

Návrh internetového elektronického obchodu

Ecommerce web design

Martin Hozík

Bakalářská práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin HOZÍK**
Osobní číslo: **A08045**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Návrh internetového elektronického obchodu**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma s důrazem na programovací a kódovací jazyky, komunikační protokoly a kryptografické metody.
2. Vytvořte kompletní internetovou aplikaci elektronického obchodu.
3. Popište model datového úložiště a přístupového rozhraní jednotlivých částí aplikace s podrobnějším zaměřením na programové řešení možných bezpečnostních rizik.
4. Vytvořte funkční instalaci aplikace.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. ZELENKA, J., ČAPEK, J., FRANCEK, J., JANÁKOVÁ, H. Ochrana dat, Kryptologie. Gaudeamus, září 2003. 171 s. ISBN 80-7041-737-4.
2. ČANDÍK, Marek. Základy informační bezpečnosti. vyd. Zlín: Univerzita Tomáše Bati, 2004. 107 s. ISBN 8073182181.
3. DOSTÁLEK, L., VOHNOUTOVÁ, M., KNOTEK, M. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. Computer Press, a.s, 2009. ISBN 978-80-251-2619-6.
4. RESIG John. Javascript a AJAX – Moderní programování webových aplikací. Computer Press, 2007. ISBN 978-80-251-1824-5.
5. KATZ, Jonathan. Introduction to Modern Cryptography: Principles and Protocols. Chapman & Hall, 1 edition. 2007. 552 s. ISBN 978-1584885511.
6. GILMORE W. Jason. Velká kniha PHP & MySQL 5 – kompendium znalostí pro začátečníky i profesionály. Zoner Press, 2006. ISBN 80-86815-53-6.
7. BITTO, O. Šifrování a biometrika. BEN, 2005. 168 s. ISBN 80-86686-48-5.
8. KOLEKTIV AUTORŮ. PHP5, MySQL, Apache – vytváříme webové aplikace. Computer Press, 2006. ISBN 80-251-1073-7.

Vedoucí bakalářské práce:

Ing. Roman Šenkeřík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

24. února 2012

Termín odevzdání bakalářské práce:

8. června 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Bakalářská práce popisuje v teoretické části klíčové technologie pro sestavení a provoz internetové webové klient-server aplikace. Soustředí se také na běžná bezpečnostní rizika při zpracování a uchování privátních dat.

Praktická část demonstruje modelový příklad sestavení aplikace v podobě elektronického obchodu. Popisuje vhodné schéma zpracování, uchování a prezentace dat a některé elektronické platební prostředky.

Klíčová slova: webová aplikace, elektronický obchod, bezpečnost, PHP, MySQL.

ABSTRACT

The theoretical part of this bachelor thesis describes key technologies for build and service of internet web client-server application. It also focuses on common security threats in processing and storing private data.

Practical part demonstrates e-commerce web application as a case study, describes an appropriate scheme for data processing and storing and some online payment solutions.

Keywords: web application, e-commerce, security, PHP, MySQL.

Děkuji především svému vedoucímu práce, Ing. Romanu Šenkeříkovi Ph.D., za odborné vedení, rady, připomínky a inspiraci během studia i sestavování této práce.

