

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

**SYSTÉM ELEKTRONICKÉHO OBCHODOVANIA V PROSTREDÍ
SOCIÁLNYCH SIETÍ S INTERAKTÍVNOU PODPOROU TVORBY
PRODUKTOV**

Diplomová práca

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

**SYSTÉM ELEKTRONICKÉHO OBCHODOVANIA
V PROSTREDÍ SOCIÁLNYCH SIETÍ S INTERAKTÍVNOU
PODPOROU TVORBY PRODUKTOV**

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	9.2.1 Informatika
Školiace pracovisko:	Katedra počítačov a informatiky (KPI)
Školiteľ:	doc. Ing. Martin Tomášek, PhD.

Abstrakt v SJ

Primárnym cieľom diplomovej práce je implementácia knižníc pre elektronické biznis riešenia s ohľadom na interaktívnu podporu úpravy produktov a ich následný predaj ako aj šírenie ich obrazových reprezentácií ako reklamy v prostredí sociálnych sietí. V práci autor analyzuje existujúce riešenia a navrhuje vlastné riešenie pokrývajúce stanovené požiadavky počnúc interaktívnym prístupom v oblasti customizácie produktu cez implementáciu medzinárodnej platobnej metódy PayPal až po oEmbed integráciu so sociálnou sieťou pre zdieľanie obrazových dát Pinterest. V závere autor poukazuje na dosiahnuté výhody v zmysle efektívnejšej a voľnejšej práce s komponentmi pri návrhu produktov, menšiu obmedzenosť, obohatenie systému o možnosť pridania vlastných komponentov oproti existujúcim systémom.

Kľúčové slová

Interaktivita, customizácia, komponentové skladanie, elektronický obchod, platobné metódy, PayPal, Pinterest, HTML5, Canvas, SVG

Abstrakt v AJ

The primary aim of the Diploma's thesis is an implementation of libraries for electronic business solutions considering interactive aid of product adjustment and subsequent sale, as well as, a spread of their graphic representations as an advertisement in social networks environment. In the paper, author analyzes existing equivalents and designs own solution covering all defined requirements starting with product customization interactive approach through PayPal the international payment method implementation up to Pinterest social network for sharing graphic data integration. The stone of the thesis is closed by the summary of achieved benefits related to more efficient and freer components manipulation throughout products design, smaller limitations, own components addition possibility compared to the existing systems.

Kľúčové slová v AJ

Interactivity, customization, components composition, eShop, payment methods, PayPal, Pinterest, HTML5, Canvas, SVG

Zadanie práce

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra počítačov a informatiky

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: 9.2.1 Informatika

Študijný program: Informatika

Názov práce:

**Systém elektronického obchodovania v prostredí sociálnych sietí s
interaktívnou podporou tvorby produktov**
E-Commerce System in the Social Networks Environment with the Interactive
Support of the Products Creation

Študent:

Bc. Pavol Daňo

Školiteľ:

doc. Ing. Martin Tomášek, PhD.

Školiace pracovisko:

Katedra počítačov a informatiky

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Vypracovať analýzu súčasného stavu v oblastiach univerzálnych grafických používateľských rozhraní, integrácii sociálnych sietí a realizácii elektronických platieb.
2. Špecifikovať požiadavky pre interaktívny systém vytvárania produktov a pre realizáciu elektronického obchodovania v prostredí sociálnych sietí.
3. Navrhnuť systém elektronického obchodovania s podporou interaktívnej tvorby produktov, s prepojením na sociálne siete ako zákaznickej bázy a s realizáciu elektronických platieb.
4. Implementovať navrhnuté riešenie a overiť jeho použitie na vybranom podnikateľskom pláne.
5. Vypracovať technickú dokumentáciu.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 02.05.2014

Dátum zadania diplomovej práce: 31.10.2013


doc. Ing. Jaroslav Porubán, PhD.
vedúci garantujúceho pracoviska




prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval samostatne s použitím odbornej literatúry uvedenej na konci práce.

Košice, 2. máj 2014

.....

vlastnoručný podpis

Pod'akovanie

Rád by som vyjadril svoju úprimnú vďaku doc. Ing. Martinovi Tomášekovi, PhD. za všetky cenné rady, nápady, pripomienky, námety, ktoré prispeli ku skvalitneniu predkladanej práce i za venovaný čas, ktorý si vyžadovalo hľadanie riešení pri vzniknutých otázkach.

Obsah

Zoznam obrázkov	9
Zoznam tabuliek	11
Zoznam symbolov a skratiek	12
Slovník termínov	14
1 Úvod.....	15
2 Analýza požiadaviek	18
2.1 Scenár	18
2.2 Zavedenie pojmov	19
2.3 Aktéri systému.....	19
2.4 Marketingová stratégia	21
2.5 Požiadavky kladené na systém	21
2.5.1 Požiadavky na funkcionality.....	22
2.5.2 Požiadavky na používateľské rozhranie	25
2.5.3 Požiadavky na softvérové rozhranie	26
2.5.4 Požiadavky na výkon	27
2.5.5 Požiadavky na bezpečnosť	27
2.5.6 Požiadavky na platformu	28
2.5.7 Požiadavky na údržbu	29
2.6 Prípady použitia.....	29
2.7 Konceptuálny model architektúry systému	33
3 Prehľad aktuálneho stavu technológií.....	35
3.1 Prehľad technológií pre realizáciu definovaného plátna	35
3.1.1 Adobe Flash	35
3.1.2 Microsoft Silverlight.....	37
3.1.3 HTML5, CSS, SVG, Javascript a AJAX	39
3.2 Prehľad technológií pre realizáciu online platieb.....	44
3.2.1 Úvod do elektronického podnikania	44
3.2.2 Platobné metódy – kategorizácia, výhody, nevýhody	44
3.2.3 Vybrané platobné metódy a analýza ich realizácie.....	49
3.3 Prehľad technológií pre integráciu so sociálnymi sieťami v oblasti internetu.....	54

3.3.1	Integrácia so sociálnou sieťou Facebook	54
3.3.2	Integrácia s Google+	55
3.3.3	Integrácia so sociálnou sieťou LinkedIn	57
3.3.4	Integrácia s Pinterest	59
4	Prehľad existujúcich riešení podobného zamerania	67
4.1	Riešenia použitím množiny HTML prvkov IMG	67
4.2	Riešenia použitím HTML prvkov MAP a AREA	70
4.3	Riešenia použitím prvkov vektorovej grafiky - SVG a PATH	73
4.4	Zhrnutie	74
5	Návrh grafického komponentu pre customizáciu navrhovaných produktov	77
5.1	Plátno	77
5.2	Základný produkt	77
5.2.1	Reprezentácia základného produktu	77
5.3	Komponenty	78
5.3.1	Stavy komponentov	78
5.3.1	Reprezentácia komponentov	79
5.3.2	Množina prípustných komponentov	79
5.3.3	Príslušnosť komponentov ku kategóriám	80
5.3.4	Triedy komponentov vzhľadom na možnosti ich umiestňovania	80
5.4	Výsledný produkt	81
5.5	Návrh tried pre modul interaktívnej tvorby produktov	81
5.6	Používateľské rozhranie pre potreby skladania komponentov	82
6	Návrh knižnice pre integráciu PayPal platieb	83
6.1	Rozhranie pre deklaráciu volaní vzdialených PayPal metód	83
6.2	Autorizačné PayPal údaje	83
6.3	URL adresy pre lokalizáciu PayPal servera	83
6.4	Výnimočné stavy	84
6.5	Stavy objednávky	84
6.6	Uchovanie dát transakcií	85
7	Návrh knižnice pre realizáciu integrácie s Pinterest	86
7.1	Návrh knižnice pre realizáciu oEmbed volaní	86
7.1.1	Reprezentácia oEmbed odpovedí	86
7.1.2	Hodnoty kľúčov oEmbed odpovedí	86

7.1.3	Výnimočné stavy	87
7.1.4	Generátor odpovedí pre neúspešné volania	88
7.1.5	Generátor odpovedí pre úspešné volania	88
7.2	Rozšírenie oEmbed knižnice pre potreby Pinterest volaní.....	89
7.2.1	Hodnoty kľúčov Pinterest odpovedí	89
7.2.2	Výnimočné stavy	89
7.2.3	Generátor odpovedí.....	89
8	Testovanie systému.....	91
8.1	Akceptačné testy pre modul customizácie	91
8.2	Integračné testy pre platbu PayPal	91
8.3	Integračné testy pre Pinterest Rich Pins	92
9	Záver.....	94
	Zoznam použitej literatúry	98

Zoznam obrázkov

Obr. 1 Používatelia navrhovaného programového riešenia a popis ich základných operácie so systémom	20
Obr. 2 Diagram aktivít používateľov systému poukazujúc na novozvolený marketingový prístup	21
Obr. 3 Členenie navrhovaného systému s ohľadom na očakávanú funkcionality. Základom je elektronický obchod doplnený funkciami pre customizáciu produktov (CP), šírenie reklamy prostriedkami sociálnych sietí (SS) a platby elektronickými platobnými metódami (PM).	22
Obr. 4 Prípád použitia s ohľadom na podsystem umožnenia prístupu.	29
Obr. 5 Model prípadu použitia pre potreby popisu správy komponentov	30
Obr. 6 Model prípadu použitia pre proces customizácie produktov	31
Obr. 7 Model prípadu použitia v prípade integrácie sociálnej siete.	32
Obr. 8 Prípád použitia systému pri realizácii platby za produkt.	32
Obr. 9 Konceptuálny návrh architektúry, súčasťou ktorej bude nami vytváraný systém[11].....	33
Obr. 10 Členenie softvérovej architektúry MVC, komunikácia medzi jednotlivými úrovňami i vzťah k báze dát[12]	33
Obr. 11 Hierarchia tried zobraziteľných objektov jazyka ActionScript	35
Obr. 12 Porovnanie modelov fungovania klasických webových aplikácií a webových aplikácií s použitím technológie AJAX[22].....	43
Obr. 13 Tok riadenia pri platobnom styku PayPal v prípade Web Checkout.....	51
Obr. 14 Priebeh komunikácie a výmena dát s popisom v príslušných krokoch pre PayPal Express Checkout	51
Obr. 15 Zjednodušený pohľad na interakciu medzi koncovým používateľom, prevádzkovateľom komerčného riešenia a sieťou Pinterest	60
Obr. 16 Bližšia interakcia medzi komunikujúcou trojicou: používateľ, prevádzkovateľ, Pinterest.....	62
Obr. 17 Enumerácia niektorých parametrov Pinterest API a oEmbed API a ich vzájomný prienik	65
Obr. 18 Výsledná reprezentácia produktu (b) vzniká správnym usporiadaním obrázkov komponentov (a) s ohľadom na ich príslušnosť do určitej vrstvy	68
Obr. 19 Aplikačné použitie skladania IMG obrázkov v prostredí strategickej hry Travian pre úpravu vzhľadu hrdinu	69
Obr. 20 Vyznačenie polygónu lemujúceho podrážku obuvi.....	70

Obr. 21 Ukážka potenciálu interaktívnej tvorby komponovaného produktu zmenou farby komponentu ako oblasti vyčlenenej polygónom opisujúceho príslušný komponent	71
Obr. 22 Priblíženie sa dojmu reálnejšieho zobrazenia zmenou viditeľnosti plochy nad upravovaným komponentom.....	71
Obr. 23 Využitie prvkov <map> a <area> v prostredí hry Travian pre explicitnú definíciu štruktúry dediny – ukážka pred a po jej výstavbe.....	72
Obr. 24 Ukážka zobrazenia športovej bundy, ktoré vzniklo vypustením uzlov v DOM HTML dokumentu, zobrazujúce množinu SVG PATH elementov popisujúcich štylovateľné komponenty produktu	73
Obr. 25 PNG obrázok v dvojakom prevedení s priehľadným a bielym pozadím (prvé dve sekcie) ako fotografia zachytávajúca skutočnú bundu a jeho umiestnenie nad vrstvou PATH elementov (tretia sekcia, tmavé pozadie z aplikácie) s cieľom dosiahnutia dojmu reálneho zobrazenia	74
Obr. 26 Stavový diagram pre komponent v procese akceptácie.....	79
Obr. 27 Diagram tried pre modul interaktívnej tvorby produktov	81
Obr. 28 Návrh používateľského rozhranie pre okno interaktívneho dizajnu.....	82
Obr. 29 Diagram tried popisujúci návrh knižnice pre realizáciu PayPal platieb.....	84
Obr. 30 Stavový diagram reprezentujúci jednotlivé stavy objednávky s ohľadom na spracovanie platby.....	85
Obr. 31 Návrh databázových tabuliek pre uloženie záznamov ako výsledkov PayPal API volaní	85
Obr. 32 Diagram tried spadajúcich pod knižnicu oEmbed	88
Obr. 33 Diagram tried pre knižnicu Pinterest nadväzujúcej na oEmbed.....	90
Obr. 34 Výpis tabuľky pre potvrdzované dáta adresy doručenia a dáta z vykonanej transakcie prenášaných z PayPal.	91
Obr. 35 Ukážka informačnej správy na fake účte klienta v PayPal prostredí pre testovanie	92
Obr. 36 Ukážka úspešnej validácie na stránkach Pinterest pre testovanie integrácie.....	93

Zoznam tabuliek

Tab. 1 Ukážka špecifikácie požiadavky v tabuľkovej forme s uvedením referenčného čísla, názvu, priority a bližšieho popisu.....	22
Tab. 2 Súhrn funkcionálnych požiadaviek na vytváraný systém.....	22
Tab. 3 Tabuľka špecifikujúca požiadavky na používateľské rozhranie systému	25
Tab. 4 Tabuľka špecifikujúca požiadavky na softvérové rozhranie systému.....	26
Tab. 5 Tabuľka špecifikujúca požiadavky na výkon systému	27
Tab. 6 Tabuľka špecifikujúca požiadavky na bezpečnosť systému.....	27
Tab. 7 Tabuľka špecifikujúca požiadavky na platformu	28
Tab. 8 Tabuľka špecifikujúca požiadavky na údržbu systému	29
Tab. 9 Výhody technológií HTML5 Canvas a SVG	41
Tab. 10 Nevýhody, ktoré so sebou prinášajú technológie HTML5 Canvas a SVG.	42
Tab. 11 Všeobecné parametre oEmbed odpovede.....	63
Tab. 12 Vlastné parametre JSON alebo XML oEmbed odpovede v prípade typu <i>photo</i>	64
Tab. 13 Vlastné parametre JSON alebo XML oEmbed odpovede v prípade typov <i>video</i> a <i>rich</i>	64
Tab. 14 Parametre JSON alebo XML odpovede pre Pinterest	65

Zoznam symbolov a skratiek

- AJAX Asynchronous JavaScript And XML**, spojenie viacerých technológií pre zobrazovanie webových dokumentov použitím XHTML a CSS, dynamické zobrazovanie a interakciu použitím DOM, výmenu dát a manipuláciu použitím XML a XSLT, asynchrónne získavanie dát použitím XMLHttpRequest a to všetko použitím jazyka JavaScript
- B2B business-to-business**, obchodovanie medzi podnikateľmi
- B2C business-to-consumer** (tiež business-to-customer), forma obchodovania medzi podnikateľom a spotrebiteľom cez maloobchodný predaj
- CSS Cascading Style Sheets**, predstavuje spolu s jazykom HTML jadro pre budovanie web stránok, pričom HTML slúži na definíciu štruktúry dokumentu, kdežto CSS dodáva dokumentu vizuálny a akustický charakter
- DES Data Encryption Standard**, šifrovacia funkcia s dvomi 64-bitovými vstupmi (kľúč a správa) poskytujúca 64-bitový šifrovaný výstup
- FTPD internet File Transfer Protocol server**, protokol pre prenos súborov v prostredí internetu
- HASH** matematická transformácia priradujúca dátam rôznej dĺžky čísla vopred stanovenej nenulovej dĺžky. Z výsledných dát nie je možné odvodiť pôvodný vstup
- HSL High Level Shading Language**, programovací jazyk pre grafické procesorové jednotky so syntaxou podobnou jazyku C pre definovanie DirectX efektov
- JSON JavaScript Object Notation**, formát pre ukladanie a výmenu dát, ktorého syntax je inšpirovaná spôsobom vytvárania objektov v jazyku JavaScript
- NFC Near Field Communication**, je súbor štandardov, ktoré definujú architektúru bezdrôtovej technológie slúžiacej na vzájomnú komunikáciu buď medzi dvomi zariadeniami alebo medzi zariadením a NFC tagmi, bezdrôtovými pamäťovými kartami s minimálnou kapacitou; cieľom je poskytnúť vlastníčkovi elektronického zariadenia možnosť spoľahlivého a bezpečného prenosu dát bez väčšej asistencie

NFS	Network File System , protokol pre distribuovaný systém súborov umožňujúci serveru zdieľať adresáre a súbory s klientom prostredníctvom siete, používateľ pracuje so vzdialenými adresármi a súbormi akoby boli umiestnené lokálne
PDA	Portable Digital Assistant , osobný digitálny asistent alebo tiež vreckový počítač, medzi základné funkcie ktorého patrí: adresár, kalendár, účtovníctvo, dokumenty, e-mail, poznámky, úlohy, hry, MP3, filmy, knihy, GPS a samozrejme komunikácia
POS	Point Of Sale , zariadenie, ktoré umožňuje akceptáciu platieb pomocou platobných kariet priamo na mieste predaja
RIA	Rich Internet Application , webové aplikácie navrhnuté tak, aby svojimi črtami používateľom pripomínali desktopové aplikácie; správanie sa RIA môžeme vnímať na dvoch úrovniach: interakcia na strane používateľa sa uskutočňuje použitím používateľského rozhrania zobrazeného webovými prehliadačmi, kdežto skutočné operácie a manipulácie nad dátami sa dejú na strane aplikačného servera
SHA	Secure Hash Algorithm , algoritmus rozšírenej hashovacej funkcie vytvárajúci odtlačok fixnej dĺžky
SSL	Secure Sockets Layer , globálny bezpečnostný štandard, ktorý vytvára zašifrované spojenie medzi web serverom a web prehliadačom
URL	Uniform Resource Locator , reťazec znakov definovanej štruktúry určujúci miesto na internete, kde informácia môže byť nájdená
XML	eXtensible Markup Language , ako už názov hovorí, ide o značkovací jazyk, ktorý si používateľ môže definovať aj sám; má textovú podobu a používa sa pre ukladanie a výmenu dát medzi ľuďmi, medzi počítačmi, ako aj medzi ľuďmi a počítačmi navzájom
XAML	eXtensible Application Markup Language , deklaratívny značkovací jazyk založený na XML firmou Microsoft pre inicializáciu štruktúrovaných hodnôt a objektov

Slovník termínov

Customizácia je zmena, resp. prispôsobenie systému individuálnym podmienkam používateľa.

Framework je programová štruktúra, ktorá slúži ako podpora pri programovaní, vývoji a organizácii iných programových projektov. Môže obsahovať podporné programy, knižnice, podporu pre návrhové vzory alebo doporučené postupy pri vývoji.

Komponent budeme v tomto dokumente používať najmä v spojitosti s vytváraním výsledného produktu, kde základný produkt bude customizovaný aplikáciou alebo zmenou prvkov – komponentov, ktoré ho tvoria.

Mobile commerce (m-commerce) je technológia využívajúca bezdrôtový prenos informácií, ako hlasových, tak aj dátových, pomocou mobilných telefónov a PDA. Môžeme ju charakterizovať aj ako využitie mobilných zariadení za účelom realizácie obchodných transakcií, kde sa mobilné zariadenie nachádza na začiatku alebo na konci celého procesu.

One-to-one marketing stratégie prístupu k zákazníkovi založená na myšlienke, že každý zákazník je rozdielny, má iné potreby, záujmy i správanie sa pri nákupe. Firma s obchodnou politikou podporujúcou one-to-one marketing sa snaží tieto odlišnosti využiť a ponúknuť zákazníkovi produkt šitý na mieru.

Platforma je počítačová architektúra a nástroje používajúce konkrétny operačný systém.

Shader je menší program alebo množina algoritmov v oblasti 3D počítačovej grafiky, ktorá má na starosti vykresľovanie, osvetľovanie a tieňovanie povrchov objektov.

1 Úvod

V súčasnosti sme svedkami rozšírenia počítačov, internetu a celkovo informačno-komunikačných technológií, z prostriedkov na komunikáciu sa stali prostriedky aj na zábavu, vzdelanie a obchodovanie a tieto technológie prenikli do všetkých oblastí ľudskej činnosti. Spoločnosť sa stáva čoraz viac informačnou. Aj v obchodných podmienkach sa tieto prostriedky začali tlačiť do popredia a preto sa v priebehu posledných rokov veľmi často začal v rôznych súvislostiach používať pojem elektronický obchod (e-commerce). Aj keď myšlienky o vykonávaní obchodu a platieb inak ako osobným kontaktom sú už dosť staré, reálnejšiu podobu však začali získavať práve s nástupom sietí a neskôr internetu[1].

Slovenská republika ako štát patriaci do Európskej únie, sa smernicou 2000/31/ES Európskeho parlamentu a rady zaväzuje vynakladať úsilie o vytvorenie ešte užších väzieb medzi štátmi a ľuďmi Európy, aby zabezpečila hospodársky a spoločenský pokrok, a to aj rozvojom elektronického obchodu v rámci informačnej spoločnosti s cieľom ponúknuť značné príležitosti na zamestnanie najmä v malých a stredných podnikoch, a na zlepšenie konkurencieschopnosti európskeho priemyslu za predpokladu, že každý bude mať prístup k internetu[2].

Nakupovanie online sa stalo značne obľúbeným až posledným desaťročím. V roku 2012 predaj realizovaný elektronickým obchodovaním v Spojených štátoch amerických dosiahol 289 miliárd dolárov, oproti 256 miliardám z predchádzajúceho roku. Počet osôb nakupujúcich digitálne sa odhaduje na 175 miliónov pre rok 2016 podľa odhadov eMarketu. Tendencii rastu potenciálu obchodovania na internete sa nevyhla ani Európska Únia, kde počet spotrebiteľov nakupujúcich tovar a služby online stúpol z 20% v roku 2005 na 40% v minulom roku.

Zo všetkých vyššie uvedených troch bodov jednoznačne vyplýva aktuálnosť témy elektronického obchodu.

Naším cieľom je v oblasti e-komercie zamerať sa na customizáciu. Customizácia vo veľkom (Mass Customization, ďalej MC) a One-to-one marketing sa stali predmetom výskumov už od roku 1992, keď sa objavilo prvé členenie MC v [3]. Otázkou vzťahu štandardného prístupu výroby a predaja produktov ku prístupu, ktorý predstavuje MC, čo je predmetom napríklad prác Xia a Rajagopolana v [4] či [5], sa zaoberať však nebudeme.

S uplatnením stratégie použitia elektronického obchodovania tak ako aj v reálnom živote, nevyhnutne súvisí reklama a propagácia predávaných výrobkov. V tomto smere čoraz väčšiu úlohu zohrávajú sociálne siete.

Sociálne siete bežne ponúkajú funkcie, ktoré v prostredí marketingu predstavujú dôležitý reklamný prostriedok pre presné cielenie marketingových aktivít[6]. Takáto možnosť ponúka priestor pre reklamné kampane efektívneho charakteru. Ba čo viac, charakteristiky sociálnych sietí – sú priestorom na konverzáciu, poskytujú možnosť zdieľania obsahu, ponúkajú prostriedky pre publikáciu obsahu vytváraného ich používateľmi, podporujú vzájomnú spoluprácu klientov na tvorbe – idú ruka v ruku s propagáciou dynamicky (prítom však používateľom samotným) interaktívne vytváraného produktu, potenciálne okamžite propagovaného i dodatočne upravovaného na základe tam prítomných vyjadrení názorov členov sociálnej siete.

Cieľom predkladanej práce je zanalyzovať technológie pre vývoj graficky interaktívneho webu v oblasti elektronického podnikania a vytvoriť šablóny vo forme knižníc a/alebo modulov podporou nielen záležitostí, ktoré by už mali byť akýmsi štandardom pre spomínanú oblasť ale aj technologických riešení, ktoré umožňujú aplikovať iné stratégie formy tvorby vlastného produktu kompozíciou a predaja v porovnaní s existujúcimi riešeniami. Rozhodnutím pre implementáciu aplikácie fungujúcej na webe podporíme myšlienku, že customizácia na internete potichu, ale isto nachádza svoje miesto[7][8].

Neodmysliteľnou súčasťou elektronického podnikania je zavedenie platobných prostriedkov do vlastných webových aplikácií, čomu sa budeme venovať ako druhému nosnému pilieru práce. Pozrieme sa na kĺb medzinárodných platobných metód, špecifickým riešeniam ako aj platobným tlačidlám bankových inštitúcií na slovenskom trhu.

S ohľadom na fakt, že navrhovaný systém má potenciál uplatniť sa v komerčnom prostredí, stojí za to vziať do úvahy vyššie opomenuté fakty týkajúce sa sociálnych sietí a uplatniť ich pre potreby reklamy. Tento štýl reklamy postavíme na voľnom šírení obrazových dát v prostredí sociálnych sietí, čomu predchádza nevyhnutnosť pochopiť podmienky ich integrácie do vlastného riešenia.

Postup pri nachádzaní odpovedí na stanovené otázky bude nasledovný. V štádiu analýzy sa budeme zaoberať všetkými aspektmi týkajúcimi sa vlastností na požadovaný systém (komponent), stanovíme základné požiadavky, charakterizujeme prípady jeho

použitia, budeme sa venovať potrebám jeho architektúry. Následne sa zameriame na technológie, ktoré by pripadali do úvahy pri realizácii stanovených cieľov, zohľadníme už existujúce aplikácie podobného charakteru, vezmeme do úvahy ich výhody a stanovíme ich slabiny, ktorým sa budeme snažiť v poskytovanom systéme vyhnúť.

Následne iniciujeme návrh systému. Pri návrhu a implementácii sa budeme pridržovať vodopádového modelu životného cyklu programového systému, ktorý je charakteristický definíciou požiadaviek, následne návrhom a až nakoniec implementáciou i testovaním. Funkčnosť a použiteľnosť sa pokúsime deklarovať na vykonštruovanom prípade použitia – elektronickom obchode s textilom s podporou tvorby a predaja vlastných odevov. Pritom však budeme vychádzať z toho, že používateľské rozhranie musí byť jednoduché pritom dostačujúce, inak rozhodnutie pre MC nebude predstavovať pre obchodníka očakávaný prínos [10].

Od úspešného riešenia zvolenej problematiky si sľubujeme poskytnutie technologického základu pre efektívnu a rýchlu realizáciu akéhokoľvek internetového obchodu s požiadavkou na interaktívny prístup k vytváraniu produktov, podporou platby cez internet ako aj reklamy a zdieľania prostredníctvom vybraných sociálnych sietí.

2 Analýza požiadaviek

Vraví sa, že každá dobrá prezentácia potrebuje na začiatku zaujímavý príbeh. Aj my si preto dovoľíme, predtým než prejdeme k samotnej analýze, tak trochu netradične začať scenárom, ktorý stojí na pozadí našej motivácie.

2.1 Scenár

Zákazník Jožko prichádza do kamenného obchodu predajne so športovým oblečením. Je jeseň, čoskoro prichádza sneh a fanúšik zimných športov je pevne odhodlaný investovať do svojej výbavy nemalý obnos peňazí, ktoré nazhŕňal počas letných brigád¹.

Vkráča dovnútra, úsmevom zdraví predajcu, prichádza k siahodhlému stojanu s bundami pre snowboardistov, s veľkým očakávaním prechádza každý jeden produkt rad za radom.

Jeho gustu je, zdá sa, ťažko vyhovieť. Prvá bunda je príliš krikľavá, na druhej chýba vrecko, tretia postráda kapucňu, štvrtá v poradí nemá na povrchu impregnačnú vrstvu, mať nejaké to vnútorné vrecko ako odkladací priestor je v prípade piatej bundy síce fajn, ale zvnútra ovešaný samými vreckami, veď všetkého veľa škodí. „**Bodaj by sa dalo čosi pridať, vymeniť z inej bundy či úplne odobrať,**“ lamentuje. Mrzutý opúšťa predajné miesto dúfajúc, že niekde inde nájde niečo, čo bude vyhovovať aj takému náročnému zákazníkovi ako je on sám. Predávajúci neutŕži žiaden zisk, tovar ostáva na pultoch a produkt výrobcu nenadobúda odbyt.

Pravdepodobne už jemne tušíte, kam tým smerujeme. Dá sa nejako predísť tomu, aby zákazník bol naozaj spokojný s tým, čo výroba a predajca ponúka? Dá sa predísť tomu, aby zakaždým predajca prišiel o potenciálny zisk iba preto, že sa stále nájde nejaký komponent na produkte, ktorý nie úplne vyhovuje zákazníkovým požiadavkám?

V dobe, keď sa snažíme znížiť výrobné náklady stratégiou výroby rovnakých produktov vo veľkom, má odpoveď skôr negatívny charakter. Akokoľvek, zmenou marketingovej stratégie a preorientovaním sa na výrobu prototypov by sme mohli doceliť nastolenie odpovede s oveľa pozitívnejšou prognózou. Je jasné, že priklonenie sa k neštandardnej predajnej politike súvisí so zameraním sa na zákazníkov s patričným

¹ Sme si vedomý marketingovej orientácie na zákazníkov, ktorí sú ochotní zaplatiť za produkt vyššie ceny než môže byť štandardom. Vyššie ceny v takomto prístupe však podľa [10] nepredstavujú problém a ráta sa s nimi.

kapitálom a ich cieľom nešetriť a zainvestovať (do produktu), no možno je takýto risk práve správnu cestou – cestou, ako sa pretlačiť do popredia v tvrdom konkurenčnom prostredí.

Vedeli by sme sa stať autormi systému orientovaného na prototypovú výrobu a predaj? Aké funkcie by pre takýto systém bolo potrebné stanoviť? Aká architektúra by preň bola vhodná? Aké by boli obmedzenia? Existujú takéto riešenia? Ak áno, čo poskytujú a čo im prípadne chýba?

Keďže, inšpirovaní skúsenosťami zo života podobnými na začiatku uvedeným sme sa rozhodli zanalyzovať existujúce možnosti a navrhnúť i zrealizovať systém, ktorý by podporil stratégiu podnikania so zameraním na výrobu prototypových produktov, odpovede na nastolené otázky nájdete na nasledujúcich stranách práce, ktorú práve držíte vo svojich rukách.

2.2 Zavedenie pojmov

Najsamprv stanovme pojmy, ktoré sa budú v nasledujúcom texte vyskytovať. Definujme pojem *základného produktu* ako produktu, ktorého vlastnosti (cena, dostupné veľkosti, farba, materiál) určuje prevádzkovateľ pri zavádzaní do systému. Jeho podstatou je vytvorenie základu pre aplikáciu ďalších *komponentov* a pre výpočet ceny, od ktorej sa výsledná cena produktu odvíja. Záverečným výsledkom customizačného procesu nech bude *finálny produkt*. Produkt v štádiu návrhu nazývame *customizovaný produkt*.

