

Webové rozhraní pro skladový systém embedded laboratoře

Bc. Filip Šichman

Diplomová práce
2020



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Filip Šichman**
Osobní číslo: **A18417**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **Prezenční**
Téma práce: **Webové rozhraní pro skladový systém embedded laboratoře**
Téma práce anglicky: **A Web Interface for an Embedded Laboratory Storage System**

Zásady pro vypracování

1. Prostudujte stávající dostupné open-source systémy pro skladovou evidenci součástek.
2. Vhodným způsobem definujte požadavky na cílovou aplikaci.
3. Vyberte vhodné technologie pro výslednou implementaci.
4. Navrhnete architekturu aplikace a sestavení jednotlivých částí.
5. Implementujte jednotlivé součásti, zkompletujte vyvíjený systém a vytvořte testovací nasazení.

Rozsah diplomové práce:
Rozsah příloh:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. SKLAR, David. *PHP 7: praktický průvodce nejrozšířenějším skriptovacím jazykem pro web*. Přeložil Jan POKORNÝ. Brno: Zoner Press, 2018. Encyklopedie Zoner Press. ISBN 978-80-741-3363-3.
2. WELLING, Luke a Laura THOMSON. *Mistrovství PHP a MySQL*. Přeložil Ondřej BAŠE. Brno: Computer Press, 2017. ISBN 978-80-251-4892-1.
3. LAURENČÍK, Marek. *Tvorba www stránek v HTML a CSS*. Praha: Grada Publishing, 2019. Průvodce (Grada). ISBN 978-80-271-2241-7.
4. ŠULC, Vladimír. *Kybernetická bezpečnost*. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk, 2018. ISBN 978-80-738-0737-5.
5. MERIDA, O., BARNARD, E., KRUPINSKI, L., PIETROLUONGO, N., RAMSEY, B., STOCKLON, D., a kol. *Web Security*. Alexandria, VA: php[architect], 2016. ISBN 978-19-401-1141-4.
6. AHMAD, Waheed-ud-din Ahmad. *Building RESTful Web Services with PHP 7*. Birmingham: Packt Publishing, 2017. ISBN 978-17-871-2774-6.
7. STAUFFER, Matt. *Laravel: up and running: a framework for building modern PHP apps*. Sebastopol, CA: O'Reilly Media, 2016. ISBN 978-14-919-3608-5.

Vedoucí diplomové práce: **Ing. Peter Janků**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: 28. listopadu 2019
Termín odevzdání diplomové práce: 15. května 2020



doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 04.08.2020

Filip Šichman, v. r.,
podpis diplomanta

ABSTRAKT

Cieľom diplomovej práce je vytvorenie webového rozhrania pre skladový systém univerzitného embedded laboratória. Návrh webového rozhrania bude vychádzať z funkcionality systému Regent (desktop organizátor elektronických súčiastok), ktorý je v súčasnosti používaný na pracovisku univerzity za účelom vedenia skladovej evidencie.

Výsledný systém umožní používateľovi riadiť evidenciu elektronických súčiastok online. Vďaka responzívne dizajnu bude zobrazenie skladového systému prispôbené jednotlivým zariadeniam, na ktorých bude používaný. Systém umožní kooperáciu s nástrojmi pre návrh dosiek plošných spojov Altium Designer. Používateľ bude mať prístup k zoznamu evidovaných súčiastok filtrovaných na základe zvolených kritérií, spolu s náhľadom na ich detailné vlastnosti. Systém bude zabezpečovať komunikáciu s programovým rozhraním výrobcu elektronických súčiastok, čím umožní používateľovi importovať údaje súčiastky priamo do systému. Okrem funkcií vzťahujúcich sa na skladovú evidenciu systém poskytne funkcionality v súvislosti s používateľskými kontami ako autentifikácia, registrácia, obnova hesla, používateľské role, správa účtov a ďalšie.

Kľúčové slová: skladový systém, web, online, organizátor elektronických súčiastok

ABSTRACT

The aim of the diploma thesis is to create a web interface for the storage system of a university embedded laboratory. The design of the web interface will be based on the functionality of the Regent system (desktop organizer of electronic components), which is currently used at the university workplace for the purpose of keeping stock records.

The final system will allow the user to manage the register of electronic components online. Thanks to the responsive design, the display of the storage system will be adapted to the individual devices on which it will be used. The system will enable cooperation with the Altium Designer PCB design tool. The user will have access to a list of registered components filtered on the basis of selected criteria together with a preview of their detailed properties. The system will provide communication with the electronic component manufacturer's program interface, which allows the user to import part data directly into the system. In addition to functions related to inventory, the system will provide functionality related to user accounts such as authentication, registration, password recovery, user roles, account management and more.

Keywords: storage system, web, online, electronic components organizer

Touto cestou by som rád vyjadril vďaku svojej rodine za neprestajnú podporu a trpezlivosť. Zároveň by som sa chcel na tomto mieste poďakovať Ing. Petrovi Jankú za vedenie diplomovej práce.

Prehlasujem, že odovzdaná verzia diplomovej práce a verzia elektronická, nahraná do IS/STAG sú totožné.

OBSAH

ÚVOD	10
I TEORETICKÁ ČASŤ	11
1 OPEN SOURCE SYSTÉMY PRE SKLADOVÚ EVIDENCIU SÚČIASTOK	12
1.1 PARTS-IN-PLACE	12
1.1.1 Poskytovaná funkcionálnosť	12
1.1.2 Zhodnotenie nástroja Parts-in-Place	14
1.2 PARTSBOX	15
1.2.1 Poskytovaná funkcionálnosť	15
1.2.2 Zhodnotenie nástroja PartsBox	17
1.3 ORGANIZÁTOR ELEKTRONICKÝCH SÚČIASTOK REGENT	17
1.3.1 Autentifikácia používateľov	18
1.3.2 Zobrazenie katalógu elektronických súčiastok	18
1.3.3 Zobrazenie podrobných informácií o vybranej súčiastke a ich editácia	19
1.3.4 Pridanie novej súčiastky do systému.....	20
1.3.5 Náhľad na zoznam schém a footprintov.....	20
2 POŽIADAVKY NA CIEĽOVÚ APLIKÁCIU	22
2.1 ROZDELENIE POŽIADAVIEK	22
2.1.1 Rozdelenie požiadaviek v závislosti od úrovne popisu.....	23
2.2 DEFINÍCIA POŽIADAVIEK	24
2.2.1 Modelovanie prípadov použitia.....	24
3 BEZPEČNOSTNÉ RIZIKÁ	28
3.1 ÚTOK HRUBOU SILOU	28
3.2 RAINBOW TABLE ATTACK.....	28
3.3 SQL INJECTION	29
3.4 CROSS-SITE SCRIPTING	29
3.5 CROSS-SITE REQUEST FORGERY	29
4 VÝBER TECHNOLOGIÍ PRE IMPLEMENTÁCIU	31
4.1 SKRIPTOVACÍ JAZYK PHP.....	31
4.2 ÚVOD K FRAMEWORKOM JAZYKA PHP	31
4.2.1 Model-View-Controller.....	32
4.3 POROVNANIE PHP FRAMEWORKOV	33
4.3.1 Modularita a škálovanie	34
4.3.2 Šablónovacie nástroje.....	34
4.3.3 Databázová podpora.....	34
4.3.4 Zhrnutie	35
II PRAKTICKÁ ČASŤ	37

5	ŠPECIFIKÁCIA POŽIADAVIEK	38
5.1	NEFUNKČNÉ POŽIADAVKY	38
5.1.1	Bezpečnostné požiadavky	38
5.2	FUNKČNÉ POŽIADAVKY	39
5.3	DIAGRAM PRÍPADOV POUŽITIA	42
6	IMPLEMENTÁCIA.....	44
6.1	ARCHITEKTÚRA APLIKÁCIE	44
6.1.1	Vybrané technológie pre implementáciu	44
6.1.2	Model klient – server	45
6.1.3	Architektúra aplikácie Laravel	45
6.2	NÁVRH IMPLEMENTÁCIE	47
6.2.1	Zostavenie jednotlivých častí	47
6.2.2	Databázové pripojenie	47
6.3	METODIKA IMPLEMENTÁCIE RIEŠENIA	48
6.3.1	Konfigurácia databázového pripojenia	48
6.3.2	Prihlásenie	49
6.3.3	Používateľské role	49
6.3.4	Registrácia	49
6.3.5	Editácia profilu používateľa	50
6.3.6	Zobrazenie zoznamu súčiastok	50
6.3.7	Zobrazenie detailu elektronickej súčiastky	52
6.3.8	Zobrazenie zoznamu schém/footprintov	53
6.3.9	Zobrazenie detailu schémy/footprintu	54
6.3.10	Správa používateľských účtov	54
6.3.11	Import údajov súčiastky z API výrobcu Farnell	54
6.3.12	Správa dodávateľov elektronických súčiastok	55
6.3.13	Podpora zmeny jazyka	56
7	TESTOVACIE NASADENIE	57
	ZÁVER	65
	ZOZNAM POUŽITEJ LITERATÚRY	66
	ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK.....	68
	ZOZNAM OBRÁZKOV	69
	ZOZNAM TABULIEK	70
	ZOZNAM PRÍLOH.....	71

ÚVOD

Použitie systémov pre skladovú evidenciu v rámci výrobných procesov vedie k efektívnemu vyhodnocovaniu zásob. Nesprávna manipulácia nad skladosť môže viesť k prebytku alebo nedostatku zásob, čo môže znamenať stratu produktivity. Systémy pre skladovú evidenciu poskytujú kontrolu nad procesom riadenia zásob, pričom jednotlivé položky skladu môžu predstavovať malé predmety o veľkosti ihly až po zložité zostavy.

Na základe tohto podnetu je v rámci laboratória univerzity používaný k skladovej evidencii elektronických komponentov systém Regent. Systém je dostupný výlučne v desktop verzií. Počas využívania funkcií systému Regent vznikla požiadavka vytvoriť nástroj pre skladovú evidenciu, ktorý bude dostupný online prostredníctvom webového rozhrania. Výhody použitia webového rozhrania sú nasledovné: zaniká potreba inštalácie aplikácie na zariadenie používateľa, požiadavky sú spracovávané na strane servera, kde sú zároveň uchovávané dáta a preto je aplikácia prístupná odkiaľkoľvek. K používaniu webovej aplikácie je potrebný len webový prehliadač a prístup k internetu. Vďaka responzívnemu dizajnu je možné prispôbiť vzhľad aplikácie zariadeniam, na ktorých bude zobrazovaná.

Navrhovaná webová aplikácia bude vychádzať zo súčasného stavu skladovej evidencie - systém Regent. Aplikácia bude disponovať základnými vlastnosťami súčasného systému, jedná sa napríklad o kooperáciu s nástrojom Altium Designer alebo komunikáciu s programovým rozhraním výrobcu súčiastok Farnell. Funkcionalita bude prispôbená a rozšírená o prvky v závislosti od požiadaviek súčasných používateľov systému Regent. Kompletnej špecifikácii požiadaviek na systém je venovaná samostatná kapitola práce.

Samotná práca pozostáva z dvoch častí, časti teoretickej a praktickej. Teoretická časť predstavuje zhrnutie informácií a poznatkov využívaných v rámci praktickej časti. Nasledujúca praktická časť potom popisuje postup a samotnú implementáciu riešenia.

I. TEORETICKÁ ČASŤ

1 OPEN SOURCE SYSTÉMY PRE SKLADOVÚ EVIDENCIU SÚČIASTOK

Cieľom prieskumu bolo zdokumentovať systémy poskytujúce skladovú evidenciu prostredníctvom webového rozhrania. Väčšina dostupných open-source riešení online je poskytovaných so všeobecným zameraním - pre skladovú evidenciu továrenských výrobkov. Poskytujú teda možnosť hromadného uplatnenia v rámci rôznych odvetví.

V rámci systémov špecifikovaných na evidenciu elektronických súčiastok, ktoré poskytujú služby prostredníctvom webového rozhrania je možné uviesť systém *Parts-in-Place* a *PartBox*. Oba zmienené systémy poskytujú open-source riešenie pre riadenie skladových zásob a sú pripravené ihneď k použitiu. Ich obmedzenie spočíva v limitovanom počte používateľov, ktorí môžu k službe pristupovať.

Ďalšie open-source riešenie evidencie predstavuje systém *PartKeepr*. Nástroj *PartKeepr* umožňuje spravovať skladové zásoby, vyhľadávať v rámci databázy a poskytovať informácie k jednotlivým súčiastkam. Z pohľadu riadenia skladových zásob poskytuje *PartKeepr* rovnakú funkcionálnosť ako vyššie uvedené systémy. Rozdiel nastáva v použití, kedy je napred potrebná inštalácia a nasadenie (vyžaduje sa webový server a databáza).

1.1 Parts-in-Place

Parts-in-Place predstavuje online systém pre správu skladu elektronických súčiastok a rozpisu materiálu (angl. *Bill of Materials*) označovaného skráteno ako BOM. Tento systém je možné použiť pre riadenie skladových zásob a správu BOM akýchkoľvek fyzických komponentov, prednostne však pre zhotovovanie elektronických dosiek.[1]

1.1.1 Poskytovaná funkcionálnosť

Nasledujúci text popisuje časti (podstránky) systému *Parts-in-Place* z pohľadu poskytovanej funkcionality.

Bills of Materials

Parts-in-Place umožňuje viesť zoznam zostáv spolu s požadovanými dielmi k ich zhotoveniu. Hlavným účelom *Bills of Materials* (rozpisu materiálu) je identifikácia elektronických dielov potrebných pre vytvorenie zostavy.

Assembly Orders

Assembly Orders je časť, ktorá predstavuje centrálny mechanizmus pre plánovanie pripravovaných zostáv. V rámci tejto časti sú reprezentované nevybavené zostavy, pre ktoré je potrebné rezervovať alebo objednať elektronické súčiastky.[1]

Part Orders

Časť reprezentujúca objednávky. Program automaticky vypočíta množstvo dielov, ktoré je potrebné objednať pre nadchádzajúce projekty.

Parts Arrivals

Parts Arrivals slúži pre dokumentáciu dielov, ktoré pri dodaní vstupujú do systému. Účelom je udržiavať aktuálnu skladovú evidenciu.

Parts Transfers

Funkciou tejto časti je poskytnúť možnosť premiestňovania elektronických komponentov medzi skladom subjektu a zmluvným dodávateľom.

Assemblies

Podstránka umožňujúca vyradenie elektronických súčiastok, ktoré boli spotrebované v rámci realizácie projektu.

Inventory

Účelom tejto časti je poskytnúť informácie o elektronických súčiastkach, ktoré sú k dispozícii k určitému dátumu (Obr. 1). Z hľadiska umiestnenia je možné voliť len dve kategórie, fyzickú dostupnosť na sklade (*office*) alebo dostupnosť u zmluvného dodávateľa (*CM*), nejedná sa teda o fyzickú lokalizáciu dielov.

STATUS	INVENTORY DAT	LOCATION	COMPANY P/N	DESCRIPTION	MANUFACTURER	MANUFACTURER P/N	SUPPLIER	SUPPLIER P/N
Executed	2013-03-17	Office	E-CAP-001	Ceramic 10uF 6.3V X5R	Murata ERI	GRM188R60J106F	arrow	490-3896-2-ND
			E-CAP-002	0.01 uF 16V X7R	Murata ERI	GRM155R71C103	arrow	490-1313-1-ND
			E-CAP-003	0.1 uF 6.3V X5r	TDK	C1005X5R0J104K	digkey	445-1266-1-ND
			E-CAP-005	47pF 50V C0G	Yageo	CC0402JRNPO98F	digkey	311-1021-1-ND
			E-CAP-006	1.0 uF 6.3V X5R	Murata ERI	GRM188R60J105F	arrow	490-1550-1-ND
			E-CAP-007	2.2 uF 10V X7R	Murata ERI	GRM21BR71A225	arrow	490-1696-1-ND
			E-CAP-009	CAP CER 2200PF 50V 10% X7R 0402	Murata	GRM155R71H222	digkey	490-1305-2-ND
			E-CAP-010	CAP CER 18PF 50V 5% NPO 0402	TDK	C1005C0G1H180J	digkey	445-1238-2-ND
			E-CAP-011	Murata GRM1555C1H300J201D, Murata GRM1555C1H300J201D	Murata	GRM1555C1H300	digkey	490-1285-6-ND
			E-CAP-014	CAP CER 6PF 50V NPO 0402	TDK	C1005C0G1H060C	digkey	445-4884-2-ND
			E-CAP-015	CAP CER 2.2UF 10V 10% X7R 0603	Murata	GRM188R71A225F	digkey	490-4520-6-ND
			E-CAP-016	CAP CER 10PF 50V NPO 0402	Taiyo Yuden	UMK105CG100DV	digkey	587-1947-2-ND
			E-CAP-018	CAP CER 18PF 50V 5% NPO 0402	Yageo	CC0402JRNPO98B	future	311-1415-2-ND
			E-CAP-019	CAP CER 4.7UF 4V 20% X5R 0402	Taiyo Yuden	AMK105BJ475MV	digkey	587-1966-2-ND
			E-CAP-020	CAP CER 22UF 6.3V 20% X5R 0805	Yageo	CC0805MX5R5B	digkey	311-1464-2-ND
			E-CAP-022	CAP CER 0.033UF 16V 10% X7R 0402	Murata	GRM155R71C333	digkey	490-3257-2-ND

Obr. 1. Ukážka zobrazenia elektronických súčiastok skladom – systém *Parts-in-Place*

Parts Write-offs

Sekcia, ktorá slúži k zníženiu množstva zásob v rámci inventára, tento proces nemusí nevyhnutne súvisieť s procesom zostavovania komponentov. Odpis môže byť použitý pre zaznamenanie straty niektorých z dielov.

Parts Library

Časť systému slúžiaca ako adresár všetkých elektronických komponentov. *Parts Library* umožňuje zobraziť a ukladať detailné informácie o elektronických súčiastkach, ktoré sú v rámci jednotlivých projektov používané, respektíve boli použité. Detail súčiastky je možné vidieť na obrázku (Obr. 2).



The image shows a web-based form titled "Edit Parts Library" with a red header bar. The form contains the following fields and values:

Company P/N	E-CAP-001
Description	Ceramic 10uF 6.3V X5R
Identifier	10uF
Manufacturer	Murata ERI
Manufacturer P/N	GRM188R60J106ME47D
Notes	
Multiple	4000
Package ID	0603
Footprint	CAPC1608X95N
Price	0.0442
ROHS Compliant	<input type="checkbox"/>
Supplier	arrow
Supplier P/N	490-3896-2-ND
Storage	
Datasheet	

Obr. 2. Detail elektronickej súčiastky – systém *Parts-in-Place*

1.1.2 Zhodnotenie nástroja *Parts-in-Place*

Na základe vyššie uvedeného popisu je možné konštatovať, že systém *Parts-in-Place* sa vo svojej podstate zameriava predovšetkým na procesy prebiehajúce počas spracovania projektov. Z tohto pohľadu poskytuje radu užitočných funkcií, ktoré však majú pevne stanovenú réžiu. Z hľadiska vedenia skladovej evidencie súčiastok však neposkytuje žiadnu kategorizáciu súčiastok ani členenie. Fyzické umiestnenie je možné nastaviť len v sekcii *Parts Library*, pričom informácia o umiestnení nie je dostupná pri zobrazení skladu. Po

uplynutí skúšobnej verzie je možné bezplatne používať systém s obmedzením na jedného používateľa.

