

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Návrh a implementace webové aplikace pro autobazar

Bakalářská práce

Vedoucí práce:
doc. Ing. Ivana Rábová, Ph.D.

David Slabý

Brno 2016

Tímto bych chtěl poděkovat paní doc. Ing. Ivaně Rábové, Ph.D. za ochotu při vedení bakalářské práce a v neposlední řadě také za její cenné rady a připomínky.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Návrh a implementace webové aplikace pro autobazar**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 3. února 2017

.....

Abstract

Slabý, D. Design and implementation of a web application for car dealer.
Bachelor thesis. Brno: Mendel university, 2016.

This Bachelor thesis is focused on an analysis of company processes of a used car dealer and on design and implementation of a web application prototype. The application enables the user to keep a record of information about the cars and to print documents of desired structure. In the theoretical part the essential terms of information systems are clarified and the used technologies are described that are related to the UML modelling and modern web application development. The practical part is devoted to a design of a new system solution that is based on company processes analysis and on the specific requirements. Included is also the implementation of the application and its testing in operation.

Keywords

UML, information systems, web applications, vehicle evidence, Bootstrap, MySQL, PHP framework

Abstrakt

Slabý, D. Návrh a implementace webové aplikace pro autobazar.
Bakalářská práce. Brno: Mendelova univerzita, 2016.

Bakalářská práce je zaměřena na analýzu firemních procesů autobazaru, návrh a následnou implementaci prototypu webové aplikace. Aplikace slouží k evidenci informací o vozidlech a umožňuje uživateli tisk dokumentů požadované struktury. V teoretické části jsou vysvětleny základní pojmy informačních systémů a popsány použité technologie související s problematikou UML modelování a moderního vývoje webových aplikací. Praktická část se věnuje návrhu nového řešení systému na základě analýzy firemního prostředí a specifikace požadavků. Součástí je implementace aplikace a její testování v provozu.

Klíčová slova

UML, informační systémy, webové aplikace, evidence vozidel, Bootstrap, MySQL, PHP framework

Obsah

1	Úvod	6
2	Cíl práce	7
3	Metodika řešení	8
4	Literární řešerše	9
4.1	Webové aplikace	9
4.2	Informační systémy	10
4.3	UML modelování	12
4.4	HTML	15
4.5	CSS	17
4.6	Javascript	18
4.7	Bootstrap	19
4.8	Programovací jazyk PHP	20
4.9	Další nástroje a techniky	24
5	Vlastní práce	25
5.1	Analýza firmy	25
5.2	UML modelování	29
5.3	Implementace	36
6	Diskuse	41
6.1	Přínosy aplikace	41
6.2	HW a SW infrastruktura	41
6.3	Ekonomické zhodnocení	42
6.4	Budoucí vývoj	42
7	Závěr	43
8	Seznam zdrojů	44
	Přílohy	46
A	LoginPresenter	47
B	Ukázka výstupu	49

1 Úvod

V souvislosti s rostoucí poptávkou po IT službách lze konstatovat, že většina firem si již nedokáže svou existenci bez moderních technologií představit. Informační systém (IS) v podnikové praxi plní významnou roli při sběru, šíření a transformaci informací na výstupy. Informace jsou uchovány v rozsáhlých databázích běžících na pozadí systému, které zajišťují bezpečnost a dostupnost dat. Pořízení systému s sebou nese spoustu přínosů spočívajících převážně ve zvýšení efektivity některých vnitropodnikových procesů. IS může mít v praxi podobu webové aplikace, kterou běžně užíváme. Přívětivé uživatelské rozhraní umožňuje práci s aplikací prakticky komukoliv. Příkladem může být třeba univerzitní informační systém, který každý z nás používá téměř každý den.

Náklady na implementaci se liší podle zvoleného řešení, volba správného řešení se odvíjí od analýzy konkrétní firmy, nicméně pokud firma eviduje větší množství informací o svých produktech a službách, je integrace IS v podstatě nutností. Ve vztahu k trendu informačních systémů bude práce popisovat právě tuto oblast IT - modelování a vývoj IS.

2 Cíl práce

Cílem bakalářské práce je na základě analýzy firemního prostředí navrhnout a realizovat prototyp webové aplikace pro autobazar. Pro splnění tohoto cíle bude nezbytné analyzovat firemní procesy a problémovou doménu firmy, následně popsat aktuální řešení systému uchování informací a specifikovat funkční a nefunkční požadavky na nový systém. Tyto poznatky poslouží k návrhu webové aplikace na míru. Návrh bude proveden pomocí nástrojů jazyka UML, které jsou vhodné pro komplexní modelování funkcionality a chování systému. V našem případě se jedná o diagramy případů užití s příslušnými scénáři a sekvenční diagramy. Dále dojde k návržení struktury databáze pro uchování dat o vozidlech, schéma vhodně zachytí ER diagram. Výsledkem inovace a zjednodušení stávajícího řešení bude webová aplikace určená pro evidenci vozidel s možností tisku výpisů a smluv, která uživateli umožní bez jakýchkoliv potíží a zdržení dospět k požadovanému výstupu, a to i na mobilních zařízeních. Na závěr proběhne test aplikace v ostrém provozu a zpracování připomínek zjištěných při testování.

3 Metodika řešení

Po nastudování teoretických poznatků objektové analýzy a jazyka UML bude provedena analýza a navržena inovace informačního systému.

Teoretická část práce bude obsahovat rozbor témat týkajících se webových aplikací, informačních systémů a jejich vývoje, ale také UML modelování. V souvislosti s informačními systémy dojde k vysvětlení obecných pojmů jako jsou data, informace apod. Dále bude následovat popis využitých technologií pro vývoj webových aplikací. Jedná se o HTML, CSS, Javascript, jQuery, PHP a MySQL. Za zmínku jistě stojí i framework Bootstrap, Nette a jeho MVP architektura.

V rámci praktické části práce bude namodelován podrobný Use Case diagram aplikace, který částečně popíšu přiložené scénáře vybraných případů užití (Use Case). Interakce mezi jednotlivými objekty systému znázorní sekvenční diagram a datovou strukturu bude vhodně prezentovat ER diagram. Implementace webové aplikace dle specifikace požadavků uživatele proběhne s využitím dříve popsaných technologií a v práci budou rozebírány jednotlivé ukázkové bloky zdrojového kódu.

4 Literární rešerše

4.1 Webové aplikace

Webovou aplikaci lze charakterizovat jako aplikaci, která je poskytována uživatelům z webového serveru pomocí Internetu nebo pomocí vnitropodnikové obdoby, tzn. intranetu. Funkci klienta plní webový prohlížeč, který nezná logiku aplikace. Význam webových aplikací roste, jelikož jsou implementovány jako podnikové, ale i jiné informační systémy, dále pak jako freemaily, internetové obchody, online aukce, diskusní fóra a další.

Velkou výhodou vývoje u aplikací stavěných na standardních funkcích prohlížeče je její multiplatformní schopnost. V praxi to znamená, že uživatel disponuje možnostmi aplikaci spustit na jakémkoliv operačním systému bez ohledu na jeho verzi. Problém může nastat až ve chvíli, kdy uživatel začne zasahovat do zobrazení aplikace, například zvolí jiný řez písma, vypne podporu skriptování apod. Tyto volby mohou narušit jednotný vzhled a funkčnost aplikace. Další výrazné potíže také mohou způsobit jiné specifikace jednotlivých prohlížečů, zejména pak nekonzistentí implementace HTML, CSS a DOM ¹.

Za nevýhodu lze označit vysokou závislost na poskytovateli aplikace, pokud se zrovna nejedná o intranetovou aplikaci. Poskytovatel může kdykoliv omezit možnosti využití aplikace, kterou provozuje na své straně. Navíc by uživatel měl k plynulému užívání disponovat dostatečnou kapacitou připojení k internetu, což by v dnešní době již nemělo být problémem. Nicméně na druhé straně má uživatel prakticky nulovou údržbu a minimální náklady, ve většině případů pouze paušální poplatky za využívání aplikace.

Webové aplikace jsou ve většině případů implementovány ve třech vrstvách. Běžně je webový prohlížeč brán jako vrstva první (prezenční), nástroje pro dynamické generování stránek (v našem případě PHP) jako střední vrstva (logická) a databáze jako vrstva třetí (datová). Webový prohlížeč posílá požadavky střední vrstvě (back-end aplikace), která je zpracovává skrze dotazy nad databází a generováním uživatelského rozhraní (front-end aplikace). (Jakub Foral, 2013)

¹DOM - objektový model dokumentu. Umožňuje přistupovat k jednotlivým objektům XML(XHTML) dokumentu a pracovat s nimi. (zdroj: <http://www.tvorba-webu.cz/dom/>)

Webové aplikace, respektive webové informační systémy dnes představují neodmyslitelnou součást běžné firemní praxe. Několik základních aspektů určuje rozdíl mezi běžným softwarem a webovou aplikací. Jedná se zejména o větší citlivost k dané oblasti, pro kterou je aplikace navrhována. Webové informační systémy lze dělit na:

- Intranetové aplikace - aplikace je dostupná jen v prostředí organizace.
- Webové aplikace - obecně se do této kategorie řadí aplikace, které jsou vhodné pro komunikaci mimo rozsah organizace. Slouží pro marketingové účely. Narozdíl od intranetových aplikací není zcela přesně známa cílová skupina uživatelů, kteří budou se systémem pracovat. Při vývoji dochází ke spolupráci mezi více subjekty.
- Elektronické obchodování - lze označit jako prodejní kanál (e-shop). Probíhá na něm přímý prodej zboží a služeb, takže prakticky nahrazuje objednávkový systém.

(Radek Šilhavý et al., 2013, s. 1)

4.2 Informační systémy

Data a informace

Obecně jsou informace považovány za data, která mají pro uživatele specifický význam a jsou schopna uspokojit konkrétní informační potřebu koncového uživatele. Pokud se podíváme blíže, pak informace reprezentuje vypovídací schopnost dat, vznik je datován k momentu zpracování dat a lze ji označit za cíl zpracování.

(Ivana Rábová, 2008, s. 8)

Co se týče souvislosti informace a informačního systému, existují různé definice pojmu informace, které ve většině případů vyplývají z oboru, ve kterém se pohybujeme. Pokud se zaměříme na projektování IS, budeme o pojmu informace uvažovat v pragmatickém slova smyslu. Data můžeme chápat jako rozpoznané signály (údaje), které vypovídají o situacích a stavech sledovaných a řízených objektů. Dalším zpracováním dochází k transformaci dat v informace. Samotné informace jsou tedy taková data, na jejichž základě získává uživatel možnost rozhodování, kterým uskutečňuje svou zpětnou vazbu na informační systém, aby dosáhl svého cílového chování. Nelze jednoznačně určit význam dat, liší se dle pohledu nebo interpretace různých uživatelů, tudíž získané informace mohou mít taktéž různý význam.

(Dominik Vymětal, 2009, s. 14)

Obecným vymezením pojmu **systém** je množina prvků a vazeb. Komplexně systém vykazuje určitou funkci (chování). Podle Berfantlyho formulace může být systém považován za "komplex prvků nacházejících se ve vzájemné interakci". Jeho prvky na dané úrovni rozlišení lze označit jako nedělitelné. Specifikací vazeb se rozumí spojení mezi prvky dvojího typu, jedná se o spojení obousměrné nebo jednosměrné. Vazby lze dále dělit na vstupní a výstupní, které získávají informace z okolí, nebo naopak informace okolí předávají. (Dominik Vymětal, 2009, s. 13)

Definice pojmu **informační systém** (IS) vychází z obecného základu, který již byl zmíněn v souvislosti s pojmem systém. Jedná se o uspořádání vztahů mezi lidmi, datovými a informačními zdroji a procedurami jejich zpracování za účelem dosažení stanovených cílů. Dle informačního obsahu lze rozlišovat mezi daty, informacemi a znalostmi, které slouží pro účely zpracování v informačním systému.