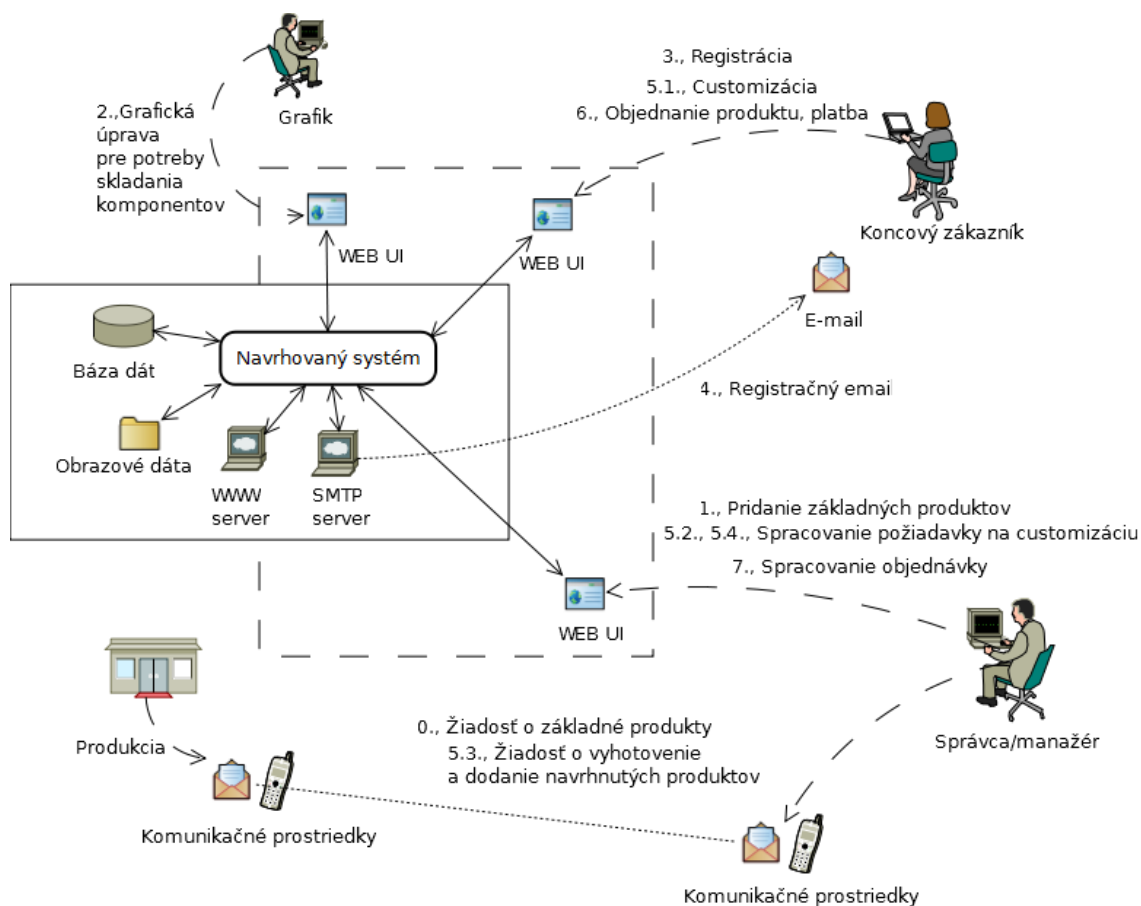
2.3 Aktéri systému

Stručne vymenujme a popíšme množinu používateľov, ktorí k nami vytváranému systému majú pristupovať:

- Producent, výrobca – organizačná jednotka zodpovedná za samotnú fyzickú výrobu produktu komunikujúca s prevádzkovateľom,
- Prevádzkovateľ, obchodník, manažér – organizačné jednotka, ktorá spravuje systém. Pridáva doň produkty, komunikuje so zákazníkmi, vybavuje objednávky a tým predstavuje premostenie medzi producentom a koncovým zákazníkom.
- Grafík – entita, ktorá má privilégium prístupu k obrazovým dátam, vie ich stiahnuť, upraviť a naspäť nasadiť podľa potreby,

- Koncový zákazník, koncový používateľ, klient – pristupuje k systému s cieľom kúpy produktu. Produkt upravuje podľa vlastných predstáv v rámci možností stanovených systémom.
- Vývojár – programovo realizuje systém, má na starosti bezstarostný chod aplikácie, stará sa o jej údržbu a zavádza nové rozšírenia.

Základné operácie používateľov vo vzťahu k aplikácii popisuje Obr. 1.

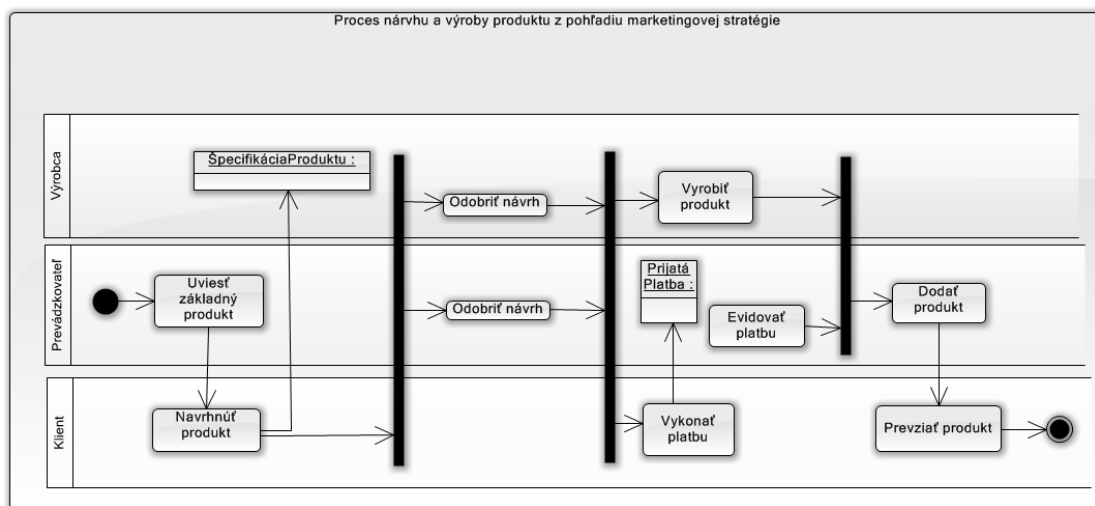


Obr. 1 Používatelia navrhovaného programového riešenia a popis ich základných operácie so systémom

Diagram prípadov použitia bude uvedený neskôr v stati 2.6.

2.4 Marketingová stratégia

Aktivitu používateľov systému vo vzťahu k marketingovej stratégii popísanej vyššie môžeme pre lepšie pochopenie znázorniť nasledujúcim diagramom aktivít na Obr. 2:



Obr. 2 Diagram aktivít používateľov systému poukazujúc na novozvolený marketingový prístup

Prevádzkovateľ zavádza do systému základný produkt. Klient pomocou dostupných komponentov stavia na základnom produkte svoj vlastný – customizovaný produkt. Prevádzkovateľ i výrobca majú právo odobrenia návrhu. Výrobca produkt fyzicky vyrába, klient medzitým vykoná platbu. Prevádzkovateľ eviduje platbu a po výrobe dodáva produkt zákazníkovi.

2.5 Požiadavky kladené na systém

Skúsme teraz detailnejšie uviesť požiadavky na cieľ našej cesty, ktorý predstavuje vytváraný systém.

Požiadavky rozčleníme do niekoľkých kategórií počnúc požiadavkami na funkcionality, používateľské rozhranie, softvérové rozhranie, údržbu, bezpečnosť, platformu. Každú označíme referenčným symbolom v tvare SKRATKA#ČÍSLO, aby sme sa na ne mohli v ďalšom texte odkazovať a ľahko orientovať. Prisúdime jej výstižný názov, charakteristiku, prioritu jej naplnenia, to všetko vo forme tabuľky nasledovného charakteru:

Tab. 1 Ukážka špecifikácie požiadavky v tabuľkovej forme s uvedením referenčného čísla, názvu, priority a bližšieho popisu

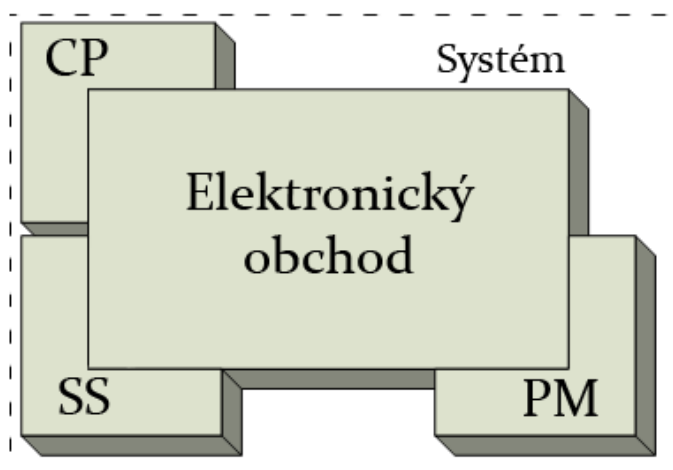
SKRATKA#ČÍSLO	Názov požiadavky/Stručný popis	Priorita
Tu bude uvedená bližšia charakteristika danej požiadavky.		

Prioritu číslujeme hodnotou v rozsahu 1 až 3 s nasledovnou sémantikou:

- 1 – voliteľná realizácia (nie až tak dôležitý aspekt systému, aspekt ponechaný na výber inými okolnosťami),
- 2 – nie nevyhnutná realizácia, neohrozí ostatné požiadavky systému,
- 3 – nevyhnutná realizácia, základná požiadavka na vytváraný systém.

2.5.1 Požiadavky na funkcionality

Požiadavky na funkcionality pripravovaného systému usporiadame v závislosti od členenia knižníc/modulov dopĺňujúcich systém, ktorý má charakter elektronického obchodu (viď Obr. 3).



Obr. 3 Členenie navrhovaného systému s ohľadom na očakávanú funkcionality. Základom je elektronický obchod doplnený funkciami pre customizáciu produktov (CP), šírenie reklamy prostriedkami sociálnych sietí (SS) a platby elektronickými platobnými metódami (PM).

Prvé riadky budú venované vlastnostiam modulu pre interaktívne komponovanie produktov, následne prejdeme k požiadavkám na platobné metódy a posledné riadky tabuľky popíšu charakteristiky požadovaných funkcií pre integráciu so sociálnymi sieťami.

Tab. 2 Súhrn funkcionálnych požiadaviek na vytváraný systém

F#1	Zavádzanie základných produktov – statická časť	3
-----	---	---

Prevádzkovateľ systému musí mať privilégium pre vkladanie základných produktov do systému. Za zavedením základných produktov stojí umiestnenie (upload) grafickej reprezentácie do bázy dát servera, špecifikácia ceny, materiálu, veľkosti základného produktu a pod.		
F#2	Zavádzanie základných produktov – dynamická časť	3
Pri zavádzaní základného produktu je potrebné dodať dynamickú časť – logiku pre podporu customizácie statickej časti. Customizácia statickej časti sa upriamuje na zmenu farby, materiálu a iných grafických reprezentácií produktu. Výber formy a technológie bude predmetom analýzy existujúcich riešení.		
F#3	Dodatočná úprava charakteristík už umiestneného základného produktu	1
Požiadavkou je dodatočná úprava charakteristík základného produktu prevádzkovateľom po jeho vložení do systému.		
F#4	Zobrazenie základných produktov	3
K dispozícii nech bude množina všetkých základných produktov pripravených na customizáciu. Z tejto množiny nech je používateľ presmerovaný do sekcie úpravy tovaru výberom konkrétneho základného produktu.		
F#5	Stanovenie množiny dostupných komponentov	2
Prislúcha právomoci prevádzkovateľa, ktorý reláciami 1:N (základný produkt:komponent) určí vzťahy aplikovateľnosti komponentov na produkt. V prípade, že bude zavedený systém viacnásobnej kontroly návrhu, je to skôr dopĺňajúca črta systému.		
F#6	Stanovenie charakteru komponentu	3/1
Aj komponent samotný má mať definovanú svoju povahu týkajúcu sa možnosti jeho umiestňovania. V prípade statických komponentov je jeho lokalizácia na plátne nemeniteľná, akceptácia statických komponentov systémom nech bude nevyhnutnosťou. V prípade dynamických komponentov je jeho umiestnenie voľné, akceptácia a manipulácia s dynamickými komponentmi by bola nad rámec základných požiadaviek.		
F#7	Zobrazenie množiny dostupných komponentov	3
Nevyhnutným základom pre dosiahnutie cieľov je získanie prostriedku pre načítanie		

množiny dostupných komponentov.		
F#8	Aplikácia komponentu na produkt – vizuálna stránka	3
Výberom komponentu z dostupnej množiny má dôjsť k jeho vizuálnemu pridaniu na plátno, grafická reprezentácia má byť umiestnená s ohľadom na charakter komponentu.		
F#9	Aplikácia komponentu na produkt – logická stránka	3
Aplikácia má zapríčiniť prepočítanie ceny vytváraného produktu (pripočítaním ceny komponentu).		
F#10	Uloženie výsledku customizácie	3/1
Akonáhle sa používateľ rozhodne, že úpravy zodpovedajú jeho predstavám, systém bude mať k dispozícii prostriedok pre pridanie produktu do hotových návrhov. Uloženie stavu navrhovaného objektu pre neskoršiu manipuláciu (akýsi save) by mohol byť nadštandardnou funkciou.		
F#11	Zdieľanie výsledkov customizácie	2
Výsledky procesu úspešného dizajnu nech sú dostupné k zhliadnutiu aj v prípade iných koncových používateľov systému.		
F#12	Úprava finálnych produktov inými používateľmi	1
Stálo by za zváženie poskytnutie možnosti pre úpravu finálnych produktov inými používateľmi než je autor sám.		
F#13	Obohatenie množiny komponentov o vlastné prvky - návrh	3
Jedným z rozšírení by mohlo byť zavedenie vlastných komponentov do systému. Proces zavádzania by prebiehal tak, že klient vkladá do systému grafický návrh komponentu v jednom z formátov pre rastrové zobrazenie (JPEG, PNG, BMP, GIF).		
F#14	Obohatenie množiny komponentov o vlastné prvky – akceptácia	3
Akceptáciu návrhu má v právomoci prevádzkovateľ systému.		
F#15	Riedenie množiny klientmi navrhovaných komponentov	1
Po odmietnutí návrhu komponentu používateľa nech je k dispozícii prostriedok pre jeho odstránenie zo systému. Odstrániť ho môže koncový používateľ alebo odstránenie vykoná prevádzkovateľ, ak to uzná za vhodné (grafická reprezentácia má vulgárny či erotický podtón a pod.).		

F#16	Integrácia vybranej platobnej metódy	3
Výsledok analýzy ako najvhodnejšej platobnej metódy nech je uskutočnený integráciou do štruktúr popisovaného programu. Pritom nech sa berie do úvahy širokospektrálnosť nasadenie riešenia a medzinárodný charakter, ak to okolnosti dovoľia (zákonné obmedzenia a pod.).		
F#17	Zavedenie tlačidla pre zdieľanie vybranej sociálnej siete	2
Nech na stránke, ktorá predstavuje detailný pohľad na produkt, bude umiestnené tlačidlo pre zdieľanie prostredníctvom zvolenej sociálnej siete. Použitím prostriedku pre zdieľanie nech dôjde k propagácii tohto produktu. Voľba najvhodnejšej sociálnej siete bude objektom ďalšieho štádia – analýzy.		
F#18	Ďalšie rozšírenie v prostredí sociálnych sietí	1
Ak to analýza umožní a bude mať zmysel nad tým uvažovať, nech je implementované aj ďalšie možné rozšírenie pre integráciu s vybranou sociálnou sieťou		

2.5.2 Požiadavky na používateľské rozhranie

Základné požiadavky na používateľské rozhranie systému uvádzame v Tab. 3.

Tab. 3 Tabuľka špecifikujúca požiadavky na používateľské rozhranie systému

PR#1	Zložitosť	2
Používateľské rozhranie musí byť jednoduché na pochopenie, intuitívne ovládateľné, nenáročné na orientáciu v prostredí.		
PR#2	Ovládanie	3
Práca so systémom bude prebiehať za pomoci ovládania klávesnice pre vstup dát a myši pre spúšťanie akcií.		
PR#3	Umiestnenie hlavného menu	1
Hlavné menu nech je umiestnené na prehľadnom mieste, najlepšie buď hore na vrchu alebo na boku stránky.		
PR#4	Obsah hlavného menu	3
Hlavné menu nech obsahuje odkaz na modul pre interaktívne skladanie komponentov. Umiestnenie ostatných položiek (prihlásenie, odhlásenie, kontakt, profil a pod.) v rámci menu nech je ponechané na voľbu programátora.		

PR#5	Logo integrovanej sociálnej siete	1
Logo sociálnej siete, s ktorou je aplikácia prepojená, musí byť na viditeľnom mieste reprezentované grafickou reprezentáciou, ktorú stanovuje dokumentácia sociálnej siete.		
PR#6	Logo integrovanej platobnej metódy	3
Vzhľadom k tomu, že organizácie ponúkajúce integráciu platobných metód prísne stanovujú spôsob reprezentácie prepojení na ich rozhrania, je nevyhnutné podriaďovať sa ich požiadavkám a zaviesť takú grafickú reprezentáciu, ktorá bude vychádzať z pravidiel uvedených v príslušnej API dokumentácii.		
PR#7	Obsah stránky pre customizáciu produktov	3
Nech obrazovka pre customizáciu obsahuje panel základných vlastností navrhovaného produktu (názov, bližší popis, autor), plátno pre zobrazenie základného produktu a skladanie komponentov, ako aj množinu komponentov (kategória, popis a pod.), pre použitie ktorých sa môže používateľ rozhodnúť.		
PR#8	Umiestnenie hlavných panelov pre customizáciu produktov	1
Umiestnenie panelu pre základnú charakteristiku nech bude na ľavej strane, interaktívne plátno nech nájde svoje umiestnenie v strede a množina komponentov nech je umiestnená na pravej strane používateľovej obrazovky.		
PR#9	Zobrazenie produktov dostupných na predaj	3
Produkty pripravené na predaj nech sú zobrazené na samostatnej stránke s možnosťou prekliknutia na ich detailné zobrazenie.		
PR#10	Prístup štandardne dostupnými webovými prehliadačmi	2
Webová aplikácia nech je optimalizovaná pre najviac rozšírené grafické webové prehliadače tak, aby práca s aplikáciou prebiehala rovnako bez ohľadu na výber prehliadača.		

2.5.3 Požiadavky na softvérové rozhranie

Požiadavky na softvérové rozhranie nájdete v nasledujúcej tabuľke.

Tab. 4 Tabuľka špecifikujúca požiadavky na softvérové rozhranie systému

SR#1	Charakter pluginu pre podporu customizácie produktov	2
Podpora customizácie tovaru nech je v systéme realizovaná vo forme modulu, ktorý pokrýva všetky vrstvy aplikácie od prezentačnej až po údajovú.		

SR#2	Charakter pluginu pre podporu integrácie sociálnej siete	3
Zavedenie nástrojov pre uskutočnenie prepojení so sociálnou sieťou nech má charakter osobitnej knižnice.		
SR#3	Charakter pluginu pre podporu integrácie platobnej metódy	3
Integrácia platobnej metódy musí spĺňať požiadavky individuálnej knižnice pre nezávislé začlenenie do systému.		

2.5.4 Požiadavky na výkon

Výkonnostné požiadavky charakterizuje nasledujúca tabuľka.

Tab. 5 Tabuľka špecifikujúca požiadavky na výkon systému

V#1	Presun čo najväčšieho počtu výpočtov na stranu klienta	3
Vykonávanie customizačných operácií nad komponentmi v sekcii pre skladanie výsledného produktu bude pravdepodobne výpočtovo najnáročnejším procesom v rámci celej aplikácie. Zátťaž na strane servera by mohla byť príliš vysoká, preto je potrebné presunúť čo najväčší počet možných výpočtov na stranu klienta.		
V#2	Asynchrónne spracovanie požiadaviek na server	3
Pri návrhu a implementácii je potrebné orientovať sa na uskutočnenie asynchrónnych dopytov na server, aby sa zamedzilo načítavaniu celej stránky v prípadoch, kde je obnovenie kritické. Príkladom môže byť zapojenie technológie AJAX v časti, kde sa používateľ prihlasuje do systému, odosiela návrh na spracovanie a pod.		

2.5.5 Požiadavky na bezpečnosť

Základné bezpečnostné požiadavky zahŕňa Tab. 6.

Tab. 6 Tabuľka špecifikujúca požiadavky na bezpečnosť systému

B#1	Autentifikácia a autorizácia	3
Systém musí podporovať autentifikáciu a autorizáciu pomocou zavedenia mena a hesla pre používateľa systému.		
B#2	Prístup k funkcionalite z hľadiska používateľských rolí	3
Splnením požiadavky B#1 získame prostriedky pre identifikáciu role používateľa. Rola používateľa bude definovať množinu dostupných funkcií, ktoré prihlásený používateľ môže realizovať. Povolené smú byť len operácie uvedené vo funkcionálnych		

požiadavkách a to s ohľadom na používateľskú rolu. Operácie mimo stanovených pravidiel musia byť odopreté.		
B#3	Prístup k zdrojovým kódom	2
Prístup k zdrojovým kódom bude mať vývojár systému, ktorý ich zavedie do produkcie umiestnením na príslušnom serveri. K umiestneniu nech je použitý FTP server alebo iný prostriedok stanovený prevádzkovateľom servera.		
B#4	Prístup k obrazovým dátam	2
Prístup ku grafickým dátam má byť umožnený návrhom a implementáciou v takom rozsahu, aby žiaden manuálny zásah v súborovom systéme na strane servera nebol potrebný. Všetka manipulácia má prebiehať výlučne prácou so systémom samotným.		
B#5	SSL komunikácia	3
Všetka komunikácia so systémom, pre ktorú je potrebné autorizovať a autentifikovať používateľa, nech prebieha v rámci štandardov stanovených šifrovaným protokolom HTTPS.		

2.5.6 Požiadavky na platformu

V rámci implementácie sa pravdepodobne prikloníme k programovaciemu jazyku PHP, a to z výkonnostných dôvodov ako i dôvodov jeho rozsiahlej podpory na webových serveroch. Databázový server nech je riadený relačnou databázou, my si vyberieme MySQL (analogické dôvody ako v prípade PHP).

Tab. 7 Tabuľka špecifikujúca požiadavky na platformu

P#1	Podpora knižnice PHP5.2+	3
Aplikačný server musí podporovať vykonanie zdrojového kódu písaného v jazyku PHP verzie 5.2.		
P#2	Podpora SRBD relačného charakteru	3
Nech systém riadenia bázy dát má charakter relačnej databázy.		
P#3	Voľba umiestnenia SRBD	1
Systém riadenia bázy dát môže byť umiestnený buď priamo na web serveri, ktorý udržiava chod aplikácie alebo mimo neho s cieľom zabezpečiť jeho nezávislosť. Prítomnosť je nevyhnutná, no výber nie je podmienený špecifickými okolnosťami, preto priority pre rozhodnutie stanovujeme hodnotu 1.		

2.5.7 Požiadavky na údržbu

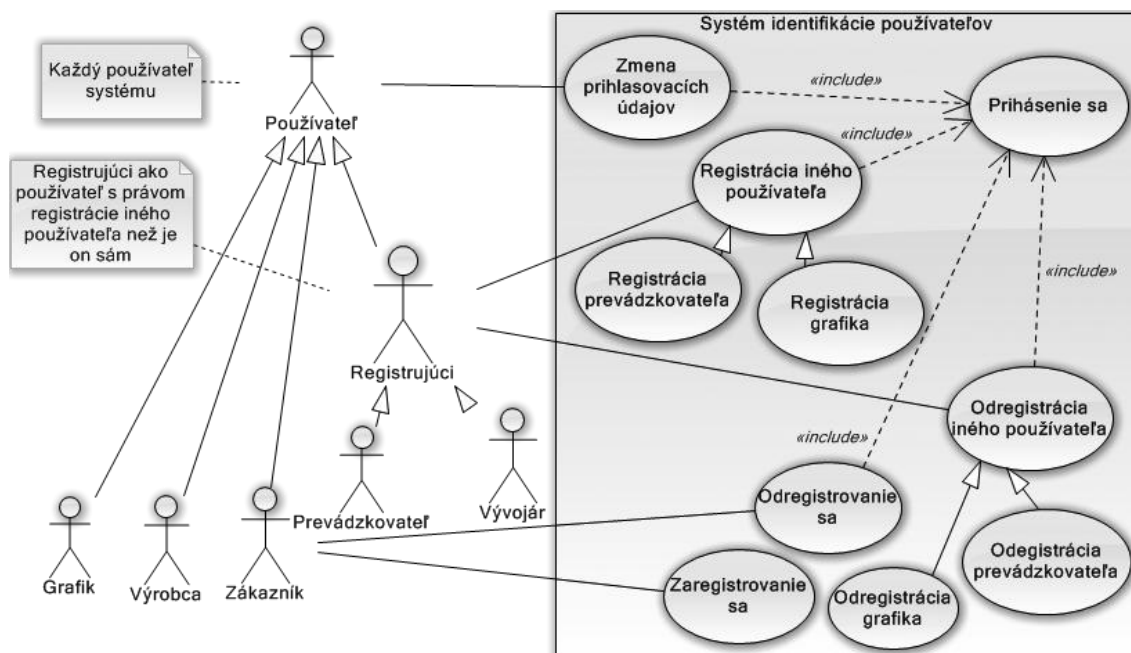
Jedinou základnou požiadavkou na údržbu systému je prístup vývojára k logovacím súborom aplikácie.

Tab. 8 Tabuľka špecifikujúca požiadavky na údržbu systému

U#1	Externý prístup k log súborom	3
Vývojár systému nech má prístup k textovým súborom, ktoré slúžia na monitorovanie toku dát a riadenia systému.		
U#2	Externý prístup k databáze	3
Vývojár systému nech má k dispozícii prostriedok pre externý prístup k používanej báze dát. Prístup nech je realizovaný buď webovým rozhraním (phpMyAdmin inštalovaným na strane databázového servera) alebo zabezpečením externého prístupu zriadením mena a hesla pre správu databázy ako aj adresy a portu pre jej dosiahnutie.		

2.6 Prípady použitia

S cieľom nájdenia hraníc systému, zavedenia jeho aktérov, nájdenia prípadov použitia s určením alternatívnych scenárov formujeme model prípadov použitia.

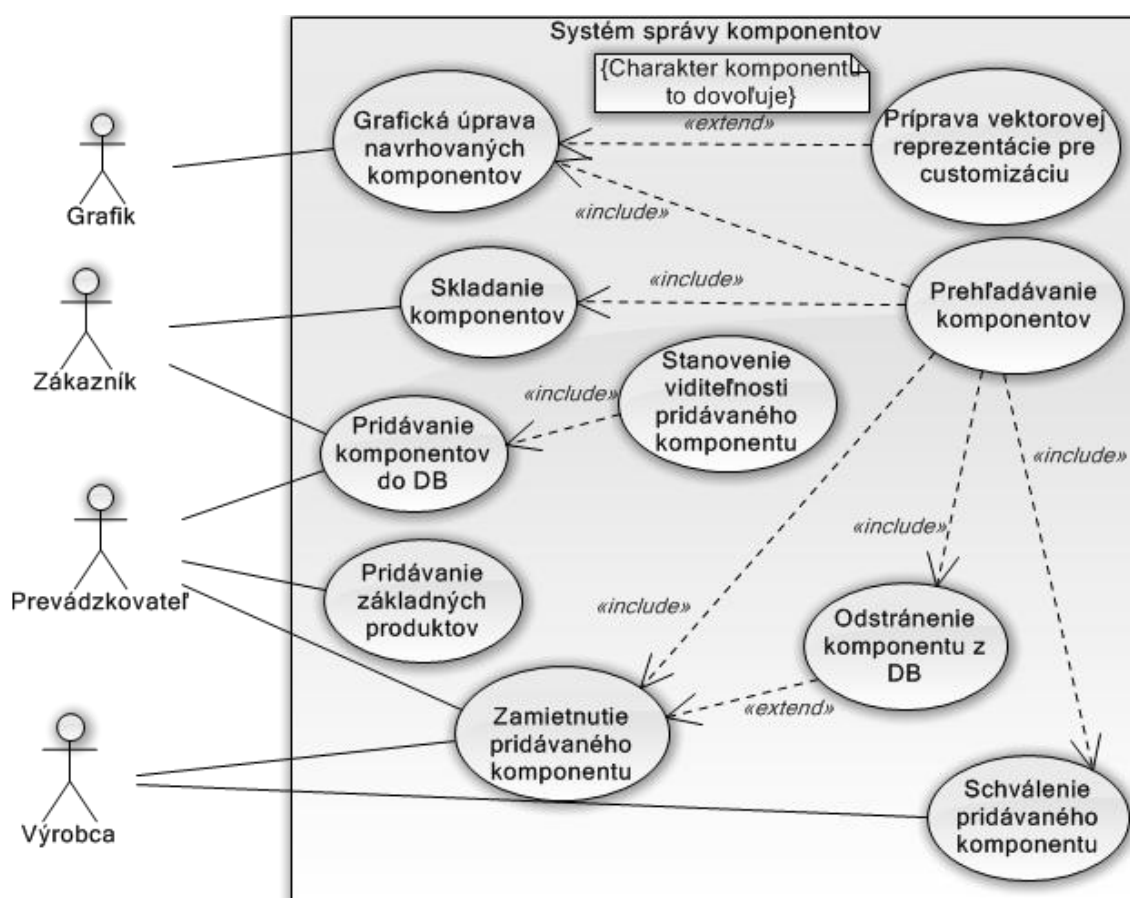


Obr. 4 Prípad použitia s ohľadom na podsystém umožnenia prístupu.

Prvý model prípadov použitia (Obr. 4) nech charakterizuje podsystém prístupu. Z modelu vyplýva nasledovné. Pre prácu so systémom je potrebné prihlásenie. Sám sa zaregistrovať a odregistrovať môže klient. Právo na registráciu iného

používateľa než je používateľ sám má prevádzkovateľ a vývojár. Ich právomoc zahŕňa vytvorenie konta pre grafika a iného prevádzkovateľa. Analogicky to platí aj v prípade odregistrovania iného používateľa. Iba pre úkon registrácie klienta nie je potrebné prihlásenie.

Obr. 5 stanovuje prípady použitia s ohľadom na správu komponentov. Pridávanie komponentov má v právomoci koncový zákazník a prevádzkovateľ. V prípade zamietnutia navrhovaného komponentu prevádzkovateľom alebo producentom je možné navrhovaný komponent z databázy (DB) odstrániť. Grafik má možnosť upraviť vizuálnu stránku pridávaného komponentu, ak mu to jeho charakter dovoľuje (napríklad lem rukáva, ktorého farbu je možné meniť).

