



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# WEBOVÁ APLIKACE PRO EVIDENCI HISTORIE SERVISNÍCH ZÁSAHŮ OSOBNÍCH VOZIDEL

## Bakalářská práce

*Studijní program:* N2612 – Elektrotechnika a informatika

*Studijní obor:* 1802R022 – Informatika a logistika

*Autor práce:* **Jakub Švéda**

*Vedoucí práce:* **Ing. Igor Kopetschke**





## Zadání bakalářské práce

# Webová aplikace pro evidenci historie servisních zásahů osobních vozidel

*Jméno a příjmení:* **Jakub Švéda**  
*Osobní číslo:* M15000156  
*Studijní program:* B2612 Elektrotechnika a informatika  
*Studijní obor:* Informatika a logistika  
*Zadávající katedra:* Ústav nových technologií a aplikované informatiky  
*Akademický rok:* **2018/2019**

### Zásady pro vypracování:

1. Proveďte rešerši existujících softwarových řešení včetně cenové dostupnosti s důrazem na koncovou klientelu.
2. Navrhněte webovou aplikaci pro evidenci osobních vozidel klienta, jeho technických parametrů a historie oprav včetně související dokumentace.
3. Navrhněte mechanismus sdílení získaných dat mezi jednotlivými klienty a ověřte případnou možnost propojení s registry typu CEBIA.
4. Navržené řešení implementujte a pokuste se získat data z testovacího provozu včetně zpětné vazby klientů.

*Rozsah grafických prací:* dle potřeby  
*Rozsah pracovní zprávy:* 30 – 40 stran  
*Forma zpracování práce:* tištěná/elektronická



**Seznam odborné literatury:**

- 1] RESIG, John. JavaScript a Ajax: moderní programování webových aplikací. Brno: Computer Press, 2007. ISBN 9788025118245.  
[2] PROCHÁZKA, David. PHP 6: začínáme programovat. Praha: Grada, 2012. Průvodce (Grada). ISBN 978-80-2-7-3899-4.

*Vedoucí práce:* Ing. Igor Kopetschke  
Ústav nových technologií a aplikované informatiky  
*Datum zadání práce:* 18. října 2018  
*Předpokládaný termín odevzdání:* 30. dubna 2019

L. S.

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

Ing. Josef Novák, Ph.D.  
vedoucí ústavu

V Liberci 18. října 2018

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

## Poděkování

Děkuji Ing. Igorovi Kopetschke za odborné vedení, poskytování rad a pomoci při zpracování mé bakalářské práce.

## Abstrakt

Bakalářská práce se zabývá návrhem a tvorbou webové aplikace sloužící k evidenci osobních vozidel uživatelů, jejich základních technických parametrů, a především všech provedených servisních zásahů a vlastních oprav včetně podrobnějšího popisu použitých dílů a fotodokumentace. Tato aplikace dále umožní potenciálnímu kupci zjistit jednoduchým způsobem historii vozidla bez nutnosti vzájemné obsáhlé komunikace, a zároveň uživateli slouží jako podrobný deník vozidla. V práci má autor navíc za úkol analyzovat existující softwarové produkty pro evidenci automobilů a servisních zásahů, návrh a implementaci systému pro převod vozidla na nového majitele a také ověření možnosti propojení aplikace s existujícími záznamy vozidel v jiné aplikaci.

Na začátku této práce se zabývám analýzou čtyř existujících softwarových produktů pro evidenci servisních zásahů vozidel, kdy každá z nich je jiného druhu. Poté jsou definovány základní pojmy týkající se webových aplikací včetně programovacích jazyků HTML, CSS, PHP, JavaScript a technologie AJAX použitých při tvorbě webové aplikace. V další kapitole se zabývám návrhem databáze, layoutem a návrhem funkcí. Návrh databáze je znázorněn vizuálně pomocí tabulek, relací a atributů. Rozložení stránky je zobrazeno na nákresu s popisem jednotlivých oblastí. Následuje návrh funkcí, kterými bude webová aplikace disponovat a návrh mechanismu pro převod vozidel mezi uživateli. Poté dochází k samotné tvorbě webové aplikace, kdy v práci popisuji vytvořený layout a vytvořené funkce včetně finálního vzhledu zobrazeného na obrázcích. V závěru práce ověřuji možnost propojení aplikace s existujícími záznamy vozidel jiné aplikace a vyhodnocuji zpětnou vazbu.

Na konci práce jsem vytvořil webovou aplikaci, která je funkční a splňuje veškeré zadané požadavky.

### **Klíčová slova:**

Webová aplikace, evidence servisních zásahů, evidence automobilů, analýza softwarových produktů, vyhledávání vozidla podle VIN

## Abstract

The bachelor thesis deals with the design and creation of a web application used for storing records of personal vehicles, their basic technical parameters, and all performed service interventions including more detailed description of used parts and photo documentation. Furthermore, this application will allow the prospective buyer to find out in a simple way the history of the vehicle without the need of a comprehensive communication, while serving the user as a detailed vehicle log. In addition, the author's task is to analyze existing software products for recording vehicles and service interventions, design and implement a system for converting a car to a new owner and also to verify the possibility of connection between the application and application with existing vehicle records.

At the beginning of this work I analyze four existing software products for recording service interventions of vehicles, each of them is of a different kind. Then there are defined basic terms related to web applications including programming languages HTML, CSS, PHP, JavaScript and AJAX technology used in web application creation. In the next chapter I deal with design of database, layout and design of functions. Database design is visualized using tables, relations, and attributes. The page layout is shown in the drawing with a description of each area. The following is a suggestion of the functions that a web application will have and a mechanism for transferring vehicles between users. Then the web application is created, where I describe the created layout and created functions including the final appearance shown in the pictures. In conclusion, I verify the possibility of connecting the application to the existing vehicle records of another application and evaluating feedback.

At the end of my work I created a web application that is functional and meets all the specified requirements.

### **Keywords:**

Web application, recording vehicle maintenance, car registration, software product analysis, vehicle search by VIN

## Obsah

Úvod .....	12
Cíl práce.....	13
1 Analýza softwarových produktů pro evidenci servisních zásahů vozidel .....	14
1.1 AdmWin + modul Autoservis .....	14
1.2 Carsys .....	15
1.3 Fuelly .....	17
1.4 Motortab .....	18
1.5 Závěrečné zhodnocení analýzy .....	20
2 Základní definice .....	21
2.1 Webová aplikace .....	21
2.2 HTTP .....	21
2.2.1 HTTP cookies .....	21
2.2.2 SESSION.....	21
2.3 HTTPS + SSL.....	22
2.3.1 SSL.....	22
2.4 Využité programovací jazyky .....	22
2.4.1 HTML.....	22
2.4.2 CSS.....	23
2.4.3 PHP .....	23
2.4.4 JavaScript.....	23
2.4.5 AJAX.....	23
3 Návrh aplikace a její realizace .....	24
3.1 Postup tvorby webové aplikace .....	24
3.1.1 Webhosting .....	24
3.1.2 Databáze .....	25
3.1.3 Layout.....	25
3.2 Návrh webové aplikace .....	25
3.2.1 Webhosting .....	25
3.2.2 Databáze .....	26
3.2.3 Návrh layoutu .....	27
3.2.4 Návrh funkcí aplikace .....	28
3.2.5 Návrh převodu získaných dat mezi uživateli.....	28
3.3 Tvorba webové aplikace.....	30



3.3.1	Layout.....	30
3.3.2	Pop-up okna.....	30
3.3.3	Vyhledávání vozidla podle VIN: .....	31
3.3.4	Registrace .....	31
3.3.5	Přihlašování .....	32
3.3.6	Přidání nového vozidla.....	32
3.3.7	Výběr vozidla a evidence servisů .....	33
3.3.8	Administrace.....	36
3.3.9	Stránka Můj účet .....	37
3.3.10	Notifikace .....	38
3.3.11	Diskuze.....	39
3.4	Propojení s registry jiné aplikace .....	40
3.4.1	API .....	40
3.4.2	Existující řešení.....	40
3.5	Zabezpečení webové aplikace .....	41
3.6	Zpětná vazba z testovacího provozu .....	41
	Závěr.....	43
	Seznam použité literatury .....	44
	Přílohy .....	47
	Tvorba databázových tabulek .....	47

## Seznam obrázků

Obrázek 1: AdmWin přehled předchozích oprav[1] .....	15
Obrázek 2: CarSystem ukázka[2].....	16
Obrázek 3: Fuelly vytvoření a výpis servisních zásahů .....	17
Obrázek 4: Motortab vytvoření servisního zásahu a přehled již provedených.....	19
Obrázek 5: Princip komunikace mezi serverem a klientem .....	24
Obrázek 6: Návrh databáze .....	26
Obrázek 7: Návrh layoutu.....	27
Obrázek 8: Úvodní strana .....	30
Obrázek 9: Vyhledat vozidlo podle VIN .....	31
Obrázek 10: Registrace.....	32
Obrázek 11: Ukázka přihlášení .....	32
Obrázek 12: Přidání nového automobilu .....	33
Obrázek 13: Výběr vozidla .....	34
Obrázek 14: Výpis servisních zásahů.....	34
Obrázek 15: Přidání nového servisního zásahu .....	35
Obrázek 16: Správa servisních zásahů .....	36
Obrázek 17: Úprava servisního zásahu .....	37
Obrázek 18: Stránka můj účet.....	38
Obrázek 19: Notifikace.....	39
Obrázek 20: Diskuze .....	39
Obrázek 21: Příklad API dotazu v PHP .....	41

## Seznam zkratek

AJAX	Asynchronous JavaScript and XML
CSS	Cascading Style Sheets
FTP	File Transfer Protocol
HTML	HyperText Markup Language, značkovací jazyk pro tvorbu hypertextových dokumentů
ID	Identifikátor (Identifikační číslo)
PHP	Hypertext Preprocessor
SPZ	Státní poznávací značka
SQL	Structured Query Language
VIN	Vehicle identification number
API	Application Programming Interface
USA	United States of America, Spojené státy americké
SSL	Secure Sockets Layer

## Úvod

Pro svou bakalářskou práci jsem si vybral téma „Webová aplikace pro evidenci historie servisních zásahů osobních vozidel“.

Automobil je věc, bez které by se v dnešní době spousta lidí neobešlo. Každý den automobily v naší zemi přepravují miliony osob, potravin, materiálu a ostatních věcí, které je potřeba dopravit z jednoho místa na druhé. V naší zemi je registrováno téměř 5,7 milionu osobních automobilů, které člověk používá pro přepravu do práce, na nákupy, za rodinou, na výlety a podobně, a proto je v našem zájmu je neustále udržovat v provozuschopném stavu. Proto je dobré mít přehled o tom, co se na automobilu provedlo za servisní úkony, kdy se provedly, při jakém nájezdu kilometrů, jaké byly použity náhradní díly a jaké byly náklady na jednotlivé servisní úkony. Pro lidi, kteří dávají svá vozidla do autoservisů, kde se o jejich vozidlo postarají není důležité si servisní zásahy nějak evidovat, protože většina autoservisů disponuje softwarovým produktem, ve kterém jsou všechny úkony evidovány. Někteří lidé svěřují svá auta do rukou živnostníků, a naopak někteří lidé jsou šikovní a zvládnou si některé servisní úkony provést doma. Pro tuto skupinu lidí je těžké si servisní úkony provedené na jejich vozidlech nějakým způsobem evidovat.[6]

Evidence servisních zásahů je přínosná ve všech směrech. Lze ji využít také při prodeji automobilu, kdy evidované záznamy servisních úkonů dokáží zvýšit prodejní cenu automobilu a pomůže novému majiteli s plánováním dalších pravidelných servisních zásahů (např. výměna olejů a kapalin, výměna filtrů, spojky atd.).