(Dominik Vymětal, 2009, s. 14)

Nejnižší složkou jsou signály, které můžeme brát jako analogové nebo digitální nosiče dat. Z hlediska IS lze signály označit za veličinu, která se mění v čase, eventuálně i v prostoru nebo místě vzniku.

Za procedury zpracování informací jsou označovány technologie, které charakterizují technické, programové a metodické prostředky. Jejich funkcí je pořízení, uchování, zpracování, prezentace a v neposlední řadě přenos dat. Další nezbytnou součástí IS je také organizace práce, řízení chodu systému.

(Ivana Rábová, 2008, s. 8)

Důvodem nasazení IS bývá povětšinou zefektivnění administrativních činností. Začlenění takových technologií může být otázkou tvorby hodnot podniku, postavení podniku na trhu, obecně pak jde o rozvoj nebo přežití podniku.

(Dominik Vymětal, 2009, s. 17)

Vývoj IS

Vývoj informačního systému lze také označit za jeho životní cyklus, který se dělí na jednotlivé etapy.

1. V první etapě vývoje dochází k **provedení analytických prací a volbě rozhodnutí**. V praxi jde o volbu, jestli je třeba implementovat systém nový, nebo jednoduše inovovat stávající řešení systému. Toto určení vychází z firemní strategie. Důležitým hlediskem může být, jak se strategie osvědčila v minulosti. Do této fáze je nutné rovněž začlenit i definici požadavků na systém, charakteristiku jeho cílů, přínosů a v neposlední řadě také vliv na chod a úroveň podniku.
2. Druhou etapu lze označit jako **výběr systému a implementačního partnera**. Nejvíce podstatnou součástí této etapy je volba produktu, tzn. hardware, software, infrastruktury a služeb, který je adekvátní nárokům organizace. Zahrnuje také neméně důležitou volbu implementačního partnera (dodavatele systému a systémového integrátora). V případě rozsáhlejších projektů lze v dnešní době využít služby poradenských společností. Důležitými ukazateli pro objednavatele jsou reference v oboru, ale také úroveň funkcionality, cena a kvalita servisních služeb.
3. Další fází bude samotné **uzavření smluvního vztahu**. Neměl by být opomíjen význam této fáze. Krátce charakterizujeme proces jako předložení sady smluv objednavateli. Jedná se o smlouvy licenční, smlouvy o implementaci a servisní podpoře.

4. Za čtvrtou etapu je považována **implementace**. Podstatou této fáze vývoje je přizpůsobení systému požadavkům podniku, tzv. customizace. V souvislosti s tímto procesem probíhá i školení uživatelů pro práci se systémem. Důležitou roli hraje časový diagram projektu, plán investic a organizace týmu, který se na vývoji podílí. Velký význam pro hladký chod má tedy personální složení týmu, způsob jeho řízení a v neposlední řadě také organizace práce.
5. Následuje etapa pátá - **užívání a údržba**. Řešení systému je uvedeno do ostrého provozu za předpokladu splněných přínosů. Nejvýznamnějším aspektem je plná funkčnost systému, která vede k naplnění přínosů z integrace systému do firemního prostředí. Lze tedy říct, že správa a údržba systému je nutností pro zajištění jeho správné funkce a předchází výpadkům, které by vedly k omezení prodeje podniku.
6. Poslední etapou je **rozvoj, inovace a "odchod do důchodu"**. Spočívá v rozšíření stávajícího řešení. V praxi jde o rozšíření již implementovaného jádra systému o další aplikace, které jsou integrovány do podnikového systému. Za důvod rozšíření lze považovat detailnější pokrytí klíčových procesů podniku.

(Hana Klčová et al., 2016, s. 97)

Na závěr můžeme konstatovat, že životní cyklus informačního systému se neustále zkracuje, nicméně rozšíření zadání a následná inovace systému je během vývoje vcelku běžnou situací. Taková situace povětšinou nastává u větších podniků.

4.3 UML modelování

Prostředkem UML modelování je samotný jazyk UML (Unified Modeling Language, unifikovaný modelovací jazyk), univerzální jazyk určený pro vizuální modelování systémů. Nejčastěji bývá spojován s modelováním objektově orientovaných softwarových systémů. Jazyk umožňuje implementaci ve všech nástrojích CASE (computer-aided software engineering). Výstupem jsou diagramy, které jsou srozumitelné pro lidi, nicméně také je možné využít takové grafické vyjádření k snadné interpretaci v nástrojích CASE. Pokud bychom chtěli jasně určit význam jazyka UML, plní svůj účel jako vizuální vyjádření syntaxe využitelné při sestavování vlastních modelů.

Za metodiku je považován Unified Process. Metodika určuje, jaké pracovníky musíme použít, jaké činnosti vykonat a jaké produkty vyrobit k úspěšnému sestavení modelu funkčního softwarového systému. Unified Process využívá jazyk UML jako vlastní syntaxi pro vizuální modelování. Při bližším pohledu je tato metodika označována jako upřednostňovaná v kontextu s nevhodnější adaptací pro jazyk UML. (Arlow, Neustadt, 2011, s. 28)

Use Case diagram

Diagram nazýváme v české terminologii jako diagram případů užití. Zachycuje chování systému z pohledu uživatele. Jeho účelem je získávání a dokumentace požadavků.

Součástí modelování diagramu jsou následující činnosti:

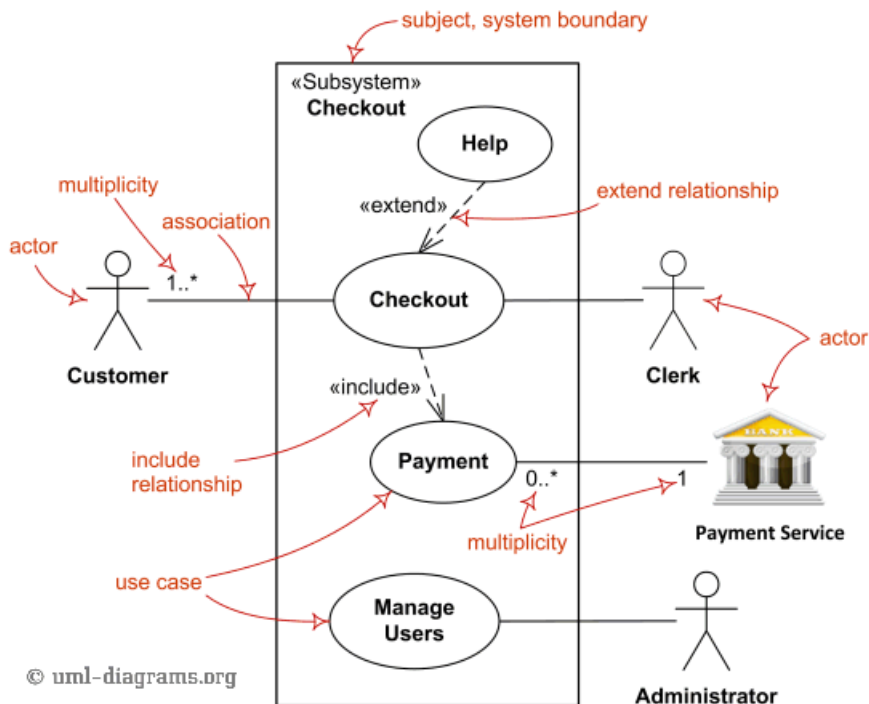
1. **Nalezení hranic systému** - Hranice označujeme jako subjekt. Slouží k určení, co je součástí systému a co naopak není, tzn. jeho okolí. Vymezení hranic má neodmyslitelný vliv na funkční a v mnoha případech i nefunkční požadavky. Subjekt udává ten, kdo systém užívá a to, co specifikuje přínos systému aktérům. Zobrazení odpovídá rámečku s popiskem názvu systému. Aktéři se nachází mimo vytyčené hranice systému a uvnitř naopak případy užití.
2. **Vyhledání aktérů** - dochází k přidělování rolí uživatelům systému. Ten ji přijímá v okamžiku, kdy začal systém bezprostředně používat. Aktér je ve vztahu k systému externí entitou a může mít spoustu rolí. Pro správnou volbu aktéra je třeba zodpovědět spoustu otázek.
3. **Nalezení případů užití**, jež zahrnuje specifikace případů užití a určení alternativních scénářů. Případy užití specifikují činnosti, které je systém schopen vykonat skrze interakci s vnějšími aktéry. Případy užití považujeme za součásti systému a lze je doplnit o případy užití podsystému i jednotlivých tříd.
4. Dochází k opakování postupu do té doby, než se dospěje k ustálení dříve jmenovaných aktivit.

(Arlow, Neustadt, 2011, s. 91)

Podoba diagramu je dána komponenty:

- **Hranice systému** (system boundary) - vymezení území nebo hranic modelovaného systému
- **Aktéři** (actors) - lze charakterizovat jako role uživatelů systému
- **Případy užití** (use cases)- aktivity, které je možné vykonat ze strany uživatele
- **Relace** (relationships)- jedná se o vztahy mezi aktéry a případy užití

(Arlow, Neustadt, 2011, s. 91)



Obrázek 1: Diagram případů užití s popisky komponent

zdroj: <http://www.uml-diagrams.org/use-case-diagrams.html>

Scénáře

Scénář, jenž jinak označujeme jako tok událostí, popisuje jednotlivé kroky případu užití. Rozlišujeme hlavní a vedlejší (alternativní) scénáře. V případě hlavního scénáře jde o situaci, kdy vše probíhá bez komplikací. V opačném případě poslouží scénář vedlejší k zahrnutí chyb, rozvětvení nebo přerušení hlavního toku. Za začátek hlavního toku je považována určitá činnost zahájení ze strany aktéra. Další tok událostí má podobu krátkých deklarativních kroků uspořádaných do časové posloupnosti. Problematika scénářů bude podrobněji řešena v praktické části práce. (Arlow, Neustadt, 2011, s. 102)

Sekvenční diagram

Cílem sekvenčních diagramů je vhodně znázornit časově orientovanou řadu zpráv předávaných mezi objekty. Jedná se o diagram spadající pod skupinu diagramů interakce, kterou nejpružněji reprezentují. Výhodnou vlastností znázornění je možnost vkládání "skriptů", jež umožňuje porozumět diagramu i netechnickým uživatelům. Jeho struktura má formu skutečných kroků vycházejících z případů užití, nebo textového shrnutí dějů zachycených v diagramu.

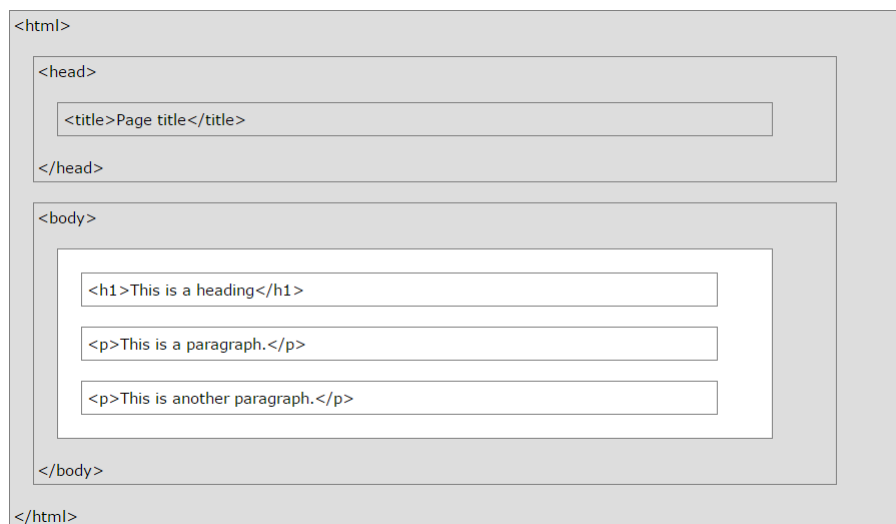
4.4 HTML

Obecně můžeme říci, že webová stránka se skládá ze tří částí.