1.2 PartsBox

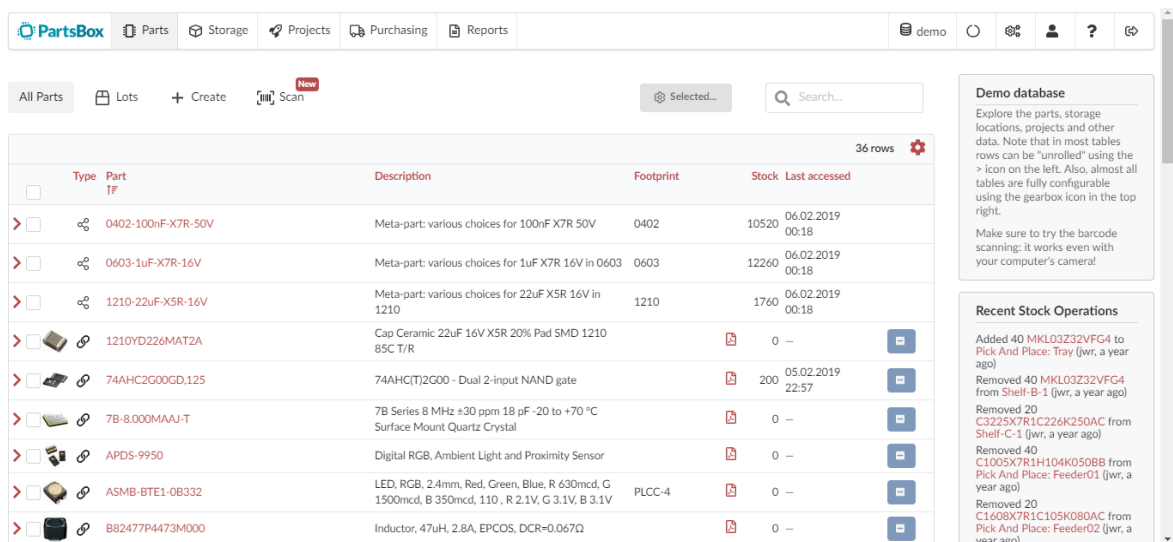
PartsBox je online nástroj pre správu elektronických komponentov. *PartBox* umožňuje jednoducho riadiť sklad elektronických súčiastok, sledovať kde sú jednotlivé súčiastky uložené, aké sú aktuálne zásoby a ktoré súčiastky sú používané v rámci projektov. *PartsBox* zobrazuje špecifikácie súčiastok a umožňuje k nim prikladať dokumenty.[2]

1.2.1 Poskytovaná funkcionálnosť

Rovnako ako v prípade systému *Parts-in-Place* obsahuje nasledujúci text popis jednotlivých častí systému *PartsBox*.

Parts

Podstránka zobrazujúca katalóg elektronických komponentov (Obr. 3). V zozname je možné dynamicky vyhľadávať. V sekcii *Parts* je možné registrovať nový komponent a to prostredníctvom výrobného čísla dodávateľa alebo manuálne. Ku každej elektronickej súčiastke má používateľ k dispozícii jej detail, ktorého ukážku je možné vidieť na obrázku (Obr. 4).



Type	Part	Description	Footprint	Stock	Last accessed
>	0402-100nF-X7R-50V	Meta-part: various choices for 100nF X7R 50V	0402	10520	06.02.2019 00:18
>	0603-1uF-X7R-16V	Meta-part: various choices for 1uF X7R 16V in 0603	0603	12260	06.02.2019 00:18
>	1210-22uF-X5R-16V	Meta-part: various choices for 22uF X5R 16V in 1210	1210	1760	06.02.2019 00:18
>	1210YD226MAT2A	Cap Ceramic 22uF 16V X5R 20% Pad SMD 1210 85C T/R		0	
>	74AHC2G00GD.125	74AHC(T)2G00 - Dual 2-input NAND gate		200	05.02.2019 22:57
>	7B-8.000MAAJ-T	7B Series 8 MHz ±30 ppm 18 pF -20 to +70 °C Surface Mount Quartz Crystal		0	
>	APDS-9950	Digital RGB, Ambient Light and Proximity Sensor		0	
>	ASMB-BTE1-0B332	LED, RGB, 2.4mm, Red, Green, Blue, R 630mcd, G 1500mcd, B 350mcd, 110 , R 2.1V, G 3.1V, B 3.1V	PLCC-4	0	
>	B82477P4473M000	Inductor, 47uH, 2.8A, EPCOS, DCR=0.067Ω		0	

Obr. 3. Katalóg elektronických komponentov – systém *PartsBox*

C1005X7R1H104K050BB
C Series 0402 0.1 uF 50 V ±10 % Tolerance X7R SMT Multilayer Ceramic Capacitor

Part Info
 Attachments
 Linked Info
 Part Storage
 Add Stock
 Remove Stock
 Move Stock
 Stock History
 Low Stock Levels
 Attrition Rate
 Offers
 Orders
 Other Operations

Manufacturer: TDK
MPN: C1005X7R1H104K050BB
Footprint: —
Where is it?
 Pick And Place: Feeder01 (9520), Shelf-A-2 (1000)

Datasheets:
 Online Datasheet (444kB)
Attachments: —

Specifications

Physical:	Electrical:	Other:
Case/Package	0402	Capacitance
Case/Package (SI)	1005	Capacitance Tolerance
Mounting Style	Surface Mount	Dielectric
Size-Length	1mm	Characteristic
Size-Thickness	0.5mm	Dielectric Material
Size-Width	0.5mm	Operating Temperature
		Voltage Rating (DC)

Regulatory & Commercial:
 Lead-Free Status: Lead Free
 Packaging: Cut Tape (CT)
 RoHS: Compliant

Notes: —
Tags: —
Auto-Tags: capacitor, SMD, 0402, X7R

Obr. 4. Detail elektronickej súčiastky – systém *PartsBox*

Storage

Storage predstavuje zoznam úložísk kde je možné komponenty skladovať. Organizácia úložiska je na výbere užívateľa. Môže sa jednať o dve úložiská reprezentujúce miestnosti ale i niekoľko menších predstavujúcich napríklad police. V rámci každého úložiska je možné zobrazit' zoznam komponentov, ktoré obsahuje.

Projects

Časť *Projects* zodpovedá súpisu materiálu (BOM) a predstavuje elektronické komponenty, ktoré sú potrebné pri vytvorení jednotlivých zostáv, zariadení. Túto časť je možné použiť pri zostavení skutočných zariadení alebo ako návrhový a cenový nástroj pri práci na nových návrhoch.

Purchasing

Purchasing poskytuje prehľadný zoznam objednávok a zároveň nákupných zoznamov. Nákupný zoznam kombinuje komponenty z jedného alebo viacerých projektov, ktoré majú byť realizované. Jednotlivé zoznamy z projektov sú následne spojené do jednotného zoznamu, ktorý je možné použiť pre vytvorenie objednávky.

Reports

Reports umožňuje získať prehľadné informácie týkajúce sa celkového vedenia. Ide o informácie ako: súhrnný počet komponentov skladom, počet skladových miest, počet projektov, celková hodnota komponentov, oznamy nízkych zásob a iné.

1.2.2 Zhodnotenie nástroja PartsBox

V porovnaní so systémom *Parts-in-Place* poskytuje systém *PartsBox* modernejšie používateľské rozhranie. Rovnako ako v prípade *Parts-in-Place* ponúka viacero funkcií súvisiacich s procesom spracovania projektov – BOM, objednávanie súčiastok. Z hľadiska skladovej evidencie však ponúka viac, ako napríklad detailnú lokalizáciu jednotlivých komponentov, kategóriu pre zobrazenie úložísk a iné. Jedná sa teda o nástroj, ktorý sa prioritne zameriava práve na vedení skladu. Systém *PartBox* poskytuje užívateľovi jednoduchú a prehľadnú navigáciu.

1.3 Organizátor elektronických súčiastok Regent

Regent je desktop systém, ktorý je aktuálne používaný v rámci univerzity za účelom evidencie skladových zásob elektronických súčiastok. Systém Regent bol vyvinutý na univerzite pre potreby používateľov embedded laboratória. Za jeho vznikom stála požiadavka na zefektívnenie riadenia skladových zásob pomocou systému, ktorý by poskytoval: informácie o umiestnení a počte elektronických súčiastok s možnosťou ich kategorizácie, umožňoval viesť záznamy obsahujúce popis jednotlivých súčiastok a zároveň kooperoval s nástrojom Altium Designer. Výsledný systém Regent, zapuzdruje širokú funkcionality, ktorej fragmenty sú zhrnuté v nasledujúcich bodoch.

Hlavné zložky poskytovanej funkcionality:

- Autentifikácia.
- Zobrazenie katalógu elektronických komponentov filtrovaných na základe kategorizácie.
- Zobrazenie podrobných informácií o vybranej elektronickej súčiastke a možnosť ich priamej editácie.
- Náhľad na zoznam schém a footprintov¹ používaných jednotlivými súčiastkami. Zobrazenie zoznamu elektronických súčiastok, ktoré využívajú zvolenú schému, respektíve zvolený footprint.
- Možnosť zaevidovať novú elektronickú súčiastku.
- Možnosť vyhľadávať v zobrazovaných zoznamoch (súčiastok, schém a footprintov).

¹ Popis fyzického rozloženia elektronického komponentu (pre pripojenie k doske plošných spojov)

- Komunikácia s programovým rozhraním výrobcu elektronických súčiastok Farnell.
- Kooperácia s nástrojom Altium Designer.

1.3.1 Autentifikácia používateľov

Autentifikácia používateľov je realizovaná prostredníctvom prihlasovacieho formuláru. Používateľ je vyzvaný k zadaniu vstupných údajov - užívateľské meno a heslo. V rámci formuláru má používateľ možnosť prostredníctvom tlačidla umiestneného vpravo od vstupného poľa otvoriť dialóg obsahujúci zoznam registrovaných používateľských účtov. Formulár umožňuje zmenu prihlasovacieho hesla, túto možnosť je potrebné označiť pred prihlásením.

Následne je používateľovi zobrazený dialóg pre nastavenie nového hesla. Po potvrdení formuláru je v prípade, že vstupné údaje boli zadané správne používateľovi poskytnutý náhľad na katalóg elektronických súčiastok. V opačnom prípade je používateľ oboznámený o neúspešnom prihlásení oznamom informujúcim, ktorý zo vstupných údajov bol zadaný nesprávne.

1.3.2 Zobrazenie katalógu elektronických súčiastok

Používateľ má k dispozícii zoznam elektronických súčiastok (Obr.5). Tento zoznam je možné filtrovať na základe zvolenej kategórie a hľadaného výrazu. Vyhľadávanie v rámci zoznamu elektronických súčiastok prebieha dynamicky, pri každom vstupe z klávesnice. Súčasťou zoznamu sú údaje prislúchajúce každej súčiastke: názov, popis, hodnota, aktuálny počet a umiestnenie. Zobrazovaný zoznam neobsahuje stránkovanie.

Název součástky	Popis	Hodnota	Aktuální počet	Umístění
1			5	Box 1/1/4
2			16	6/5/10
3			0	
4	AA-8.000MALV-T TXC - AA-8.000MALV-T - CRYSTAL, 8MHz, 8PF, 5 X 3.2MM	AA-8.000MALV-T	42	
5	ABL57M2-12.000MHZ-D-... ABRACON ABL57M2-12.000MHZ-D-ZY-T CRYSTAL, 12MHz, 18P...	12 MHz	0	
6	ABMM2-25.000MHZ-E2-T ABRACON - ABMM2-25.000MHZ-E2-T - CRYSTAL, 25M, 18PF CL, ...	25 MHz	16	Box 5/8/5
7	ABMM2-8.000MHZ-E2-T ABRACON - ABMM2-8.000MHZ-E2-T - CRYSTAL, 8M, 18PF CL, 6...	8 MHz	11	Box 5/8/5
8	ABM10-20.000MHZ-E20-T ABRACON - ABM10-20.000MHZ-E20-T - CRYSTAL, 20MHz, 10PF, ...	ABM10-20.000MHZ-E20-T		
9	ABM3-10.0000MHZ ABRACON - ABM3-10.0000MHZ-D2Y-T - CRYSTAL, 10MHz, 18PF, ...	ABM3-10.0000MHZ-D2Y-T	0	
10	ABM3-25.000MHZ-D2Y-T ABRACON - ABM3-25.000MHZ-D2Y-T - CRYSTAL, 25MHz, 18PF, ...	25.000MHz	0	
11	ABM3-8.000MHZ-D2Y-T ABRACON - ABM3-8.000MHZ-D2Y-T - CRYSTAL, 8MHz, 18PF, S...	ABM3-8.000MHZ-D2Y-T	10	Box 4/8/4
12	ABM8-12.000MHZ-B2 Resonator type: ceramicFrequency: 12MHz	12 MHz	5	Box 5/8/5
13	ABS06-107-32.768KHZ-T ABRACON - ABS06-107-32.768KHZ-T - CRYSTAL, 32.768KHZ, 4PF, ...	ABS06-107-32.768KHZ-T	0	
14	ABS06-32.768KHZ-T ABRACON - ABS06-32.768KHZ-T - CRYSTAL, 32.768KHZ, 12.5PF, ...	32.768 kHz	1	Box 5/9/1
15	ABS07-120-32.768KHZ-T ABRACON - ABS07-120-32.768KHZ-T - CRYSTAL, 32.768KHZ, 6PF, ...	ABS07-120-32.768KHZ-T		
16	ABS07-32.768KHZ-T ABRACON - ABS07-32.768KHZ-T - CRYSTAL, 32.768KHZ, 12.5PF, ...	ABS07-32.768KHZ-T		
17	ACPL-247-500E AVAGO TECHNOLOGIES - ACPL-247-500E - OPTOCOUPLER, QU...	ACPL-247-500E	0	
18	ACPL-247-560E BROADCOM LIMITED - ACPL-247-560E - OPTOCOUPLER, QUAD, ...	ACPL-247-560E	0	
19	ACS711KLCTR-12AB-T ALLEGRO MICROSYSTEMS - ACS711KLCTR-12AB-T - CURRENT S...	ACS711KLCTR-12AB-T	4	Box 2/2/1
20	ACS711KLCTR-25AB-T ALLEGRO MICROSYSTEMS - ACS711KLCTR-25AB-T - CURRENT S...	ACS711KLCTR-25AB-T	0	

Obr. 5. Zobrazenie zoznamu elektronických súčiastok – systém *Regent*

1.3.3 Zobrazenie podrobných informácií o vybranej súčiastke a ich editácia

System umožňuje zvolením súčiastky zo zoznamu zobrazíť používateľovi detail súčiastky v modálnom okne. V rámci čoho je používateľovi sprístupnená editácia všetkých zobrazovaných údajov. Medzi tie patrí aj výber z ponuky kategórií, footprintov, schém a dodávateľov, pričom v súčasnosti používateľ nemá možnosť vyhľadávania v ponuke. Používateľ môže importovať údaje o zobrazovanej súčiastke pomocou funkcie „Doplniť“. Údaje sú importované z API (Application Programming Interface; aplikačné programové rozhranie) výrobcu súčiastok Farnell, predtým je však potrebné správne vyplniť Kód dodávateľa. Detail ponúka možnosť odstrániť súčiastku, funkcia však momentálne nie je dostupná. Súčasná verzia aplikácie nezobrazuje náhľad na súčiastku a neobsahuje validáciu zadaných vstupov. Jednotlivé údaje o súčiastke je možné vidieť na obrázku (Obr.6).

The screenshot shows a window titled "Vlastnosti součástky" (Component Properties). It contains several sections:

- Název součástky:** ABM10-20.000MHZ-E20-T
- Umístění:** (empty)
- Popis:** ABRACON - ABM10-20.000MHZ-E20-T - CRYSTAL, 20MHZ, 10PF, 2.5 X 2MM
- Obrázek:** (empty)
- Kategorie:** Crystals
- Dodavatel:** Farnell
- Hodnota:** ABM10-20.000MHZ-E20-T
- Kód u dodavatele:** 2467767
- Odkaz k objednávce:** <http://cz.farnell.com/2467767?Ntt=2467767>
- Odkaz na datasheet:** farnell.com/datasheets/1749654.pdf
- Footprint:** None
- Schéma:** None
- Aktuální počet:** 0
- Minimální počet:** 0
- Rezervováno:** 0
- Other:** SZ: , RC:

Buttons include "Přidat" (Add), "Doplnit" (Add), "Open", "Download", "Zobrazit rezervace" (Show Reservations), "Smazat součástku" (Delete Component), "OK", and "Cancel". A red box highlights the "Doplnit" button.

Obr. 6. Modálne okno reprezentujúce detail súčiastky – systém *Regent*

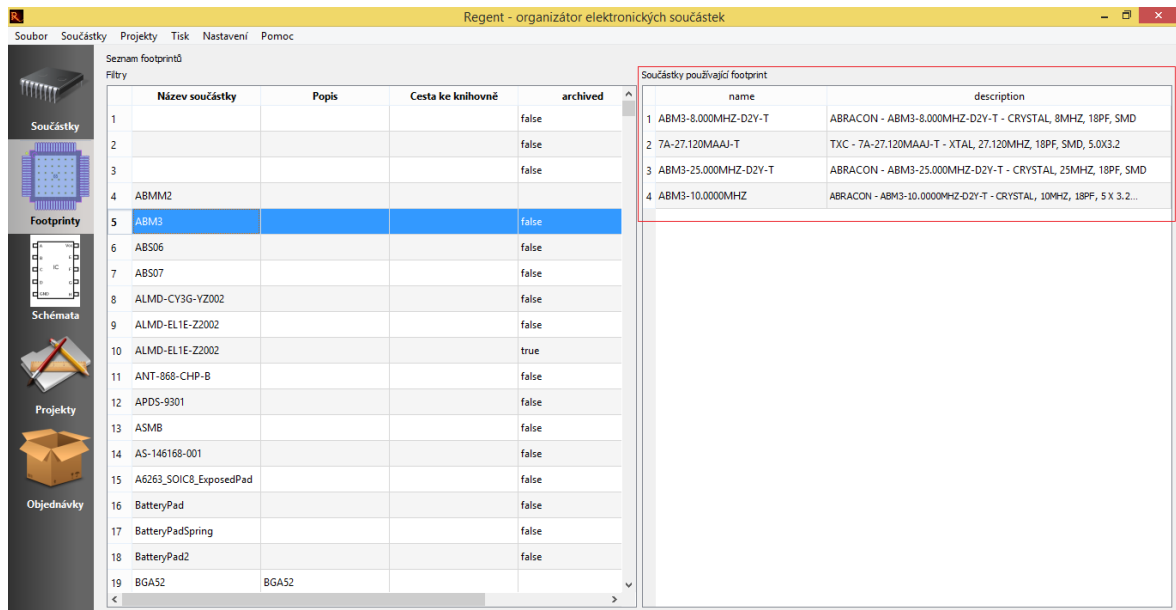
1.3.4 Pridanie novej súčiastky do systému

Novú súčiastku je možné pridať prostredníctvom totožného formuláru, aký je zobrazený pri detaile súčiastky. Vo formulári však chýba možnosť pre vloženie odkazu na obrázok (náhľad na súčiastku) a taktiež validácia, teda je možné vytvoriť prázdny objekt. Druhým spôsobom je importovať údaje o elektronickej súčiastke z API výrobcu Farnell. V tomto prípade používateľ zadá kód dodávateľa. Táto funkcia v súčasnej verzii nefunguje podľa očakávaní a je ju možné využiť len v rámci detailu elektronickej súčiastky.

