

Vývoj e-commerce systému napojeného na katalóg TecDoc

Diplomová práce

Vedúci práce:

Ing. Jiří Lýsek, Ph.D.

Bc. Jakub Vátrt

Brno 2017

Ďakujem vedúcemu práce Ing. Jiřímu Lýsekovi, Ph.D. za odborné vedenie a cenné rady pri vypracovaní tejto práce a konateľovi spoločnosti Hart Slovakia Mgr. Jánovi Judákovi za príkladné sprostredkovanie materiálov a spätnej väzby spoločnosti.

Čestné prehlásenie

Prehlasujem, že som prácu: **Vývoj e-commerce systému napojeného na katalog TecDoc** vypracoval/a samostatne a všetky použité zdroje a informácie uvádzam v zozname použitej literatúry. Súhlasím, aby moja práca bola zverejnená v súlade § 47b zákona č. 111/1998 Sb., o vysokých školách v znení neskorších predpisov a v súlade s platnou *Směrnici o zveřejňování vysokoškolských závěrečných prací*.

Som si vedomý/a, že sa na moju prácu vzťahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzatvorenie licenčnej zmluvy a použitie tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

Ďalej sa zaväzujem, že pred spísaním licenčnej zmluvy o použití diela inou osobou (subjektom) si vyžiadam písomné stanovisko univerzity, že predmetná licenčná zmluva nie je v rozpore s oprávnenými záujmami univerzity a zaväzujem sa uhradiť prípadný príspevok na úhradu nákladov spojených so vznikom diela, a to až do ich skutočnej výšky.

V Brne dňa 18. mája 2017

Abstract

Vatrt, J., Development of e-commerce system connected with TecDoc catalogue. Brno: Mendel University in Brno, 2017.

This thesis deals with design and implementation of e-commerce solution for automotive segment connected with TecDoc catalogue. According results of analytics of the current solution, new one solution has been projected and implemented.

An abstract is in (British) English.

Keywords

E-commerce, TecDoc, PHP, HTML, CSS, XML, Codeigniter, orders, catalogues, cluster

Abstrakt

Vatrt, J., Vývoj e-commerce systému napojeného na katalóg TecDoc. Diplomová práca. Brno: Mendelova univerzita v Brně, 2017.

Práca obsahuje návrh a implementáciu e-commerce riešenia pre automotive segment napojeného na katalóg TecDoc. Podľa výsledkov analýzy súčasného riešenia bolo navrhnuté a nové riešenie.

Kľúčové slová

E-commerce, TecDoc, PHP, HTML, CSS, XML, Codeigniter, objednávky, katalógy, cluster

Obsah

1	Úvod	11
2	Cieľ práce	12
2.1	Metodika práce	12
3	Súčasný stav	14
3.1	Rešerš trhu a možných riešení	14
4	Návrh riešenia systému	16
4.1	Požiadavky na systém	16
4.1.1	Formálne požiadavky na systém	16
4.1.2	Technické požiadavky na systém	20
4.2	Zloženie e-commerce systému	21
4.3	Prehľad navrhovaných technológií	22
4.3.1	Webserver	23
4.3.2	Skriptovací jazyk PHP	24
4.3.3	Framework Codeigniter	24
4.3.4	MariaDB	25
4.3.5	HTML5 Boilerplate + Twitter Bootstrap	25
4.3.6	jQuery a React.js	26
4.3.7	XML prenos dát	26
4.4	Návrh štruktúry databáz	26
4.4.1	Centrálne databáza	28
4.4.2	Databáza TecDoc	31
4.5	Grafický návrh frontendu	32
4.5.1	Úvodná stránka	34
4.5.2	Katalóg TecDoc	35
4.5.3	Nákupný košík	36
4.6	Návrh administračnej časti	37
5	Implementácia systému	39

5.1	Implementácia centrálnej časti	39
5.1.1	Nastavenie frameworku.....	39
5.1.2	Inštalácia potrebných zásuvných modulov	40
5.1.3	Príprava administračnej časti.....	41
5.1.4	Programovanie frontendu – hlavička a pätička	41
5.1.5	Úvodná stránka.....	42
5.1.6	Nákupný košík.....	43
5.1.7	Objednávka tovaru	43
5.1.8	Profil používateľa.....	44
5.1.9	Prihlásenie a zabudnuté heslo.....	45
5.1.10	Administračná časť	45
5.2	Implementácia katalógu TecDoc.....	49
5.2.1	Importovanie položiek databázy.....	49
5.2.2	Kódovanie frontendu.....	53
5.3	Nasadenie na server.....	53
5.4	Prepojenie so skladovým systémom.....	55
5.5	Testovanie systému.....	56
5.5.1	Titulná stránka a jadro stránky	56
5.5.2	Prihlásenie, zabudnuté heslo a profil používateľa	56
5.5.3	Katalóg TecDoc.....	56
5.5.4	Nákupný košík a dokončenie objednávky.....	56
5.5.5	Prenos dát do skladového systému.....	57
5.5.6	Zhodnotenie	57
6	Diskusia	58
7	Záver	60
8	Literatúra	61
9	Zoznam obrázkov	63
10	Zoznam tabuliek	65
A	Priložené CD so zdrojovými kódmi a ukážkovou databázou	67

1 Úvod

Internetové obchodovanie sa stalo v poslednej dobe jednou z najčastejších služieb využívaných prostredníctvom internetu. Spoločnosti plynule prešli od objednávok prostredníctvom faxu alebo emailov k pohodlnejšej variante výberu tovaru prostredníctvom katalógov a internetových obchodov. Väčšina dnešných spoločností má vyvinuté komplexné informačné systémy, ktoré zvyšujú ich produktivitu a automatizujú čo najväčšie množstvo procesov aby sa zabránilo chybám ľudského faktoru. Tieto systémy sú často súčasťou firemného ERP riešenia, ktoré zabezpečí aj jednotlivé post objednávkové procesy.

Bol som oslovený firmou Hart Slovakia, aby som im navrhol a zrealizoval nový internetový obchod pre segment B2B a neskôr aj informačný systém pre celú spoločnosť. Spoločnosť sa zaoberá veľkoobchodnou distribúciou náhradných dielov pre automobily a zariadením autodiagnózy. Základ tvorí katalóg TecDoc, ktorý je dodávaný externou firmou z Nemecka a obsahuje takmer všetky diely, ktoré spoločnosť ponúka svojim partnerom. Spoločnosť používa okrem katalógu TecDoc aj iné katalógy, ktoré potrebuje v budúcnosti do navrhnutého riešenia implementovať. Dôraz sa kladie najmä na rýchlosť a jednoduchosť objednávkového procesu, keďže z týchto nedostatkov súčasného riešenia vznikla potreba realizácie nového systému.

Verím že v kooperácii s konateľom spoločnosti a zamestnancami pripravím kvalitný systém na mieru spoločnosti a splňujúci všetky ich požiadavky. Zároveň som sa snažil celé riešenie navrhnuť nadčasovo tak, aby prípadné inovácie niektorých častí sa dali nasadiť bez nutnosti znovu vyvíjať celý systém odznovu.

2 Cieľ práce

Cieľom diplomovej práce je zanalyzovať súčasne riešenie firmy, navrhnúť a implementovať nové riešenie, pričom dôraz sa kladie na rýchlosť a jednoduchosť objednávok a prepojenie so súčasným skladovým systémom. Súčasťou analýzy a rešerše bude aj porovnanie s už existujúcimi riešeniami e-commerce systémov na trhu, inšpirovať sa ich výhodami a niektoré z nich nasadiť aj v novo pripravovanom systéme spoločnosti.

Po vypracovaní analýzy a rešerši podobných systémov na trhu sa vytvorí celkový návrh systému, ktorý musí hlavne byť rýchly a modulovateľný, keďže sa plánuje zapojenie ďalších obdobne veľkých katalógov do systému. Samozrejmosťou je použitie čo najnovších dostupných technológií tak, aby novo navrhnutý systém bol schopný fungovať dlhšiu dobu bez nutnosti programátorského zásahu alebo prerábania niektorej jeho časti. Jednotlivé katalógy musia byť nezávislé na sebe, bude ich možné kedykoľvek aktualizovať alebo vypnúť.

Po návrhu systému sa prejde k implementácii riešenia na vybranom programovacom jazyku a databázovom systéme. Po dokončení implementácie sa systém interne otestuje a napojí na súčasný skladový systém. Spoločnosť na základe týchto testov rozhodne o nasadení do produkčného prostredia.

Záveru nasadenia budú spracované v diskusii, kde budú zhrnuté aj možné vylepšenia a námety do budúcnosti.

2.1 Metodika práce

Na základe rešerše obdobných systémov a vytýčení hlavných vlastností, bude vypracovaný návrh nového riešenia, pričom jednotlivé procesy budú demonštrované pomocou UML diagramov tak, aby sa opísala celá formálna funkcionálna. Na modelovanie diagramov využijem nástroj Enterprise architect. Funkčné celky ako databáza budú načrtnuté ako EER diagramy na lepšiu demonštráciu prepojení jednotlivých tabuliek pomocou cudzích kľúčov, ktoré budú namodelované prostredníctvom nástroja MySQL Workbench. Export návrhu štruktúry databázy sa importuje prostredníctvom nástroja phpMyAdmin alebo Adminer na databázový server.

Vyhotovenie aplikačnej časti bude založené na frameworku Codeigniter, ktorý bude doplnený rozšíreniami tretích strán podľa potreby, napríklad pre komunikáciu so skladovým systémom. Do frameworku bude možné vkladať rozšírenia tretích strán podľa potreby, pre zabezpečenie požadovanej funkčnosti.

Na spracovanie dodaného katalógu do databázy sa naprogramujú vlastné skripty na import, ktoré pobežia v cez rozhranie PHP v príkazovom riadku. Spoločnosť TecDoc dodáva katalóg na štvrťročnej báze, čiže treba mať pripravený univerzálny skript spracovania dát do navrhovanej štruktúry databázy schopný zvládať nové verzie s drobnými obmenami.

Grafický návrh frontendu bude spracovaný podľa dodaných materiálov od grafika spoločnosti, ktorý bude zohľadňovať požiadavky na funkcionality, či už technickú alebo formálnu.

Nasadenie systému bude podľa výsledku a náročnosti nasadené na vlastnom virtualizovanom serveri VPS, pričom podľa potreby bude zakúpených viacero VPS serverov na oddelenie databázy a ostatných častí systému. Aplikácia sa bude priebežne interne testovať a podľa výsledkov testov sa rozhodne o jej nasadení v produkcii. Súčasťou testov bude aj prepojenie so súčasným skladovým systémom od spoločnosti Nextis.

3 Súčasný stav

Momentálne spoločnosť využíva kompletný informačný systém od spoločnosti Nextis. Tento systém v súčasnej verzii prestáva spĺňať požiadavky spoločnosti Hart hlavne čo sa týka rýchlosti a komfortu nákupu. Celý informačný systém je založený na technológiách spoločnosti Microsoft, hlavne .NET a ASP.

Systém beží na single-server riešení, ktoré je navyše zdieľané aj s inými používateľmi a nie je schopné replikácie a zapojenia do clusteru v prípade požiadavky na vyšší výkon. To negatívne ovplyvňuje celú používateľskú skúsenosť a súčasne narastá nespokojnosť používateľov aj manažérov systému. Zistiť, či by nebolo možné súčasný systém replikovať na viacero serverov sa mi nepodarilo, nakoľko po pripojení cez RDP klienta sa som na súčasnom serveri obmedzený právami používateľa, ale podľa použitej verzie Windows Server 2008 by to možné bolo, keďže spoločnosť Microsoft funkcionality clusterov podporuje už od verzie 2000 (Microsoft, 2017).

Súčasný e-commerce systém má zastaraný dizajn, ktorý nespĺňa napríklad požiadavky responzivity pre mobilné zariadenia, optimalizácie kódu pre vyhľadávače a URL adres, často sa objavujú výpadky kvôli novej nekompatibilitate s katalógom TecDoc. Obchodných zástupcov trápi absencia histórie a možnosti krokov vzad, čo sa prejavuje hlavne pri náhodnom znovu načítaní stránky a reštarte objednávkového procesu na začiatok, čo spôsobuje nejednoznačnú identifikáciu prostredníctvom vyššie spomínanej URL adresy.

Vytvorené objednávky následne sa synchronizujú do skladového systému kde sú spracované a dodané zákazníkovi. Súčasná firma Nextis, ktorá dodáva spoločnosti aj skladový systém, má riešenie komunikáciu s externými riešeniami pomocou výmeny XML súborov s dátami. Celé súčasné e-commerce riešenie je riadené prostredníctvom skladového programu, kde sú aj viaceré možnosti ako napríklad nastavovanie priorít jednotlivým výrobcam, zlučovanie kategórií, nastavovanie noviniek a iné. Nakoľko oba systémy sú na rovnakej platforme .NET, je zabezpečená bezproblémová komunikácia medzi nimi. Na skladový software sa pripája takisto pomocou RDP klienta a dáta sú na strane dodávateľa, čo vzbudzuje obavy vedenia spoločnosti z možného vyzradenia obchodného tajomstva a cenotvorby konkurencii.

3.1 Rešerš trhu a možných riešení

Na začiatok rešerše treba poznamenať, že požadovaný spôsob napojenia katalógu TecDoc, teda implementáciou na strane klienta nie on-line vyhľadávaním priamo v spoločnosti TecDoc, ponúkala v dobe písania tejto práce z tuzemských spoločností iba firma Nextis, ktorá zároveň dodáva aj skladový systém k svojmu eshopu.

Okrem spoločnosti Nextis, ponúka spracovanie katalógu TecDoc aj spoločnosť ShopSys, ktorá ponúka prenájom systému za mesačné platby, ktoré sú odvíjané od obratu spoločnosti. Toto riešenie bolo konečne zamietnuté, nakoľko množstvo

objednávok stále rastie a tento model financovania by nebol rentabilný pre spoločnosť. Takisto aj spoločnosť ShopSys aj iná spoločnosť inShop dodáva implementáciu iba vo WEB SHOP verzii, teda katalóg je na serveroch v Nemecku a nie na vlastných serveroch. Dáta obchodu sú takisto na serveroch dodávajúcich spoločností a nie na strane klienta, čo znovu spôsobuje vyššie spomínané obavy z nechceného vyzradenia obchodného tajomstva.

Okrem toho zvyšné firmy v zahraničí ako napríklad v Poľsku alebo Nemecku si systém vytvárajú na mieru ich potrebám a možnému prenájmu alebo odkúpení ich riešenia neboli naklonení, nakoľko tiež chceli systémy ponúkať ako Software as a Service solution (AWARDS-BA³ SaaS, 2017). Niektoré z nich mali aj problém s rýchlosťou platformy pri vyššom počte produktov, čo nerieši hlavný problém spoločnosti. Taktiež niektoré riešenia neumožňovali preklad rozhrania, čo je zásadný problém nasadenia v slovenskom prostredí.

Keď spoločnosť uvažovala nad nasadením nejakého voľne dostupného OpenSource riešenia (Woocommerce, Magneto, ZenCart), bohužiaľ ani jedno riešenie nebolo natoľko robustné, aby bolo schopné obslúžiť katalóg TecDoc, z ktorého spoločnosť momentálne využíva orientačne okolo 2 000 000 produktov. Zaujímavá prednáška na túto tému bola od Davida Hilditcha, ktorý sa snažil optimalizovať systém WooCommerce pre 1 milión produktov, pričom narazil na limit 5000 produktov a nemožnosť cachovania a škálovania celého systému (Hilditch, 2016). Keďže WooCommerce je momentálne jeden z najpoužívanejších e-commerce riešení na internete (Henderson, 2016) a súčasťou platformy Wordpress, obdobný problém majú aj ostatné riešenia s výnimkou Magneta, kde je potrebná Enterprise licencia, ale aj tak sa objavujú problémy s indexáciou a vyhľadávaním (Winkle, 2013), čo nevyhovuje z pohľadu rýchlosti.

Nakoľko jedna z požiadaviek spoločnosti je rýchlosť celého systému, bolo treba navrhnuť a vyvinuť vlastný systém na mieru a následne ho prepojiť s existujúcim skladovým systémom.

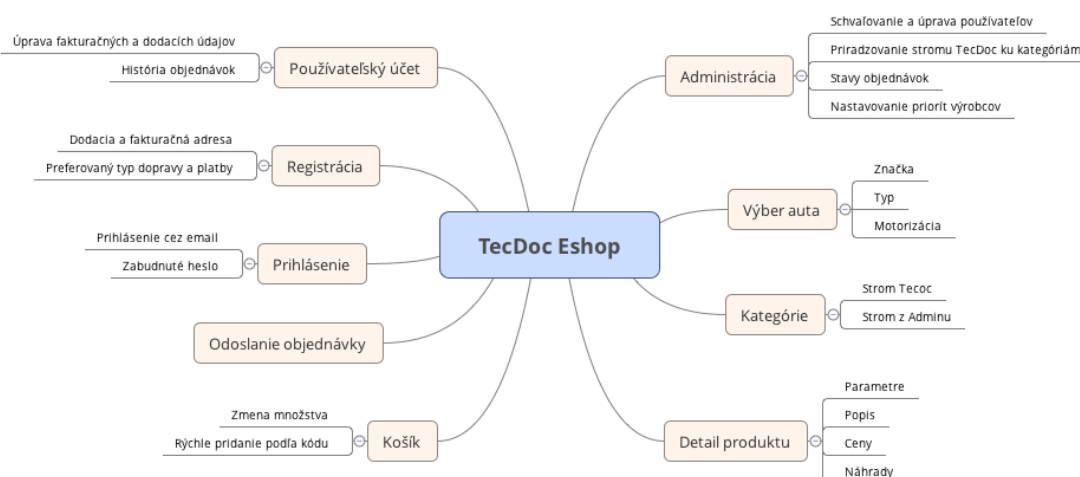
4 Návrh riešenia systému

4.1 Požiadavky na systém

Pri tvorbe technických a formálnych požiadaviek na nový systém som úzko spolupracoval s vedením spoločnosti a dodávateľmi katalógu TecDoc. Vedenie spoločnosti chcelo hlavne zachovať súčasnú funkcionality a zlepšiť jej výkon a pridať niektoré chýbajúce časti. Taktiež by malá byť zlepšená spoľahlivosť a bezpečnosť systému. Po zapracovaní ich a mojich pripomienok a zlepšení som vytvoril viacero myšlienkových máp a use case diagramov, pre lepšiu demonštráciu funkčnosti. Jednotlivé schémy sa môžu počas vypracovania systému jemne meniť, ale základ dohodnutej a schválenej schémy platí pre celý projekt.

4.1.1 Formálne požiadavky na systém

Na obrázku 1. je rozkreslená myšlienková mapa základnej funkcionality e-commerce systému. Každá časť zapadá do určitého stavu procesu objednávky, či už je to výber značky, jednoduché vyhľadávanie v kategóriách jednotlivých dielov.