Jsou to:

- **Textový obsah** v podobě prostého textu, který má čistě informační účel, tedy informovat návštěvníka o prezentovaném produktu, službě apod.
- **Odkazy na jiné stránky** umožňují nahrávání obrázkových, zvukových a videosouborů. Slouží jako prostředek k nalinkování jiných HTML stránek, stylů (určujících vzhled stránky) a také Javascriptových souborů starajících se o chování stránky.
- **Značky** jazyka HTML, které popisují samotný textový obsah a definují odkazy.
- Kromě těchto tří hlavních komponent obsahuje struktura stránek ještě kód jazyka, který ji popisuje jako celek. Jedná se o tzv. meta tagy. Určují primární jazyk obsahu, znakovou sadu textu (nejčastěji UTF-8) a také informace pro vyhledávací roboty.

(Castro, Hyslop, 2012, s. 23)



Obrázek 2: HTML struktura stránky

zdroj: Introduction to HTML: http://www.w3schools.com/html/html_intro.asp

HTML označuje hypertextový značkový jazyk (značky označovány jako tagy). Funkce jazyka HTML je tedy jednoduchá, spočívá v uchování obsahu stránek a popisu jeho významu. Umožňuje zpřístupnit obsah široké mase uživatelů bez ohledu na skutečnost, že webové stránky mohou na různých zařízeních a prohlížečích vypadat odlišně.

Jeho vznik se datuje k počátku 90. let 20. století. Zpočátku měl podobu stručného dokumentu, který popisovalo několik elementů sloužících jako základní kameny webových stránek. Jazyk prošel postupem času vývojem, který se vyznačoval rozrůstajícím se množstvím dostupných elementů a také přizpůsobení nejnovějším trendům v tvorbě webových stránek. Vývoj udává jednotlivé verze jazyka, poslední verzí je HTML5. (Castro, Hyslop, 2012, s. 16)

HTML5

Tato nová verze HTML poskytuje vývojáři nové nástroje k tvorbě přívětivějšího uživatelského rozhraní.

Za výhody lze považovat:

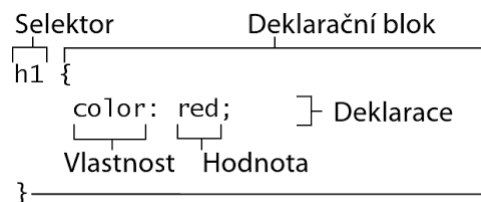
- **Popisnější zdrojový kód** - nové elementy definující jednotlivé části stránky, tj. záhlaví, zápatí, navigace, články apod.
- **Multimédia s menší závislostí na zásuvných modulech** - podpora videí, audia a vektorové grafiky bez závislosti na Flash modulu
- **Lepší aplikace** - vývoj bohatších a interaktivnějších webových aplikací
- **Posílání zpráv mezi dokumenty** - skriptování mezi weby
- **Webové sockety** - umožňují trvalé připojení k serveru bez neustálé aktualizace webu
- **Úložiště na straně klienta** - rozšíření o webové technologie pro uchování dat, jde o API pro webové úložiště a webové SQL databáze
- **Lepší formuláře** - vylepšení ovládacích prvků uživatelského rozhraní. Prakticky se jedná o nejrůznější posuvné jezdce, kalendáře pro výběr data, roletové nabídky, zatrhávací políčka, přepínače a spoustu dalších potřebných komponent. Problém může nastat v podpoře u různých prohlížečů. Nicméně velkou výhodou, stojící za zmínku, je zlepšená přístupnost. Web se stává dostupnější i pro uživatele s omezeními.
- **Vylepšená přístupnost** - implementace nových elementů, které přispívají k lepšímu popisu obsahu. Čtečky obrazovek dokáží takový obsah lépe zpracovat. Dalším rozšířením jsou atributy elementů, které určují jejich roli.

HTML5 je kompatibilní se všemi prohlížeči. Některé starší verze prohlížečů však mohou mít problémy s podporou některých technologií, důsledkem by ve většině případů mělo být pouze odlišné zobrazení. Použití nových elementů, např. audio, video, nebo canvas, přináší výhodu spojenou se zlepšenou přístupností, jež souvisí s integrací webové stránky uživatelem ve starším prohlížeči, s pomalým internetovým připojením, nebo třeba přes mobilní zařízení. (Hogan, 2011, s. 20)

4.5 CSS

CSS (Cascading Style Sheets), nazývané jako kaskádové styly, určuje vzhled webu. K tomuto jazyku můžeme přistupovat jako k souboru pravidel, která skrze své vlastnosti a hodnoty přidělují určitým HTML elementům jejich vzhled. Jazyk disponuje vlastnostmi pro **formátování textu** (velikost, barva písma atd.), pro **definici rozvržení**, které napomáhají při umístění elementů, dále pak vlastnosti pro řízení tisku. Neměli bychom opomenout možnost využít některou z dynamických vlastností, které umožňují elementy zobrazovat a skrývat.

Vznik CSS je spojován s rokem 1996, což znamená několik let po vzniku HTML. Zpočátku byly dokumenty formátovány pomocí značek HTML a jejich atributů. K požadovanému rozvržení prvků stránek byly využívány tabulky. Postupem času ale došlo k zvětšení objemu prezentovaných informací. To vedlo ke značnému zpomalení načítání stránek vlivem velkého množství použitých značek a také docházelo k nárůstu nákladů na režii serveru. CSS si s těmito problémy dokáže poradit. Jeho počátky byly spojeny s nekompatibilitou tehdejších prohlížečů, nicméně dnes už jsou běžným nástrojem při tvorbě webových stránek. V dnešní době již jsou velmi rozšířené preprocesory Sass a Less. Tyto jazyky jsou kompilované a poskytují vývojáři možnost využít proměnných, funkcí a spoustu dalších prostředků, které mu velice usnadní práci a dělají kód přehlednější. (David Procházka, 2011, s. 108)



Obrázek 3: CSS pravidlo stylu

CSS3

Jazyk prochází vývojem podobně jako HTML.

Výhody oproti předchozí verzi:

- **Lepší uživatelské rozhraní** - stejně jako u HTML5 snadněji dosáhneme požadovaného výsledku ve stylování jednotlivých elementů.
- **Pokročilé selektory** - možnost jednoduše identifikovat elementy dle různých kritérií a díky tomu učinit zdrojový kód mnohem přehlednější. Příkladem je indentifikace lichých a sudých řádků tabulky, všech vybraných zaškrtačacích políček nebo posledního odstavce ve skupině.
- **Lepší aplikace** - vývoj bohatších a interaktivnějších webových aplikací
- **Vizuální efekty** - napomáhají při přidávání stínů a přechodů elementům bez nutnosti spoléhat na obrázky na pozadí, nebo další zdrojový kód. Lze použít i transformace, které zajišťují vytvoření kulatých rohů, zkosení či otočení elementů.

(Hogan, 2011, s. 20)

Co se týče kompatibility, situace je velice podobná jako u již zmiňovaného HTML5.

4.6 Javascript

Jedná se o skriptovací jazyk. Za jeho zrodem stála v 90. letech 20. století společnost Netscape a byl vyvinut jako doplněk k jazyku Vizual Basic. Javascript je implementován ve všech moderních prohlížečích, tudíž má široké uplatnění, v dnešní době spíše v podobě nejrůznějších frameworků. Na počátku sloužil k validaci formulářů a drobné manipulaci s obsahem stránky, nicméně postupem času se stal mocným nástrojem k tvorbě klientských aplikací. Zapříčinil do jisté míry ústup zásuvného modulu Flash. (Ara Pehlivanian et al., 2014, s. 11)

Hlavní význam jazyka spočívá v zajištění potřebné interaktivity webových stránek. Příkladem jsou vysouvací navigační nabídky, transformace textu, dynamické přidání elementů do stránky atd. Limitující faktor je spojen s funkcionalitou jazyka, přesněji řečeno jde o jeho závislost na jiném rozhraní, nebo hostitelském programu, což je ve většině případů webový prohlížeč. Užívá se pro něj označení uživatelský agent (user agent), jelikož pracuje na straně klienta. Javascript tedy dokáže reálně vykonat jen to, co mu uživatel umožní. Do jisté míry tuto skutečnost ovlivní i horší podpora starších prohlížečů, nicméně v dnešní době se daří ve většině případů problém podpory skriptování eliminovat. Dalším omezením je garance bezpečnosti dat, která není plně zajištěna. Vstup ze strany klienta může nabývat nesmyslných rozměrů. S ohledem na tuto skutečnost je třeba brát zřetel na validaci dat na straně serveru. I přes zmíněné nedostatky se vyplatí Javascript využít k zajištění správného

chování webu, avšak je třeba pamatovat na to že prezentovaný obsah by měl být oddělený a měl by poskytnout uživateli hledanou informaci nezávisle na podpoře jazyka. (Steve Suehring, 2008, s. 24)

JQuery

JQuery je Javascriptový framework. Nabízí jednoduché vyhledávání elementů v DOMu, jejich modifikace a vytváření. Mezi další možnosti využití patří nepochybně také práce s událostmi, velké množství pokročilých funkcí pro práci s poli, podpora AJAXu a animací. (Richard Šerý, 2007)

Klady použití:

- usnadňuje práci s DOMem
- jednoduchost
- zajímavé funkce
- kvalitně zpracovaná dokumentace

(Richard Šerý, 2007)

Framework je díky svým přednostem vhodný pro menší a středně velké projekty.

4.7 Bootstrap

Bootstrap lze označit jako HTML, CSS a Javascript framework. Obsahuje sadu nástrojů usnadňující práci při vývoji frontendu webové aplikace. Jeho vznik byl spojen se sociální sítí Twitter, pro jejíž potřeby vznikl. Nabízí mnoho komponent, které usnadňují práci při tvorbě kvalitní webové prezentace. Pokud budeme konkrétnější, pravděpodobně největší předností je snazší vytvoření responzivního frontendu, tzn. zobrazitelného na různých rozlišeních displeje. Nicméně poskytuje i další řešení problémů, které úzce souvisí s vývojem, např. práce s typografií, tvorba layoutu, vytváření elementů uživatelského rozhraní a v neposlední řadě také zobrazení napříč různými platformami. Můžeme konstatovat, že díky tomuto frameworku je vývoj rychlejší a snadnější. (Martin Michálek, 2013)

4.8 Programovací jazyk PHP

PHP (Hypertext Preprocessor) je skriptovací jazyk s širokým využitím pro vývoj webových aplikací v interakci s HTML kódem. Kód skriptu psaného v PHP je spuštěn na straně serveru, což zamezí uživateli (klientovi) odhalení logické podstaty vykonávaných funkcí. Klient tímto docílí požadovaných výsledků, nicméně základní kód aplikace mu zůstává utajen. Za největší výhodu je považováno rychlé a snadné pochopení logiky jazyka začátečníkem, ačkoliv jazyk poskytuje spoustu možností i zkušeným programátorům. Mezi hlavní funkce patří sběr dat z formulářů, generování dynamického obsahu stránky nebo příjem a odesílání souborů cookies.