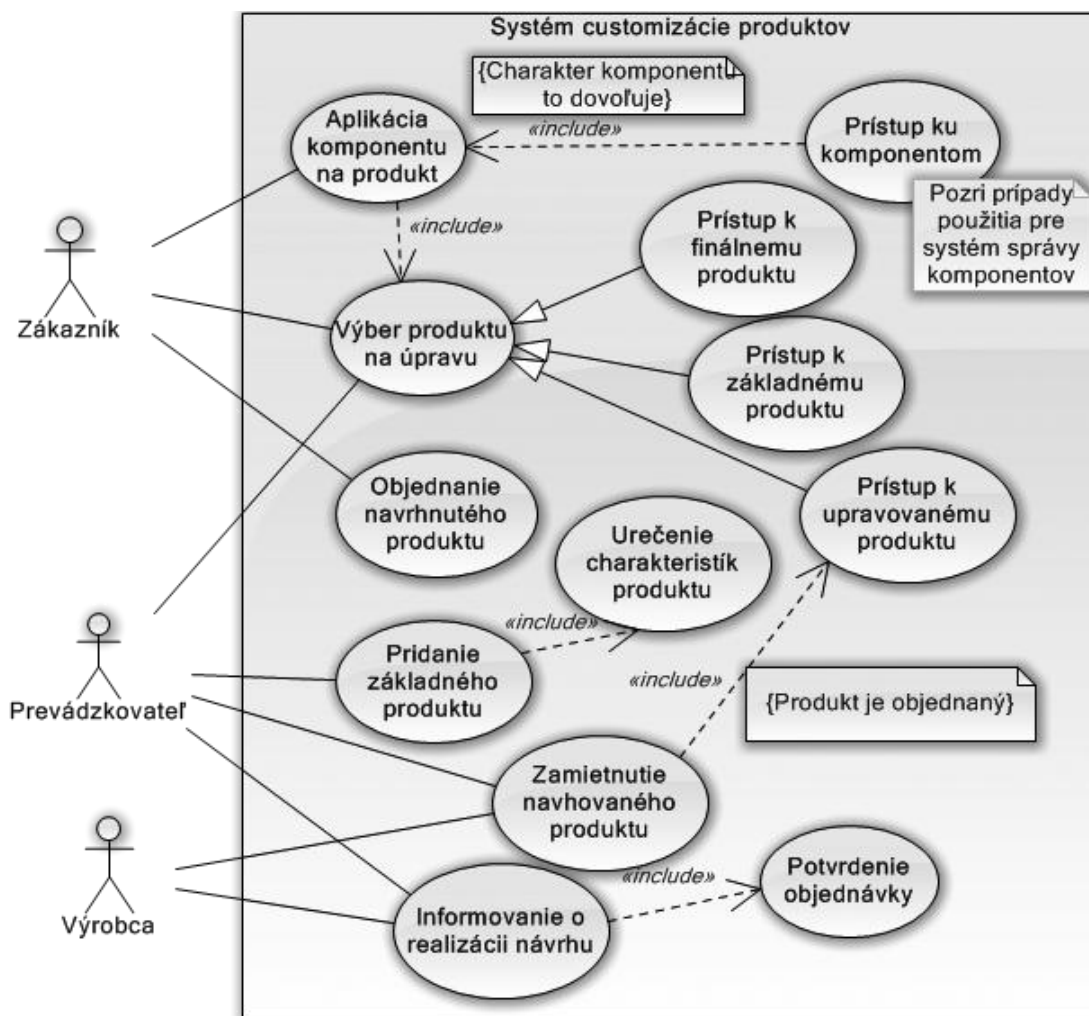


Obr. 5 Model prípadu použitia pre potreby popisu správy komponentov

Na správu komponentov nadväzuje customizácia. Klient má mať možnosť upravovať produkt podľa vlastných predstáv aplikáciou dostupných komponentov. Pritom nech stavia na základnom produkte pridaným prevádzkovateľom systému. Prevádzkovateľ pri zavádzaní základného produktu do systému určuje jeho základné charakteristiky (napríklad východzia cena). Po ukončení fáze dizajnu (objednávka)

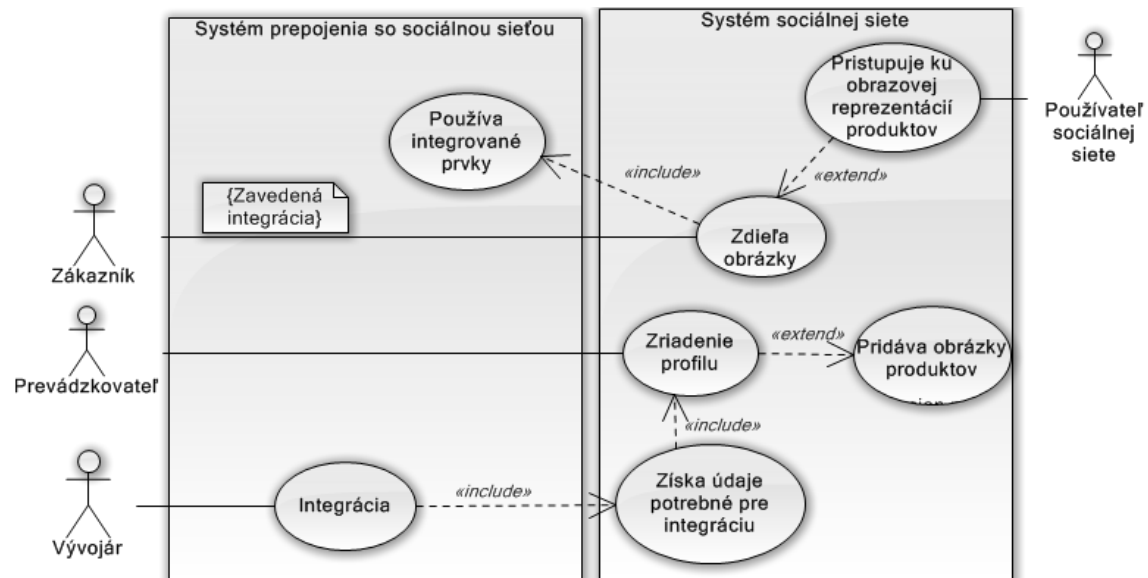
je návrh zaradený k produktom na schválenie. Ak je objednaný produkt schválený (prevádzkovateľom a výrobcom), zákazník je o tom informovaný. Pozri Obr. 6.

Prepojenie so sociálnou sieťou má docieľiť nasledovné. Integračnými prvkami, ktoré zavádza vývojár po získaní špecifických údajov registráciou prevádzkovateľa, klient systému zdieľa obsah na integrovanej sociálnej sieti. Akýkoľvek používateľ sociálnej siete má prístup k týmto obrazovým dátam.



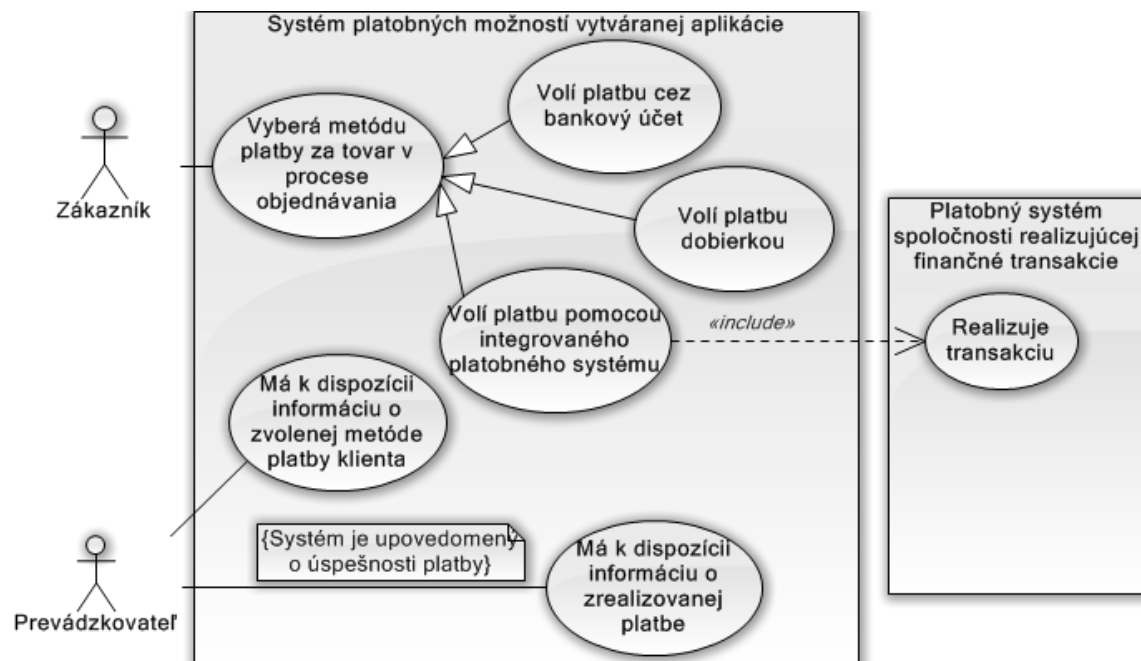
Obr. 6 Model prípadu použitia pre proces customizácie produktov

Môže ich voľne šíriť ďalej vo svojom profile, zoskupovať do kolekcií a podobne (ak to sociálna sieť umožňuje, predmet analýzy). Vid' ilustráciu modelu nižšie.



Obr. 7 Model prípadu použitia v prípade integrácie sociálnej siete.

Obr. 8 opisuje prípad použitia systému z hľadiska metódy platby, pre ktorú sa klient rozhoduje. V prípade platobnej metódy integrovanej systémom má mať aplikácia nástroj pre zistenie úspešnosti realizácie platby tak, aby prevádzkovateľ mal túto informáciu k dispozícii. Cieľom je čo najpromptnejšie vybavenie objednávky a zabezpečenie pohodlia nákupu odkiaľkoľvek.

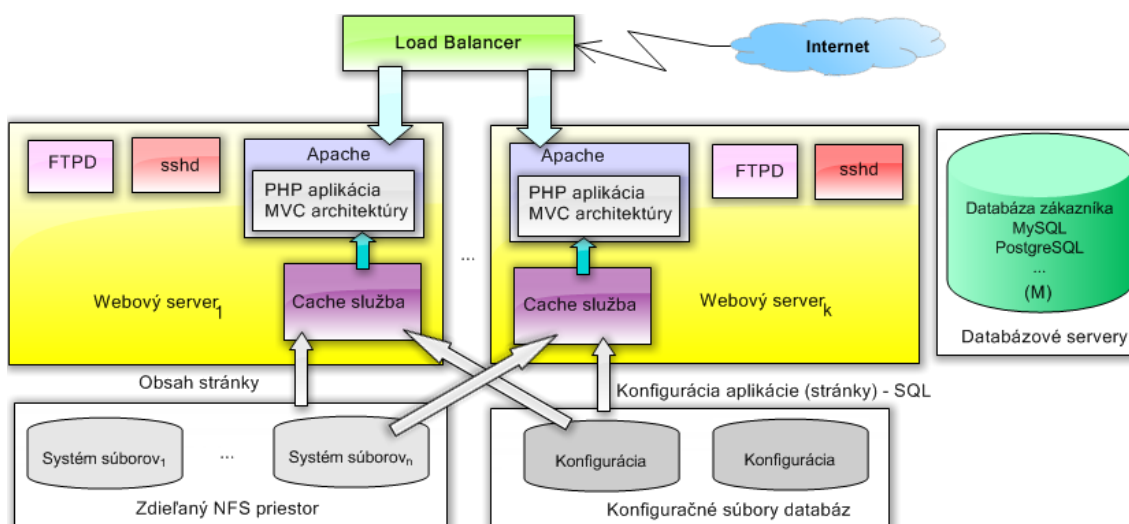


Obr. 8 Prípád použitia systému pri realizácii platby za produkt.

2.7 Konceptuálny model architektúry systému

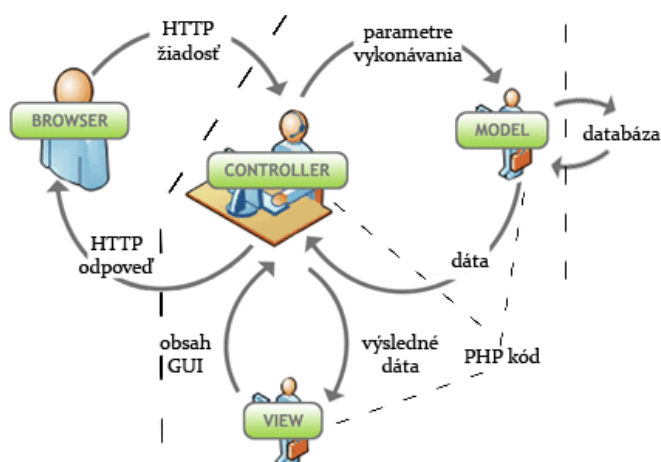
Ako už bolo naznačené, projekt by mal byť nasadený tak, aby bol dostupný po internetovej sieti. Niet dôvodu na jeho decentralizáciu, takže pôjde o centralizovaný nedistribučovaný systém. K jeho uskutočneniu budeme potrebovať WWW server pre umiestnenie výslednej aplikácie, najlepšie s podporou zasielania emailov (vstavaný SMTP server alebo existujúci osobitne), manipulujúcou nad databázou s prístupom k súborom s grafickou reprezentáciou produktov tak, ako to naznačuje Obr. 1.

Konceptuálny návrh architektúry samotnej vrátane webového servera Apache, cache služby, konfiguračných súborov databázového servera, uloženia grafických dát v rámci súborového systému NFS prístupného cez FTPD, ktoré vyhovujú stanoveným požiadavkám, zachytáva nasledujúci obrázok.



Obr. 9 Konceptuálny návrh architektúry, súčasťou ktorej bude nami vytváraný systém[11]

Softvérovou architektúrou bude MVC (Model View Controller, Obr. 10) model.



Obr. 10 Členenie softvérovej architektúry MVC, komunikácia medzi jednotlivými úrovňami i vzťah k báze dát[12]

MVC model umožňuje izolovať dátovú vrstvu (M) od prezentačnej (V) a aplikačnej (C) implementovaný v jazyku PHP v niektorom frameworku. Výhodou MVC architektúry je, že zmenou na jednej úrovni nedochádza k nevyhnutnosti vykonania zmien na úrovni inej. Preto je napríklad možné presedlať zo systému riadenia bázy dát jedného typu na typ druhý (napríklad MySQL na PostgreSQL) bez propagácie zmien na úrovni prezentačnej či aplikačnej.

3 Prehľad aktuálneho stavu technológií

Nasledujúce kapitoly sa venujú prehľadu existujúcich technológií pre realizáciu customizácie, integráciu platobných metód aj sociálnych sietí.

3.1 Prehľad technológií pre realizáciu definovaného plátna

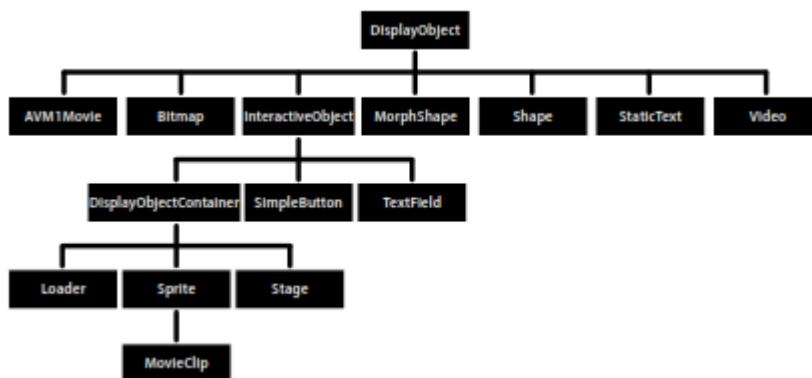
V rámci tejto podkapitoly sa pokúsime o zostavenie stručného prehľadu existujúcich technológií, ktorých využitie prichádza do úvahy pri implementácii modulu pre grafické plátno.

Prvé z opísaných technológií budú založené na spolupráci s webovým prehliadačom prostredníctvom tzv. sandboxu, ktorý predstavuje bezpečnostnú obálku, v rámci ktorej sa môže daná technológia realizovať. Môžeme ich takisto popísať aj ako programy, ktoré presúvajú aplikačnú logiku zo strany servera na stranu klienta, čo je podmienené potrebou odbremeniť záťaž servera a schopnosťou znášať väčšiu záťaž na strane klienta[13], čo v našom prípade, keďže pôjde o interaktívnu manipuláciu nad obrazovými dátami, bude nosnou požiadavkou.

3.1.1 Adobe Flash

Adobe Flash (pôvodne Macromedia Flash) je multimediálna a softvérová platforma pre použitie vektorovej grafiky, animácií, hier a RIA (Rich Internet Applications). Z dôvodu pokrytia našich požiadaviek na aplikáciu je pre nás podstatné tzv. *programovanie zobrazení* v objektovo-orientovanom jazyku ActionScript).

Obrázok nižšie popisuje hierarchiu tried balíka `flash.display` jazyka ActionScript, ktoré sú šablónami pre inšanciovanie zobraziteľných objektov.



Obr. 11 Hierarchia tried zobraziteľných objektov jazyka ActionScript

Pre nás sú podstatné najmä tieto triedy:

- `Bitmap` (slúži na definíciu bitmapových objektov, tie môžu byť vykreslené pomocou jazyka AS alebo načítané z externých zdrojov; podporované formáty sú JPG, PNG, GIF; pre úpravu bitmáp je k dispozícii trieda `BitmapData`),
- `InteractiveObject` (nadtrieda pre všetky objekty schopné interakcie s klávesnicou alebo myšou),
- `Stage` (reprezentuje práve jednu a jedinou plochu aplikácie, je umiestnená na prvom mieste v hierarchickom usporiadaní zobraziteľných objektov),
- `Loader` (používa sa na čítanie grafických dát, ako aj SWF súborov, z externých dátových zdrojov),
- `SimpleButton` (reprezentuje symbol tlačidla; dá sa použiť pre spúšťanie udalostí; má štyri stavy: *hore*, *dolu*, *nad* a *zahájenie testu* – `hitTestState`),
- `TextField` (slúži jednak pre zobrazenie textu, ako aj pre načítanie a prácu so vstupom vo forme textu),
- `Shape` (pripravená pre vytváranie vlastných vektorových grafických objektov – obdĺžnikov, čiar, kruhov a pod.).

Zobrazované objekty sú súčasťou stromovej štruktúry – zoznamu zobrazení. Vrcholom stromu je plocha, tá môže obsahovať viacero kontajnerov alebo objekty zobrazenia, ktoré opäť môžu obsahovať kontajnery a/alebo ďalšie objekty zobrazenia. Spomínaná hierarchická štruktúra inklinuje k tomu, že dokážeme z referencie na plochu postupne získať všetky zobrazené objekty, napríklad rekurziou.

Spomenúť môžeme ďalšie operácie nad zobrazovanými objektmi podporované v jazyku `ActionScript`, napríklad posun, otáčanie, manipuláciu s veľkosťou a zmenu merítka, aplikovanie rôznych režimov prekrytia (`alpha`, `darken`, `difference`, `erase`, `invert`, `lighten`, `multiply`), transformáciu farieb (úplná zmena alebo úprava s ohľadom na pôvodné farby – grafické filtre).

Medzi pozitíva môžeme zaradiť:

- podpora programovania v jazyku `ActionScript` open source vývojovými prostrediami (`FlashDevelop`, `MTASC`, `Haxe`),
- podpora paradigmy objektovo-orientovaného programovania,

- podpora spracovania dát vo forme súborov jazyka XML (eXtended Markable Language).

Negatíva:

- nevyhnutná[14] prítomnosť pluginu pre podporu Flash technológií,
- tlačidlo „krok späť“ v prehliadači nespôsobí návrat k predchádzajúcim viditeľným zmenám, ale vráti používateľa na pôvodnú webovú stránku, aj je vôbec povolené, pretože ak povolené nie je, môžeme to považovať za absenciu často používanej akcie,
- neštandardné zobrazovanie štandardných záležitostí, napríklad odkazu na inú stránku (záleží od vývojára, ako navrhne grafické zobrazenie premostenia – linku, odkazu – s inou stránkou), čo môže spôsobiť nutnosť zvyknúť si na nové pravidlá pri každej inej webovej stránke,
- odmietnutie podpory zo strany mobilných zariadení, napríklad iPhone,
- úpadok Flash technológií[15], čo bude mať v našom prípade pravdepodobne dôležité slovo pri konečnom výbere použitej technológie.

3.1.2 Microsoft Silverlight

Ďalšou technológiou, ktorá spočíva v rozšírení webového prehliadača o interaktívnu prezentačnú vrstvu (Rich Presentation Tier) je Silverlight od spoločnosti Microsoft, ktorej architektúra pozostáva z 2 vrstiev: *prezentačnej* (komponenty a služby orientované na generovanie používateľského rozhrania) a *inštaláčno-aplikačnej* (inštaláčny a aplikačný komponent pre internetový prehliadač).

Na manipuláciu s prezentačnou vrstvou programovo sa môže použiť JavaScript, pričom sa pracuje s objektovým modelom pracovnej plochy. Pre definíciu takéhoto rozhrania sa používa obdoba jazyka XML, jazyk XAML (eXtended Application Markup Language), ktorý vytvára základ pre dynamickú interakciu používateľa s ovládacími prvkami na stránkach.

Rozdielom v porovnaní s technológiou Flash je zasielanie textových dát prehliadaču oproti binárnym.

Prezentačná vrstva je spätá s objektovou hierarchiou, ktorú vytvárame pomocou objektov Canvas, Stack Panel a Grid. Canvas predstavuje našu pracovnú plochu, plátno, na ktorú umiestňujeme grafické objekty. Hierarchické usporiadanie umožňuje vnárať jednotlivé plátna. Objekt Stack Panel umožňuje rozmiestniť

objekty nad sebou alebo vedľa seba, implicitne vertikálne. V prípade potreby rozvrhnutie pracovnej plochy na niekoľko samostatných častí máme k dispozícii objekt `Grid`. Podporované sú napríklad aj elementy ako `PasswordBox`, `ComboBox`, `ProgressBar`, `DataGrid` (pre tvorbu tabuliek), `RadioButton`, `CheckBoxes`, `DatePicker`. Ďalšími podporovanými operáciami sú: obojsmerný prenos súborov, ukladanie súborov (podporované len v obmedzenej forme, tzv. `Isolated Storage`, kde ide o izolovanú časť disku a používateľ nemá priamy prístup k týmto súborom, po vypnutí aplikácie sú automaticky odstránené), reakcie na mnohé variácie manipulácie myšou, `Right-Click` (pri kliku pravým tlačidlom myši na konkrétny UI element sa zobrazí príslušná nápoveda) a `MultiTouch` (obsluha udalosti vzniknutej napríklad viacnásobným kliknutím tlačidla myši), `Clipboard Acces` (po akceptácii povolenia prístup napríklad k textu z iných práve bežiacich programov), tlač, zobrazenie obsahu HTML v rámci Silverlight aplikácie, lokalizácia, interakcia s jazykom JavaScript z pohľadu vonkajšieho prostredia aplikácie atď. Pre prácu s databázou sa používa technológia zvaná LINQ (*Language Integrated Query*), no nič nám nebráni použiť na dočasné lokálne ukladanie dát XML.

Silverlight zahŕňa všetky základné primitíva ako obdĺžnik, elipsa, cesta ako aj element pre prácu s médiami, umožňuje aplikovať efekty na bitmapové objekty (rozostrenie, vrhanie tieňa či iné efekty - napríklad vytvárané cez jazyk pre procesorové jednotky HLSL), podporuje HLSL shadre i Perspective 3D, z nášho hľadiska je pre nás tiež zaujímavá podpora základných transformácií. Silverlight podporuje jednoduché 2D transformácie, jednoduché 3D transformácie, viacnásobné 2D transformácie ako aj 2D a 3D transformácie aplikované naraz.

Silverlight takisto umožňuje rotovať ľubovoľný objekt `UIElement` vo vlastnom trojdimenzionálnom priestore nastavením vlastnosti `Projection`.

Medzi pozitíva môžeme zaradiť nasledovné:

- prístupnosť balíka Microsoft Expression Studio pre návrh a vývoj webových stránok (Microsoft Expression Web), vektorovej a bitmapovej grafiky (Microsoft Expression Design), kódovanie a úpravu video súborov (Microsoft Expression Encoder), ako aj katalogizačný nástroj pre dávkové spracovanie, úpravy, prevody a vytváranie katalógov multimediálnych súborov (Microsoft Expression Media). Súčasťou balíka je aj grafické vývojové prostredie pre tvorbu moderných aplikácií s interaktívnou

podporou 3D zobrazovania (Microsoft Expression Blend), čo umožňuje relatívne intuitívnym spôsobom – drag and drop – vytvárať a editovať prvky vektorovej grafiky i iné, ovládacie prvky,

- dostupnosť mnohých predefinovaných tém (kolekcie štýlov, ktoré popisujú vizáž všetkých kontrolných objektov, tlačidiel a pod.),
- Silverlight 2 a vyššie podporuje programovanie už aj v jazykoch rodiny .NET, nie len v JavaScripte,
- napojenie na zdroj dát prostredníctvom rozšírení ASP.NET AJAX, čo umožňuje volanie webových služieb z prostredia Javascriptu vo formáte XML alebo JSON,
- hardvérová akcelerácia na platformách operačného systému MS Windows[16].

Negatíva:

- nevyhnutná inštalácia doplnku prehľadávača, aj s ohľadom na kombináciu operačného systému a druh samotného prehľadávača,
- v prípade potreby využívania balíka Microsoft Expression Studio (alebo jeho častí) nevyhnutná inštalácia Microsoft .NET Framework, minimálne však potrebná inštalácia knižnice Silverlight Software Development Kit (SDK) pre vývoj takýchto aplikácií.

Všetky doposiaľ uvedené technológie podporujú zobrazenie na celú obrazovku.

3.1.3 HTML5, CSS, SVG, Javascript a AJAX

Mohli by sme povedať, že v predchádzajúcich kapitolách sme popisovali technológie, ktoré pracovali s jazykom HTML a CSS nepriamo. To znamená, že v rámci HTML stránky existoval pre aplikácie určitý vymedzený (a zároveň obmedzený) priestor, v ktorom aplikácia vykonávala svoju činnosť. Pôvodne boli však webové prehliadače určené na zobrazovanie dokumentov na strane klienta písaných priamo v jazyku HTML a CSS[13], či už išlo o textové alebo neskôr grafické prehliadače.

Uvádza sa, že výhodou technológií uvedených v kapitolách 3.1.1 a 3.1.2 je možnosť obohatenia klientskej strany o rozmanité grafické rozhranie, no s vývojom technológií ako JavaScript a príchodom technológie AJAX (Asynchronous JavaScript And XML) sa táto výhoda stáva skôr samozrejmosťou, a to aj bez potreby dodatočnej

inštalácie[16]. Uvedené vyplýva z podstaty technológie AJAX, ktorá je stále len zmesou HTML, CSS a JavaScriptu a XML[17], čo pre nás znamená značnú výhodu v porovnaní s predchádzajúcimi technológiami.

Prvky jazyka HTML (značky alebo tzv. elementy) uvádzané v zátvorkách “<” a “>” definujú štruktúru dokumentu², vizáž určujú kaskádové štýly a dynamiku dodávajú skripty písané v jazyku JavaScript.

Obrázky vníma prehliadač ako multimediálne elementy, ktoré sa do dokumentu načítajú zo zdroja. Zdrojom môže byť buď obrázok špecifikovaný cestou, uložený na strane servera, ktorý poskytuje otváraný HTML dokument (tzv. vloženie, používa sa element) alebo zdrojom môže byť iný server – externý zdroj (tzv. odkazovanie, používa sa element <a>). Podporovaný formát závisí od prehliadača, nie je teda stanovený štandardom, základné formáty ako JPEG, GIF či PNG (dokonca aj BMP) sú však kompatibilné s najpoužívanejšími[18] prehliadačmi (Internet Explorer, Google Chrome, Mozilla Firefox, Opera, Chromium).

Značka <canvas> je HTML elementom použiteľným pre kreslenie pomocou skriptov (zvyčajne ide o JavaScript)[19]; ako už jej názov napovedá, ide o plátno. Je fixnej veľkosti reprezentované povrchom pre kreslenie vystavujúce 2D kontext³, ktoré sa dá použiť pre vykresľovanie grafov, kompozíciu fotiek či tvorbu jednoduchých i zložitejších animácií. Keďže značka <canvas> prichádza s HTML5, nie je ešte podporovaná staršími prehliadačmi (príkladom môže byť Internet Explorer pred verziou 9). Je štylovateľná ako ľubovoľný obrázok (okraj, ohraničenie, pozadie a pod.). Ak sa na <canvas> neaplikujú žiadne štýly, je plne transparentná.

Čo sa týka primitívnych grafických objektov, podporované je síce len vykresľovanie obdĺžnika, ale použitím ciest sa dajú zostrojiť veľmi komplexné objekty. Dajú sa kresliť čiary, oblúky, Bézierove a kvadratické krivky.

Jednou zo signifikantných črt <canvas> je schopnosť narábať s obrázkami. Obrázky z externých zdrojov môžu byť v ľubovoľnom formáte podporovanom prehliadačom samotným. Je možné dokonca použiť ako zdrojový obrázok grafické dáta iného plátna. Vykresliť obrázok sa najjednoduchšie podarí zadáním zdroj a (cesty v zmysle URL) a súradníc ľavého horného rohu požadovaného umiestnenia, pri potrebe

² Všetky opisované HTML značky budeme uvádzať v zátvorkách < a >.

³ V prípade experimentovania sa dá použiť aj 3D kontext, napríklad v prípade WebGL. Akokoľvek, podpora 3D na strane prehliadačov je zatiaľ na slabej úrovni[20], takže od myšlienkových pochodov orientovaných týmto smerom upustíme.

zmeny veľkosti je možné použiť rovnakú funkciu s pridaním ďalších dvoch príslušných parametrov. Funkcia s najväčším počtom parametrov pre vykreslenie obrázka umožňuje zobrazit' iba jeho časť – orezanie. Explicitne je možné nastaviť priehľadnosť vykresľovaného objektu, od absolútne priehľadného po úplne nepriehľadný. S cieľom dosiahnuť reálny efekt osvetlenia ponúka HTML5 tiež funkciu pre vrhanie tieňa. Z hľadiska nášho záujmu sú zaujímavé aj funkcie pre zapamätanie si stavu plátna a načítanie už zapamätaného stavu plátna. Plátno sa dá posúvať, rotovať, zväčšovať a zmenšovať, čím sa dotvára efekt približovania a vzdďľovania. Na zneviditeľnenie časti objektu vymedzením hraníc sa dá použiť uzatvorená vektorová cesta.

`<canvas>` manipuluje s obrázkami ako s **rastrovými** objektmi, čo so sebou môže priniesť určité nevýhody.

Na strane druhej tu však máme technológiu, ktorá pracuje s **vektorovou** grafikou - SVG.

SVG je jazyk vyvíjaný W3C konzorciom založený na jazyku XML pre popis 2D grafiky vo vektorovom formáte[21], ten sa nám môže hodiť, najmä ak ide o farebné vyplňanie plôch definovaných polygónom oproti plochám definovaným na úrovni pixelov prostredníctvom prvku `<canvas>`. Zhrnutie výhod porovnávaných riešení deklaruje nasledujúca tabuľka.

Tab. 9 Výhody technológií HTML5 Canvas a SVG

HTML5 <Canvas>	SVG
vysoký výkon pri kreslení v 2D	nezávislosť od rozlíšenia – umožňuje platformovo nezávislú zmenu veľkosti pre akékoľvek rozlíšenie
konštantný výkon – všetko je pixel; výkon klesá so vzrastajúcim rozlíšením	výborná podpora pri animáciách
schopnosť uložiť výsledok v PNG či JPG formáte	výborná kontrola nad každým elementom prostredníctvom SVG DOM API v JavaScripte
najvhodnejší pre generovanie rastrovej grafiky, úpravu obrázkov a operácií na úrovni pixelov	keďže SVG je XML formát, webové prehliadače implementujú prístup k SVG dokumentom lepšie v porovnaní s <code><canvas></code> elementmi; to robí zo SVG

	lepšie riešenie pre používateľské rozhrania webových aplikácií
--	--

Prameň: DEV. OPERA: SVG alebo Canvas? Výber medzi dvomi. [online]. Jún 2010 [cit. 2014-03-23]. Dostupné na internete: <<http://dev.opera.com/articles/view/svg-or-canvas-choosing-between-the-two/>>

Každá z týchto technológií, ako to už býva zvykom, so sebou prináša aj množinu nevýhod, ktorú popisuje Tab. 10.

Tab. 10 Nevýhody, ktoré so sebou prinášajú technológie HTML5 Canvas a SVG.