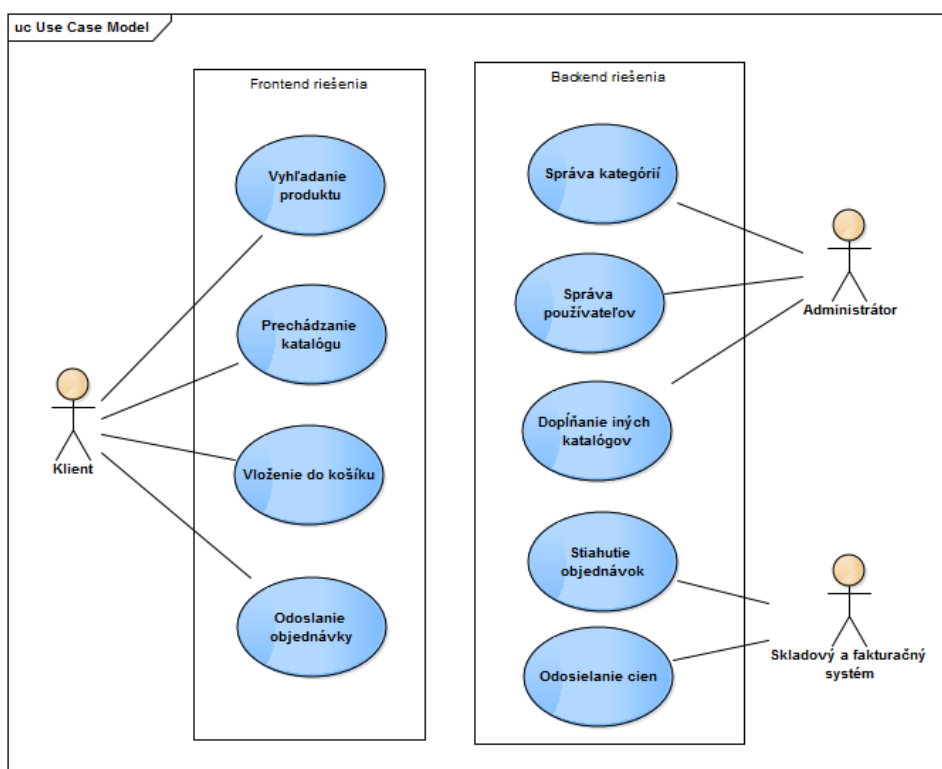
Obr. 1 Myšlienková mapa funkcií návrhu nového e-commerce riešenia

Úvodnú stránku bude tvoriť akciová ponuka spoločnosti, ktorá bude dostupná aj neprihláseným používateľom. Samotný obchod a riešenie bude dostupné až po prihlásení klienta, respektíve jeho zaregistrovaním a schváleným administrátorom. Po prihlásení bude mať klient na výber z ponúkaných značiek automobilov, z ktorých si klient vyberie a následne sa mu otvorí zoznam typov aut danej značky. Po výbere typu bude zobrazený strom kategórií dielov, ktoré si vie spoločnosť upravovať a zlučovať pomocou webovej administrácie.

Každá kategória obsahuje produkty, ktorých poradie a priority značiek výrobcov si spoločnosť obdobne vyberie prostredníctvom administrácie. Každý produkt po roztvorení v kategórii má svoj detail, kde sú vidieť hlavne montážne parametre dielu, ktoré sú pre zákazníkov mimoriadne dôležité. Nie je nutné vytvárať detail produktu z dôvodu urýchlenia objednávkového procesu. Oblíbená funkcia pri terajšom internetovom obchode je taktiež zobrazenie kompatibilných náhrad so súčasným dielom, ktoré by chceli zachovať aj v novom e-commerce systéme.

Po vložení tovaru do košíku sa prejde k pokladni kde sa vyberie spôsob dodania, schválenie obchodných podmienok a celá objednávka sa odošle na spracovanie. Tieto údaje bývajú spravidla pred vyplnené už od doby registrácie zákazníka, možnosť zmeny bude mať len administrátor cez skladový program. Zaregistrovaný klient je schopný tieto údaje vo svojom profile meniť aby sa mu pri ďalšej objednávke samostatne vyplnili a taktiež aj zobrazovať jeho historické objednávky.

Objednávky sa potom budú ďalej spracovávať v skladovom a fakturačnom systéme, s ktorými bude obchod prepojený pomocou predom dohodnutého rozhrania, spravidla XML formátu dokumentu.

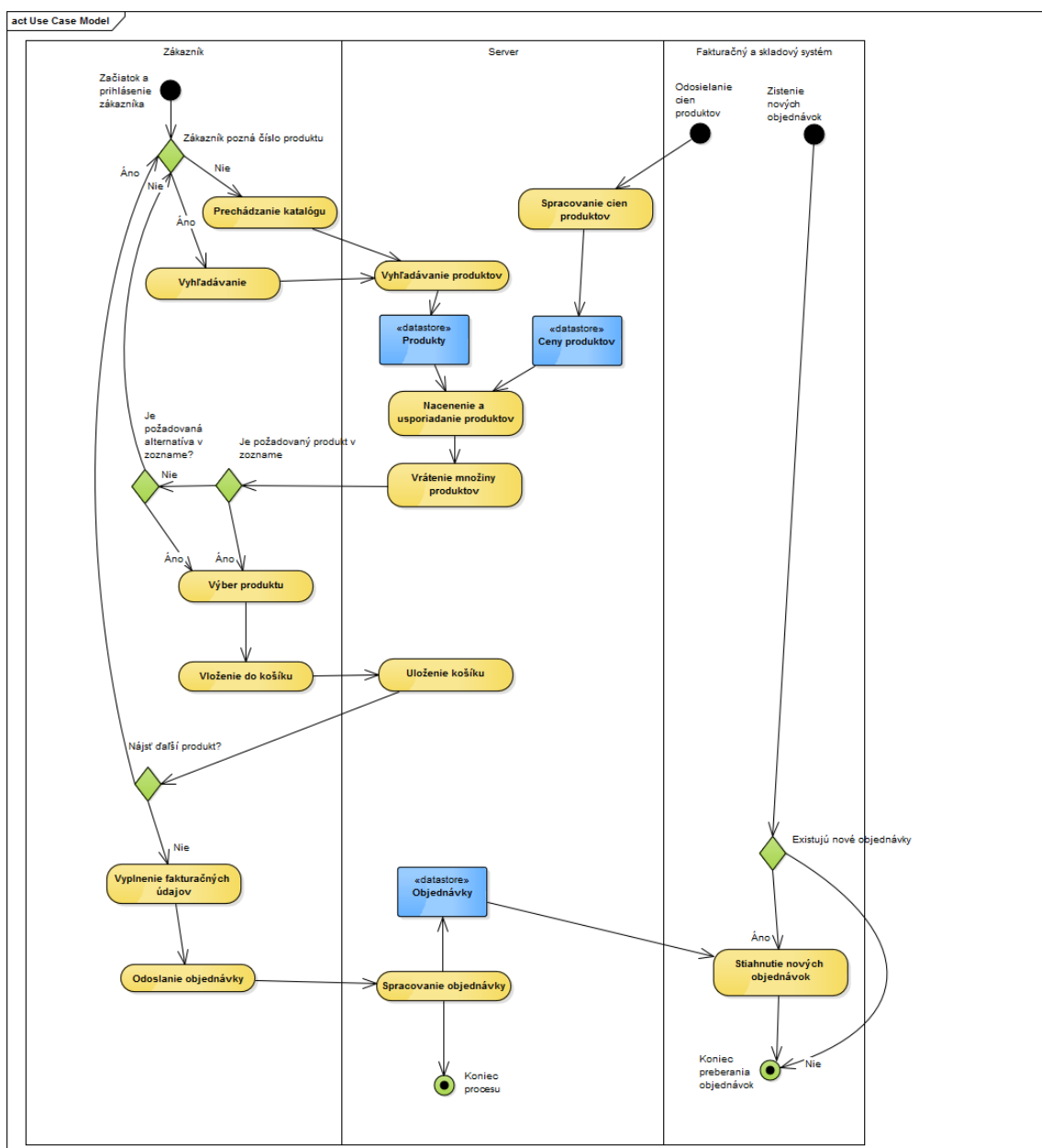


Obr. 2 Use Case diagram funkcionality systému

Na obrázku 2 je vidieť celú funkcionality a vzťahy v Use case diagrame. Celý diagram je rozdelený na frontendovú časť dostupnú klientovi a pozadia spracovania úkonov – Backend.

Za predpokladu že aktor klient je prihlásený, má možnosť si prezerat' katalóg respektíve katalógy spoločnosti, vkladať jednotlivé produkty do košíku a tvorbu objednávok. Aktor administrátor, čo bude poverený pracovník, bude mať možnosť spravovať kategórie, používateľov a taktiež dopĺňať položky do iných katalógov.

Spracované objednávky si následne vyžiada aktor Skladový a fakturačný systém ktorý ich spracuje. Zároveň tento aktor bude odosielať aktuálne ceny produktov.



Obr. 3 Schéma nákupného procesu používateľa

Na schéme nákupného procesu je vidieť jednotlivé zapojenie všetkých komponentov systému a jednotlivé úkony jednotlivých aktorov.

Zákazník po prihlásení pokračuje zadaním kódu produktu do vyhľadávania, alebo si manuálne produkt vyhladá prechádzaním stromu katalógu, teda výberom značky auta, typu auta a motorizácie. Serverová časť prevezme požiadavku a vyhladá v datastore (v našom prípade databáza TecDoc), vhodné produkty podľa zadaných parametrov a priradí ceny vráteným produktom z jej vlastného datastoru, následne zoradí podľa priorit značiek a celú túto množinu produktov odošle zákazníkovi, ktorému sa zobrazia ako výpis.

Ak sa produkt nachádza vo výpise, zákazník si vyberie požadovaný produkt a množstvo a vloží si ho do košíku. V opačnom prípade si zákazník má možnosť vybrať z množiny alternatív (v systéme označované aj ako náhrady). Pokiaľ si nevybral ani náhradu prechádza späť k vyhľadávaniu produktu. Výber sa môže viacnásobne zopakovať, pokiaľ košík zákazníka neobsahuje želané produkty.

Nakoniec zákazník prejde k výberu spôsobu dodania a schváleniu objednávky. Objednávku server uloží do svojho datastore objednávok, kde bude čakať na vyzdvihnutie mechanizmom skladového a fakturačného systému.

Skladový a fakturačný systém je prakticky nezávislý od objednávkového procesu, ale podieľa sa na poslednej časti a to vyzdvihnutí objednávok z datastoru do fakturačného software.



Obr. 4 Funkcionalita administrácie

Poslednou časťou obchodu je administrácia, kde povolani pracovníci sú schopní meniť strom kategórií dielov a zlučovať jeho časti. Taktiež môžu schvaľovať jednotlivé používateľské registrácie, nakoľko sa jedná o B2B segment, tak pri registrácii je nutné uviesť firemnú a daňovú identifikáciu IČO a DIČ. Po overení týchto údajov je admin schopný klienta schváliť a pripustiť k možnosti objednávania tovaru. Z terajšieho riešenia je požadovaná už spomínaná funkcionlita určovania priorít zobrazovania jednotlivých výrobcov. Administrátor bude mať možnosť pridať vlastné produkty mimo katalógu TecDoc, ktoré budú podobne ako ostatné katalógy do budúca oddelené. Posledná časť Akcie a statické stránky bude mať na starosti zobrazovania akciových letákov na úvode a možnú editáciu statických stránok ako sú Kontakt, Obchodné podmienky a iné.

4.1.2 Technické požiadavky na systém

Nový systém elektronického obchodu by mal spĺňať viacero požiadaviek. Hlavná požiadavka je rýchlosť celého systému, jeho stabilita a jednoduchá rozšíriteľnosť. Systém musí zvládať plynulo narastajúce množstvo objednávok a to hlavne v špičkách cez deň. Väčšina odberateľov spoločnosti má spustený internetový

obchod na pozadí praktický celý deň, čiže do úvahy pripadá aj jednostránková webová aplikácia prípadne desktopová aplikácia.

Taktiež by sa mala zlepšiť spoľahlivosť systému. Podľa Sommerville, spoľahlivý software by v prípade zlyhania systému nemal spôsobiť fyzickú alebo eknomickú škodu (Sommerville, 2013), čo v našom prípade znamená poškodenie databázy s objednávkami.

Čo sa týka dizajnu, mal by mať všetky prvky responzívne, aby bolo jednoduché spracovávať objednávky z tabletu, čo v teréne využijú obchodný zástupcovia. Dizajn sa predpokladá minimalistický s čo najmenšou záťažou na server a zariadenie používateľa. Predpokladá sa aj určitá spätná kompatibilita, avšak táto požiadavka nesmie brániť plynulosti a funkčnosti systému.

Systém musí byť prepojený so súčasným skladovým a fakturačným systémom prostredníctvom výmeny správ vo formáte XML, spravidla objednávok, stavu skladov, cien produktov a zákazníckych kariet. Špecifikácia a formát bude dodaný dodávateľom fakturačného systému.

Samozrejma je možnosť viacerých administrátorov systému, nakoľko spoločnosť ma viacero pobočiek a špecifických produktov mimo katalóg TecDoc, ktoré je nutné do systému zaradiť. Taktiež sa touto funkcionalitou chce vyhnúť možným výpadkom v prípade dovolenky alebo choroby správcu jednotlivej skupiny produktov.

Do budúca sa počíta s rozšírením celého systému a to nie len počtom produktov, ale aj zakomponovaním konkurenčných katalógov ako je napríklad ACI. Samozrejme s narastajúcim počtom položiek musí byť zachovaná rýchlosť a stabilita celého systému. Preto je nutné jednotlivé databázy a katalógy oddeliť tak aby vedeli fungovať nezávisle. Celý systém bude nutné oddeliť na jednotlivé katalógy, pričom vyhľadávanie bude fungovať len pre daný otvorený katalóg. Jediná časť, ktorú budú mať systémy spoločné je nákupný košík, do ktorého budú môcť spadať produkty z viacerých katalógov.

Neskôr je v pláne aj možnosť vytvoriť natívnu aplikáciu pre mobilné platformy Android a iOS, aby zákazníci mohli pohodlnejšie a rýchlejšie zadávať svoje objednávky. Preto pri návrhu a vypracovaní je treba myslieť aj na túto dodatočnú funkcionalitu aby následné napojenie naprogramovaných aplikácií bolo možné.

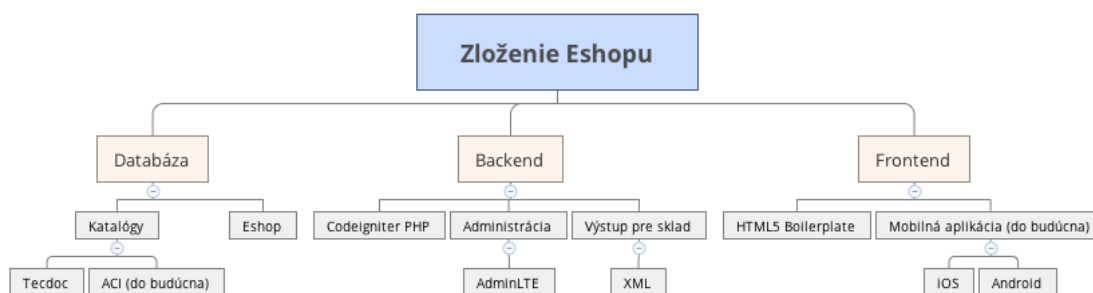
4.2 Zloženie e-commerce systému

Z početných konzultácií som dospel k záveru že jednotlivé zložky systému ako databáza, frontend a backend bude nutné oddeliť tak, aby boli schopné nezávisle od seba schopné replikácie v cloude. Preto treba navrhnúť oddelenú štruktúru databáz pre katalógy a jednu centrálnu databázu, ktorá bude uchovávať spoločné dáta ako prihlasovacie údaje či objednávky. Tieto databázy sa budú môcť viacnásobne replikovať, aby vo výkonnostných špičkách bola dodržaná rýchlosť systému. Replikáciu bude zabezpečovať zvolený cloudový systém pomocou jeho

mechanizmov, zvažuje sa použitie riešení Forpsi Cloud alebo Micorosoft Azure či Amazon AVS.

Samotnú aplikácia bude navrhnutá ako MVC architektúra, čiže bude mať oddelené časti na model – databáze, view – zobrazovanie dát a kontroler – centrálna časť, ktorá spája model a kontroler do funkčného celku. Kontroler, respektíve vybraný aplikačný framework, musí mať možnosť práce s viacerými databázami a samozrejme musí byť rozšíriteľný o funkcionality pomocou zásuvných modulov tretích strán. View časť by mala byť podporovať responzívny dizajn a byť čo najúspornejšia, aby sa skrátila doba načítavania stránky.

Po spojení databáz a aplikačnej časti, bude nasledovať testovacia fáza, jednak pohodlnosti použiteľnosti e-commerce riešenia, ale aj sprístupneniu potrebnej funkcionality pre administráciu stránky. Časti systému budú obsluhované pomocou skladového systému, ktorého prepojenie bude pripravené podľa spoločnosti dodávajúcej tento systém.



Obr. 5 Rozdelenie platformy na nezávislé funkčné celky, ktoré je možné replikovať

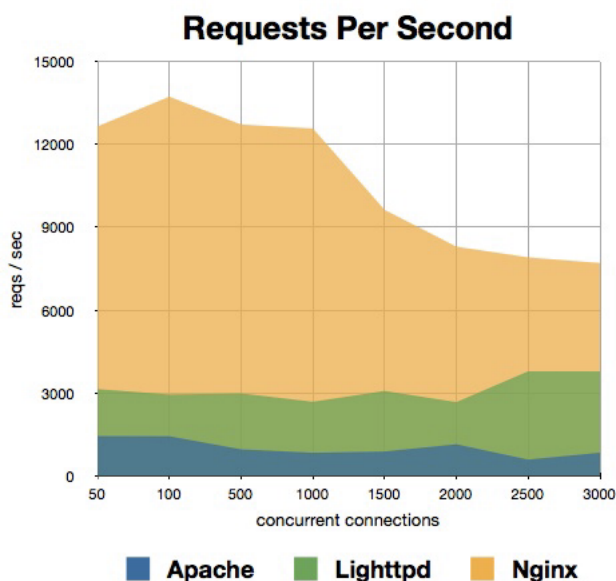
Na obrázku je rozdelenie hlavných zložiek systému, jednotlivé databázy sú pre ilustráciu rozdelené na katalógy a centrálnu databázu (eshop). Na túto databázu sa napája backend, čo je v našom prípade kontroler, ktorý slúži pre obsluhovanie mechanizmov frontendu – stránky a administrácie a synchronizácie so skladom. Frontend je zabezpečený pomocou zobrazovania HTML a CSS súborov – ideálne cez nejaký framework a do budúca sa počíta aj s nasadením mobilných aplikácií. Frontend treba zvoliť čo najviac minimalistický a čiastočne spätne kompatibilný so staršími prehliadačmi.