(Achour, Betz, Dovgal, 2014)

Roku 1994 jazyk vytvořil Rasmus Lerdorf pro svou potřebu, což v praxi znamenalo vývoj jednoduchého systému pro kontrolu přístupů na stránku. Sada skriptů, psaných v jazyku C a starajících se o chod systému, byla vydána jako Personal Home Page Tools, zkráceně PHP. Později jazyk zaznamenal velký rozmach, který přinesl i jeho rozvoj ve formě různých verzí. Důležitým milníkem je rok 1998, kdy byla k dispozici verze PHP 3.0 vyznačující se vyšší rychlostí a větší vybaveností než doposud vydané verze. Navíc byla dostupná i pro operační systém Windows. Vývoj pokračoval a v současné době, lze za poslední verzi brát PHP 7. Průběh vývoje jednotlivých verzí je charakterizován zejména vyšší poskytovanou rychlostí, menší pamětovou náročností a širšími možnostmi využití nejrůznějších funkcí.

(Petr Zajíc, 2004)

Proč jazyk používat?

- je jednoduchý na pochopení
- podporuje širokou řadu souvisejících technologií, formátů a standardů
- lze nalézt spoustu již řešených problémů
- dobře komunikuje s databázemi, např. MySQL
- je multiplatformní, provoz na webovém serveru na straně webhostingových poskytovatelů je tedy prakticky samozřejmostí

Možnosti použití:

- internetové obchody
- podnikové informační systémy
- redakční systémy
- diskusní fóra
- a mnoho dalších, např. dynamické webové prezentace

(Petr Zajíc, 2004)

Nette

Nette lze charakterizovat jako kvalitně objektově navržený framework pro PHP, který se setkává s velkou oblibou. Jeho autorem je český vývojář David Grudl. Umožňuje snadnější vývoj webových aplikací, proto na jeho základě byla vybudována řada velkých projektů, např. Slevomat, ČSFD apod. Své uplatnění ale nachází také u spousty menších webových stránek a e-shopů.

Framework je komplexní soubor vzájemně spolupracujících knihoven. Pokud přihlédneme blíže k Nette, můžeme konstatovat, že framework je stavěn na architektuře MVC, která je pro něj pojmenována jako MVP. Obsahuje následující komponenty, které obstarávají podobnou funkci jako u MVC. (David Čápka, 2014)

Architektura MVC (Model-View-Controller) dělí aplikaci na 3 logické části. To přináší možnost upravovat jednotlivé části samostatně, s co nejmenším dopadem na ostatní části. Jedná se o Model, View a Controller.

Model zahrnuje data a logiku aplikace. Příkladem jsou výpočty, databázové dotazy, validace apod. Přijímá parametry zvenčí a navrácí data ve formě výstupů. Model se nestará, odkud data v parametrech přišla a jakým způsobem budou výstupní data zformátována a vypsána.

View zobrazuje uživatelské rozhraní, jinak řečeno výstup pro uživatele. Obvykle jde o phtml šablonu, která obsahuje HTML stránku a tagy nějakého značkovacího jazyka. Skrze tento jazyk lze do šablony vkládat proměnné, eventuálně provádět iterace (cykly) a podmínky.

Controller se stará o tok událostí v aplikaci a obecně aplikační logiku. Zařizuje spojení dříve zmíněných komponent a tedy komplexnost celého systému jako funkčního celku.

(Borek Bernard, 2009)

Ze základu této architektury MVC vychází i již zmíněná architektura MVP. Liší se zejména pojmenováním jednotlivých komponent.

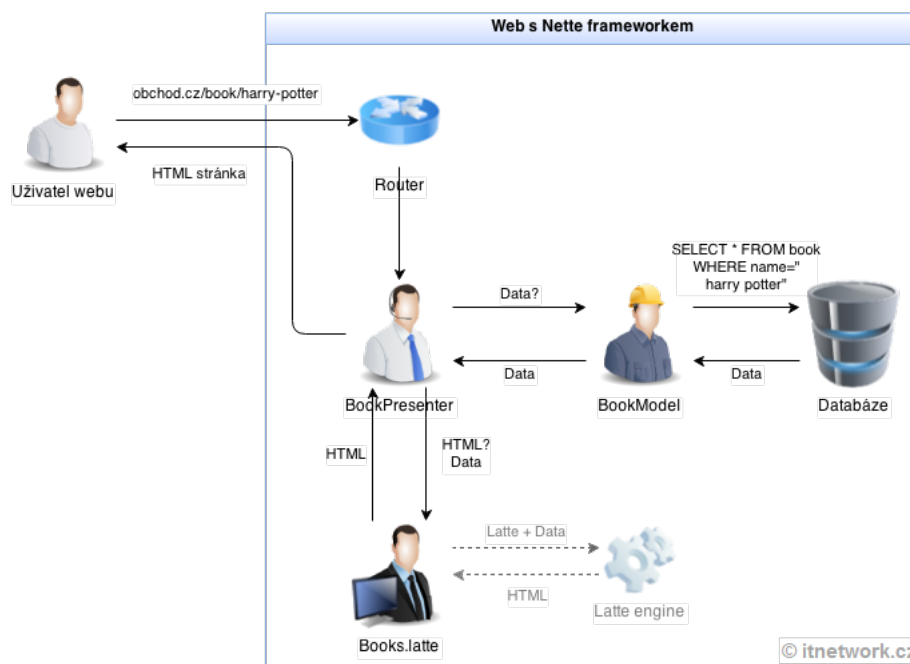
Presenter, řízení - komponenta, která zajišťuje komunikaci s uživatelem. Vstupem jsou parametry ze strany uživatele, navrácený výstup je v podobě HTML stránky. Dá se nazvat prostředníkem, podstata tohoto označení vyplývá ze samotného procesu zpracování parametrů, které jsou v první fázi předány modelům. Tím získá presenter data, která předá pohledům (šablony), poté jsou data začleněna do HTML kódu. V poslední fázi presenter HTML kód zašle do prohlížeče uživatele. Každý presenter disponuje několika svými šablonami pro různé akce, příkladem je přidání, detail aj.

Model, logika - zahrnuje logiku aplikace, tzn. práci s databází a různé výpočty. Každá datová entita má ve většině případů svůj model, např. uživatel, článek atd.

Pohledy, výstup - součástí jsou Latte šablony s HTML kódem. Latte šablony umožňují do HTML vkládat data z PHP s využitím speciálních značek.

Související modely a presentery (se šablonami) jsou začleňovány do tzv. **modulů**. Jedná se o balíček komponent pro určitou část webu.

Funkci MVP znázorňuje na konkrétním příkladu následující diagram, který zachycuje životní cyklus stránky.



Obrázek 4: Životní cyklus stránky

zdroj: <http://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette/>

Než dojdou hodnoty jednotlivých parametrů ze strany uživatele k presenteru, prochází přes síťové zařízení - router (směrovač), který podle URL dokáže rozeznat požadavek a zavolat příslušný presenter. Ten se postará o obsluhu dotazu. Poté už vše probíhá podle schématu popsaného výše ve funkcích jednotlivých komponent. (David Čápka, 2014)

Vyjma komponenty MVC jsou další součástí Nette také tzv. Filtry. Ty lze brát jako malé pomocné funkce pro šablony, které se starají o formátování výstupu. Prakticky jde např. o formátování data a spoustu dalších využití.

Pojmem, který bychom měli také zmínit, je **sandbox**. Označuje kostru webové aplikace, do které jsou přidávány jednotlivé stránky aplikace. (David Čápka, 2014)

Důvody použití frameworku:

- rozšíření funkcí standardních knihoven PHP
- znovupoužitelnost již dříve implementovaných funkcí
- vytvoření aplikace v kratším čase
- zjednodušení kódu aplikace, V souvislosti s tímto faktem je aplikace snadno udržovatelná.

(David Čápka, 2014)

Výhody:

- dokonalé zabezpečení - umožňuje zamezit zneužití bezpečnostních děr
- ladící nástroje - poskytují širokou škálu ladících aplikací, které umožní vývojáři vypořádat se s chybami
- výkon
- promyšlený objektový návrh - využití verze PHP 5
- a spoustu dalších

(Nette, 2015)

4.9 Další nástroje a techniky

K uchování dat poslouží databáze MySQL. Pro návrh UML diagramů bude použit nástroj Visual Paradigm se školní licencií.

MySQL

Jedná se o databázi relační. Způsobem uložení dat jsou tabulky. Každá z nich obsahuje záznamy jednoho typu, v našem případě např. vozidla, majitele a jiné. Na řádcích tabulky jsou uloženy jednotlivé položky a sloupce pak reprezentují atributy (vlastnosti) položek. Atributy jsou určeny jejich datovým typem podle toho, o jaký typ dat se jedná. Za příklad lze uvést textový řetězec, číslo, datum apod. Veškeré záznamy tabulky mají přiřazen unikátní identifikátor, pomocí kterého lze k hodnotě přistupovat. Jak je ze slova relační patrné, mezi tabulkami a entitami jsou vztahy, které je k sobě vážou.

Databáze jako celek je označována zkratkou RDBMS (Relation DataBase Management System), český překlad zní systém řízení báze dat. Tento systém neslouží pouze jako prostředek k uložení dat. Pomocí jazyka SQL dotazů lze nad daty vykonávat nejrůznější operace a dosáhnout požadovaného výstupu. Nejčastěji jde o selekci záznamů dle požadovaných kritérií, dále je možné data editovat, mazat apod. Jazyk SQL je pro jeho uživatele jednoduchým na pochopení, jelikož jeho formulace vychází ze srozumitelných vět pro člověka. Princip užití bude vysvětlen v části implementace aplikace na praktické ukázce. (David Čápka, 2012)

5 Vlastní práce

Obsahem této části práce bude analýza podniku a modelování funkčního řešení evidence vozidel. Výstup následně poslouží k implementaci prototypu webové aplikace dle požadavků zadavatele. Prezentované poznatky budou vycházet z teorie popsané v předchozí části práce.

5.1 Analýza firmy

Představení firmy

Firma Perfekt - cars s.r.o., jejíž vznik se datuje k roku 1996, je umístěna v Jihomoravském kraji, přesněji nedaleko města Znojma. Jedná se o autobazar, který nabízí výkup vozidel hotově, dovoz na objednávku, ale také možnost využít vozidlo jako protihodnotu při nákupu nového. Mimo jiné firma zprostředkovává havarijní pojištění, povinné ručení a také financování koupě vozidla formou spotřebitelského úvěru.

Současný stav a prostředí firmy

Umístění autobazaru lze označit jako strategické, nachází se u hlavního tahu na Prahu. Společně s vstřícným chováním všech pracovníků tento fakt lze brát jako významnou konkurenční výhodu vůči firmě podobného zaměření, která se nachází nedaleko. Velkou výhodou je také rozlehlá plocha využitelná pro vystavení prodávaných vozů, proto je autobazar rozdělen na dvě části, pro vozidla osobní a pro vozidla nákladní, nebo staršího data výroby. Většina zákazníků pochází z okolí firmy, avšak tato skutečnost není pravidlem, jelikož je nabídka vozů dostupná i na webových stránkách.

Aktuálně firma vlastní dva stolní počítače a tiskárnu v místě výkonu práce, mimoto je využíván notebook pro vyřizování korespondence přímo z domova. Počítače jsou propojeny do místní sítě s přístupem na internet a obsluhovány třemi zaměstnanci, kteří umí využívat moderní technologie a nemají s obsluhou větší problémy, nicméně o správný chod zařízení je dbáno externě formou outsourcingu. Podobně je tomu u webových stránek, na kterých si zákazník může jednoduše zjistit detailnější informace o nabízených vozech s možností vozy filtrovat dle požadovaných parametrů. O jejich údržbu a vývoj se stará marketingová agentura, avšak nejaktuálnější nabídka vozů je zadávána zaměstnanci.