Proto mě napadlo vytvořit přehlednou webovou aplikaci, ke které by se lidé mohli připojit přes internet odkudkoliv, a ve které by si mohli jednoduše vytvářet a číst záznamy servisních zásahů jejich vozidel. Záznamy, které by si mohli poté upravovat a ke kterým by mohli přidávat fotodokumentaci.

## **Cíl práce**

Cílem práce je vytvořit fungující webovou aplikaci pro evidenci vozidel a na nich provedených servisních zásahů. Aplikace bude přístupná pomocí webového prohlížeče, ve kterém si klient po vytvoření uživatelského účtu a následném přihlášení bude moci přidávat vozidla a provedené servisních zásahy. Tyto záznamy uživatel bude moci upravovat a mazat. Při vytvoření nového servisního zásahu uživatel bude moci přiložit k záznamu fotodokumentaci. Aplikace bude umožňovat vyhledávání vozidel podle VIN, výpis základních informací o automobilu včetně stručného výpisu provedených servisních zásahů. Pro převod automobilu na nového majitele bude navržen bezpečný mechanismus. Dalším cílem práce bylo analyzovat existující softwarové produkty s ohledem na jejich cenovou dostupnost a ověřit možnost propojení této aplikace se záznamy vozidel jiné aplikace.

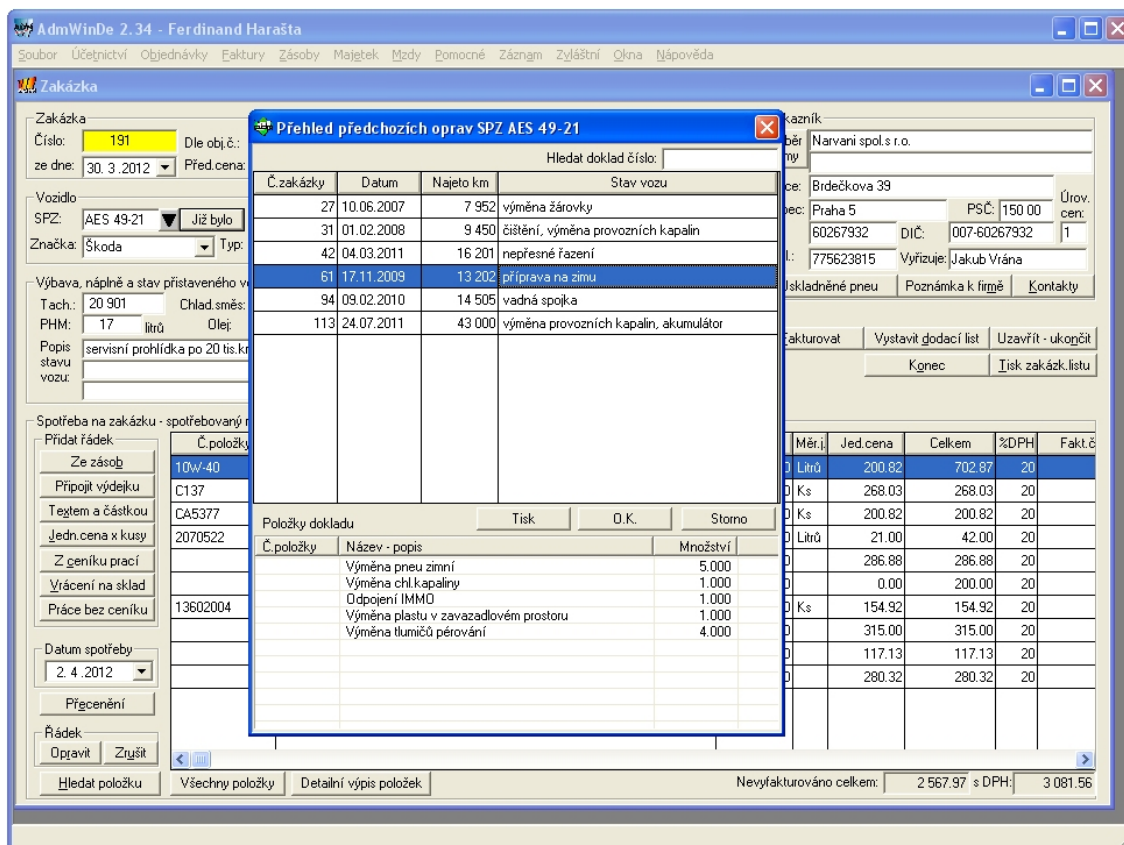
# 1 Analýza softwarových produktů pro evidenci servisních zásahů vozidel

V této kapitole budou představeny čtyři příklady aplikací pro evidenci servisních zásahů vozidel. Každá z nich je jiného druhu, proto bude popsáno, co tyto aplikace nabízí klientům včetně cenové dostupnosti a obrázků pro představu, jak rozhraní jednotlivých aplikací vypadá.

## 1.1 AdmWin + modul Autoservis

AdmWin je účetní software určený pro firmy, který disponuje různými moduly pro různá odvětví firem. Pro nás důležitý je modul Autoservis, který je zaměřený pro menší i velké autoservisy. Spojením ekonomického softwaru s modulem pro evidenci servisních zakázek vznikne ideální software pro vedení účetnictví, zakázek a servisních úkonů provedených na vozidlech zákazníků. [1]

Modul autoservis nabízí upravenou formu zakázkových listů a zpracování zakázek vhodné pro autoservisy. Nabízí také široké spektrum evidování záznamů jako například evidenci zakázek, u které se evidují požadavky zákazníka, datum a čas přistavení vozidla a předpokládaný termín dokončení servisních úkonů, náhradní díly, spotřebovaný materiál, vykonanou práci a výpis přehledu předchozích zakázek provedených na vozidle. K zakázkám lze přikládat fotodokumentaci či oskenované dokumenty. Dále umožňuje evidovat vozidla, u kterých se eviduje jejich SPZ, zákazník, výrobce, model, VIN, druh vozidla, druh karoserie, rok výroby, výbava, barva, datum konce STK, typ a stav pneumatik a údaje o motoru. Tento modul také disponuje evidencí uskladněných pneumatik, notifikacemi blížícího se ukončení STK a speciálními výpisy a přehledy zakázek.[1]



Obrázek 1: AdmWin přehled předchozích oprav[1]

Tento obrázek zobrazuje okno s přehledem oprav, které se provedly na určitém vozidle. V pozadí jdou na pravé straně vidět informace o zákazníkovi a na levé straně číslo zakázky a informace o aktuálně vybraném vozidle.

## Požizovací náklady

Modul Autoservis je nabízen včetně základního účetního systému AdmWin pro daňovou evidenci za 6500 Kč za licenci pro jeden počítač.[1]

## 1.2 Carsys

Jedná se o informační systém, který se dělí na dvě verze. Nám postačí ta základní s názvem Easy servis, která je určená pro menší servisy. Tato verze systému Carsys je ideálním softwarovým produktem pro menší servisy, který disponuje jednoduchou evidencí a fakturací servisních zásahů vozidel.[2]

Verze Easy servis se skládá ze čtyř modulů. První z nich je modul Autoservis, který eviduje údaje o zákazníkovi, jeho vozidle a stavu vozidla při převzetí včetně požadavků na opravu, naplánování termínu opravy a odhadované ceny. Disponuje

také fakturační položek. Druhým je modul Vozidla. V tomto modulu se evidují údaje jednotlivých vozidel včetně údajů o majiteli či uživateli vozidla a všech realizovaných servisních zásazích vozidla. Třetím modulem je Adresář, který eviduje veškeré údaje o zákaznících spolu s informacemi o dokladech zákazníka a jeho vozového parku. Posledním modulem je modul Katalog/Sklad. Tento modul disponuje katalogem, ve kterém si uživatelé tohoto softwaru ukládají často se opakující položky zakázek jako je například materiál, práce a subdodávky. Modul obsahuje také sklad, který slouží pro správu skladového hospodářství firmy. [2]

The screenshot shows a window titled "Servisní zakázka ZAK-20080027". At the top, there are tabs for "Obecné", "Zadání oprav", "Položky", and "Deník". Below the tabs, there are fields for "Číslo zakázky: ZAK-20080027", "Kniha: Servisní zakázky", "Rok: 2008", and "Stav: Nezahájeno". There are also dropdown menus for "Datum založení: 25.6.2008" and "Datum fakturace: 5.8.2008".

The form is divided into two main sections: "Zákazník" and "Vozidlo".

**Zákazník:**

- Firma: New Service
- Název firmy: New Service
- Adresa: Ulice 1234, 599 00 Město nad Řekou
- Kontakt: 112 221 333
- Zástupce: [empty]
- IČO: 25051857
- Kontakt: [empty]
- DIČ: CZ25051857

**Vozidlo:**

- VIN interní: 4A3 5678
- SPZ: 4A3 5678
- Rok výr. / up.: 1997
- Kód modelu: [empty]
- Verze: [empty]
- Značka: VW
- Karosérie: [empty]
- Model: Golf
- Motor: 1,6 tdi
- Rok: [empty]
- Kritérium 1: [empty]
- Stav tachometru: 0
- Stav nádrže: [empty]

Below the form, there is a section "Ostatní zakázky vozidla" with a table:

	Číslo zakázky	Stav	Datum	Stav tach.	Rozdíl km	Celkem bez DPH	Celkem s DPH	Zaměstnanec zapsal	Zarr
▶	ZAK-20080020	Nezahájeno	20.5.2008	0	-160000	630,20	750,00	sa	
▶	ZAK-20080019	Dokončeno	20.5.2008	160000	160000	1 452,10	1 728,00	sa	

Obrázek 2: CarSystem ukázka[2]

Na tomto obrázku lze vidět obecné informace servisní zakázky, což jsou informace o zákazníkovi, jeho vozidle, a ostatních servisních zakázkách tohoto vozidla.