4.3 Prehľad navrhovaných technológií

Pre komplexný systém, akým navrhované riešenie bude, treba vybrať tie najlepšie technológie zabezpečujúce rýchlosť a bezpečnosť riešenia. Po konzultáciách a testoch som návrh orientoval hlavne na zachovanie čo najvyššej rýchlosti hlavne v špičkách – prevádzky počas dňa. Všetky použité technológie majú otvorený zdrojový kód a neustály vývoj, takže je dosiahnutá aj nezávislosť na dodávateľovi, čo bol problém terajšieho riešenia.

4.3.1 Webservice

Webservice je centrálna entita, spracováva požiadavky používateľa prostredníctvom URL adresy a serverových premenných. Takisto zabezpečuje beh serverových skriptov ako PHP alebo Ruby. Nakoľko serverové skripty budú programované pre jazyk PHP, tak pre návrh sú uvažované dva najpopulárnejšie webservery a to Apache alebo Nginx. Spolu distribuujú viac ako 50% obsahu na internete. Jazyk PHP je plne kompatibilný s oboma webovými servermi, takisto aj dostupnosť databázových modulov pre napojenie databázy. V našom prípade je výhodnejšie použiť serveru Nginx a to z viacerých dôvodov. Nginx v porovnaní so serverom Apache je výrazne rýchlejší v počte požiadavkou za sekundu a to hlavne pri strednom množstve pripojení, v našom prípade okolo 1000 – 1500. Rýchlosť Nginxu spočíva v použití asynchrónnych soketov, teda Nginx nevytvára nové procesy keď dostane požiadavku. Jeden proces na jedno procesorové jadro vie tak zvládnuť tisíce pripojení, čo vedie k nižšej záťaži procesoru a spotreby pamäte. (Nedelcu, 2015) Porovnanie so serverom Apache je zhrnuté na nasledujúcom obrázku:

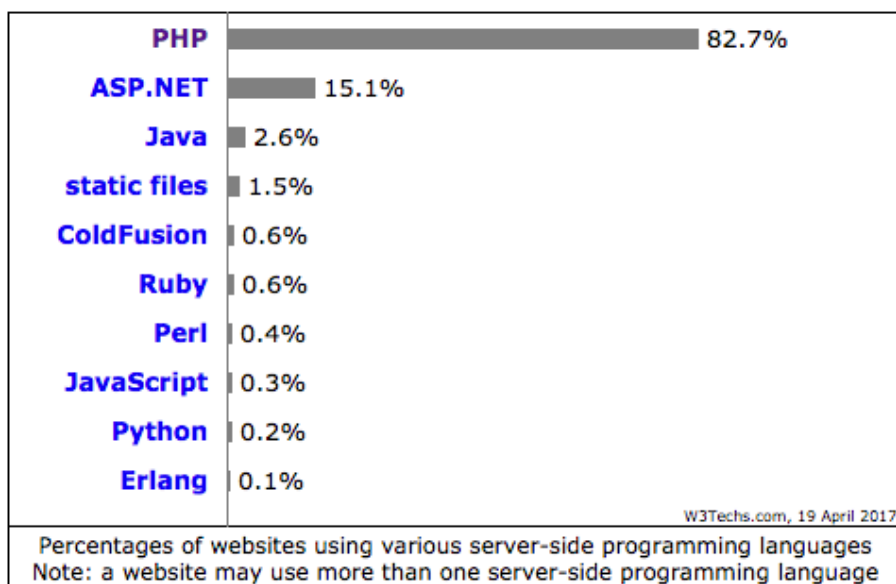


Obr. 6 Počet požiadavkov za minútu, ktoré sú schopné spracovať jednotlivé webservery
Zdroj: Dreamhost, 2017

Nginx server sa dá použiť najmä na serveri, kde bude bežať databáza TecDoc a jednotlivé obrázky produktov budú uložené ako fyzické súbory na disku na ktoré sa bude možno odkazovať priamo prostredníctvom URL adresy, čo je z výkonnostného hľadiska výhodnejšie ako ukladať obrázky do databázy a dolovať ich cez PHP.

4.3.2 Skriptovací jazyk PHP

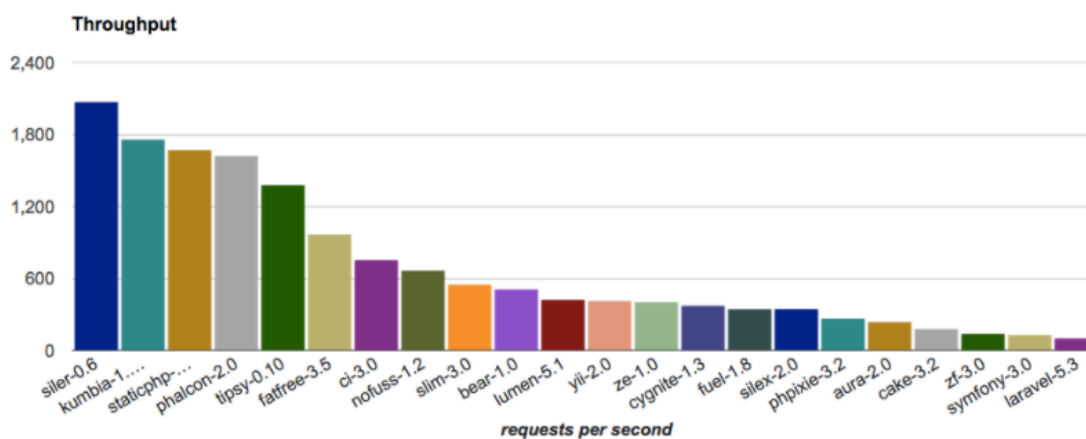
Skriptovací jazyk PHP je dnes najrozšírenejší jazyk pre vývoj webových aplikácií. Prostredníctvom neho vieme spracovávať dynamicky webové stránky, meniť ich obsah. V dobe písania tejto práce bola najnovšia verzia jazyka 7.1, ktorá hlavne s príchodom verzie 7 priniesla nárast výkonu v niektorých prípadoch až o 30% oproti verzii 5.6. Zároveň pravidelne zavádza moderné podporné funkcie ako napríklad pre spracovanie XML súborov či šifrovania dát. Pre samotný jazyk PHP je dostupná celá rada rozšírení, pre ktoré vznikol centrálny repozitár Composer, pomocou ktorého vieme do jazyku pridávať želané funkcie a zároveň obmedzovať kompatibilné verzie. Takisto vzniklo pre jazyk viacero frameworkov, aby sa urýchlil vývoj písania rutinných funkcií a programátori sa mohli zamerať na celkovú funkčnosť riešenia.



Obr. 7 Použitie jednotlivých serverových skriptovacích jazykov v praxi
Zdroj: W3Techs, 2017

4.3.3 Framework Codeigniter

Framework Codeigniter je PHP framework vyvíjaný v dnešnej dobe na British Institute of Technology v Montreali. CodeIgniter je inteligentný framework pre vývoj aplikácií s flexibilným a rozšíriteľným jadrom, ktoré má vysoký výkon a záťažou na operačnú pamäť. (Orr, 2013) Jedná sa o dlhodobo vyvíjaný framework so širokou používateľskou základňou a možnosťami použitia. Hlavnou z jeho predností je rýchlosť frameworku, kedy v niektorých prípadoch dokonca predčí necachovaná verzia cachované verzie aplikácií v konkurenčných frameworkoch.



Obr. 8 Porovnanie počtu požiadavkou, ktoré sú jednotlivé frameworky schopné spracovať, za sekundu
Zdroj: (Kenjis, 2017)

Ďalšou z jeho predností je jednoduché učenie a pochopenie frameworku, kde ponúka používateľovi voľnosť a nie je tak striktný ako ostatné frameworky. V tejto kombinácii je framework vhodný pre agilný vývoj aplikácie a jej zmeny podľa potreby. Codeigniter prináša integráciu viacerých typov databáz, čo je výhodné ak bude potrebné premigrovať databázu na robustnejšie riešenie. Taktiež sa framework hodí pre vývoj REST API systému, ktorý bude neskôr použitý pre napojenie mobilných aplikácií.

4.3.4 MariaDB

Databázový systém MariaDB je odnož databázového systému MySQL, ktorá vznikla po prevzatí spoločnosti Sun, zastrešujúcou vývoj MySQL, konkurenčnou spoločnosťou Oracle, ktorá vyvíja rovnomenné databázové riešenie. MariaDB je v porovnaní s MySQL výkonnejšia hlavne vo fulltextovom vyhľadávaní, čo bude kľúčové pre náš systém. Podľa Fínskej konzultačnej spoločnosti Seravo, prechod z MySQL na MariaDB priniesol rádovo zlepšenie výkonu v rozmedzí 3-5% (Seravo, 2015). Takisto je plne podporované zvoleným frameworkom, čiže je zabezpečená plná kompatibilita s aplikáciou. Pre škálovanie a zapojenie do clusteru je zabezpečená pokročilá replikácia. MariaDB verzie 10 vie spustiť niektoré príkazy, replikované z hlavnej databáze, na dcérskych databázach paralelne (MariaDB, 2017)

4.3.5 HTML5 Boilerplate + Twitter Bootstrap

HTML5 Boilerplate je základná šablóna od spoločnosti Google, ktorá hlavne rieši kompatibilitu vzhľadu aplikácie naprieč všetkými mobilnými prehliadačmi, založená na HTML5 a CSS3. Podľa Hogana, HTML5 obsahuje nové elementy definujúca štruktúru, čo znamená že nemusíme používať zdrojové kódy naviac

alebo zásuvné moduly a ďalšie doplnky. (Hogan, 2011) V tejto práci je dôležitá aj čiastočná spätná kompatibilita, avšak nemôže ísť na úkor rýchlosti aplikácie.

Prehliadače používajú ich základné štýly na elementy, ktorých vlastnosti programátor bližšie nešpecifikuje. Problém je že tieto štýly ma každý prehliadač iné. Preto framework obsahuje súbor normalize.css, ktorý zabezpečuje rovnaké zobrazenie základných vlastností naprieč všetkými prehliadačmi. (Manian, 2012)

Frontendový framework Twitter Bootstrap ponúka hlavne zjednodušenie vývoja responzívneho vzhľadu, nakoľko prináša mriežkový systém zarovnávania blokov do mriežky, ktoré pri zariadeniach s menším displejom ako sú smartfóny, sa skladajú pod seba a netreba pre každé rozlíšenie písať výnimky v CSS súboroch. Taktiež prináša základnú funkcionality napríklad roletového menu alebo modálnych okien, čím odpadá vyhľadávanie potrebných súčastí a ich implementácia z tretích strán.

4.3.6 jQuery a React.js

Pre zabezpečenie dynamickosti riešenia je vhodné použitie kombinácia JavaScriptových knižníc jQuery a React.js. Samotná knižnica React.js je výhodná na použitie uchovávanie stavu aplikácie prostredníctvom jej state managementu, napríklad pri použití vo filtrovaní produktov. JQuery je monolitická knižnica so širokou funkcionality, ktorá je potrebnou súčasťou vyššej spomínaného frameworku Twitter Bootstrap, kde sa používa ako podpora funkčnosti pre rozšírenia. Ich kombináciou v tejto práci chcem dosiahnuť kvalitnejšiu používateľskú skúsenosť.

4.3.7 XML prenos dát

XML sa podľa Koska využíva pre prenos dát medzi backedami jednotlivých aplikácií a v Enterprise aplikáciách ktoré nevyužívajú prosté XML ale komplexnejší mechanizmus webových služieb (Kosek, 2009). XML bude slúžiť ako formát pre výmenu dát medzi e-commerce systémom a skladovým systémom.

4.4 Návrh štruktúry databáz

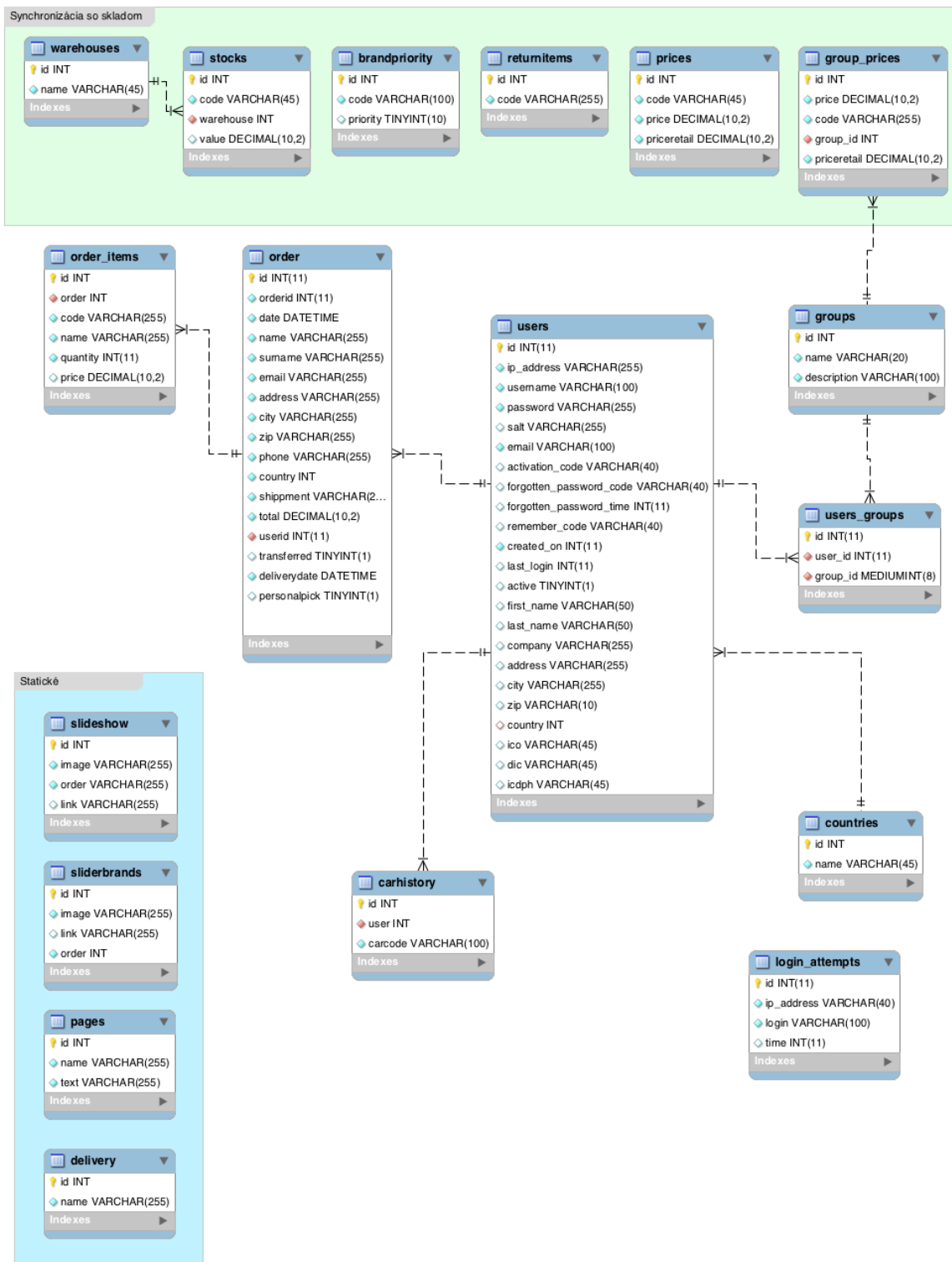
Nakoľko celý systém sa bude skladať z viacerých databáz reprezentujúcich jednotlivé katalógy a centrálny systém, treba ich štruktúru naplánovať tak, aby v čo najvyššom stupni spĺňali databázové normálne formy. Normalizácia predstavuje predovšetkým odstránenie redundantných (nadbytočných) dát a prevenciu vzniku aktualizáčnych anomálií pri vkladaní, rušení a editácií dát (Otte, 2013). Jednotlivé formy sú odstupňované nasledovne:

- ONF – aspoň jeden atribút obsahuje viac než jednu hodnotu (je neatomický)
- 1NF – každý atribút obsahuje iba atomické hodnoty a každý riadok obsahuje primárny kľúč

- 2NF – tabuľky musia spĺňať 1NF a zároveň každý neklúčový atribút je plne závislý na kandidátnom kľúči – zavedenie väzieb typu 1:N medzi tabuľkami
- 3NF – všetky neklúčové atribúty musia byť navzájom nezávislé – využívanie cudzích kľúčov k priradeniu
- Boyce-Coddova normálna forma (BCNF) – atribúty, ktoré nie sú súčasťou primárneho kľúča musia byť nezávislé – uloženie dát na menšie relácie, zhodné s 3NF, zväčša platí, že ak je tabuľka v 3NF spĺňa aj BCNF

Tieto stupne budú stačiť, aby databáza zbytočne neobsahovala duplicitné dáta a jednotlivé prvky boli previazané pomocou cudzích kľúčov, ktoré odkazujú na ostatné prvky.

4.4.1 Centrálna databáza



Obr. 9 Schéma centrálnej databázy navrhnutá v programe MySQL Workbench

Základom celého systému bude centrálna databáza, ktorá bude obsahovať tabuľky s dátami o zákazníkoch, objednávkach, stavy skladov a ceny. Nakoľko ostatné databáze budú fungovať ako katalógy, táto jediná databáza bude synchronizovaná so skladovým systémom. Popis jednotlivých tabuliek:

Tabuľka users, groups, users_groups

Tabuľka users obsahuje údaje o používateľoch v systéme. Nakoľko v aplikácii sa plánuje využitie knižnice Ion Auth, jej základ je daný štruktúrou podľa autora. Do tabuľky používateľov sa doplnia len možné fakturačné údaje, nakoľko väčšina používateľov budú zákazníci. Každý používateľ môže byť členom viacerých skupín, kde prostredníctvom väzby N:M v tabuľke users_groups sú spojené používateľské skupiny – groups a tabuľka používateľov users. Tabuľka má cudzí kľúč na tabuľku countries, kde sú uložené jednotlivé krajiny.

Tabuľka login_attempts

Tabuľka je rovnako základom prihlasovacej knižnice, kde sa ukladajú pokusy o prihlásenie na základe prihlasovacieho mena (emailu), času a IP adresy. Tabuľka slúži ako podpora ochrany prihlásenia, kedy budeme vedieť po N pokusoch o prihlásenie vyvolať želanú akciu.

Tabuľka countries

Tabuľka obsahuje krajiny, kam spoločnosť dodáva tovar.