Vděk bych také rád vyjádřil autorům aplikačních rámců Nette a Flex, díky kterým se praktická část práce může soustředit na samotná podstatná návrhová řešení aplikace.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 KLÍČOVÉ TECHNOLOGIE APLIKACE	11
1.1 KOMUNIKACE A PŘENOS DAT	11
1.1.1 World Wide Web (WWW) v síti Internet.....	11
1.1.1.1 Web 2.0	11
1.1.2 Hypertextový přenosový protokol (HTTP).....	11
1.1.2.1 Udržování relací.....	12
1.1.3 Hypertextový značkovací jazyk (HTML) a kaskádové styly (CSS)	12
1.1.3.1 Rozšiřitelný značkovací jazyk (XML)	13
1.2 PROGRAMOVÉ ZPRACOVÁNÍ DAT	13
1.2.1 PHP: Hypertextový preprocesor	14
1.2.2 Strukturovaný dotazovací jazyk (SQL) a Systém řízení báze dat (DBMS)	14
1.2.3 JavaScript	14
1.2.4 Platforma Flash	15
1.3 KRYPTOGRAFIE A ŘÍZENÍ PŘÍSTUPU K PRIVÁTNÍM DATŮM.....	15
1.3.1 Zabezpečená přenosová vrstva nad HTTP (TLS/SSL).....	15
1.3.2 Symetrické šifrování dat	16
1.3.2.1 DES a AES	17
1.3.3 Asymetrické šifrování dat.....	17
1.3.3.1 RSA.....	17
1.3.4 Asymetrický otisk dat.....	18
1.3.4.1 MD5 a SHA	18
2 HLAVNÍ BEZPEČNOSTNÍ RIZIKA INTERNETOVÝCH APLIKACÍ	19
2.1 CHYBNÉ ZPRACOVÁNÍ NEDŮVĚRYHODNÝCH DAT	19
2.1.1 Injektáž kódu	19
2.2 KOMPROMITACE OSOBNÍCH ÚDAJŮ.....	19
2.2.1 Zabezpečení dat úložiště	20
2.2.2 Phishing a Pharming.....	21
2.3 NEOPRÁVNĚNÁ AUTORIZACE	22
2.3.1 Hrubá síla	22
2.3.2 Zabezpečení hesel	23
2.3.3 Podvržení požadavků klienta (XSRF).....	24
2.4 NARUŠENÍ FUNKČNOSTI (DENIAL OF SERVICE).....	24
2.4.1 Strojové zneužití služby.....	25
II PRAKTICKÁ ČÁST	26
3 VOLBA VHODNÝCH TECHNOLOGIÍ, NÁSTROJŮ A PODOBY WEBOVÉ APLIKACE	27
3.1 SERVEROVÝ PROGRAMOVÝ INTERPRET, DATABÁZOVÝ SYSTÉM A APLIKAČNÍ RÁMCE.....	27
3.1.1 Nette Framework.....	27
3.1.2 Apache Flex	28
3.1.3 JQuery.....	28

3.2	FUNKČNÍ ROZSAH APLIKACE.....	28
4	MODEL DATABÁZOVÉHO ÚLOŽIŠTĚ.....	29
4.1	PRODUKT	31
4.2	OBJEDNÁVKA	32
4.3	ÚDAJE O PLATEBNÍCH KARTÁCH	32
4.4	ZÁKAZNÍCI A ADMINISTRÁTOŘI.....	33
5	DŮLEŽITÉ ČÁSTI UŽIVATELSKÝCH ROZHRAŇÍ	35
5.1	SEZNAM POLOŽEK	36
5.2	DETAIL POLOŽKY.....	36
5.3	SPRÁVA KOŠÍKU	37
5.4	OBJEDNÁVKA	38
5.4.1	Šifrovací modul údajů o platební kartě	39
5.5	PŘIHLÁŠENÍ ZÁKAZNÍKŮ A ADMINISTRÁTORŮ.....	40
5.6	ADMINISTRACE.....	41
5.6.1	Úprava obsahu obchodu.....	41
5.6.2	Správa objednávek	42
5.6.3	Dešifrování údajů o platební kartě.....	43
6	ZÁKLADNÍ DRUHY BEZHOTOVOSTNÍCH ON-LINE PLATEB	45
6.1	E-COMMERCE 3D-SECURE	45
6.2	PAYPAL EXPRESS CHECKOUT	46
6.3	PŘEDÁNÍ ÚDAJŮ O KARTĚ OBCHODNÍKOVÍ.....	47
	ZÁVĚR.....	50
	ZÁVĚR V ANGLIČTINĚ.....	51
	SEZNAM POUŽITÉ LITERATURY	52
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	54
	SEZNAM OBRÁZKŮ	56

ÚVOD

Vývoj prostředí webu v síti Internet dal v posledních deseti letech impuls ke vzniku nového odvětví informačních technologií. Technologií, které jsou často decentralizované a nezávislé na lokalitě i platformách uživatelů.