### Pořizovací náklady

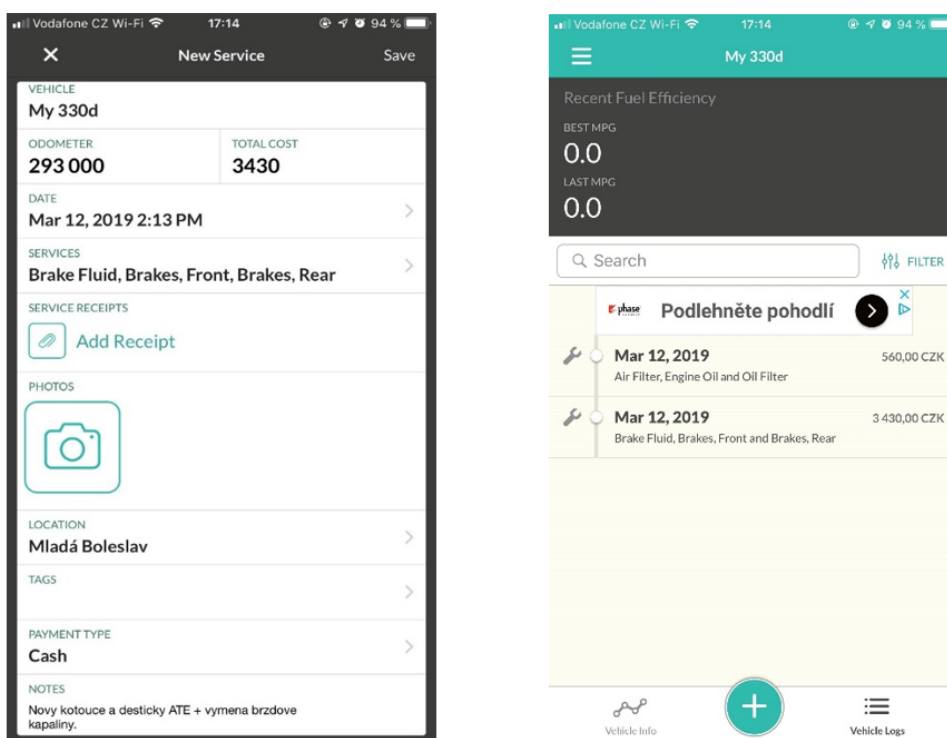
Licence Small dostačující pro malý autoservis stojí 7300 Kč. Jedná se o plovoucí licenci, kterou lze spouštět na libovolném počítači, ale v jeden moment smí být spuštěna pouze na jednom.[3]



### 1.3 Fuely

Jedná se o mobilní aplikaci v anglické jazyce, která slouží k měření spotřeby pohonných hmot a evidenci servisních zásahů automobilů, motocyklů a těžké techniky. Lze se do ní přihlásit i pomocí webového prohlížeče, a zde porovnávat své náklady za pohonné hmoty a servisní úkony s vozidly ostatních uživatelů. Tato aplikace je určena spíše pro fyzické osoby, které si chtějí počítat průměrnou spotřebu, stručně evidovat servisní zásahy a náklady svých vozidel.[4]

Fuely nabízí evidenci vozidel uživatele která obsahuje rok výroby vozidla, jeho značku a model, motorizaci, VIN, fotografii, druh převodovky, SPZ, pojištění, velikost pneumatik, stát, ve kterém je vozidlo přihlášeno, počet najetých kilometrů včetně poznámky k vozidlu. Po vytvoření vozidla lze evidovat servisní zásahy na něm provedené, u kterých se zadává druh servisního úkonu, počet najetých kilometrů, datum provedení zásahu, celkové náklady zásahu, lze přiložit dokumentaci či fotografii, místo provedení zásahu, druh platby a poznámku k provedenému servisnímu zásahu. Tato mobilní aplikace také obsahuje funkci připomínky, kdy po výběru druhu, termínu a frekvence uživatele upozorní na blížící se termín uživatelem vytvořené události. [4]



Obrázek 3: Fuely vytvoření a výpis servisních zásahů

## Požizovací náklady

Aplikaci lze využívat zdarma včetně reklam nebo za 29 Kč měsíčně bez reklam, včetně prémiových doplňků. [4]

### 1.4 Motortab

Motortab je webová aplikace v anglickém jazyce pocházející z USA, určená pro evidenci servisních zásahů osobních vozidel. Aplikace tohoto druhu v České republice pravděpodobně neexistuje, jelikož jsem po důkladném prozkoumání internetu žádnou nenašel. Jedná se tudíž o druh aplikace, která je předmětem této bakalářské práce a kterou se chystám vytvořit. Tato aplikace se na úvodní stránce pyšní nápisem, že se jedná o nejlepší webovou aplikaci pro údržbu vozidel na internetu. [5]

V této aplikaci je přihlášenému uživateli umožněno evidovat svá vozidla zapsáním jména vozidla, roku výroby, značky a modelu, státu registrace, doby zakoupení vozidla, počtu najetých kilometrů a lze přidat obrázek vozidla. Poté uživatel může evidovat servisní zásahy provedené na některém z jeho vozidel. Při vytvoření nového servisního zásahu uživatel nejprve vybírá název vozidla, na kterém byl servisní zásah proveden, poté uživatel zadává druh servisního zásahu, datum, cenu, počet najetých kilometrů, poznámku k zásahu a může přiložit dokumentaci. Aplikace disponuje výpisem provedených servisních zásahů vozidla, přehledem zajímavostí jako například průměrný počet najetých kilometrů za měsíc nebo celkové náklady servisních zásahů vozidla a nabízí také upozornění pomocí emailové zprávy o blížícím se potřebu výměny dílu či konce platnosti STK. [5]

V této aplikaci jsou také nedostatky kdy jedním z nich je nemožnost odstranění vozidla. V případě prodeje vozidla bude uživateli vozidlo překážet ve virtuální garáži aplikace a bude mu zabírat zbytečně místo, jelikož místa v garáži jsou placená. Nedostatkem pro českého uživatele může být rozhraní pouze v anglickém jazyce a náklady servisních zásahů zapisovány v amerických dolarech. [5]

The screenshot displays the Motortab web application interface. At the top, there is a navigation bar with tabs for 'Vehicles', 'Maintenance', 'Dashboard', 'Trends', and 'Account'. The 'Maintenance' tab is active, showing a form to 'Add new maintenance item in the fields below:'. The form includes fields for vehicle selection (Moje auto), service type (oil change), date (03/20/2019), cost (\$50.00), mileage (302,000), and notes (Olej Castrol 5 litrů). There are 'Upload Doc' and 'Save Item' buttons. Below the form is a table of existing maintenance records:

Vehicle	Service	Date	Cost	Mileage	Notes	Upload	Edit
Moje auto	clutch	03/13/19	\$510	312,000	Nova spojka SACHS.	<input checked="" type="checkbox"/>	
Moje auto	air filter	03/03/19	\$120	310,000	Vymena vzduchoveho filtru.	<input checked="" type="checkbox"/>	

On the left, there is a 'Vehicles' sidebar with a 'Moje auto' section showing a car icon and 'Mileage: 312,000'. Below it is an 'Upgrade' button and a list of benefits: 'Pay thru PayPal', 'Flexible options', and 'Annual subscriptions'. On the right, there is a 'How To' sidebar with a search icon and links for 'Search for Maintenance Records', 'Customize Email Reminder', 'Settings', and 'Send Maintenance Reminders to Additional Emails'.

Obrázek 4: Motortab vytvoření servisního zásahu a přehled již provedených

## Požizovací náklady

Zdarma uživatel může přidat jedno vozidlo do garáže, přičemž se mu budou zobrazovat reklamy, nebude moci ukládat dokumenty a exportovat data. Při zaplacení částky 25\$ (567 Kč) uživatel může mít v garáži až 10 vozidel, bude moci ukládat dokumenty a exportovat data a využívat aplikaci bez reklam. [5]

## 1.5 Závěrečné zhodnocení analýzy

V této kapitole byly popsány čtyři softwarové produkty určené pro evidování servisních zásahů vozidel. Každý z produktů je jiného typu a každý má své výhody a nevýhody. Zatímco ekonomický software AdmWin s modulem Autoservis a informační systém Carsys jsou ideálními softwarovými produkty pro autoservisy, tak na druhou stranu jsou zbytečné pro fyzické osoby, které si chtějí pouze evidovat provedené servisní zásahy. Obsahují velké množství možností, kterými lze obohatit záznamy a disponují mnoha užitečnými funkcemi pro autoservisy. Naopak webová aplikace Motortab a mobilní aplikace Fueelly jsou vytvořeny pro fyzické osoby, a také jsou k tomu přizpůsobeny. Mají přehledné rozhraní a disponují vytvářením stručných záznamů servisních zásahů. Největší nevýhodou aplikací Motortab a Fueelly je absence českého jazyka, ale bohužel aplikace tohoto typu v českém jazyce nejsou k dispozici.

## **2 Základní definice**

### **2.1 Webová aplikace**

Jedná se o druh aplikace, kterou lze spustit přes webový prohlížeč na jakémkoliv zařízení bez nutnosti instalace aplikace. Tato aplikace totiž běží na webovém serveru, a uživatel si jí může zobrazit přes webový prohlížeč. Na první pohled může webová aplikace vypadat jako webová stránka, ale není tomu tak. Webová aplikace totiž využívá databázi, provádí složitější úlohy a vesměs umí provádět to, co nainstalovaná aplikace v počítači. [7]

### **2.2 HTTP**

Jedná se o aplikační protokol umožňující zobrazování webových stránek. Zajišťuje přenášení objektů (obrázky, stránky atd.) mezi webovým serverem a internetovým prohlížečem. HTTP je bezstavový protokol, který nerozlišuje to, od kterého klienta požadavek přichází, tudíž odpoví každému požadavku. Při tvorbě webové aplikace tato vlastnost je nežádoucí, proto je potřeba jí obejít pomocí různých metod (např. Cookies).[8]

#### **2.2.1 HTTP cookies**

Takto jsou označována data vytvořená webovým serverem, která se ukládají do počítače uživatele prostřednictvím webového prohlížeče. Ukládaná data obsahují informace o provedených krocích uživatele, a v momentě kdy uživatel znovu navštíví webovou aplikaci, webový prohlížeč odešle cookies zpět na server a ten opět získá uložená data o uživateli.[9]

Cookies ve webových aplikacích slouží k rozpoznávání jednotlivých uživatelů a následnému ukládání konkrétních údajů o něm. Dále se využívají pro ukládání identifikátorů či ukládání potvrzení, že je uživatel přihlášený. [9]

#### **2.2.2 SESSION**

Jedná se o technologii používanou ve webových aplikacích, která umožňuje ukládat uživatelská data během komunikace se serverem. Systém přidělí uživateli unikátní identifikační číslo, které se označuje jako session id, a to je uloženo jako cookie

v počítači uživatele. K tomuto identifikátoru jsou poté přidělena data spojená s uživatelem, která jsou uložena v paměti serveru.[9]

Session uchovává data dočasně, k jejich zániku dojde v momentě vypnutí webového prohlížeče nebo manuálním smazáním cookies v prohlížeči, přičemž dojde ke ztrátě session id. Také lze session ukončit příkazem v aplikaci (využívá se při odhlašování uživatele) nebo jí příkazem přidělit dobu po které se neaktivní komunikace ukončí.

## **2.3 HTTPS + SSL**

HTTPS je zkratka z anglického názvu HyperText Transfer Protokol Secure. Jedná se o šifrovanou variantu protokolu HTTP určenou pro bezpečnou komunikaci v počítačové síti. Tento protokol zajišťuje šifrování přenášené komunikace algoritmem SSL mezi webovým prohlížečem a webovým serverem, a díky tomu zajišťuje chráněný přístup k webovému serveru. Protokol HTTPS je důležitý pro bezpečné přenášení citlivých informací (telefonní čísla, čísla kreditních karet, hesla atd.) a pro autorizaci přístupu k webovým aplikacím.[11]

### **2.3.1 SSL**

O bezpečný přenos dat mezi klientem a serverem se stará SSL protokol, který zaručuje pravou identitu klienta a serveru, aby nedošlo ke krádeži identity. Dále protokol zajišťuje šifrování komunikace mezi klientem a serverem, kdy zařídí vše okolo šifrovacího algoritmu, tvorby společného tajemství a bezpečné výměny klíčů.[12]

O potvrzení pravosti serveru a jeho provozovatele se starají SSL certifikáty které jsou důležitým prvkem bezpečné komunikace. Lze pomocí nich také identifikovat klienta vůči serveru.[13]

## **2.4 Využití programovací jazyky**

### **2.4.1 HTML**

HTML je textový značkovací jazyk, který se využívá k vytváření objektů a struktury webových stránek. Tímto jazykem popisujeme pomocí párových tagů (značek) jaký text se stane nadpisem, odstavcem, odkazem nebo jiným objektem.