1.3.5 Náhľad na zoznam schém a footprintov

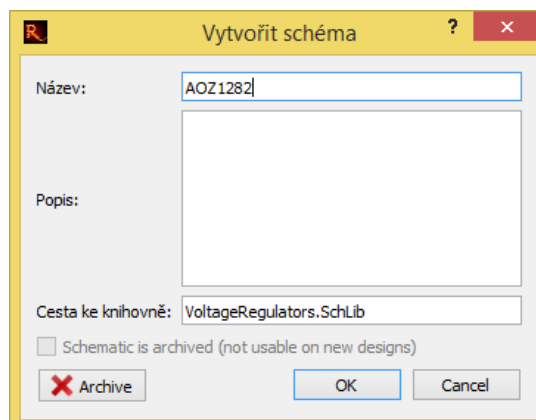
System Regent umožňuje používateľovi zobraziť zoznam schém a zoznam footprintov (Obr.7), ktoré jednotlivé elektronickej súčiastky využívajú. Jedná sa o dva rozdielne pohľady s totožnou funkcionalitou. Z tohto dôvodu bude v nasledujúcej časti popísaný iba pohľad zobrazujúci zoznam schém, pričom uvedené funkcie sú dostupné i pre pohľad so zoznamom footprintov.

Časť zameraná na používané schémy obsahuje zoznam schém a zároveň zoznam súčiastok, ktoré zvolenú schému využívajú. V rámci zoznamu schém môže používateľ vidieť informácie týkajúce sa schémy ako: názov, popis, cestu ku knižnici a či je schéma archivovaná (rovnaké informácie sú zobrazované v prípade footprintov). V zozname súčiastok sú zobrazené údaje o názve a popise súčiastky.



Obr. 7. Zoznam footprintov a súčiastok používajúcich zvolený footprint

V súčasnom stave nie je v systéme dostupné vyhľadávanie v zoznamoch schém/footprintov, zároveň zmienené zoznamy neimplementujú stránkovanie. Používateľ má rovnako ako v prípade zoznamu súčiastok filtrovaných na základe kategórie možnosť zobrazit' detail vybranej súčiastky a prístupit' k jeho editácii. Používateľ má tiež možnosť zobrazit' detail prislúchajúci k schéme a rovnako údaje editovať, pričom schému nie je možné odstrániť. Na obrázku (Obr.8) je zobrazený detail schémy spolu s dostupnými údajmi. Novú schému/footprint je možné registrovať prostredníctvom detailu elektronickej súčiastky (pomocou dialógového okna). Schéma/footprint, ktorá je evidovaná v systéme predstavuje referenciu na schému/footprint v rámci knižníc nástroja Altium Designer.

Obr. 8. Detail schémy – systém *Regent*

2 POŽIADAVKY NA CIEĽOVÚ APLIKÁCIU

Definícia požiadaviek na cieľovú aplikáciu predstavuje základnú, prvotnú fázu procesu vývoja softvéru. Samotný proces následne pozostáva z etáp ako návrh systému, implementácia modulov, integrácia (spájanie) modulov, testovanie a prevádzka spojená s údržbou. Definícia požiadaviek predstavuje aktivitu, ktorá vedie k vymedzeniu funkcionality aplikácie spolu s obmedzeniami pre vývoj a následnú prevádzku. Požiadavky je možné vo všeobecnosti definovať ako špecifikáciu toho, čo má byť v rámci výsledného systému implementované. Požiadavky slúžia na vyjadrenie chovania (funkcionality), ktoré bude systém poskytovať, nedefinujú spôsob akým je možné danú funkcionality dosiahnuť. [3]

2.1 Rozdelenie požiadaviek

Požiadavky je možné rozdeliť na základe viacerých aspektov. Jedným zo základných rozdelení je kategorizácia požiadaviek v závislosti od častí systému, ktoré popisujú. Z tohto pohľadu sa požiadavky delia na *funkčné*, *nefunkčné* a *doménové požiadavky*. Ďalší spôsob rozdelenia vyplýva z úrovne popisu požiadaviek.

Funkčné požiadavky

Funkčné požiadavky popisujú funkcionality systému. Určujú čo má systém vykonávať, respektíve definujú funkcie, ktoré majú byť poskytnuté používateľovi.

Nefunkčné požiadavky

Nefunkčné požiadavky, označované aj ako „mimofunkčné“ sú obmedzujúce podmienky kladené na systém. Nefunkčné požiadavky predstavujú vlastnosti systému, špecifikujú, respektíve vymedzujú spôsob akým bude daný systém implementovaný. Sú to napríklad požiadavky kladené na efektivitu, spoľahlivosť alebo bezpečnosť vyvíjaného systému. [3]

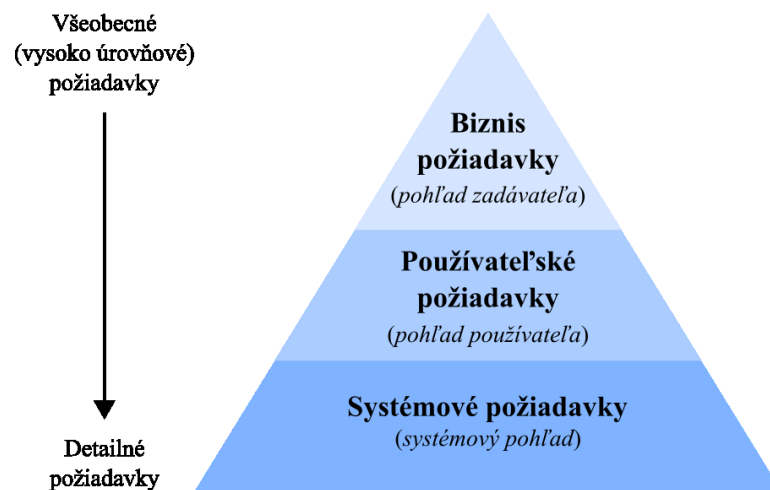
Doménové požiadavky

Cieľom doménových požiadaviek je popis problémovej domény, teda oblasti nasadenia. Môže sa jednať o predpisy, legislatívu alebo technické obmedzenia ako napríklad zmena chovania systému v závislosti od meteorologických podmienok. Doménové požiadavky môžu viesť k vzniku nových funkčných požiadaviek, prípadne k stanoveniu obmedzujúcich podmienok pre funkčné požiadavky.

Pre rozšírenie základného rozdelenia je možné požiadavky usporiadať do *taxonómie* - hierarchie typov, ktorá podrobnejšie rozdeľuje jednotlivé požiadavky. V rámci nefunkčných požiadaviek, kategorizovaných na základe typu, sa tak môže jednať o rozdelenie do skupín ako výkon, kapacita alebo zabezpečenie. V prípade funkčných požiadaviek môže ísť napríklad o platby, autentifikáciu, produkty a ďalšie. Zavedenie rozdelenia podľa typu má zmysel najmä v prípade práce s rozsiahlym počtom požiadaviek.[3]

2.1.1 Rozdelenie požiadaviek v závislosti od úrovne popisu

Z pohľadu *úrovne popisu* je možné požiadavky rozdeliť do troch skupín: biznis, používateľské a systémové. Rozdelenie požiadaviek na základe úrovne popisu je znázornené na obrázku (Obr. 9).



Obr. 9. Delenie požiadaviek podľa úrovne popisu

Biznis požiadavky

Biznis požiadavky vyjadrujú požiadavky kladené organizáciou alebo zákazníkom na výsledný systém či produkt. V praxi sa zvyčajne jedná o sadu potrieb, z ktorých plyní vznik nového produktu, prípadne zmena produktu. Biznis požiadavky majú zvyčajne formu len niekoľkých bodov alebo stručného dokumentu. Vznik biznis požiadaviek je najčastejšie iniciovaný na základe strategického rozhodnutia, spätnej väzby od zákazníkov alebo požiadaviek trhu.[4]

Používateľské požiadavky

Používateľské požiadavky sú zvyčajne zdokumentované použitím prostého textu. Opisujú na čo používateľ využíva systém, respektíve aké akcie musia byť používateľom sprístupnené. Jedná sa o stručnú charakteristiku funkcií. Používateľské požiadavky bližšie

špecifikujú biznis požiadavky a ďalej slúžia ako primárny vstup pre vytvorenie systémových požiadaviek.[5]

Systémové požiadavky

Systémové požiadavky obsahujú podrobný popis funkcionality systému, predstavujú „stavebné bloky“, od ktorých sa odvíja tvorba systému. Súčasťou systémových požiadaviek sú požiadavky funkčné aj nefunkčné. Okrem uvedeného popisu funkcionality teda systémové požiadavky zahŕňajú aj popis obmedzení a vlastností systému.

2.2 Definícia požiadaviek

Požiadavky vyplývajú z kontextu systému. K zberu a následne k definícii požiadaviek existuje viacero metód, medzi ktoré patrí: pozorovanie činnosti používateľov, práca s používateľmi, štúdium existujúcich systémov, dotazník a rozhovor. Z uvedených metód patrí k najvýznamnejším rozhovor, pri ktorom je možné kladením adekvátne formulovaných otázok získať najviac informácií. Následne je vytvorený dokument, ktorý obsahuje prehľad všetkých požiadaviek – funkčných, nefunkčných a doménových. K stanoveniu požiadaviek je možné aplikovať nasledovný postup.[3]

Postup definície a špecifikácie požiadaviek:

- Nájdenie funkčných požiadaviek.
- Stanovenie nefunkčných požiadaviek.
- Priradenie priority jednotlivým požiadavkám.
- Integrácia požiadaviek do prípadov použitia (modelovanie prípadov použitia).[3]

2.2.1 Modelovanie prípadov použitia

Okrem *modelu požiadaviek* (funkčné, nefunkčné a doménové požiadavky) sa k zachyteniu požiadaviek na systém používa *model prípadov použitia* (angl. *use case model*). Model požiadaviek spolu s modelom (diagramom) prípadov použitia reprezentujú požiadavky na systém rôznymi, ale vzájomne sa doplňujúcimi spôsobmi. Diagram prípadov použitia sa používa k formulácii požiadaviek v grafickom jazyku UML (Unified Modeling Language; Unifikovaný modelovací jazyk)² a ich vizuálnej reprezentácii.

² UML sa využíva k vizualizácii, návrhu a špecifikácii v rámci všetkých vývojových etáp programového systému

Prípady použitia predstavujú metódu pre zachytenie kontextu systému a popis funkčných požiadaviek. Každá z požiadaviek je realizovaná prostredníctvom jedného prípadu použitia. Jednotlivé prípady použitia teda definujú spôsob využitia systému z pohľadu používateľa. Reprezentácia požiadaviek formou prípadov použitia poskytuje nasledovné benefity: [3, 6]

- uľahčuje formulovať spôsob používania systému a komunikáciu s klientmi,
- zachytáva interakcie používateľa so systémom,
- zachytáva dynamické aspekty systému,
- pomáha identifikovať faktory ovplyvňujúce systém,
- umožňuje modelovať základný tok udalostí.[7]

Modelovanie prípadov použitia pozostáva z aktivít: stanovenie hraníc systému, vyhľadanie aktérov (angl. *actors*) a nájdenie prípadov použitia. Tento postup sa opakuje pokiaľ nedôjde k ustáleniu jednotlivých prvkov (hranice, aktéri, prípady použitia). Výstupom procesu je model prípadov použitia, ktorý pozostáva zo štyroch komponentov – hranice systému, aktéri, prípady použitia a relácie (vzťahy).[3]

Hranice systému

Hranice systému, tiež označované ako subjekt, slúžia na vymedzenie častí modelovaného systému.

Aktéri

Aktér je entita popisujúca rolu pridelenú osobám alebo predmetom, ktoré bezprostredne používajú systém.

Prípady použitia

Prípady použitia popisujú činnosti, ktoré systém umožní aktérom vykonávať. Zachytávajú sled udalostí a prípadov použitia systému špecifickým aktérom. Prípady použitia sú iniciované a popísané z pohľadu aktéra.

Relácie

Relácie, označované aj ako asociácie alebo väzby, sú vzťahy medzi aktérmi a prípadmi použitia. V rámci diagramu prípadov použitia existuje päť druhov väzieb:

- Asociácia medzi aktérom a prípadom použitia.
- Dedičnosť (generalizácia) medzi aktérmi.

- Dedičnosť medzi prípadmi použitia.
- Extend (rozšírenie) medzi prípadmi použitia.
- Include (zahrnutie) medzi prípadmi použitia.[8]

Generalizácia aktéra

Generalizácia aktéra umožňuje vyčleniť chovanie spoločné pre viacero aktérov. Je to väzba, ktorá značí zdedenie role aktéra iným aktérom. Potomok dedí všetky prípady použitia (asociácie) predka. Následne má potomok pridelený jeden alebo viacero prípadov použitia, ktoré sú špecifické pre jeho rolu (dochádza k špecializácii). [8]

Generalizácia prípadu použitia

Generalizácia prípadu použitia slúži k vyčleneniu spoločných funkcií pre viacero prípadov použitia do všeobecného, rodičovského prípadu. Chovanie predka je zdedené potomkom. Táto väzba je aplikovaná pokiaľ medzi prípadmi použitia existuje spoločné správanie a zároveň správanie špecializované, ktoré je špecifické pre daný prípad použitia.[8]

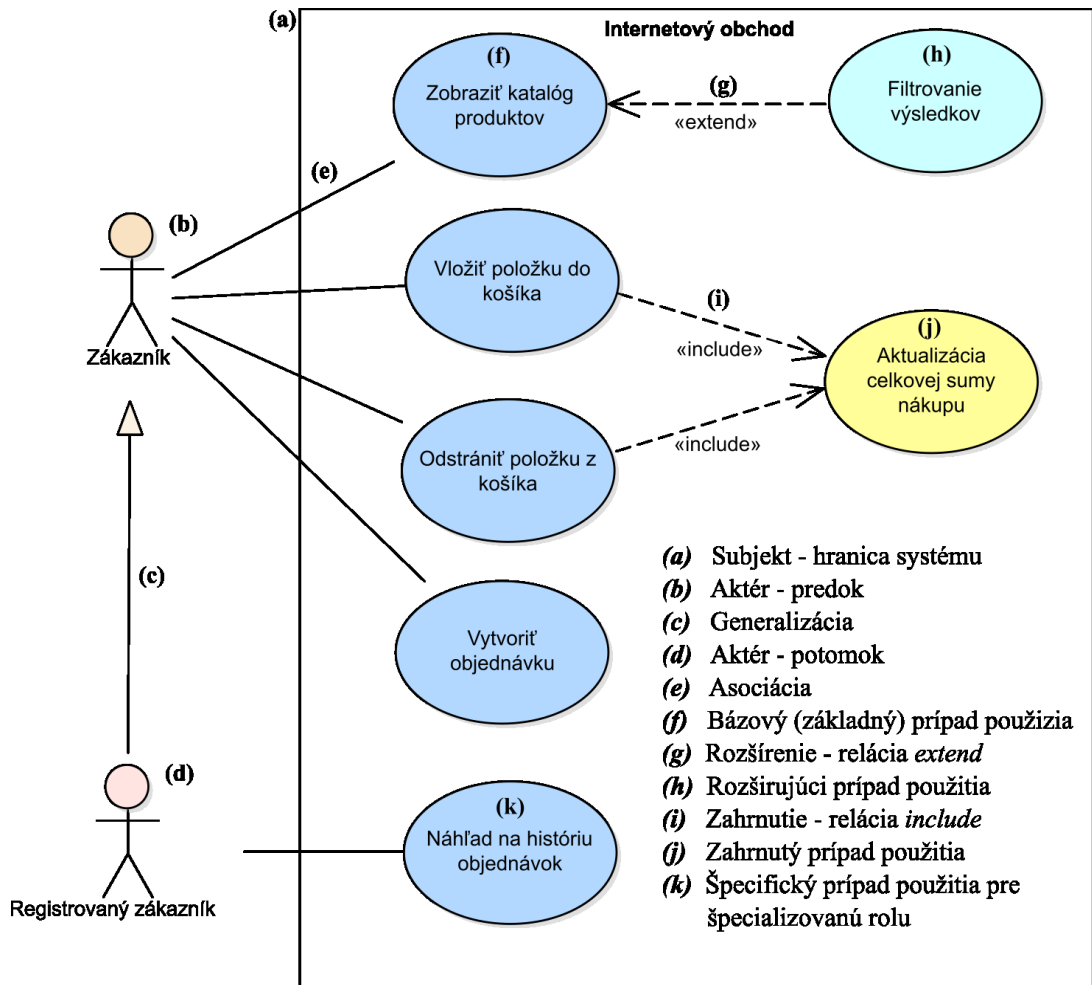
Zahrnutie

Zahrnutie (angl. *include*) je väzba, ktorá sa využíva k včleneniu správania prípadu použitia do iného prípadu použitia (základného). Hlavným dôvodom je možnosť opakovane použiť spoločné funkcie, ktoré sa opakujú sa viacerých prípadov použitia. Včlenenie sa zároveň používa k zjednodušeniu zložitého správania v modeli.[3]

Rozšírenie

Rozšírenie (angl. *extends*) sa používa medzi prípadmi použitia v situácii, keď jeden z prípadov rozširuje správanie druhého (základného) prípadu použitia. Rozširujúci prípad použitia predstavuje spôsob, ako je možné vložiť novú funkcionálnosť do už existujúceho prípadu. Rozširujúci prípad použitia závisí od základného (existujúceho), jeho prevedenie je zvyčajne voliteľné alebo podmienené.[8]

Na obrázku (Obr. 10) je znázornený vzorový diagram prípadov použitia určený pre internetový obchod. Obrázok obsahuje popis jednotlivých komponentov diagramu a ich vzájomných prepojení.



Obr. 10. Vzorový diagram prípadov použitia obsahujúci popis komponentov

3 BEZPEČNOSTNÉ RIZIKÁ

S poskytovaním služieb online, prostredníctvom webového rozhrania, je spojená rada bezpečnostných rizík. Nezabezpečenie dostatočnej ochrany môže viesť k neoprávnenému prístupu, k narušeniu prevádzky služby alebo manipulácii s informáciami. K vytýčeniu požadovaného zabezpečenia je potrebné preštudovať najčastejšie sa vyskytujúce útoky a následne podľa ich charakteru špecifikovať systémové požiadavky.

3.1 Útok hrubou silou

Útok hrubou silou (angl. *Brute-force attack*) je metóda typu pokus - omyl používaná k získaniu informácií pomocou uhádnutia vstupných údajov. V praxi sa teda jedná o systematické testovanie variant z množiny potencionálnych vstupov s využitím automatizovaného softvéru.