Tabuľka prices a group_prices

Tabuľka obsahuje ceny jednotlivých produktov, kde aplikácia bude referovať unikátny kód produktu podľa stĺpca code. Tabuľka sa bude synchronizovať so skladovým systémom. Tabuľka group_prices je pomocou parametru group_id previazaná s tabuľkou groups, kde bude možné tvoriť individuálne cenové skupiny priradené k používateľovi. Každá z tabuliek obsahuje aj odporúčanú maloobchodnú cenu v parametri priceretail.

Tabuľka order a order_items

V daných tabuľkách sa budú nachádzať objednávky používateľa, kde pomocou cudzieho kľúču userid bude možné tieto objednávky prepojiť s existujúcim používateľom v databázi. Fakturačné polia sú zdublikované a to z dôvodu možnej zmeny fakturačnej adresy v budúcnosti. Ak by boli tieto informácie bola výlučne previazané s používateľom a jeho kontom, nastali by dva možné problémy. Prvý je, že používateľovi sa môžu fakturačné údaje meniť, čo by samozrejme ovplyvnilo aj jeho historické objednávky a údaje by nesedeli s údajmi zadanými v minulosti. Druhý problém by mohol nastať zmazaním používateľa, kedy by nebolo možné odkazovať na jeho údaje, nakoľko by sa už v databáze nenachádzali. Preto sa v tabuľke tieto údaje uložia v dobe zadania objednávky. Pre uloženie dátumu dodania (oneskorenie objednávky) sa používa parameter deliverydate, ktorý obsahuje aj čas dodania pre možnosť dodania v poobedňajších hodinách. V prípade osobného vyzdvihnutia na pobočke je zavedený parameter personalpick. Na

objednávku v tabuľke order_items viažu objednané položky. Obe tabuľky si pomocou prepojenia sťahuje skladový systém, je použitý parameter transferred, aby nedochádzalo k prenosu už prevzatých objednávok.

Tabuľka stock a warehouses

Tabuľky slúžia na uchovávanie aktuálnych skladových hodnôt, pričom sú previazané so skladovým systémom. Tabuľka warehouses slúži na zoznam skladov – v našom prípade centrálny sklad a pobočky a tabuľka stock na množstvá jednotlivých položiek na jednotlivých skladoch, ktoré sa podobne ako ceny produktov párujú pomocou kódu produktu.

Tabuľka brandpriority

Tabuľka slúži na nastavenie priority zobrazovania jednotlivých značiek v katalógu. Páruje sa so značkami v jednotlivých katalógoch pomocou unikátneho kódu, ktorý je rovnaký aj v TecDocu aj v skladovom systéme s ktorým je tento zoznam synchronizovaný. Škála priorít je od 0 (nezobrazovať) do 10 (maximálna priorita), pričom základný stupeň je 5.

Tabuľka returnitems

Tabuľka obsahuje zoznam položiek, ktoré je možné nepoškodené vrátiť v prípade že diel nebude sedieť.

Tabuľka carhistory

Tabuľka obsahuje historicky vyhľadávané vozidlá používateľa v katalógu TecDoc, ktoré sa párujú pomocou ID používateľa a unikátneho kódu vozidla.

Tabuľky slideshow a sliderbrands

Tabuľky obsahujú jednotlivé obrázky buď v slideshow alebo v slideri na úvode stránky, ktoré majú nastavené poradie a voliteľné prelinkovanie.

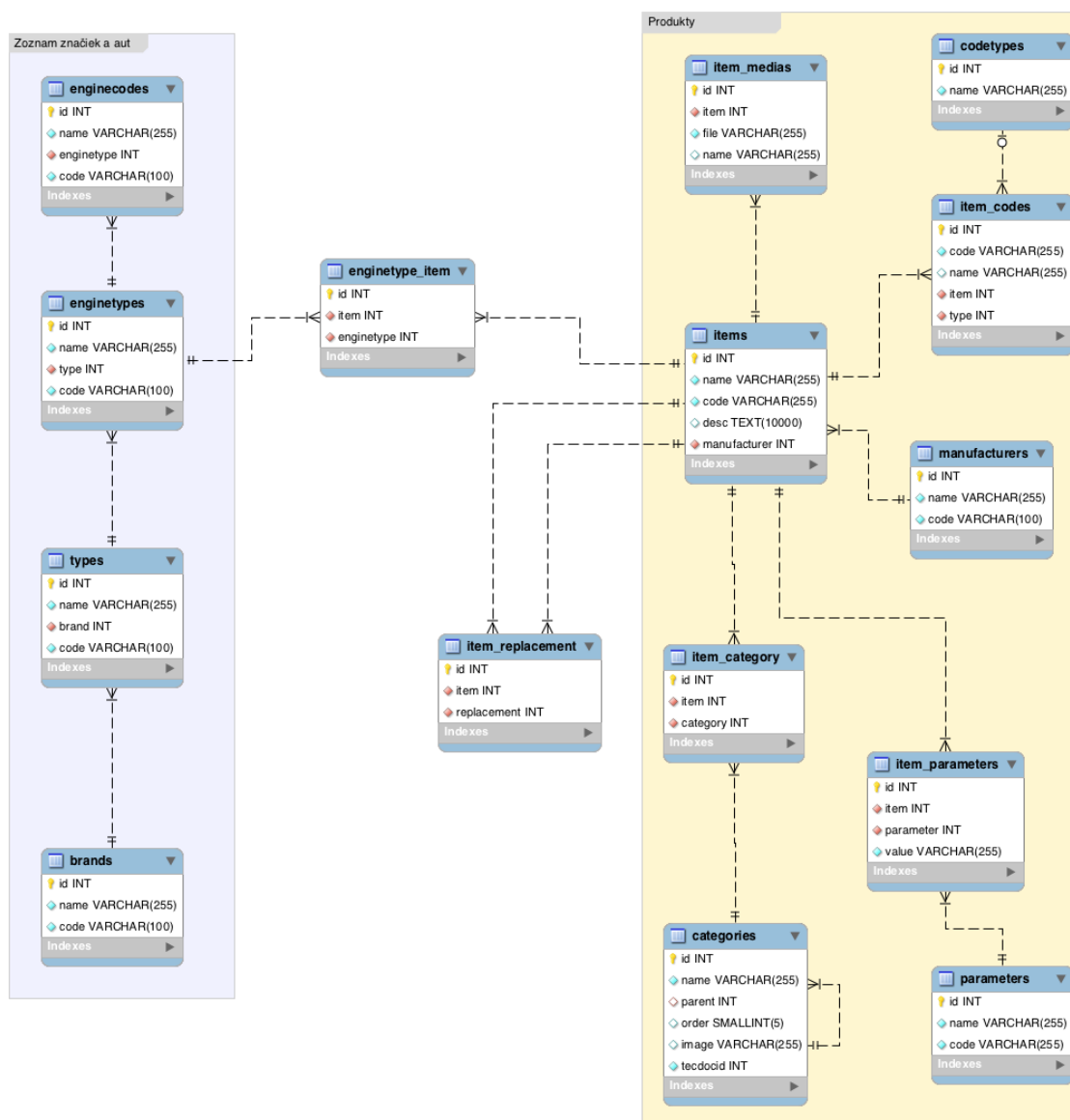
Tabuľka delivery

Tabuľka obsahuje zoznam možností dopravy.

Tabuľka pages

Tabuľka obsahuje texty statických stránok ako Obchodné podmienky alebo Kontakt.

4.4.2 Databáza TecDoc



Obr. 10 Schéma databázy TecDoc navrhnutá v programe MySQL Workbench

Databáza TecDoc obsahuje vlastné tabuľky, kde sú dve veľké skupiny a to zoznam automobilov podľa značiek, typu a motorizácie a zoznam produktov. Tieto dve skupiny sú navzájom prepojené podľa takzvaných aplikačných skupín, kde je definovaná množina navzájom kompatibilných produktov a vozidiel.

Tabuľky brands, types, enginetypes a enginecodes

Tabuľka brands obsahuje zoznam značiek automobilov ako napríklad Škoda. Jednotlivé typy (Octavia, Fabia) obsahuje tabuľka types, ktorá je prepojená s tabuľkou brands. Typy motorizácie obsahuje tabuľka enginetypes, pričom je

prepojená na typ automobilu. Tabuľka `enginecodes` obsahuje kódové označenia motorov u niektorých výrobcov, keďže pre jeden typ motorizácie sa niekedy používa rozdielne označovanie kódov motorov. Každý prvok v tabuľkách obsahuje unikátny kód, podľa čoho sa dá identifikovať jasne výrobca, typ, motorizácia alebo kódové označenie motoru.

Tabuľky `items`, `item_codes`, `codetypes`, `item_medias`, `manufacturers`, `categories` a `item_category`

Centrálnou tabuľkou je tabuľka `items`, ktorá obsahuje všetky produkty. Produkt má základný kód, ostatné voliteľné kódy produktov sa nachádzajú v tabuľke `item_codes` a jednotlivé typy kódov v tabuľke `codetypes`. Produkt môže mať mediálne súbory ako obrázky alebo PDF súbory, kde cesta k nim je uložená v tabuľke `item_medias`. Výrobca produktu sa nachádza v tabuľke `manufacturers`, pričom tento výrobca má vlastný unikátny kód. Produkt je zaradený v kategórií, ktoré sa nachádzajú v tabuľke `categories`, ktoré obsahujú meno, poradie, ID rodičovskej kategórie a prípadne obrázok kategórie. Produkt môže sa nachádzať vo viacerých kategóriách, väzba M:N je v tabuľke `item_category`.

Tabuľky `item_parameters` a `parameters`

V tabuľkách sa nachádzajú parametre, pričom názvy parametrov sú definované v tabuľke `parameters` a tabuľka `item_parameters` obsahuje hodnoty parametrov a prepojenie na názov parametru a produkt.

Tabuľka `item_replacement`

Tabuľka obsahuje dvojice navzájom nahraditeľných produktov. Odkazuje na ID produktov.

Tabuľka `enginetype_item`

Tabuľka tvorí spojenie medzi jednotlivými druhmi motorizácie a produktami väzbou M:N.

4.5 Grafický návrh frontendu

Spolu s konateľom sme nakreslili jednoduché wireframy, ktoré sme ďalej posunuli na grafické spracovanie. Wireframy – alebo drôtené modely – sú podkladom na spracovanie grafického návrhu e-commerce systému. Odrážajú presný pohľad na rozmiestenie prvkov na stránkach eshopu. (Mikulášková, 2015).

Grafik spoločnosti navrhol jednotlivé časti frontendu tak, aby boli podľa predstáv konateľa spoločnosti a súčasne spĺňali funkčné hľadisko projektu. Na základe týchto návrhov sa grafické návrhy prenesú do konečnej podoby vo formáte HTML + CSS. Grafiku frontendu ako napríklad logá a ikon pre zachovanie kvality je vhodné ukladať do vektorového formátu SVG. Ostatné obrázky budú podľa potreby sa ukladať do formátov PNG a JPEG. Hlavný rozdiel medzi nimi je ten, že

bitmapové obrázky majú pevne nastavené PPI¹, takže na obrazovke s nízkym DPR² vyzerajú pekne, ale na obrazovke s vysokým DPR nebudú príliš lahodiť oku. (Gasston, 2015)

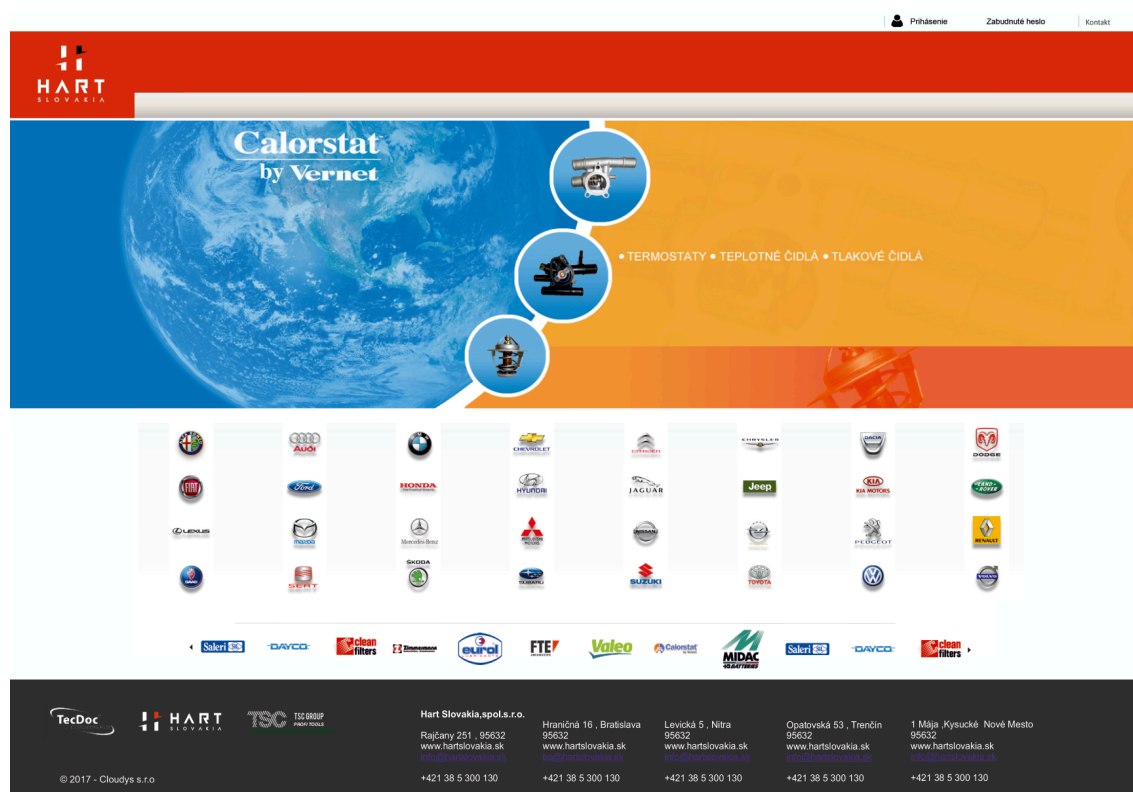
Každá stránka bude pozostávať z hlavičky a pätičky. V hlavičke sa pre neprihlásených používateľov bude zobrazovať len logo spoločnosti a vpravo hore prihlásenie a kontakt. Po prihlásení sa v hlavičke budú zobrazovať jednotlivé katalógy, možnosť fulltextového vyhľadávania a nákupný košík. Hlavička bude obsahovať panel akcií, ktorého obsah sa bude meniť s ohľadom na aktívnu stránku. Pätička bude statická, bude obsahovať logá partnerov a kontaktné údaje na jednotlivé pobočky spoločnosti.

Prihlásený používateľ bude mať dostupný vlastný profil, kde si bude môcť meniť prihlasovacie údaje ako email a heslo a zobrazovať historické objednávky. Menu pre každý katalóg bude zobrazované vľavo a na mobilných zariadeniach by sa malo zobraziť pomocou prepínaču pre ušetrenie vertikálneho miesta.

¹ PPI – Points Per Inch je hustota bodov na palec, často označovaná aj ako DPI. Štandardná hodnota pri displayoch je 96.

² DPR – Device Pixel Ratio je pomer medzi fyzickými a logickými pixelmi (bodmi), kedy je obraz prepočítaný z logického rozmeru na fyzický tak, aby na vyššej hustote bodov na palec nestratil obraz svoje proporcie.

4.5.1 Úvodná stránka



Obr. 11 Úvodná obrazovka systému

Jadro statickej stránky bude pozostávať z centrálnej slideshow, kde sa bude prelínať akciová ponuka spoločnosti, ktorú bude možné meniť v administrácii, log jednotlivých dodávaných značiek a horizontálneho slideru značiek náhradných dielov. Tieto logá značiek bude možné meniť v administrácii, logá výrobcov automobilov budú statické.

4.5.2 Katalóg TecDoc

The screenshot displays the TecDoc catalog interface for finding oil filters. The top navigation bar includes the HART SLOVAKIA logo and various vehicle categories. A search bar is present with the text 'Zadajte hľadany výraz'. The main content area shows search results for 'CLEAN FILTER' oil filters, listing items like DO 1815, DO 930, and PH 9727. Each item includes a small image, a description, and pricing information (MOC bez DPH and MOC s DPH). A table at the bottom provides a summary of the search results.

Náhrady	Vlastnosti	Použití	OE čísla	Obrázky	MOC bez DPH	MOC s DPH	Služby
DO 1815	Clean Filter	Olejový filter	2,70	3,24	RA >0	12 H >5	
Z 449	Misfat	Olejový filter	6,70	8,04	RA >0	24 H >5	
SM 158	SCT Germany	Olejový filter	2,85	3,42	RA >0	24 H >5	
H 26	Mecafilter	Olejový filter	2,70	3,24	RA >0	24 H >5	

Obr. 12 Detailný náhľad do kategórie katalógu TecDoc

Pri prepnutí sa do katalógu TecDoc nákupný proces bude mať dve možnosti:

1. Vyhľadanie dielu pomocou fulltextového vyhľadávača – kedy sa zobrazia iba výsledky nájdených dielov bez nutnosti výberu auta.
2. Prechádzanie katalógu v dvoch krokoch – najskôr používateľ vyberie auto v roletovom zozname a to štýlom značka auta, druh auta a typ motorizácie. Následne vľavo si vyberie kategóriu produktu a zobrazia sa dané položky pre dané vozidlo.

Roletový zoznam pre výber značky bude súčasťou hlavičky. Pre rýchlejšiu orientáciu v minulosti hľadaných vozidlách bude naľavo rolovacie menu s históriou výberu automobilu. Naľavo v centrálnej časti sa nachádza stromová štruktúra kategórií, ktoré pre zjednodušenie majú iba hlavnú kategóriu a pod kategóriu prvej úrovne. Strednú časť zaberá filter, kde je možné filtrovať želané značky podľa kategórie, prípadne zobrazit' všetky.

Produkty sú zoskupené podľa výrobcov, každý produkt obsahuje jeho unikátne číslo, obrázok, názov, krátky popis, cenu a stav na skladoch. Po kliknutí na oko sa rozbalí detail produktu, v ktorom sú dôležité náhrady produktu – tabuľka kompatibilných dielov podľa TecDocu. Ďalej doplnkové vlastnosti, použitie, OEM čísla a prípadné doplnkové obrázky. Kliknutím na košík sa produkt vloží do košíku.