Charakteristickým rysem moderních webových technologií je jejich transformace z elektronických dokumentů na obousměrně komunikující elektronické aplikace. Běžní koncoví uživatelé se tak mění na autory informací. Zapojením běžných uživatelů do tvorby informací se webové technologie samovolně zapojují do běžného života lidí. Web tak přestává být pouhým nosičem a distributorem informací, ale je nyní i vhodným prostředím pro zábavu, komunikaci a obchod.

Posun rolí uživatelů webu s sebou ale také přinesl vážná bezpečnostní rizika a odhalil slabá místa v zažité koncepci mnohých právních, komunikačních a obchodních modelů.

Typickým představitelem aplikace modelu z reálného světa do elektronického je webový elektronický obchod.

I. TEORETICKÁ ČÁST

1 KLÍČOVÉ TECHNOLOGIE APLIKACE

1.1 Komunikace a přenos dat

1.1.1 World Wide Web (WWW) v síti Internet

World Wide Web, česky řídce *Světová pavučina*, běžněji jednoduše *Web*, je souhrn propojených (často proto označovaných jako *hypertextových*) dokumentů v síti Internet. Každý z těchto dokumentů je jednoznačně identifikován pomocí identifikátoru *URI*, který typicky zároveň plní i funkci jednoznačného lokalizátoru, tedy *URL*, směřujícího na konkrétní webový server. (1)

Strukturu těchto dokumentů tvoří běžně, nikoliv však výhradně, značkovací jazyk *HTML*, doplněný typicky o vizuální definici stylů *CSS* a případně také o soubor programových instrukcí, nejčastěji v jazyce *JavaScript*.

Komunikaci s webovým serverem, grafické zobrazení dokumentů webu a interpretaci skriptů obstarává program – *webový prohlížeč*. Komunikace probíhá prostřednictvím protokolu *HTTP* nad protokolem *TCP* s případnými dalšími zabezpečenými vrstvami. (2)

1.1.1.1 Web 2.0

Rozvojem schopností programových interpretů webových prohlížečů a webových serverů se naskytla možnost využití technologie webu nikoliv jen pro distribuci dokumentů, ale nově také jako rámec vhodný pro provoz obousměrně komunikujících internetových aplikací.

Tento posun filozofie použití webu bývá někdy označován jako *Web 2.0*, přestože souhrnný vývoj webu jako takového není nijak verzován. (1)

1.1.2 Hypertextový přenosový protokol (HTTP)

Protokol *HTTP* je hlavním komunikačním prostředkem webové aplikace. Spadá do poslední, aplikační, vrstvy modelu *OSI*.

HTTP je bezstavový, synchronní, jednobláknový protokol, pracující v modelu hierarchie klient-server, kde klient je nejčastěji webový prohlížeč. (2)

V praxi komunikace probíhá tak, že klient sestaví *požadavek* a server pak na základě něj poskytne *odpověď*.

Požadavek i odpověď je v libovolném formátu, textovém v případě samotného HTML dokumentu, vždy je však uvozen textovou hlavičkou s procesními instrukcemi a informacemi. (2)

1.1.2.1 Udržování relací

Bezstavovost HTTP komunikace je výhodná z hlediska jednoduchosti obsluhy klientských požadavků, ale je problematická u internetových aplikací, které ze své podstaty vyžadují trvalé udržování některých dat mezi požadavky nebo používají asynchronní komunikaci se serverem.

Vzhledem k tomu, že webový server považuje každý pár požadavku a odpovědi za nezávislý, relace je udržována až v programové vrstvě serverového a případně i klientského programového interpreta předáváním unikátních identifikačních parametrů.

Kompromitace identifikátoru relace je pak velmi závažným bezpečnostním rizikem a je vhodné přijmout i některá další opatření na jeho ochranu respektive zneplatnění.

