na obrázku 6.6.

Obrázek 6.6: Nastavení spuštění stahování

Nastavení stahování je přichystáno pro použití aplikace na serveru (momentálně tedy nefunkční). Při stisku tlačítka `Vyprázdnit databázi` se chod programu přesměruje do skriptu vyprazdneni_db.php, ve kterém se pomocí SQL příkazu `truncate` vymažou všechny záznamy

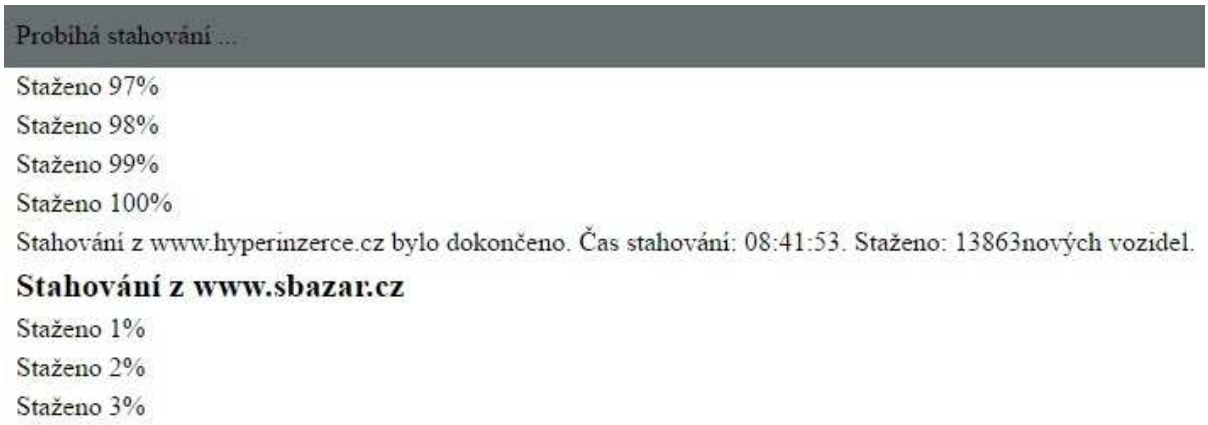
z databáze a chod programu se přesměruje zpátky do skriptu prihlaseni.php. Při stisku tlačítka `Spustit stahování` se chod programu přesměruje do skriptu spusteni.php.

Jelikož stahování dat trvá dlouhou dobu a prohlížeč automaticky přeruší načítání po 30 sekundách, je na začátku skriptu spusteni.php přenastavena maximální délka načítání. Přenastavení maximální délky načítání je možno vidět na ukázce Ukázka 6.16: Maximální délka načítání. Tak velké magické číslo, které je uvedeno v závorkách, je zvoleno proto, aby se vždy stihlo dokončit celé stahování.

```
set_time_limit(999999999999999);
```

Ukázka 6.16: Maximální délka načítání

Podle vyplněného formuláře před spuštěním stahování se pomoci funkce `include_once` spustí stahování z příslušného serveru. Průběh stahování je vypisován v procentech a na konci stahování daného serveru je vypsána doba stahování a počet nově stažených vozidel. Průběh stahování můžeme vidět na obrázku 6.7.



Obrázek 6.7: Průběh stahování

7 Testování

Nejprve probíhalo testování zisku dat z jednotlivých webových serverů během implementace. Po dokončení implementace byl celý systém otestován i včetně uživatelského rozhraní.

V první části implementace jednotlivých skriptů pro zisk dat bylo prováděno jen krátké testování, kterým byla nalezena spousta možností, jakými jsou informace zapisovány na serverech autobazarů. Díky tomuto testování aplikace zvládá zpracovat data zapsána v různém formátu. Jakmile se zdála být implementace konkrétního skriptu hotová, tak se testovalo spuštění celého skriptu, které odhalilo další množství nedostatků, které byly opraveny. Jedním z hlavních nedostatků byla situace,

kdy došlo k výpadku internetu. Data z jednoho autobazaru se stahují přibližně 10 – 40 hodin a výpadek připojení k internetu byl velkým problémem, který se naštěstí povedlo vyřešit.