Co se týče programového vybavení, klasická kancelářská sada a základní software zajišťující bezpečný chod zařízení jsou samozřejmostí. Hlavním nástrojem pro operace spojené s prodejem se stal program, nazývaný jako Car IS, který je nainstalován na jednom z počítačů. Car IS slouží jako evidence vozidel a informací s nimi spojenými. Systém umožňuje export dokumentů na základě informací vedených o vozu, jeho majiteli a technickém stavu v době přijetí. Některé vstupní informace jsou automaticky doplněny na základě vybraných parametrů vozidla, jež jsou zadány uživatelem,

v našem případě vedoucím autobazaru. Příkladem je identifikační číslo VIN, podle kterého lze zjistit spoustu cenných informací. Mechanismus získává potřebné data z informačního systému externích registrů vozidel, jedním z nich je asi nejznámější systém pro prověřování vozidel Cebia.

Finální výstup systému má formát pdf a je vhodný k tisku dokumentu s požadovanou strukturou. Konkrétně se jedná o smlouvu o zprostředkování, kupní smlouvu a nebo výpis technického stavu, který je důležitý při převzetí vozidla kupujícím.

Program mimo jiné poskytuje spoustu dalších funkcí.

Funkce:

- archivace prodaných vozidel
- pokladní kniha
- ekonomika podniku
- evidence zájemců
- export inzerce z internetu
- formulář
- cash-flow
- inventura
- interní pošta
- výpis historie ojetých vozidel
- a některé další funkce

Problémová doména

Současné řešení systému neodpovídá představám uživatele. Jeho integrace ve zmiňované podobě pro firmu znamená spoustu nevyužitých funkcí a zbytečně vysoké poplatky za licenci softwaru. Tato problematika byla ze strany majitele autobazaru několikrát řešena s tím, že byl vznesen návrh na zúžení funkcionality systému s následným snížením poplatků. Požadavek ovšem nebyl ze strany poskytovatele naplněn. Ačkoliv se nepodařilo najít kompromis mezi uživatelem a dodavatelem, firma nadále prodlužuje licenci, jelikož se systém stal nezbytnou součástí běžné firemní praxe. Poplatky za užívání jsou stále placeny paušálně každý půlrok. Tato skutečnost má negativní dopad na náklady firmy a z nich vyplývající neefektivní alokaci zdrojů.

Požadavky

Specifikace požadavků na nový systém vychází z již popsané problematiky neefektivního čerpání zdrojů. Rozsah stávajícího řešení systému neodpovídá množství využívaných funkcí. Po konzultaci s vedoucím autobazaru byly upřesněny požadavky na funkcionalitu. Výsledná webová aplikace bude generovat prakticky totožné výstupy, ačkoli dozná značných úprav.

Funkční požadavky

Uživatel musí mít možnost se do aplikace přihlásit a následně odhlásit.

V důsledku umístění aplikace na internetu by měly být důležité informace chráněny před přístupem zvenčí.

Aplikace musí umožnit uživateli evidovat údaje o nově přijatém vozidle.

Požadavkem zadavatele bylo snížit množství evidovaných informací oproti stávajícímu řešení a výsledkem by měla být aplikace na míru s databází menšího rozsahu. Prezentace údajů je realizována tabulkou formou výpisu z databáze.

Záznamy je možné editovat a mazat

Při kliknutí na detail vozidla je uživateli zpřístupněn předvyplněný formulář s údaji, které pak lze jednoduše aktualizovat. Před smazáním je uživatel nejdříve dotázán, jestli chce operaci opravdu provést.

Aplikace umožní uživateli filtrovat záznamy dle zadaných parametrů

Filtrovat lze podle několika parametrů, jsou to: výrobce, model, najeté kilometry, SPZ, jméno a příjmení majitele vozidla. Uživatel je upozorněn na aplikaci filtru a může filtr zrušit, nebo si vyčistit formulář pro zadávání kritérií filtrace.

Vozidlo lze přesunout do archivu

Archiv zobrazuje informace o vozidlech totožně jako výpis. Rozlišení archivních záznamů probíhá na základě informace o stavu prodeje vozidla. Znamená to, že pokud je nastaven vozidlu stav Vráceno nebo Prodáno, vozidlo se automaticky vloží do archivu. Nicméně uživatel má možnost záznam přenést jedním kliknutím na tlačítko archivu u každého vozidla, opačně vozidlo lze vrátit z archivu zpět do prodeje.

Na základě vedených údajů musí být uživateli umožněn tisk výpisu technického stavu a smluv.

Kontrétně se jedná o Kupní smlouvu a Smlouvu o zprostředkování. Do smluv je doplněna částka určená na základě ceny vozidla požadované ze strany majitele, provize autobazaru za prodej a daně z přidané hodnoty. Před tiskem smluv je možné text smlouvy ještě upravit. Výstup je generován ve formě PDF a uživatel si sám zvolí, kam chce soubor uložit.

Nefunkční požadavky

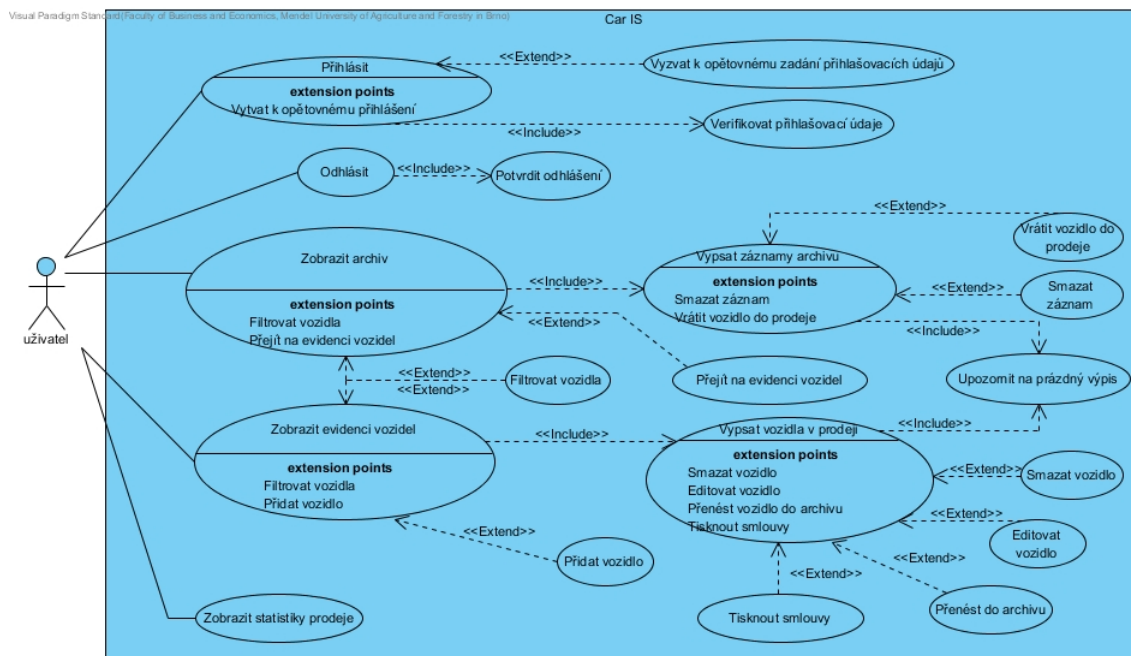
Pro spuštění aplikace bude primárně používán webový prohlížeč Chrome, nicméně uživatel by neměl mít problém s podporou také na jiných platformách, např. Internet Explorer (verze 9 a vyšší), Mozilla Firefox apod. S ohledem na tuto skutečnost by neměl být opomenut požadavek na multiplatformní vývoj aplikace. Dostupnost aplikace prostřednictvím internetu bude zajištěna na webhostingu, který umožňuje využít databázi MySQL pro uložení dat.

Co se týče prostorové náročnosti, na straně serveru aplikace zabere zhruba 150 MB paměti. O efektivní načítání souborů se postará použitý framework, který umožňuje přednačítání šablon, jež jsou plněny daty z databáze. Celková náročnost zobrazení by tedy měla být minimální a měla by zaručovat uživateli možnost aplikaci spustit i na starších zařízeních. Neméně důležitým požadavkem je responzivita aplikace, tzn. možnost užívání aplikace na zařízeních s různou velikostí a rozlišením displeje.

Smlouvy a výpisy generované aplikací musejí mít podobu odpovídající standardu, který určuje stávající forma výstupů.

5.2 UML modelování

Use Case Diagram



Obrázek 5: Use Case Diagram aplikace

Diagram znázorňuje jednotlivé funkce systému, které je možné vykonat. Hraniče systému určuje informační systém evidence vozidel. Vně hranice vystupuje aktér, kterým byl zvolen uživatel a v konkrétním případě autobazaru se bude jednat o jeho zaměstnance. Jednotlivé případy užití značí činnosti proveditelné ze strany zaměstnance. Po správném zadání přihlašovacích údajů je uživateli zpřístupněna možnost výběru několika operací, v opačném případě je vyzván k opětovnému zadání údajů pro přihlášení.

Případy užití:

1. Zobrazit archiv

Po zobrazení archivu dojde k výpisu záznamů vozidel se stavem vráceno nebo prodáno. Pokud nebyla žádná vozidla nalezena, uživatel je upozorněn na prázdný výpis. S každou položkou výpisu lze manipulovat, uživatel může záznam smazat, nebo vrátit zpět do prodeje. Tato operace se projeví na výpisu evidovaných vozidel v prodeji. Uživatel dále může vozidla filtrovat dle zadaných kritérií. Totožná filtrace se vyskytuje i v evidenci vozidel a její princip bude popsán později. Poslední možností je přímý přechod k evidenci vozidel.

2. Zobrazit evidenci vozidel

Podrobněji je tento případ užití popsán v detailním rozboru níže. Jakmile už-

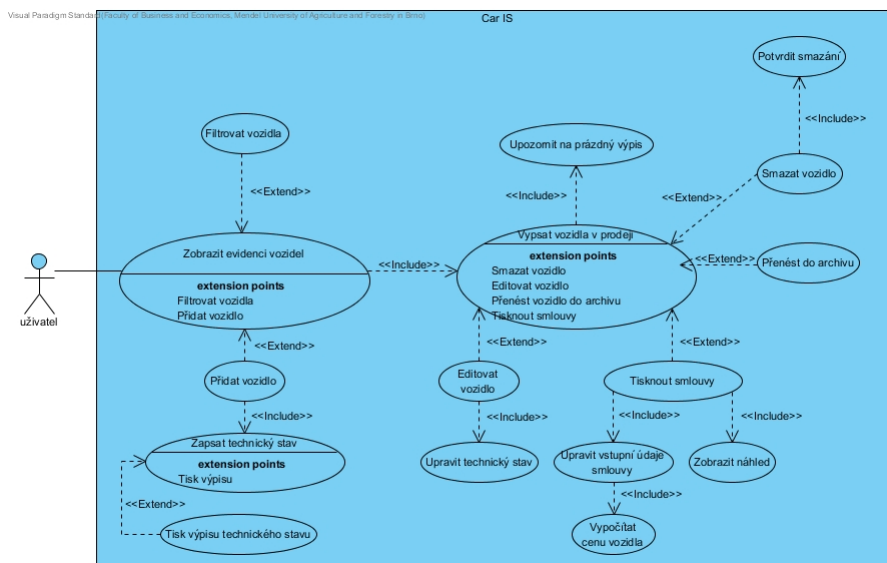
vatel zobrazí evidenci, podobně jako u archivu jsou vypsány všechny záznamy vozidel s příslušnou vlastností - v prodeji. Princip výpisu a manipulace se záznamy se do velké míry podobá archivu, liší se v operacích, které může uživatel s daty provést.