1.1.3 Hypertextový značkovací jazyk (HTML) a kaskádové styly (CSS)

HTML je hlavním kódovacím jazykem webových dokumentů a aplikací. Je odvozen z obecnějšího *SGML* a jeho strukturu představuje obyčejný čistý text, jehož části jsou ohraničeny značkami, které jim dávají sémantický význam. (3)

Vzhledem k tomu, že HTML je jazyk *hypertextový*, umožňuje prostřednictvím některých značek připojovat nebo odkazovat na další dokumenty a prostředky.

Prvotní verze HTML umožňovaly kromě sémantického významu definovat také vizuální podobu dat. S rozvojem grafických schopností prohlížečů se postupně prosadila oddělená definice vizuálních stylů v samostatném formátu *kaskádových stylů*.

Soubor *CSS* direktiv pak může být jak součástí HTML dokumentu, tak i jako dokument s vlastním URI. Charakteristickou vlastností *CSS* direktiv je jejich vzájemná *kaskádová dědičnost*, kdy jedna direktiva může doplňovat druhou. (4)

1.1.3.1 Rozšiřitelný značkovací jazyk (XML)

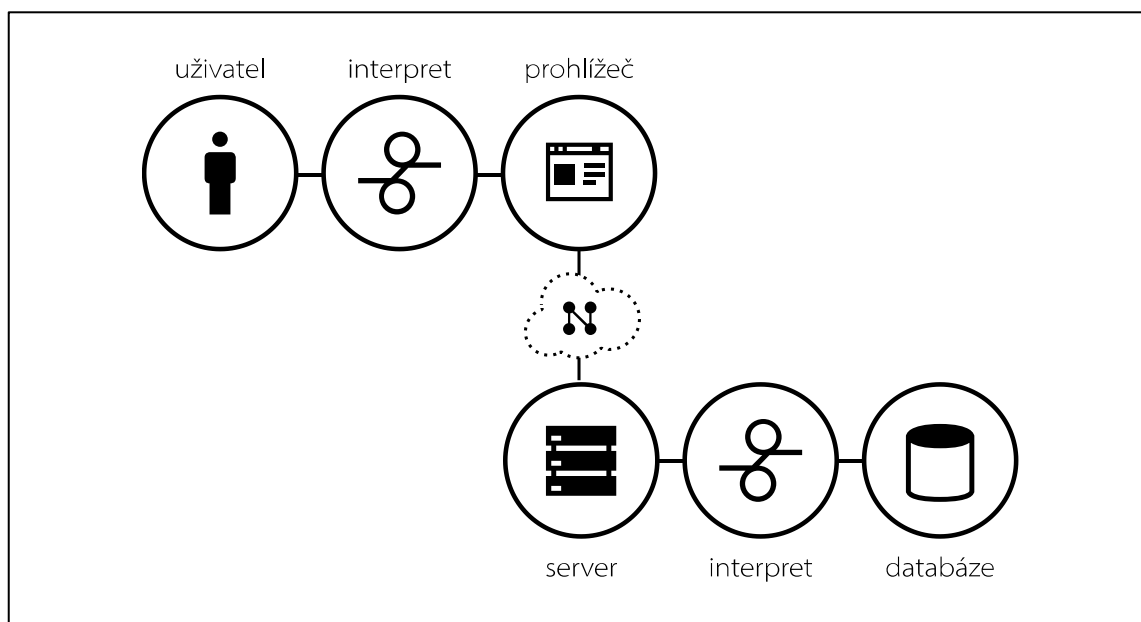
O něco mladší implementací SGML je jazyk *XML*. Ten slouží především jako platformě-nezávislý formát strukturovaných dat.

Na rozdíl od HTML vyžaduje striktní dodržení pravidel syntaxe a nestanovuje žádnou předdefinovanou kolekci značek. (5)

Pro svoji univerzálnost je běžně využíván jako základní formát komunikace mezi webovými aplikacemi.