Tagy se ve většině případů nacházejí ve dvojici, kdy první tag otevírá objekt a druhý tag objekt zavírá.[20]

### **2.4.2 CSS**

CSS je zkratka z anglického Cascading Style Sheets, česky kaskádové styly. Jde o moderní stolovací jazyk, který popisuje vizuální způsob zobrazení dokumentů psaných ve strukturálních jazycích HTML, XHTML a XML. CSS definuje vlastnosti objektů jako jsou barva, písmo, umístění a jejich další vizuální vlastnosti.[19]

### **2.4.3 PHP**

Jedná se o jeden z nejvíce rozšířených programovacích jazyků určených pro vytváření webových aplikací. PHP umí pracovat se soubory webové aplikace a s databázemi, umí upravovat a vytvářet grafiku, odesílat emaily, vytvářet export dat a mnoho dalšího. Tento jazyk pracuje na straně serveru, tudíž PHP skript se provede na serveru, který poté prohlížeči odešle výsledek ke zobrazení.[18]

### **2.4.4 JavaScript**

*„JavaScript je objektově orientovaný programovací jazyk, využívaný při tvorbě webových stránek. Na rozdíl od serverových programovacích jazyků (například PHP) sloužících ke generování kódu samotné stránky, JavaScript běží na straně klienta, tedy v prohlížeči až po stažení do vašeho počítače.“ [16]*

JavaScript vytváří interaktivní webové stránky, a tím pomáhá šetřit místo na stránce zobrazováním skrytých formulářů. Také dokáže měnit obsah nebo vizuální podobu webových stránek. [16]

### **2.4.5 AJAX**

Zkratka AJAX pochází z anglického Asynchronous JavaScript and XML. AJAX je moderní technologií která se často využívá ve webových aplikacích. Aplikace díky této technologii dokáže odeslat data a jiná data získat zpět ze serveru, aniž by bylo potřeba znovu načíst celou stránku.[17]

## 3 Návrh aplikace a její realizace

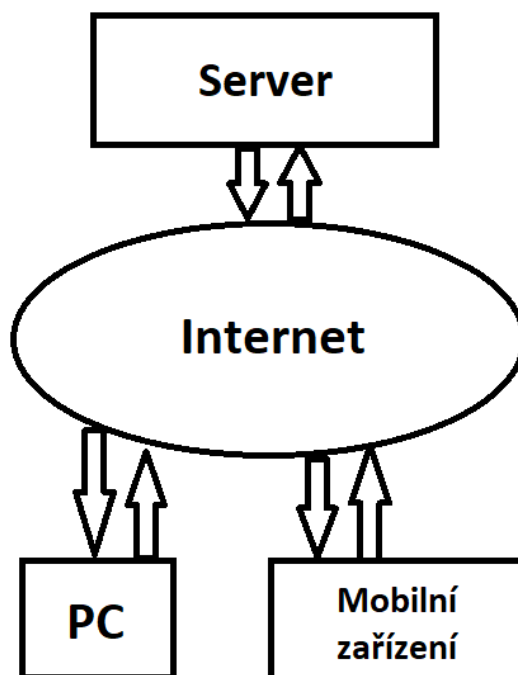
### 3.1 Postup tvorby webové aplikace

V této části práce jsou popsány potřebné úkony, které je nutné podniknout k tvorbě fungující webové aplikace.

#### 3.1.1 Webhosting

Před samotnou tvorbou webové aplikace je potřeba vybrat poskytovatele webhostingu, který pronajme část prostoru na svém serveru kde bude aplikace uložena a spuštěna. Poskytovatelé webhostingu mají ve většině případů zřízenou webovou aplikaci pro klienty, kteří v ní mohou vytvářet domény, ukládat soubory na server, vytvářet databáze a mnoho dalšího. [24]

Aby si webovou aplikaci mohli uživatelé zobrazit, je potřeba mít zaregistrovanou doménu, díky které se po zadání názvu domény do vyhledávače aplikace zobrazí uživateli.



Obrázek 5: Princip komunikace mezi serverem a klientem

Princip komunikace mezi serverem a klientem funguje tak, že nějaké zařízení pomocí webového prohlížeče vykoná žádost o zobrazení webové aplikace, tato žádost koluje internetem až dokud nedojde k serveru, ve kterém je webová aplikace



spuštěna. Server poté odpoví uživateli zobrazením webové aplikace ve webovém prohlížeči.

### **3.1.2 Databáze**

Jedná se o systém řízení báze dat, který je základním pilířem každé webové aplikace. Slouží totiž k ukládání dat. Proto je třeba si pro svou webovou aplikaci databázi vytvořit, například pomocí webové aplikace webhostingu. Pro webové aplikace se nejčastěji využívá databáze typu MySQL, ve které komunikace mezi serverem a databází probíhá pomocí jazyka SQL.[21]

### **3.1.3 Layout**

*„Layout je rozvržením stránky, udržuje její logickou organizaci nejen pro počítač, ale hlavně pro uživatele, který vnímá jednotlivé části stránky jako logické celky (například logo, navigační menu, vlastní obsah, zápatí a další). Každý tento celek by měl mít svou stálou pozici, aby jej mohl uživatel rychle a bez problémů najít očima, kdykoli to potřebuje.“[22]*

Tvorba layoutu probíhá tvorbou objektů pomocí jazyka HTML. Těmto objektům se poté pomocí kaskádových stylů (CSS) přiřadí grafické vlastnosti tak, aby vytvořili přehledný a hezky vypadající layout webové aplikace.

## **3.2 Návrh webové aplikace**

Dá se říci, že webová aplikace je množina skriptů běžících na straně serveru. Programátor má za úkol tyto skripty vytvářet pomocí jazyka PHP a JavaScript. V těchto skriptech se pracuje s tokem dat pomocí příkazů SQL, s objekty, emaily, soubory a dá se říci, že se vším, na co si člověk při tvorbě aplikace vzpomene. Po naprogramování každého skriptu je třeba vždy otestovat funkčnost a popřípadě opravit vzniklé chyby ve skriptu.[7]

### **3.2.1 Webhosting**

Díky snaze o minimalizaci nákladů při tvorbě této bakalářské práce bude zřízen free webhosting. Tudíž bude vybrán poskytovatel webhostingu, který nabídne pronájem místa na serveru včetně domény zdarma. Při výběru mezi nabízenými webhostingy

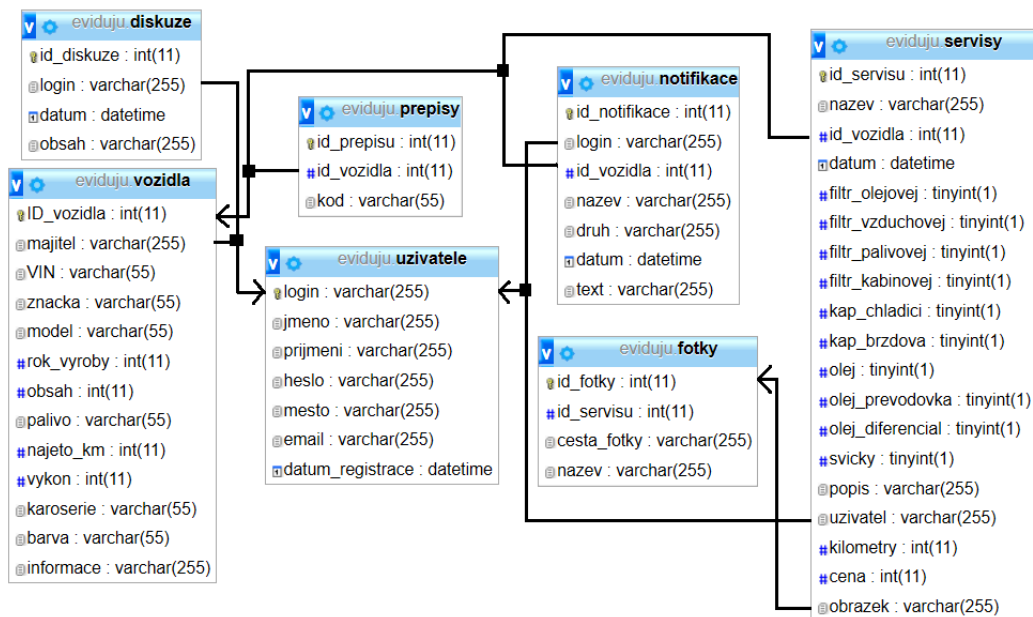
zdarma je důležité zkontrolovat nabízené technické parametry, které budou splňovat požadavky pro tvorbu této webové aplikace.

Pro naši webovou aplikaci byl vybrán webhosting Endora od společnosti Stable.cz. Ve webové administraci byla vytvořena doména s názvem „www.eviduju-servisy.mzf.cz“, díky které se do webové aplikace bude moci klient připojit z webového prohlížeče.[14]

### 3.2.2 Databáze

Před tvorbou databáze je důležité si databázi graficky navrhnout, promyslet si jaké tabulky bude obsahovat a jaké v nich budou atributy. U atributů je potřeba se zamyslet nad tím, jaký budou mít datový typ a jaké vlastnosti. Některé tabulky budou propojeny relacemi, tudíž bude potřeba určit primární a cizí klíče a promyslet jejich typy relací.

Databáze byla vytvořena ve webové administraci webhostingu Endora. Vytvořená databáze je typu MySQL, do které se lze přihlásit pomocí softwarového nástroje phpMyAdmin, ve kterém lze spravovat celou databázi. Pomocí PHP skriptu a databázových dotazů SQL jsou vytvořeny tabulky včetně atributů (viz příloha). Poté je databáze připravena pro příjem dat z webové aplikace.



Obrázek 6: Návrh databáze

### 3.2.3 Návrh layoutu

Tvorbou layoutu programátor určí budoucí vzhled webové aplikace. Proto před jeho tvorbou je potřeba si představit, jak by webová aplikace měla být rozložena a jak by měla vypadat. Proto jsem nakreslil jednoduchý návrh layoutu, podle kterého bude layout aplikace tvořen.

<b>Logo + jméno webové aplikace</b>		<b>Přihlašování</b>
<b>Navigační menu</b>		
<b>Výpis evidovaných servisních zásahů vozidla</b>	<b>Obsah stránek, podrobný výpis servisních zásahů, administrace, diskuze atd.</b>	

Obrázek 7: Návrh layoutu

Na nákresu lze vidět rozložení stránky, kde v levém horním rohu se bude nacházet oblast určená pro logo a název aplikace, které by mohli v budoucnu sloužit k propagačním účelům. V pravém horním rohu bude vytvořena oblast pro formulář sloužící k přihlašování uživatelů, kdy po přihlášení se zde objeví výpis informací o uživateli. Pod těmito oblastmi se bude nacházet ve vodorovné poloze navigační menu. Pod ním se bude na levé straně nacházet svisle oblast pro výpis názvů všech provedených servisních zásahů vozidla včetně odkazů na podrobné výpisy. Ve zbytku stránky se bude nacházet oblast s obsahem. V něm se budou zobrazovat veškerý důležitý obsah aplikace (podrobné výpisy, administrace, popis aplikace atd.).