4.5.3 Nákupný košík

Vratka	PLU	Výrobca	Názov položky	Skladom RA	24h	Malobchodná cena bez DPH	s DPH	Nákupná cena bez DPH	s DPH	Objednané	Cena celkom bez DPH	s DPH
X	0001243	Filtern	Destilovaná voda	>5	>0	10,85€	13,02€	5,00€	6,00€	- 2 +	10,00€	12,00€

Import súborov

Typ súboru : (kód , výrobca , množstvo)

Nahrať objednávku : (csv , txt)

Spolu	bez DPH	S DPH
	10,00€	12,00€

Obr. 13 Nákupný košík riešenia

Nákupný košík je spoločný pre všetky katalógy, kde vybrané položky združuje do objednávok. Zaujímavá položka je prvý stĺpec – vratka, znamená možnosť vrátenia nepoužitého dielu v prípade nekompatibility. Ďalej sú tu položky ako PLU (kód produktu), výrobca, názov produktu, stav na sklade, necenenie, počet kusov a konečná cena. Od konateľa prišla požiadavka na implementáciu funkčnosti zo starého systému, kedy nahrať CSV alebo textového súboru vo formáte kód, výrobca a množstvo bolo možné hromadne pridať položky do košíku. Po potvrdení sa prejde ku konečnej rekapitulácii objednávky.

The screenshot shows the checkout page of the Hart Slovakia website. At the top, there is a navigation bar with the Hart Slovakia logo and various icons for product categories. Below this is a search bar and a user account link. The main content area features a table of items in the cart:

Vratka	PIU	Výrobca	Názov položky	Skladom RA	24H	Malooberhodná cena bez DPH	s DPH	Nákupná cena bez DPH	s DPH	Objednané	Cena celkom bez DPH	s DPH
✓	0001243	Filoon	Destilovaná voda	>5	>0	10,85€	13,02€	5,00€	6,00€	2	10,00€	12,00€

Below the table, there is a 'Možnosti:' section with a dropdown menu for 'Doprava:' set to 'Prieváža', a date selector for 'Deň doručenia:' set to '23.7.2017', and a text area for 'Poznámka k objednávke:'. A checkbox for 'Súhlasím so všeobecnými obchodnými podmienkami' is checked. A large red 'Objednať' button is positioned to the right. The footer contains logos of various brands and contact information for Hart Slovakia, spol. s r.o. across multiple locations.

Obr. 14 Výsledné okno rekapitulácie objednávky

Po schválení obsahu nákupného košíku sa vyberie spôsob a dátum dopravy, prípadné doplnkové potvrdenie a poznámka k objednávke. Dôležitým prvkom je súhlas s obchodnými podmienkami spoločnosti. Po odoslaní objednávky už ďalej systém spracuje objednávku.

4.6 Návrh administráčnej časti

Administráčna časť, alebo backend slúži povereným osobám na spravovanie vybraných položiek v systéme. Administrátori budú mať možnosť upravenia typov dodania, statických stránok, sliderov na úvode, kategórií atď...

Vzhľad administrácie bude riešený takisto responzívne. Nakoľko existuje viacero predpripravených voľne dostupných šablón administráčného systému, rozhodol som sa pre výber šablóny AdminLTE 2 od dizajnéra Abdulla Almsaeeda (Almsaeed, 2017). Šablóna ponúka už predpripravené časti pre administráciu ako prihlasovanie, tabuľky, grafy, komplexné menu či editory. Založená je na Bootstrap 3 frameworku, spĺňa parametre responzivity a je jednoducho upraviteľná.



Obr. 15 Demonštračná stránka šablóny administrácie AdminLTE 2
Zdroj: AdminLTE2, Almsaeed, 2017

Pre administračnú časť, nakoľko sa často bude opakovať jednoduchá úprava tabuliek, je výhodné použiť generátor CRUD tabuliek. Skratka CRUD vznikla z anglického Create, Read, Update a Delete, čo sú základné funkcie pre perzistentné úložisko (Heller, 2007). Framework sám o sebe generátor alebo knižnicu neponúka, bude treba nejakú knižnicu z tretej strany nájsť a implementovať ju.

Pre úpravu statických stránok je vhodné nasadenie WYSIWYG editoru. WYSIWYG editor ponúka správcovi zobrazenie konečného výsledku, skratka vznikla z prekladu anglického „What you see is what you get“ teda čo vidíte to dostanete. (Rouse, 2011) Tento editor bude vhodný pre menej technický zdatných používateľov, aby vedeli upravovať obsah statických stránok prípadne popisov podobne, ako sú zvyknutý z editoru Microsoft Word.

5 Implementácia systému

Implementácia systému prebehne podľa plánu, v tejto kapitole rozpíšem detaily a zaujímavosti, s ktorými som sa počas implementácie stretol. Využíval som hlavne poznatky ľudí, ktorý sa s podobnými problémami už stretli. Programovanie aplikácie a implementáciu som vykonával na webovom serveri MAMP Pro pre operačný systém MacOS a programoval pomocou IDE PhpStorm.

5.1 Implementácia centrálnej časti

Centrálnu časť tvorí vyššie spomínaný Framework Codeigniter. Framework je voľne dostupný na stiahnutie na oficiálnych stránkach, v dobe písania tejto práce bola najnovšia stabilná verzia 3.1.4. Framework sa stiahne ako archív a rozbalí do príslušného priečinku. Celý framework sa skladá z dvoch základných častí a to priečinku application – kde sa nachádza kód samotnej aplikácie a priečinku system – kde sa nachádza jadro frameworku. Pre ukladanie externých súborov ako sú CSS, JS, obrázky a ostatné súbory som v koreni vytvoril priečinok assets. Schému centrálnej databázy som vyexportoval priamo v MySQL Workbench a cez phpMyAdmin naimportoval na databázový server.

5.1.1 Nastavenie frameworku

Najskôr som nastavil v konfiguračnom súbore config.php, ktorý sa nachádza na application/config/ čo je priečinok s nastaveniami frameworku základnú cestu k projektu a jazyk projektu. V konfiguračnom súbore databázy som nastavil údaje na pripojenie k základnej databáze – centrálnej. Ostatné databázy je možné nastaviť ako vedľajšie a pripojiť ich obdobným spôsobom, čo bude rozobraté v ostatných častiach. Pre automatické pripojenie k databázi, treba databázovú triedu zadať do zoznamu knižníc, ktoré sa automaticky načítavajú v súbore autoload.php.

Vytvoril som hlavný kontroler aplikácie – Site, ktorý v konfiguračnom súbore ciest (routes.php) som nastavil ako predvolený. Pre test funkčnosti stačí vytvoriť funkciu triedy index() a skúsiť výpis prostredníctvom príkazu echo. Možné chyby v tejto fázy sú zlé nastavenie pripojenia k databázovému serveru – adresa, meno, heslo, databáza a port, prípadne zle napísaná cesta v routes.php. Codeigniter vyžaduje mená jednotlivých kontrolerov aby začínali veľkým písmenom.

Pre nastavenie základných ciest k jednotlivým katalógom som si vytvoril konfiguračný súbor catalogues.php. Tieto nastavenia sa potom jednoducho nahrávajú pomocou príkazu:

```
$this->config->item(basepath', 'tecdoc')
```

5.1.2 Inštalácia potrebných zásuvných modulov

Pre prihlasovanie a správu používateľov som použil voľne dostupnú knižnicu Ion Auth od programátora Bena Edmunda (Edmunds, 2017). Knižnica obsahuje nasledovné funkcie:

- Prihlasovanie a registrácia používateľov
- Správa používateľov administrátorom
- Tvorbu používateľských skupín
- Funkcia zabudnutého hesla
- Ochrana účtu pri viacnásobných prihláseniach
- Použitie robustného šifrovania Bcrypt
- Vlastné spúšťače (triggery) podľa potreby programátora

Rozvrhnutie tabuliek databázy a základné prihlasovacie údaje boli už implementované do návrhu. Nakoľko prihlasovanie používateľov budeme potrebovať všade na stránke, je výhodné si knižnicu automaticky načítavať. S ňou treba načítavať aj dodatočné knižnice session – správa sedení a database – pripojenie na databázu, ktorú sme si už načítali vyššie. Aplikácia má vlastný kontroler, ktorý slúži ako administrácia používateľov, ktorý bude neskôr implementovaný do administrácie stránky.

Framework sám o sebe neobsahuje šablónovací systém, no je preň dostupná jednoduchá knižnica od programátora Jérôme Jaglale (Jaglale, 2009) na správu šablón tak, aby bolo možné oddeliť opakujúce sa časti ako hlavička a pätička a centrálnu časť, ktorá sa mení. Pomocou tejto knižnice sa jednoducho dajú aj definovať vlastné premenné pre view ako napríklad titulok stránky.

Pre tvorbu CRUD funkcionality v administrácií som použil zásuvný modul GroceryCRUD (Skoumbourdis, 2017) s bohatou funkcionalitou:

- Tvorba CRUD funkcií priamo v Codeigniteri
- Možnosť väzieb 1:N a M:N tabuliek
- Zabudovaný mechanizmus pre nahrávanie súborov
- Zabudovaný CKEditor WYSIWYG Editor a podpora iných editorov
- Možnosť využitia rôznych šablón
- Export tabuliek do formátu CSV

Knižnicu treba nastaviť v jej konfiguračnom súbore, kde som nastavil jazyk, CKEditor editor, formát dátumu, povolené súbory na nahrávanie a šablónu. Pre editor CKEditor som doinštaloval jednoduchého správcu súborov KCFinder, ktorý dokáže spravovať a nahrávať súbory priamo do editora. Správca súborov obsahuje aj náhľad bežných typov súborov ako sú obrázky alebo PDF súbory, pre ich jednoduchšiu identifikáciu.

Pre rozsiahle tabuľkové výpisy som implementoval do administrácie doplnok Datatables, ktorý prostredníctvom asynchrónnych AJAX volaní dokáže vyhľadávať a spravovať aj väčšie množstvá riadkov – príklad na ich domovskej stránke, kde demonštrovali nastavenie doplnku pre tabuľku s 5 miliónmi riadkov (Jardine,

2017). Pre tento doplnok navrhnutý v JavaScriptovej knižnici jQuery je nutné mať serverový obslužný skript, ktorý naprogramoval pre Codeigniter Vincent Bambico a kolektív (Bmbamico a kol., 2015).

5.1.3 Príprava administratívnej časti

Súbory šablóny administratívnej časti som najskôr nakopíroval do separátneho priečinku v assets – AdminLTE 2 tak, aby boli oddelené od zvyšných súborov. Z šablóny administrácie som odstránil nadbytočné a nepotrebné veci a rozdelil ju na hlavičku s menu, centrálny obsah a pätičku pomocou knižnice na šablóny. Vytvoril som vlastný kontroler Admin. Do tohto kontroleru som implementoval základnú časť pre správu používateľov z kontroleru knižnice Ion auth, a implementoval prihlasovacie okno do šablóny AdminLTE 2. V konštrukture administrácie je zadefinované presmerovanie neprihlásených používateľov na prihlasovacie okno a zabránené prihlasovanie nepovolaným používateľom – mimo skupiny administrátorov. Správa používateľov je implementovaná kombináciou rozšírenia Datatables a vzoru od programátora rozšírenia Ion Auth. Správu tabuľky krajín som pripravil pomocou rozšírenia GroceryCRUD, pričom bolo potrebné nastaviť vkladanie potrebných CSS a JS súborov rozšírenia do šablóny. Prostredníctvom GroceryCRUD a editoru CKEditor sa spravujú aj statické stránky ako Kontakt či Obchodné podmienky.

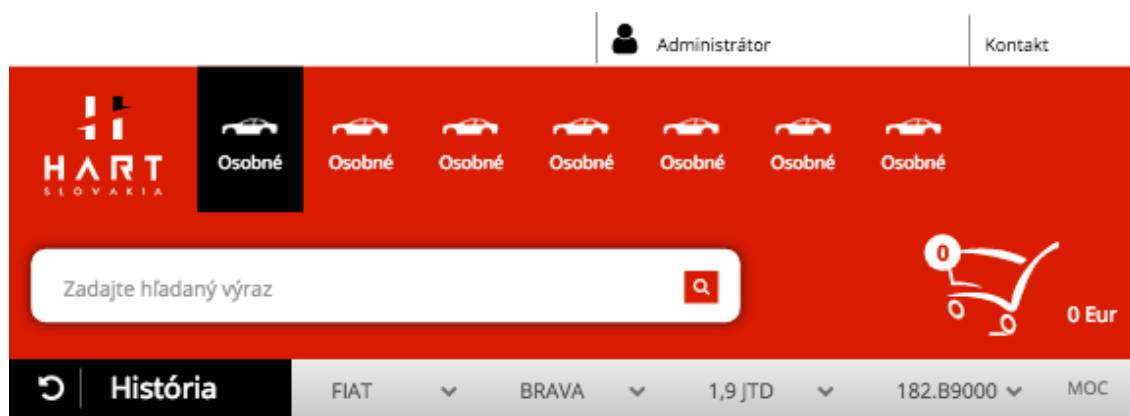
5.1.4 Programovanie frontendu – hlavička a pätička

Ako základ frontendu je použitá kombinácia frameworkov HTML5 Boilerplate a Bootstrap 3. Podľa grafických návrhov bol frontend preprogramovaný do HTML a CSS súborov, dynamické prvky boli spracované prostredníctvom JavaScriptu, respektíve frameworkov pre JavaScript – jQuery a React.js.

Frontend je rozdelený na hlavičku, centrálnu časť a pätičku, pričom hlavička a pätička je na všetkých stránkach rovnaká s drobnými obmenami. Tieto časti som pomocou šablónovacej knižnice oddelil a postupne spracoval.

Hlavička obsahuje logo spoločnosti, ikony s jednotlivými katalógmi (ktoré budú do budúcnosti pridávané), vyhľadávacie pole, košík, panel s akciami – ktoré budú rôzne v závislosti na katalógu a prepojenie na prihlásenie v hornej časti okna. Časti okrem loga a prihlasovacieho okna budú dostupné až po prihlásení používateľa. Prihlasovanie používateľa obsahuje ikonku osoby zo základnej sady vektorových ikon frameworku Twitter Bootstrap. Logo stránky som previedol do formátu SVG, čo je otvorený vektorový formát, kvôli malej veľkosti a zachovaniu kvality na obrazovkách s vysokým rozlíšením. Podobne aj ikonky jednotlivých katalógov budú vo formáte SVG. Vyhľadávacie pole má tiež generovaný pomocou atribútu v CSS3 a ikonku vyhľadávania, použitej zo základnej sady. Obrázok košíka je dodaný od grafika, taktiež vo formáte SVG. Nakoľko prehliadač Internet Explorer 11 a starší nepodporuje SVG formát v CSS ako pozadie, bolo nutné ho vložiť do stránky ako bežný obrázok a nastaviť absolútne pozicovanie v CSS. Posledná časť hlavičky, panel s akciami, obsahuje farebný prechod ako pozadie, ktorý je

vykreslený prostredníctvom CSS. Obsah tohto panelu sa mení podľa otvorenej stránky, zvyčajne bude obsahovať omrvinkovú navigáciu. Pre katalóg TecDoc sa tu bude nachádzať roletové menu pre výber auta a roletová ponuka historicky zvolených modelov. Táto ponuka je tvorená ako select boxy základného HTML, kde šípka na roztvorenie menu je v každom prehliadači iná a tento fakt esteticky kazí dojem z hlavičky. Preto som pomocou CSS toto správanie zakázal, nahradil šípku ikonkou zo sady frameworku a prekryl select box pre prirodzenejší vzhľad. Pomocou mriežkového systému Bootstrapu som nastavil bloky tak, aby sa na rôznych zariadeniach skladali pod seba a pôsobili prirodzene. Niektoré prvky bolo nutné upraviť manuálne, dodatočne pomocou tzv. media queries, čo sú špeciálne nastavenia pre definované prostredie ako rozlíšenie displaya či typu zobrazenia. Tieto manuálne nastavenia som aplikoval pre niektoré rozlíšenia mobilných telefónov, hlavne pri orientácii mobilného telefónu na šírku.



Obr. 16 Zobrazenie hlavičky katalógu TecDoc na tablete Apple Ipad Air 2

Pätička stránky je jednoduchá časť, ktorá sa skladá z log partnerov a kontaktných údajov na jednotlivé pobočky spoločnosti. Logá partnerov boli prevedené do formátu SVG. Pätička by mala byť vždy ukotvená o spodnú časť webového prehliadača, čo som dosiahol absolútnym pozicovaním a odsadením zvyšného obsahu o výšku pätičky. Na tento úkon bolo potrebné znovu manuálne napísať media queries, nakoľko sa obsah pätičky nemení, bolo možné použiť absolútne hodnoty. V opačnom prípade by bolo nutné pri zmene rozlíšenia spočítať výšku pätičky a zvyšný obsah odsadiť prostredníctvom JavaScriptu.

5.1.5 Úvodná stránka

Úvodná stránka obsahuje slideshow s reklamami, ktoré sa horizontálne prelínajú. Pre zabezpečenie efektu som použil knižnicu pre framework jQuery Cycle 2 (Alsup, 2017). Knižnica má malú veľkosť a ponúka bohaté možnosti rozširovania. Jednotlivé obrázky sa dajú nahrávať prostredníctvom administrácie, kde je voliteľne možnosť nastaviť ku každému obrázku odkaz a poradie obrázkov. Do frontendu sa načítava cez model `site_model` a do view vrstvy je posielaný

pomocou poľa `$data`. Obdobne pracuje aj scroller značiek na úvode, využíva vyššie spomínanú knižnicu pre efekt skrolovania obrázkov a načítavanie pomocou modelu. Značky sa rovnako dajú pohodlne spravovať cez administráciu.

5.1.6 Nákupný košík

Nákupný košík aplikácie je centrálna časť, kde sa združujú všetky produkty a neskôr sa z neho tvorí objednávka. Framework Codeigniter priamo obsahuje knižnicu Shopping Cart library, ktorá obsahuje základnú podporu pre funkcionality nákupného košíku. Po odoslaní položky do košíku prostredníctvom POST metódy sa položka pridá do nákupného košíku, pričom povinné parametre sú id položky, cena, množstvo a meno položky. Taktiež je možnosť zdefinovať si voliteľné parametre produktu. Knižnica potom vie spočítať množstvo produktov v košíku, výslednú cenu, či zaobstarat' jednoduché pridávanie a odoberanie položiek. Nakoľko frontend košíku obsahuje priamo pri položke prepínač, tento som spracoval ako AJAX volanie a priamy prepočet cien v košíku. Pre AJAX volania som si vytvoril priamo vlastný kontroler a model – ajax, ktorý nemá view a priamo vracia odpovede o spracovaní vo formáte JSON. Toto jednoduché REST rozhranie aktualizuje množstvá produktov v košíku a vracia výslednú sumu produktu a celého košíku. Táto cena potom prepisuje príslušné hodnoty bez nutnosti znovu načítania stránky, čo je pre používateľsky pohodlné riešenie. Rovnako zmazanie produktu prebieha prostredníctvom AJAXu, pričom pred samotným zmazaním je používateľ upozornený hlásením, ktoré musí potvrdiť, aby nedošlo k zmazaniu položky omylom.