HTML5 <Canvas>	SVG
nevytvárajú sa žiadne DOM uzly, všetko sa pohybuje na úrovni pixelov	so vzrastajúcou komplexnosťou dokumentu dochádza k spomaľovaniu renderovania, pretože všetko, čo používa vo vysokej miere DOM, má spomaľovací účinok
pre animácie neexistuje žiadne API	SVG nemusí byť vhodná pre oblasť hier; v tomto prípade sa odporúča kombinácia technológií Canvas a SVG
slabá podpora renderovania textu	
nemusí predstavovať najlepší výber v prípade, kde kľúčová je prístupnosť	
Nevhodné pre web stránku či používateľské rozhranie aplikácií kvôli dynamike a interaktivite (vyplýva to z nevyhnutnosti prekresľovať každý element v rozhraní)	

Prameň: DEV. OPERA: SVG alebo Canvas? Výber medzi dvomi. [online]. Jún 2010 [cit. 2014-03-23]. Dostupné na internete: <<http://dev.opera.com/articles/view/svg-or-canvas-choosing-between-the-two/>>

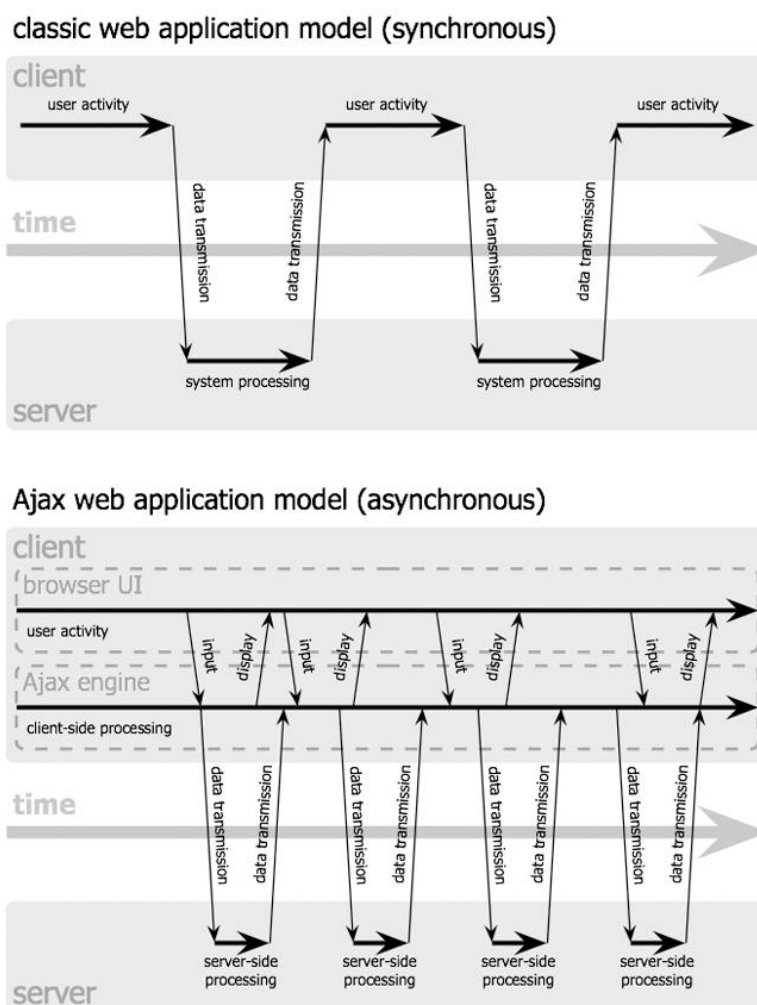
JavaScript vníma HTML dokument a prehliadač ako kolekciu určitých častí, objektov, ktoré majú detekovateľné a meniteľné vlastnosti.

Čo však posúva štandardné webové stránky bližšie k aplikáciám realizovaným ako applety je technológia AJAX. Tá predstavuje z hľadiska komunikácie medzi

klientom a serverom medzivrstvu, čo síce vedie k presvedčeniu, že doba reakcie na interakciu používateľa sa predlžuje, no podľa [22] opak je pravdou.

Medzivrstvu technologicky predstavuje *AJAX engine*, ktorý narúša „start-stop-start-stop“ prístup konvenčných webových aplikácií, a to nasledovne: namiesto načítania stránky sa načíta AJAX engine, ktorý sa stará o vykresľovanie grafického rozhrania a o komunikáciu medzi serverom a klientom – asynchrónne. Asynchrónnosť v tomto kontexte indikuje nezávislosť na komunikácii so serverom z pohľadu interakcie používateľa. Používateľ je tak úspešne „ukrátený“ o pohľad na prázdnu stránku či indikátor načítavania nového obsahu stránky.

Obr. 12 zachytáva čakanie používateľa na načítanie stránky ako myslennú vodorovnú čiaru vedúcu od konca šípky indikujúcej aktivitu používateľa po začiatok nasledujúcej šípky rovnakého charakteru vo vrstve klienta – čaká sa na spracovanie vo vrstve servera.



Obr. 12 Porovnanie modelov fungovania klasických webových aplikácií a webových aplikácií s použitím technológie AJAX[22]

Medzi ďalšie pozitíva takéhoto prístupu patrí schopnosť stránky byť prehľadávanou/nájdenu ako aj indexovanou, takisto už zmienená doba načítania stránky je kratšia než v prípade použitia appletov, pretože sa nemusia načítavať siahodlhé knižnice, dizajn je skôr webovo-orientovaný než desktopovo-orientovaný v porovnaní so skôr uvádzanými technológiami. AJAX sa dá použiť aj pri konvenčnej webovej stránke, kdežto applety je potrebné programovať od piky.

3.2 Prehľad technológií pre realizáciu online platieb

3.2.1 Úvod do elektronického podnikania

Elektronické podnikanie (e-business) je akákoľvek činnosť, ktorú organizácia vykonáva použitím počítačových sietí s cieľom obchodu[23], čo zodpovedá aj nami vytýčenému programovému riešeniu. Tento širší pojem vo svojej podstate evokuje, že elektronické podnikanie je elektronickou formou už existujúcich aktivít. Každá transakcia realizovaná prostredníctvom počítačovej siete, čo má za následok zmenu vlastníctva či práv na užívanie tovaru alebo služby je pomenúvaná ako e-komercia (e-commerce). Táto definícia predstavuje užší pojem, definuje konkrétnu transakciu pri predaji, počnúc objednávkou cez finančné vysporiadanie až po samotné dodanie tovaru zákazníkovi.

Aplikáciou Internetu v oblasti konvenčného obchodu sa získal potenciál pre vyhľadanie informácií o produkte, objednanie produktu elektronicky, platenie za služby či tovar(elektronickými peniazmi) i realizáciu zákazníckeho servisu.

Autori v [23] zaraďujú medzi ostatné vplyvy elektronického obchodu na marketing aj nami už viackrát spomínanú customizáciu, ktorá je, ako bolo v prvých kapitolách naznačené, dôvodom realizácie tejto témy. Aspekt customizácie bude pre náš modul príznačným.

3.2.2 Platobné metódy – kategorizácia, výhody, nevýhody

Možností platenia za tovar alebo služby je viacero. Môžeme konštatovať, že forma platby závisí od vzájomnej dohody medzi predávajúcim a kupujúcim. Najčastejšie typy platenia sú:

- **Platba kreditnou kartou** – ide o tzv. *priamu* platbu kreditnou kartou predávajúcemu. Negatívom je, že predávajúci sa dozvedá citlivé bankové údaje kupujúceho a musí mať takúto službu s bankou vopred dohodnutú, z čoho vyplýva aj jej spoplatnenie. Osoba, ktorá má za cieľ nakúpiť,

musí byť držiteľom karty, ktorú vydáva banka (tzv. vydavateľská banka). Predávajúca osoba musí byť evidovaná v banke, s ktorou má uzatvorenú zmluvu o prijímaní platobných kariet (tzv. zúčtovacia banka) a ktorá jej preposiela peniaze na podnikateľský účet. Transakčné poplatky sa v tomto prípade pohybujú okolo 1,5% až 4%. Príkladom pre systém platby kreditnou kartou je Visa, Visa Electron, Maestro, MasterCard, Diners Club. Zúčtovacie banky na slovenskom trhu ponúkajú zavedenie platby kartou pod nasledovnými menami: Web-Pay (ČSOB), eCard (VÚB), CardPay (Tatra banka), iTerminál (Poštová banka), WebCard (UniCredit), Bezpečná i-platba (Sberbank).

- **Peňažná poukážka a personal checks** – reprezentuje platobný systém analogický peňažnej poukážke. Mohli by sme sem zaradiť napríklad Money order od Wester Union alebo paysafecard.
 - **Money order** – takáto forma platenia je viazaná na kreditnú kartu. Pomocou webovej stránky sa môže poslať „money order“ priamo online. Výhodou sú menšie poplatky v porovnaní s platbami realizovanými kreditnou kartou priamo. Takýto typ platby podporuje eBay.
 - **Paysafecard** – v tomto prípade už ide o voľnosť v zmysle nie nevyhnutného vlastníctva kreditnej karty či bankového účtu[24]. Vo svojej podstate je to platba v hotovosti, ktorá je anonymná. Základom je výmena hotovosti za platobnú poukážku v jednej z pevne stanovených hodnôt (10EUR, 25EUR, 50EUR alebo 100EUR) na niektorom z 450 000 predajných miest. Platba prebieha zadaním jedného 16-miestneho PIN kódu alebo v prípade väčších čiastok zadaním viacerých PIN kódov (max. 10). Systém Paysafecard akceptuje cez 4 000 internetových obchodov. Niektorí autori tento prístup radia k platbám v hotovosti.
- **Hotovosť** – v zmysle online obchodovania je výhodou neprítomnosť bankových poplatkov, nevýhodou je nutnosť vopred sa dohodnúť s predajcom. Výmena hotovosti sa uskutoční napríklad poštou – listom prvej triedy – alebo poisťiteľným (cenným) listom.
- **Priamy bankový prevod** – štandardný bankový prevod z ľubovoľného bankového účtu na bankový účet predávajúceho ako služba ponúkaná

komerčnými bankami. Platba prebieha ešte pred samotným obdržaním tovaru. Čím skôr kupujúci zaplatí, tým skôr predávajúci zašle tovar. Pri cezhraničných platbách musíme brať do úvahy vyššie poplatky a časové zdržanie, i keď sa to považuje za najlacnejší spôsob platenia⁴.

- **Dobierka** – predstavuje najobľúbenejšiu platobnú metódu. Nie je potrebné vlastniť kreditnú kartu, bankový účet a pod. K platbe dochádza pri preberaní tovaru, ktorý kupujúcemu prináša balíková/zásielková služba (u nás DPD, TNT) alebo pošta (Slovenská pošta). Často krát sa dobierka uplatňuje v prípade nákupu od neznámeho obchodníka. Náklady na dobierku sú predmetom dohody medzi kupujúcim a predávajúcim, no zvyčajne sa povinnosť uhradiť poplatky prikláňa na stranu nakupujúceho, pričom platí sa zásielkovej službe buď v hotovosti alebo kartou. Zásielková služba sa potom vyrovná s predávajúcim.
- **Elektronická peňaženka** – ideálna metóda pre platby malých finančných čiastok. K registrácii nie je nevyhnutné vlastniť bankový účet, a teda registrácia nepodlieha kontrole ani overovaniu. K platbe dochádza ešte pred prebratím tovaru, pričom zostatok elektronickej peňaženky sa dá zvýšiť pomocou bežného bankového účtu, platobnej karty či inej elektronickej peňaženky. Samozrejmosťou je opačný transfer peňazí, teda aj tzv. *vybitie* na bežný bankový účet. Stanovenie minimálneho zostatku či minimálneho prevodu býva taktiež zvykom. Príkladom môže byť Skrill, PayPal, Trust, Google Wallet či Lemon Wallet.
 - **PayPal** – pôvodne Billpoint⁵, dcérska spoločnosť patriaca pod eBay, predstavuje virtuálne účty a transakcie prebiehajúce medzi nimi. PayPal môžeme v súčasnosti považovať za lídra medzi mikropłatbami. Už to nie je len o realizácii platieb pre potreby online aukcií na eBay, ale PayPal zaujal pevné miesto aj v oblasti finančného vyrovnávania pri predaji nehnuteľností, v oblasti cestovania, web designu, digitálneho obsahu či elektroniky[25], preto bude našou prioritou pokúsiť sa o zavedenie platobnej brány PayPal do nášho systému. Používajú ho právnici, lekári,

⁴ Po vstupe do Európskej Únie a zavedení IBAN formátu pre čísla účtov by v súčasnosti mali byť poplatky za cezhraničné prevody minimálne.

⁵ Istú dobu tieto platobné systémy existovali paralelne, jedna spoločnosť konkurovala tej druhej. Neskôr sa eBay namiesto toho, aby zdokonalil svoj systém Billpoint, radšej rozhodol pre odkúpenie svojho konkurenta.

požiadavky na platby sa môžu realizovať už aj z mobilných zariadení s internetovým pripojením. Momentálne je k službe prihlásených viac než 230 miliónov používateľov[26], čo predstavuje značnú používateľskú základňu. Pre členstvo je potrebná emailová adresa a heslo, platby sa realizujú temer okamžite (real time). Komunikácia so serverom je šifrovaná. Používa sa v 190 krajinách a platby sa dajú realizovať v 24 rôznych menách (prevod medzi menami za určitý poplatok, inak sú poplatky minimálne). Overenie konta je bezplatné, akokoľvek, pri registrácii sa uskutočňuje overenie vykonaním istého poplatku, ktorý je pri prvej transakcii následne používateľovi vrátený. K uskutočneniu platby sa pristupuje pomocou virtuálnych účtov, ktorými musia disponovať aj kupujúci aj predávajúci. Dobíť účet si kupujúci môže prostredníctvom bankového účtu. Novinkou je schopnosť výmeny peňazí medzi klientmi len na základe emailovej adresy či telefónneho čísla prostredníctvom mobilných zariadení. Adresát platby pritom nemusí byť vlastníkom PayPal účtu. Peniaze sa pošlú v rámci zásad bezpečnosti a súkromia s potvrdením odosielateľa.

- **Google Wallet** – mobilný platobný systém vyvinutý spoločnosťou Google umožňujúci používateľom smartfónov udržiavať v nich informácie z debetných a kreditných kárť pre platby online ako aj platby priamo v obchodoch[27]. Na tieto účely sa používa NFC a *cloud* technológia, čo má za dôsledok rýchlu, bezpečnú platbu, hoci len priložením telefónu na ľubovoľný NFC POS terminál pri pokladni. Elektronická „peňaženka“ sa dá použiť online v prípade prítomnosti „Google Wallet Buy“ tlačidla na príslušných stránkach obchodu. V prípade platby mobilným zariadením sa zneužitiu predchádza použitím štvorčíselného PIN kódu či použitím viacerých bezpečnostných vrstiev znemožňujúcim útoky počítačových kriminálnikov v prípade online platieb alebo podvodov s kreditnými kartami. Kontrolou na diaľku (*remote control*) sa poskytuje pôda pre deaktivovanie funkcií v prípade udalosti akou je krádež alebo strata smartfónu. Podobným prístupom ako má PayPal sa nevyhol ani Google, najmä keď ide o utajenie citlivých platobných informácií – čísla (kariet) platobných sú ukryté, pri platbách online je viditeľné len posledné štvorčísle. I napriek nedávnemu

uviedení technológie sa už v blízkej budúcnosti očakáva jej plná podpora na všetkých Android smartfónoch a kreditných kartách „veľkých hráčov“, čo taktiež znamená, a to vo všeobecnosti[28], nie len v prípade smartfónov, že zatiaľ sa na Slovensku sa takýto spôsob platby zatiaľ v rámci elektronických obchodov **nedá** integrovať.

- **Lemon Wallet** – mobilná aplikácia určená pre platformy iOS, Android aj Windows Phone umožňujúca nastaviť a použiť kreditnú i debetnú kartu, vernostné karty. Informácie je schopná zobrazit' ako čiarový kód čitateľný obchodníkmi. Prepojením platobnej karty s aplikáciou sa dajú kontrolovať účty priamo na mobilnom zariadení. Ochranu zabezpečuje 4-číselný PIN kód pre vstup, tzv. *passcode*, ktorý si aplikácia vypýta pri štarte a aj pri pokuse o prístup na prepojenú kartu. Priamo z aplikácie sa dá vyvolať akcia pre vyhotovenie fotky účtenky/pokladničného bloku pre uchovanie informácie o daniach či iné monitorovanie výdavkov.
- **Skrill eWallet** – predstavuje ďalšiu alternatívu k tradičným platobným metódam[29], umožňuje (registrovaným zákazníkom) realizovať online platby bez nutnosti udávania osobných finančných údajov, posielat' a prijímat' peniaze použitím len emailovej adresy a hesla. Počet registrovaných používateľov presahuje niečo cez 30 miliónov. Klient má k dispozícii predplatenú kartu *Skrill Prepaid MasterCard*, ktorá ho prepája so svojím účtom a umožňuje mu nad ním správu aj v režime *offline*[30]. Ročný poplatok za používanie je stanovený na 10 EUR. Transakcie za nákupy, či už online alebo offline, sú bezplatné.
- **TrustPay** – klient uzatvorením zmluvy získa platobný účet, v prípade dohody aj viacero účtov, vedených na jeho meno a priezvisko alebo obchodné meno[31]. Vznikom získa nárok na manipuláciu s účtom (zriadenie, zrušenie, zadávanie, zmena a rušenie osôb oprávnených disponovať s platobným účtom), ako aj s peňažnými prostriedkami na platobnom účte. Podporované sú platby kartou (*TrustCard*, podporované formáty VISA aj MasterCard), ako aj platby realizované pomocou bankových prevodov (*TrustPlatby*, 14 krajín strednej a východnej Európy v 12 menách) či Trustpay platobnou bránou[32], čo pre nás znamená, že jedinou integráciou s TrustPay je možné vyriešiť

potreby platobných tlačidiel aj platobných kariet jednotným spôsobom bez potreby ďalšej integrácie s jednotlivými bankovými inštitúciami. Na strane druhej, znamená to prispôbiť sa finančným pravidlám a platiť stanovené poplatky (poplatok za podporu platieb z kreditných kárt, transakčný poplatok za každú vykonanú platbu atď.).

- **Platobné tlačidlo** – metóda prevádzkovaná finančnou inštitúciou. K uskutočneniu platby dochádza ešte pred obdržaním tovaru, predávajúci ihneď získa informáciu o platbe. Obaja, aj kupujúci aj predávajúci, musia byť *zvyčajne* klientmi banky poskytujúcej platobné tlačidlo a používaťmi elektronického bankovníctva. Kupujúci pri online nákupe sa rozhodne pre platbu platobným tlačidlom príslušnej finančnej inštitúcie, na stránku ktorej je následne presmerovaný aj s predvyplneným formulárom. V rámci bezpečnosti sa zadávajú prihlasovacie údaje a heslo. Po vstupe do takejto platobnej brány sa kupujúci rozhodne vyplnený formulár potvrdiť alebo ho môže odmietnuť. Obchodník obvykle platí náklady vo výške okolo 1%, ale podmienky sa rôznia. Patrí sem napríklad E-comm (ČSOB), SporoPay (SLSP), ePlatby (VÚB), TatraPay (Tatra banka), E-obchod (Poštová banka), UniPlatba (UniCredit), OTPPay (OTP), WEBpay (Sberbank), mTransfer (mBank).

3.2.3 Vybrané platobné metódy a analýza ich realizácie

V rámci nášho riešenia sa pozrieme bližšie na kľb postupom, ako zaviesť možnosti pre medzinárodnú platbu pomocou PayPal účtu a pre jedno vybrané platobné tlačidlo, nech ním bude Sberbank VeBPay.

K takémuto rozhodnutiu sme sa priklonili z nasledovných dôvodov. Z analýzy vyplývajú zákonné obmedzenia pre integráciu s niektorými platobnými metódami (Google Wallet) ako i nevyhnutnosť platiť viaceré transakčné poplatky (TrustPay) či jednorazový poplatok (Skrill eWallet). Preto, s ohľadom na klientsku základňu a medzinárodný charakter, sme sa rozhodli pre PayPal. Skúsime sa v ďalšej podkapitole bližšie pozrieť na podmienky jej integrácie, implementačné detaily a iné informácie uvedené na stránkach PayPal ako i API dokumentácie.

Aj napriek tomu, že nášmu systému bohato vystačí PayPal integrácia, urobíme súhrn podmienok a pravidiel integrácie pre platobné tlačidlo WEBpay. Výber

bol ovplyvnený najmä dostupnosťou technickej dokumentácie a promptným vybavením jej dodania zo strany finančnej inštitúcie.

Podmienky integrácie platobných tlačidiel ďalších finančných inštitúcií pokrýva práca [36].

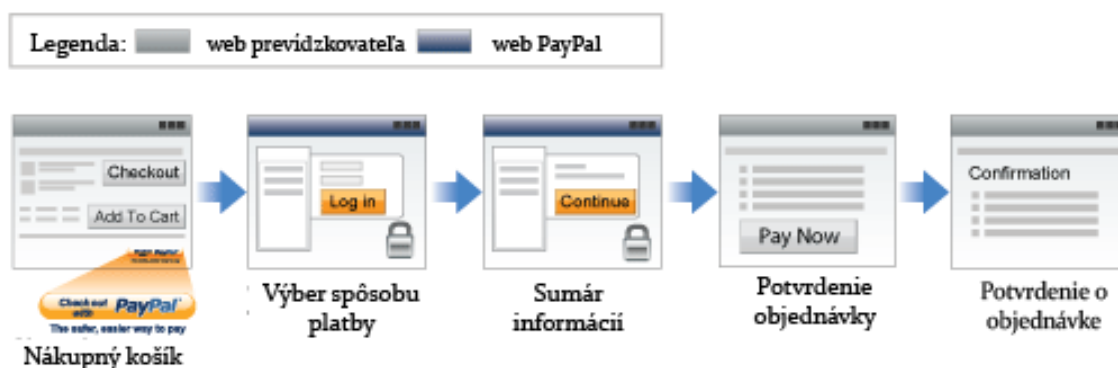
3.2.3.1 Možnosti integrácie PayPal platby

PayPal ponúka široké spektrum API volaní pre realizáciu rôznych operácií nad účtami svojich zákazníkov. My svoj záujem upriamime len na platby, ktoré pokrývajú potreby realizácie elektronického obchodu v zmysle predaja produktu a vykonania prevodu peňazí medzi PayPal účtami. Túto integráciu PayPal dokumentácie označuje ako *Express Checkout*.

Existujú dva spôsoby, ako je možné integrovať PayPal platby. Prvým krokom je však v oboch prípadoch získanie dvojice ID klienta a kľúča (tzv. Sandbox OAuth kľúče), ktorá je platná len po určitú dobu a je nevyhnutná pre realizáciu akéhokoľvek volania ďalších akcií na strane PayPal. V prípade testovania sa generuje aj heslo, to všetko sa následne zakomponováva do volaní z aplikácie komunikujúcej s PayPal rozhraním. V prípade nasadenia do produkcie je dvojicu možné získať registráciou aplikácie na oficiálnej PayPal stránke.

Prvým spôsobom predstavuje použitie tzv. *Classic API*. Hlavnými znakmi sú: prevádzkovateľ má značnú kontrolu nad priebehom realizácie platby, platobný proces je dôveryhodný, bezpečný a jednoduchý na použitie pre koncového používateľa, používajú sa prvky účtu zákazníka (použitie ID platiaceho, nemusí sa implementovať všetko od piky).

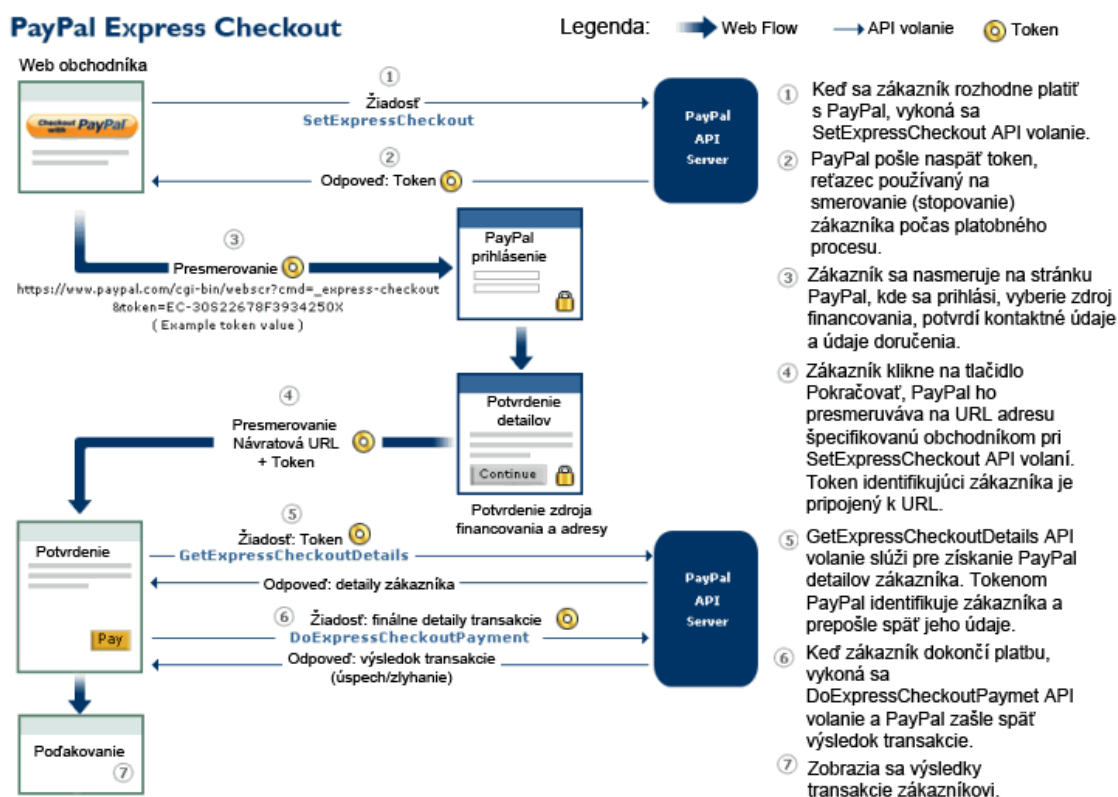
Keď sa zákazník rozhodne zaplatiť prostredníctvom PayPal, je presmerovaný na samotný PayPal s následným prihlásením sa a kontrolou detailov transakcie. Po súhlase s financovaním platby pomocou PayPal účtu je zákazník presmerovaný späť na web stránku prevádzkovateľa, kde stačí finálne potvrdiť platbu a tá sa vykoná. Priebeh takéhoto vyplatenia charakterizuje nasledujúci obrázok.



Obr. 13 Tok riadenia pri platobnom styku PayPal v prípade Web Checkout

Technicky, realizácia sa uskutočňuje pomocou zavedenia tlačidiel do vlastnej web stránky na inicializovanie platobného procesu. Následne sa po zakliknutí integrovaného tlačidla aktivuje buď *OnClick* akcia alebo *POST* akcia formuláru, čím sa volá PayPal API. Po tom, čo sa zákazník prihlási do svojho PayPal účtu, zobrazia sa mu transakčné údaje, môže zmeniť v následnom kroku svoju adresu doručenia, čomu nadchádza potvrdenie transakcie a možné presmerovanie na stránku predajcu.

Priebeh komunikácie, volané API metódy a dátové výmena je naznačená na nasledujúcom obrázku.



Obr. 14 Priebeh komunikácie a výmena dát s popisom v príslušných krokoch pre PayPal Express Checkout

Druhý spôsob predstavuje použitie **REST API**. Principiálne ide o použitie HTML prvkov, nad ktorými sú volané vzdialené metódy (servera REST ako adresáta a takisto aj odosielateľa) pomocou správ formátovaných ako objekty JSON. Pre vývojárov sú pripravené podporné knižnice v jazykoch a platformách pre Ruby, Python, Javascript (Node.js), Java, PHP, .Net, iOS a Android. Ako API autorizačný framework sa používa OAuth 2.0.

Naším potrebám postačuje Classic API, aj keď neexistuje žiaden špecifický dôvod, prečo by sme sa nemohli rozhodnúť pre implementáciu pomocou REST API. Existujúce riešenia ako [34] však pre platformu PHP vyžadujú verziu 5.3+, čo je v rozpore s našou požiadavkou P#1, preto bude potrebné navrhnúť a realizovať svoje vlastné riešenie.

3.2.3.2 Možnosti integrácie platobných tlačidiel

Keďže vo väčšine prípadov majú podmienky integrácie rovnaký charakter, popíšeme bližšie len jednu z nich, a tou je platobné tlačidlo Sberbank WEBpay.

Platobný proces prebieha takto. Po tom, čo si koncový zákazník na web stránke elektronického obchodu zvolí platbu WEBpay, je presmerovaný na aplikáciu banky. Údaje o platbe sú prenášané URL adresou, ktorá má jasne definovaný tvar. Zákazník je vyzvaný k prihláseniu sa pomocou svojich údajov, ktoré používa pre elektronické bankovníctvo. Generuje sa preddefinovaný platobný príkaz (údaje použité pri generovaní – suma, VS, KS, SS, popis k platbe – sú získané zo spomínanej URL adresy), v ktorom sa dajú meniť len položky účtu platiteľa (svoje účty, ktoré má k dispozícii) a popisu platby. Číslo účtu adresáta (obchodníka) sa nedá meniť, keďže vyplýva z uzatvorenej zmluvy a je pevne dané. Klient je o výsledku platby informovaný prostredníctvom ďalšieho okna, do ktorého je presmerovaný. Klient informáciu o výsledku potvrdí, následne je presmerovaný na aplikáciu obchodníka, obchodník je zo strany banky upovedomený o úspešnosti či neúspešnosti realizácie platby, z čoho vyplýva schopnosť obchodníka zaznamenať zmenu stavu objednávky a patrične zareagovať (pripraviť produkt, zabaliť, odoslať a pod.). Obchodníkovi je fyzicky doručená dvojica: šesťciferný kód MID a osemciferný šifrovací kľúč (možná zmena na požiadanie). Kód MID slúži k jednoznačnej identifikácii obchodníka pre potreby komunikácie medzi ním a WEBpay aplikáciou. Ako už názov napovedá, úlohou šifrovacieho kľúča je zašifrovať prebiehajúcu komunikáciu (prenášané dáta) s WEBpay. Z technického hľadiska sa komunikácia odohráva medzi servermi obchodníka a bankovej inštitúcie volaniami HTTP metód GET a POST so šifrovanou

komunikáciou. Server WEBpay aplikácie je dostupný na <https://ibs.sberbank.sk/vebpay/> a očakáva nasledovné parametre⁶:

- **MID** – identifikačné číslo prevádzkovateľa obchodu,
- **AMT** – suma v EUR, pričom desatinné čísla sú oddelené bodkou,
- **VS** – variabilný symbol ako (maximálne) desaťmiestne číslo bez medzier,
- **CS** – konštantný symbol, ak je uvedený, max. desiatich znakov bez medzier, ak nie je uvedený, je považovaný za reťazec nulovej dĺžky,
- **SS** – špecifický symbol, ak je uvedený, max. desiatich znakov bez medzier, ak nie je uvedený, je považovaný za reťazec nulovej dĺžky,
- **RURL** – URL adresa návratu, WEBpay aplikácia presmeruje klienta na danú adresu aj s pridaním návratových parametrov,
- **DESC** – maximálne tridsaťpäťznakový popis platby,
- **SIGN** – bezpečnostný prvok (podpis), ako uvádza popis vyššie.

Bezpečnostný podpis požiadavky sa získa nasledovne. Použitím hodnôt parametrov MID, AMT, VS, CS, SS a RURL sa vytvorí jeden reťazec (poradie parametrov sa musí dodržať), ten sa prostredníctvom algoritmu SHA1 prevedie na HASH, z ktorého sa v ďalšom kroku použije iba prvých 8 znakov a tie sa zašifrujú algoritmom DES pomocou bezpečnostného kľúča obchodníka. Vzniknutý osemznakový reťazec sa po konverzii do hexa zápisu mení na šestnásťznakový, čo predstavuje náš SIGN parameter pripojený k požiadavke.