Za nejvýznamnější lze označit tisk smluv. Před samotným tiskem (exportem) smlouvy uživatel upraví vstupní údaje, do kterých spadá i cena vypočtená ze zadaných parametrů. V dalším kroku je uživateli zobrazen náhled do smlouvy, ve kterém lze modifikovat text smlouvy a její strukturu. Poté již dochází k exportu smlouvy.

Přidat vozidlo umožní uživateli vložit nový záznam do evidence. Po uložení základních dat o vozidle a osobách s ním spojených následuje přechod k podrobnému zápisu technického stavu vozidla. Poté je možné vytisknout výpis, nicméně této události musí předcházet uložení záznamů, které zapříčiní doplnění evidovaných údajů do výpisu. S dokončením předchozích fází přidávání vozidla je zpřístupněn i tisk smluv k danému záznamu.

Vyjma přidání systém umožňuje i editaci vozidla, pod kterou spadá i editace jeho technického stavu.

Vozidlo lze smazat, po uživateli je požadováno potvrzení operace, a také přesunout do archivu, např. v případě dokončení prodeje.

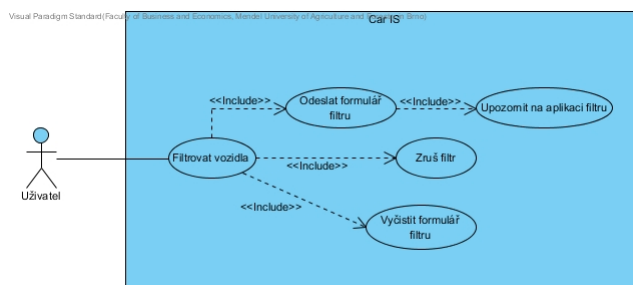


Obrázek 6: Detailní Use Case Diagram evidence vozidel

3. Zobrazit statistiky prodeje

V současné době není tato možnost zdokumentována. Jejím vývojem se bude autobazar zabývat v budoucnu.

Mimo jmenované operace se může uživatel z aplikace odhlásit. Tato volba musí být před provedením potvrzena.



Obrázek 7: Use Case Diagram filtrace

Filtrace umožňuje uživateli zúžit výpis záznamů dle zadaných kritérií. V okamžiku aplikace filtru je uživatel upozorněn. Dalšími operacemi jsou zrušení filtru a vyprázdnění formuláře pro zápis kritérií.

Scénář

Přihlásit

Uživatel se přihlašuje do aplikace pro evidenci vozidel. Přihlašovací údaje obdržel již při zavádění systému do provozu. Registrace není v systému řešena.

Tabulka 1: Scénář přihlášení do aplikace

	Actor	System
1	Případ užití začíná ve chvíli, kdy uživatel spustí aplikaci ve webovém prohlížeči.	
2		System zobrazí uživateli přihlašovací formulář.
3	Uživatel zadá přihlašovací údaje a odešle je ke zpracování.	
4		System ověří správnost údajů.
4.1		Dochází k přesměrování na domovskou stránku aplikace.
4.2		KDYŽ není formulář vyplněn, nebo jsou zadány nesprávné údaje, system uživatele upozorní.

Tisk smlouvy

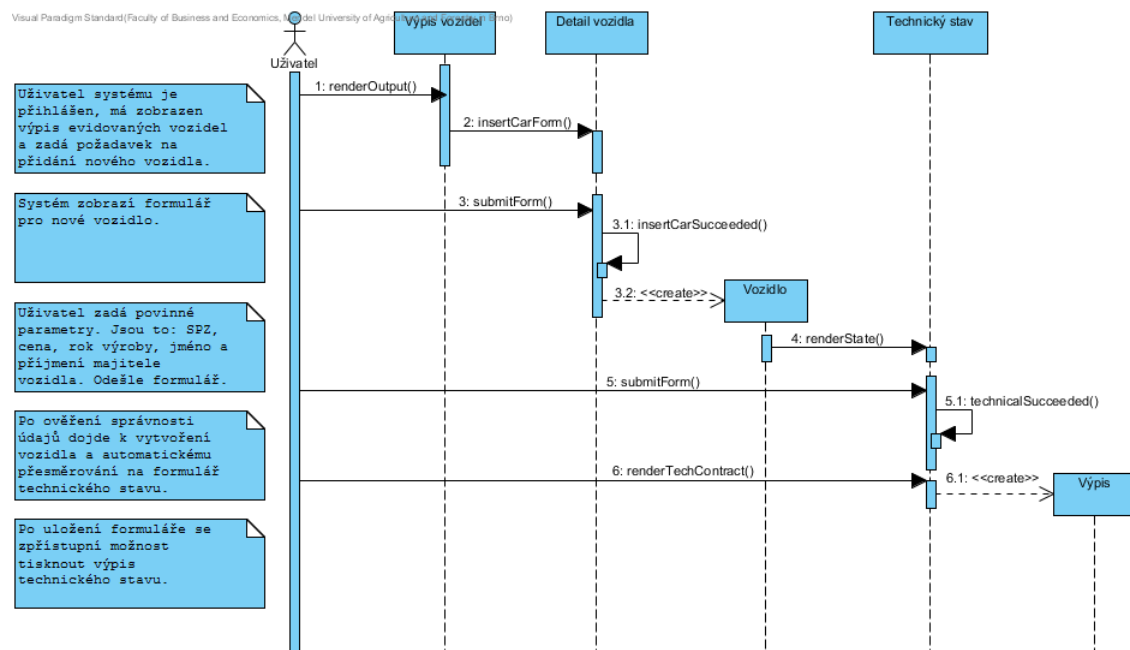
Následující scénář popisuje proces exportu smlouvy u již evidovaného vozidla. U nového vozidla předchází tisku několik kroků zadávání informací ze strany uživatele.

Tabulka 2: Scénář tisku smluv

	Actor	System
1	Uživatel na domovské stránce zvolí možnost Evidence vozidel.	
2		Systém zobrazí výpis evidovaných vozidel.
3	Uživatele klikne na ikonu tisku u daného vozidla.	
4		Dochází k přesměrování na stránku Smlouvy a systém vykreslí formulář k úpravě informací smlouvy.
5	Uživatel doplní potřebné položky a odešle formulář kliknutím na tlačítko Tisknout smlouvu.	
5.1	KDYŽ požaduje přepočítání celkové ceny, klikne na příslušné tlačítko.	
6		Systém doplní informace do smlouvy a zobrazí náhled smlouvy.
7	Uživatel zkontroluje smlouvu a potvrdí tisk.	
8		Systém generuje požadovaný výstup ve formě dokumentu PDF.

Sekvenční diagram

Diagram znázorňuje interakci mezi jednotlivými kroky zadávání nového vozidla až po vytvoření výpisu technického stavu. Pro zdokumentování diagramu byly zvoleny poznámky umístěné na levé straně diagramu.

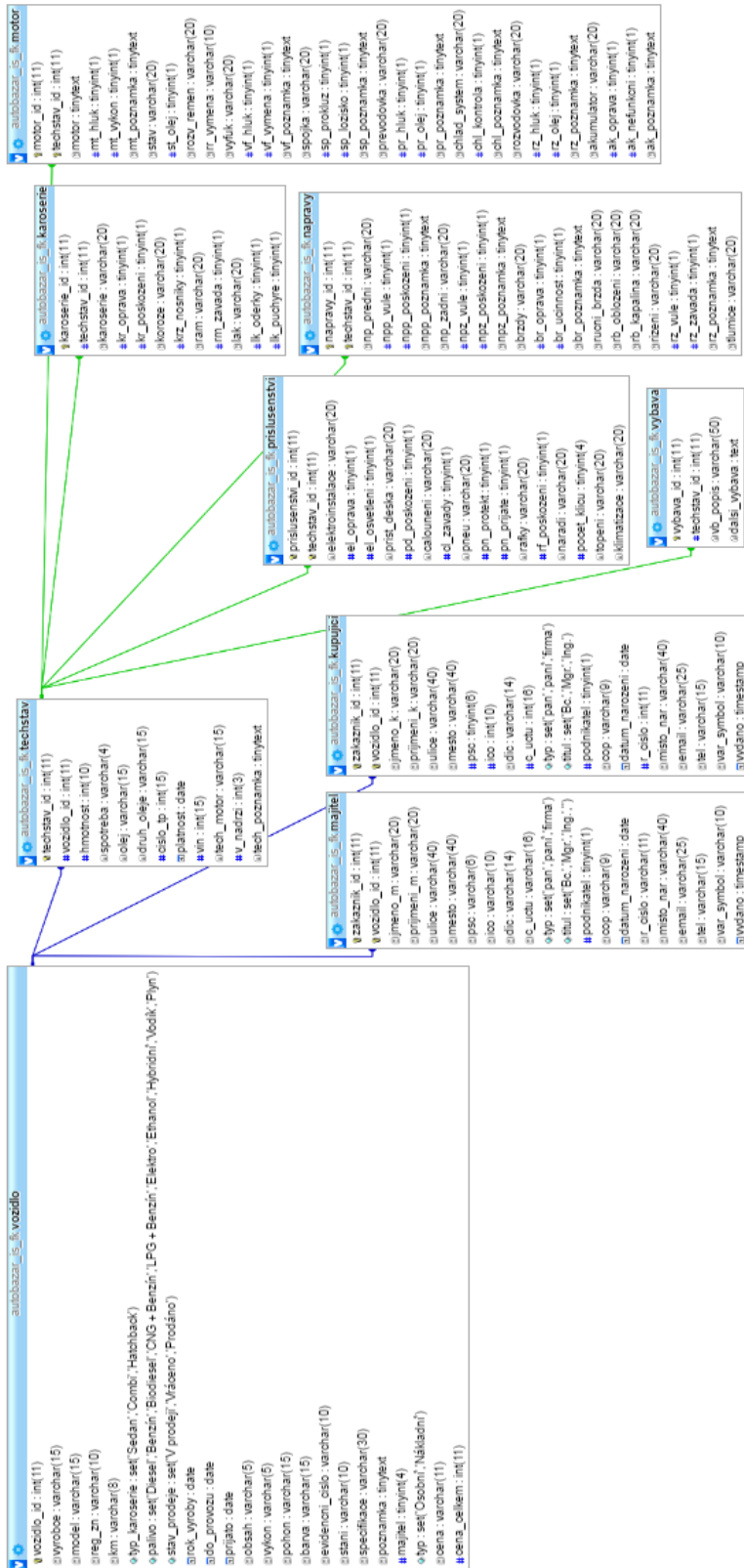


Obrázek 8: Sekvenční diagram evidence vozidel

ERD diagram

Entitně - relační diagram popisuje datovou strukturu aplikace, přesněji schéma databáze pro uložení dat. Obdélníkové útvary reprezentují jednotlivé entity, které se vyznačují specifickými vlastnostmi (atributy). Jejich propojení pomocí čar zachycuje vztahy mezi entitami. Obvykle jsou relace doplněny o parcialitu n nebo kardinalitu.

Databáze webové aplikace se skládá z devíti tabulek (entit), které nesou informace o vozidlu, jeho technickém stavu, majiteli a nakonec i kupujícím. Technický stav je dále rozdělen na menší celky, které se týkají stavu konkrétních součástí vozidla. Každý záznam je od ostatních rozlišen unikátním identifikátorem, který nazýváme primární klíč. Jak je z diagramu zřejmé, všechny tabulky spojují cizí klíče, které jsou vázány na jednotlivé identifikátory entit.