Funkciu importu položiek do košíku chcela z pôvodného riešenia spoločnosť zachovať, nakoľko viacerí z ich zákazníkov si robia export zoznamu položiek na objednanie z ich interných systémov a manuálne vyhľadávajú a vkladajú do košíku by im zabralo zbytočne veľa času. Pomocou formuláru sa prijímajú súbory buď vo formáte TXT alebo CSV. Mechanizmus v kontroleri zistí, o aký druh súboru sa jedná a či má dodržanú predom stanovenú štruktúru. Ak súbor prejde týmto testom, prejde sa celý riadok po riadku a v prípade súhlasu kódu produktu sa vloží do košíku, pričom skript automaticky zistí jeho cenu.

V strede košíku nájdeme tlačidlo pre zmazanie celého košíku, čo spôsobí reštart objednávkového procesu. Pre pokračovanie k metódam dodania a odoslania objednávky slúži tlačidlo prejsť k objednávke.

5.1.7 Objednávka tovaru

Objednávka tovaru je podobná s košíkom, rozdiel je len v ľavom paneli na výber dopravy – dodacej trasy, ktorý sa prostredníctvom kontroleru a modelu načítava ako zoznam z databáze. Jednu dodáciu bude mať používateľ predvolenú. Na konci boxu je voliteľný notifikačný email, poznámka a povinné potvrdenie súhlasu s obchodnými podmienkami. Dátum dopravy sa dá vybrať pomocou rozšírenia DatePicker, ktoré zobrazuje vyskakovací box s kalendárom a jednoduchší výber pre používateľa. Po stlačení tlačidla objednať sa musí ešte objednávka potvrdiť

pomocou modálneho okna, kde je výrazne vybraný spôsob dopravy s voliteľnou možnosťou dodania tovaru v poobedňajších hodinách, prípadný osobný odber.

Wartka	PIU	Výrobca	Názov položky	Skladom RA	24h	Malobchodná cena bez DPH	s DPH	Nákupná cena bez DPH	s DPH	Objednané	Cena celkom bez DPH	s DPH
✓	0001243	Filson	Destilovaná voda			10,85€	13,02€	5,00€	6,00€	2	10,00€	12,00€
											bez DPH	5 DPH
											10,00€	12,00€

Obr. 17 Potvrdenie objednávky

5.1.8 Profil používateľa

Každý prihlásený používateľ má dostupnú jednoduchú správu svojich údajov ako fakturačná a dodacia adresa a zmena prihlasovacieho hesla. Do profilu sa štandardne dostane rozkliknutím ponuky vpravo hore. Profil obsahuje najskôr fakturačné údaje, a menu na zmenu hesla. Fakturačné údaje sa jednoducho spravujú pomocou vstavanej funkcie `update()` v knižnici `ion_auth`. Pre zmenu hesla treba zadať súčasné heslo a potom nové heslo a potvrdenie nového hesla z dôvodu možného preklepu. Pre validáciu vložených údajov existuje vo frameworku pomocná knižnica `form_validator`, kde je možné pre vybrané polia zadať viacero možností overovania ako:

- Povinné položky
- Správnosť emailovej adresy
- Minimálna a maximálna dĺžka
- Zhoda poľa s iným polom
- Iba číselný/textový vstup

Centrálne v konfigurácií sa dajú zapnúť aj priama ochrana proti útokom XSS³ a CSRF⁴ tak, aby framework priamo ošetroval vstupy a formuláre. Pre formulárové vstupy ako dodacie a fakturačné adresy je zabezpečenie vítané, nakoľko sa jedná o citlivé údaje. Súčasťou používateľského profilu je aj odkaz na zobrazenie zoznamu historických objednávok a ich detailu. Táto funkcionality má len informatívny charakter.

5.1.9 Prihlásenie a zabudnuté heslo

Prihlásenie je zabezpečené pomocou knižnice `ion_auth`, kde odoslaním prihlasovacieho formuláru – emailu a hesla, sa spracuje prihlásenie pomocou funkcie `login()`, ktorá v prípade zhody zapíše nové sedenie o tom že používateľ bol prihlásený, inak vypíše chybovú hlášku o nesprávnom mene a hesle. Nakoľko registrácia nových používateľov sa spracováva v skladovom systéme, registrácia cez webovú stránku nie je možná.

Dôležitou funkciou systému je možnosť obnovenia hesla pomocou funkcie `zabudnuté heslo`. V prípade neznalosti hesla používateľ navštíví stránku `zabudnuté heslo`, kde vloží svoj email. Ak sa emailová adresa našla v systéme, odošle sa pomocou funkcie `forgetten_password()` email s kódom na resetovanie hesla. Tento kód ma obmedzenú platnosť, ktorá sa nastavuje v konfigurácií knižnice (v našom prípade 3 hodiny). Ak zákazník navštíví adresu v odoslanom emaili, samozrejme za predpokladu že je kód platný, vygeneruje sa mu nové heslo a odošle sa mu znovu na kontaktný email.

5.1.10 Administračná časť

Po prihlásení povolaným používateľom do administrácie môže používateľ spravovať používateľov, bannery na stránkach, kontrolovať objednávky, meniť prioritu značkám. Pri niektorých katalógoch je tu možnosť správy produktov, kategórií a zlučovania jednotlivých kategórií.

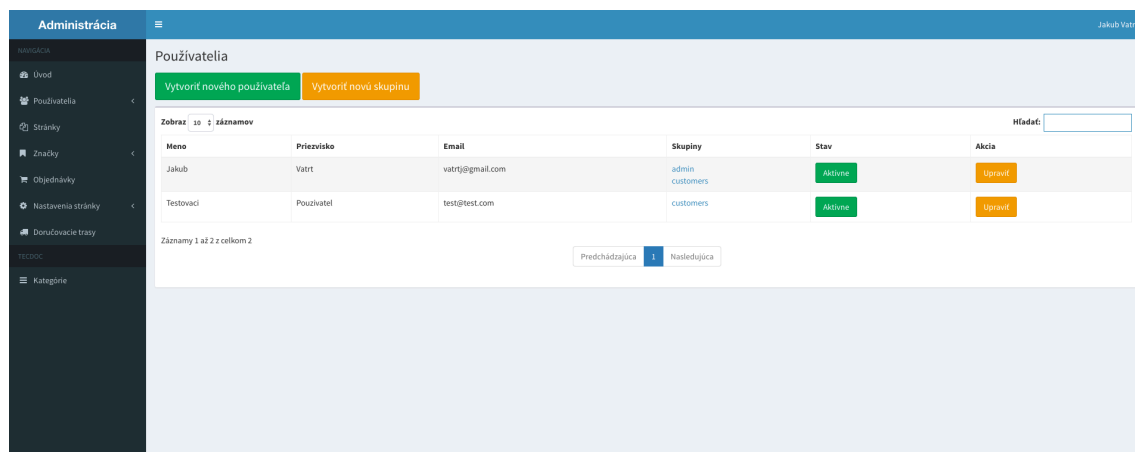
Správa používateľov

Pre správu používateľov obsahuje autentizačná knižnica `Ion Auth` už predpripravenú jednoduchú správu, ktorú som implementoval do šablóny administrácie `AdminLTE` a pridal novovzniknuté fakturačné polia. Používateľov nemožno mazať, iba ich nastaviť ako neaktívnych, čo bude riadené aj aktualizáciami pri synchronizácii z adresáru skladového systému. Používatelia majú základnú skupinu `customers`, ktorá má základné ceny, pričom ak ma používateľ inú cenovú skupinu, táto sa pri synchronizácii so skladovým systémom najskôr vytvorí, potom sa vytvorí používateľ, ktorý sa do nej priradí a potom bude

³ XSS je útok pomocou vloženia JavaScriptového kódu do neošetrených formulárových vstupov. Útok môže slúžiť k modifikácií, mazaniu alebo získaniu citlivých údajov

⁴ CSRF je útok pomocou podvrhnutia URL tak aby došlo k modifikácií prípadne zmazaniu údajov, pomocou odoslania požiadavky na cieľový server cez metódy POST a GET

nasledovať synchronizácia cien. Pre rýchlejšie vyhľadávanie používateľov som použil rozšírenie DataTables, ktoré pridáva funkciu stránkovania a fulltextového vyhľadávania. V prípade vyššieho množstva používateľov sa toto rozšírenie dá prepnúť do AJAXového režimu, kedy každá akcia je volaním prostredníctvom AJAXu priamo knižnice DataTables pre Codeigniter, ktorá je prepojená s databázou a vráti výsledok ktorý je schopné rozšírenie vykresliť do tabuľky.



Obr. 18 Vzhľad správy používateľov cez rozšírenie DataTables

Statické stránky

Statické stránky ako kontakt alebo používateľské podmienky sa spravujú pomocou rozšírenia GroceryCRUD, ktoré obsahuje viacero rozšírení pre jednoduchšiu editáciu ako napríklad WYSIWYG editor. Taktiež z pohľadu programátora použitie tohto rozšírenia na tabuľku je veľmi jednoduché. Nasledujúcim kódom som dosiahol vygenerovanie tabuľky zobrazenej na obrázku nižšie:

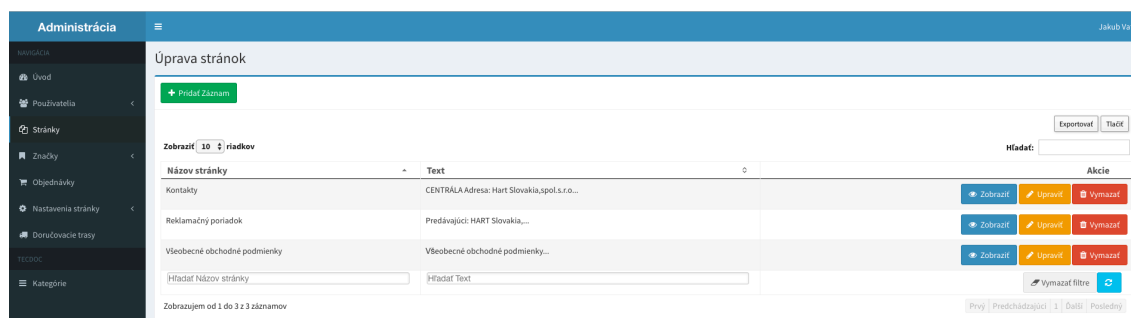
```
public function pages(){
    $data = array();

    $crud = new grocery_CRUD();
    $crud->set_table('pages');

    $crud->display_as('name', 'Názov stránky');

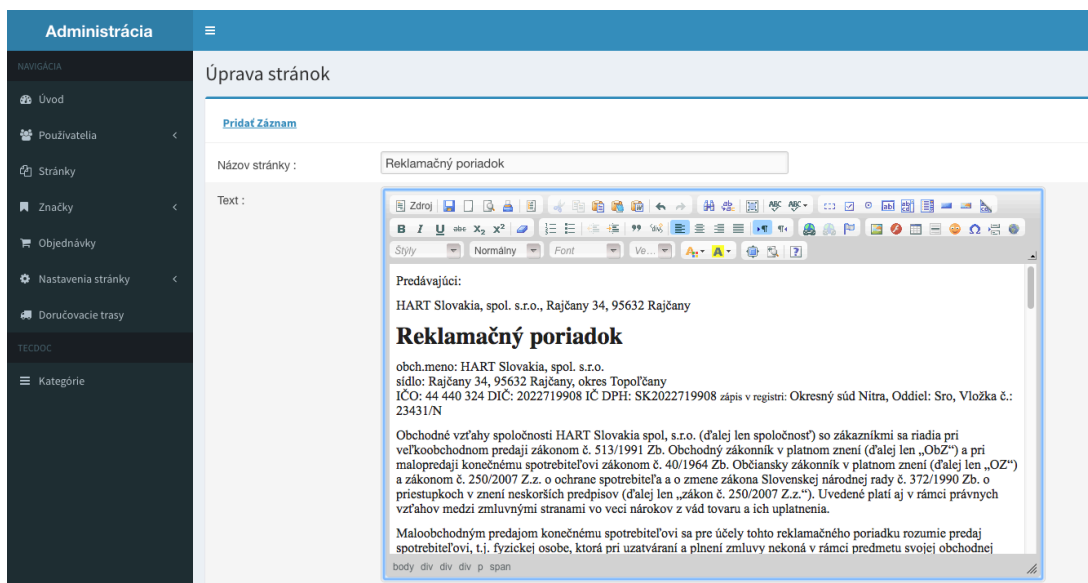
    $this->template->set('title', 'Úprava stránok');
    $data["crud"] = $crud->render();
    $this->template->set('js_files', $data["crud"]->js_files);
    $this->template->set('css_files', $data["crud"]->css_files);
    $this->template->load('admin/theme', 'admin/crud', $data);
}
```

Stačí naviť pomocou funkcie `set_table` cieľovú tabuľku a následne pomocou funkcie `render` ju vygenerovať. Zvyšné funkcie len premenovávajú alebo pomenovávajú jednotlivé polia a posielajú potrebné CSS a JS súbory do šablóny na vykreslenie.



Obr. 19 Tabuľka so statickými stránkami vygenerovaná pomocou GroceryCRUD

Tabuľka obsahuje tri tlačidlá na správu a to Zobraziť, Upraviť a Zmazať. Taktiež sa dajú záznamy ľahko exportovať do formátu CSV. Pri pridávaní alebo úprave, sa zobrazí formulár nižšie, pričom pole text sa automaticky zmení na editor CKeditor podobný Microsoft Wordu, čo je pre používateľa administrácie pohodlnejšie riešenie, než nutnosť písať HTML kód.



Obr. 20 Úprava statickej stránky s WYSIWYG editorom

Správa priorít značiek

Jednotlivé značky v systéme majú rôzne priority na zobrazenie. Ich správu som riešil tiež pomocou rozšírenia GroceryCRUD podobne ako správu statických stránok, s tým rozdielom že pole priority som zmenil na roletové menu pre zamedzenie preklepov miesto klasického vstupného poľa. To som dosiahol nasledujúcim kódom:

```
Scrud->change_field_type('priority', 'dropdown',
    array(
        '0' => '0 (najnižšia)',
        '1' => '1',
```

```

'2' => '2' ,
'3' => '3' ,
'4' => '4' ,
'5' => '5 (predvolená)' ,
'6' => '6' ,
'7' => '7' ,
'8' => '8' ,
'9' => '9' ,
'10' => '10 (najvyššia)' ,
)
);

```

Všimnite si že najnižšiu, strednú a najvyššiu prioritu som označil aj pomocným slovom pre jednoduchšiu orientáciu. Priorita sa bude načítavať a párovať so značkami pomocou ich kódu, ak ich nenájde bude základná (5).

Správa objednávok

Zoznam objednávok je znovu spracovaný pomocou GroceryCRUD, pričom tento zoznam je len záložný v prípade nefunkčnosti prenosu dát do skladového systému. Dá sa v ňom vyhľadávať podľa čísla objednávky, dátumu, zákazníka alebo jeho kontaktných údajov ako telefón a email. Každé objednávke sa dá zobrazit' detail, kde sú podrobné fakturačné údaje, vrátane zoznamu položiek, typu a dátumu dopravy.

Kód	Meno	Množstvo	Cena	Celkovo
44 015 55 200	CompacPlus	1	75,00 EUR	75,00 EUR
49 000 51 241	NLG Pico	1	25,00 EUR	25,00 EUR
Súčet				
Celkovo k úhrade				100,00 EUR

Obr. 21 Detail objednávky zákazníka v administrácii

Správa slideshow a obrázkov značiek na úvode

Jednotlivé obrázky slideshow sa spravujú tiež pomocou GroceryCRUD, pričom na nahrávanie obrázkov plugin ponúka priamo zabudovanú funkciu upload. Stačí nastaviť len meno poľa a cestu, kam sa majú obrázky nahrávať:

```
$crud->set_field_upload('image', 'assets/uploads/slideshow');
```

Pre zamedzenie nahrávania zbytočne veľkých obrázkov som pridal ešte voliteľné spracovanie obrázku po nahraní a to pomocou externej funkcie, ktorá si volá knižnicu na spracovanie obrázkov image_moo, kde prebehne zmena veľkosti nahraného obrázku.


```
$crud->callback_after_upload(array($this, '_slideshow_image_upload'));  
  
$this->image_moo->load($file_uploaded)->resize(1920,880)-  
>save($file_uploaded,true);
```

Obrázok má maximálne rozlíšenie 1920x880px, čo plne postačuje dnešným štandardom a jeho veľkosť pri formáte JPG neprekročí 500kB, čo je pri dnešných rýchlych pripojeniach štandardná hodnota pre veľkoplošný obrázok. Do databáze sa potom už len uloží meno obrázku. Následne je možné nastaviť poradie jednotlivých obrázkov, pričom základná hodnota 0 nastaví obrázok ako prvý.

Správa značiek je podobná, s tým rozdielom že veľkosť obrázku značky je nastavená na 400x400px.

Správa dodacích trás

Dodacie trasy sú jednoduchá tabuľka spravovaná pomocou GroceryCrud. Tieto predvolené trasy sú potom zobrazované používateľovi pri vytváraní objednávky.

5.2 Implementácia katalógu TecDoc

Ako bolo v návrhu databázy už načrtnuté, katalóg TecDoc sa delí na dve hlavné skupiny a to značky a typy aut a jednotlivé položky a ich kategórie. Medzi týmito dvoma skupinami sú spojovacie tabuľky, ktoré spájajú kompatibilné autá a diely – niektoré diely sú kompatibilné s viacerými druhmi aut. Taktiež pokiaľ sa originálny diel alebo iný prestal vyrábať, katalóg obsahuje zoznam náhrad za tento diel. Dodávateľ katalógu TecDoc spoločnosť TecAlliance GmbH po podpise zmluvy dodáva katalóg prostredníctvom online distribúcie – spístitupnením ich FTP serveru s dátovými súbormi. Aktualizácia katalógu prebieha pravidelne každé tri mesiace, čiže pre import dát z dátových súborov do databáze je vhodné pripraviť si automatizačný skript, ktorý je možné viacnásobne spúšťať. Spoločnosť TecAlliance dodáva spolu s dátami aj obsiahly manuál, ako dáta spracovávať, respektíve význam jednotlivých položiek a tabuliek. Nakoľko celá TecDoc databáza podlieha licenčným právam, v tejto práci sa budú nachádzať importované demo dáta, ktoré spoločnosť posielala každému záujemcovi na prípravu.