Po vykonaní transakcie je klient presmerovaný na URL adresu špecifikovanú parametrom RURL v pôvodnom volaní. Používa sa HTTP metóda GET. Pripojené sú nasledovné parametre:

- **VS** – variabilný symbol príkazu platby,
- **SS** – špecifický symbol príkazu platby,
- **RES** – jedna z dvojice hodnôt OK alebo FAIL indikujúca úspešnosť platby,
- **SIGN** – bezpečnostný podpis odpovede.

Pre overenie bezpečnostného podpisu sa používa podobný postup ako v prípade uvedenom vyššie, rozdiel je len v tom, že tento krát sa používajú pre tvorbu reťazca na

⁶ Parametre označené tučným rezom písma sú povinné.

začiatku len dostupné parametre, teda VS, SS, RES (poradie ani v tomto prípade nie je náhodné, ale musí sa dodržať).

V prípade Sberbank má obchodník k dispozícii aj službu, ktorá ho o úspešnosti platby informuje notifikačným emailom alebo SMS správou presne definovaného tvaru:

SBERBANK SK PORTAL RES=OK AMT=XXX.XX CUR=EUR VS=XXX CS=XXX SS=XXX SIGN=XXXXXXXXXXXXXXXXXX, čo sa môže použiť pre potreby ďalšieho automatizovaného spracovania.

3.3 Prehľad technológií pre integráciu so sociálnymi sieťami v oblasti internetu

Sociálne siete ako prostriedky pre zoskupovanie ľudí na základe spoločných prvkov života sa stali príchodom WEB 2.0 fenoménom 21. storočia vytvárajúcim bohatú pôdu pre reklamu a marketing[37]. Ich masový rozvoj je príčinou toho, že spoločnosti hľadajú potenciálnych zákazníkov na webe. Koncom marca 2012 výskum v oblasti slovenských domácností vo vzťahu k sociálnym sieťam ukázal, že až 54% slovenskej populácie využíva sociálne siete, každý piaty Slovák (21%) pristupuje na niektorú sociálnu sieť každodenne[38]. Potenciál sociálnych sietí pre potreby rozšírenia zákazníckej základne je nepopierateľný. Čo je pre nás a náš systém nesmierne dôležité, dnes to už nie je len profesionálny jednostranný verejný monológ firiem k svojim zákazníkom, ale aj obojstranná online komunikácia v rámci vytváraných komunit s cieľom výmeny nápadov, zdieľania skúseností a podpory pri inovovaní[39]. V oblasti B2B je to tiež cesta ako podnietiť vznik povedomia o (novej) značke u zákazníkov, ako informovať o pripravovaných akciách, ako spravovať a manažovať online reputáciu, ako realizovať nábor, spoznať novinky, technológie, konkurenciu v danej oblasti a tak ďalej[40].

V rámci tejto podkapitoly sa teda pokúsime zamerať na internetu vládnuce sociálne siete a na ich potenciálnu integráciu s deklarovanou aplikáciou prostredníctvom vyvíjaného modulu.

3.3.1 Integrácia so sociálnou sieťou Facebook

Spoločnosť Facebook poskytuje rozsiahle možnosti pre integráciu s vlastnými programovými riešeniami. Poskytnutie možnosti zdieľania informácií jej používateľov s webovými aplikáciami tretích strán siaha do augusta 2006[41]. V máji 2007 došlo k rozšíreniu podpory pod názvom Facebook Platform, čím nástroje pre budovanie RIA

kooperujúcimi s Facebook-om dostali zelenú. Ďalšie novinky z dielne tvorcov spomínanej sociálnej siete uzreli svetlo sveta v máji 2008. Ich dôsledkom je možnosť použitia identity (prihláseného) používateľa na akejkolvek webovej stránke implementujúcej definované API, prítomnosť najaktuálnejších informácií o používateľovi (zmena propagovaná z Facebook-u ihneď na aplikáciu tretích strán), zvýšená podpora bezpečnosti na základe nastavení používateľa.

Na jej stránkach určených pre vývojárov⁷ je možné nájsť podporné knižnice i postupy napríklad pre začlenenie sociálnych pluginov (tlačidlá: Páči sa mi to, Zdieľať, Poslať, Vložiť príspevok, Sledovať atď.), pre manipuláciu s tzv. Open Graph⁸, pre realizáciu prihlásenia cez Facebook Login či pre vývoj hier fungujúcich na rôznych platformách.

Vo všeobecnosti, pre vývojárov sú pripravené tri základné cesty, ako integrovať Facebook do vlastných aplikácií. Prvé dve ponúkané riešenia sa bezprostredne týkajú integrácie natívnych aplikácií bežiacich pod mobilnými operačnými systémami iOS a Android. Tretie riešenie je stavané všeobecne pre webové aplikácie, v prípade ktorých sú k dispozícii na výber knižnice pre komunikáciu s Facebook rozhraním na strane klienta (Facebook SDK pre JavaScript) alebo na strane servera (Facebook SDK pre PHP). V prvom prípade ide o podporu prihlásenia sa, prístupu na API ako aj vkladanie sociálnych pluginov. V druhom prípade ide skôr o volania pre prístup k rozličným funkciám na strane servera. Ak sa webové riešenie realizuje inak, k dispozícii sú knižnice pre C#, Ruby, Flash, Python, Java a v JavaScripte napísanú knižnicu Node.js.

3.3.2 Integrácia s Google+

Podľa blogu pre vývojárov⁹ z 26.februára 2013 prichádzajú na trh Google Play služby pod verziou v3.0. Okrem vylepšeného prihlásenia do sociálnej siete Google+ (Google+ Sign-In) či máp (Google Maps) došlo všeobecne k rozšíreniu podpory integrácie pomocou verejnej API. Okrem Over-the-Air inštalácie aplikácií (možnosť inštalovať mobilnú aplikáciu priamo zo svojej webstránky), dostupnejšej manipulácii s profilom (priame nastavenie fotky na stránkach tretích strán) tu sú *interaktívne*

⁷ To jest <https://developers.facebook.com/docs/plugins/>.

⁸ Open Graph umožňuje pridávať udalosti na časovú os používateľa vo forme príbehov definovaných štyrmi charakteristikami: účinkujúci (actor), akcia, objekt a názov aplikácie.

⁹ Dostupný na <http://android-developers.blogspot.sk/2013/02/google-sign-in-now-part-of-google-play.html>.

príspevky (*Interactive Posts*, tiež nazývaný aj *Interactive Sharing Button*), ktoré nás budú zaujímať najviac.

Interaktívne príspevky umožňujú nám ako autorom webovej aplikácie prepojiť Google+ používateľa prístupujúceho na našu stránku s jeho profilom s cieľom zdieľania medzi jeho *kruhmi* (Google Circles). Kruhy predstavujú používateľom formované skupiny ľudí, ku ktorým sa zachováva jednotný prístup virtuálneho prístupu. Znamená to, že používateľ môže vypustiť informáciu v sociálnej sieti len pre istý (o)kruh ľudí (napríklad rodina, najlepší kamaráti a pod.), čo by pre nás znamenalo istý zdroj voľne šíriteľnej reklamy potenciálne medzi ľuďmi s rovnakými záujmami, ako aj medzi ľuďmi mimo kruhov (príspevok zdieľaný verejne). Interaktívny príspevok je rozšírením aj v tom zmysle, že nám umožňuje definovať štýl akcie vykonanej po kliknutí používateľom na tlačidlo na strane sociálnej siete ako aj uviesť URL adresu, na ktorú bude používateľ po kliknutí presmerovaný (*deep linking*) zo svojej mobilnej aplikácie.

Prítomnosť prvku pre zdieľanie interaktívnym príspevkom je podmienený vykonaním nasledovných krokov na platforme Google pre vývojárov ako i na vyvíjanej web stránke. Na stránke <https://code.google.com/apis/console/> je potrebné vygenerovať užívateľské ID. Na vlastnej stránke potom je potrebné načítať knižnicu vložením nasledovného skriptu v dokumente niekam ešte pred uzatvárajúcou `</body>` značkou, napríklad takto:

```
<body>
...
<script type="text/javascript">
    (function() {
        var po = document.createElement('script');
        po.type = 'text/javascript';
        po.async = true;
        po.src = 'https://apis.google.com/js/client:plusone.js';
        var s = document.getElementsByTagName('script')[0];
        s.parentNode.insertBefore(po, s);
    })();
</script>
...
</body>
```

Následne potrebujeme náš dokument obohatiť o element (vizuálne) reprezentujúci dialóg pre vyvolanie vytvorenia interaktívneho príspevku. Google+ API definuje 3 výsledné prvky: `<div>`, `<span>` a `<button>` (generované buď staticky alebo dynamicky JavaScriptom). Príklad pre statické zavedenie by mohol vyzeráť nasledovne:

```
<button
  class="g-interactivepost"
  data-contenturl="https://plus.google.com/pages/"
  data-contentdeeplinkid="/pages"
  data-clientid="xxxxx.apps.googleusercontent.com"
  data-cookiepolicy="single_host_origin"
  data-prefilltext="Engage your users today, create a Google+ page for
  your business."
  data-calltoactionlabel="CREATE"
  data-calltoactionurl="http://plus.google.com/pages/create"
  data-calltoactiondeeplinkid="/pages/create">
  Povedzte svojim priateľom.
</button>
```

Hodnotu atribútu `data-clientid` je potrebné nahradiť vygenerovaným ID používateľa a hodnota atribútu `data-calltoactionlabel` bližšie určuje charakter príspevku („vytvoriť“, „kúpiť“, „predať“, „navštíviť“, „zobraziť“). Celý zoznam dostupných hodnôt je dostupný na príslušných webových stránkach¹⁰. Podľa charakteru webovej aplikácie je odporúčané nastaviť aj hodnotu pre atribút `data-cookiepolicy`. Množina dostupných hodnôt je šesťprvková, v závislosti od komunikačného protokolu (HTTP vs. HTTPS), obmedzenej viditeľnosti v rámci servera (`https://example.com` vs. `https://sub.example.com`), obmedzeného použitia HTML5 databázového priestoru alebo cookies či portu, na ktorom server počúva.

3.3.3 Integrácia so sociálnou sieťou LinkedIn

Predpokladáme, že nami poskytnuté technické riešenie nájde uplatnenie v e-commerce, preto popri integrácii s komunitnými sociálnymi sieťami stojí za to venovať svoju pozornosť aj možnostiam integrácie so sociálnymi sieťami profesijnými, akou je napríklad LinkedIn.

LinkedIn sa považuje za najväčšiu a najznámejšiu sociálnu sieť s profesijným zameraním[42] so špecializáciou na združovanie a vyhľadávanie kvalifikovaných osôb,

¹⁰ Viď <https://developers.google.com/+/features/call-to-action-labels>.

obchodných vzťahov, pracovných odvetví[43], čo s jej viac ako dvesto miliónmi členmi[44] predstavuje výbornú základňu pre nadväzovanie kontaktov, spolupráce a výmeny názorov v oblasti interaktívneho návrhu (výrobkov).

Pozrime sa teda na to, aké možnosti integrácie teda LinkedIn API dokumentácia ponúka.

Samozrejmosťou je tlačidlo pre zdieľanie, ktoré je prítomné na danej webovej stránke a prepája ju s LinkedIn platformou, čo v podstate znamená, že všetkých používateľov, ktorí na LinkedIn sieti kliknú na vygenerovaný odkaz, budú spätne presmerovaní na spomínanú webovú stránku. Svoje opodstatnenie má aj tlačidlo pre sledovanie spoločnosti, po kliknutí na ktoré používateľ začína dostávať príspevky zverejňované touto spoločnosťou na svojej LinkedIn domovskej stránke. Profil používateľa je možné integrovať s vlastným riešením prostredníctvom pluginu *Member Profile*. Čo je ale pre nás jednoznačným prínosom, je rozšírenie možnosti registrácie či prihlásenia sa na výlučne vlastnom webovom riešení registrovaných používateľov cez registráciu či prihlásenie prostredníctvom LinkedIn. Spoločnosť si taktiež môže vytvoriť svoju vizitku bez obmedzenia použitia na svojej či na iných webových stránkach. Vizitka môže zahŕňať stručné údaje o firme, miesto pôsobenia, logo a pod. V prípade, že by náš modul bol implementovaný so stratégiou navrhnutia produktu a následnej žiadosti o prácu v spoločnosti, prostriedkom pre takúto realizáciu je tlačidlo *Apply with LinkedIn*. Firma implementujúca tlačidlo na svojej stránke môže rozšíriť základnú funkcionality o pridanie maximálne troch áno/nie otázok či zmeny vizáže tlačidla samotného. Ak si predstavíme aplikáciu, kde chceme k produktu, ktorý navrhuje určitá osoba, pripojiť aj dodatočné informácie o danej osobe, poslúži k tomuto účelu plugin *Full Member Profile*. Ak používateľ chce odporučiť produkt alebo službu ale bez samotného zdieľania na vlastnej LinkedIn stránke, môže zakliknúť tlačidlo *odporúčať*.

Pre komunikáciu s cieľom manipulácie dát uložených na strane LinkedIn servera je dostupný REST server ako aj JavaScript API, pričom JavaScript API predstavuje akýsi tenký obal štandardných LinkedIn REST rozhraní[45], čo znamená, že pre dosiahnutie rovnakého cieľa je možné použiť rôzne riešenia (t.j. priama komunikácia medzi servermi alebo zasielanie XML či JSON objektov z prehliadača klienta).

3.3.4 Integrácia s Pinterest

Gro nášho modulu pre interaktívnu kompozíciu prvkov do výsledného návrhu bude (z pohľadu bežného používateľa) predstavovať generovanie finálneho obrázka z pôvodných komponentov. Keďže výsledkom generatívneho procesu je obrázok v niektorom z bežne podporovaných formátov, stojí za úvahu pozrieť sa na kľb najznámejšej sociálnej siete pre zdieľanie obrazových dát.

Pinterest slúži na zostavovanie fotografií a obrázkov z webu do kolekcií, ku ktorým sa môže používateľ následne vracat'; je založená na vizuálnom obsahu, určená (nielen) pre ľudí, ktorí si niekedy v živote čosi vystrihovali alebo zbierali[46], čo súvisí práve s naším zámerom navrhnuté grafické riešenia uschovať, triediť, zoskupovať do zbierok a ďalej šíriť. Pinterest sa pre nás stáva zaujímavou, pretože disponuje špeciálnym mechanizmom a štýlom, ktorý ju odlišuje od predchádzajúcich sociálnych sietí so značne odlišným charakterom svojich používateľov[47].

Okrem vizuálneho zdieľania obrázkov by v prostredí elektronického podnikania bolo uvítané aj vloženie dodatočných informácií o záležitosti, ktorú si používateľ do svojej zbierky pridáva. Predstavme si, že obyčajný používateľ si do kolekcie vkladá produkt, ktorý ho zaujal na stránkach predajcu. Používateľ Pinterestu má vo svojom profile k dispozícii produkt, ktorý ho zaujal, iní používatelia tento produkt na profile daného používateľa vidia, no môžu postrádať potenciálne dodatočné informácie, napríklad o použití, výrobcovi, pôvode, cene, dostupnosti produktu, atď. Prvok uchovávajúci dodatočné informácie podľa dokumentácie¹¹ nesie meno *Rich Pin*. Tvorcovia Pinterestu na uvedený fakt mysleli a preto je už možné bližšie špecifikovať parametre či iné informácie k zdieľanému obsahu piatich typov:

- film,
- recept,
- článok,
- **produkt,**
- miesto.

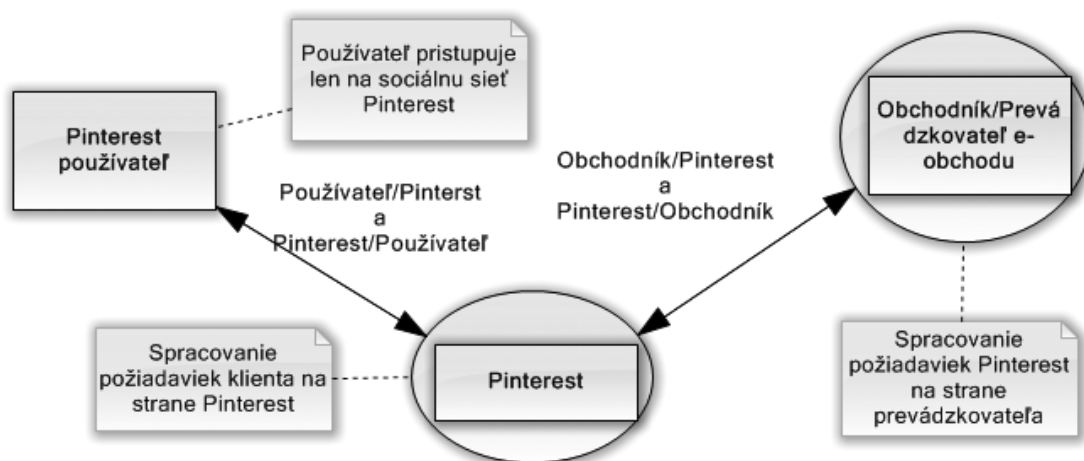
Je zrejmé, k akému potenciálu nás to privádza. Otvoria sa dvere k možnostiam šírenia sa reklamy produktov bez ďalšej námahy. Používatelia by mohli sami zdieľať vizuálny obsah z webového riešenia integrujúceho náš modul a s pridanými ďalšími

¹¹ Dostupná na https://developers.pinterest.com/rich_pins/.

informáciami o vlastnostiach produktu to bude už len krok k tomu, aby sa potenciálny kupujúci preklikol zo sociálnej siete na obchod a produkt kúpil.

Čo je teda potrebné spraviť, aby sme dosiahli vyššie popísané možnosti? Na našej strane to znamená naštudovať dokumentáciu k integrácii so sociálnou sieťou Pinterest, navrhnúť a implementovať interaktívnu tvorbu produktov tak, aby sme naštudované princípy mali napamäti, čím sa vytvorí pôda pre registráciu biznis účtu na strane záujmovej skupiny využívajúcej náš modul. Po dosiahnutí týchto úloh už nič nebude brániť k tomu, aby sa vizuálny obsah medzi používateľmi zdieľal a podporil tak predajaschopnosť produktov záujmovej skupiny.

Kontextuálny model interakcie medzi koncovým používateľom, prevádzkovateľom a službou Pinterest reprezentuje Obr. 15.



Obr. 15 Zjednodušený pohľad na interakciu medzi koncovým používateľom, prevádzkovateľom komerčného riešenia a sieťou Pinterest

Z obrázka vyplýva, že ku priamej komunikácii vo vzťahu koncový používateľ – prevádzkovateľ nedochádza, komunikačný most predstavuje Pinterest udávajúci pravidlá svojou API dokumentáciou. Pozrime sa teda bližšie na API so zameraním na Rich Pins integráciu.

V prípade Rich Pins prvkov existujú 3 metódy zavedenia do systému, a to nasledovné:

- sémantická značka *Open Graph*,
- sémantická značka *Schema.org*,
- metóda *oEmbed*.

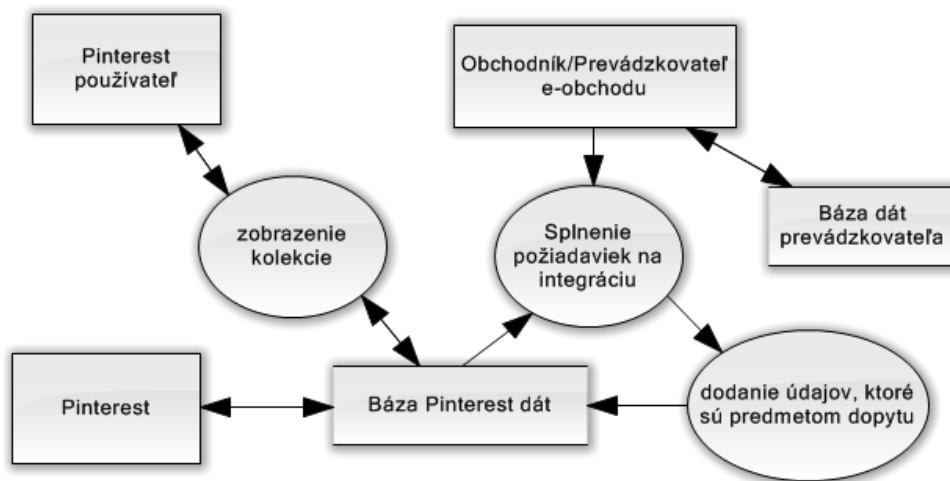
Open Graphs je štandard vyvíjaný spoločnosťou Facebook umožňujúci ponímať web stránku ako akýkoľvek iný Facebook objekt, kde namiesto viacerých rôznych technológií sa uplatňujú meta značky jazyka HTML, čo je užitočné najmä pri vychádzaní v ústrety Pinterest API v prípade, že stránka spotrebiteľa má vo svojom obsahu už meta značky zahrnuté. Keďže my výstup prezentačnej vrstvy ešte len navrhujeme a nie sme zatiaľ nijak viazaný na Facebook, touto cestou sa nemusíme uberať a viazať sa takýmto spôsobom.

Sémantické meta značky známe ako Schema.org pochádza z dielne spoločnosti Google s cieľom vložiť do webových stránok meta informácie za účelom dosiahnutia štruktúrovanosti dát bez ďalšieho dopytovania databázy[48], čo znamená, že akonáhle je web stránka raz vygenerovaná vrátane meta značiek obklopujúcich príslušné informácie, vyhľadávacie nástroje sa dostávajú k relevantným údajom jednoduchšie a poskytujú tak presnejšie výsledky. Keďže meta značky je potrebné zahrnúť (s ohľadom na požadovaný efekt) aj v rámci tela HTML stránky, v tomto prípade je zavádzanie prácnejšie. Na druhej strane je však možné parsovať údaje týkajúce sa viacerých produktov zobrazených na jednej jedinej webovej stránke. Pozrime sa na tretí spôsob, ktorý v prípade Pinterestu ešte prichádza do úvahy.

Vo všeobecnosti je metóda oEmbed formátom pre vloženie vnorenej reprezentácie URL na stránkach tretích strán umožňujúcim zobrazit' vnorený obsah ako obrázok či video na webe pomocou linku na daný zdroj bez priameho parsovania obsahu[49], čo by mohlo zodpovedať nárokom pre náš systém. oEmbed výmena prebieha medzi spotrebiteľom a poskytovateľom. Spotrebiteľ má v záujme zobrazit' na svojej webovej stránke zdroj, ktorý sa nachádza na serveri tretej strany. Ak má tretia strana zimplementovanú podporu oEmbed, potom spotrebiteľovi nič nebráni v stanovenom oEmbed formáte požiadať danú tretiu stranu o očakávanú reprezentáciu. Výmena sa uskutočňuje pomocou komunikácie založenej na JSON a XML objektoch, čo predstavuje technologicky moderný spôsob podobný spracovaniu REST serverov.

Voľbou eEmbed cesty by sme sa neviazali na žiadneho veľikána na pôde vyhľadávačov. Takisto by to pre nás znamenalo, že aj iné aplikácie tretích strán môžu posilať dopyty na náš server bez nevyhnutného parsovania obsahu generovaného prezentačnou vrstvou. Ba čo navyše, jedným volaním služby by sme vedeli propagovať informácie o viacerých produktoch naraz, čo je výhodou oproti Open Graph metóde.

Pozrime sa na to, čo nám dokumentácia bližšie k realizácii oEmbed napovie, už teraz je však jasný detailnejší spôsob komunikácie naznačený ilustráciou Obr. 16.



Obr. 16 Bližšia interakcia medzi komunikujúcou trojicou: používateľ, prevádzkovateľ, Pinterest

Na strane servera je v prvom rade nevyhnutné pomenovať a zriadiť verejnú službu, na ktorú sa budú aplikácie tretích strán vedieť posielat' svoje dopyty. Predstavme si takúto adresu: `http://www.api.domain.com/service/` s neurčeným prihlasovacím menom a neurčeným heslom, subdoménou *api*, doménou *domain.com* a službou *service*, na ktorú môžu byť smerované dopyty. Služba, ktorá požiadavky spracováva, musí byť pripravená pre akceptáciu 4 GET parametrov:

- url,
- maxwidth,
- maxheight,
- format,

kde prvý parameter je povinný a zvyšné 3 sú voliteľné. Parameter *url* predstavuje zdroj (spracovania) vložených informácií. *Maxwidth* a *maxheight* určujú maximálnu výšku a maximálnu šírku vkladaných zdrojov v prípade typov volaní pre video, obrázok (photo) a tzv. obohatený typ (rich). Pre typ „odkaz“ sú tieto parametre nepovinné, môžu byť ignorované. Pre akceptovaný URL reťazec nie je presne stanovený tvar, ale určitý vzor je potrebné dodržať (v inom prípade volanie nebude platné či vôbec podporované). Prípustné šablóny sú napríklad:

- `http://www.api.domain.com/service/*`,
- `http://www.api.doman.com/service/*/foo/`,

- `http://*.domain.com/service/`.

Posledný zo štvorice akceptovateľných parametrov *format* určuje spôsob komunikácie, t.j. očakávaný formát odpovede – JSON objekt alebo XML dokument. Pre lepšie pochopenie uvádzame príklad akceptovateľnej adresy s vyznačením parametrov tučným rezom písma¹²:

`http://www.api.domain.com/oembed?url=http://domain.com/photo/2362225867/&maxwidth=300&maxheight=400&format=json`.

Vo svojej podstate je sémantika adresy nasledovná. Služba *oembed* dostupná komunikačným protokolom *http*, načúvajúca na doméne *api.domain.com* dostáva (všetky) 4 parametre definujúce: *url* pre ďalšie spracovanie, obmedzenie výšky a šírky pre vkladaný zdroj a JSON ako očakávaný formát odpovede, a to v uvádzanom poradí.

Parametre v odpovedi, ktorá je výsledkom spracovania služby sú zachytené v Tab. 11. Povinné parametre sú v nasledovných tabuľkách zobrazené na šedom pozadí.

Tab. 11 Všeobecné parametre oEmbed odpovede

Názov parametra	Charakteristika
type	Typ zdroja z množiny {photo, video, link, rich}.
version	Číslo verzie oEmbed. Povinná hodnota je 1.0
title	Označenie vo forme textu popisujúce zdroj.
author_name	Meno autora alebo vlastníka zdroja.
author_url	URL navigujúca na autora alebo vlastníka zdroja.
provider_name	Meno poskytovateľa zdroja.
provider_url	URL navigujúca na poskytovateľa zdroja.
cache_age	Odporúčaná životná doba pre zdroj udávaná v sekundách. Používateľ môže ale nemusí použiť túto hodnotu.
thumbnail_url	URL na miniatúru charakterizujúcu zdroj. Ak je prítomný, musia byť prítomné aj parametre <i>thumbnail_width</i> a <i>thumbnail_height</i> .
thumbnail_width	Šírka miniatúry (ak miniatúra je špecifikovaná).
thumbnail_height	Výška miniatúry (ak je miniatúra špecifikovaná).

Každý konkrétny typ dopytovaného zdroja má ešte okrem všeobecných parametrov svoje parametre.

¹² Bodka za adresou nie je jej súčasťou.

Tab. 12 Vlastné parametre JSON alebo XML oEmbed odpovede v prípade typu *photo*

Názov parametra	Charakteristika
url	Zdrojová URL obrázka. Používateľ by mal byť schopný použiť URL adresu v elemente. Podporované protokoly sú HTTP a HTTPS.
width	Šírka obrázka v pixeloch špecifikovaného URL.
height	Výška obrázka v pixeloch špecifikovaného URL.

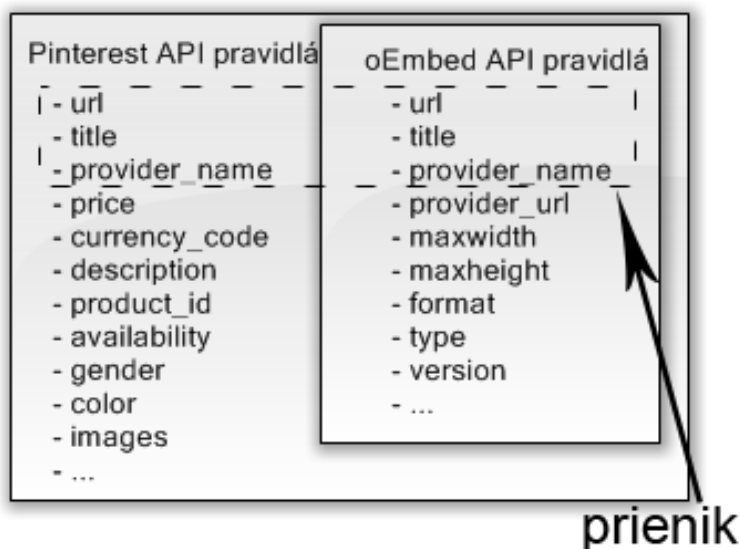
Pre typ *video* a *rich* sú vlastné parametre toho istého charakteru, obsiahnuté v Tab. 13.

Tab. 13 Vlastné parametre JSON alebo XML oEmbed odpovede v prípade typov *video* a *rich*

Názov parametra	Charakteristika
html	HTML kód pre vloženie video prehrávača alebo zobrazenie zdroja.
width	Šírka obrázka v pixeloch potrebná pre zobrazenie HTML prvku.
height	Výška obrázka v pixeloch potrebná pre zobrazenie HTML prvku.

Odpoveď typu *link* umožňuje prevádzkovateľovi vrátiť ľubovoľné vkladané dáta (všeobecnými parametrami) bez nutnosti špecifikácie parametrov URL či HTML.

Nad vyššie uvádzanými oEmbed parametrami si Pinterest následne definuje vlastné povinné aj voliteľné parametre, čím dochádza k prekryvaniu parametrov, isté parametre sú definované v oboch špecifikáciách, iné patria len jednej API, pozri Obr. 17 nižšie.



Obr. 17 Enumerácia niektorých parametrov Pinterest API a oEmbed API a ich vzájomný priemik

Najvýznamnejšie Pinterest parametre uvádzame v Tab. 14.

Tab. 14 Parametre JSON alebo XML odpovede pre Pinterest

Názov parametra	Charakteristika
price	Cena produktu bez uvedenia symbolu meny, napríklad "5.50".
currency_code	Reťazec reprezentujúci kód meny podľa ISO 4217.
description	Reťazec vyjadrujúci popis produktu.
product_id	Unikátne ID produktu na strane dotazovaného.
availability	Dostupnosť produktu. Hodnotou je reťazec z množiny {"in stock", "preorder", "backorder", "out of stock", "discontinued"}.
quantity	Celé číslo ako počet dostupných produktov.
color	Zoznam špecifikácií farieb, ak sa produkt vyrába vo viacerých farbách. Napríklad "black".
images	Zoznam maximálne 6 URL adries ukazujúcich na obrázky s vysokým rozlíšením.
gender	Reťazec popisujúci pohlavie z množiny {"male", "female", "unisex"}.

Prameň: For Developers: Rich Pins. [online]. Citované 06-04-2014. Dostupné na internete: <https://developers.pinterest.com/rich_pins/>

Príklad odpovede vo formáte JSON môže byť takáto:

```
{
  "provider_name": "FAMSF",
```

```
"url": "http://shop.famsf.org/do/product/BK5160",
"title": "The Young Copper Bookmark",
"description": "Cool copper exterior signature bookmark.",
"product_id": "BK5160",
"price": 15.00,
"currency_code": "USD",
"availability": "in stock",
"standard_price": 20.00,
"sale_start_date": 2013-12-20T00:00:00,
"sale_end_date": 2013-12-31T00:00:00,
"gender": "unisex",
"images":
["http://shop.famsf.org/storeImages/images/store/CPG-RP_1.jpg",
"http://shop.famsf.org/storeImages/images/store/CPG-RP_02_1.jpg"]
}
```

V prípade nesprávneho dopytu sa od prevádzkovateľa očakáva odpoveď obsahujúca príslušný stavový kód popisujúci vzniknutú príčinu. Nasledujúce stavové kódy a hlásenia sú súčasťou oEmbed špecifikácie:

- 404 Not Found (prevádzkovateľ nemá žiadnu odpoveď pre prijatú URL),
- 501 Not Implemented (prevádzkovateľ nevie splniť požiadavku odpovede v požadovanom formáte),
- 401 Unauthorized (URL žiadosť sa odkazuje na neverejný zdroj).