Obrázek 9: ER Diagram

5.3 Implementace

Latte šablony

O grafickou prezentaci dat se starají Latte šablony, které jsou nastýlovány pomocí CSS stylů.

Ukázka udává strukturu stránky přihlášení do aplikace.

```

1   {block content}
2
3   <div class="login-page">
4
5       <h1>Perfekt-cars <span class="color-red">IS</span>
        /h1>
6
7       <div class="jumbotron vertical-align">
8           <div class="container">
9               <div class="row login-form">
10                  <div class="col-md-4 col-md-offset-4">
11                      
13
14                          {control loginForm}
15
16                          {ifset $error}
17                              <div class="login-err">
18                                  <p>Zadejte správné
19                                      řspihlaovací údaje!</p>
20                                  </div>
21                              {/ifset}
22                          </div>
23                      </div>
24                  </div>
25              </div>
26          </div>
27      {/block}

```

Jak je z ukázky zdrojového kódu zřejmé, ve struktuře stránky jsou hojně používána Latte makra, která manipulují s DOMem stránky. Celý obsah je obalen v bloku označeném jako content, který je znovupoužitelný v jiných šablonách a lze na něj jednoduše odkazovat. Za povšimnutí stojí vykreslení formuláře pomocí makra *control loginForm*. Formulář je komponentou příslušného presenteru (LoginPresenter.php). Možností využití maker je spousta. V uvedeném kódu ošetřuji zobrazení

chybového hlášení pomocí podmínky *ifset*. Pokud uživatel nesprávně vyplní přihlašovací údaje, příslušný presenter vyhodnotí podmínku za splněnou a následně je vykreslen požadovaný HTML blok.

Napojení na databázi a dotazování

Nastavení databáze a souvisejících PHP direktiv je řešeno externě v souboru *config.neon*. Soubor obsahuje autorizační údaje a další informace. Následné připojení databáze probíhá v konstruktoru presenteru.

```
1     public function __construct(Nette\Database\Context
2         $database) {
3         $this->database = $database;
4     }
```

K manipulaci s daty z databáze se používají SQL dotazy.

```
1     public function renderOutput($filterQuery) {
2
3         $this->template->filter = False;
4         if (isset($filterQuery)) {
5             $this->template->filter = True;
6         }
7
8         $this->template->cars = $this->database->query('
9             SELECT V.vozidlo_id, M.jmeno_m, M.prijmeni_m, K.
10              jmeno_k, K.prijmeni_k, V.vyrobce, V.model, V.km,
11              V.reg_zn, V.rok_vyroby, V.prijato
12             FROM majitel M, vozidlo V, kupujici K~           WHERE
13              M.vozidlo_id=V.vozidlo_id
14              AND M.vozidlo_id=K.vozidlo_id
15              AND V.stav_prodeje="V~prodeji"' . $filterQuery);
16     }
```

Funkce slouží k výpisu záznamů evidovaných vozidel v prodeji. Pokud je aplikován filtr dochází k selekci záznamů podle zadaných parametrů. Výstup je vykreslen v příslušné Latte šabloně.

Presentery

Ukázka presenteru pro přihlašovací stránku je umístěna v přílohách.

Presenter se skládá z funkcí ošetřujících:

- vytvoření komponenty formuláře
- úspěšné odeslání formuláře a následné přesměrování na domovskou stránku aplikace
- upozornění při neúspěšném odeslání formuláře

Export smluv

Jednotlivé smlouvy mají klasickou HTML strukturu a jejich grafickou podobu lze upravit pomocí CSS stylů. Údaje z databáze jsou doplněny příslušným presenterem (PrintPresenter). Export dokumentů obsluhuje knihovna mPDF, která z HTML struktury vytvoří PDF dokument. Knihovna je volně dostupná na internetu.

Mobilní verze

Pro dosažení responzivity webu byl využit Bootstrap Grid, který má sloupcovou strukturu.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4				
span 4				span 8								
span 6						span 6						
span 12												

Obrázek 10: Bootstrap mřížka

Praktický příklad poukazuje na kontejner s řádkem o dvou sloupcích stejné šířky. Ve sloupcích jsou umístěna tlačítka výběru na úvodní obrazovce. V mobilní verzi se sloupce skládají pod sebe.

```

1  <div class="container">
2    <div class="row">
3      <div class="col-md-6">
4        <a href="{link_Output:output}">
5          <button type="button" class="btn btn-default
6            btn-block btn-lg">
7            <i class="fa fa-car fa-3x" aria-hidden="true"
8              ></i>
9            <span class="description">Evidence úvoz</span
10           >
11          </button>
12         </a>
13       </div>
14      <div class="col-md-6">
15        <a href="{link_Archive:archive}">
16          <button type="button" class="btn btn-default
17            btn-block btn-lg">
18            <i class="fa fa-archive fa-3x" aria-hidden="
19              true"></i>
20            <span class="description">Archiv úvoz</span>
          </button>
        </a>
      </div>
    </div>
  </div>

```

Aplikace na internetu

Aplikace je dostupná na doméně <http://sdevelop.cz/>.

Tabulka 3: Přihlašovací údaje

Uživatel	Heslo
perfektcars	auto123

Testování

Aplikace byla testována během vývoje procházením jednotlivých kroků evidence a následným zafixováním nalezených chyb. Ve většině případů se jednalo o chybné předání identifikátoru (ID) daného vozidla. Při přesunu mezi jednotlivými položkami v menu docházelo k padání aplikace. Následně byly chyby logovány pomocí tzv. laděnky, kterou poskytuje framework Nette jako ladící nástroj. S pomocí tohoto nástroje byl každý problém rychle odhalen a snadno vyřešen.

Finální verze funkční aplikace byla posléze zpřístupněna uživateli na internetu. Došlo k testování v běžném provozu a bylo zjištěno několik nedostatků. Většina z nich neměla na správný chod aplikace vliv, nicméně v plném provozu je třeba docílit efektivity procesu získávání požadovaných vstupů, aby byla možná jejich transformace na výstupy v co nejkratším čase.

Asi nejvýraznějším nedostatkem byla částečně nefunkční filtrace. Problém spočíval v absenci podmínek pro některá formulářová pole filtrace. Pro zajištění správného chodu bylo zapotřebí pouze upravit konkrétní presenter (`OutputPresenter.php`). Další nedostatek, na který bylo upozorněno, se zakládal na odhlášení z aplikace po nezáměrném kliknutí na příslušné tlačítko. Úprava byla velice jednoduchá, stačilo přidat do šablony javascriptovou událost potvrzení vyvolanou po kliknutí na tlačítko. Podobná situace nastávala u mazání jednotlivých záznamů.

Následovaly úpravy, které se týkaly převážně frontendu aplikace.

- změna povinných údajů při zadávání nového vozidla
- do výpisu vozidel přidáno tlačítko pro přímý přesun k tisku smluv
- přidána upozornění pro uživatele nabádající k uložení dat před tiskem
- přidán zpětný odkaz do náhledu před tiskem

6 Diskuse

6.1 Přínosy aplikace

Nejvýznamnějším přínosem oproti stávajícímu řešení je snížení provozních nákladů autobazaru. Jak již bylo zmíněno, tyto náklady plynuly z nutnosti zakoupení licence, která byla prodlužována každého půl roku. Základní funkcionalita systému evidence zůstala zachována, nicméně výrazné změny zaznamenala struktura vedených informací. Dle požadavků zadavatele se podařilo zredukovat množství evidovaných údajů. Tento fakt byl přínosem při implementaci responzivního designu. Za výhodu lze označit také dostupnost aplikace prostřednictvím internetu. Narozdíl od dosavadního řešení, instalovaného lokálně v místě výkonu, disponuje uživatel možnostmi provést potřebné operace kdykoliv a odkudkoliv, pokud má k dispozici připojení k síti internetu.

6.2 HW a SW infrastruktura

Jako poskytovatel webhostingových služeb a registrace domén byla zvolena firma INTERNET CZ, která vystupuje pod obchodním názvem FORPSI. Následně byl vybrán základní balíček služeb Easy, který obsahuje vše potřebné k zpřístupnění aplikace prostřednictvím internetu. Přesněji registraci domény, neomezený prostor pro web, podporu PHP 7.0, neomezenou databázi dle výběru, zálohování webu a spoustu dalších funkcí. Připojení k internetu se dá považovat za nejdůležitější kritérium pro plynulý chod aplikace. Doporučil bych zajištění externí firmou, která má v místě výkonu nejstabilnější připojení. Pro využívání aplikace mimo sídlo firmy je dostačující i mobilní připojení na úrovni Edge, avšak uživatel si bude muset na některé operace déle počkat.

Nároky na hardware zařízení jsou minimální a je spíše na uživateli, na jakém stroji aplikaci spustí. V úvahu připadají i mobilní zařízení a tablety. V našem případě uživatel disponuje stolními počítači, lze tedy konstatovat že s plynulým chodem aplikace nebude mít uživatel větších problémů.

Doporučené softwarové vybavení odpovídá prohlížeči Chrome, nicméně pro běžné užívání aplikace uživateli postačí předinstalovaný webový prohlížeč dle operačního systému. K zobrazení výstupu ve formě PDF je nejvhodnějším programem Adobe Reader, který umožňuje čtení a tisk dokumentů.

Aplikace byla testována hned na několika zařízeních a výsledky potvrdily skutečnost, že je aplikace správně optimalizována pro širokou škálu zařízení.

6.3 Ekonomické zhodnocení

Z hlediska časové náročnosti realizace je třeba rozdělovat projekt do více etap vývoje od definice požadavků po samotnou realizaci prototypu a jeho testování.

Výsledná cena projektu je tedy odvozena z kalkulace nákladů na:

- konzultace se zadavatelem
- návrh a vytvoření struktury databáze
- implementace aplikace - interní charakter aplikace, tzn. není dbáno na grafickou prezentaci, zadavatel ušetří peníze za grafické práce

Pokud firma po testování přistoupí k uváděnému řešení, ušetří část nákladů na implementaci aplikace. Pro běžné používání bude nutné aplikaci drobně modifikovat, jelikož se jedná o prototyp. Dá se předpokládat, že provozní náklady by měly dosahovat celkové výše rovnající se poplatkům za internetové připojení, které jsou již delší dobu placeny, společně s nově vznikajícími náklady na paušální poplatky poskytovateli webhostingu a na registraci domény. Ty se pohybují okolo 320 Kč. Správu a údržbu aplikace zajistí formou outsourcingu firma Inflexity s.r.o., které bude vyplácena hodinová sazba za odvedenou práci, avšak tyto náklady jsou očekávány jen v případě úprav funkcionality nebo implementace rozšíření.

6.4 Budoucí vývoj

Do budoucna je plánován vývoj s cílem dalšího snížení časové náročnosti jednotlivých procesů evidence. Konkrétně se jedná o procesy spojené se zadáváním nových záznamů do aplikace. V praxi to znamená zajištění napojení aplikace na veřejné registry vozidel, které poskytují velké množství informací o vozech. Výsledkem by mělo být automatické předvyplnění některých údajů a usnadnění práce uživatele.

Důležitým bodem je také implementovat plynulejší přechod mezi jednotlivými kroky zadávání nových záznamů do evidence. Uživateli by tato změna umožnila zapsat údaje o vozidle v libovolném pořadí a přesouvat se mezi jednotlivými položkami v menu bez omezení.

Dalším vývojem by mohl být sběr dat o prodejkách. Přínos pro firmu by spočíval v možnosti dohledat si statistiky prodeje v závislosti na technických parametrech vozidla a dále třeba na ročním období. Na domovské stránce aplikace je pro toto rozšíření připraveno tlačítko Statistiky prodeje.