Útok hrubou silou sa môže vyskytovať v rôznych podobách. V prípade reverzného útoku hrubou silou (angl. *Reverse brute-force attack*) sa jedná o testovanie jedného hesla k mnohonásobnému počtu používateľských mien, prípadne šifrovaných informácií. Tento postup sa opakuje pre vybranú množinu hesiel. Ďalšou z metód útoku hrubou silou je použitie vopred pripraveného slovníku, táto metóda je označovaná ako slovníkový útok (angl. *Dictionary attack*).[9, 10]

3.2 Rainbow table attack

Útok označovaný ako „rainbow table attack“ sa zameriava na prelomenie hesiel, ktoré boli prevedené do formy otláčku (angl. *hash*) pomocou kryptografickej hashovacej funkcie³. K útoku je používaná **rainbow table**, ktorú môžeme definovať ako databázu, obsahujúcu vopred pripravené reťazce pozostávajúce zo vstupného a koncového bodu. Reťazce sú vytvárané opakovaným aplikovaním hashovacej a redukčnej funkcie (reverzná funkcia) na zvolený vstupný bod. Reťazce znižujú výpočtovú náročnosť a eliminujú nároky na pamäť. Pri hľadaní pôvodného textu je na výraz opakovane aplikovaná redukčná a hashovacia funkcia, pričom výsledok je porovnávaný s koncovými bodmi reťazcov pokiaľ nenastane zhoda. V prípade zhody nasleduje rekonštrukcia reťazca.[11, 12]

³ Funkcia k jednosmernému prevodu dát na reťazec definovanej dĺžky, bez možnosti získania pôvodného reťazca z výstupu

3.3 SQL injection

Útoku SQL injection sa zameriava na všetky typy databáz používajúcich jazyk SQL (Structured Query Language) a jeho formy (MySQL, PostgreSQL, SQL Server...). Jedná sa o útok, ktorý využíva vkladanie kódu – SQL príkazu do polí určených pre vstup s cieľom získať obsah databázy alebo manipulovať s uloženými dátami. SQL injection využíva zraniteľnosti, medzi ktoré patrí: nesprávne filtrovaný vstup od používateľa – tzn. znakov s priradeným významom v rámci jazyka SQL alebo ak používateľský vstup nie je pevne stanovený - používateľ zadá neočakávanú hodnotu vstupu.[12, 13]

3.4 Cross-site scripting

V prípade cross-site scripting sa jedná o útok na strane klienta pomocou podsunutia kódu. Cieľom útočníka je spustiť škodlivý kód (skript) vo webovom prehliadači obeť. Útočník zahrnie skript do oprávnenej časti webovej aplikácie. K samotnému útoku dochádza v momente keď obeť navštívi webovú aplikáciu, na ktorej sa vykoná spustenie škodlivého kódu. Webová aplikácia sa v tomto prípade stáva prostriedkom pre doručenie škodlivého skriptu do webového prehliadača obeť.

Cross-site scripting je často spájaný s inými typmi útokov. Patrí medzi jeden z najrozšírenejších spôsobov umožňujúcich útok session hijacking. Cieľom útoku session hijacking je odcudzenie session⁴ obeť, čo môže mať za následok krádež identity obeť a jej následné zneužitie.[14]

3.5 Cross-site request forgery

Cross-site request forgery, skrátene CSRF je útok, pri ktorom útočník vo svoj prospech oklame a prinúti obeť vykonať akcie v jej mene. Možný dopad útoku závisí od úrovne povolení, ktoré obeť má.

CSRF využíva skutočnosť, že webové aplikácie kompletne dôverujú používateľovi, ktorý je autentifikovaný. Vychádza z faktu, že množstvo webových stránok využíva dáta z iných webových lokalít. Ako príklad možno uviesť zobrazovanie videa na webovej stránke, ktorého úložisko je zvyčajne na inej webovej lokalite zameranej na streaming (napr. YouTube). Množstvo webových aplikácií taktiež ukladá svoje zdroje na vyhradených úložiskách (CDN), z ktorých sú následne sťahované. Túto skutočnosť môže útočník využiť

⁴ Dáta používateľa iniciované počas komunikácie so serverom – „identita“ používateľa

tak, že použije webovú stránku k získaniu údajov obete z inej webovej lokality bez jeho súhlasu. [15]

Útok CSRF je často podceňovaný, pričom sa môže jednať o ťažko detekovateľný a účinný útok v prípade, že je správne vykonaný. CSRF je pomerne rozšírený typ útoku, čo potvrdzuje aj jeho niekoľkonásobné umiestnenie v rebríčku desiatich najlepších útokov v OWASP (Open Web Application Security Project) Top 10⁵[16].

⁵ Dokument pre zvýšenie informovanosti vývojárov a zabezpečenia webových aplikácií

4 VÝBER TECHNOLOGIÍ PRE IMPLEMENTÁCIU

K tvorbe webovej aplikácie je možné využiť širokú škálu programovacích jazykov. Ich primárne rozčlenenie je možné urobiť na základe architektúry webovej aplikácie. Tá pozostáva z časti operujúcej s dátami – serverová časť (označovaná angl. *back-end*) a časti, ktorá poskytuje používateľské rozhranie – označovaná aj ako klientska časť (*front-end*). Programovacie jazyky na strane klienta sú: HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) a JavaScript. Na strane servera sa jedná o jazyky: PHP (PHP: Hypertext Preprocessor), Java, Python, Node.js/JavaScript, C# a ďalšie. Táto kapitola práce sa zameriava na zvolenie vhodnej technológie pre pokrytie oboch častí, popis vybraných technológií a zdôvodnenie ich výberu.

4.1 Skriptovací jazyk PHP

Skriptovací⁶ jazyk PHP je využívaný k tvorbe webových aplikácií postavených na architektúre klient-server. Predstavuje serverovú časť, kde dochádza k vykonávaniu kódu. Výstup je následne odoslaný na stranu klienta. PHP umožňuje logické operácie, zber dát formulárov, generovanie dynamického obsahu a iné. [17]

Použitie jazyka PHP v porovnaní s inými jazykmi pre serverovú časť, t.j. Python, C#, Java a iné, prináša nasledovné výhody: PHP kooperuje s najznámejšími webovými servermi, poskytuje nízku časovú náročnosť zahájenia projektu, PHP je multiplatformový a značne rozšírený, má rozsiahlu komunitnú základňu, z čoho vyplýva široká dostupnosť podpory a služieb tretích strán. Štandardne je podpora PHP dostupná u väčšiny poskytovateľov umožňujúcich webhosting. Ďalším benefitom jazyka PHP je spolupráca s mnohými databázovými systémami (skrátene angl. *DBMS*) ako napríklad MySQL, PostgreSQL, SQLite, Mssql, IBM DB2 a ďalšie. [17, 18]

4.2 Úvod k frameworkom jazyka PHP

Najčastejším riešením pre zefektívnenie vývoja je využitie frameworkov, ktoré sú k dispozícii pre programovacie jazyky na strane servera ako i klienta. Základnou ideou frameworku je automatizácia réžií spojených s bežnými činnosťami v rámci vývoja. Cieľom je poskytnúť základnú štruktúru pre tvorbu aplikácie, zamedziť duplicite kódu, zaistiť stabilitu, zjednodušiť a urýchliť vývoj. V rámci jazyka PHP je k dispozícii viacero

⁶ Inštrukcie sú interpretované za behu

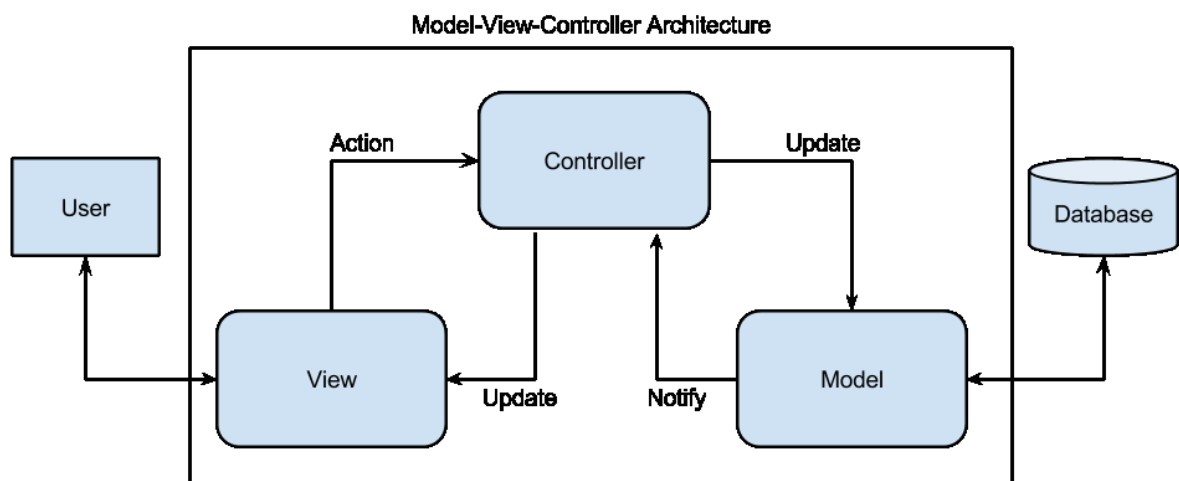
frameworkov. Základná myšlienka ich funkcionality vychádza z architektúry MVC (Model-View-Controller).[18]

4.2.1 Model-View-Controller

MVC je architektonický návrhový vzor v programovaní, ktorý izoluje logické procesy (biznis logiku) od používateľského rozhrania (skrátene angl. *UI; User Interface*). To umožňuje modifikovať jednotlivé časti separovane. MVC rozdeľuje proces vývoja aplikácie a preto je možné spracovávať jednotlivé časti individuálne zatiaľ čo zvyšné zostávajú neovplyvnené. MVC návrhový vzor pozostáva z troch komponentov:[19]

- **M** : Model – reprezentuje dáta,
- **V** : View – zodpovedá za vizuálnu prezentáciu dát (výstupu),
- **C** : Controller – odpovedá logickým operáciám, predstavuje spoj medzi komponentom *model* a *view* (pohľad), reaguje na udalosti, čím zaisťuje zmeny v týchto komponentoch.[20]

Na obrázku (Obr. 11) sú prostredníctvom diagramu zachytené jednotlivé komponenty MVC návrhového vzoru.



Obr. 11. Softvérová architektúra MVC návrhového vzoru [21]

4.2.2 Nástroje na správu závislostí

Pri práci s frameworkami sú využívané nástroje pre správu závislostí – kódu (kníhnic) tretích strán. Tieto nástroje automatizujú proces správy (aktualizácie, inštalácie, a odstránenia) jednotlivých závislostí.

Composer slúži ako nástroj pre správu závislostí jazyka PHP. Umožňuje inštalovať knižnice, komponenty jazyka PHP, ktoré sú formou balíkov dostupné na hlavnom repozitári *Packagist*. Composer spravuje deklarované závislosti projektu, čo znamená, že požadované knižnice sú stiahnuté, aktualizované a dostupné k použitiu.

Ďalším nástrojom je **npm** (Node Package Manager). Npm slúži pre správu závislostí a distribúciu kódu pre programovací jazyk JavaScript s hlavným repozitárom *npmjs*. kde sú jednotlivé knižnice dostupné.

4.3 Porovnanie PHP frameworkov

V rámci popularity jednotlivých frameworkov jazyka PHP sa na popredných priečkach (na základe Google Trends[22], GitHub[22]) najčastejšie umiestňoval *Laravel* a *Symfony*. Obe uvedené technológie sa používajú na strane servera, zároveň však poskytujú nástroje pre tvorbu používateľského rozhrania. Nasledujúca časť je venovaná ich popisu formou porovnávania vlastností, ktoré poskytujú.

Laravel

Laravel je open-source framework, ktorý sa riadi návrhovým vzorom MVC. K vytvoreniu webovej aplikácie používa existujúce komponenty rôznych frameworkov, vďaka čomu ponúka širokú sadu funkcií. Laravel poskytuje štruktúrovanosť a pragmatickosť pri vývoji webovej aplikácie. Laravel je známy svojou jednoduchosťou prevedenia, čo vedie k skráteniu časovej náročnosti vývoja. Okrem toho Laravel poskytuje bezpečnostné opatrenia zabráňujúce viacerým potencionálnym útokom.[23, 24]

Symfony

Symfony je voľne dostupný framework predstavujúci radu opakovane použiteľných PHP komponentov. Symfony rovnako ako Laravel vychádza z návrhového vzoru MVC a je založený na niekoľkých open-source PHP projektoch ako napríklad Doctrine, Propel, Swift Mailer a PHP Unit. Symfony je vo všeobecnosti používaný pre komplexné podnikové projekty. Symfony je stavebným pilierom pre PHP projekt Drupal a vyššie uvedený Laravel, ktoré používajú komponenty z frameworku Symfony.[23]

4.3.1 Modularita a škálovanie

Symfony obsahuje opakovateľne použiteľné komponenty, ktoré poskytujú väčšiu modularitu. Symfony organizuje kód spôsobom, ktorý je o niečo prehľadnejší, vďaka čomu sa výborne hodí pre rozsiahlejšie a komplexnejšie projekty.[25]

Princíp a efektivita Laravelu je založená na MVC paradigme. Laravel nie je natoľko flexibilný, ale v prípade webových aplikácií s MVC návrhovým vzorom je používateľsky prívetivejší.

Pri tvorbe aplikácie od základu poskytuje Laravel užitočnú skupinu predefinovaných závislostí a pripravených implementácií, ktoré výrazne urýchlia samotný vývoj. V prípade ak sa však jedná o rozsiahlu škálovateľnú aplikáciu, je pre rozširovanie aplikácie vhodnejšie vybrať si potrebné závislosti a doplnky na základe požiadaviek. Z tohto dôvodu je pre takýto typ projektov preferovanejší framework Symfony.[25]

4.3.2 Šablónovacie nástroje

Jazyk PHP predstavuje sám o sebe šablónovací prostriedok. Poskytnutie ďalších prostriedkov pre šablónovanie prináša pokročilejšie možnosti. Použitie šablónovacích nástrojov pomáha udržať biznis logiku v rámci PHP a prezentačnú časť v použitom šablónovacom prostriedku. [25]

Symfony poskytuje *Twig* ako predvolený šablónovací nástroj. V tomto ohľade poskytuje Laravel viac, vďaka šablónovaciemu nástroju *Blade*. Jeho zásadná výhoda oproti nástroju *Twig* spočíva v tom, že umožňuje opakované použitie kódu prostredníctvom komponentov. Oba uvedené nástroje slúžia k tvorbe UI a sú založené na hypertextovom jazyku HTML, ktorý definuje štruktúru webovej stránky. K definícií štýlov je možné použiť jazyk CSS, v prípade poskytnutia interaktívnosti skriptovací jazyk JavaScript.[24, 25]

4.3.3 Databázová podpora

Laravel ako aj Symfony poskytujú objektovo-relačné mapovanie (angl. *ORM*; *Object-Relational Mapping*) pre prístup k dátam. *Doctrine* riadi ORM pre Symfony, zatiaľ čo *Eloquent* poskytuje ORM pre Laravel. V oboch prípadoch sa vďaka ORM stáva manipulácia s dátami o mnoho jednoduchšia. V rámci databázy existuje medzi frameworkami viacero odlišností. Jednou z nich je poskytovaná podpora, ktorá je uvedená v tabuľke (Tab. 1).[25]

Tab. 1. Databázová podpora (Laravel, Symfony)[25]

Databáza	Laravel	Symfony
Drizzle	-	✓
MySQL	✓	✓
Oracle	-	✓
PostgreSQL	✓	✓
SAP Sybase SQL Anywhere	-	✓
SQLite	✓	✓
SQLServer	✓	✓

Databázové migrácie

V prípade Symfony sú databázové migrácie uskutočňované automaticky. Je potrebné len jednoduchá definícia polí v rámci modelu. Pri použití Laravelu je nutné vykonávať databázové migrácie manuálne, ale na rozdiel od Symfony nevyžadujú definíciu polí.[25]

Modelovanie dát

Pri použití Laravelu vyžaduje prístupovanie k dátam o niečo viac znalostí jazyka SQL. Symfony tieto nároky znižuje, avšak je potrebné aby bola pre každé volanie definovaná funkcia v repozitári. Takýto spôsob môže viesť k veľkému počtu týchto funkcií v prípade, že aplikácia využíva komplexnú databázovú štruktúru.[25]

Eloquent ORM (Laravel) zväzuje aplikáciu s návrhom schémy databázy. Zmena názvu stĺpca môže mať za následok zmenu zasahujúcu do celého kódu. Doctrine, Symfony poskytuje rozšírenie pre použitie mapovacej schémy, takže je možné oddeliť databázovú schému a reprezentáciu údajov v pamäti (PHP objekty).[25]

4.3.4 Zhrnutie

Z uvedeného porovnania frameworku Laravel a Symfony vyplýva, že nie je možné jednoznačne určiť, ktorý z frameworkov poskytuje viac. Oba prinášajú pridanú hodnotu v rozličných aspektoch. Preto je pri výbere potrebné vziať v úvahu zámer projektu. V prípade menších projektov sa Laravel javí ako lepšia voľba, dokáže efektívne poskytnúť potrebné komponenty (implementácie) pripravené na použitie, ktoré uľahčia štart aplikácie.

Symfony je odpoveďou pre robustnejšie projekty. Poskytuje väčšiu modularitu a škálovateľnosť a teda aj lepšie prispôsobenie špecifickým požiadavkám. Symfony poskytuje len skutočne najdôležitejšie a umožňuje pridávať iba prvky, ktoré sú v rámci projektu požadované.

K implementácii výslednej aplikácie bol vybraný framework **Laravel**, s ohľadom na rozsah projektu a vyššie uvedené porovnanie. Jeho použitie umožní výrazne skrátiť čas vývoja. Vďaka poskytovanému súboru bezpečnostných opatrení poskytne pokročilú ochranu, bez potreby implementácie ďalších bezpečnostných prvkov. Vysoká popularita a široká používateľská základňa zároveň znamená dlhodobu pretrvávajúcu podporu. V prípade klientskej časti Laravel poskytne výhodu vo forme opakovateľne použiteľných komponentov šablónovacieho nástroja Blade.

II. PRAKTICKÁ ČASŤ

5 ŠPECIFIKÁCIA POŽIADAVIEK

Výsledné riešenie musí implementovať vlastnosti a funkcionality, ktorú zapuzdruje desktop systém Regent, aktuálne používaný systém pre riadenie skladových zásob. Vzhľadom k tejto skutočnosti, vychádzajúc z časti práce zameranej na existujúce open-source riešenia skladovej evidencie (kapitola 1) a kapitolu popisujúcu bezpečnostné riziká (kapitola 3), boli definované nasledovné systémové požiadavky.

5.1 Nefunkčné požiadavky

Systém bude spĺňať nasledovné kritéria:

- bude implementovaný prostredníctvom webového rozhrania,
- poskytne podporu pre spoluprácu s nástrojom Altium Designer,
- umožní komunikáciu s API výrobcu Farnell,
- bude disponovať responzívnym dizajnom,
- k správe dát bude použitý databázový systém PostgreSQL,
- každý používateľ musí mať unikátnu e-mailovú adresu,
- všetky vstupy od používateľa musia byť validované (formát vstupu, existencia výberu v tabuľke, ustanovenie povinných vstupov...),
- používateľ bude pri vykonávaní jednotlivých akcií notifikovaný,
- v rámci systému budú zavedené používateľské role – administrátor a používateľ.

5.1.1 Bezpečnostné požiadavky

Bezpečnostné požiadavky reagujú na definované bezpečnostné riziká popísané v rámci teoretickej časti (kapitola 3). Cieľom týchto požiadaviek je zamedziť potencionálnym útokom. V prípade **útoku hrubou silou**, ktorého úspech je závislý od výpočtového výkonu, sa jedná o opatrenia:

- Obmedzenie počtu neúspešných pokusov o prihlásenie.
- Použitie neutrálnej spätnej väzby pri autentifikácii – používateľovi nebude oznámené, ktorý zo vstupných údajov bol zadaný nesprávne.

- Definícia dĺžky a komplexnosti hesla - stanovenie regulárneho výrazu (skr. angl. *regex*) pre heslo: minimálna dĺžka hesla musí byť viac ako osem znakov, heslo musí obsahovať aspoň jednu číslicu.

K zamedzeniu ďalším opísaným typom útokov nie je potrebná explicitná definícia požiadaviek. Jedná sa o bezpečnostné opatrenia, ktoré poskytne framework Laravel pri zachovaní zavedených štandardov. Prehľad týchto opatrení sa nachádza v tabuľke (Tab.2).