5.2.1 Importovanie položiek databázy

Dodané dátové súbory majú formát textových súborov, kódovanie UTF-8 a CRLF⁵ ukončovanie riadkov. Pomocou jednoduchého skriptu PHP som nastavil prechádzanie súborov po riadkoch a spracovával každý riadok zvlášť. Tento skript predvediem napríklad na súbore 012, ktorá je pri importe kľúčový, nakoľko obsahuje popisy, teda je vhodné si ho pripraviť do poľa tak, aby bolo jednoduché vyhľadávanie v ňom pomocou popisného čísla. Tú som spracoval nasledujúcim kódom:

⁵ Carriage Return and Line Feed (Windows) v ASCII tabuľke kód 13 (\r) a následne 10 (\n)

```
<?
$pole = array();

$subor = fopen("012.DAT", "r");
while(!feof($subor)) {
    $line = fgets($subor);

    if (substr($line, 38, 3) == "SK "){
        $pole[substr($line, 29, 9)]=substr($line, 44, 60);
    }

}
fclose($subor);

$data = serialize($pole);
file_put_contents('slovník.txt', $data);

?>
```

Pomocou prechádzania súboru 012.DAT funkciami feof (detekcia konca súboru) a fgets (načítanie jedného riadku súboru) som si do premennej line načítal jeden riadok. Tento riadok zodpovedá rozložením popisu z manuálu, takže sa môžem riadiť pozíciami jednotlivých reťazcov. Nakoľko súbor obsahuje reťazce rôznych jazykov (predpoklad) a pre databázu potrebujem reťazce so slovenskou jazykovou mutáciou, skontroloval som, či je daný reťazec v slovenčine, čo som vykonal porovnaním reťazca z 38 pozície s dĺžkou 3 znaky. Ak reťazec zodpovedá jazykovému kódu Slovenska – tj. “SK “ reťazec sa zaradí do poľa. Ako kľúč poľa som použil popisný kód, podľa ktorého som neskôr vyhľadával a ku ktorému som priradil popis. Procedúra sa opakuje na ďalšom riadku. Po prejdení celého súboru som použil na pole funkciu serialize, ktorá vytvorí reprezentáciu dát vhodnú na uloženie (PHP Group, 2017) a ktorú som uložil si do pomocného súboru slovník.txt.

Tento súbor nasledovne môžem načítať pri spracovaní ostatných tabuliek nasledovne:

```
<?
$subor = file_get_contents('slovník.txt');
$lang = unserialize($subor);
?>
```

Teda reverzným spôsobom sa načíta súbor a späťne sa z neho dostane pole funkciou unserialize. Potom jednoducho volanie poľa s indexom vracia popisnú hodnotu. Keď už mám pripravenú tabuľku popisov, môžem spracovávať ďalšie. Napríklad dátový súbor 100 obsahuje výrobcov aut v nasledovnom formáte:

```
999910000106SKODA      1000001061000100
```

V manuáli som sa dozvedel nasledovné údaje:

Meno parametru	Pozícia	Dĺžka	Typ	Popis
DLNr	000	4	N	Konštanta dodávateľa databáze = 9999
SA	004	3	N	Konštanta dátovej tabuľky = 100
HerNr	007	5	N	Číslo výrobcu
HKZ	012	10	C	Krátky kód výrobcu
LBezNr	022	9	N	Popisné číslo (->012)
PKW	031	1	N	PC výrobca (0 = No, 1 = Yes)
NKW	032	1	N	CV Výrobca (0 = No, 1 = Yes)
VGL	033	1	N	Výrobca na porovnanie (0 = No, 1 = Yes)
Achse	034	1	N	Výrobca osí (0 = No, 1 = Yes)
Motor	035	1	N	Výrobca motorov (0 = No, 1 = Yes)
Getriebe	036	1	N	Výrobca prevodoviek (0 = No, 1 = Yes)
Delete	037	1	N	1: Výrobca bude zmazaný v budúcnosti, inak 0

Tab. 1 Popis dát tabuľky výrobcov

Pre našu potrebu sú potrebné parametre číslo výrobcu, popis výrobcu – meno značky. Tieto parametre viem vyššie pripraveným skriptom obdobne spracovať tak, aby mi vzniklo pole žiadaných hodnôt, ktoré viem uložiť do databáze, konkrétne do tabuľky brands.

Po naimportovaní jednotlivých značiek som spustil import sérií, ktoré sa nachádzajú v súbore 110. Tu už dochádza k vytváraniu relácie pomocou cudzích kľúčov, nakoľko v tabuľke types sa pomocou hodnoty v stĺpci brand odkazujem na už naimportovanú značku. Musel som si zistiť najskôr ID značky z databáze pomocou kódu výrobcu, teda SQL príkazom

```
Select id from brands where code = '0004'
```

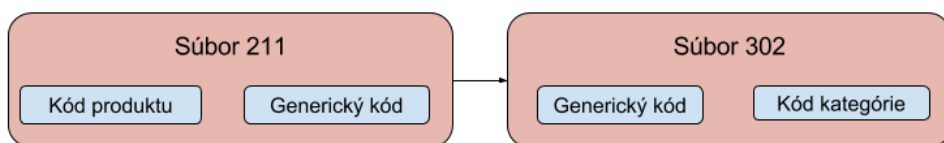
kde premenná brandcode je kód značky. Do tabuľky types teda potrebujem meno typu, získané z popisnej tabuľky pomocou kódu, kód typu a id značky. Pokračoval som importom tabuľky typov podľa motorizácie, ktorá prebiehala obdobne ako import sérií. Motorizácia sa nachádzala v súbore 120, pričom odkazovala sa cudzím kľúčom odkazuje na sériu. Pomocou kódu motorizácie som spracovaním dátového súboru 125 a následne 150 priradil typy motorov jednotlivým motorizáciám obdobne ako v prípadoch vyššie, pričom som spracoval názov motoru jeho kód a priradenie k motorizácií.

Potom som importoval strom kategórií dodávaných TecDocom. Tie sa nachádzajú v súbore 301. Je to stromová štruktúra pričom som spracoval meno kategórie – odkaz z dátového súboru 030, tecdocid (vlastné číslo kategórie), poradie. Po importe bez štruktúry som import musel spustiť znovu, aby sa pomocou tecdocid nadradenej položky vyhládalo ID v databáze obdobne ako pri značkách vyššie a zapísalo sa ako ID nadradenej položky. Väzba obrázkov kategórií

sa nachádza v súbore 307 a sú previazané podľa tecdocid s produktom a číslom ikonky zo súboru 305.

Pre položky bolo ešte nutné pripraviť zoznam jednotlivých parametrov do tabuľky parameters. Tie sa nachádzajú v súbore 50 a názvy sú prepojené podobne ako v ostatných prípadoch v súbore 30 a kód, ktorým sa budú parametre párovať pri importovaní položiek.

Nasledovalo importovanie jednotlivých položiek. Tie sú rozdelené podľa jednotlivých výrobcov, v testovacom prípade máme položky výrobcu MANN. Najskôr som si spracoval hlavičkový súbor 001, kde sa nachádza meno a kód výrobcu, ktorý som uložil do tabuľky manufacturers. Následne všetky položky majú unikátny kód, ktorý sa nachádza v súbore 200 a na základe ktorého sa budú párovať popisy, obrázky a ostatné parametre. Súbor ešte obsahuje kód ktorý obsahuje odkaz na názov v súbore 30, ktorý som si pripravil podobne ako súbor 012 vyššie a nazval si ho slovník2.txt. Prípadný popis produktu som získal zo súboru 206, ktorý je cez kód prepojený na súbor 320 a následne na 323 a na slovník2.txt. Pre prepojenie s kategóriami najskôr v súbore 211 som našiel takzvaný generický kód, ktorým som vyhľadával množinu kódov tecdocid pre kategórie v súbore 302. Preto som si súbor 302 pripravil do poľa. Pre zjednodušenie som nakreslil nasledujúcu schému.



Obr. 22 Prepojenie produktov a kategórií v dátových súboroch

Pomocou kódu produktu a kategórie som si našiel príslušné ID a tie uložil do tabuľky item_category.

Potom som na importoval jednotlivé parametre, ktoré sú v súbore 210 a obsahujú hodnotu a kód parametru podľa ktorého nájdem ID podobne ako v predchádzajúcich prípadoch.

Položkám som potom priradil jednotlivé kódy. Na kódy som použil pomocnú tabuľku codetypes, kde som uložil názvy jednotlivých typov kódov – OE (kód výrobcu), predajné kódy a EAN kódy. OE kódy od výrobcov sú v súbore 203, ktorý obsahuje aj odkaz na výrobcu čiže názov si viem získať SQL volaním na tabuľku výrobcov. Predajné kódy sú v súbore 207, EAN 209, import je obdobný ako pri OE s rozdielom ID typu kódu.

Prepojenia na jednotlivé médiá ako napríklad obrázky alebo PDF súbory sa sú v súbore 232 s odkazom na tabuľku 231 sa nachádza meno súboru.

Produkty majú aj svoje náhrady, ktoré som získal zo súboru 204, kde je vždy kód produktu a kód náhrady. Obdobne som si získal ID oboch a uložil do tabuľky `item_replacement`.

Na záver bolo potrebné prepojiť jednotlivé súčiastky a automobily. Prepojenia sa nachádzajú v súbore 400, kde sú prepojené jednotlivé položky s kódmi motorizácie. Podobne ako pri kategóriách sa ID produktu a ID motorizácie uloží do tabuľky `enginetype_item`. Je zrejmé, že jedna súčiastka je kompatibilná s viacerými typmi automobilov.

5.2.2 Kódovanie frontendu

Pri prepnutí do katalógu TecDoc sa v hlavičke aktivuje vyhľadávanie a pomocná lišta sa zmení na kaskádový výber aut a menu aut vyhľadávaných v minulosti. Zákazník má dve možnosti nájdania tovaru – výberom auta alebo priamym vyhľadávaním pomocou kódu. V prípade výberu auta si v roletovom menu vyberie značku, prostredníctvom AJAXu sa načítajú typy aut a po výbere typu auta sa znovu AJAXom načíta motorizácia. V prípade zmeny nadradeného parametru (značka, typ) sa podradené parametre vynulujú. Po výbere značky vľavo sa nachádza strom kategórií, kde si vyberie zákazník kategóriu produktu z dvojúrovňového menu.

V prípade vyhľadávania sa hľadajú produkty odpovedajúce zadanému kódu. V tomto prípade sa menu kategórií vľavo nezobrazuje a zobrazuje sa priamo len výpis výsledkov vyhľadávania.

Výpis výsledkov vyhľadávania je zoskupený podľa značiek a preferencií zadaných v administrácii. Na stránku sa maximálne zobrazuje 50 položiek, v prípade vyššieho množstva sa pod výpisom produktov zobrazí stránkovanie. Výpis sa dá filtrovať podľa značiek, kedy sa pomocou formuláru odošle množina ID značiek, ktoré chceme zobraziť. Filter funguje v oboch prípadoch vyhľadávania.

Každá položka obsahuje meno, TecDoc kód (hlavný kód produktu), meno produktu, obrázok (ak existuje, inak sa zobrazí ikona neexistujúceho obrázku), cenu, ikonu na zobrazenie detailu, stavy skladov a ikonku vloženia do košíku. Po kliknutí na ikonu oka sa prostredníctvom Ajaxu načíta detail, ktorý je zobrazený prostredníctvom záložiek. Jednotlivé záložky obsahujú náhrady, detailný popis (vlastnosti), možnosti použitia, ostatné identifikačné kódy a zvyšné mediálne súbory. Funkcionalitu záložiek obsahuje framework Bootstrap už v sebe a nie je nutné importovanie cudzieho riešenia.

5.3 Nasadenie na server

Systém sa nasadil pre interné testovanie zamestnancov spoločnosti. Centrálny server tvoril virtuálny server od spoločnosti Forpsi, ktorý sa nastavil pomocou vzdialeného prístupu. Ako serverový operačný systém sa nainštaloval Ubuntu Server 16.04 LTS 64bit. Následne sa nainštaloval takzvaný LEMP balíček, čo je skratka pre Linux, (E)Nginx, Maria (MySQL) a PHP. Nakoľko sa jednotlivé časti systému inštalujú postupne, je nutné ich nakonfigurovať tak, aby fungovali spolu.

V konfiguračnom súbore Nginxu treba spustiť spracovanie PHP súborov a zakázať spracovanie .htaccess súborov, nakoľko ich server nepodporuje. To sa dosiahlo nasledujúcimi nastaveniami:

```
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/run/php/php7.0-fpm.sock;
}

location ~ /\.ht {
    deny all;
}
```

Následne bolo treba zabezpečiť prepisovanie URL adries vo frameworku Codeigniter a to pridaním nasledujúcich riadkov:

```
location / {
    rewrite ^system.* /index.php?/$1 break;
    rewrite ^application.* /index.php?/$1 break;
    if (!-e $request_filename){
        rewrite ^(.*)$ /index.php?/$1 break;
    }
}
```

Kde je zabezpečený aj prístup používateľom zvonku na systémový priečinok system a application a umožnený prepis URL adries do prívetivejšej podoby používateľom. Následne som nahral do priečinku /var/www/html (základný priečinok pre webové skripty) celú centrálnu časť, teda framework + aplikáciu. Pre testovacie potreby som spustil na rovnakom serveri aj MariaDB databázu, kde som štruktúru tabuliek a základne dáta nahral pomocou nástroju Adminer.

Pre databázu TecDoc som vytvoril separátne klon serveru s tým rozdielom, že na sa nachádza len databáza a rozbalené mediálne súbory. Obrázky sú rozdelené podľa výrobcov do jednotlivých archívov ktoré stačilo len rozbaľiť a premenovať tak, aby sa priečinok nazýval TecDoc číslom výrobcu (0001...). Nakoľko obrázky sa nemenia a sú zväčša statické, nastavil som ich spracovanie do vyrovnávacej pamäte pre zvýšenie rýchlosti serveru. Stačilo pridať jednoduché nastavenie do konfigurácie Nginx-u:

```
location~* \.(?:jpg|jpeg|gif|png|pdf|doc|docx|xls|xlsx)$ {
    expires max;
    access_log off;
    add_header Cache-Control "public";
}
```

Kde som nastavil maximálnu možnú dobu držania súborov vo vyrovnávacej pamäti. V budúcnosti pri importe novej verzie databáze sa táto vyrovnávacia

pamäť manuálne premaže. Taktiež som nahral celú TecDoc databázu pomocou nástroja Adminer a na pôvodnom serveri nastavil pripojenie k tejto databáze v konfigurácií. Serveri boli tým pádom pripravené na interné testovanie zamestnancami.

5.4 Prepojenie so skladovým systémom

Prepojenie so skladovým systémom sa vykonáva na základe výmeny XML súborov medzi systémom a skladovým systémom. Zo skladového systému e-commerce systém importuje nasledujúce položky:

- Stav skladov
- Údaje zákazníkov
- Ceny položiek
- Priorita značiek
- Položky na vrátenie
- Ceny pre cenové skupiny

Opačným smerom sa odosielajú na sklad jednotlivé objednávky. V dobe písania tejto práce bol vo vývoji modul na prijímanie externých objednávok treťou stranou – firmou Nextis. Spracovanie odoslaných stavov skladov a ostatných vecí bolo jednoduché, stačilo využiť v php funkciu `simplexml_load_file`, ktorá načítala súbory odoslané na FTP v centrálnom hostingu. Táto funkcia premenila dáta v XML tagoch na štruktúrovanú inštanciu objektu `SimpleXMLElement`, s ktorou sa dalo ďalej pracovať. V prípade stavov skladov to bolo párovanie pomocou kódu položky jednoduchým SQL príkazom:

```
Update stocks, set value = '1' where code = '23 091 32 101' and  
warehouse = '1'
```

Trochu komplikovanejšie bolo pridávanie používateľov nakoľko najskôr pomocou funkcie `identity_check()` v knižnici `Ion auth`, som musel zistiť prítomnosť používateľa v systéme a podľa toho sa rozhodnúť či ho aktualizovať alebo vytvoriť nové konto. Pri vytváraní konta sa vytvára aj nové heslo používateľovi, inak sa iba aktualizujú dodacie a fakturačné údaje. Po spracovaní používateľov skript pristúpi k spracovaniu cien pre jednotlivých používateľov, ktoré musia byť spracované po nahraní/aktualizácii používateľov, nakoľko sa v tabuľke cien pre používateľov nachádza ID používateľa a tým pádom by mohlo dôjsť k nekompatibilite dát kvôli neúplnosti tabuľky `users`. Následne sa môžu spracovať všetky nasledujúce dáta. Skript na importovanie dát bude spúšťaný pravidelne podľa potreby, minimálne však dvakrát denne pomocou služby CRON na linuxovom serveri. Táto služba zabezpečí periodické spúšťanie skriptu v predvolený čas.

5.5 Testovanie systému

Systém pred uvedením do produkčného prostredia je nutné kvalitne otestovať a prípadné chyby odstrániť. Hlavná časť testovania je nákupný proces popísaný v kapitole návrhu riešenia systému. Pre detailnejšie pochopenie rozpišem potrebnú funkcionálnu jednotlivých podstránok systému.

5.5.1 Titulná stránka a jadro stránky

Na titulnej stránke by mal byť funkčný hlavný banner s reklamami a pás so značkami. V hornej časti by malo byť funkčný odkaz na prihlásenie, zabudnuté heslo a kontakt. V pätičke stránky by sa mali nachádzať správne jednotlivé adresy na pobočky spoločnosti.

5.5.2 Prihlásenie, zabudnuté heslo a profil používateľa

Používateľ by sa mal v prípade zadania správneho mena a hesla prihlásiť na stránku, taktiež by sa mala prepnúť hlavička stránky, sprístupniť jednotlivé katalógy aktivovať košík a vyhľadávanie na jednotlivých katalógoch. V prípade neúspešného pokusu o prihlásenie tuto skutočnosť oznámi hlásenie nad prihlasovacím formulárom a v prípade viac ako 6 neúspešných pokusov dôjde k zablokovaniu účtu a nutnosti kontaktovať pracovníka spoločnosti. Zabudnuté heslo sa bude dať obnoviť pomocou odkazu, ktorý po vygenerovaní príde na kontaktný email zákazníka.

V profile používateľa bude možné zmeniť si heslo k účtu. Nakoľko dodacie a fakturačné adresy spravuje skladový systém, nie je ich možné cez profil meniť.

5.5.3 Katalóg TecDoc

Bude možné funkčné vyhľadávanie dielu, alebo prezerania stromu dielov prostredníctvom výberu značky, typu a motorizácie automobilu. Pri vyhľadávaní sa priamo zobrazuje výpis dielov bez stromu kategórií, v prípade výberu značky sa strom kategórií zobrazí po výbere motorizácie auta, ktorá sa zakaždým načíta automaticky.

Vybraný diel bude možné vložiť do košíku, prípadne roztvoriť si jeho detaily a vybrať náhradu dielu a tú vložiť do košíku.

5.5.4 Nákupný košík a dokončenie objednávky

V nákupnom košíku by som mal mať možnosť meniť počet kusov jednotlivým položkám a priamo by sa mala prepočítavať konečná cena položky a celej objednávky. Taktiež pri odstránení položky z nákupného košíku sa táto cena upraví. V prípade nahratia txt alebo csv súboru v dodržanom formáte, sa vyhľadajú jednotlivé položky a v prípade zhody sa pridajú do košíku. V prípade nenulového stavu košíku sa dá prejsť k objednávke.