Bodku po výbere vhodnej metódy a jej následnej implementácii predstavuje validácia, ktorá sa uskutočňuje zadaním URL adresy do Rich Pin validátora na stránkach Pinterest API dokumentácie.

Za zmienku ešte stojí schopnosť realizovať určité štatistické analýzy, ktoré sa týkajú väzieb medzi webovým riešením a prvkami Pinterestu na ňom aplikovanými. Medzi štatistiky patrí zistenie 500 naposledy vytvorených Pinov viazaných na doménu, zistenie zoznamu Pinov (max. 100) z požadovanej domény zoradených podľa počtu na nich realizovaných kliknutí za určitú periódu času (denne, týždenne, dvakrát týždenne) či zistenie zoznamu Pinov, ktoré boli pripnuté (Repin) na nástenky používateľov najväčší počet krát.

4 Prehľad existujúcich riešení podobného zamerania

V tejto kapitole sa pokúsime pozrieť bližšie na už existujúce riešenia podobného typu, a to s ohľadom na všetky aspekty na vytváraný systém.

Vo všeobecnosti platí, že klient postupuje pri vytváraní vlastného produktu v konsekvencii krokov, kde každý krok pozostáva zo štylovania práve upravovaného komponentu, ktorý môže alebo aj nemusí meniť charakter výsledného produktu (napríklad, zmena charakteru opätku mení charakter celej obuvi, ale zmena farby materiálu nemá dopad na zmenu tvaru obuvi). V každom takomto kroku potom môže byť premenlivá aj množina vlastností, ktoré sa dajú aplikovať na komponent (podľa typu zvolenej kože sa mení škála použiteľných farieb kože).

Spočiatku sme plánovali rozdeliť analyzované systémy na základe aplikačného použitia z pohľadu koncového používateľa, no keďže cieľ, ktorý sa týka gra našich zámerov, sa opiera o kompozíciu grafických prvkov, rozhodli sme sa rozčleniť existujúce riešenia do troch skupín na základe vnútorného (implementačného) riešenia.

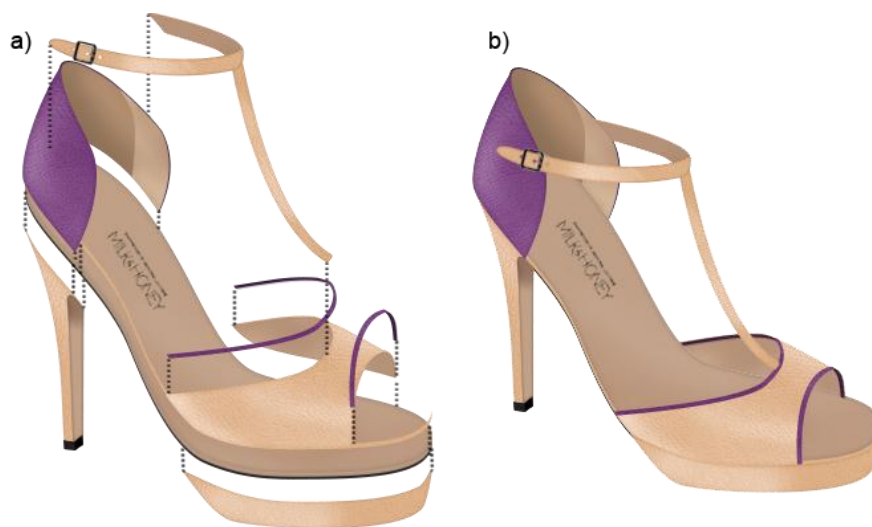
4.1 Riešenia použitím množiny HTML prvkov IMG

Ako prvé aplikácie, ktorým sa pozrieme na kĺb, budú tie, ktoré si pri svojej realizácii vystačili so základnými HTML prvkami. Opretie sa o základné prvky so sebou nesie istú nevýhodu, ktorá vyplynie z charakteristiky uvedenej nižšie.

Zástupcov skupiny môžeme charakterizovať analýzou vlastností webového riešenia na predaj obuvi, zvaného Milk&Honey. Postavené je na technológiách HTML, CSS a Javascript. Hlavným prvkom predstavujúcim prostredie pre manipuláciu a customizáciu produktov je obyčajný `<div>` element, na ktorom sú umiestnené PNG obrázky s priehľadným pozadím, plnofarebným vykreslením netransparentných častí, fixnou dĺžkou a fixným posunom. V niektorých prípadoch môže byť tento obalujúci prvok nahradený prvkom `<canvas>`. Prekrývajúcich sa obrázkov nie je veľa, základ tvorí viditeľná časť obuvi navrhovaného modelu v základnom prevedení. V prípade výberu zložitejších kombinácií sa dodávajú ďalšie obrázky prekrývajúce tie pôvodné základné. Dosahuje sa to poradím umiestnenia v rámci množiny prvkov patriacich pod `<canvas>` použitím z-indexu¹³ v príslušných kaskádových štýloch.

¹³ Z-index ako číselná hodnota predstavuje umiestnenie prvku na pomyselnéj osi Z ako normály plochy zobrazovacieho zariadenia. Čím je hodnota vyššia, tým je prvok k používateľovi umiestnený bližšie.

Takýmto spôsobom sa na pozadí z množiny obrázkov predstavujúcich reprezentácie jednotlivých komponentov (Obr. 18 - a) skladá reprezentácia výsledného produktu (Obr. 18 – b).



Obr. 18 Výsledná reprezentácia produktu (b) vzniká správnym usporiadaním obrázkov komponentov (a) s ohľadom na ich príslušnosť do určitej vrstvy

Zákazník má k dispozícii jediný pohľad na produkt, čo eliminuje množinu obrázkov potrebných na kompozíciu vlastného produktu.

Príslušným komponentom je pridelený z-index v CSS súboroch na základe ID atribútu použitého IMG prvku.

Čo sa týka platobných možností, platba je realizovateľná pomocou platobných kariet Visa, Mastercard a American Express, štandardne je použiteľný aj PayPal účet alebo bankový prevod.

Riešenie je prepojené so sieťou Facebook, Twitter, Pinterest i Tumblr. Každé prepojenie ostáva v rovine jednoduchého presmerovania na firemnú stránku na príslušnej sociálnej sieti až na prípad Facebooku, pri ktorom sa dá aj navrhovaný produkt zdieľať na nástenke používateľa.

Obdobne je riešený aj návrh obuvi pre značku Scarosso. Oproti Milk&Honey chýba platba cez Discovery Network, zdieľať navrhovaný produkt nie je možné cez Pinterest ani Twitter, no prínosom je pohľad na produkt z rôznych uhlov.

Pri návrhu outfitu ešte môžeme ostať a poukázať na Zazzle, kde sa riešenie obmedzuje na pridávanie vlastných obrázkov či textu, nad ktorým je umožnená rotácia

či zmena veľkosti. Výhodou je však možnosť umiestnenia v závislosti od požiadavky používateľa.

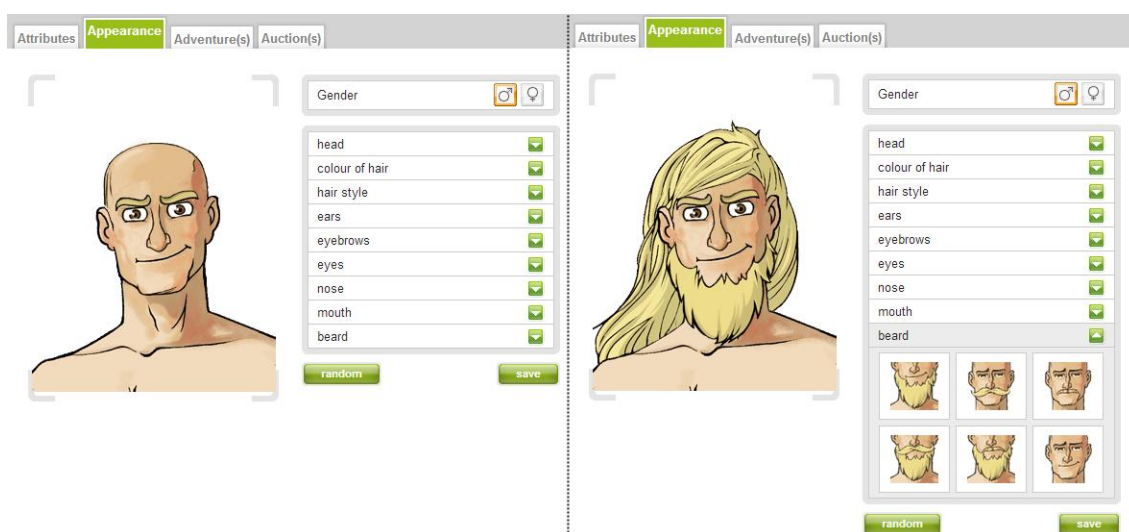
Zo slovenského trhu spomeňme výrobu košiel na mieru Grandesa, kde platobné možnosti sa opierajú o platobné tlačidlá TatraPay, SporoPay či VÚBePlatby a platbu kartou Visa či Mastercard.

Aplikačné použitie výlučne množiny IMG prvkov je rozsiahle, vidno je to aj napríklad v prípade strategickej online hry Travian, kde sa podobným postupom docieľuje interaktívna úprava vizáže hrdinu každého hráča (Obr. 19).

Ako už bolo načrtnuté, v spojení s takouto realizáciou sú nepopierateľne prítomné nedostatky alebo nevýhody, medzi ktoré môžeme zaradiť nevyhnutnosť pripravenosti – a na strane servera aj fyzickej prítomnosti – obrázkov pre každú kombináciu *komponent* a *vlastnosť*. Prakticky to znamená, že pre opäť 3 rôznych tvarov, pre ktoré sú použiteľné 4 rôzne materiály, je nevyhnutná prítomnosť 12 vopred pripravených zodpovedajúcich obrázkov na strane servera.

Ak ešte vezmeme do úvahy, že počet pohľadov nie je obmedzený iba na základný pohľad, potom tu máme ďalšieho činiteľa, ktorý pri takom malom čísle ako je 2 už dáva súčin 24 (3x4x2).

Je jasné, že pre univerzálne systémy s podporou viacerých pohľadov a zároveň so širokou škálou modifikovateľných komponentov je takéto riešenie neefektívne a neperspektívne, ak chceme dosiahnuť nezávislosť náročnosti údržby systému od počtu a rozmanitosti produktov v ňom.



Obr. 19 Aplikačné použitie skladania IMG obrázkov v prostredí strategickej hry Travian pre úpravu vzhľadu hrdinu

4.2 Riešenia použitím HTML prvkov MAP a AREA

Stále sa pohybuje v základných webových technológiach bez použitia sandboxu, tentokrát však prichádzajú do hry HTML prvky `<map>` a `<area>`, ktoré posúvajú možnosti interaktívnej tvorby produktov na vyššiu úroveň.

V predchádzajúcom prípade sa používali obrázky, ktoré síce mali transparentné pozadie, no tie časti, ktoré mali byť viditeľné, ostali plne viditeľné, teda so 100% viditeľnosťou. Prekladáním takýchto obrázkov sa časti obrázkov zoradených na nižších vrstvách (či) s nižším z-indexom nezobrazovali, ale boli plne prekryté, z čoho teda vyplývala nevyhnutnosť pre každú verziu materiálu dodať samostatnú grafickú reprezentáciu.

Použitím značiek `<map>` a `<area>` sa nám otvárajú nové možnosti pre zavedenie interakcie. Tým pádom, že nadradený prvok `<map>` umožňuje vnoreným prvkom `<area>` vymedziť plochy reprezentovaného objektu/obrázka, je k dispozícii schopnosť špecifikovať oblasti 2D reprezentácie, ktorých charakter je možné interaktívne meniť.

Skúsme si teda predstaviť grafickú reprezentáciu obuvi vo forme fotky z bočného uhla pohľadu. Prvok `<map>` bude reprezentovať danú fotku. Vložme do rodičovského `<map>` prvku jeden `<area>` objekt polygonálneho typu, a to tak, že dvojice súradníc vymenovaných v atribúte `coords` budú predstavovať polygón lemujúci podrážku obuvi. Pre ilustráciu prikladáme nasledovný obrázok.



Obr. 20 Vyznačenie polygónu lemujúceho podrážku obuvi

Je zjavné, že ak s danou plochou môžeme interaktívne manipulovať (napríklad meniť farbu ľubovoľne alebo výberom z prípustnej škály, ktorá zodpovedá možnostiam

realizácie producenta, vid' Obr. 21), značne obmedzíme nevyhnutnosť uchovávaní a prenos toľkého množstva dát na strane/zo strany servera.



Obr. 21 Ukážka potenciálu interaktívnej tvorby komponovaného produktu zmenou farby komponentu ako oblasti vyčlenenej polygómom opisujúceho príslušný komponent

Keďže, ako z obrázka vyplýva, dojem zobrazenia blízkeho realite sa stráca (kompletné prekrytie komponentu farbou), používa sa trik, ktorý je podmienený prítomnosťou obrázkov PNG formátu. Ten spočíva v tom, že buď farbou vyplnená oblasť alebo obrázok reprezentujúci komponent je rovnomerne (priehľadnosť celého obrázka je nastavená na rovnakú hodnotu) alebo nerovnomerne (svetlejšie časti sú priehľadnejšie, tmavšie si zachovávajú vyššiu viditeľnosť) priehľadný, čo pri zmene poradia vrstiev zmenou z-indexu, kde skutočná fotka obuvi upravená podľa vyššie zmieneného postupu je uložená nad plochou <area> konkrétnej plnej farby, vyvoláva dojem skutočnejšieho zachytenia reality.



Obr. 22 Priblíženie sa dojmu reálnejšieho zobrazenia zmenou viditeľnosti plochy nad upravovaným komponentom

Zástupcom skupiny je napríklad Reebok¹⁴, taktiež aplikácia na tvorbu obuvi, pretentokrát športovej. K dispozícii je značný počet pohľadov (narátaných 21), kde pri počiatočnej polohe – pri pohľade z profilu – je možné „otáčať“ produktom buď v horizontálnom alebo buď vo vertikálnom smere. Pri bližšom pohľade na implementáciu je zrejmé, že nejde o prácu s 3D grafikou ale dojem priestoru a voľnosti otáčania je dosiahnutý početným súborom pripravených náhľadov na každý komponent (pre každý komponent daného strihu obuvi maximálne 21 obrázkov¹⁵). Taktiež platí, že výberom komponentu na úpravu dochádza k automatickej zmene uhla pohľadu.

Aranžovaný produkt je možné zdieľať prostredníctvom sociálnej siete Facebook i Twitter ako aj prostredníctvom zaslania emailom priamo zo stránky aplikácie.

Platiť sa dá cez bránu PayPal, kartou Visa, Mastercard, American Express, či Discover.

Opäť môžeme spomenúť hru Travian, ktorá výhody <map> a <area> prvkov používa pri vytváraní štruktúry dediny, ktorú hráč postupne stavia, čím nahrádza voľné miesta v dedine určené na výstavbu (špecifikované práve prvkami <area>) konkrétnymi budovami.



Obr. 23 Využitie prvkov <map> a <area> v prostredí hry Travian pre explicitnú definíciu štruktúry dediny – ukážka pred a po jej výstavbe

¹⁴ Dostupná na <http://shop.reebok.com/us/custom/your-reebok>.

¹⁵ Maximálne preto, lebo ľubovoľný komponent nemusí byť viditeľný pre všetky pohľady. Príkladom môže byť jazyk obuvi, ktorý nie je viditeľný pri pohľade zospodu.

4.3 Riešenia použitím prvkov vektorovej grafiky - SVG a PATH

Táto cesta je presunom od rastrovej grafiky ku prvkom pracujúcim s vektorovou grafikou. SVG z hľadiska tvorby interaktívneho webu našich zámerov je ďalšou alternatívou pre to, čo sa dá dosiahnuť použitím `<map>` a `<area>` značiek. Tak ako nad grafickým objektom zahrnutým v prvku `<map>` sa dali vymedzovať oblasti atribútmi prvkov `<area>` v ňom vnorených, tak je možné aj v rámci prvku SVG definovať prvky `<path>` pre špecifikáciu plôch. Keďže rastrová grafika so sebou prináša nevýhody napríklad pri zmene veľkosti obrázka (viď 3.1.3), pozrieme sa na potenciálne riešenie pomocou vektorovej grafiky SVG či, lepšie povedané, kombinácie vektorovej a rastrovej grafiky.

Reprezentantom riešenia popísaného typu je webová aplikácia na návrh oblečenia dostupná na NW3TK¹⁶.

Nad množinou `<img>` obrázkov PNG formátu je implementovaná množina polygonálne definovaných ciest špecifikujúcich (farebne vyplnené) plochy vo forme prvkov `<path>` (viď obrázok nižšie) spadajúcich pod SVG prvok.



Obr. 24 Ukážka zobrazenia športovej bundy, ktoré vzniklo vypustením `<img>` uzlov v DOM HTML dokumentu, zobrazujúce množinu SVG PATH elementov popisujúcich štylovateľné komponenty produktu

¹⁶ Dostupná na <http://nwt3k.com/>.

Cesty sú definované tak, aby svojím umiestnením presne opísali komponent produktu uloženého akoby na nižšej vrstve. Charakter výplne plochy, ktorú tá-ktorá cesta definuje, je určený <path> atribútom *fill*. Výberom zo škály prípustných farieb je potom možné dynamicky meniť atribút – farbu plochy, čo v spolupráci s čiastočne priehľadným obrázkom charakteru komponentu (Obr. 22, prvá a druhá sekcia zľava) vytvára opäť dojem vizuálne reálne vyzerajúceho objektu (Obr. 22, tretia sekcia). PNG obrázky zachovávajú taký istý princíp týkajúci sa priehľadnosti ako to bolo v prípade predchádzajúceho riešenia.

Výhodou je fakt, že zmenou veľkosti nedochádza ku skresleniu, čo by sa dalo využiť napríklad pri detailnejších náhľadoch na objekt.



Obr. 25 PNG obrázok v dvojakom prevedení s priehľadným a bielym pozadím (prvé dve sekcie) ako fotografia zachytávajúca skutočnú bundu a jeho umiestnenie nad vrstvou PATH elementov (tretia sekcia, tmavé pozadie z aplikácie) s cieľom dosiahnutia dojmu reálneho zobrazenia

Akokoľvek, pri realizácii priblíženia bude nevyhnutné zmeniť aj veľkosť obrázka pod SVG vrstvou, čo sa bude musieť riešiť napríklad dodaním obrázka vo väčšom rozlíšení s cieľom vyvolať dojem skutočného priblíženia. Táto nevýhoda ostáva prítomná v oboch analyzovaných prípadoch.

4.4 Zhrnutie

Aplikácií podobného zamerania je na trhu mnoho, no každá z nich je charakteristická (len) určitými črtami, žiadna nepredstavuje komplexné riešenie z hľadiska voľnosti a slobody návrhu či rozmanitosti použitia, čo vyplýva buď z jej

aplikačného použitia alebo obmedzení v oblasti výroby daného produktu. My sa však pokúsime navrhnúť takú technologickú realizáciu, ktorá vyhovie všetkým požiadavkám na dynamický návrh výrobku prostredníctvom grafického rozhrania vrátane čo najväčšej integrácie so sociálnymi sieťami rôzneho zamerania, a to s použitím natívnych webových technológií a kombináciou rastrovej i vektorovej grafiky.

Existujúcim softvérovým produktom ako sú Nike, Shirtinator, GelaSkis či Skinflips, založeným na technológiách tretích strán, čo predstavuje nevyhnutnosť dodatočných inštalácií, keďže programy sú spúšťané v osobitnom prostredí, ktoré nevyhnutie vyžadujú inštaláciu dodatočných programov na strane zákazníka[33] už bližšie nevenovali. Uprednostnili sme preto bližšie zameranie na použitie natívnych webových technológií a využili priestor pre analýzu riešení ako Reebok, Milk&Honey, Scarosso a pod.

V nasledujúcich kapitolách sa pokúsime o návrh realizácie grafického plátna pre potreby skladanie komponentov. Zamyslíme sa nad tým, z akých komponentov ako takých bude pozostávať vizuálna reprezentácia plátna, konceptuálnym zmýšľaním v sfére objektovo-orientovaného prístupu priradíme komponentom plátna ich reprezentáciu v rámci modelu – vytvoríme konceptuálny model entít systému. Následne definujeme vlastnosti entít, možnosti manipulácie s týmito vlastnosťami a vzájomné vzťahy medzi entitami. Keďže náš modulárny návrh rešpektuje fakt, že akcie nad plátnom budú vykonávané rôznymi druhmi používateľov, budeme musieť vziať do úvahy rôzne pohľady na ich manipuláciu. Stavy entít sa budú môcť meniť, interakcia bude ošetrená istými pravidlami zamedzujúcimi totálne svojvoľnú manipuláciu z pohľadu grafického rozhrania i z pohľadu aplikačnej logiky.

S cieľom popísať vyššie uvedené aspekty aj naďalej budeme používať modelovací jazyk UML.

Druhá kapitola bude venovaná téme integrácie pluginov ponúkaných sociálnymi sieťami s našim systémom. V rámci hľadania riešení sa oprieme o poznatky získané z patričných knižných zdrojov ako i API rozhraní sociálnych sietí, čo predstavuje gro bázzy informácií pre uskutočnenie uvedeného rozšírenia. Už zo samotnej analýzy vyplýva niekoľko faktov, na ktoré bude treba dbať a prispôbiť tak návrh.

Tretiu kapitolu zasväťme odpovedi na otázku, akým spôsobom je potrebné vyriešiť zavedenie platobných metód v rámci našej vyvíjanej aplikácie. Potrebám

integrácie prispôbíme náš návrh a deklaruujeme naštudovanie a pochopenie oblasti realizáciou integrácie vybranej platobnej metódy.

5 Návrh grafického komponentu pre customizáciu navrhovaných produktov

Základným predmetom interaktívnej manipulácie bude objekt predstavujúci záchytný celok pre aplikáciu ďalších objektov. Budeme ho nazývať plátnom a reprezentovať ako upravovaný produkt – `CustomizedProduct`.

5.1 Plátno

Plátno si môžeme predstaviť ako tabuľu bez akýchkoľvek vlastností, ktorej grafická reprezentácia v tom najjednoduchšom prípade bude mať špecifikovaný iba svoj tvar (pre jednoduchosť nech je obdĺžnikovitý) a vyskytne sa na obrazovke v procese dizajnu ako jediná svojho druhu. Takouto výhodou je jasné určenie medzí, v ktorých sa pridávané prvky môžu pohybovať.

5.2 Základný produkt

Vnútorne bude plátno predstavovať kolekciu, ktorej jedným prvkom bude základný produkt (`BasicProduct`), na ktorý bude používateľ umiestňovať ostatné komponenty (návrh komponentov je popísaný v 5.3). Základný produkt používateľ vyberie z množiny základných produktov (F#4). Je zrejmé, že návrh základného objektu následne ovplyvní možnosti komponovania zložitejších objektov, aby bola splnená požiadavka F#7 a F#8.

5.2.1 Reprezentácia základného produktu

Pri používaní nášho modulu sa prevádzkovateľ sám rozhodne, akým spôsobom možnosť podpory zobrazenia z viacerých strán uplatní, čomu sa bude musieť patrične prispôbiť a pripraviť tak dáta pre zavedenie do systému. Na jednej strane to bude znamenať viacnásobné uloženie reprezentácií pre ten istý produkt či komponent, na strane druhej dátový priestor ušetríme zavedením prvkov vektorovej grafiky, čím eliminujeme nevyhnutnosť ukladania grafických dát pre každú kombináciu *pohľad na komponent* a *použiteľný materiál*.

Pri zavádzaní základného produktu do systému prevádzkovateľ dbá však na to, aby sa okrem rastrovej grafickej reprezentácie (F#1) dodala do systému aj logická reprezentácia *upravovateľných častí* (F#2).

5.2.1.1 Rastrová reprezentácia základného produktu

Rastrová reprezentácia (`RasterRepresentation`) pre jednu inštanciu produktu je len jedna, vtedy a len vtedy, ak na systém neexistuje požiadavka pre pohľad na produkt z viacerých uhlov. Ak takáto požiadavka existuje, dosiahne sa dojem 3D priestoru, aj keď priamo prácu v 3D sme vylúčili¹⁷. V prípade, že pohľadov je viac, pre každý pohľad je potrebné dodať vlastnú rastrovú reprezentáciu, a to práve jednu.

5.2.1.2 Vektorová reprezentácia základného produktu

Logická reprezentácia (`VectorRepresentation`) upravovateľných častí finálneho produktu (netreba si to mýliť s komponentmi aplikovanými na produkt) sa pre jednu inštanciu môže – a ak bude prítomná, tak sa pravdepodobne aj vo väčšine prípadov bude – vyskytovať vo viacerých inštanciách. Vyplýva to z faktu, že pre zobrazenie produktu stačí jeden obrázok, ale pre špecifikovanie jeho upravovateľných častí už je potrebných vektorových obrázkov viacero. Pre lepšie pochopenie sa vráťte naspäť na ilustráciu Obr. 24 a Obr. 25.

Definície prvkov vektorovej grafiky budú uložené v relačnej databáze, a to vo forme textu (SVG element `PATH`), v porovnaní s oveľa náročnejším uložením rastrových dát s ohľadom na pamäťové nároky.

5.3 Komponenty

Triedu pre komponent pomenujeme `Component`. Pridávanie komponentov má mať v právomoci ako koncový používateľ, tak aj prevádzkovateľ obchodu (F#13), klientove návrhy však musia prejsť procesom schválenia (F#14). K tomu budeme potrebovať stavy, v ktorých sa navrhovaný komponent môže ocitnúť.

5.3.1 Stavy komponentov

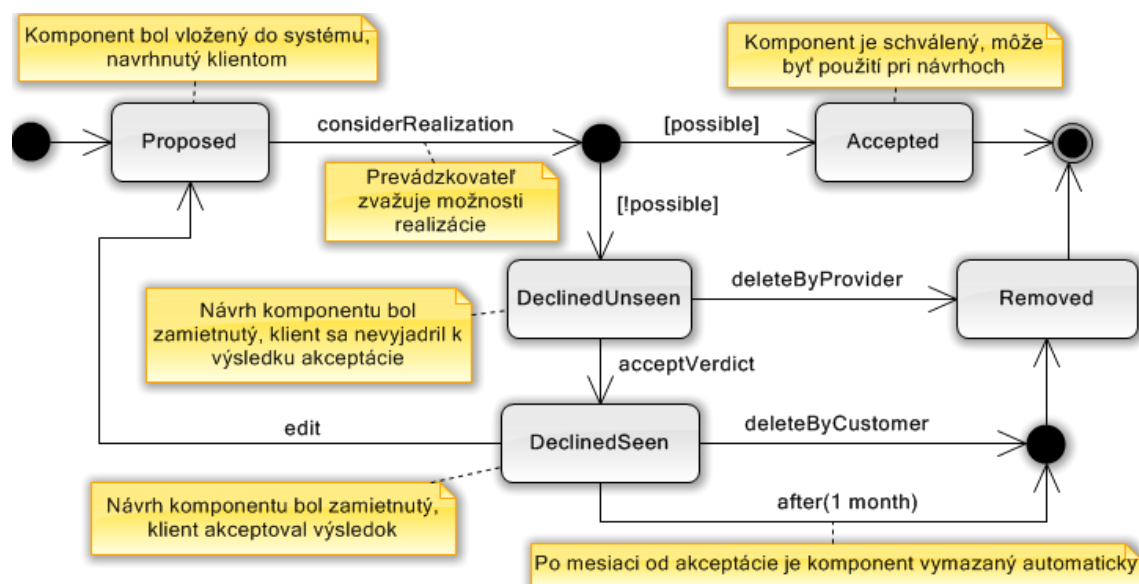
Komponent sa pridaním do systému ocitá v stave *PROPOSED*, v prípade, že jeho tvorcom je zákazník. V prípade, že návrhárom pridávaného komponentu je prevádzkovateľ, jeho stav je automaticky inicializovaný na *ACCEPTED*. Iba komponent v stave *ACCEPTED* môže byť použitý pri vytváraní produktov. Zo stavu *PROPOSED* do stavu *ACCEPTED* sa dostáva procesom schválenia. Ak komponent neprechádza úspešne procesom schválenia, nadobúda stav

¹⁷ Pozri 3.1.3.

DECLINEDUNSEEN, a to do doby, pokiaľ sa s týmto stavom neoboznámí jeho tvorca – prechod do *DECLINEDSEEN*.

Iba v prípade, že komponent bol zamietnutý, môže byť vymazaný zo systému. A to buď tvorcom na základe vlastného rozhodnutia, alebo systémom po uplynutí stanovenej doby alebo aj prevádzkovateľom, a to v stave, keď zákazník výsledok schválenia svojho návrhu nevidel¹⁸. (F#15). Klient môže navrhovaný produkt upraviť a požiadať o obnovenie procesu schválenia.

Pridaním komponentu musí dôjsť k prepočítaniu ceny vytváraného komponentu, aby sa naplnila požiadavka F#9. Cena bude jeho atribútom, určuje ju tvorca (prevádzkovateľ má nástroj na odmietnutie, ak je cena nereálna).



Obr. 26 Stavový diagram pre komponent v procese akceptácie

5.3.1 Reprezentácia komponentov

Takisto ako aj finálne produkty, aj komponenty budú môcť mať dve formy reprezentácie, rastrovú a vektorovú. Rastrovú stanoví tvorca, vektorovú grafik.

5.3.2 Množina prípustných komponentov

Pre komponenty platí, že sa budú môcť vyberať z istej množiny, ktorá to dovoľuje. Nazvime túto množinu *množinou prípustných komponentov* (MPK). MPK bude zjednotením 2 množín. Prvou množinou bude množina komponentov, ktorých použitie je prípustné z hľadiska ich charakteru (predíde sa napríklad tomu, aby sa na rukáv

¹⁸ K takémuto rozhodnutiu dospievame na základe predpokladu, že prevádzkovateľ je schopný rozhodnúť o tom, dokedy je vhodné ponechať zamietnutý produkt v systéme. Ak zamietnutý produkt nebol návrhárom zhladený a od procesu schválenia ubehol polrok, niet dôvodu na to, aby sa takýto komponent v systéme ďalej evidoval (nezáujem tvorca).

nemohla umiestniť kapučňa). Druhou množinou bude množina, ktorú definuje návrhár produktu alebo (mení) administrátor systému s cieľom uchovať v systéme komponenty, ktoré sú výlučne vlastné, iným používateľom nedostupné (keďže sme sa rozhodli pre stratégiu obohatenia bázy systému o koncovým používateľom navrhované komponenty, požiadavka F#5).

5.3.3 Príslušnosť komponentov ku kategóriám

Samozrejme, komponenty budú v systéme zložkami aj iných množín, v závislosti od potrieb systému a uhla pohľadu na ne. Môžeme uvažovať o množine komponentov, ktoré majú spoločné aplikačné použitie (vrecká, zipsy, lemy rukávov), výrobný materiál (vlňa, bavlna, polyester), veľkosť (napríklad kapucne a traky veľkosti L), môžu byť určitého, niekým pomenovaného druhu (puzdro kladivové, tajné, USB puzdro, puzdro pre mobil) a pod. Takéto zložky nazveme kategóriami - `ComponentCategory`.

5.3.4 Triedy komponentov vzhľadom na možnosti ich umiestňovania

Komponenty budú dvoch tried, vychádzajúc z možností ich umiestnenia.