Tab. 2. Zoznam útokov a poskytnutých opatrení (Laravel)

Typ útoku	Poskytované opatrenie
SQL injection	Parametrizované (prepared) dotazy
Rainbow table attack	Aplikácia soli pri použití hashovacej funkcie (<i>Bcrypt</i>)
Cross-site request forgery	Použitie tokenov v rámci formulárov
Cross-site scripting	Filtrácia údajov zobrazovaných v pohľade

5.2 Funkčné požiadavky

Funkčné požiadavky mapujú funkcionality v závislosti od súčasného riešenia – systém Regent, ktorého popis sa nachádza v prvej kapitole práce. Táto časť obsahuje ich stručný popis a definíciu novej funkcionality a úprav, ktoré budú implementované.

Registrácia

Vytváranie nových používateľských účtov bude umožnené výhradne administrátorovi. Pri registrácii účtu administrátor zadá e-mailovú adresu. Zadaná e-mailová adresa musí byť unikátna a správnom tvare. Administrátor nezadáva heslo k vytváranému účtu. Heslo je vygenerované automaticky. Pri registrácii bude na zadanú e-mailovú adresu odoslaný e-mail, vyzývajúci používateľa k nastaveniu hesla prostredníctvom obnovovacieho formuláru. Po nastavení hesla bude účet používateľa označený za aktivovaný a pripravený k použitiu.

Prihlásenie

Prihlásenie do systému bude realizované prostredníctvom formuláru. Pre prihlásenie je potrebné aby mal používateľ nastavené heslo podľa pokynov prijatých po registrácii. V rámci prihlasovacieho formuláru používateľ vyplní e-mailovú adresu a heslo. Používateľ je oboznámený v prípade neúspešného pokusu o prihlásenie, nie je mu však poskytnutá informácia, ktorý z údajov bol zadaný nesprávne. Pred prihlásením bude používateľovi

sprístupnené označiť možnosť „zostať prihlásený“. Používateľ viac nebude mať prístup k zoznamu registrovaných účtov.

Obnova hesla

Používateľovi bude poskytnutá možnosť obnoviť zabudnuté heslo. Pri obnovovaní hesla používateľ zadá e-mailovú adresu. Následne bude na zadanú adresu odoslaný e-mail obsahujúci odkaz slúžiaci k verifikácii. Po kliknutí na odkaz, v prípade že sa jedná o odkaz platný, bude používateľ presmerovaný na stránku s formulárom, kde mu bude umožnené nastaviť si nové heslo.

Zobrazenie profilových informácií

Používateľ bude mať k dispozícii zobrazenie profilových informácií. Medzi tieto údaje patrí: meno, e-mailová adresa, používateľská rola a dátum posledných zmien týkajúcich sa profilu. V profile bude používateľovi poskytnutá možnosť odstrániť svoj účet.

Editácia profilu

V rámci editácie profilu má používateľ možnosť zmeniť údaje používané k prihláseniu sa do systému (e-mailová adresa a heslo). Pri zmene hesla používateľ zadáva aktuálne heslo, nové heslo a potvrdenie hesla. V prípade ak sa rozhodne pre zmenu e-mailovej adresy je pre dokončenie operácie potrebné aby zmenu potvrdil kliknutím na potvrdzovací odkaz, ktorý je pri žiadosti o zmenu odoslaný na novú e-mailovú adresu.

Správa používateľských účtov

Používateľ s rolou administrátora bude môcť zobraziť zoznam zaregistrovaných účtov, pričom bude mať k dispozícii informácie o mene používateľa, e-mailovej adrese a o tom či bol účet aktivovaný. Taktiež bude mať možnosť účet odstrániť, ale len v prípade, že sa nejedná o účet administrátora. V tomto prípade je potrebné použiť funkciu pre odstránenie účtu v profilovej sekcii.

Zobrazenie zoznamu elektronických komponentov filtrovaných na základe kategorizácie

Zoznam elektronických súčiastok bude zobrazovaný na základe zvolenej kategórie. Zobrazovaný zoznam bude obsahovať stránkovanie s možnosťou výberu počtu zobrazovaných záznamov. Používateľ bude môcť zoznam usporiadať podľa zvolených kritérií (abecedne, podľa počtu kusov). V zozname bude môcť používateľ dynamicky vyhl'adávať a to s ohľadom na aktívne filtre (usporiadanie, zvolená kategória).

Zobrazenie podrobných informácií o vybranej súčiastke a ich editácia

Po zvolení položky zo zoznamu elektronických súčiastok bude používateľovi zobrazený detail príslušnej súčiastky. Detail poskytne podrobné údaje o súčiastke, vrátane vizuálnej reprezentácie súčiastky (pokiaľ bude k dispozícii). V rámci detailu bude mať používateľ možnosť editovať jednotlivé údaje, prípadne súčiastku odstrániť, čomu bude predchádzať potvrdenie modálneho dialógu. Pri editácii bude každý z údajov validovaný. V tomto prípade už viac nebude viac mať v databáze záznam o súčiastke bez priradeného názvu alebo počtu kusov.

Pridanie novej súčiastky do systému

Pridanie novej súčiastky do skladového systému bude realizované prostredníctvom vyhradenej stránky. Pri pridaní bude mať používateľ možnosť importovať údaje súčiastky z programového rozhrania výrobcu Farnell. Predtým je však potrebné aby vyplnil číslo u dodávateľa, ktoré je priradené súčiastke. Používateľ bude môcť formulár pre pridanie vyplniť aj manuálne a zároveň tak i upravovať údaje poskytnuté prostredníctvom importu. V rámci formuláru pribudne možnosť nastaviť odkaz pre obrázok súčiastky. Odoslaný formulár pre vytvorenie súčiastky bude validovaný.

Náhľad na zoznam schém a footprintov, zoznam elektronických súčiastok, ktoré využívajú zvolenú schému/footprint

Zoznam používaných schém a zoznam footprintov bude stránkovaný, zároveň bude stránkovaný aj zoznam elektronických súčiastok, ktoré zvolenú schému/footprint používajú. Taktiež v rámci tejto funkcionality pribudne vyhľadávanie, takže používateľ bude môcť dynamicky hľadať v týchto zoznamoch.

Zobrazenie detailných informácií o schéme a footprinte

Rovnako ako v prípade elektronických súčiastok bude mať používateľ možnosť zobrazit detailné informácie o schéme alebo footprinte. Pri detaile okrem možnosti editácie údajov pribudne možnosť zvolenú schému odstrániť.

Vytvorenie novej schémy a footprintu

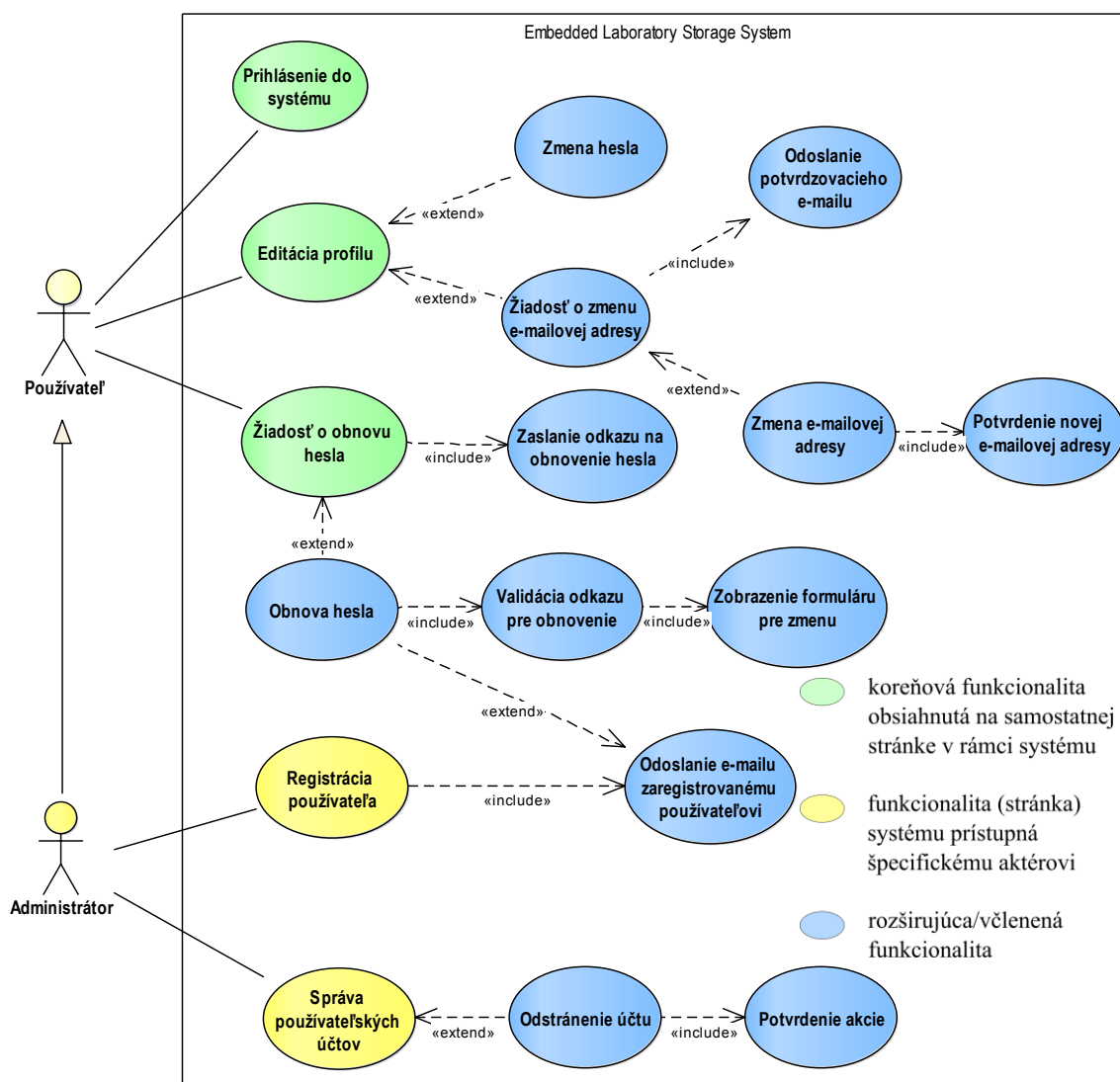
K vytvoreniu schémy alebo footprintu bude môcť používateľ prístupit priamo prostredníctvom tlačidla na jednotlivých stránkach. K vytváraniu pribudne požadovaná validácia.

Import údajov z API výrobcu

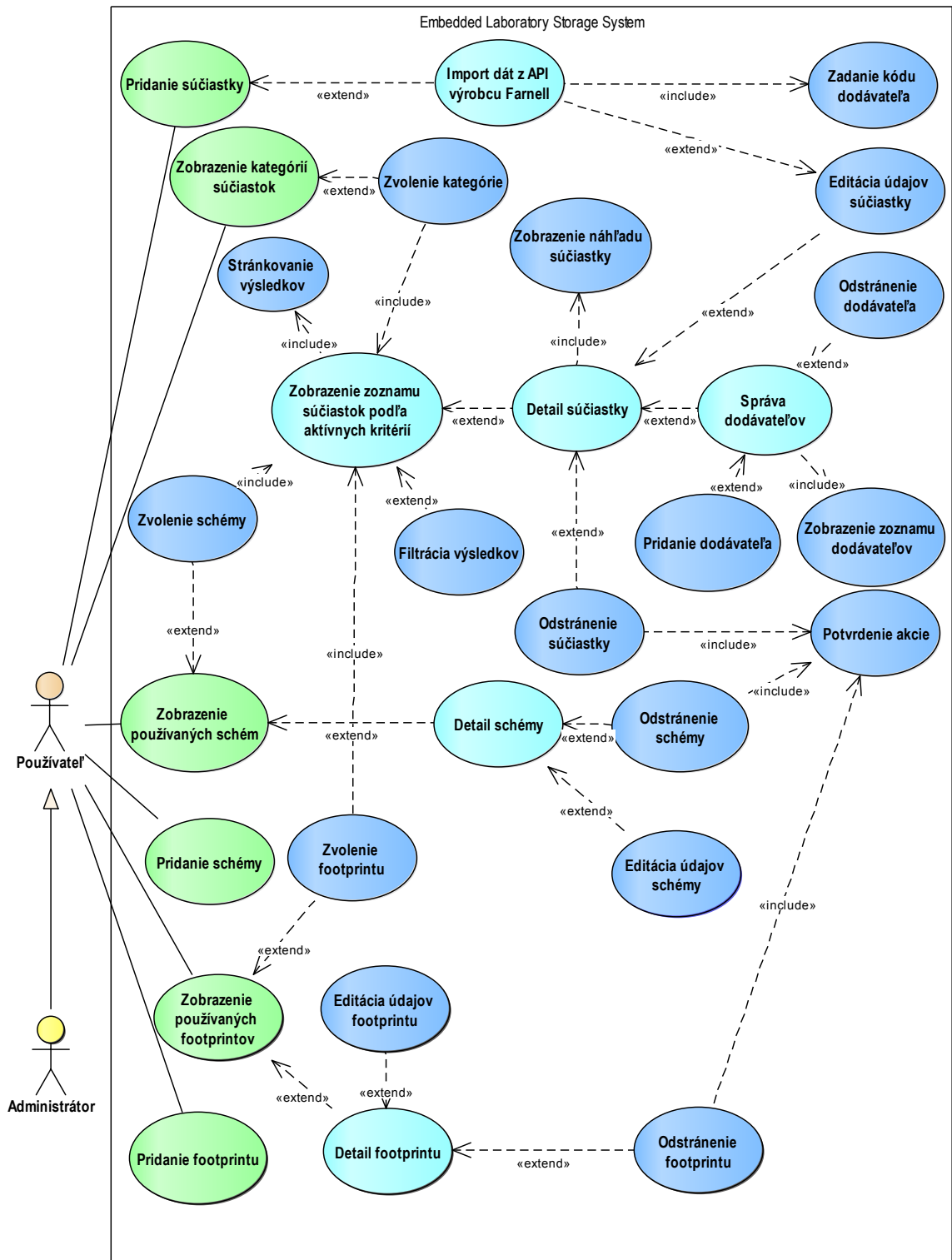
Rovnako ako v prípade systému Regent, bude používateľ môcť importovať údaje elektronickej súčiastky z programového rozhrania výrobcu Farnell tak, že zadá kód dodávateľa prislúchajúci súčiastke. Ako bolo uvedené, táto funkcia bude dostupná pri pridávaní súčiastky, ale zároveň aj v rámci detailného náhľadu súčiastky.

5.3 Diagram prípadov použitia

Na obrázku (Obr. 12) je zobrazený diagram prípadov použitia zachytávajúci funkčné požiadavky súvisiace s autentifikáciou a používateľskými účtami. Zvyšné funkčné požiadavky sú zahrnuté v diagrame prípadov použitia na obrázku (Obr. 13).



Obr. 12. Funkčné požiadavky spojené s autentifikáciou a používateľskými účtami (*use case diagram*)



● koreňová funkcionlita obsiahnutá na samostatnej stránke v rámci systému
 ● esenciálna funkcionlita
 ● rozširujúca/včlenená funkcionlita

Obr. 13. Funkčné požiadavky na vyvíjaný systém (use case diagram)

6 IMPLEMENTÁCIA

Obsahom tejto kapitoly je popis architektúry, návrh zostavenia častí vyvíjaného systému ako aj zhrnutie postupu pri implementácii jednotlivých funkcií.

6.1 Architektúra aplikácie

Táto časť sa zameriava na popis technológií, ktoré budú použité v rámci implementácie, popis architektúry klient – server, priblíženie architektúry aplikácie vytvorenej prostredníctvom frameworku Laravel a popis dizajnového vzoru MPA (Multi-Page Applications).

6.1.1 Vybrané technológie pre implementáciu

Na základe teoretickej časti práce (kapitola 4) bol k implementácii, ako primárna technológia, vybraný framework **Laravel** (6.17.0). K definícii klientskej časti aplikácie je Laravel (šablónovací nástroj Blade) použitý v kombinácii s nižšie uvedenými technológiami.

Bootstrap 4

Front-end framework Bootstrap 4 je sada nástrojov využívajúcich jazyk HTML, CSS, a JavaScript pre vývoj používateľského rozhrania. Bootstrap 4 poskytuje šablónu pre dizajn elementov ako napríklad vstupné polia, formuláre, tlačidlá, tabuľky, typografia a ďalšie. Jednotlivé grafické komponenty je zároveň možné globálne prispôsobiť prostredníctvom vstavaných premenných preprocesora Sass (Syntactically Awesome Style Sheets). Hlavnú výhodu Bootstrapu však predstavuje mriežkový systém zabezpečujúci responzívny dizajn a množstvo komponentov a doplnkov postavených na knižnici jQuery.

Sass

Sass umožňuje efektívnejšie definovanie štýlov webovej aplikácie. V stručnosti sa jedná o preprocesor CSS. Sass poskytuje funkcie, ktoré štandardne nie sú dostupné v jazyku CSS, jedná sa napríklad o premenné, vnáranie, dedičnosť a ďalšie.

Vue.js

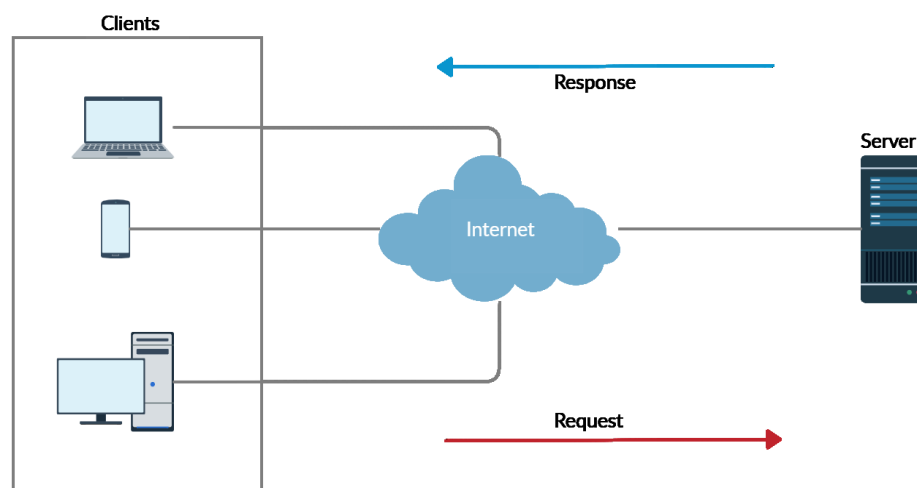
Vue je „reaktívny“ front-end framework pre budovanie používateľského rozhrania. Na rozdiel od iných porovnateľných frameworkov je Vue navrhnutý od základov tak, aby v čo najväčšej miere poskytoval modularitu.