7 Závěr

V rámci bakalářské práce byla provedena analýza firemního prostředí autobazaru, která odhalila nedostatky týkající se stávajícího řešení systému pro evidenci vozidel. Tyto nedostatky byly shrnuty a popsány v problémové doméně, na jejímž základě došlo po konzultaci s vedoucím firmy ke specifikaci požadavků na nový systém. Následovalo vytvoření návrhu nového informačního systému za pomoci UML diagramů. Diagramy byly modelovány s přihlédnutím na dříve stanovené funkční požadavky a popisují kompletní funkcionalitu systému. Implementaci předcházelo stanovení datové struktury, která vedla k vytvoření schématu databáze vhodné k uchování potřebných dat. Výsledná realizace má po předchozí domluvě podobu webové aplikace, přesněji prototypu, který byl testován v běžném provozu. Pokud se majitel autobazaru v budoucnu rozhodne, mohla by aplikace po drobné úpravě nahradit dosavadní řešení systému.

Teoretická část práce vysvětluje pojmy, které úzce souvisí s vývojem informačních systémů. Dále jsou popsány některé technologie využívané při vývoji moderních webových aplikací. Vlastní práce se skládá hned z několika částí určených pro analýzu, návrh a implementaci webové aplikace. Důležitou součástí analýzy je specifikace požadavků, která posloužila k následnému návrhu Use Case diagramů a popisu scénářů jednotlivých Use case. Návrh obsahuje také sekvenční diagram k znázornění interakcí uvnitř systému a ER diagram pro grafické znázornění datové struktury aplikace. V implementační části se nachází bloky zdrojového kódu a souhrn poznatků z testování aplikace. Ve své podstatě lze tuto část brát jako dokumentaci aplikace. Na konci praktické části jsou diskutovány přínosy aplikace, ekonomické zhodnocení, budoucího vývoj a doporučená hardwarová a softwarová infrastruktura aplikace.

Závěrem lze konstatovat, že soudě podle hladkého průběhu spolupráce a spokojenosti zadavatele výsledný prototyp aplikace, který vycházel z pečlivě zpracovaného návrhu, splnil požadavky dané zadáním práce.

8 Seznam zdrojů

- ACHOUR, M., BETZ, F., DOVGAL, A. *PHP: Introduction - Manual*. 2014 [online]. [cit. 2015-12-18].
Dostupné z: <http://php.net/manual/en/introduction.php>.
- ARLOW, J., NEUSTADT, I. *UML 2 a unifikovaný proces vývoje aplikací*. 2. aktualizované a doplněné vydání.
Brno: Computer Press, 2011. 568 s. ISBN 978-80-251-1503-9.
- BERNARD, B. *Úvod do architektury MVC*. 2009 [online]. [cit. 2016-10-29].
Dostupné z: <https://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>.
- CASTRO, E., HYLOP, B. *HTML5 a CSS3*.
Brno: Computer Press, 2012. 140 s. ISBN 978-80-251-3733-8.
- ČÁPKA, D. *MVC architektura*. 2014 [online]. [cit. 2016-10-29].
Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor>.
- ČÁPKA, D. *Úvod do Nette frameworku pro PHP*. 2014 [online]. [cit. 2016-10-29].
Dostupné z: <http://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette>.
- ČÁPKA, D. *Úvod do MySQL a příprava prostředí*. 2012 [online]. [cit. 2016-11-20].
Dostupné z: <http://www.itnetwork.cz/mysql/mysql-tutorial-uvod-a-priprava-prostredi>.
- FORAL, J. *Webová aplikace*. 2013 [online]. [cit. 2016-10-24].
Dostupné z: http://wiki.knihovna.cz/index.php/Webová_aplikace.
- HOGAN, B. *HTML5 a CSS3*.
Brno: Computer press, a.s., 2011. 272 s. ISBN 978-80-251-3576-1.
- KLČOVÁ, H., SODOMKA, P. *Informační systémy v podnikové praxi*.
Brno: Computer Press, a.s., 2016. 504 s. ISBN 978-80-251-2878-7.
- MICHÁLEK, M. *K čemu je dobrý Bootstrap a frontend frameworky?*. 2013 [online]. [cit. 2016-11-13].
Dostupné z: <https://www.zdrojak.cz/clanky/k-cemu-je-dobry-bootstrap-frontend-frameworky/>.
- NETTE *Framework Nette*. 2015 [online]. [cit. 2015-12-18].
Dostupné z: <https://nette.org/cs/>.
- PEHLIVANIAN, A., NGUYEN, D. *Javascript Okamžitě*.
Brno: Computer Press, 2014. 160 s. ISBN 978-80-251-4163-2.

- PROCHÁZKA, D. *CSS a XHTML tvorba dokonalých webových stránek krok za krokem*. 2. aktualizované vydání.
Praha: Grada Publishing, a.s., 2011. 175 s. ISBN 978-80-247-3897-0.
- RÁBOVÁ, I. *Podnikové informační systémy a technologie jejich vývoje*.
Brno: Mendelova univerzita, 2008. 139 s. I-04270/2011.
- SUEHRING, S. *Javascript krok za krokem*.
Brno: Computer Press, 2008. 336 s. ISBN 978-80-251-2241-9.
- ŠERÝ, R. *Javascript s jQuery - lehký úvod*. 2007 [online]. [cit. 2016-11-27].
Dostupné z: <https://www.interval.cz/clanky/javascript-s-jquery-lehky-uvod/>.
- ŠILHAVÝ, R., ŠILHAVÝ, P., PROKOPOVÁ, Z., SYSEL, M., POKORNÝ, P., MATÝSEK, M., VLČEK, K., SVOBODOVÁ, L. *Vybrané aspekty návrhu webových informačních systémů*.
Vsetín: Scientific Press, 2013. 178 s. ISBN 978-80-904741-3-0.
- VYMĚTAL, D. *Informační systémy v podnicích - teorie a praxe projektování*.
Praha: Grada Publishing a.s., 2009. 142 s. ISBN 978-80-247-3046-2.
- ZAJÍC, P. *PHP - Historie a budoucnost*. 2004 [online]. [cit. 2016-11-20].
Dostupné z: http://www.linuxsoft.cz/article.php?id_article = 171.

Přílohy

A LoginPresenter

```
1 <?php
2     namespace App\Presenters;
3
4     use Nette;
5     use Nette\Application\UI\Form;
6     use Nette\Database\Context;
7     use Nette\Application\UI\Multiplier;
8
9     use Nette\Forms;
10    use Nette\Forms\Controls;
11
12    class LoginPresenter extends Nette\Application\UI\
13        Presenter
14    {
15        private $database;
16
17        public function __construct(Nette\Database\Context
18            $database) {
19            $this->database = $database;
20        }
21
22        // podminka splnena v~pripade neuspesneho odeslani
23        // formulare
24        public function renderLogin($err) {
25            if ($err) {
26                $this->template->error = True;
27            }
28        }
29
30        // vytvoreni formulare
31        protected function createComponentLoginForm() {
32            $form = new Form;
33            $form->addText('name', 'žUivatel:');
34            $form->addPassword('password', 'Heslo:');
35            $form->addSubmit('login', 'řPihlásit se');
36            $form->onSuccess[] = [$this, '
37                loginFormSucceeded'];
38            return $form;
39        }
40
41        // vola se po uspesnem odeslani formulare
```

```
38     public function loginFormSucceeded($form, $values)
39     {
40         $user = $this->getUser();
41
42         try {
43             // test prihlaseni
44             $user->login($values->name, $values->
45                 password);
46             // v~pripade uspechu probehne presmerovani
47             // na domovskou stranku
48             $this->redirect("Homepage:default");
49         } catch (Nette\Security\AuthenticationException
50             $e) {
51             // $this->flashMessage('Chyba: ');
52             $this->redirect('Login:login', 1);
53         }
54     }
55 }
```


B Ukázka výstupu

PDF na samostatné stránce níže

TECHNICKÝ STAV VOZIDLA

Autobazar PERFEKT - CARS
Olbramkostel 177
671 51 Znojmo tel.515255467

Příloha č.1
Ke smlouvě č. 11111

Reg.zn: 2B6 4568 Datum uv. do provozu: 2016-11-02	Tovární značka: BMW X5 Druh vozidla: Osobní Výkon motoru: kW	VIN: 0 Motor:
Tachometr: 25 300 Druh paliva: Diesel	Spotřeba l/100km: 6.3	V nádrži: Olej: Garance:
Motor: Původní <input checked="" type="checkbox"/> Hlučný <input type="checkbox"/> Nižší výkon Stav: Původní <input type="checkbox"/> Únik oleje Rozv. řemen: Původní <input type="checkbox"/> Výměna		
Výfuk: Původní <input type="checkbox"/> Zvýš. hlučnost <input type="checkbox"/> Nutná výměna		Doporučujeme návštěvu u autorizovaného servisu
Spojka: Původní <input type="checkbox"/> Prokluzuje <input type="checkbox"/> Hlučné ložisko		
Převodovka: Manuální <input type="checkbox"/> Hlučná <input type="checkbox"/> Únik oleje		
Chlad.syst.: Původní <input type="checkbox"/> Nutná kontrola hustoty		
Rozvodovka: Původní <input type="checkbox"/> Hlučná <input type="checkbox"/> Únik oleje		
Akumulátor: Původní <input type="checkbox"/> Po opravě <input type="checkbox"/> Nefunkční		
Nápravy	Přední: Původní <input type="checkbox"/> Vůle uložení <input type="checkbox"/> Poškození Zadní: Původní <input type="checkbox"/> Vůle uložení <input type="checkbox"/> Poškození	
Brzdy: Původní <input type="checkbox"/> Nutná oprava <input type="checkbox"/> Sniž. účinnost Ruční brzda: Původní Obložení:		Kapalina:
Řízení: Původní <input type="checkbox"/> Zvýšená vůle <input type="checkbox"/> Funkční závada		
Karoserie: <input type="checkbox"/> Opravovaná <input type="checkbox"/> Poškozená Koroze: Původní <input type="checkbox"/> Koroze nosných částí		
Rám: Nezjištěno <input type="checkbox"/> Zjevná závada		
Lak, Barva: Nezjištěno <input type="checkbox"/> Oděrky <input type="checkbox"/> Puchýře		
Elektroinst.: Původní <input type="checkbox"/> Nutná oprava <input type="checkbox"/> Neuč. vnější osv.		
Příst. deska.: Nezjištěno <input type="checkbox"/> Poškozená		
Čalounění: <input type="checkbox"/> Zjevné povrchové závady		
Pneumatiky: Původní <input type="checkbox"/> Protektory		
Ráfky: Nezjištěno <input type="checkbox"/> Vizually poškozené		
Tlumiče: Původní Topení: Původní Klimatizace: Původní		
Nářadí: Nezjištěno		
Výbava: OK		
Další výbava:		
plní "EURO II"		
Příslušenství:		
Poznámka:		

Prohlášení majitele

Potvrzuji správnost shora uvedených údajů a prohlašuji, že jsem nezatajil(a) žádné vady, vlastnosti, poškození a zásahy do technického stavu, které by měly vliv na technický stav vozidla a tím i na jeho cenu.

V **Znojmo** dne **30.1.2017**

podpis prodávajícího

Prohlášení kupujícího

Potvrzuji převzetí vozidla a prohlašuji, že s jeho technickým stavem jsem byl seznámen včetně zkušební jízdy, že tyto skutečnosti beru na vědomí a že akceptuji stanovenou prodejní cenu.

V dne

podpis kupujícího