5.3.4.1 Komponenty bez možnosti zmeny umiestnenia

Komponenty prvej skupiny budú také, ktorých umiestnenie bude definované explicitne, budú statické (`StableComponent`) ako v prípade väčšiny analyzovaných riešení. Ich umiestnenie bude pevne dané a nebude premenlivé. V prípade statického umiestnenia sa nebudeme musieť príliš zaoberať kontrolou umiestnenia (môžeme sa spoľahnúť na zdravý rozum administrátora systému).

5.3.4.2 Komponenty s možnosťou zmeny umiestnenia

Nech komponenty druhej skupiny budú pozične flexibilné (`MovableComponent`, jeden z chýbajúcich nedostatkov existujúcich riešení, F#6), dynamicky umiestniteľné s ohľadom na komponent či základný produkt na nižšej vrstve tak, aby umiestňovaný komponent bolo vo fáze produkcie možné pripevniť na výsledný produkt („nesmie lietať vo vzduchu“).

V prípade flexibilnej manipulácie s miestom aplikácie komponentu je potrebné zvážiť, akým spôsobom budeme môcť predísť potenciálnemu nevhodnému nasadeniu komponentu. Nič nám nebráni v tom, aby sme zimplementovali viacnásobnú kontrolu akcie používateľa, a to v zmysle jednak kontroly navzájom prekrývaných objektov špecifikovanými reláciami (opäť potrebná definícia vzťahov, predchádzanie situácii,

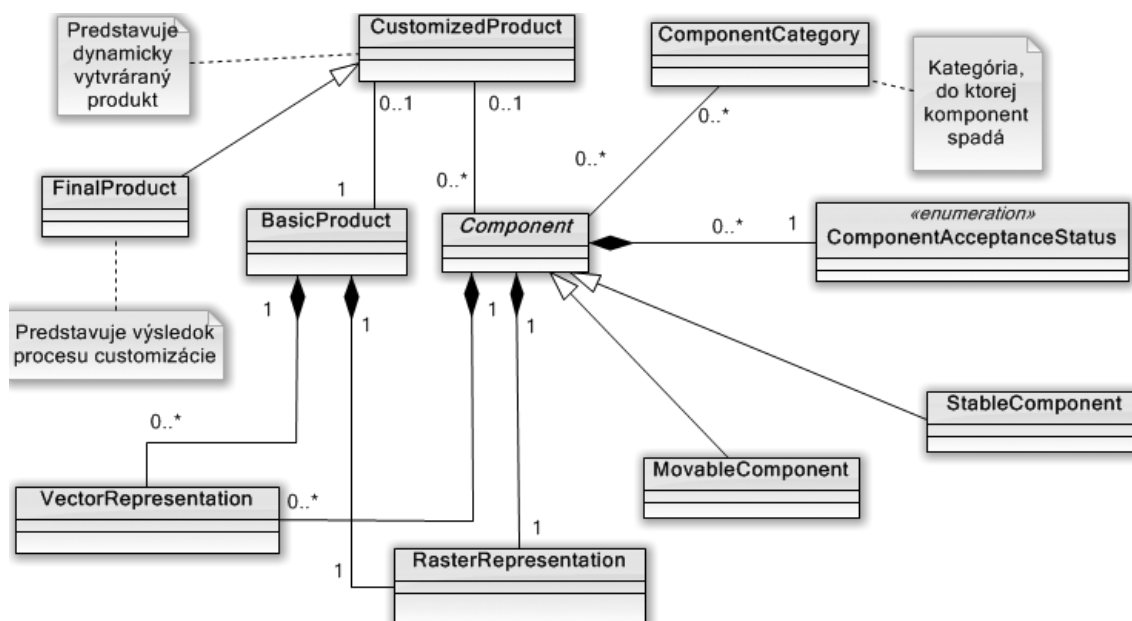
kde by mohol byť, napríklad, gombík umiestnený na zips¹⁹), ako aj neautomatizovanej kontroly v procese spracovania samotným administrátorom systému.

5.4 Výsledný produkt

Pre uloženie výsledku customizácie (F#10) budeme potrebovať zaviesť ďalšiu triedu do návrhu. Pomenujme ju *FinalProduct*. Vo svojej podstate je triedou *CustomizedProduct*, len už na nej neprebíha návrh. Produkty, ktoré sú výsledkom dizajnu budú prístupné aj pre iných používateľov (F#11), a to po úspešnom procese ich schválenia. Tým pádom, že trieda *FinalProduct* dedí od *CustomizedProduct* uchovaním informácií o aplikovaných komponentoch a ich charakteristikách, máme vyriešenú aj otázku dodatočných úprav vytvorených produktov inými používateľmi (F#12).

5.5 Návrh tried pre modul interaktívnej tvorby produktov

Návrh entít systému a ich vzájomných vzťahov pre podporu customizácie produktov zachytáva nasledovný diagram.

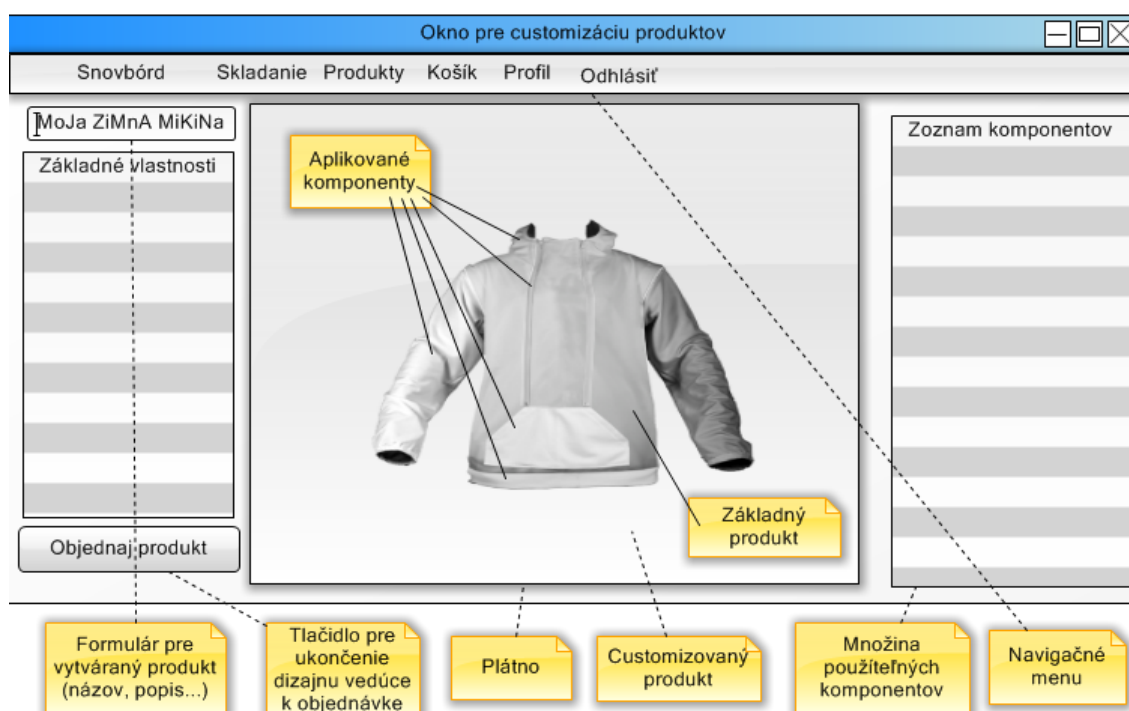


Obr. 27 Diagram tried pre modul interaktívnej tvorby produktov

¹⁹ Čo by v prípade neprítomnej kontroly relácií kontrola prekrytia sama o sebe povolila.

5.6 Používateľské rozhranie pre potreby skladania komponentov

S ohľadom na požiadavky uvedené v Tab. 3 uvádzame, ako by mala vyzeráť obrazovka pre skladanie produktov. Na ľavej strane nech bude panel so základnou charakteristikou vytváraného produktu, gro záujmu nech predstavuje plátno umiestnené v strede obrazovky a na pravej strane nech sa nachádzajú dostupné komponenty vo forme zoznamu (požiadavky PR#7 a PR#8). Návrh samotného používateľského rozhrania pre customizáciu by mohla predstavovať nasledovná ilustrácia.



Obr. 28 Návrh používateľského rozhranie pre okno interaktívneho dizajnu

Z uvedenej ilustrácie vyplýva aj prítomnosť a rozmiestnenie položiek menu, čo vyhovuje požiadavkám PR#3 a PR#4.

6 Návrh knižnice pre integráciu PayPal platieb

V poradí siedma kapitola pojednáva o návrhu podpornej knižnice pre realizáciu PayPal platieb.

Vychádzajúc z analýzy uvedenej v 3.2.3.1 navrhujeme rozhranie pre definíciu metód zodpovedajúcich PayPal API volaniam tak, ako to spomínaná analytická časť uvádza.

6.1 Rozhranie pre deklaráciu volaní vzdialených PayPal metód

Rozhranie pomenujeme `IPayPal` a jeho metódy a názvy príslušných metód API volaní nech sú nasledovné:

- `CallShortcutExpressCheckout` – volanie neúplnej PayPal API metódy `SetExpressCheckout` len pre získanie tokenu pri autorizácii (nie nevyhnutná, poslúži skôr na testovacie účely),
- `CallMarkExpressCheckout` – volanie PayPal API metódy `SetExpressCheckout` s uvedením všetkých potrebných parametrov,
- `GetShippingDetails` – volanie PayPal API metódy `GetExpressCheckoutDetails` pre získanie detailov zákazníka (adresa, kontakt),
- `ConfirmPayment` – volanie PayPal API metódy `DoExpressCheckoutPayment` pre získanie detailov transakcie.

Popísané rozhranie nech realizuje `PayPalImpl`.

6.2 Autorizačné PayPal údaje

Pre uskutočnenie komunikácie a dátovú výmenu je potrebné zahrnúť do volaní aj autentifikačné údaje dopytujúcej aplikácie (pozri 3.2.3.1), čo bude mať na starosti trieda `PayPalUserSettingsHolderImpl` implementujúca rozhranie `IPayPalUserSettingsHolder` s metódami pre získanie mena, hesla a signatúry.

6.3 URL adresy pre lokalizáciu PayPal servera

API zároveň definuje adresy, na ktoré sa je potrebné vo volaniach odvolávať a to podľa toho, či nasadenie zodpovedá testovacím účelom alebo reálnym dopytom.

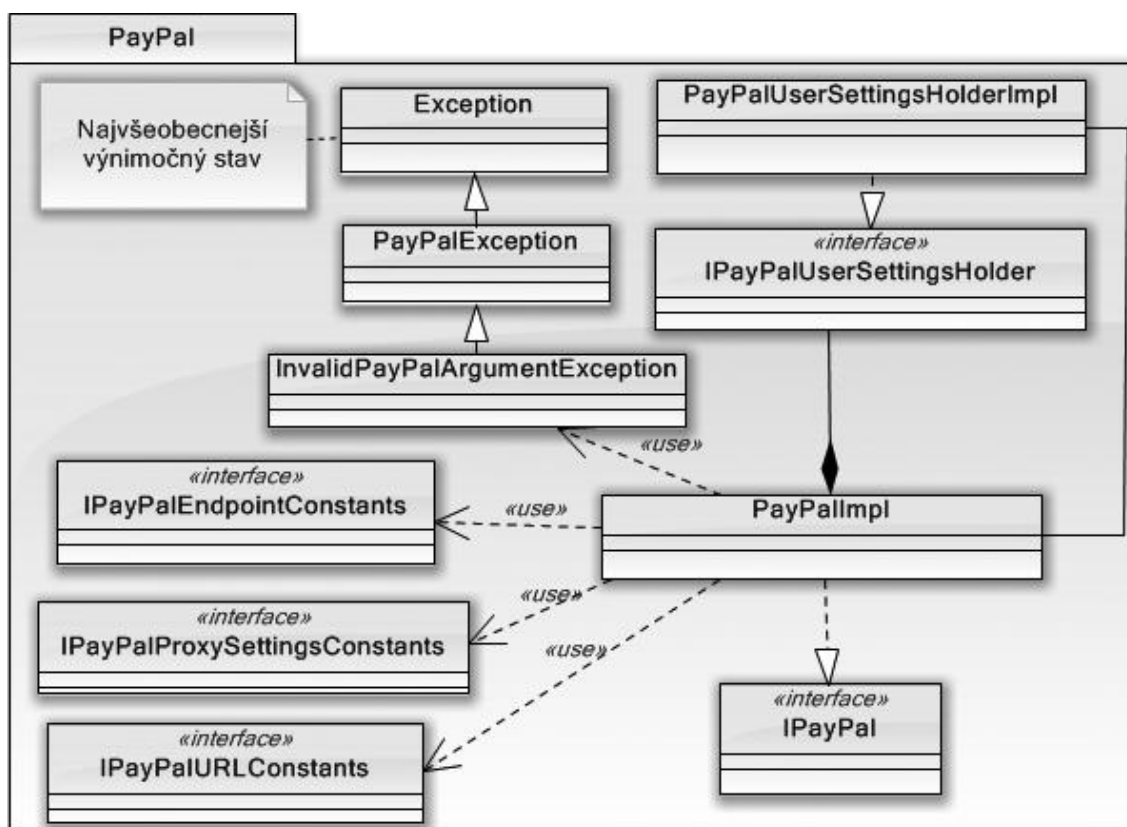
Zavádzame tak dve entity ako držiteľov týchto konštánt: PayPalEndpointConstants a PayPalURLConstants.

Rozhraním pre zachovanie nastavení v prípade proxy volaní bude PayPalProxySettingsConstants s adresou, portom a príznakom použitia.

6.4 Výnimočné stavy

V prípade inicializácie PayPalImpl potrebujeme prostriedok na propagovanie výnimočnej situácie zadania nesprávnych parametrov, na čo posluží výnimka InvalidPayPalArgumentException. Všetky výnimočné situácie v knižnici nech sú obalené výnimkou PayPalException.

Návrh všetkých tried knižnice a ich vzájomných vzťahov reprezentuje uvedený obrázok.

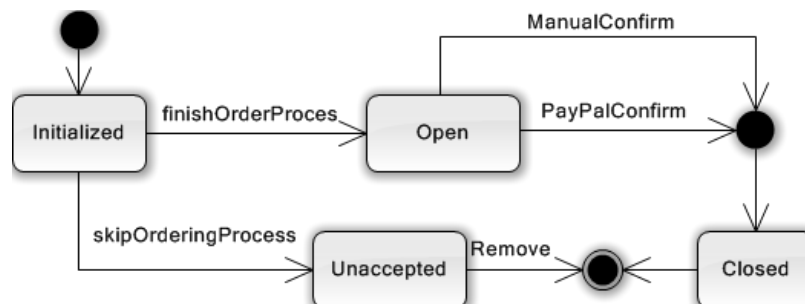


Obr. 29 Diagram tried popisujúci návrh knižnice pre realizáciu PayPal platieb

6.5 Stavy objednávky

V štádiu prijatia a spracovania objednávky v súvislosti s formou platby môže existovať viacero stavov, v ktorých sa objednávka nachádza. Pri vytvorení objednávky nech je objednávka v počiatočnom stave, nazvime ho INITIALIZED. Po súhlase

s obchodnými podmienkami, spôsobom doručenia a podobne nech je jej stav zmenený na OPEN. Zaplatením za tovar nech nadobúda stav hodnotu CLOSED.

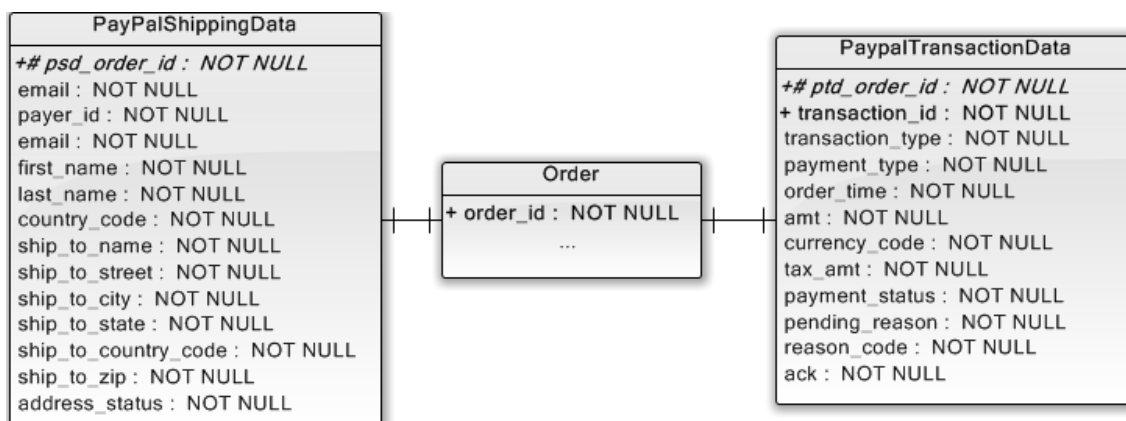


Obr. 30 Stavový diagram reprezentujúci jednotlivé stavy objednávky s ohľadom na spracovanie platby

Tu nastáva vetvenie procesov z pohľadu automatizácie uloženia stavu objednávky. V prípade, že klient si zvolí platbu na dobierku, je najsamprv nutné postarať sa o doručenie produktu, kdežto pri výbere platbou integrovanej platobnej metódy, ku platbe dochádza ako k prvej. V prípade, že zákazník teda platí až pri prebraní tovaru alebo cez internetbanking, musíme zabezpečiť nástroj pre manuálne nastavenie objednávky ako zaplatenej. Prechody medzi stavmi zobrazuje diagram na Obr. 30.

6.6 Uchovanie dát transakcií

Pre potreby monitorovania úspešnosti volaní vzdialených API metód navrhujeme dve jednoduché databázové tabuľky pre ukladanie prenášaných údajov. Každý riadok oboch tabuliek bude priradený konkrétnej objednávke, ktorej sa platba týka. Zavedenie tabuliek nie je nevyhnutné, ale poslúži svojmu účelu aj v procese vývoja aj v procese nasadenia do produkcie. Ich atribúty a vzťah k objednávke znázorňuje Obr. 31.



Obr. 31 Návrh databázových tabuliek pre uloženie záznamov ako výsledkov PayPal API volaní

7 Návrh knižnice pre realizáciu integrácie s Pinterest

V tejto podkapitole navrhujeme podpornú knižnicu pre realizáciu prepojenia s Pinterest sociálnou sieťou.

Výsledkom analýzy boli tri cesty, ktorými by sa pre dosiahnutie integrácie dalo uberať. Z dôvodov už uvedených v 3.3.4 sme sa rozhodli pre metódu oEmbed.

Metóda oEmbed je integračným nástrojom, no treba mať napamäti fakt, že je len technologickým základom, na ktorý Pinterest API nadväzuje (pre pripomenutie pozri 3.3.4). Znamená to, že pri návrhu knižnice musíme okrem špecifikácií stanovených v dokumentácii oEmbed vziať do úvahy aj pravidlá pre rozšírenie z uhla pohľadu Pinterest API, ktoré stáva akousi obálkou.

7.1 Návrh knižnice pre realizáciu oEmbed volaní

Začnime špecifikáciou tried pre reprezentáciu oEmbed odpovedí.

7.1.1 Reprezentácia oEmbed odpovedí

Odpovede oEmbed budú reprezentovať inštancie triedy `OEmbedResponseMessage` implementujúcej rozhranie `IOEmbedSendable`. Obsah odpovede generovanej na základe požiadavky používa pevne stanovené definície reťazcov, ktoré sú v prípade JSON formátu kľúčmi a v prípade XML formátu názvami elementov a ich hodnoty sú určené výpočtom na strane poskytovateľa (až na jediný prípad kľúča *version* sú ostatné hodnoty voľne ponechané na poskytovateľa). Z tohto faktu vyplýva návrh niekoľkých tried pre stanovenie hodnôt pre polia kľúčov.

7.1.2 Hodnoty kľúčov oEmbed odpovedí

Polia kľúčov nech sú reprezentované nasledovne:

- `OEmbedRequiredResponseKeyConstants` – stanovuje hodnoty kľúčov/elementov povinných parametrov,
- `OEmbedOptionalResponseKeyConstants` – určuje hodnoty kľúčov/elementov nepovinných, voliteľných parametrov,
- `OEmbedVersionValueConstants` – definuje jedinú konštantu prislúchajúcu parametru pre špecifikáciu oEmbed verzie,

- `OEmbedTypeValueConstants` – definuje štyri konštanty ako alternatívy prislúchajúce parametru pre špecifikáciu typu zdroja v žiadosti,
- `IOEmbedErrorKeyConstants` – stanovuje hodnoty kľúčov/elementov pre odpovede charakterizujúce chybné dopyty,
- `IOEmbedErrorValueConstants` – stanovuje hodnoty priradené kľúčom pre odpovede charakterizujúce chybné dopyty.

Následne je potrebné zoskupiť konštanty pre požadované a voliteľné parametre do zodpovedajúcich množín, enumeračných typov, aby sa následne bolo možné obrátiť na operáciu zistenia príslušnosti do skupiny (tieto funkcie nech deklaruje rozhranie `IAbstractEnum` realizované rodičovskou triedou `AbstractEnum`). Pre tento účel poslúžia dcérske triedy:

- `OEmbedRequiredKeys` – reprezentuje enumeračný typ pre zoskupenie povinných parametrov,
- `OEmbedOptionalKeys` – reprezentuje enumeračný typ pre zoskupenie voliteľných parametrov.

7.1.3 Výnimočné stavy

V prípade vykonávanej kontroly nad predávanými parametrami, ak stanovenie požadovaných typov alebo voliteľných typov nezodpovedá špecifikácii, navrhovaná knižnica má reagovať propagáciou výnimočného stavu, a to použitím tried:

- `InvalidRequiredOEmbedKeyException` – charakterizuje výnimočný stav pre prípad kontroly hodnoty kľúča nepatriaceho do množiny povinných parametrov,
- `InvalidOptionalOEmbedKeyException` – špecifikuje výnimočný stav v prípade kontroly hodnoty kľúča mimo množiny voliteľných parametrov.

Keďže dopyty sú akceptované len na základe (nami) určeného vzoru URL adresy, poskytuje nám to pôdu pre návrh pomocného nástroja pre spracovanie žiadosti v zmysle určenia jej správnosti, vyvodenia reakcia, ak je forma nesprávna alebo obsah nesprávny, rozčlenenie volaných parametrov, filtrácia očakávanej predpony URL adresy (tá musí byť vopred definovaná a ak nie je, nech sa reaguje výnimočnou situáciou

AcceptedURLNotInitializedException), pridanie kľúča s hodnotou do odpovede vrátane príslušného ošetrenia hodnôt a pod. Je zrejmé, že takýto nástroj bude používať aj ostatné čiastkové nástroje, už vyššie popísané.

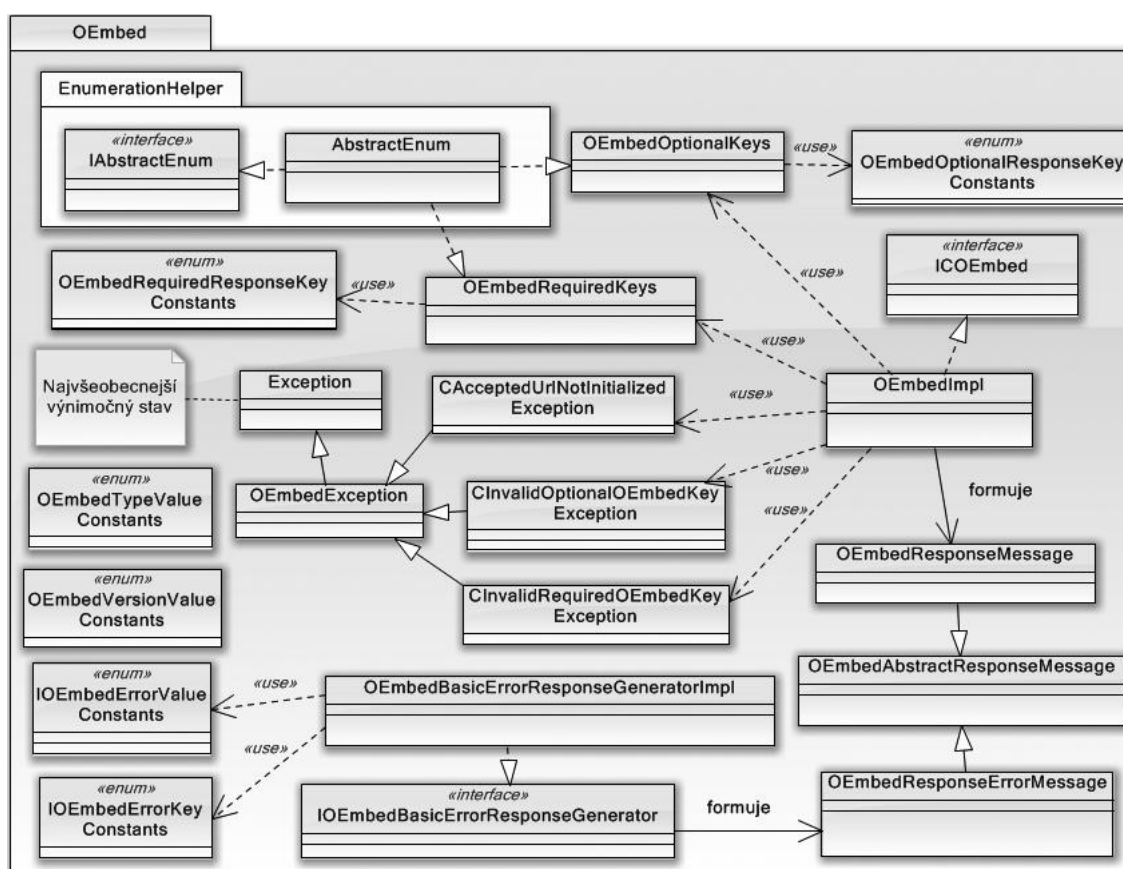
Všetky výnimočné situácie, ktoré sme tu uvádzali, nech budú rozšírením najvšeobecnejšej výnimky OEmbedException, aby sme mohli reagovať na všetky z nich nie len jednotlivo ale aj globálne.

7.1.4 Generátor odpovedí pre neúspešné volania

Potrebuje taktiež nástroj, ktorý by generoval správu na základe chyby dopytu. Konkrétne funkcie nech stanoví IOEmbedBasicErrorResponseGenerator a úlohu nech prevezme jeho implementácia, nazvime ju OEmbedBasicErrorResponseGeneratorImpl.

7.1.5 Generátor odpovedí pre úspešné volania

Množinu stanovených funkcií pre základnú manipuláciu a generovanie odpovedí v prípade úspešných dopytov nech deklaruje rozhranie IOEmbed a jeho implementáciu pomenujme menom OEmbedImpl. Navrhované triedy popisuje Obr. 32.



Obr. 32 Diagram tried spadajúcich pod knižnicu oEmbed

7.2 Rozšírenie oEmbed knižnice pre potreby Pinterest volaní

Uvažujme teraz nad Pinterest knižnicou, ktorú postavíme na základoch eEmbed knižnice.

Tak ako v prípade oEmbed, aj v tomto prípade potrebujeme rozhrania pre definíciu konštánt vychádzajúcich z dokumentácie.

7.2.1 Hodnoty kľúčov Pinterest odpovedí

Pre účel definície konštánt nám poslúži rozhranie `IPinRequiredResponseKeyConstants` obsahujúce nevyhnutné parametre Pinterest odpovede a rozhranie `IPinOptionalResponseKeyConstants` obsahujúce voliteľné parametre Pinterest odpovede.

S cieľom získať prostriedok pre zistenie príslušnosti k jednej či druhej skupine navrhujeme triedy `CPinRequiredKeys` a `PinOptionalKeys`.

O špecifikáciu prípustných hodnôt v prípade pohlavia, pre ktoré je produkt určený sa postará `IPinGenderValueConstants` a v prípade dostupnosti produktu `IPinAvailabilityValueConstants`.

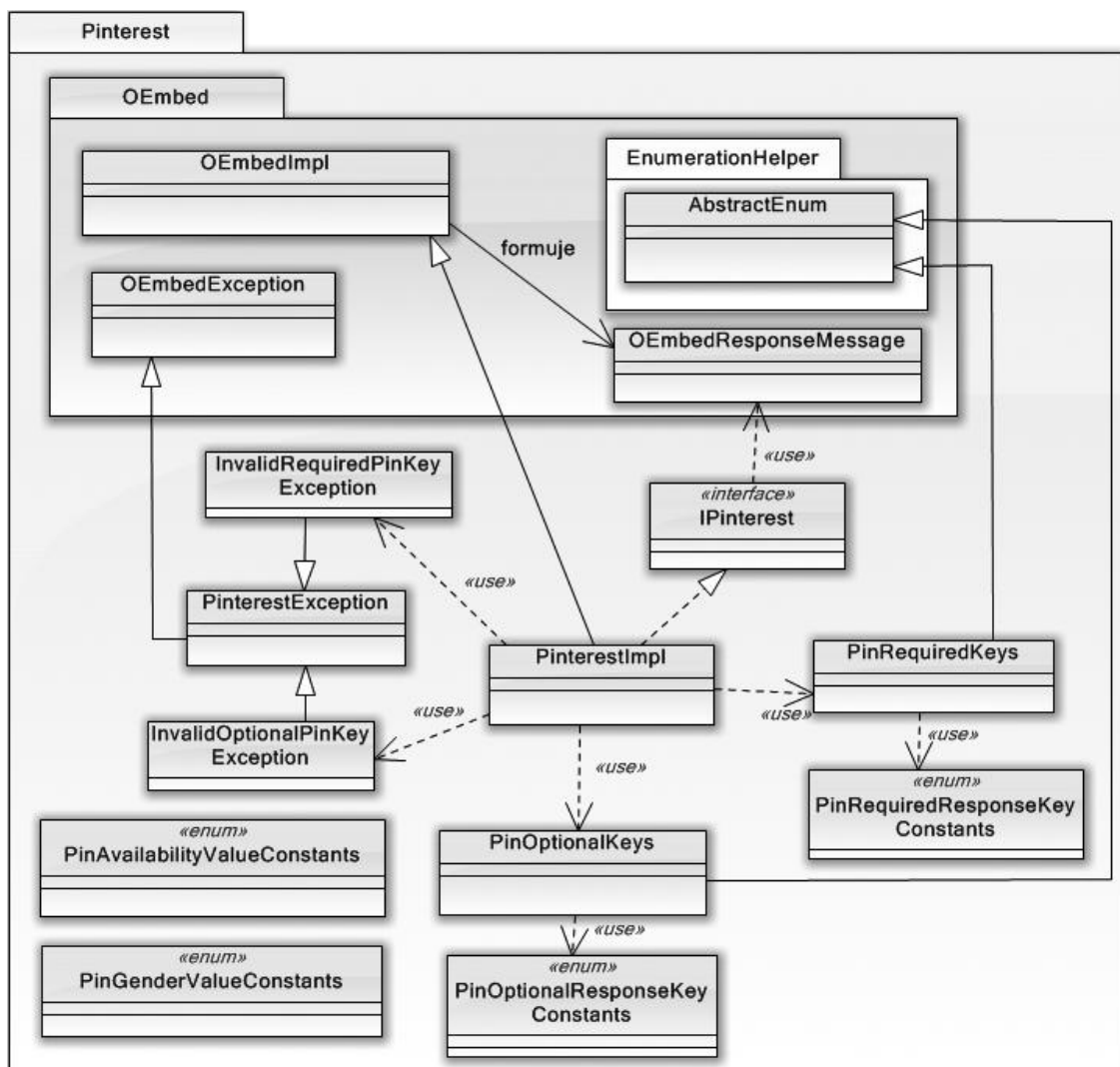
7.2.2 Výnimočné stavy

Ak pokus o vloženie požadovaného parametra alebo voliteľného parametra zlyhá, chybový stav popíšu výnimočné situácie reprezentované ako `InvalidRequiredPinKeyException` a analogicky aj `InvalidOptionalPinKeyException`. Prítomná nech je aj všeobecná výnimka `PinterestException`.