Knižnica tvoriaca jadro frameworku je zameraná výhradne na vrstvu pohľadu aplikácie (UI) a je ľahko integrovateľná aj s inými knižnicami alebo existujúcim projektom. Vďaka štandardnej podpore Laravelu je Vue možné bez problémov začleniť do projektu a využiť jeho poskytované komponenty.[26]

6.1.2 Model klient – server

Riešenie bude vo všeobecnosti vychádzať z architektúry (modelu) klient-server (Obr. 14). V prípade modelu klient-server sa jedná o distribuovanú aplikačnú štruktúru, ktorá rozdeľuje úlohy (spracovanie) na poskytovateľa služby – server a žiadateľa o službu – klient. [27]

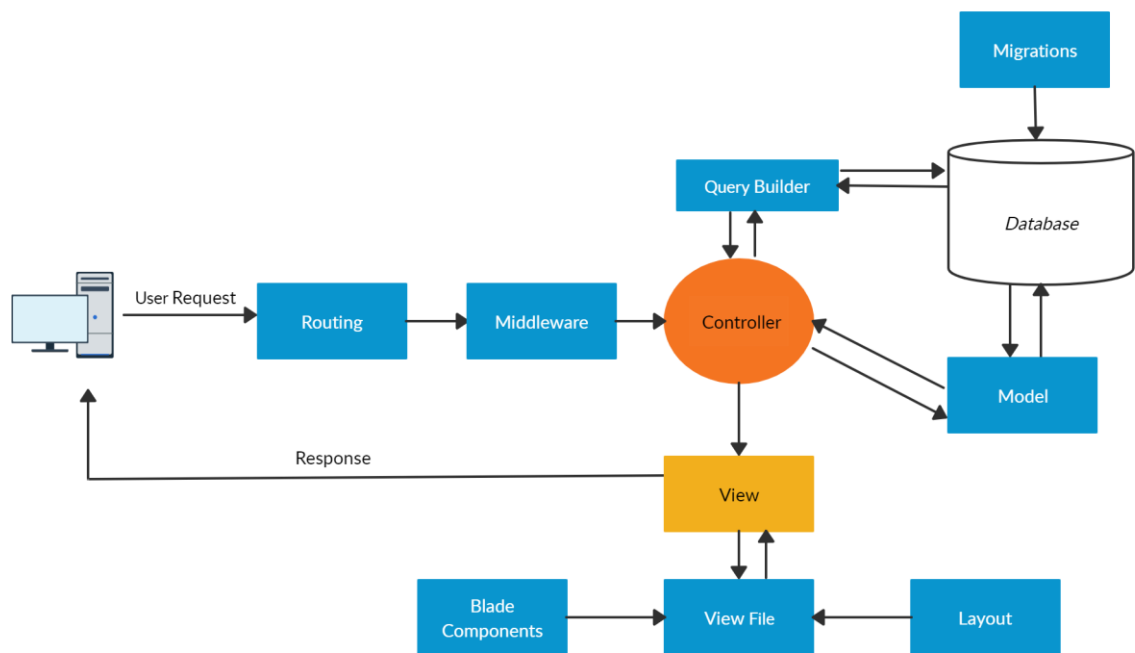
V klient-server architektúre ukladá všetky údaje a spravuje väčšinu procesov server. Klient posiela žiadosti o údaje na server cez internet, server prijme požadovaný proces a odovzdá procesný výstup klientovi. Klient v niektorých prípadoch obstaráva spracovanie, ale k dokončeniu vyžaduje dáta zo servera.[28]



Obr. 14. Model klient – server

6.1.3 Architektúra aplikácie Laravel

Framework Laravel sa opiera o MVC návrhový vzor, čo sa premieta aj do samotnej architektúry aplikácie. Na obrázku (Obr. 15) sú zobrazené časti z ktorých pozostáva aplikácia vytvorená s použitím Laravelu. Z obrázku je možné vidieť, že jadro aplikácie je tvorené jednotlivými komponentami MVC návrhového vzoru.



Obr. 15. Architektúra Laravel aplikácie

6.1.4 Dizajnový vzor multi-page application

K implementácii aplikácie bude použitý dizajnový vzor *multi-page application* (MPA). Dizajnový vzor MPA predstavuje tradičný spôsob implementácie webových aplikácií, kedy pri interakcii používateľa dochádza pri výmene dát k načítaniu celej lokality. Tento dizajnový vzor je použitý z dôvodu, že procesný výstup serveru nie je spracovávaný inými aplikáciami a preto nie je potrebné striktné oddeliť časti front-end a back-end. Zároveň vďaka responzívnemu dizajnu zaniká nutnosť vytvorenia mobilnej aplikácie, ktorá by mohla vyžadovať výstup servera.

V rámci aplikácie je kladený zámer viac na obsah ako na grafické spracovanie. Potrebnú dynamickosť je možné zabezpečiť použitím modálne zobrazovaných komponentov (Bootstrap 4), Vue.js komponentov, knižnice JQuery a asynchrónnych žiadostí (AJAX).

Dizajnový vzor MPA zároveň nevyžaduje použitie zvlášť technológie pre front-end, je teda možné využiť všetky predpripravené funkcie poskytované Laravelom. A to od funkcionality poskytovanej šablónovacím nástrojom Blade až po pripravené implementácie ako autentifikačný systém.

6.2 Návrh implementácie

Táto časť je venovaná všeobecnému návrhu pre implementovanie funkčných požiadaviek aplikácie.

6.2.1 Zostavenie jednotlivých častí

Pri zostavovaní bude aplikácia v rámci implementácie rozdelená na dve časti: časť týkajúca sa autentifikácie a správy používateľských účtov a časť zameraná na jadro systému (operácie nad elektronickými súčiastkami).

Návrh zostavenia aplikácie sa opiera o diagram prípadov použitia. Časti aplikácie (UI), ktoré sú spoločné pre viacero pohľadov budú implementované pomocou opakovaním použiteľných komponentov nástroja Blade. Týmto spôsobom je možné zamedziť vzniku duplicity kódu a zvýšiť samotnú udržateľnosť kódu. Medzi zdieľané komponenty patrí napríklad zobrazenie detailu elektronickej súčiastky, zobrazenie potvrdzovacieho dialógu alebo detail schémy/footprintu.

Rovnaký postup bude použitý v prípade vyhradených funkcií, ktoré sú využívané v rámci viacerých častí aplikácie (pohľadov). Jedná sa tak napríklad o funkciu zabezpečujúcu zobrazenie zoznamu elektronických súčiastok. Z diagramu prípadov použitia vyplýva, že funkcia je volaná pri zobrazení súčiastok na základe aktívnej kategórie rovnako ako pri zozname používaných footprintov/schém, kedy zoznam predstavuje elektronické súčiastky filtrované na základe zvoleného footprintu/schémy. Táto funkcia bude implementovaná tak, aby ako parametre obsahovala jednotlivé filtračné kritéria a bolo možné ju použiť pre každý z pohľadov.

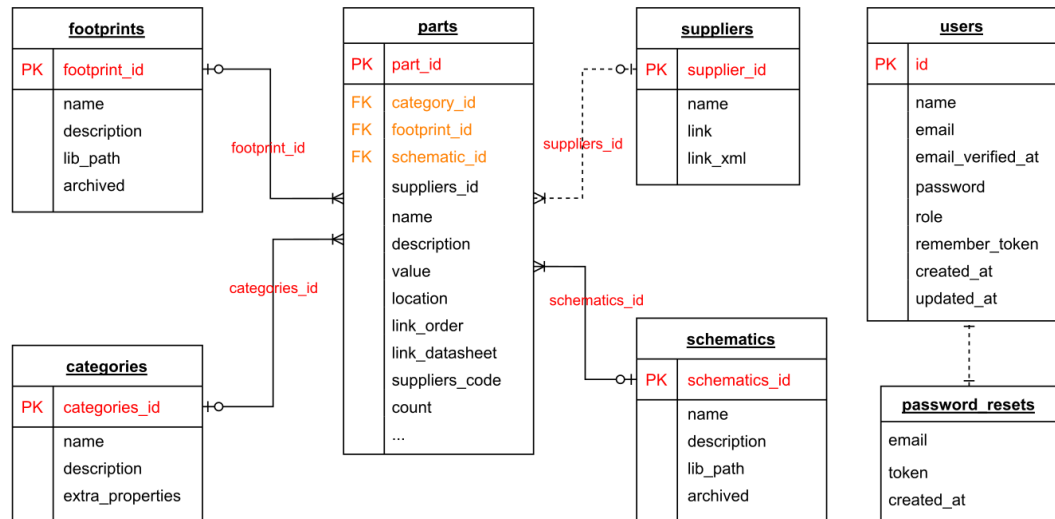
Táto časť reprezentuje všeobecný návrh zostavenia častí. Podrobnejší popis postupu pri implementácii jednotlivých funkcií sa nachádza v nasledujúcej sekcii - metodika postupu práce.

6.2.2 Databázové pripojenie

Aplikácia bude v rámci vývoja pripojená k objektovo-relačnému databázovému systému PostgreSQL. Lokálna databáza bude vytvorená prostredníctvom *dump*⁷ súboru a teda bude kopírovať súčasne používanú databázu. Tento postup má zaistiť plnú kompatibilitu pri presune z testovacieho nasadenia. Architektúra databázy je navrhnutá v súvislosti

⁷ Databázový výpis, ktorý obsahuje štruktúru a dáta databázy vo forme SQL príkazov

s požiadavkou kladenou na kooperáciu systému s nástrojom Altium Designer (sú definované virtuálne tabuľky - pohľady). Jednotlivé pohľady databázy (angl. *views*) reprezentujú kategórie v rámci elektronických súčiastok. Nasledujúci obrázok (Obr. 16) zobrazuje generálnu časť databázovej štruktúry, ktorá je využívaná počas interakcie so systémom.



Obr. 16. Schéma databázy

6.3 Metodika implementácie riešenia

Nasledujúci text zachytáva stručný popis postupu pri implementácii vybraných častí aplikácie.

6.3.1 Konfigurácia databázového pripojenia

Pred samotnou implementáciou systému bolo potrebné zabezpečiť databázové pripojenie s štruktúrou odpovedajúcou reálnej databáze. Za týmto účelom bola vytvorená lokálna databáza PostgreSQL pomocou *dump* súboru reálnej databázy. Následne boli vytvorené príslušné modely, ktoré boli upravené tak aby zodpovedali entitám databázy. Jednalo sa o úpravu preddefinovaných konvencií Eloquent ORM ako napríklad: privátny kľúč tabuľky nesie názov *id*, názov tabuľky odpovedá názvu modelu v množnom čísle (tabuľke *parts* odpovedá model *Part*), tabuľky obsahujú stĺpce pre časové známky (*created_at*, *updated_at*). Keďže existujúca databáza neprihliada na tieto konvencie, boli potrebné vlastnosti explicitne definované v rámci modelu spolu s jednotlivými vzťahmi (one-to-many, many-to-one, a ďalšie).

6.3.2 Prihlásenie

Pre implementáciu prihlásenia ako aj ďalšej funkcionality spojenej s autentifikáciou bol použitý balík **laravel/ui**. Balík poskytuje pripravenú implementáciu autentifikácie, ktorá je dostupná spolu s frameworkom Bootstrap 4, vrátane pohľadov, riadiacich tried (angl. *controllers*) a príslušného smerovania. Následne bol pohľad (UI) dizajnovu upravený a prispôsobený. V rámci prihlásenia bol stanovený počet neúspešných pokusov o prihlásenie (päť), pričom po prekročení je prihlásenie zablokované po dobu piatich minút. Záznamy o neúspešných pokusov o prihlásenie sú ukladané na základe kombinácie IP adresy a e-mailu používateľa.

6.3.3 Používateľské role

K stanoveniu používateľských rolí bol vytvorený middleware⁸: *RoleMiddleware*, do ktorého ako parameter vstupuje používateľská rola. Hodnota parametra je porovnávaná s rolou používateľa. Následne bol tento middleware aplikovaný pre skupinu trás (angl. *routes*), ktoré vyžadujú autorizáciu – rolu používateľa administrátor ('middleware' => ['role:admin']). Takto implementovaný middleware je možné použiť aj pri rozšírení systému o ďalšie používateľské role, pričom stačí upraviť len hodnotu parametra (napr. role:super-admin).

6.3.4 Registrácia

Registrácia je rovnako, ako tomu bolo v prípade prihlásenia založená na použití balíka **laravel/ui**. Pripravená implementácia registrácie je základne nastavená, tak aby bolo vytvorenie účtu umožnené každému používateľovi.

V tomto prípade bolo potrebné riešenie upraviť, aby mal k vytvoreniu účtu prístup len autentifikovaný a autorizovaný používateľ (administrátor). Pre prehľadnosť kódu boli trasy určené pre registráciu explicitne definované a zahrnuté pod skupinu trás s prefixom *admin*. Pre tieto trasy bol zároveň priradený vyššie spomenutý middleware - *role:admin*, ktorý zabezpečuje autorizáciu. Úpravou prešli aj ďalšie prvky ako presmerovanie po registrácii a umiestnenie príslušného pohľadu.

Samotná registrácia bola upravená tak, že pri vytváraní účtu je vyžadované aby administrátor zadal meno, e-mailovú adresu a rolu používateľa. Heslo je vygenerované náhodne a následne zašifrované pomocou funkcie *bcrypt()*. V prípade, že registrácia

⁸ Vrstva pre kontrolu údajov na základe stanovených kritérií

prebehla úspešne (boli splnené všetky validačné kritéria) je na e-mailovú adresu registrovaného účtu odoslaný e-mail prostredníctvom triedy *Notification (RegistrationNotification)*, ktorý obsahuje odkaz pre nastavenie hesla k účtu. K nastaveniu hesla je následne použitá funkcia zabezpečujúca obnovu hesla. Po nastavení hesla je účet označený za aktivovaný.

6.3.5 Editácia profilu používateľa

V rámci editácie profilu je používateľovi umožnená zmena prihlasovacích údajov (heslo, e-mailová adresa). Pri zmene hesla používateľ zadáva súčasné heslo, nové heslo a potvrdenie hesla. Zmena hesla je úspešná v prípade, že bolo správne zadané aktuálne heslo, nové heslo je totožné s potvrdzovacím heslom a zároveň spĺňa stanovený *regex* (dĺžka hesla minimálne 8 znakov, heslo musí obsahovať aspoň jednu číslicu).

Pre zmenu e-mailovej adresy používateľ zadá novú adresu a heslo. Ak je zadané heslo platné a nová e-mailová adresa je unikátna v rámci používateľov systému, je na zadanú adresu odoslaný e-mail. Súčasťou e-mailu je „podpísaný“ odkaz (angl. *signed URL*⁹) s nastavenou dobou expirácie desať minút.

Po kliknutí na odkaz je používateľ presmerovaný späť na pohľad pre editáciu profilu. V prípade, že podpis je validný, používateľ je prihlásený a jeho *id* je zhodné s poskytnutým *id* odkazu a zároveň nevypršala expirácia, dôjde k zmene e-mailu o čom je používateľ informovaný.

6.3.6 Zobrazenie zoznamu súčiastok

Keďže v rámci zoznamu má byť používateľovi umožnené dynamické vyhľadávanie, je pre zobrazenie zoznamu implementovaná funkcia využívajúca asynchrónne volanie (AJAX). Táto funkcia (*showPartsList*) po zavolaní odošle žiadosť na server a odpoveď vo formáte JSON (JavaScript Object Notation) následne spracuje do záznamov tabuľky, ktorá reprezentuje daný zoznam.

Vyhľadávanie

Pre prípad vyhľadávania obsahuje spomínaná funkcia *showPartsList* parameter reprezentujúci hľadaný výraz (*searchQuery*). Pri každom zadaní znaku, ktorý predstavuje

⁹ URL obsahujúca podpis (*hash*), ktorý slúži k zabezpečeniu integrity

tlačový znak (reprezentujúci symbolom) je funkcia opätovne zavolaná a zoznam aktualizovaný.

Z pohľadu implementácie vyhľadávania na back-end časti sa jedná o „fulltext“ vyhľadávanie v rámci viacerých stĺpcov databázy. Pre tento prípad bol vytvorený nový príkaz *whereLike* registrovaný ako *Macro* pre *Laravel Query Builder*. Nová direktíva prijíma ako parameter hľadaný výraz a pole stĺpcov. Následne je výraz porovnávaný s údajmi stĺpcov prostredníctvom case-insensitive komparácie. Výsledná kolekcia je potom odoslaná na front-end.

Filtrácia

Zoznam elektronických súčiastok je zobrazovaný vo viacerých pohľadoch (stránkach) systému : pohľad zobrazujúci zoznam súčiastok na základe kategorizácie, pohľad obsahujúci zoznam súčiastok používaných v rámci zvoleného footprintu a pohľad obsahujúci zoznam súčiastok pre vybranú schému.

Pre zobrazenie zoznamu volaním funkcie *showPartsList* je teda potrebné, pridať kritérium, ktoré bude reprezentovať filter (kategória/footprint/schéma). Je taktiež potrebné zabezpečiť aby pri zmene hodnoty filtra (zvolenie inej kategórie) bol zoznam súčiastok súčasne filtrovaný aj na základe ďalších aktívnych kritérií (napr. hľadaný výraz). Z tohto dôvodu sú pre každý pohľad obsahujúci zoznam súčiastok definované globálne premenné reprezentujúce hodnotu daných kritérií. Základný filter (kategória/footprint/schéma) je možné pre jednotlivé pohľady definovať ako konštantu. Následne jednotlivé hodnoty vstupujú ako parametre funkcie *showPartsList*, čo zabezpečuje, že zobrazovaný zoznam spĺňa vždy všetky aktívne kritéria.

Takto navrhnuté filtrovanie zoznamu bolo následne rozšírené o **možnosť usporiadania zobrazovaných záznamov**. Zoznam je možné usporiadať abecedne a podľa počtu dostupných kusov a to zostupne alebo vzostupne. Keďže zoznam bude stránkovaný, je potrebné opätovne volať funkciu *showPartsList*, keďže nie je možné len zmeniť usporiadanie už zobrazených záznamov. Spôsob usporiadania si používateľ volí prostredníctvom rozbaľovacieho menu

Stránkovanie

Pre implementáciu stránkovania v rámci front-end časti je funkcia *showPartsList* rozšírená o parameter *page*. Týmto spôsobom je zabezpečené, že pri prechádzaní jednotlivých strán zostanú všetky stanovené filtre aktívne.

Pre odosielanie stránkovaných výsledkov zo serveru bola na filtrovanú kolekciu výsledkov aplikovaná metóda *paginate*, ktorá umožňuje prostredníctvom argumentu *perPage* definovať počet zobrazovaných záznamov na stránku. Táto vlastnosť je použitá k implementácii nastavenia zobrazovaného počtu záznamov. Funkcia *showPartsList* využíva premennú reprezentujúci počet záznamov (*perPage*), ktorá je použitá pri každom volaní funkcie a jej hodnota je odosielaná spolu s ostatnými filtrami. Pri zmene stránkovania prostredníctvom menu je táto hodnota zmenená a zároveň je opäť zavolaná funkcia *showPartsList* spolu s aktívnymi filtrami.

6.3.7 Zobrazenie detailu elektronickej súčiastky

Keďže je potrebné zabezpečiť prístup k detailu súčiastky na viacerých stránkach, je k implementácii využitý komponent šablónovacieho nástroja Blade. Komponent je následne vložený do jednotlivých pohľadov. Takto navrhnuté riešenie umožňuje lepšiu udržateľnosť kódu a tým jednoduchšie vykonávanie zmien.

V rámci komponentu bol samotný detail elektronickej súčiastky implementovaný ako modálne dialógové okno, ktoré je používateľovi zobrazené po kliknutí na konkrétny záznam zo zoznamu súčiastok. Parametre súčiastky sú zobrazované prostredníctvom formuláru, z dôvodu požiadavky na ich priamu editáciu.

Pred zobrazením okna je potrebné formulár naplniť údajmi príslušnej súčiastky. Pre tento účel slúži funkcia *showPartDetail*, ktorá ako parameter preberá *id* zvolenej súčiastky. Potom funkcia prostredníctvom asynchrónneho volania odošle žiadosť na server, kde je žiadosť spracovaná a následne je odoslaná odpoveď vo formáte JSON obsahujúca informácie o súčiastke. Po prijatí odpovede dochádza k parsovaniu údajov.