### 3.2.4 Návrh funkcí aplikace

Před tvorbou aplikace je nutno si stanovit kterými funkcemi bude aplikace disponovat. V našem případě to budou následující funkce.

**Vyhledávání vozidla podle VIN** – Tuto funkci bude moci využít každý, jak přihlášený, tak i nepřihlášený uživatel. Jedná se o stručný výpis informací o vozidle včetně zjednodušeného výpisu provedených servisních zásahů vozidla, jehož VIN je zadáno.

**Registrace a přihlašování uživatelů** – Každý uživatel webové aplikace musí mít vytvořený účet, který bude sloužit k jeho identifikaci. Přihlášenému uživateli se budou vypisovat pouze data, která jsou s ním spojená nebo s jeho vozidlem.

**Evidence vozidel** – Uživatel bude mít možnost vytvářet, upravovat a mazat vozidla.

**Evidence servisních zásahů** – Uživatel bude moci vytvářet, upravovat a mazat servisní zásahy včetně přiložených obrázků.

**Vybrání vozidla a výpis jeho servisních zásahů** – Uživateli se bude zobrazovat obsah garáže, kde po kliknutí na vybrané vozidlo se zobrazí název a datum provedených servisních zásahů v levém objektu kdy po kliknutí na vybraný servisní záznam se zobrazí jeho podrobný výpis.

**Administrace** – Možnost upravovat vytvořené servisní zásahy, vytvořená vozidla a svůj účet.

**Notifikace** – Přihlášenému uživateli bude umožněno vytvořit notifikaci která ho pomocí emailové zprávy upozorní na blížící se konec platnosti STK, blížící se splatnost pojištění vozidla nebo něčeho jiného.

### 3.2.5 Návrh převodu získaných dat mezi uživateli

V případě prodeje vozidla jednoho z uživatelů zde musí být možnost převést vozidlo včetně všech jeho servisních zásahů na nového majitele. Proto je potřeba vymyslet mechanismus, který musí být bezpečný, aby nedošlo k ukradení záznamů.

Inspirujme se existujícím mechanismem pro obnovu zapomenutého hesla uživatele u aplikací. Ten funguje tak že po kliknutí na tlačítko pro obnovu hesla se odešle

dotyčnému email s kódem, který zadá do aplikace a poté má uživatel možnost si nastavit nové heslo.

Tudíž mechanismus, který bude vytvořen bude fungovat tak, že přihlášený uživatel v sekci „převod vozidla na nového majitele“ vybere vozidlo, které chce převést, a nechá si vygenerovat náhodný kód, který se odešle do jeho osobní emailové schránky. Ten si kód z emailové schránky zkopíruje, a pošle novému majiteli. Ten musí obdržený kód zadat do sekce „převod vozidla na nového majitele“ a poté mu bude automobil přiřazen, a s ním i přístup k jeho veškerým servisním zásahům.

## 3.3 Tvorba webové aplikace

### 3.3.1 Layout

Při vytváření layoutu dochází k tvorbě logických oblastí stránky, kterým přidáváme určité vlastnosti. Při tvorbě layoutu webové stránky se využívají jazyky HTML a CSS. Náš layout obsahuje hlavičku a tělo. Hlavička je rozdělena na banner, přihlašovací formulář a navigační menu. Tělo je rozděleno na postranní oblast pro přehled servisních zásahů a na obsah.[22]



Obrázek 8: Úvodní strana

Pro uživatele mobilních zařízení jsem vytvořil pozměněné rozložení layoutu, aby tato aplikace byla přehledná i na mobilních zařízeních.

### 3.3.2 Pop-up okna

Pro ušetření místa, přehlednost a interaktivitu webové aplikace jsem se rozhodl využít vyskakovacích oken. Jedná se o oblast, která je skrytá, ale kliknutím na určité tlačítko se zobrazí. V této oblasti se budou vyskytovat formuláře pro registraci, vytvoření vozidla, vytvoření nového servisního zásahu, výpis vozidel pro jejich výběr, zobrazení obrázků a výpis notifikací včetně jejich vytvoření. Pro jejich tvorbu jsem využil HTML, PHP, JavaScript a CSS.

### 3.3.3 Vyhledávání vozidla podle VIN:

Tato funkce slouží pro výpis informací o vozidle včetně základních informací o provedených servisních zásazích vozidla. Uživatel zadá VIN vozidla a kliknutím na tlačítko „Vyhledat“ odešle metodou POST zadané VIN vozidla na server, kde PHP skript získá z databáze informace o vozidle a ty následně vypíše uživateli. Vypsání informací o vozidle včetně základních informací o provedených servisních zásazích jsou vypsány do tabulky, ve které jsou přehledně čitelné.

Datum	Název	Servisy: Popis	Stav kilometrů	Cena servisu
04.04.2019	Zhavičky	Žhavičky Bosch vyměněny s relém žhavení, jelikož diagnostika psala chybu žhavení.	312000	1685
04.04.2019	Tlumice	Sachs tlumice, pružiny Elbach pro kit.	316000	16000
11.04.2019	Výměna oleje	Výměna oleje za Castrol FST Titanium, 5 litrů, vzduchový filtr MANN a olejový Bosch.	293000	1685
21.04.2019	Výměna oleje	Castrol zase	200000	1000

Obrázek 9: Vyhledat vozidlo podle VIN

### 3.3.4 Registrace

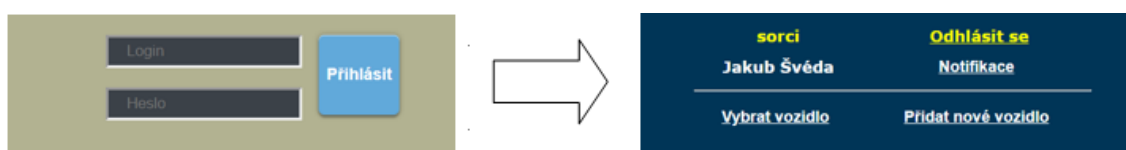
Registrace nového uživatele se provádí kliknutím na tlačítko „Registrovat“ které se nachází v navigačním menu. Po kliknutí se zobrazí vyskakovací okno s formulářem pro registraci uživatele. Po vyplnění potřebných dat a kliknutí na tlačítko „Registrovat“ se odešlou zadané údaje metodou POST na server, kde jsou uloženy v proměnných  $\$_POST['\text{název\_objektu}']$  a poté PHP skript zadané údaje o uživateli uloží do databáze. JavaScript zařídí zobrazení zprávy o úspěšném vytvoření uživatelského účtu.



Obrázek 10: Registrace

### 3.3.5 Přihlašování

V pravém horním rohu aplikace se nachází oblast pro přihlášení uživatele. Po zadání uživatelského jména a hesla se po kliknutí na tlačítko „Přihlásit“ odešle metodou POST požadavek na PHP skript, který zkontroluje, zda nějaký účet s takovým jménem a heslem existuje v databázi. Pokud ano, tak PHP skript spustí SESSION příkazem `session_start()`, díky které můžeme pracovat s daty o která při přecházení mezi stránkami naší webové aplikace nepřijedeme. Data se nám ukládají do SESSION proměnných `$_SESSION['nazev_proměnné']`, které jsou vázány na přihlášeného uživatele. Po přihlášení uživatele se v pravém horním rohu aplikace zobrazí místo formuláře pro přihlášení oblast s vypsáním jménem, příjmením a loginem uživatelského účtu včetně tlačítek pro odhlášení, zobrazení notifikací, výběr vozidla a přidání nového vozidla.



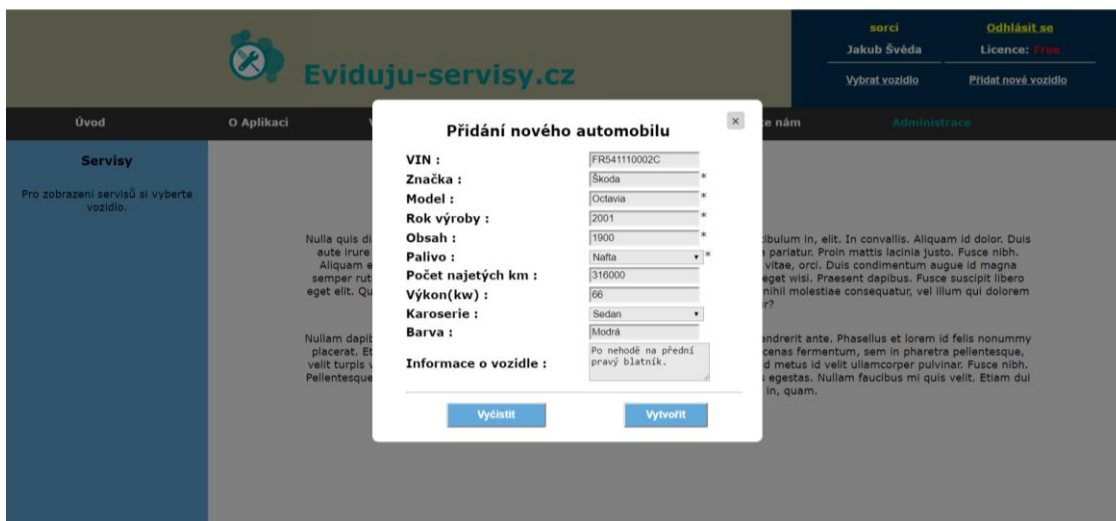
Obrázek 11: Ukázka přihlášení

### 3.3.6 Přidání nového vozidla

Funkce pro přidání nového vozidla je vytvořena obdobně jako registrace uživatele. Po kliknutí na tlačítko „Přidat nové vozidlo“ se zobrazí vyskakovací okno s formulářem pro zadání údajů o vozidle. Kliknutím na tlačítko „Vytvořit“ se metodou POST obnoví stránka a PHP skript se postará o přidání nového vozidla do databáze pomocí dotazu SQL. Majitelem vozidla se stane právě přihlášený uživatel.



Formulář nabízí výběr z přednastavených druhů paliv a typů karoserií. JavaScript zařídí zobrazení zprávy o úspěšném přidání automobilu do databáze.



The screenshot shows a web application interface for 'Eviduju-servisy.cz'. A modal window titled 'Přidání nového automobilu' is open, allowing a user to add a new car. The form contains the following fields and values:

Field	Value
VIN	FRS41110002C
Značka	Škoda
Model	Octavia
Rok výroby	2001
Obsah	1900
Palivo	Nafta
Počet najetých km	310000
Výkon(kw)	66
Karoserie	Sedan
Barva	Modrá

Below the form, there is a note: 'Po nehodě na přední pravý blatník.' At the bottom of the modal, there are two buttons: 'Výčistit' and 'Vytvořit'.