7.2.3 Generátor odpovedí

Rozšírenie knižnice o pridávanie parametrov do odpovedí na dopyty prevezme `PinterestImpl` implementujúce rozhranie `IPinterest`. Nech má zároveň na starosti generovanie odpovedí, v tomto prípade aj úspešných aj neúspešných, čo umožňuje obalenie triedy `OEmbedImpl`.

Pre znázornenie vzájomných vzťahov a závislostí medzi knižničnými triedami prikladáme Obr. 33.



Obr. 33 Diagram tried pre knižnicu Pinterest nadväzujúcej na oEmbed

8 Testovanie systému

V poslednej kapitole pred záverom sa nachvíľu zastavíme pri téme testovania implementovaného riešenia. Fázu testovania sme rozdelili do troch rovín v rámci aplikácie a tak tomu bude aj v rámci práce. V prvej podkapitole len v krátkosti načrtujeme, ako sme testovali modul pre interaktívnu kompozíciu, v druhej podkapitole popíšeme spôsob a výsledky testovania PayPal integrácie a v kapitole tretej sa bližšie pozrieme na kľb testovania Pinterest integrácie.

8.1 Akceptačné testy pre modul customizácie

Modul pre interaktivitu skladania komponentov do zložitejších produktov sme testovali manuálne. Vykonávali sme rôzne akcie nad systémom sledujúc kontrolné výpisy, ako aj obsah databázy pred a po ich vykonaní.

8.2 Integračné testy pre platbu PayPal

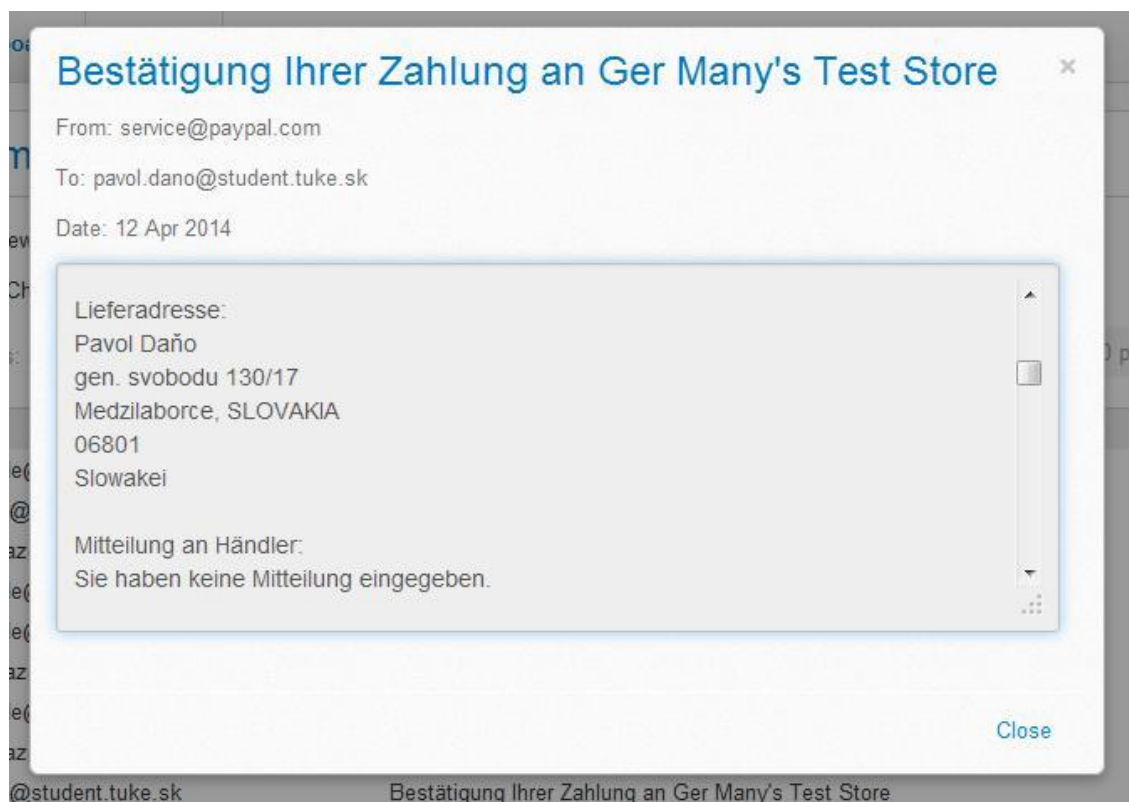
Integračné testovanie pre PayPal prebieha sa dá vykonať na viacerých úrovniach. Snáď najjednoduchším prostriedkom je výpis a ukladanie výsledkov transakcií po ich vykonaní. Práve pre tento účel sme pripravili v štádiu návrhu databázové tabuľky uvedené v podkapitole 6.6 a ukážku ich obsahu uvádzame obrázkom Obr. 34.

```
mysql> select * from sb_paypal_shipping_data;
+-----+-----+-----+-----+-----+-----+
| psd_order_id | email | payer_id | payer_status | first_name |
| last_name | country_code | ship_to_name | ship_to_street | ship_to_city | s
| hip_to_state | ship_to_country_code | ship_to_zip | address_status |
+-----+-----+-----+-----+-----+-----+
| 1 | lobogito@azet.sk | UH4SPZER4RLLQ | verified | Stefan |
| Dano | DE | Pavol Da?o | gen. svobodu 130/17 | Medzilaborce | S
| LOUAKIA | SK | 06801 | Unconfirmed |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from sb_paypal_transaction_data;
+-----+-----+-----+-----+-----+-----+
| ptd_order_id | transaction_id | transaction_type | payment_type | order_tim
| e | amt | currency_code | tax_amt | payment_status | pending_reason
| reason_code | ack |
+-----+-----+-----+-----+-----+-----+
| 1 | 4TR87362DY3856101 | expresscheckout | instant | 2014-04-2
| 2014-04-21T18:58:21Z | 137.50 | EUR | 0.00 | Completed | None
| None | Success |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Obr. 34 Výpis tabuľky pre potvrdzované dáta adresy doručenia a dáta z vykonanej transakcie prenášaných z PayPal.

V PayPal prostredí pre vývojára si môže následne vývojár skontrolovať zrealizované i nezrealizované platby, jednak z pohľadu fake účtu predajcu i z pohľadu fake účtu kupujúceho. Jednu z takýchto správ prezentuje Obr. 35²⁰.



Obr. 35 Ukážka informačnej správy na fake účte klienta v PayPal prostredí pre testovanie

Takýmto testovaním na viacerých úrovniach sme zabezpečili integritu bázy dát i korektnosť správania sa aplikácie v procese komunikácie s PayPal platobným rozhraním.

8.3 Integračné testy pre Pinterest Rich Pins

Po naštudovaní API dokumentácie Rich Pins sociálnej siete Pinterest sme sa zaoberali otázkou testovania funkčnosti integrácie a výsledkom bolo získanie adresy na validačný nástroj, ktorý používateľsky prívetivým spôsobom informuje o úspešnosti či neúspešnosti integrácie Rich Pins.

Validačný nástroj v procese validácie prijíma URL adresu servera, na ktorej je umiestnená služba integrujúca Pinterest. Službu následne volá a vyhodnocuje návratovú správu.

²⁰ Jazyk výpisu správy je nemecký a to z dôvodu uvedenia Nemecka ako krajiny, odkiaľ kupujúci i predávajúci pochádzajú. Názov obchodníka v prípade účtu predávajúceho je nastavený na Ger Many, preto je uvedený ako adresát platby Ger Many's Test Store.

Stránka nástroja pred a po vykonaní validačného procesu zachytáva obrázok Obr. 36.

The image shows two screenshots of the Pinterest Rich Pins validator tool. The top screenshot shows the initial validation step where a URL is entered and validated. The bottom screenshot shows the successful validation result and the preview of the Rich Pin.

Top Screenshot: Validation Step

for developers Login

[Go back](#)

Rich Pins validator

Validate your Rich Pins and apply to get them on Pinterest:

1. Make sure you've read the [rich pin documentation](#).
2. Validate your rich pins below by entering a URL below. This could be a page with metatags ([preview](#)), an oEmbed endpoint ([preview](#)) or a Shopify link ([preview](#)).
3. Preview your Pins to make sure they're correct.
4. Once your Rich Pins are validated, apply to get them approved.

Enter a valid URL [Validate](#)

for developers Login

[http://puwel.sk/snovbord/index.php/c_services/oembed?url=http%3A//puwe](#) [Validate](#)

✓ Your Pin's been validated! [Apply now](#).

Bottom Screenshot: Rich Pin Preview

Product rich pin
[See documentation](#)

- Site name
Snovbord
- Favicon
1
- Product Name*
In The Snovbord Store: Snovbord Hoodie I
- Availability
--
- Price*
120.0
- Currency*
EUR

What your Rich Pin will look like:

We can't load your image right now, but that's just because no one's Pinned it yet!

(P.S. You don't need the image to show up here to validate your Pins.)

In The Snovbord Store: Snovbord Hoodie I

€120 (pôvodne €120)

1 Snovbord

Obr. 36 Ukážka úspešnej validácie na stránkach Pinterest pre testovanie integrácie

9 Záver

Cieľom predkladanej práce bolo navrhnúť a zimplementovať systém s charakterom podobným elektronickým obchodom. Avšak, v porovnaní s ohromným množstvom elektronických komerčných riešení sme sa vymykali spod rámca klasických e-obchodov a orientovali sa na podporu interaktívnej tvorby produktov. Predstavuje to iný pohľad na používateľov systému, keďže sám zákazník je tiež tvorcom vlastného produktu. Zanalyzovali sme požiadavky komunikáciou so záujmovou skupinou, ktorej nápad stál pri zrode témy predkladanej práce. Stanovili sme charakteristiky vytváraného softvérového produktu a po implementácii sme uskutočnili akceptačné testy a odkonzultovali dosiahnuté výsledky. Odlišné marketingová stratégia však nebola jediným prínosom v porovnaní s klasickými e-obchodmi.

Reprezentácia produktov samotných mala byť v rámci systému realizovaná tak, aby sa generované obrazové dáta dali aj ďalej šíriť na sociálnej sieti orientovanej práve na obrazové dáta. Keďže neoddeliteľnou súčasťou systémov pre podporu podnikania sú integrované elektronické platobné metódy, jedným z vytýčených cieľov bola analýza platobných možností na slovenskom i medzinárodnom trhu a zavedenie vybranej platobnej metódy do nami vytváraného systému.

Aby sme sa vyhli chybám či nedostatkom, ktoré prichádzajú ruka v ruke s existujúcimi riešeniami zameranými na customizáciu, podrobili sme ich analýze. Zistené technologické pozadie sme mohli konfrontovať s poznatkami nadobudnutými pri získavaní prehľadu webových technológií pre podporu skladania komponentov do formy výsledných produktov. Rozhodli sme sa upustiť od technológií akými sú Adobe Flash či Microsoft Silverlight. Naproti tomu sme sa rozhodli pre použitie natívnych technológií, ktoré s WEB 2.0 prichádzajú, menovite HTML5, Javascript s podporou asynchrónnej komunikácie pomocou XML známej ako AJAX. Prínosom tohto rozhodnutia bolo, že každý používateľ systému k nemu môže pristúpiť bez nutnosti dodatočnej inštalácie akejkoľvek softvérovej podpory [33] či zásuvných modulov tretích strán. Pre prácu so systémom postačí akýkoľvek z bežne používaných webových prehliadačov, čo predstavuje značnú výhodu v porovnaní s existujúcimi riešeniami akými sú Nike, Shirtinator, GelaSkins či Skinflips.

Ak sa zamýšľate nad tým, či sme sa takýmto výberom nepripravili o podporu hardvérovej akcelerácie, ktorá by mohla byť použitá v prípade, ak by sme zvolili

technológiu Microsoft Silverlight, musíte vziať do úvahy, že pre prácu s obrazovými dátami v zmysle načítania, jednoduchých 2D transformácií a preskupení v rámci vrstiev, by hardvérová akcelerácia nenašla svoje skutočné uplatnenie.

Naopak, voľbou natívnych technológií sme nestratili na potenciálnej zákazníckej základni. Systém je schopný konkurencie v porovnaní s riešením poskytovaným elektronickým obchodom NWT3K, ktorý predstavoval najvyššiu našu métu pre konkurovanie. Elektronický obchod NWT3K pracujúci nad množinou IMG obrázkov a SVG prvkov nemá nástroje pre generovanie výsledného obrázku z množiny SVG elementov prekrytých IMG elementmi. Pravdepodobne preto jeho tvorcovia neboli schopní implementovať zdieľanie v sociálnych sieťach v takej miere, v akej to je dnes, dá sa povedať, už zvykom, no prichýlili sa len k jednoduchým odkazom na sociálne siete s ich obchodnými profilmi. To isté platí aj pre komerčne zamerané aplikácie ako Milk&Honey či Scarosso.

Nám sa však podarilo tieto problémy preklenúť a to kombináciou HTML prvkov pracujúcich s rastrovou grafikou s prvkami pracujúcimi s vektorovými dátami. Konkrétne ide o postupnú aplikáciu obrazových dát v nasledovnom poradí: najsamprv dochádza k vykresleniu vektorových informácií vrátane použitých farieb na pomocný HTML prvok `<canvas>`, následne sa vykresľuje základný produkt na iný pomocný prvok `<canvas>` vo forme priehľadného PNG obrázka, v treťom kroku sú aplikované na ten istý pomocný element rastrové reprezentácie komponentov, celý proces je zavŕšený načítaním rastrového obsahu pomocného plátna a jeho prekreslením na plátno s už rastrovými informáciami reprezentujúcimi pôvodné SVG dáta. Následné volanie Javascript metód pre získanie obsahu plátna poskytne očakávaný výsledok – individuálnu PNG reprezentáciu použiteľnú v ďalších systémových funkciách.

Úspešná implementácia tohto algoritmu má nielen dopad na vytvorenie pôdy pre zdieľanie v prostredí sociálnych sietí, ba čo viac, jej dopad na efektivitu systému je citelný. Prečo? Odpoveď je jednoduchá. Ak by výsledkom kompozície produktu bola výlučne množina rastrových a vektorových informácií, okrem neschopnosti zdieľania v sociálnych sieťach by akékoľvek snahy o vizuálne zobrazenie komponovaného produktu končili pri manipulácii s celou množinou i rastrových i vektorových elementov. A to na všetkých úrovniach MVC rámca. Jednak na úrovni bázy dát musí dôjsť k buď viacnásobným dopytom alebo ku komplikovanejším dopytom pre selekciu obrazových dát, na úrovni aplikačnej logiky sa musia tieto výsledky následne

univerzálne cyklicky spracovávať a v štádiu generovania výstupu vo forme webovej stránky je DOM objekt výsledného HTML dokumentu zaťažený o výskyt všetkých generovaných prvkov. Poskytované riešenie odbúrava vymenované neduhy nasledovným spôsobom. Pre reprezentáciu výsledného produktu sa používa práve jedna rastrová informácia v podobe PNG obrázka, ktorá je priamo priradená komponovanému produktu. Pri načítaní jedného produktu na úrovni databázy nedochádza hneď k selekcii informácií o použitom základnom produkte a všetkých aplikovaných komponentoch, no dopyt obsahuje požiadavku pre získanie jednej URL adresy jeho rastrovej reprezentácie, ktorú následne na výstupe generuje bez potreby akéhokoľvek cyklického prechádzania množinou reprezentácií použitých komponentov. Automaticky potom dochádza k integrácii informácie o umiestnení (obrazovej reprezentácie) produktu s vybranou sociálnou sieťou, ktorá by inak nebola možná.

Siahnutím po SVG sme taktiež dosiahli zníženie pamäťových nárokov pre ukladanie obrázkov na strane servera, ktoré boli nevyhnutné v niektorých analyzovaných riešeniach pre potrebu dosiahnutia dojmu interaktivity (Milk&Honey, Scarosso). Namiesto toho, aby pre každú verziu komponentu bol na strane servera uložený obrázok a na strane databázy položka k jeho dosiahnutiu, prítomnosťou SVG informácie eliminujeme túto nevyhnutnosť a tak s jednou rastrovou a s jednou vektorovou reprezentáciou úplne vystačíme pre dosiahnutie dojmu manipulácie nad *niekoľkými* rôznymi komponentmi. Cestou bolo obohatenie aplikačnej logiky a ukladanie SVG prvkov do databázy, čo nie je nič iné, len text. Nevýhodou tohto princípu je potreba pracovať s vektorovým formátom a časť jeho textovej reprezentácie do systému zavádzať. V tomto duchu by sme mohli iniciovať ďalšiu prácu na systéme na budúcu, ktorá by sa mohla zaoberať otázkou zefektívnenia procesu zavádzania SVG prvkov do systému, čo v podstate znamená dynamické parsovanie obsahu, validáciu SVG súboru a v našom prípade selekcii len podstatných `<PATH>` elementov. Uvedeným sa zaoberá napríklad Zhan a Engelen v [50], Zhenghong v [51] či Ma v [52].

Na ceste za získaním prostriedku pre šírenie reklamy pomocou potenciálu sociálnych sietí sme sa rozhodli systém integrovať so sieťou Pinterest, ktorá sa teší veľkej obľube[47] a zodpovedá svojím charakterom stanoveným požiadavkám. Jej výber bol týmto jednoznačný. Podarilo sa nám zabezpečiť jej integráciu zasielaním JSON správ spĺňajúcich pravidlá stanovené aplikačným programovým rozhraním Pinterest, ktoré tak zastrešovalo všeobecné pravidlá oEmbed technológie,

ako jeho základu. Obrazové dáta na strane sociálnej siete sa týmto stali obohatenými o dodatočné informácie o zdieľaných produktoch – z pohľadu Pinterest používateľa stále len obrazových dátach, čo nemá podporu napríklad ani v prípade NWT3K, keď sa k zdieľanému obsahu dodatočné dáta nepripájajú.

Poskytli sme tak knižnicu, ktorej nasadenie rýchlo umožní integráciu so sociálnou sieťou Pinterest na vyššej úrovni v prostredí elektronických riešení s podporou customizácie.

Integrácia iných sociálnych sietí, ktoré nie sú orientované prioritne na šírenie obrazových informácií, môže byť predmetom ďalšej expanzie systému. Ich integrácia však nakoniec nebola výzvou pre realizáciu, keďže ich charakter nevytváral až také prívetivé prostredie pre šírenie obrazových dát ako Pinterest. Uprednostnili sme progresívnejší prístup a zamerali sa tak na aktuálnejšie aspekty šírenia reklamy vo svete sociálnych sietí.

Po analýze platobných možností na Internete sme sa priklonili k integrácii s PayPal, ktorá má medzinárodný charakter a širokú základňu používateľov. Možnosti jej integrácie boli rozsiahle, no pre potreby e-komerčného riešenia sme sa zamerali len na najpodstatnejšiu podporu, to jest prevod peňazí medzi účtami PayPal zákazníkov, pretože platobný systém sa stáva úspešným, keď jeho návrh vyhovuje požiadavkám používateľa [34]. Jedným z aktérov je zákazník, ktorý pristupuje na stránky predajcu a rozhoduje sa zaplatiť za tovar zo svojho PayPal účtu. Prevádzkovateľ ho naviguje na platobné rozhranie PayPal, dodáva svoje obchodné údaje ako aj údaje o platbe, zákazník buď súhlasí s podmienkami platby alebo odmieta vykonať transakciu. V oboch prípadoch je nasmerovaný na stránku predajcu, kde už dostáva informácie o úspešnosti zrealizovania alebo nezrealizovania platby. Existujúce riešenia, ako napríklad [34], sa opierali o verziu PHP5.3+, čo nás viedlo k vlastnej implementácii. Vlastnou implementáciou v rámci možností jazyka PHP verzie 5.2 sa nám tak podarilo dodať nástroj pre zavedenie PayPal platieb použiteľný kdekoľvek, kde si to situácia žiada.

Jedným z vylepšení, ktoré by sa dali do predkladaného systému zapracovať, je presun z klasického WWW servera do cloudu. Tejto otázke sme istý čas venovali a môžeme s pokojným svedomím prehlásiť, že návrh nepredstavuje prekážku a presun je možný. Podmienkou jeho realizácie je zmena alebo dodanie konfiguračných súborov v závislosti od podmienok, ktoré si *konkrétne* cloudové riešenie vyžaduje.

Zoznam použitej literatúry

- [1] GAJDOŠÍKOVÁ, Jana: Elektronický obchod – nedoriešené problémy. Referát [cit. 2013-06-17]. Dostupné na internete: <<http://edi.fmph.uniba.sk/~winczer/SocialneAspekty/gajdosikova.htm>>.
- [2] Smernica č. 2000/31/ES Európskeho parlamentu a rady. Dostupné na internete <<http://www.economy.gov.sk/e%EF%BF%BD-obchod-6019/127704s>>.
- [3] PINE, Joseph: Mass Customization: The New Frontier in Business Competition. Harvard College: Harvard Business Review Press, 368 s. ISBN 978-0875843728.
- [4] XIA, Nan – RAJAGOPALAN, Sampath: Standard vs. Custom Products: Variety, Lead Time and Price Competition. In: Marketing Science, 2009. s. 887 – 900.
- [5] XIA, Nan – RAJAGOPALAN, Sampath: A Competitive Model of Customization with Lead-time Effects. In: Decision Sciences, 2009. s. 727 – 758.
- [6] Marketing na sociálnych sieťach. Článok [cit. 2013-06-18]. Dostupné na internete: <<http://marketing.krea.sk/clanky/socialne-siete/>>.
- [7] BIN, Tan – Fuchun, Peng: Unsupervised Query Segmentation Using Generative Language Models and Wikipedia. In: Proceeding of World Wide Web. 2008. s. 347 – 356.
- [8] KANGPYO, Lee et al.: A Subsumption-based Folksonomy Visualization Using Wikipedia Texts. In: Proceedings of World Wide Web. 2008. s. 1093 – 1094.
- [9] YIN, Zhang et al.: An Adaptive Web Toolkit for Various Mass Customization Users. In: Management and Service Science, Wuhan: 2009. s. 1–4. ISBN 978-1-4244-4639-1.
- [10] CAVUSOGLU, Hasan – Cavosuglu, Huseyin – Raghunathan, Srinivasan: Selecting a Customization Strategy under Competition: Mass Customization, Targeted Mass Customization, and Product Proliferation. In: IEEE transactions of engineering management. 2007.
- [11] Parallels: Linux Shared Web Hosting Next Generation Service Module. Citované 15-04-2014. Dostupné na internete: <<http://www.parallels.com/products/automation/services-and-applications/sharedng/>>.
- [12] MVCC: Model-View-Controller-Confusion. Aktualizované 03-02-2010 [cit 2014-04-14]. Dostupné na internete: <<http://psychopathya.wordpress.com/2010/02/03/mvcc-model-view-controller-confusion/>>.
- [13] MUSCIANO, Chuck – KENNEDY, Bill: HTML a XHTML – Kompletní průvodce. Praha: Computer Press, 2000. 676 s. ISBN 80-7226-407-9.

-
- [14] ORNDORFF, Jessie: Disadvantages of Flash. Aktualizované 02-12-2010 [cit 2013-06-26]. Dostupné na internete: <<http://www.churchwebsiteideas.com/resources/the-disadvantages-of-flash/>>.
- [15] GARNEY, Ben: Flash Player: A Declining Asset [online]? Publikované 21-11-2011 [cit 2013-06-25]. Dostupné na internete: <<http://blog.bengarney.com/2011/11/21/flash-player-a-declining-asset/>>.
- [16] ROTH, Stefan: HTML5 vs. Silverlight aka Ajax vs. Java Applets – A Trip in The Time Machine [online]. Aktualizované 13-04-2011 [cit 2013-07-02]. Dostupné na internete: <<http://rothisblog.blogspot.sk/2011/04/html5-vs-silverlight-aka-ajax-vs-java.html>>.
- [17] Comparison of Ajax and Applet [online]. Citované 02-07-2013. Dostupné na internete: <<http://www.ajax.thiyagaraaj.com/comparison-of-ajax-and-applet>>.
- [18] Comparison of Web Browsers. Dostupné na internete: <http://en.wikipedia.org/wiki/Comparison_of_web_browsers#Image_format_support>.
- [19] SHEPHERD, Eric: Canvas Tutorial. Aktualizované 27-06-2013 [cit 2013-07-03]. Dostupné na internete: <https://developer.mozilla.org/en-US/docs/Canvas_tutorial?redirect=no>.
- [20] GRAVELLE, Robert: HTML5 Canvas vs. SVG – Choose the Best Tool for the Job. Citované 22-03-2014. Dostupné na internete: <http://www.htmlgoodies.com/html5/other/html5-canvas-vs.-svg-choose-the-best-tool-for-the-job.html#fbid=SeT3EK0_3yx>.
- [21] KALÁB, Miloš: SVG-based Knowledge Visualization : Diploma thesis. Brno: FI MU, 2012. 64 strán.
- [22] GARRET, Jesse James: Ajax: A New Approach to Web Applications [online]. Aktualizované 18-02-2005 [cit 2013-07-02]. Dostupné na internete: <<http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>>.
- [23] DELINA, Jaroslav – VAJDA, Viliam – BOLHA, Jozef: Elektronický obchod v slovenskom priemysle. Košice: EF TU, 2008. 110 s. ISBN 978-80-969953-1-8.
- [24] Paysafecard. Citované 15-07-2013. Dostupné na internete: <<https://www.paysafecard.com/sk-sk/>>.
- [25] PayPal – About us. Dostupné na internete: <<https://www.paypal.com/webapps/mpp/about>>.
- [26] PayPal – About Paypal. Dostupné na internete: <https://www.paypal.com/us/cgi-bin/webscr?cmd=p/pop/about_paypal_wa-outside>.
- [27] ROUSE, Margaret: Google Wallet Definition [online]. Citované 04-07-2013. Dostupné na internete: <<http://www.techopedia.com/definition/27518/google-wallet>>.
-

-
- [28] Google developers: Frequently Asked Questions [online]. Aktualizované 13-02-2014 [cit 2014-04-05]. Dostupné na internete: <<https://developers.google.com/wallet/digital/faq>>.
- [29] Skrill eWallet. Dostupné na internete: <<https://www.moneybookers.com/ads/online-platby/epenezenka/>>.
- [30] Skrill Prepaid MasterCard. Dostupné na internete: <<http://g.moneybookers.com/mastercard/cz/>>.
- [31] TrustPay – Všeobecné obchodné podmienky. Dostupné na internete: <<http://www.trustpay.eu/assets/Uploads/VOP-a-cenniky-2013SKod-1.5.7.pdf>>.
- [32] TrustPay – Produkty. Dostupné na internete: <<http://www.trustpay.eu/produkty-paypal-ukash-paysafecard-zaplat-to/>>.
- [33] GONG, Xini et al.: Web Visualization of Distributed Network Measurement System Based on HTML5. In: Cloud Computing and Intelligent Systems. Beijing: 2012. s. 519 – 523. ISBN 978-1-4673-1855-6.
- [34] SAHUT, Jean-Micheal: Security and Adoption of Internet Payment. In: Emerging Security Information, Systems and Technologies. Cap Esterel: 2008. s. 218 – 223. ISBN 978-0-7695-3329-2.
- [35] Angell EYE PayPal PHP Library. Dostupné na internete: <<https://github.com/angelleye/paypal-php-library>>.
- [36] VALDNER, Martin: Možnosti platby za tovar a služby v prostredí internetu [online]. Bratislava: UK, FM, 2009. [cit. 2014-01-10]. Dostupné na internete: <http://school.valdner.com/data/FM%20UK%20BA/Statnice/diplomovka/diplomova_praca-moznosti%20platby%20za%20tovar%20a%20sluzby%20v%20prostredi%20internetu.pdf>.
- [37] HORNÝ, Juraj – PEKAROVIČ, Martin – PIZANO, Veronika: Sociálne siete a marketing. In: Zborník prác zo Študentskej vedecko-odbornej a umeleckej konferencie 2010. Trnava: Fakulta masmediálnej komunikácie UCM v Trnave, 2010. s. 236-256. ISBN 9788081051845.
- [38] VELŠIC, Marián: Sociálne siete na Slovensku [online]. Bratislava: Inštitút pre verejné otázky, 2012. Citované 17-07-2013. Dostupné na internete: <http://www.ivo.sk/buxus/docs/publikacie/subory/Socialne_siete_SR.pdf>. ISBN 978-80-89345-36-6.
- [39] BOSWELL, Wendy: Social Media: What Is It [online]? Citované 17-07-2013. Dostupné na internete: <http://websearch.about.com/od/blogsforumsocialsites/f/social_web.htm>.
- [40] NIMETZ, Jody: Emerging Trends in B2B Social Networking [online]. Aktualizované 18-11-2007 [cit 2013-07-17]. Dostupné na internete: <<http://www.marketing-jive.com/2007/11/jody-nimetz-on-emerging-trends-in-b2b.html>>.
-

-
- [41] MORIN, Dave: Announcing Facebook Connect [online]. Zverejnené 09-05-2008 [cit. 2013-12-22]. Dostupné na internete: <<https://developers.facebook.com/blog/post/2008/05/09/announcing-facebook-connect/>>.
- [42] BARINKOVÁ, Veronika: Zapojenie sociálnych sietí do informačného vzdelávania [online]. Brno: Masarykova univerzita, Filozofická fakulta, Ústav české literatury a knihovnictví, 2012. 111 s. [cit. 2013-12-23]. Dostupné na internete: <http://is.muni.cz/th/384448/ff_m/DP.pdf>.
- [43] BAZALOVÁ, Lenka: Pohľad na sociálne siete a využívanie [online]. Brno: Masarykova univerzita, Filozofická fakulta, Ústav české literatury a knihovnictví, 2012. 78 s. [cit. 2013-12-23]. Dostupné na internete: <http://is.muni.cz/th/384483/ff_m/DS_Bazalova.pdf>.
- [44] NISHAR, Deep: 200 Million Members [online]. Zverejnené 09-01-2013 [cit. 2013-12-22]. Dostupné na internete: <<http://blog.linkedin.com/2013/01/09/linkedin-200-million/>>.
- [45] LinkedIn API: Overview [online]. Citované 23-12-2013. Dostupné na internete: <<https://developer.linkedin.com/documents/overview>>.
- [46] RŮŽIČKA, Jan: Pinterest je sociální síť založená na kráse a inspiraci. Roste nejrychleji ze všech [online]. Aktualizované 15-06-2012 [cit. 2013-12-23]. Dostupné na internete: <<http://tech.ihned.cz/c1-56149650-pinterest-je-socialni-sit-zalozena-na-krase-a-inspiraci-roste-nejrychleji-ze-vsech>>.
- [47] ZIMING, Feng et al.: An Empirical Study of User Behaviors on Pinterest Social Network. Web Intelligence and Intelligent Agent Technologies, Atlanta, 2013. s. 402 – 409. ISBN 978-1-4799-2902-3.
- [48] Schema.org: What is Schema.org? Citované 14-03-2013. Dostupné na internete: <<http://schema.org/>>.
- [49] oEmbed. Citované 14-03-2013. Dostupné na internete: <<http://oembed.com/>>.
- [50] ZHANG, Wei – ENGELN, Robert: An Adaptive XML Parser for Developing High-Performance Web Services. In: eScience. Fourth IEEE International Conference on eScience, Indianapolis: 2008. s. 672 – 679. ISBN 978-1-4244-3380-3.
- [51] ZHENGHONG, Gao et al.: A High Performance Schema-Specific XML Parser. In: e-Science and Grid Computing. IEEE International Conference, Bangalore: 2007. s. 245 – 252. ISBN 978-0-7695-3064-2.
- [52] MA, Jianliang et al.: Paraller Speculative Dom-based XML Parser. In: High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, Liverpool: 2012. s. 33 – 40. ISBN 978-1-4673-2164-8.
-