Z dôvodu hromadného priradenia (angl. *mass assignment*) hodnôt pri editácii má každý vstup formuláru nastavený atribút *name*, tak aby odpovedal príslušnému stĺpcu tabuľky. Vďaka tomu je možné v cykle prechádzať prijaté dáta a na základe kľúča (názvu stĺpca) je možné nastaviť potrebnú hodnotu vstupu, pričom jednotlivé elementy reprezentujúce vstup, sú vybrané na základe atribútu *name* (atribút *name* musí byť zhodný s hodnotou kľúča). Po úspešnom vykonaní všetkých uvedených operácií je používateľovi zobrazený detail. Nakoľko je pre získanie údajov použité asynchrónne volanie, tak v prípade, že by došlo k oneskoreniu, je používateľovi zobrazená animácia načítania až pokiaľ nedôjde k zobrazeniu detailu.

Editácia parametrov vybranej súčiastky

Jednotlivé informácie o súčiastke sú zobrazované prostredníctvom formuláru, pričom hodnoty vstupných polí a výberov reprezentujú hodnoty parametrov. Výberové polia boli prispôsobené prostredníctvom jQuery knižnice **Select2**, tak aby v nich bolo umožnené používateľovi dynamické vyhľadávanie. Bootstrap zabezpečuje základnú validáciu údajov pre front-end. Odoslané údaje sú rovnako podrobené validácií na strane servera.

Keďže je detail zobrazovaný v modálnom okne, je odosielanie formuláru vykonávané asynchrónne. Vďaka tomu okno zostáva otvorené a všetky zvolené filtre pre zoznam aktívne. V prípade, že nastane chyba pri editácii na strane servera (neplatný vstup pri validácii), je o tom používateľ informovaný, prostredníctvom upozornenia (obsahujúce príslušnú chybu), ktoré sa zobrazí vo vrchnej časti detailu. Po úspešnej editácii je na pozadí opätovne volaná funkcia *showPartsList*, ktorá prekreslí zobrazovaný zoznam, takže vykonané zmeny sa okamžite premietnu aj do zobrazovaného zoznamu súčiastok.

6.3.8 Zobrazenie zoznamu schém/footprintov

S cieľom zahrnúť možnosť dynamického vyhľadávania v zozname, je k implementácii zobrazenia zoznamu použitý rovnaký postup ako v prípade zoznamu elektronických súčiastok. V zozname schém a zozname footprintov však nie je potrebné definovať žiaden globálny filter.

Pre vykreslenie zoznamov sú osobitne definované funkcie. Tieto funkcie, rovnako ako funkcia *showPartsList* obsahujú argumenty reprezentujúce jednotlivé kritéria zobrazenia zoznamu. V tomto prípade sa jedná o hľadaný výraz a číslo strany zobrazovaných výsledkov. Keďže zoznamy obsahujú totožné údaje, uvedené funkcie vo svojom tele volajú spoločnú funkciu zabezpečujúcu zobrazenie, ktorá ako parameter okrem zmiených argumentov navyše preberá identifikátor, ktorý slúži pre rozlíšenie zoznamu schém a zoznamu footprintov. Takto navrhnuté riešenie je možné škálovať, napríklad v prípade, že by vznikla potreba separovať jednotlivé funkcie. Rovnako je možné vďaka identifikátoru stanoviť vykonanie rozdielnych operácií a zobraziť údaje v závislosti od typu zoznamu. Z dôvodu, že hodnoty niektorých údajov v databáze chýbali a pri prenose vo formáte JSON boli nahradené hodnotou *null*, bola vytvorená funkcia, ktorá príslušnú hodnotu späťne pretransformuje na prázdny reťazec.

6.3.9 Zobrazenie detailu schémy/footprintu

Detail footprintu/schémy je rovnako ako v prípade elektronickej súčiastky implementovaný s využitím komponentu (Blade). Detail je zobrazovaný v modálnom dialógovom okne. Komponent je využívaný z dôvodu, že schéma a footprint zdieľajú rovnaký dizajn ako aj rovnaké vlastnosti, ktoré sú v rámci detailu zobrazované: názov, popis, archivácia. Pre zabezpečenie škálovania komponentu obsahuje komponent možnosť prispôbiť jednotlivé časti s využitím slotov. V detaile je používateľovi umožnená editácia parametrov schémy/footprintu a zároveň má používateľ možnosť odstrániť príslušnú schému/footprint. Pred odstránením je potrebné schválenie akcie prostredníctvom potvrdzovacieho dialógového okna.

6.3.10 Správa používateľských účtov

Na správu používateľských účtov má prístup výhradne používateľ s rolou administrátor prostredníctvom odkazu v navigácii, ktorý nie je viditeľný pre používateľov s inou rolou. Trasy použité v súvislosti so správou účtov spadajú pod skupinu trás s prefixom */admin*, na ktoré je aplikovaný už vyššie popísaný middleware ('middleware' => ['role:admin']) zabezpečujúci autorizáciu. Administrátorovi je v rámci správy používateľských účtov sprístupnený náhľad na zoznam registrovaných účtov. Administrátor má možnosť odstrániť vybraný účet zo zoznamu. Odstráneniu predchádza schválenie potvrdzovacieho dialógu.

6.3.11 Import údajov súčiastky z API výrobcu Farnell

Import údajov je vykonaný na základe zadaného výrobného čísla elektronickej súčiastky. Na tento účel slúži funkcia *importPartData*, ktorá je k dispozícii v detaile súčiastky a zároveň na stránke určenej k registrácii nových elektronických súčiastok. Funkcia ako parameter preberá výrobné číslo, ktoré je následne prostredníctvom asynchrónneho volania odoslané na server. Server potom prostredníctvom HTTP (Hypertext Transfer Protocol) klienta *GuzzleHttp* odošle žiadosť na adresu API výrobcu Farnell.

V prípade, že bolo poskytnuté správne výrobné číslo, sú dáta odoslané z API výrobcu na serveri deserializované a spracované do požadovanej formy. Následne sú dáta odoslané späť na front-end ako odpoveď vo formáte JSON. Dáta sú ďalej vo funkcií *importPartData* parsované a vo formulári sú pre jednotlivé vstupy priradené príslušné hodnoty.

Serverová časť v tomto prípade slúži ako proxy server. Je to z dôvodu implementovanej politiky rovnakého pôvodu (angl. *same origin policy*) vo webovom

prehliadači, ktorá slúži pre zabránenie *cross site request forgery* útoku. Tá povoľuje spojenie iba v prípade, že lokalita klientskeho rozhrania je zhodná s lokalitou uvedenou v *Access-Control-Allow-Origin* odpovedi servera, ktorá špecifikuje, ktoré lokality môžu pristupovať k jeho zdrojom. V prípade, že sa lokalita nezhoduje s uvedenými lokalitami, prehliadač zablokuje požiadavku s chybou politiky CORS (Cross-Origin Resource Sharing). Táto kontrola sa vzťahuje na komunikáciu medzi prehliadačom a serverom. Uvedená bezpečnostná politika nie je aplikovaná na komunikáciu medzi dvoma servermi, preto je možné využiť serverovú časť ako proxy server pre komunikáciu s API výrobcu.

6.3.12 Správa dodávateľov elektronických súčiastok

Správa dodávateľov zahŕňa zobrazenie zoznamu dodávateľov, pridanie dodávateľa a odstránenie dodávateľa. Všetky uvedené funkcie sú k dispozícii v rámci jedného modálneho dialógového okna. K zabezpečeniu potrebnej „reaktivity“ je dialógové okno implementované prostredníctvom frameworku **Vue.js**, konkrétne s využitím technológie **BootstrapVue** (komponent *b-modal*). Používateľ si zobrazí dialógové okno kliknutím na označené tlačidlo umiestnené v detaile elektronickej súčiastky pri výberovom poli s príslušnými dodávateľmi.

Pred otvorením dialógového okna je prostredníctvom asynchrónneho volania (knihnica **axios**) odoslaná žiadosť pre získanie aktuálneho zoznamu evidovaných dodávateľov. Následne je zoznam vykreslený vo forme tabuľky a používateľovi je zobrazené dialógové okno.

Pridanie nového dodávateľa

Pridanie dodávateľa je umožnené používateľovi cez vstupné pole. V prípade, že pridanie prebehlo úspešne, používateľ môže okamžite pozorovať zmenu v zobrazenom zozname. Zároveň je príslušný dodávateľ pridaný do výberového poľa dodávateľov v detaile elektronickej súčiastky.

Odstránenie dodávateľa

Používateľ má možnosť odstrániť dodávateľa kliknutím na určené tlačidlo zobrazené v tabuľke. Pred odstránením je používateľ upozornený a vyzvaný k potvrdeniu akcie. Vykonané zmeny sú ihneď premietnuté do zobrazovaného zoznamu. Odstránený dodávateľ je následne odobraný aj z možností výberového poľa dodávateľov v detaile súčiastky.

V prípade elektronických súčiastok, kde odstránený dodávateľ predstavoval zvoleného dodávateľa, je na miesto dodávateľa automaticky priradená prednastavená hodnota: *None*.

6.3.13 Podpora zmeny jazyka

Výsledná aplikácia obsahuje podporu pre zmenu jazyka zobrazenia. Jazyk aplikácie je možné nastaviť prostredníctvom výberového poľa na stránke pre prihlásenie alebo v editácii profilu. Po výbere jazyka je jeho príslušná hodnota uložená do *session* premennej. Na všetky trasy systému je aplikovaný middleware *locale*, ktorý slúži na detekciu hodnoty jazyka v premennej *session*. V prípade, že došlo k nastaveniu jazyka, je lokalizácia aplikácie zmenená prostredníctvom funkcie *setLocale* na hodnotu zo *session*. Je to z dôvodu, že nastavenie jazyka nie je perzistentné, je uložené len pre konkrétny beh (požiadavku). Pri prechode medzi stránkami je potrebné ju explicitne definovať, k čomu slúži uvedený middleware.

Texty v pohľadoch sú syntakticky uvedené vo forme prekladových reťazcoch, tie zároveň predstavujú kľúč pre preklad, ktorý je definovaný v JSON súboroch pre príslušný jazyk (ako napr.: {"Cancel": "Zrušiť" }). V prípade, že by niektoré z uvedených slov bolo potrebné zobrazit' vo viacerých významoch je možné použiť kľúč, ktorý má príslušný text pre jednotlivé jazyky definovaný v rámci PHP súborov pre podporované jazyky.

7 TESTOVACIE NASADENIE

Po kompletizácii príslušných komponentov vyvíjaného systému bolo výsledné riešenie nasadené prostredníctvom služby Elastic Beanstalk na cloud platformu Amazon Web Services (Amazon EC2). V rámci inštancie Amazon EC2 je využívaný server NGINX. Nasledujúce obrázky (Obr. 17 – Obr. 30) zobrazujú jednotlivé časti testovacieho nasadenia.

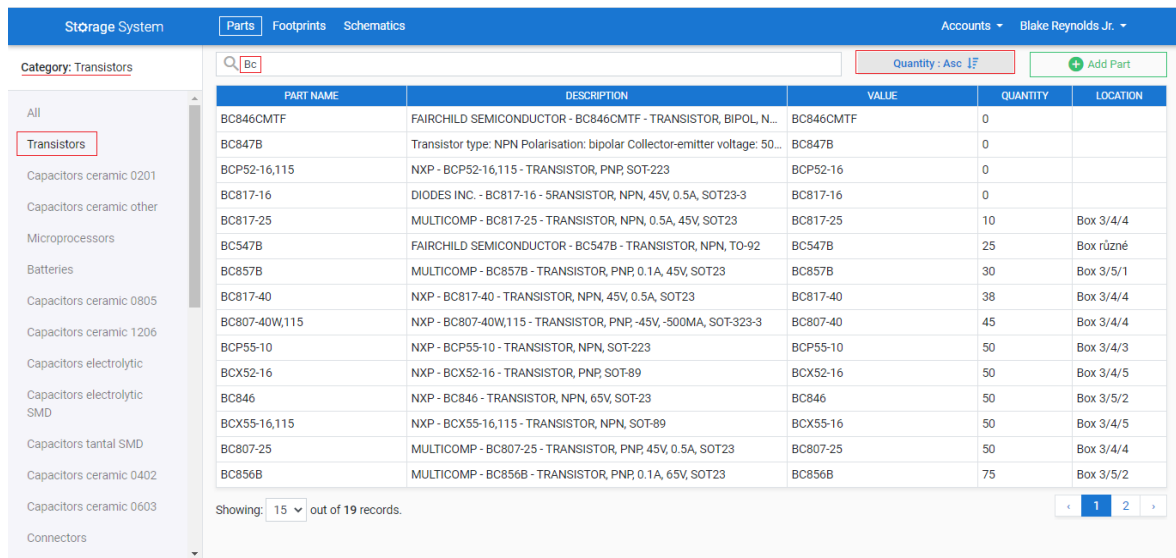
Na obrázku (Obr. 17) je možné vidieť formulár slúžiaci k autentifikácii. V prípade úspešného prihlásenia je používateľ presmerovaný na pohľad obsahujúci zoznam všetkých evidovaných elektronických súčiastok (Obr. 18).

Obr. 17. Prihlasovací formulár (výsledný systém)

PART NAME	DESCRIPTION	VALUE	QUANTITY	LOCATION
NCP5501DT33G	Typ Výstupu: Pevný Vstupní Napětí Min: - Vstupní Napětí Max: 18V Pe...	3V3	0	
0.01uF / 630V	EPCOS - B32529C8103J189 - CAP, FILM, PET, 10NF, 630VDC, RAD	0.01uF / 630V		
0.1uF/35V		0,1u/35V	33	Box 4/8/2
0.22u/35V		0.22u/35V	35	Box 4/8/2
0.33u/35V		0.33u/35V	35	Box 4/8/2
0.47u/35V		0.47u/35V	35	Box 4/8/2
0.9pF/50V	0.90pF 50V Ceramic Capacitor COG, NP0 0402 (1005 Metric) 0.039" L ...	0.9pF/50V	0	
0402B104K250CT	WALSIN - 0402B104K250CT - CAPACITOR, MLCC, X7R, 0.1UF, 25V, 0402	100nF 25V	0	
0402N270J500CT	WALSIN - 0402N270J500CT - CAPACITOR, MLCC, NP0, 27PF, 50V, 0402	27pF 50V	0	
0603A101JAT2A				
0603B104K160CT	WALSIN - 0603B104K160CT - CAPACITOR, MLCC, X7R, 0.1UF, 16V, 0603	100nF / 16V	0	
0603B332J500CT	WALSIN - 0603B332J500CT - CAPACITOR, MLCC, X7R, 3300PF, 50V, 0...	3.3nF 50V		
0603B473K500CT	WALSIN - 0603B473K500CT - CAPACITOR, MLCC, X7R, 0.047UF, 50V, 0...	47nF / 50V	0	
0603F474Z160CT	WALSIN - 0603F474Z160CT - CAPACITOR, MLCC, Y5V, 0.47UF, 16V, 06...	0.47uF 16V	0	
0603N1R8C500CT	WALSIN - 0603N1R8C500CT - CAP, MLCC, NP0, 1.8PF, 50V, 0603, REEL	1.8p 50V	0	

Obr. 18. Zobrazenie zoznamu elektronických súčiastok (výsledný systém)

Zoznam je zobrazovaný v závislosti od kritérií špecifikovaných v predošlej kapitole (kategória, typ usporiadania – abecedne/podľa kvantity, hľadaný výraz, číslo strany, počet zobrazovaných výsledkov na stránku). Na obrázku (Obr. 19) je ukážka zoznamu filtrovaného na základe hľadaného výrazu a zvolenej kategórie, ktorý je usporiadaný vzostupne podľa počtu kusov. Jednotlivé filtre sú vyznačené červenou farbou.



PART NAME	DESCRIPTION	VALUE	QUANTITY	LOCATION
BC846CMTF	FAIRCHILD SEMICONDUCTOR - BC846CMTF - TRANSISTOR, BIPOL, N...	BC846CMTF	0	
BC847B	Transistor type: NPN Polarisation: bipolar Collector-emitter voltage: 50...	BC847B	0	
BCP52-16,115	NXP - BCP52-16,115 - TRANSISTOR, PNP, SOT-223	BCP52-16	0	
BC817-16	DIODES INC. - BC817-16 - 5RANSISTOR, NPN, 45V, 0.5A, SOT23-3	BC817-16	0	
BC817-25	MULTICOMP - BC817-25 - TRANSISTOR, NPN, 0.5A, 45V, SOT23	BC817-25	10	Box 3/4/4
BC547B	FAIRCHILD SEMICONDUCTOR - BC547B - TRANSISTOR, NPN, TO-92	BC547B	25	Box různé
BC857B	MULTICOMP - BC857B - TRANSISTOR, PNP, 0.1A, 45V, SOT23	BC857B	30	Box 3/5/1
BC817-40	NXP - BC817-40 - TRANSISTOR, NPN, 45V, 0.5A, SOT23	BC817-40	38	Box 3/4/4
BC807-40W,115	NXP - BC807-40W,115 - TRANSISTOR, PNP, -45V, -500MA, SOT-323-3	BC807-40	45	Box 3/4/4
BCP55-10	NXP - BCP55-10 - TRANSISTOR, NPN, SOT-223	BCP55-10	50	Box 3/4/3
BCX52-16	NXP - BCX52-16 - TRANSISTOR, PNP, SOT-89	BCX52-16	50	Box 3/4/5
BC846	NXP - BC846 - TRANSISTOR, NPN, 65V, SOT-23	BC846	50	Box 3/5/2
BCX55-16,115	NXP - BCX55-16,115 - TRANSISTOR, NPN, SOT-89	BCX55-16	50	Box 3/4/5
BC807-25	MULTICOMP - BC807-25 - TRANSISTOR, PNP, 45V, 0.5A, SOT23	BC807-25	50	Box 3/4/4
BC856B	MULTICOMP - BC856B - TRANSISTOR, PNP, 0.1A, 65V, SOT23	BC856B	75	Box 3/5/2

Obr. 19. Náhľad na filtrovaný katalóg elektronických súčiastok (výsledný systém)

Po zvolení záznamu (elektronickej súčiastky) zo zoznamu je používateľovi prostredníctvom modálneho dialógového okna zobrazený odpovedajúci detail (Obr. 20). V rámci detailu má používateľ prístup k jednotlivým údajom súčiastky s možnosťou ich editácie. Po zadaní výrobného čísla elektronickej súčiastky (zvýraznená časť – Obr. 20) je možné importovať údaje z programového rozhrania výrobcu Farnell. V detaile súčiastky je k dispozícii dialógové okno pre správu dodávateľov (Obr. 21). To je zobrazené po kliknutí na tlačidlo umiestnené vedľa výberového menu s dodávateľmi (zvýraznená časť – Obr. 21).

PART INFORMATION

NAME: OR33 - 5% VALUE: OR33 COUNT: 150 LOCATION: Box 6/2/2

DESCRIPTION: BOURNS - CRL0805-JW-R330ELF - RESISTOR, 0805, R33, 5%, 0.125W

CATEGORY: Resistors 0805

SCHEMA: UniversalResistor0805

FOOTPRINT: R0805

IMAGE LINK: <http://cz.farnell.com/productimages/farr>

DATASHEET: <http://www.farnell.com/datasheets/>

PART SUPPLIER

SUPPLIER: Farnell CODE: 2008299 ORDER LINK: <http://cz.farnell.com/2008299?Ntt=2008299> SZ: RC:

Update Remove

Obr. 20. Detail elektronickej súčiastky (výsledný systém)

SUPPLIER MANAGEMENT

SEARCH: + Add

#	NAME	REMOVE
1	Ecom	
2	Farnell	
3	EZK	
4	GME	
5	unknown	
6	TME	
7	SOS Electronic	
8	UTB	
9	Official electronic	
10	M12 Lenses	
11	None	
12	Arrow	

Obr. 21. Dialógové okno pre správu dodávateľov (výsledný systém)

K zaevidovaniu novej elektronickej súčiastky do systému slúži samostatná stránka. Na obrázku (Obr. 22) je ukážka využitia importu dát z API výrobcu Farnell do formuláru, ktorý slúži k pridaniu súčiastky. Formulár s importovanými údajmi je zobrazený na nasledujúcom obrázku (Obr. 23). Na obrázku (Obr. 23) je zároveň možné vidieť dynamické vyhľadávanie v rámci menu s kategóriami.