Obrázek 12: Přidání nového automobilu

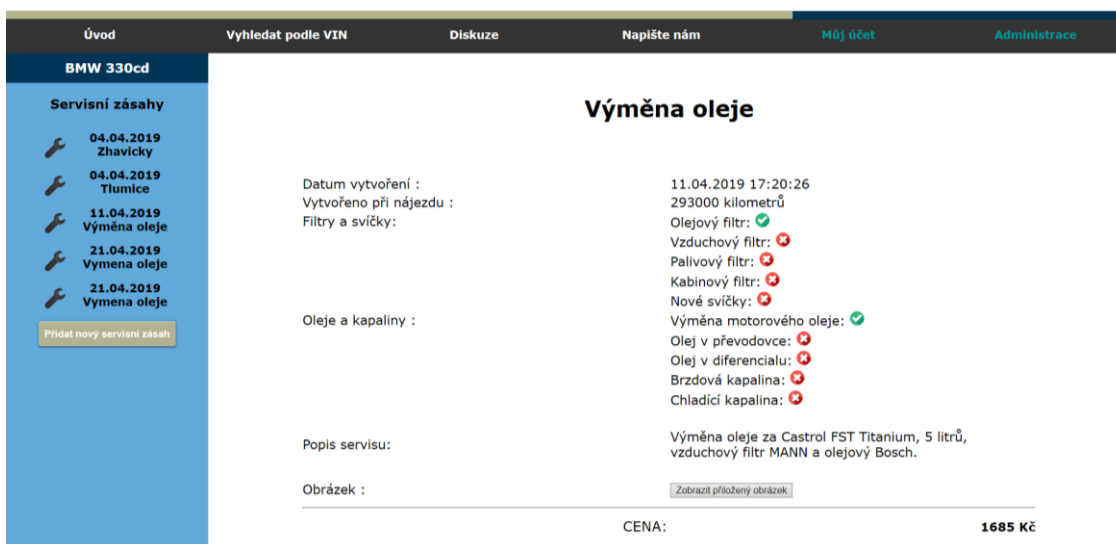
### 3.3.7 Výběr vozidla a evidence servisů

Uživateli se po kliknutí na tlačítko „Vybrat vozidlo“ zobrazí vyskakovací okno s vypsánými automobily které jsou v jeho vlastnictví. Výběr vozidla je potřebný pro přidávání a výpis jeho servisních zásahů. Pro výběr vozidla je zapotřebí kliknout na ikonu automobilu před názvem vozidla jehož servisní zásahy chceme přidávat a zobrazovat. Každá ikona neboli tlačítko před názvem automobilu nese jeho ID, tudíž po kliknutí na ikonu se pomocí metody POST uloží ID vozidla do SESSION proměnné `$_SESSION['vybrane_auto_id']` a díky tomu s tímto vozidlem můžeme pracovat i při přecházení mezi stránkami aplikace. ID vozidla je v této proměnné uloženo do doby, než si uživatel vybere jiné vozidlo, odhlásí se nebo ukončí internetový prohlížeč.



Obrázek 13: Výběr vozidla

Po kliknutí na ikonu se v levé oblasti pro výpis servisních zásahů zobrazí název vybraného vozidla s názvy a datумы provedených servisních zásahů automobilu a v oblasti pro obsah se nám zobrazí podrobný výpis nejstaršího provedeného servisního zásahu na vybraném vozidle. Servisní zásahy jsou vypsaný pomocí PHP skriptu a SQL dotazu a jsou vsazeny do tabulky uvnitř levé oblasti. Před názvem servisního zásahu a jeho datumm vytvoření se generují tlačítka s ikonou klíče, které v sobě nosí ID jednotlivých servisních zásahů. Po kliknutí na jedno z tlačítek se ID servisního zásahu odešle metodou POST, při které PHP skript provede podrobný výpis vybraného servisního zásahu. Jednotlivá data jsou uspořádána do tabulky kvůli přehlednosti.



Obrázek 14: Výpis servisních zásahů

Chce-li uživatel vytvořit nový zápis servisního zásahu, musí kliknout na tlačítko „Přidat nový servisní zásah“, které se nachází pod vypsanými servisními zásahy.

Poté se zobrazí vyskakovací okno s formulářem pro vytvoření nového servisního zásahu.

The screenshot shows a web application interface with a modal window titled "Přidání nového servisního zásahu k BMW 330cd". The modal contains the following fields and options:

- Název :
- Aktuální nájezd :  kilometrů
- Filtry a svíčky:  Olejový filtr:   
 Vzduchový filtr:   
 Palivový filtr:   
 Kabinový filtr:   
 Nové svíčky:
- Oleje a kapaliny :  Výměna motorového oleje:   
 Olej v převodovce:   
 Olej v diferenciálu:   
 Brzdová kapalina:   
 Chladičí kapalina:
- Popis servisu:
- CENA:  Kč
- Načítáte fotografii (pouze JPG,PEG,PNG,GIF a max 20MB)  Procházet...

Buttons: "Výčistit" and "Vytvořit".

Background text: "Popis servisu: Žhavičky Bosch vyměněny s relém žhavení, jelikož diagnostika psala chybu žhavení."

Obrázek 15: Přidání nového servisního zásahu

Formulář obsahuje elementy `<input>` sloužící pro zadávání informací o provedeném servisním zásahu, které se po kliknutí na tlačítko „Vytvořit“ odešlou metodou POST na server a ty se následně díky PHP skriptu s SQL dotazem uloží do databáze. Elementy `<input>` typu checkbox nabývají hodnoty „true“ v okamžiku, kdy uživatel kliknutím element zaškrtně. V opačném případě, kdy elementy zůstanou nezaškrtnuté nabývají hodnot „false“.

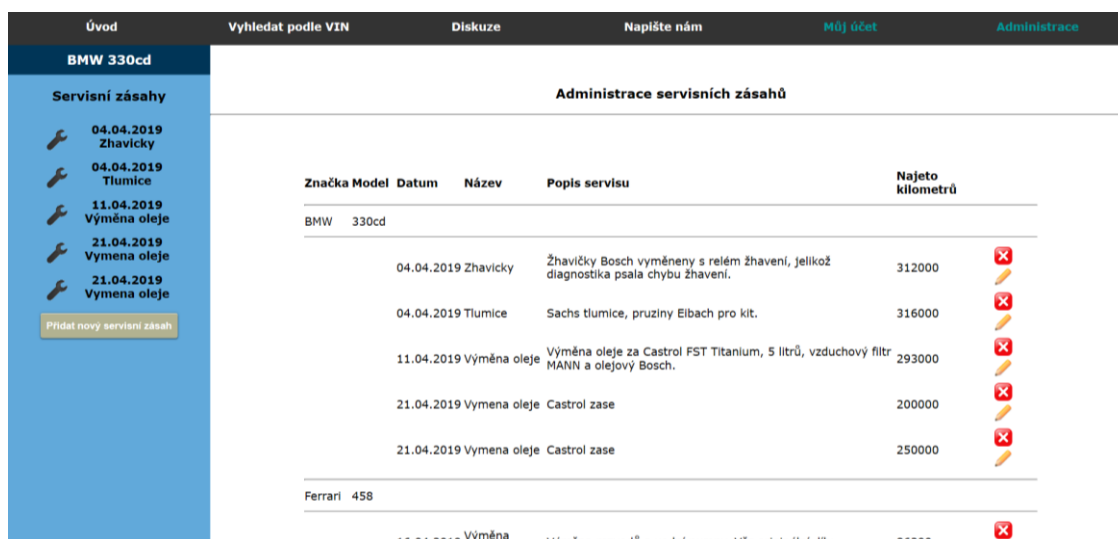
Do databáze se tyto hodnoty ukládají číslicemi 1 (true) nebo 0 (false). Ve formuláři jsou elementy `<input>` typu checkbox vytvořeny pro rychlý zápis nejčastěji prováděných servisních úkonů. Uživatel má možnost k servisnímu zásahu připojit obrázek, který si poté bude moci zobrazit v podrobném výpisu servisního zásahu.













### 3.3.8 Administrace

Administrace je určena pro správu evidovaných záznamů uživatele. V této sekci je uživatel oprávněn upravovat a mazat data svého uživatelského účtu, svých vozidel a na nich provedených servisních zásahů. Administrace také disponuje převodem vozidla na nového majitele.

Po přihlášení uživatele se v navigačním menu místo tlačítka „Registrace“ zobrazí tlačítko „Administrace“. Po kliknutí na tlačítko se uživateli zobrazí tlačítka s ikonami pro výběr mezi správou vozidel, správou servisních zásahů, úpravou uživatelského účtu a převodem vozidla na nového majitele.

Po kliknutí na některé z tlačítek se nastaví do SESSION proměnné druh módu úpravy, díky kterému aplikace pozná, jaké záznamy se budou upravovat. Pro ukázkou je zde zobrazena správa servisních zásahů, ve které se zobrazí výpis vozidel uživatele včetně na nich provedených servisních zásahů.



Značka	Model	Datum	Název	Popis servisu	Najeto kilometrů	
BMW	330cd					
		04.04.2019	Zhavicky	Žhavičky Bosch vyměněny s relem žhavení, jelikož diagnostika psala chybu žhavení.	312000	 
		04.04.2019	Tlumice	Sachs tlumice, pružiny Eibach pro kit.	316000	 
		11.04.2019	Výměna oleje	Výměna oleje za Castrol FST Titanium, 5 litrů, vzduchový filtr MANN a olejový Bosch.	293000	 
		21.04.2019	Vymena oleje	Castrol zase	200000	 
		21.04.2019	Vymena oleje	Castrol zase	250000	 
Ferrari	458					
		16.04.2019	Výměna	Výměna rozvodů a vodní pumov. Vše originální dílv.	86200	 

Obrázek 16: Správa servisních zásahů

Servisní zásahy lze upravovat kliknutím na tlačítko s ikonou tužky. Tlačítko má v sobě uložené ID příslušného servisního zásahu. Po kliknutí na tlačítko se spustí JavaScript funkce `UpravitServis(id_servisu)`, která pomocí AJAX dokáže obsah stránky měnit bez opětovného načtení. Tato funkce metodou GET odešle ID servisního zásahu PHP skriptu, který pomocí SQL dotazu získá data vybraného záznamu z databáze, a ty pomocí formátu JSON vrátí funkci nazpátek. Funkce poté zobrazí skrytý formulář s elementy `<input>` typu text, které poté naplní získanými

daty. Obsah elementů `<input>` typu text lze přepisovat. Upravená data pak lze uložit do databáze, kdy stisknutím tlačítka „Uložit“ se pomocí JavaScriptu odešlou AJAX metodou POST všechna data z formuláře PHP skriptu, který dotazem SQL upraví záznam v databázi.

Mazání záznamů je řešeno JavaScript funkcí `SmazatServis(id_servisu)`, která AJAX metodou POST odešle PHP skriptu ID servisního zásahu, který chceme odstranit a následně SQL dotazem záznam vymaže z databáze. Funkce je vyvolána kliknutím na tlačítko s ikonou křížku, které v sobě má uložené ID toho servisního záznamu, u kterého se tlačítko nachází. Před odstraněním záznamu se JavaScript zeptá, zda uživatel opravdu chce vymazat vybraný záznam, aby se předešlo ztrátě dat při nechtěném kliknutí na odstraňovací tlačítko.

Obrázek 17: Úprava servisního zásahu

### 3.3.9 Stránka Můj účet

Jedná se o přehled informací o uživatelském účtu, počtu servisních záznamů a celkových nákladů investovaných do údržby vozidel přihlášeného uživatele. Uživatel si vybráním jednoho ze svých vozidel může interaktivně pomocí JavaScriptu, AJAX a formátu JSON vypsat množství a celkové náklady servisních zásahů vybraného vozidla.

Tato stránka webové aplikace také nabízí zrychlenou tvorbu zápisu výměny oleje vybraného vozidla s přednastavenými hodnotami. Po kliknutí na tlačítko „Vytvořit výměnu oleje“ se interaktivně zobrazí textová pole pro dodatečný zápis počtu

najetých kilometrů, ceny a popisu výměny oleje pro zapsání například výrobce motorového oleje a výrobce olejového filtru. Tyto textová pole není nutné vyplňovat. Pro funkci zrychlené tvorby zápisu výměny oleje bylo využito JavaScriptu, AJAX a formátu JSON.