V objednávke bude možné vybrať miesto doručenia (rozvozová trasa) a dátum doručenia. Objednávku bude možné odoslať až potvrdením súhlasu

s obchodnými podmienkami spoločnosti. Pri potvrdení objednávky je možné v modálnom okne vybrať osobný odber alebo poobedný rozvoz.

5.5.5 Prenos dát do skladového systému

Otestovať prepojenie so skladovým systémom, hlavne sťahovanie objednávok a prenos cien a stavov skladov. Pokusne vytvoriť používateľa v skladovom systéme a zosynchronizovať ho s e-commerce systémom.

5.5.6 Zhodnotenie

Aplikácia bola primárne testovaná obchodnými zástupcami spoločnosti a internými zamestnancami. Po odstránení drobných nedostatkov a doplnení nových katalógov sa plánuje spustenie do produkčného prostredia.

V budúcnosti sa plánuje testovanie pomocou automatizovaných PHP Unit testov a pomocou riešenia Codeception na komplexné testovanie celej aplikácie.

6 Diskusia

Systém spĺňa všetky požiadavky spoločnosti formulované v kapitole návrhu riešenia systému. V dobe písania práce prebiehalo interné testovanie systému spoločnosťou, pričom bolo zadané pridať do systému ďalšie štyri katalógy, ktoré nie sú až tak obsiahle ako TecDoc, ale dátovo a štruktúrou úplne odlišné. Preto bolo v práci namieste rozdeliť aplikáciu na centrálnu časť s košíkom a profilmi používateľa a jednotlivé databázy katalógov. Taktiež jednotlivé katalógy je možné škálovať, čo bude nutné sledovať a následne spracovať po nasadení do produkcie a reálnej prevádzky systému, nakoľko na systém vieme spustiť reálne iba automatické záťažové testy.

Na zistenie vyťaženia systému by bolo vhodné nainštalovať na virtuálne servery nejaký monitorovací systém záťaže, či už riešenie Nagios, Cacti alebo Zabbix, ktoré nám bude vedieť poskytnúť informácie o výkonovo slabých miestach na platforme. Tieto miesta by sa ošetrili buď replikáciou serverov, alebo auditom pomalých miest v aplikáciách a ich odstránením.

Tu nastáva otázka do budúcnosti, či jednotlivé zložky systému nebude potrebné ešte viac oddeliť na samostatnú zobrazovaciu časť a časť kontrolera. To by sa dá dosiahnuť vytvorením centrálnej API vrstvy, ktorá by fungovala len ako jednoduché odosielanie textových dát v nejakom štandardizovanom formáte ako XML alebo JSON. Samotné vykresľovanie týchto elementov by mal na starosti nejaký JavaScriptový framework ako napríklad ReactJS. Narozdiel od objektov vykresľovaných prehliadačom, objekty vykresľované prostredníctvom ReactJS sú jednoduchšie na vytvorenie a aktualizáciu (Facebook Inc, 2017). Týmto spôsobom by sa odľahčila centrálna časť, nakoľko vykresľovanie celej stránky by prebiehalo u používateľa v jeho prehliadači a server by posielal len jednoduché štruktúrované textové dáta a obrázky. Navzdory všetkému API vrstva sa určite bude pripravovať do budúcnosti, nakoľko je požiadavka odberateľov na priame prepojenie ich informačných systémov a objednávok, bez nutnosti zásahu respektíve manuálneho nákupu ich pracovníka.

API vrstva tiež bude využiteľná pre programovanie mobilných aplikácií pre IOS a Android. Do budúca zamýšľa spoločnosť vybaviť obchodných zástupcov prevažne tabletmi, ktoré sú mobilnejšie a uľahčovali by obchodným zástupcom priamo nákup a jednanie o odberateľských cenách. V prípade, že sa firma rozhodne pre vývoj vlastného skladového programu alebo priamo CRM informačného systému, ten sa bude dať jednoducho cez API prepojiť s webovým riešením.

Nakoľko katalóg TecDoc obsahuje aj jazykové mutácie, spoločnosť premýšľa nad prekladom celého riešenia. Zamýšľané sú angličtina, čeština a poľština. Samotný framework podporuje zmenu jazykov, takisto má nato prispôbené rozšírenie, ktoré dokáže zobrazovať preklady jednotlivých jazykových reťazcov, ktoré sa zobrazujú na frontende, čiže implementácia by z pohľadu frameworku nemala robiť problém. Určite by sa museli rozšíriť databázy jednotlivých katalógov o nové tabuľky, kde by sa jednotlivé reťazce ukladali. Možným riešením je aj

vytvorenie tabuľky iba s reťazcami, ktorá by slúžila ako slovník, podobne ako to majú riešené dátové súbory TecDocu.

Po nasadení na do produkcie by bolo vhodné vykonanie bezpečnostného auditu s cieľom zabezpečenia aplikácie proti prípadným útokom. Základ by tvorilo nasadenie SSL certifikátu, ktorý zašifruje prenos medzi klientom a serverom, hlavne čo sa týka citlivých údajov ako sú napríklad heslá. Ďalej by bolo vhodné skontrolovať bezpečnosť formulárov na stránke a to hlavne prihlasovanie do administrácie, klientskemu účtu a formuláre v nákupnom procese. Samotný framework ma už integrovanú ochranu proti bežným XSS a CSRF útokom, avšak môžu sa vyskytovať iné, menej bežné útoky tohto typu proti ktorým by bolo vhodné sa zabezpečiť.

So zabezpečením priamo súvisí aj zálohovanie a to hlavne hlavnej databáze s objednávkami, nakoľko tá sa dynamicky mení. Nakoľko spoločnosť Forpsi poskytuje zálohy pomocou obrazu diskov, čo by bolo vhodné pri zrútení celého systému, pri priamej zálohe databázy bolo by vhodné použiť aj nejaký zálohovací program, buď unixový mysqldump a plánovať to CRONom prípadne iný nástroj ktorý priamo zálohu databáze odošle na predom stanovené miesto pomocou FTP alebo nejakej synchronizácie do cloudu napríklad prostredníctvom služby Dropbox.

V prípade nedostatočného výkonu PHP aplikácie môže v budúcnosti nastať situácia, kedy bude nutné celú aplikáciu prepísať do iného skriptovacieho jazyku. Do úvahy pripadá buď jazyk Python alebo aplikácia bežiaca cez NodeJS v JavaScripte. Nakoľko sa výkon PHP zlepšil prechodom z verzie 5.6 na verziu 7 až desaťnásobne, nepredpokladám že by tento prípad nastal. Nakoľko je platforma dostatočne oddelená, vedia sa jednotlivé časti replikovať tak, aby bol výkon stále dostatočný.

Do aplikácie by bolo vhodné doprogramovať zaznamenávanie a vyhodnocovanie jednotlivých akcií prostredníctvom štandardných logov, ktoré bude môcť nejaký systém spracovávať. Vhodná použítie na zaznamenávanie logov by bola zabudovaná knižnica vo frameworku Error Handling, prípadne univerzálna knižnica Logging. Na spracovanie logov do prehľadných grafov s cieľom odhaliť slabé alebo problematické miesta existuje nástroj GoAccess.

7 Záver

V spolupráci s firmou Hart Slovakia, vedúcim diplomovej práce a nezávislými konzultantmi sa mi podarilo navrhnuť, implementovať a spustiť riešenie, ktoré splňuje požiadavky rýchlosti, bezpečnosti a prehľadnosti systému. Rieši ich najpálčivejšie problémy z minulého riešenia, pričom zachováva niektoré obľúbené funkcie. Systém sa naďalej vyvíja a pridávajú sa nové katalógy, ktoré rozširujú sortiment súčasných produktov. Spoločnosť do budúcnosti uvažuje o maloobchodnom predaji, pričom maloobchod by bol založený na pozmenenej kópii spracovaného riešenia, ale táto aktivita je stále v štádiu úvah. Naďalej sa rieši prepojenie so skladovým systémom, nakoľko niektoré položky sa zdajú byť náročné na prenos a hľadá sa nové rýchlejšie riešenie. Navrhnutý systém je vhodný nielen na nasadenie pre automobilový segment, ale jeho nasadenie je vhodné aj v iných segmentoch, kde pri vysokom počte produktov je nutné zachovanie výkonu aplikácie. Systém má výhodu v centrálnej časti, na ktorú je možné napojenie ľubovoľného počtu a druhov katalógov, taktiež nakoľko sa jedná o vlastné riešenie, napojenie ľubovoľných knižníc tretích strán pre podporu napríklad platby alebo správy skladov je jednoduché. Riešenie využíva potenciál najmodernejších technológií dostupných v dobe písania tejto práce a všetky časti systému majú otvorený zdrojový kód. Verím, že navrhnutý a implementovaný e-commerce systém bude spoločnosti dlho slúžiť a bude s ním spokojná. Naša spolupráca pokračuje a verím v dlhodobú spoluprácu aj v budúcnosti.

8 Literatúra

- KOSEK, J. *PHP a XML*. 1. vyd. Praha: Grada, 2009. 367 s. ISBN 978-80-247-1116-4.
- HOGAN, B P. *HTML5 a CSS3 : výukový kurz webového vývojáře*. 1. vyd. Brno: Computer Press, 2011. 272 s. ISBN 978-80-251-3576-1.
- GASTON, P. *Moderní web*. Brno: Computer Press, 2015. ISBN 978-80-251-4345-2.
- MIKULÁŠKOVÁ, P. -- SEDLÁK, M. *Jak vytvořit úspěšný a výdělečný internetový obchod*. Brno: Computer Press, 2015. ISBN 978-80-251-4383-4.
- SOMMERVILLE, I. *Softwarové inženýrství*. 1. vyd. Brno: Computer Press, 2013. 680 s. ISBN 978-80-251-3826-7.
- NEDELCEU, CLÉMENT. *Nginx HTTP Server*. 3. vyd. Birmingham: Packt Publishing, 2015. ISBN 978-1-78528-033-7.
- ORR, ELI A YEHUDA ZADIK. *Programming with CodeIgniter MVC*. Birmingham: Packt Publishing, 2013. ISBN 978-1-84969-470-4.
- MANIAN, DIVYA. *HTML5 Boilerplate Web Development*. Birmingham: Packt Publishing, 2012. ISBN 978-1-84951-850-5.
- MICROSOFT. *Windows 2000 Clustering Technologies: Cluster Service Architecture* [online]. 2017 [cit. 2017-03-10] Dostupné z: <https://msdn.microsoft.com/en-us/library/bb727117.aspx>
- AWARDS-BA³. *SaaS: (Software as a Service) Solution. Brentech Data Systems Ltd*. [online]. [cit. 2017-04-01]. Dostupné z: <http://www.bds.ie/solution/awards-ba%C2%B3-saas-software-as-a-service-solution.406.html>
- HILDTICH, DAVID. *Scaling WooCommerce to 1 million products*. [online]. 2016 [cit. 2017-04-02]. Dostupné z: <https://www.wpintense.com/2016/08/16/scaling-woocommerce-1-million-products-talk-wordcamp-brighton/>
- HENDERSON, NICOLE. *Magento, WooCommerce Lead Ecommerce Platform Market Share: Report*. Whir [online]. , 1 [cit. 2017-04-02]. Dostupné z: <http://www.thewhir.com/web-hosting-news/magento-woocommerce-lead-e-commerce-platform-market-share-report>
- WINKLE, PHIL. *Is Magento the right platform for 1M products?* [online]. [cit. 2017-04-02]. Dostupné z: <http://magento.stackexchange.com/questions/3652/is-magento-the-right-platform-for-1m-products?answertab=votes#tab-top>
- DREAMHOST.COM. *Request per second*. [online]. DreamHost: DreamHost, 2017 [cit. 2017-04-10]. Dostupné z: <https://help.dreamhost.com/hc/en-us/articles/215945987-Web-server-performance-comparison>
- W3TECHS. *Usage of server-side programming languages for websites*. [online]. W3Techs: W3Techs, 2017 [cit. 2017-04-10]. Dostupné z: https://w3techs.com/technologies/overview/programming_language/all

- SERAVO. *10 reasons to migrate to MariaDB (if still using MySQL)*. [online]. Seravo: online, 2015 [cit. 2017-04-10]. Dostupné z: <https://seravo.fi/2015/10-reasons-to-migrate-to-mariadb-if-still-using-mysql>
- MARIADB. *Parallel Replication*. [online]. MariaDB: online, 2015 [cit. 2017-04-10]. Dostupné z: <https://mariadb.com/kb/en/mariadb/parallel-replication/>
- OTTE, LUKÁŠ. *Databázové systémy: Normalizace dat* [online]. 2013, , 21 [cit. 2017-04-13]. Dostupné z: http://projekty.fs.vsb.cz/463/edubase/VY_01_044/
- ALMSAEED ABDULLAH. *AdminLTE Control Panel Template* [online]. online: autor, 2017 [cit. 2017-04-15]. Dostupné z: <https://almsaeedstudio.com/>
- HELLER MARTIN. *REST and CRUD: the Impedance Mismatch* [online]. Developer World. InfoWorld, 2007 [cit. 2017-04-15]. Dostupné z: <http://www.infoworld.com/article/2640739/application-development/rest-and-crud--the-impedance-mismatch.html>
- ROUSE, MARGARET. *WYSIWYG (what you see is what you get)*. [online]. Techtarget: Techtarget, 2011 [cit. 2017-04-18]. Dostupné z: <http://whatis.techtarget.com/definition/WYSIWYG-what-you-see-is-what-you-get>
- EDMUNDS BEN. *Ion Auth Docs* [online]. online: autor, 2017 [cit. 2017-04-15]. Dostupné z: http://benedmunds.com/ion_auth/
- JAGLALE JÉRÔME. *Most Simple Template Library for CodeIgniter* [online]. online: autor, 2017 [cit. 2017-04-15]. Dostupné z: http://jeromejagla.com/doc/php/codeigniter_template
- SKOUMBOURDIS JOHN. *Grocery CRUD | Auto PHP Codeigniter CRUD* [online]. online: autor, 2017 [cit. 2017-04-16]. Dostupné z: <http://www.grocerycrud.com/>
- JARDINE ALLAN. *DataTables example - Server-side processing (5,000,000 rows)* [online]. online: autor, 2017 [cit. 2017-04-16]. Dostupné z: https://datatables.net/extensions/scroller/examples/initialisation/server-side_processing.html
- BAMBICO VINCENT A KOL. *Codeigniter-DataTables* [online]. online: autor, 2017 [cit. 2017-04-17]. Dostupné z: <https://github.com/zepernick/Codeigniter-DataTables>
- ALSUP MIKE A KOL. *Cycle2* [online]. online: autor, 2017 [cit. 2017-04-18]. Dostupné z: <http://jquery.malsup.com/cycle2/>
- PHP GROUP. *Serialize function* [online]. PHP Group, 2017 [cit. 2017-04-22]. Dostupné z: <http://php.net/manual/en/function.serialize.php>
- FACEBOOK INC. *Rendering elements - React* [online]. PHP Group, 2017 [cit. 2017-05-10]. Dostupné z: <https://facebook.github.io/react/docs/rendering-elements.html>

9 Zoznam obrázkov

Obr. 1	Myšlienková mapa funkcií návrhu nového e-commerce riešenia	16
Obr. 2	Use Case diagram funkcionality systému	17
Obr. 3	Schéma nákupného procesu používateľa	19
Obr. 4	Funkcionalita administrácie	20
Obr. 5	Rozdelenie platformy na nezávislé funkčné celky, ktoré je možné replikovať	22
Obr. 6	Počet požiadavkov za minútu, ktoré sú schopné spracovať jednotlivé webservery Zdroj: Dreamhost, 2017	23
Obr. 7	Použitie jednotlivých serverových skriptovacích jazykov v praxi Zdroj: W3Techs, 2017	24
Obr. 8	Porovnanie počtu požiadavok, ktoré sú jednotlivé frameworky schopné spracovať, za sekundu Zdroj: (Kenjis, 2017)	25
Obr. 9	Schéma centrálnej databázy navrhnutá v programe MySQL Workbench	28
Obr. 10	Schéma databázy TecDoc navrhnutá v programe MySQL Workbench	31
Obr. 11	Úvodná obrazovka systému	34
Obr. 12	Detailný náhľad do kategórie katalógu TecDoc	35
Obr. 13	Nákupný košík riešenia	36
Obr. 14	Výsledné okno rekapitulácie objednávky	37
Obr. 15	Demonštračná stránka šablóny administrácie AdminLTE 2 Zdroj: AdminLTE2, Almsaeed, 2017	38
Obr. 16	Zobrazenie hlavičky katalógu TecDoc na tablete Apple Ipad Air 2	42
Obr. 17	Potvrdenie objednávky	44
Obr. 18	Vzhľad správy používateľov cez rozšírenie DataTables	46

Obr. 19	Tabuľka so statickými stránkami vygenerovaná pomocou GroceryCRUD	47
Obr. 20	Úprava statickej stránky s WYSIWYG editorom	47
Obr. 21	Detail objednávky zákazníka v administrácii	48
Obr. 22	Prepojenie produktov a kategórií v dátových súboroch	52

10 Zoznam tabuliek

Tab. 1 Popis dát tabulky výrobcov

51

Prílohy

A Priložené CD so zdrojovými kódmi a ukážkovou databázou

Nastavenie a spustenie aplikácie:

Aplikácia pre svoj beh vyžaduje webový server so spusteným modulom pre PHP a databázu MariaDB (MySQL). Pre testovacie účely postačí server XAMPP, ktorý je voľne dostupný na adrese: <https://www.apachefriends.org/index.html> Tento server je nutné si stiahnuť, nainštalovať a spustiť server Apache a databázový server MariaDB.

Na CD v priečinku databáze sa nachádzajú dva súbory – hart.sql a tecdoc.sql. Tieto súbory je nutné na adrese <http://localhost/phpmyadmin> na importovať, pričom si samé vytvoria databáze, tabuľky a naplnia ich obsahom.

Samotná aplikácia sa nachádza v priečinku hart, ktorú je nutné nakopírovať do htdocs priečinku v serveri XAMPP - vznikne virtuálna adresa <http://localhost/hart> V aplikácii v súbore application/config/config.php je nutné nastaviť basepath na vyššie uvedenú adresu. V súbore application/config/database.php treba nastaviť pripojenie pre jednotlivé databázy (štandardne má XAMPP meno root a heslo prázdne).

Mediálne súbory pre Tecdoc sa nachádzajú v priečinku tecdoc, pričom ich treba podobne ako aplikáciu nakopírovať do priečinku htdocs. Nakoniec v súbore application/config/catalogues.php treba nastaviť cestu ku katalógu Tecdoc.

Prihlasovacie meno a heslo testovacieho používateľa:

Meno: test@test.com

Heslo: test1234

Adresa administrácie: <http://localhost/hart/admin>

Pre demonštračné účely je naimportovaných pár produktov od výrobcu MANN-Filter pod kódmi:

- 23 091 32 101
- 23 098 31 102