Obr. 22. Zobrazenie prebiehajúceho importu údajov z API výrobcu Farnell (výsledný systém)

Obr. 23. Formulár pre zaevidovanie novej elektronickej súčiastky

Ďalšou časťou výsledného systému je pohľad na zoznam používaných schém a pohľad na zoznam používaných footprintov. Zoznam používaných schém spolu s elektronickejšími súčiastkami využívajúcimi zvolenú schému je zobrazený na obrázku (Obr. 23). Detail zvolenej schémy je možné vidieť na obrázku (Obr. 24). K registrácii novej schémy slúži formulár, ktorý je umiestnený na osobitnej stránke (Obr. 25).

NAME	DESCRIPTION	PATH: LIBRARY	ARCHIVED	#	NAME	DESCRIPTION
TestSchematic		Resistors.SchLib	false	1	2.2u/25V	
ADP2302			false	2	22u/20V	
ADP2302		VoltageRegulators.SchLib	false	3	47u/16V	KEMET - T491C476K016AT - C...
UniversalCapacitor		Capacitors.SchLib	false	4	33u/10V	
UniversalCapacitorPol		Capacitors.SchLib	false	5	2.2uF/35V	KEMET - T491A225K035AT - C...
UniversalCapacitorPol		Capacitors.SchLib	false	6	15u/16V	
RJ-45_2X4		Connectors.SchLib	false	7	EEETG1H101UP	PANASONIC ELECTRONIC CO...
Header3W3_5		Connectors.SchLib	false	8	4.7u/16V	VISHAY SPRAGUE - 293D475X...
HDR2X5-lock		Connectors.SchLib	false	9	1uF/16V	NP16V105M5X11 - CAP, ALU E...
HDR2x5male		Connectors.SchLib	false	10	CE 2200u/35VT JAM-TK 16x2...	CE 2200u/35VT JAM-TK 16x2...
HDR2x20male		Connectors.SchLib	false	11	100uF/35V	
MUSB		Connectors.SchLib	false	12	47000UF, 5.5V	Supercapacitors
HDR1x2male		Connectors.SchLib	false	13	3.3u/16V	
None			false	14	47u/6.3V	
MOSFETND		Transistors.SchLib	false	15	0.33u/35V	

Obr. 24. Náhled na zoznam používaných schém a katalóg elektronických súčiastok v závislosti od zvolenej schémy (výsledný systém)

SCHEMA DETAIL

NAME:

DESCRIPTION:

LIBRARY PATH:

ARCHIVED
Not usable on new designs

Obr. 25. Detail schémy (výsledný systém)

The screenshot shows a web interface for adding a new schema. At the top, there are tabs for 'Footprints' and 'Schematics'. Below the tabs is a header 'ADD NEW SCHEMA'. The form contains the following elements:

- NAME:** A text input field.
- DESCRIPTION:** A larger text area with a diagonal resize handle in the bottom right corner.
- LIBRARY PATH:** A text input field.
- ARCHIVED**
Not usable on new designs
- + Add Schema** button

Obr. 26. Formulár pre registráciu novej schémy (výsledný systém)

Nezávisle od pridelenej role má používateľ k dispozícii náhľad na profil (Obr. 27) a editáciu prihlasovacích údajov (Obr. 28). V prípade používateľa s rolou administrátor je poskytovaná funkcionálna rozšírená o riadenie používateľských účtov – správa (Obr. 29), registrácia.

The screenshot shows a user profile page. At the top, there are tabs for 'Schematics', 'Accounts', and 'Blake Reynolds Jr.'. The main content area displays the following information:

- Account:** Blake Reynolds Jr.
- E-mail:** admin@example.com
- Role:** admin
- Registered:** 2020-03-04 18:32:31
- Settings:** [Edit Profile](#) [Delete Account](#)
- Last updated: 2020-03-15 12:44:53

At the bottom left, there is a 'Locale: English' dropdown menu. On the right side, there is a user menu with the following options:

- Profile
- Edit Profile
- Logout

Obr. 27. Profil používateľa (výsledný systém)

The image shows two side-by-side forms for account management. The left form, titled 'Change Password', contains three input fields: 'Password', 'New Password', and 'New Password Confirm'. Below the 'New Password' field, there is a note: 'The minimum password length is 8 characters. Password must contain at least one number.' A blue 'Change Password' button is at the bottom. The right form, titled 'Change E-mail', shows the current email address as 'admin@example.com'. It has two input fields: 'New E-Mail' and 'Password'. A note below the 'Password' field states: 'NOTE: A confirmation message will be sent to your new e-mail address.' A blue 'Change E-mail' button is at the bottom. The top navigation bar includes 'Parts', 'Footprints', 'Schematics', and 'Accounts'.

Obr. 28. Editácia prístupových údajov (výsledný systém)

The image shows a user management interface. At the top, there is a navigation bar with 'Storage System', 'Parts', 'Footprints', 'Schematics', 'Accounts', and 'Blake Reynolds Jr.'. A green notification bar at the top says 'Account was successfully removed.' Below this is a table with columns: Username, E-mail Address, Role, Verified, and a delete icon. There are two rows of data. A '+ Create Account' button is in the top right corner of the table area.

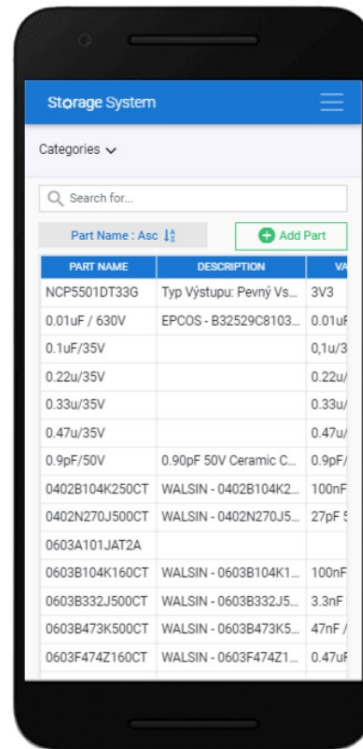
Username	E-mail Address	Role	Verified	
Blake Reynolds Jr.	admin@example.com	admin	Yes	
test	test@example.com	user	No	

Obr. 29. Správa používateľských kont (výsledný systém)

Všetky časti systému boli implementované v súlade s princípmi responzívneho dizajnu. Rozloženie elementov vybraných stránok systému na mobilnom zariadení je možné vidieť na obrázku (Obr. 30).



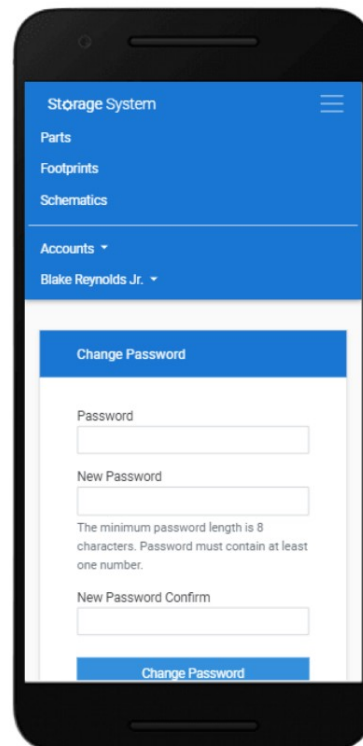
(1) Zobrazenie detailu elektronickej súčiastky



(2) Náhľad na katalóg elektronickej súčiastky



(3) Detail elektronickej súčiastky (pokračovanie)



(4) Ukážka navigačného menu

Obr. 30. Ukážka zobrazenia výsledného systému na mobilnom zariadení

ZÁVER

Diplomová práca bola koncipovaná v súlade s definovanými zásadami pre vypracovanie. Všetky stanovené body uvedené v zadaní boli dodržané. Výstupom praktickej časti diplomovej práce je systém, ktorý poskytuje funkcie v závislosti od špecifikovaných požiadaviek. Jednotlivé časti systému boli implementované za použitia moderných technológií (Laravel, Vue.js a ďalšie).

Výsledný systém zapuzdruje funkcionalitu, ktorá umožňuje efektívne riadenie a organizáciu skladových zásob embedded laboratória. Vo všeobecnosti sa jedná o kľúčové funkcie: zobrazenie zoznamu evidovaných elektronických súčiastok na základe zvolených kritérií, náhľad na detailné informácie príslušnej súčiastky a ich editácia, zoznamy používaných schém a footprintov, riadenie jednotlivých zoznamov, kooperácia s nástrojom pre návrh dosiek plošných spojov Altium Designer, komunikácia s programovým rozhraním výrobcu súčiastok Farnell slúžiaca k importu dát. Jednotlivé funkcie sú podrobne popísané v rámci praktickej časti práce.

V súvislosti s používateľskými kontami a prístupom k službe, systém umožňuje autentifikáciu, obnovu hesla, registráciu, manažment účtov a editáciu prístupových údajov. Z pohľadu autorizácie sú používateľom pridelené role, ktoré špecifikujú prístupové práva k jednotlivým funkciám. Vďaka dostupnosti online, prostredníctvom webového rozhrania je systém k dispozícii na širokej palete zariadení, ihneď pripravený k použitiu. Systém disponuje responzívnym dizajnom s cieľom skvalitnenia poskytovaných služieb pri použití odlišných zobrazovacích jednotiek. K zabezpečeniu používateľsky prívetivejšieho rozhrania obsahuje systém podporu pre zmenu jazyka.

Skompletizovaný systém bol za účelom poskytnutia skúšobného obdobia nasadený na cloud platformu Amazon Web Services. Následne je možné systém presunúť na iný server alebo bez dodatočných úprav, vďaka kompatibilnej databázovej štruktúre, pripojiť k súčasnému databázovému systému a začať ihneď s jeho používaním.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] How to Manage Inventory and BOMs When Producing Electronic Boards. *Parts-in-Place* [online]. [cit. 2020-01-10]. Dostupné na internete: <http://partsinplace.com/user-guide/general-concepts/>
- [2] PartsBox User's Guide (PartsBox). *PartsBox* [online]. [cit. 2020-01-11]. Dostupné na internete: <https://partsbox.com/users-guide.html>
- [3] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [4] Business požiadavky (Business Requirements). *ManagementMania.com* [online]. [cit. 2020-01-25]. Dostupné na internete: <https://managementmania.com/sk/business-poziadavky-business-requirements>
- [5] PARKER, John. Business, User, and System Requirements. *Enfocus Solutions* [online]. 18. február 2012 [cit. 2020-02-03]. Dostupné na internete: <https://enfocussolutions.com/business-user-and-system-requirements/>
- [6] PILONE, Dan a Neil PITMAN. *Uml 2.0 in a Nutshell*. 2nd edition. Beijing; Sebastopol, CA: O'Reilly Media, 2005. ISBN 978-0-596-00795-9.
- [7] *UML - Use Case Diagrams - Tutorialspoint* [online]. [cit. 2020-02-10]. Dostupné na internete: https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm
- [8] Use Case Diagram Relationships Explained with Examples. *Creately Blog* [online]. 17. február 2015 [cit. 2020-02-23]. Dostupné na internete: <https://creately.com/blog/diagrams/use-case-diagram-relationships/>
- [9] What is a Brute Force Attack? - Definition from Techopedia. *Techopedia.com* [online]. [cit. 2020-02-25]. Dostupné na internete: <https://www.techopedia.com/definition/18091/brute-force-attack>
- [10] SHANKDHAR, Pavitra. Popular Tools for Brute-force Attacks [Updated for 2019]. *Infosec Resources* [online]. 12. február 2019 [cit. 2020-02-25]. Dostupné na internete: <https://resources.infosecinstitute.com/popular-tools-for-brute-force-attacks/>
- [11] Rainbow tables. *IONOS Digitalguide* [online]. [cit. 2020-03-12]. Dostupné na internete: <https://www.ionos.com/digitalguide/server/security/rainbow-tables/>
- [12] MERIDA, Oscar a kol. *Web Security 2016*. Alexandria, VA.: php[architect], 2016. ISBN 978-1-940111-41-4.
- [13] What is SQL Injection (SQLi) and How to Prevent It. *Acunetix* [online]. [cit. 2020-03-16]. Dostupné na internete: <https://www.acunetix.com/websitesecurity/sql-injection/>
- [14] What is Cross-site Scripting and How Can You Fix it? *Acunetix* [online]. [cit. 2020-03-16]. Dostupné na internete: <https://www.acunetix.com/websitesecurity/cross-site-scripting/>

- [15] Cross-Site Request Forgery Attacks. *Netsparker* [online]. 15. apríl 2020 [cit. 2020-03-20]. Dostupné na internete: <https://www.netsparker.com/blog/web-security/csrf-cross-site-request-forgery/>
- [16] MANNINO, Jack. *OWASP Top 10 2007-2017: The Fall of CSRF* [online]. [cit. 2020-03-20]. Dostupné na internete: <https://nvisium.com/blog/2017/11/30/owasp-top-10-2007-2017-the-fall-of-csrf.html>
- [17] *What is PHP? Write your first PHP Program* [online]. [cit. 2020-03-29]. Dostupné na internete: <https://www.guru99.com/what-is-php-first-php-program.html>
- [18] SKLAR, David. *PHP 7: praktický průvodce nejrozšířenějším skriptovacím jazykem pro web*. Preložil Jan POKORNÝ. Brno: Zoner Press, 2018. Encyklopedie Zoner Press. ISBN 978-80-7413-363-3.
- [19] REYES, Joel. Discussing PHP Frameworks: What, When, Why and Which? *Jotform Blog* [online]. [cit. 2020-04-02]. Dostupné na internete: <https://www.jotform.com/blog/discussing-php-frameworks/>
- [20] KUMAR, Saket. MVC Design Pattern. *GeeksforGeeks* [online]. 18. august 2017 [cit. 2020-04-04]. Dostupné na internete: <https://www.geeksforgeeks.org/mvc-design-pattern/>
- [21] *Overview of Model-View-Controller (MVC) – Patrick’s Software Blog* [online]. [cit. 2020-04-04]. Dostupné na internete: <https://www.patricksoftwareblog.com/overview-of-model-view-controller-mvc/>
- [22] SHARMA, Rashmi. 8 Popular PHP Frameworks For Web Development in 2020. *Hacker Noon* [online]. [cit. 2020-04-06]. Dostupné na internete: <https://hackernoon.com/8-popular-php-frameworks-for-web-development-in-2020-od3f38ez>
- [23] *Laravel vs Symfony in 2020 – which framework choose for your project?* [online]. 16. august 2019 [cit. 2020-04-07]. Dostupné na internete: <https://asperbrothers.com/blog/laravel-vs-symfony/>
- [24] STAUFFER, Matt. *Laravel: Up and Running*. 1st edition. Sebastopol, CA: O’Reilly Media, Inc., 2016. ISBN 978-1-4919-3608-5.
- [25] FRONCZAK, Sylvia. Laravel vs Symfony: PHP Framework Comparison. *Stackify* [online]. 23. marec 2019 [cit. 2020-04-07]. Dostupné na internete: <https://stackify.com/laravel-vs-symfony-php-framework-comparison/>
- [26] *Introduction — Vue.js* [online]. [cit. 2020-04-15]. Dostupné na internete: <https://vuejs.org/v2/guide/>
- [27] Client-Server Model. *GeeksforGeeks* [online]. 23. október 2019 [cit. 2020-04-21]. Dostupné na internete: <https://www.geeksforgeeks.org/client-server-model/>
- [28] What is the Client-Server Model? - Definition from Techopedia. *Techopedia.com* [online]. [cit. 2020-04-21]. Dostupné na internete: <https://www.techopedia.com/definition/18321/client-server-model>

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

BOM	Bills of Materials
API	Application Programming Interface
UML	Unified Modeling Language
SQL	Structured Query Language
CSRF	Cross-Site Request Forgery
OWASP	Open Web Application Security Project
PHP	PHP: Hypertext Preprocessor
DBMS	Database Management System
MVC	Model-View-Controller
UI	User Interface
npm	Node Package Manager
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
ORM	Object-Relational Mapping
MPA	Multi-Page Applications
Sass	Syntactically Awesome Style Sheets
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
CORS	Cross-Origin Resource Sharing

ZOZNAM OBRÁZKOV

Obr. 1. Ukážka zobrazenia elektronických súčiastok skladom – systém <i>Parts-in-Place</i>	13
Obr. 2. Detail elektronickej súčiastky – systém <i>Parts-in-Place</i>	14
Obr. 3. Katalóg elektronických komponentov – systém <i>PartsBox</i>	15
Obr. 4. Detail elektronickej súčiastky – systém <i>PartsBox</i>	16
Obr. 5. Zobrazenie zoznamu elektronických súčiastok – systém <i>Regent</i>	19
Obr. 6. Modálne okno reprezentujúce detail súčiastky – systém <i>Regent</i>	20
Obr. 7. Zoznam footprintov a súčiastok používajúcich zvolený footprint	21
Obr. 8. Detail schémy – systém <i>Regent</i>	21
Obr. 9. Delenie požiadaviek podľa úrovne popisu	23
Obr. 10. Vzorový diagram prípadov použitia obsahujúci popis komponentov	27
Obr. 11. Softvérová architektúra MVC návrhového vzoru [21].....	32
Obr. 12. Funkčné požiadavky spojené s autentifikáciou a používateľskými účtami (<i>use case diagram</i>).....	42
Obr. 13. Funkčné požiadavky na vyvíjaný systém (<i>use case diagram</i>)	43
Obr. 14. Model klient – server	45
Obr. 15. Architektúra Laravel aplikácie	46
Obr. 16. Schéma databázy	48
Obr. 17. Prihlasovací formulár (výsledný systém)	57
Obr. 18. Zobrazenie zoznamu elektronických súčiastok (výsledný systém)	57
Obr. 19. Náhľad na filtrovaný katalóg elektronických súčiastok (výsledný systém).....	58
Obr. 20. Detail elektronickej súčiastky (výsledný systém)	59
Obr. 21. Dialógové okno pre správu dodávateľov (výsledný systém).....	59
Obr. 22. Zobrazenie prebiehajúceho importu údajov z API výrobcu Farnell (výsledný systém).....	60
Obr. 23. Formulár pre zaevidovanie novej elektronickej súčiastky	60
Obr. 24. Náhľad na zoznam používaných schém a katalóg elektronických súčiastok v závislosti od zvolenej schémy (výsledný systém)	61
Obr. 25. Detail schémy (výsledný systém)	61
Obr. 26. Formulár pre registráciu novej schémy (výsledný systém)	62
Obr. 27. Profil používateľa (výsledný systém).....	62
Obr. 28. Editácia prístupových údajov (výsledný systém)	63
Obr. 29. Správa používateľských kont (výsledný systém)	63
Obr. 30. Ukážka zobrazenia výsledného systému na mobilnom zariadení	64

ZOZNAM TABULIEK

Tab. 1. Databázová podpora (Laravel, Symfony)[25]	35
Tab. 2. Zoznam útokov a poskytnutých opatrení (Laravel).....	39

ZOZNAM PRÍLOH

Príloha P I: CD nosič

PRÍLOHA P I: CD NOSIČ

Elektronické dáta vzťahujúce sa k diplomovej práci sú k dispozícii v rámci CD prílohy, ktorá obsahuje:

- diplomovú prácu vo formáte PDF,
- zdrojové kódy praktickej časti práce.