Obrázek 18: Stránka můj účet

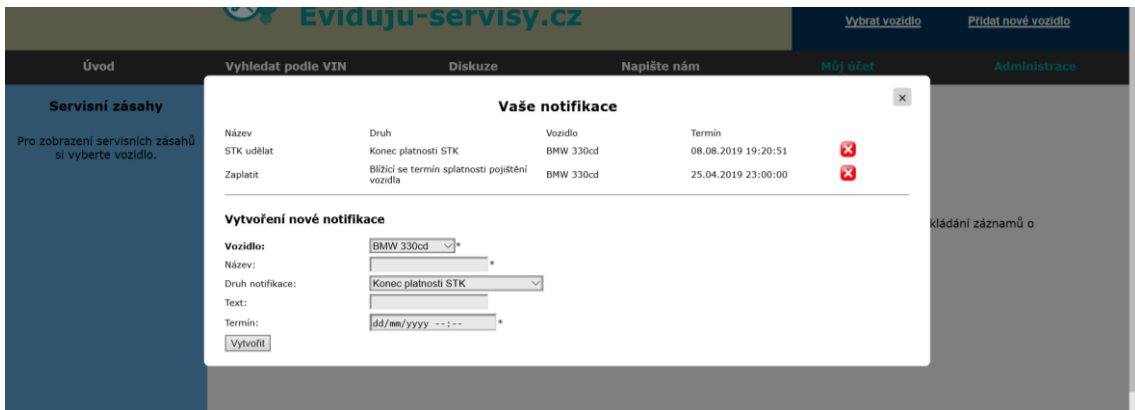
### 3.3.10 Notifikace

Uživatel může vytvářet a mazat notifikace, které slouží pro upozornění na blížící se události spojené s jeho vozidly. Upozornění se odesílají pomocí emailových zpráv do emailové schránky uživatele. Při vytváření nové notifikace uživatel ve formuláři vybere vozidlo, zadá název notifikace, druh notifikace, text notifikace a její termín. Po kliknutí na tlačítko „Vytvořit“ se metodou POST odešlou zadaná data, a pomocí PHP skriptu se uloží do databáze.

K mazání notifikací se používá JavaScript funkce, která metodou POST odešle ID vybrané notifikace do PHP skriptu, který pomocí dotazu SQL notifikaci odstraní z databáze.

K samotnému odesílání emailů slouží PHP skript, který se spouští každý den v 5:00 ráno, kdy pomocí SQL dotazu zjistí, které notifikace mají termín připomenutí v daný den a poté odešle vybrané emailové notifikace uživatelům. Existují tři druhy notifikací, tudíž každý druh má rozdílnou šablonu textu emailu. O samotné spouštění PHP skriptu se stará serverová služba CRON, jejíž rozhraní pro provozovatele webové aplikace poskytuje webhosting. Tato služba dokáže podle

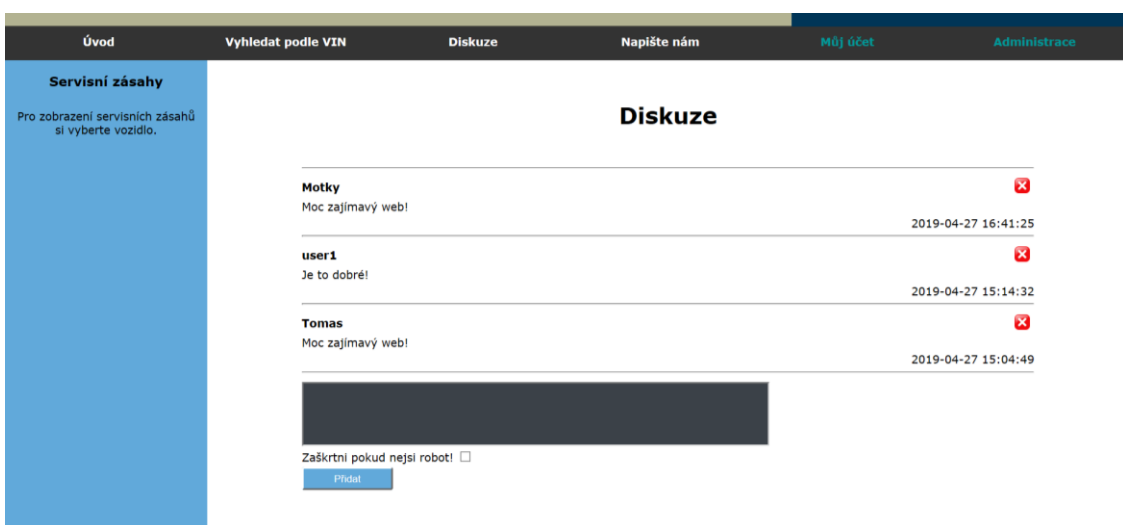
nastaveného intervalu spouštět skripty a tím se nám postará o každodenní odeslání emailových notifikací uživatelům.



Obrázek 19: Notifikace

### 3.3.11 Diskuze

Tato stránka webové aplikace je určena pro dotazy, rady, hodnocení a komunikaci uživatelů. Může sloužit pro získávání negativní zpětné vazby pro případně zdokonalení webové aplikace. Přihlášený uživatel může číst a vkládat příspěvky, ovšem nepřihlášený uživatel smí jen příspěvky číst. Načtením stránky se vypíše příspěvky uživatelů včetně elementů `<input>` pro vytvoření nového příspěvku. Pro přidání příspěvku je nutné kliknout na tlačítko „Přidat“ kdy se metodou POST odešlou data na server a PHP skript vytvoří příspěvek. Oprávnění pro odstraňování příspěvků má pouze uživatel s názvem účtu „admin“.



Obrázek 20: Diskuze

## **3.4 Propojení s registry jiné aplikace**

Jedním z úkolů bylo prověřit propojení webové aplikace s registry typu společnosti Cebia, tedy s údaji o vozidlech uloženými v jiné aplikaci. Princip fungování by měl vypadat tak, že by se uživatelé po zadání VIN vozidla do této webové aplikace zobrazily údaje o vozidle včetně historie vozidla získané z jiné aplikace. Realizace této funkce by se dala provést pomocí rozhraní API.

### **3.4.1 API**

Jedná se o rozhraní využívané v programování aplikací, které slouží programátorovi pro využívání různých funkcí, procedur nebo tříd z nějaké vytvořené knihovny nebo programu.

### **3.4.2 Existující řešení**

Společnost Cebia neposkytuje možnost sdílení svých získaných dat o vozidlech s jinými webovými aplikacemi.

V České republice existuje webová aplikace jménem Vindecoder.eu, která klientům poskytuje prověření vozidla a dekodování VIN vozidla. Nabízí sdílení podrobnějších informací o vozidlech s ostatními webovými aplikacemi pomocí API rozhraní. Bohužel tato aplikace prodává uživatelům podrobné informace jednoho vozidla za 7,29\$, tudíž z důvodů minimalizace nákladů propojení nebylo uskutečněno.

Funkce pro získávání dat z webové aplikace Vindecoder pomocí API je tvořena pomocí PHP skriptu, ve kterém se vytvoří API dotaz za pomoci REST služby o třech parametrech (ID, API key, kontrolní součet). Zájemci o tento typ propojení aplikací jsou přiděleny klíče API key a secret key, které za pomoci ID naplněném VIN vyhledávaného vozidla slouží k vytvoření kontrolního součtu. Kontrolní součet je vytvořen posledními deseti znaky hashe typu SHA1 vytvořené ze zadého ID, API key a secret key. Poté stačí odeslat požadavek pro získání dat, které jsou uloženy ve formátu JSON.



```

<?php
$apiPrefix = "https://api.vindecoder.eu/2.0";
$apikey = "*****"; // Your API key
$secretkey = "*****"; // Your secret key
$vin = "XXXDEF1GH23456789"; // Requested VIN
$id = $vin;

$controlsum = substr(sha1("${id}|${apikey}|${secretkey}"), 0, 10);

$data = file_get_contents("${apiPrefix}/${apikey}/${controlsum}/decode/${vin}.json", false);
$result = json_decode($data);

```

Obrázek 21: Příklad API dotazu v PHP

### 3.5 Zabezpečení webové aplikace

**SQL Injekce** – „*SQL injekce patří mezi jednu z nejrozšířenějších zranitelností s až kritickou závažností. Zneužití této zranitelnosti vede k úniku libovolných dat z databáze, která slouží aplikaci jako úložiště dat.*“ [15]

Stručně řečeno, pokud bychom při přihlašování nezabezpečili proti SQL injekci, mohl by se nějaký hacker přihlásit na účet některého uživatele kde by mohl číst, vytvářet a mazat data.[15]

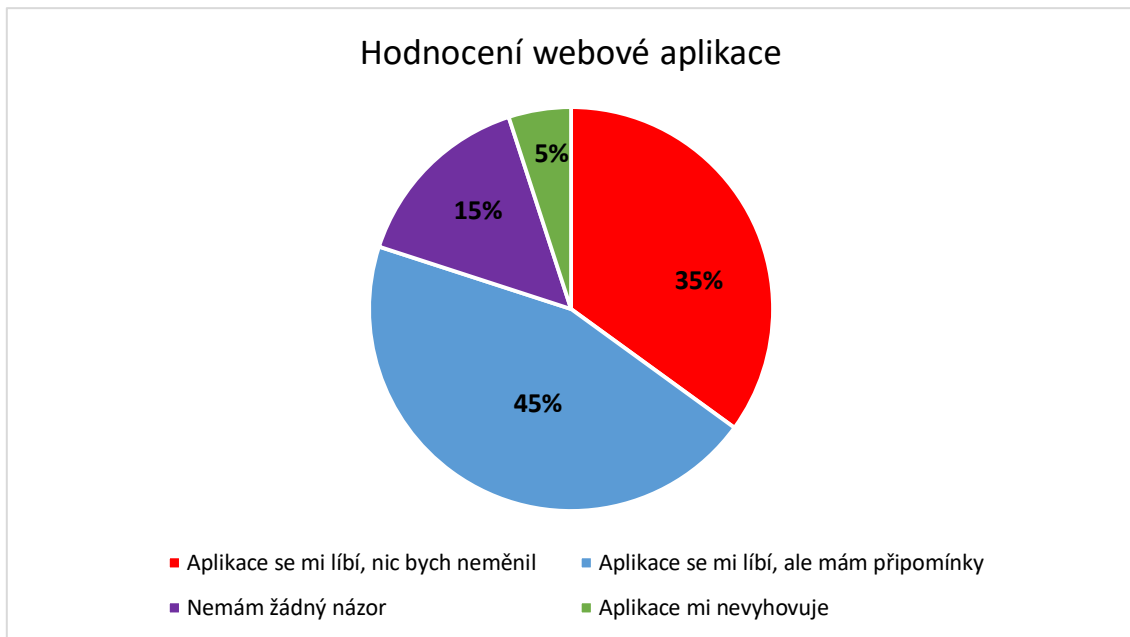
**Hash uživatelského hesla** – Uživatelské heslo se ukládá do databáze zašifrované pomocí hashovací funkce MD5. Pomocí této funkce se výrazně ztíží hackerům útoky pro získávání hesel uživatelů.

**Zabezpečené přímé odkazy** – Při vstupu přes přímý odkaz nepřihlášeného uživatele na některou ze stránek určených pro uživatele přihlášené dojde k přesměrování na úvodní stránku.