Po implementaci všech skriptů byla celá aplikace otestována včetně uživatelského rozhraní. Tyto testy proběhly jak s vkládáním informací do prázdné databáze, tak s vkládáním informací do databáze, která již obsahovala nějaké záznamy. Stahování do prázdné databáze ze serveru www.cars.cz trvalo přibližně 10 hodin při množství přibližně 31000 inzerátů. Při stahování do databáze, která již obsahovala záznamy trvalo stahování necelé 2 hodiny a bylo staženo asi 800 nových inzerátů. Zisk dat z www.sbazar.cz trval téměř stejně dlouho jako z www.cars.cz. U serveru www.hyperinzerce.cz už je čas stahování o poznání delší. Přibližně 41000 inzerátů se stahovalo okolo 35 hodin. Samozřejmě při stahování dat do databáze, která již obsahovala záznamy, se čas stahování pohyboval okolo 2 hodin při stažení 500 nových inzerátů.

Největším problémem celé aplikace je, pokud daný server, ze kterého se získávají data změní HTML strukturu. S tímto problémem si skript neporadí a je ho potřeba upravit podle nové HTML struktury. Tento problém nastal i při testování přibližně týden před odevzdáním, kdy si server www.cars.cz změnil HTML strukturu. Naštěstí tato změna byla jen minimální, takže i úprava skriptu byla jen minimální.

V aplikaci se mohou objevit ještě nějaké chyby, které nebyly při testování odhaleny. Aby aplikace byla bezchybná, tak by bylo potřeba aplikaci testovat alespoň rok nebo dva v plném provozu, protože sémantika přirozeného jazyka je velmi komplikovaná.

8 Závěr

Cílem bakalářské práce bylo vytvořit aplikaci pro zisk dat ze serverů autobazarů napříč ČR, která zvládne data stáhnout a uložit do databáze v předem určeném formátu.

Výsledkem je webová aplikace vytvořená pomocí jazyků HTML, CSS, PHP a MySQL, která získá potřebné informace ze serverů www.cars.cz, www.sbazar.cz a www.hyperinzerce.cz. Aplikace je spustitelná pouze na lokálním zařízení.

Aplikaci je možné do budoucna ještě rozšiřovat a vylepšovat. Hlavním rozšířením do budoucna by mohlo být umístění celé aplikace na nějaký webový server, na kterém by se stahování dat mohlo spouštět v pravidelných intervalech (například jednou týdně). Dalším rozšířením by mohlo být vytvoření dalších skriptů pro stahování dat i třeba z menších autobazarů. Tímto rozšířením by se navýšil celkový počet vozidel a tím pádem by aplikace jako celek mohla se získanými daty ještě více pracovat. U serveru www.sbazar.cz se pomoci této aplikace povede získat pouze jedna třetina inzerátů. Tento fakt by se dal do budoucna také vylepšit, ale musel by se vymyslet nějaký efektivní algoritmus pro zisk dat z volného textu. Vymyšlení tohoto algoritmu by mohlo vést na jedno celé téma nějaké další bakalářské práce.

Literatura

[1] DUCKETT, Jon. HTML & CSS: design and build websites. Indianapolis, IN: Wiley, c2011. ISBN 978-1118008188.

[2] CSS styly - úvod. Jak psát web [online]. [cit. 2017-05-16]. Dostupné z: https://www.jakpsatweb.cz/css/css-uvod.html

[3] What is PHP?. PHP [online]. [cit. 2017-05-16]. Dostupné z: https://secure.php.net/manual/en/intro-whatis.php

[4] About MySQL. MySQL [online]. [cit. 2017-05-16]. Dostupné z: https://www.mysql.com/about/

[5] URL. Adaptic [online]. [cit. 2017-05-16]. Dostupné z: http://www.adaptic.cz/znalosti/slovnicek/url/

[6] Ing. Bohuslav Křena, Ph.D., Ing. Radek Kočí, Ph.D.: Studijní opora úvod do softwarového inženýrství. FIT VUT v Brně, 2010.

Seznam příloh

Příloha A: Obsah CD

Příloha A: Obsah CD

Složka **Databaze** – obsahuje inicializační sql soubory

Složka **Technická zpráva** – obsahuje technickou zprávu v pdf a ve zdrojovém formátu

Složka **Zdrojové soubory** – obsahuje všechny zdrojové soubory a knihovny

Readme.txt – Stručný návod k použití aplikace