### 3.6 Zpětná vazba z testovacího provozu

Testovací vzorek tvořilo 20 klientů mužského pohlaví.

Graf ukazuje, jak klienti hodnotili webovou aplikaci. Nejvíce klientů 9 (45%) odpovědělo, že se jim aplikace líbí, ale mají připomínky (viz níže). Dále 7 (35%) klientů odpovědělo, že se jim aplikace líbí a že by na ní nic neměnili. 3 (15%) klienti nevyjádřili na webovou aplikaci žádný názor. A 1 (5%) klient odpověděl, že mu aplikace nevyhovuje z důvodu nepřehlednosti.



Graf: Hodnocení webové aplikace

**Připomínky od klientů k webové aplikaci:**

- 1) Při podrobném výpisu servisních zásahů by se mohly vypisovat pouze provedené výměny filtrů a kapaliny.
- 2) Při přejetí myší přes tlačítko s ikonou pro podrobné zobrazení servisního zásahu zvýraznit toto tlačítko pomocí CSS efektu hover.
- 3) Doplnit více typů karosérií vozidel.
- 4) Výpis provedených servisních zásahů vozidla v levém oblasti stránky obohatit o počet najetých kilometrů při realizaci zásahu.
- 5) Výběr vozidla realizovat kliknutím na název vozidla.
- 6) Umožnit uživatelům smazat své příspěvky v diskuzi.
- 7) Vytvořit lepší vizuální vzhled stránky.
- 8) Poskytnout možnost změny jazyka pro anglicky mluvící uživatele.
- 9) Při úpravě vozidel vytvořit větší element <input> typu text pro přehlednější úpravu informací o vozidle.

## Závěr

Cílem této bakalářské práce bylo vytvořit fungující webovou aplikaci určenou pro evidenci servisních zásahů vozidel. Mým úkolem bylo najít a analyzovat existující aplikace určené pro evidenci servisních zásahů vozidel, navrhnout webovou aplikaci včetně funkcí které bude vykonávat a ověřit možnost propojení aplikace s existujícími záznamy jiné aplikace. Tyto úkoly jsem splnil a na závěr implementoval navržená řešení a webovou aplikaci vytvořil.

Webová aplikace pomáhá uživateli udržovat přehled o provedených servisních úkonech vozidel, poskytuje přesné plánování dalšího intervalu výměny komponenty vozidla a počítá celkové náklady spojené s údržbou vozidel. Aplikace také slouží k podrobnějšímu popisu technického stavu vozidla a tím pádem může vést ke snadnějšímu prodeji. Aplikace disponuje notifikacemi, které uživatele upozorní na blížící se události spojeny s jeho vozidly.

Díky této práci jsem získal nové znalosti ve tvorbě webových aplikací a vytvořil jsem aplikaci, která bude používána pro mé osobní účely.

Do budoucna by aplikace mohla umožňovat výpis informací a historie vozidla získaných z jiné aplikace pomocí implementace API dotazu popsaného v práci. Inovací aplikace by mohlo být umožnění výběru typu servisního zásahu, kdy by se uživateli zobrazila šablona s přednastavenými hodnotami a souvisejícími možnostmi výběru použitých autodílů. Další inovací by mohl být výpis blížících se pravidelných servisních úkonů (výměna oleje, rozvodů atd.), kde by uživatel viděl, při jakém nájezdu kilometrů se má blížící se úkon provést.

## Seznam použité literatury

- [1] AdmWin - modul Autoservis. *AdmWin* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.admwin.cz/nabidka/ucetni-program-autoservis/>
- [2] Carsys: software Autoservis. *Carsys* [online]. [cit. 2019-04-29]. Dostupné z: <http://www.carsys.cz/software/autoservis/>
- [3] Carsys: ceny licencí. *Carsys* [online]. [cit. 2019-04-29]. Dostupné z: [http://www.carsys.cz/uzivatele/dokumentace/Ceny\\_licenci\\_CARSYSTEM\\_1\\_2.pd](http://www.carsys.cz/uzivatele/dokumentace/Ceny_licenci_CARSYSTEM_1_2.pd)
- [4] Fuelly: mobilní aplikace. *Fuelly* [online]. [cit. 2019-04-29]. Dostupné z: <http://www.fuelly.com>
- [5] Motortab: webová aplikace. *Motortab* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.motortab.com/>
- [6] Počet osobních vozidel v ČR. *České Noviny* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.ceskenoviny.cz/zpravy/v-cesku-je-registrovano-pres-5-7-milionu-osobnich-aut/1650740>
- [7] Webová aplikace (Web Application). In: ManagementMania.com [online]. Wilmington (DE) 2011-2019, 18.10.2018 [cit. 29.04.2019]. Dostupné z: <https://managementmania.com/cs/webova-aplikace-web-application>
- [8] HTTP (Hypertext Transfer Protocol). In: ManagementMania.com [online]. Wilmington (DE) 2011-2019, 07.12.2016 [cit. 29.04.2019]. Dostupné z: <https://managementmania.com/cs/http-hypertext-transfer-protocol>
- [9] Cookies. In: ManagementMania.com [online]. Wilmington (DE) 2011-2019, 27.05.2018 [cit. 29.04.2019]. Dostupné z: <https://managementmania.com/cs/cookies>
- [10] SESSION. *ITslovník.cz* [online]. [cit. 2019-04-29]. Dostupné z: <https://it-slovník.cz/pojem/session>

- [11] HTTPS definice. *Adaptic* [online]. [cit. 2019-04-29]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/https/>
- [12] HANÁK, Jiří. SSL definice. *Master* [online]. 29-03-2016 [cit. 2019-04-29]. Dostupné z: <https://www.master.cz/blog/co-jsou-ssl-certifikaty-a-ssl-protokoly-jak-funguji-vysvetleni-navod/>
- [13] SSL certifikáty. *SSLmentor* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.sslmentor.cz/ssl/ssl-certifikaty>
- [14] Webhosting Endora. *Endora* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.endora.cz/>
- [15] SQL Injection. *ITnetwork* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.itnetwork.cz/php/bezpecnost/tutorial-bezpecnost-v-php-utok-sql-injection-a-obrana>
- [16] Definice JavaScript. *Adaptic* [online]. [cit. 2019-04-29]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/javascript/>
- [17] Definice AJAX. *Adaptic* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.adaptic.cz/znalosti/slovnicek/ajax/>
- [18] Definice PHP. *Adaptic* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.adaptic.cz/znalosti/slovnicek/php/>
- [19] Definice CSS. *Adaptic* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.adaptic.cz/znalosti/slovnicek/css/>
- [20] ROLEČEK, Luděk. Úvod do HTML a CSS. *Czechitas* [online]. 18.01.2019, , 1 [cit. 2019-04-29]. Dostupné z: <https://www.czechitas.cz/cs/blog/zaciname-s-it/online-kurz-uvod-do-html-css-1-lekce>
- [21] Databáze. *ITnetwork* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.itnetwork.cz/programovani/sql/kterou-sql-databazi-pouzit>

- [22] BŘÍZA, Petr. Tvorba layoutu. *Interval* [online]. 03.03.2004 [cit. 2019-04-29]. Dostupné z: <https://www.interval.cz/clanky/tvorba-layoutu-webu-teoreticky-uvod>
- [23] Vincario: API sdílení. *Vincario* [online]. [cit. 2019-04-29]. Dostupné z: <https://vindecoder.eu/cz/my/api/2.0/docs>
- [24] Webhosting teorie. In: *Český hosting* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.cesky-hosting.cz/webhosting/co-je-webhosting/>

# Přílohy

## SQL tvorba databázových tabulek

### Tabulka uživatelé:

```
CREATE TABLE uzivatele (  
login varchar(255) NOT NULL,  
jmeno varchar(255) NOT NULL,  
prijmeni varchar(255),  
heslo varchar(255) NOT NULL,  
mesto varchar(255),  
email varchar(255) NOT NULL,  
datum_registrace datetime NOT NULL,  
PRIMARY KEY (login));
```

### Tabulka vozidla:

```
CREATE TABLE vozidla (  
ID_vozidla int(11) AUTO_INCREMENT,  
majitel varchar(255) NOT NULL,  
VIN varchar(55),  
znacka varchar(55) NOT NULL,  
model varchar(55) NOT NULL,  
rok_vyroby int(11) NOT NULL,  
obsah int(11) NOT NULL,  
palivo varchar(55) NOT NULL,  
najeto_km int(11),  
vykon int(11),  
karoserie varchar(55),  
barva varchar(55),  
informace varchar(255),  
PRIMARY KEY (ID_vozidla),  
FOREIGN KEY (majitel) REFERENCES uzivatele(login));
```

### Tabulka přepisy:

```
CREATE TABLE prepisy (  
id_prepisu int(11) AUTO_INCREMENT,  
id_vozidla int(11) NOT NULL,  
kod varchar(55) NOT NULL,  
PRIMARY KEY (id_prepisu),  
FOREIGN KEY (id_vozidla) REFERENCES vozidla(ID_vozidla));
```

### **Tabulka servisy:**

```
CREATE TABLE servisy (  
id_servisu int(11) AUTO_INCREMENT,  
nazev varchar(255) NOT NULL,  
id_vozidla int(11) NOT NULL,  
datum datetime NOT NULL,  
filtr_olejovej tinyint(1),  
filtr_vzduchovej tinyint(1),  
filtr_palivovej tinyint(1),  
filtr_kabinovej tinyint(1),  
kap_chladici tinyint(1),  
kap_brzdova tinyint(1),  
olej tinyint(1),  
olej_prevodovka tinyint(1),  
olej_diferencial tinyint(1),  
svicky tinyint(1),  
popis varchar(255),  
uzivatel varchar(255) NOT NULL,  
kilometry int(11) NOT NULL,  
cena int(11),  
obrazek varchar(255),  
PRIMARY KEY (id_servisu),  
FOREIGN KEY (id_vozidla) REFERENCES vozidla(ID_vozidla),  
FOREIGN KEY (uzivatel) REFERENCES uzivatele(login));
```

### **Tabulka fotky:**

```
CREATE TABLE fotky (  
id_fotky int(11) AUTO_INCREMENT,  
id_servisu int(11) NOT NULL,  
cesta_fotky varchar(255) NOT NULL,  
nazev varchar(255) NOT NULL,  
PRIMARY KEY (id_fotky),  
FOREIGN KEY (id_servisu) REFERENCES servisy(id_servisu));
```

### **Tabulka diskuze:**

```
CREATE TABLE diskuze (  
id_diskuze int(11) AUTO_INCREMENT,  
login varchar(255) NOT NULL,  
datum datetime NOT NULL,  
obsah varchar(255) NOT NULL,  
PRIMARY KEY (id_diskuze),  
FOREIGN KEY (login) REFERENCES uzivatele(login));
```

### **Tabulka notifikace:**

```
CREATE TABLE notifikace (  
id_notifikace int(11) AUTO_INCREMENT,  
login varchar(255) NOT NULL,  
id_vozidla int NOT NULL,  
nazev varchar(255) NOT NULL,  
druh varchar(255) NOT NULL,  
datum datetime NOT NULL,  
text varchar(255),  
PRIMARY KEY (id_notifikace),  
FOREIGN KEY (login) REFERENCES uzivatele(login),  
FOREIGN KEY (id_vozidla) REFERENCES vozidla(ID_vozidla));
```