1.2 Programové zpracování dat

Základní model fungování webu předpokládá jednoduché předání dokumentu serverem na základě požadavku klienta. Jak na straně klienta, tak i na straně serveru však mohou být na základě požadavku respektive odpovědi vykonány další programové instrukce a rovněž tak samotný požadavek i odpověď mohou být sestaveny programově, jak ukazuje obrázek 1.



Obrázek 1: Model interpretované komunikace klienta a serveru

1.2.1 PHP: Hypertextový preprocesor

PHP je serverový skriptovací jazyk. Z hlediska jeho programovacích vzorů je procedurální, synchronní, objektově orientovaný, se slabou typovou disciplínou. (6)

Z hlediska zpracování je běžně, nikoliv však výhradně, interpretován.

PHP interpret funguje jako modul HTTP serveru, případně jako externí program připojený k HTTP serveru prostřednictvím rozhraní *CGI*. Jeho oficiální implementace je napsána v *jazyce C*.

Původně byl určen k jednoduchému sestavování HTML dokumentů. V dnešní době, díky velkému množství extenzí, je schopen pracovat s velmi širokým spektrem dat i služeb.

1.2.2 Strukturovaný dotazovací jazyk (SQL) a Systém řízení báze dat (DBMS)

Roli dlouhodobého datového úložiště aplikace plní, vzhledem k relačnímu charakteru dat, typicky *relační databáze*. Přesněji řečeno systém, který ji obsluhuje. Ten především zajišťuje konzistentní sestavení, čtení a zápis relacemi svázaných dat a rovněž řídí autorizaci obecné manipulace s nimi.

Pro komunikaci s databázovým systémem se používá výhradně *strukturovaný dotazovací jazyk*.

Z hlediska bezpečnosti je spojení s databázovým systémem typicky omezeno na úzký okruh lokální podsítě serveru, případně přímo jen na mezi-procesovou komunikační frontu v rámci operačního systému serveru. Za těchto podmínek pochopitelně bezprostředně s databázemi pracuje výhradně serverový interpret.

Přestože databázový systém řídí i autentizaci spojení, samotné fyzické úložiště běžně nebývá nijak zabezpečeno. Neoprávněný přístup k systému souborů databáze je tak velmi vážným bezpečnostním rizikem.

1.2.3 JavaScript

JavaScript je, opět nikoliv výhradně, interpretovaný skriptovací jazyk běžící převážně v prostředí webového prohlížeče. Je dynamický, prototypově orientovaný, se slabou typovou disciplínou. (6)

Formálně jde o implementaci jazyka *ECMAScript* Evropské asociace počítačových výrobců (ECMA).

JavaScript slouží převážně k programové interakci a zpracování událostí *Objektového modelu dokumentu* (DOM), případně dalších extenzí webového prohlížeče, jako jsou například ovladače Microsoft Active-X.

Klíčovou schopností JavaScriptu je, v perspektivě moderních webových aplikací, možnost asynchronní komunikace s webovým serverem.

Přes svůj název nemá, mimo předlohy v jazyce C, nic společného s *Javou*.

1.2.4 Platforma Flash

Flash je multimediální platforma, dříve určena především pro tvorbu křivkových animací, posléze interaktivních animací, později pak webových aplikací a her, v současné době pak obecně internetových aplikací.

Podobně jako JavaScript, i Flash používá implementaci ECMAScriptu – označovanou zde jako *ActionScript*. (7)

Flashové programy jsou vždy kompilovány. Součástí HTML dokumentu pak mohou být jako externí vložené objekty zobrazené pomocí zásuvného modulu prohlížeče pro programové aplikační rozhraní Netscape (*NPAPI*) nebo pomocí ovladače Active-X.

Vzhledem ke koherentnímu vývoji technologie poskytuje ActionScript, na rozdíl od JavaScriptu, některé spolehlivé nadstandardní nástroje především co se internetové komunikace, interakce s uživatelem a práce se surovými binárními daty týče.

1.3 Kryptografie a řízení přístupu k privátním datům

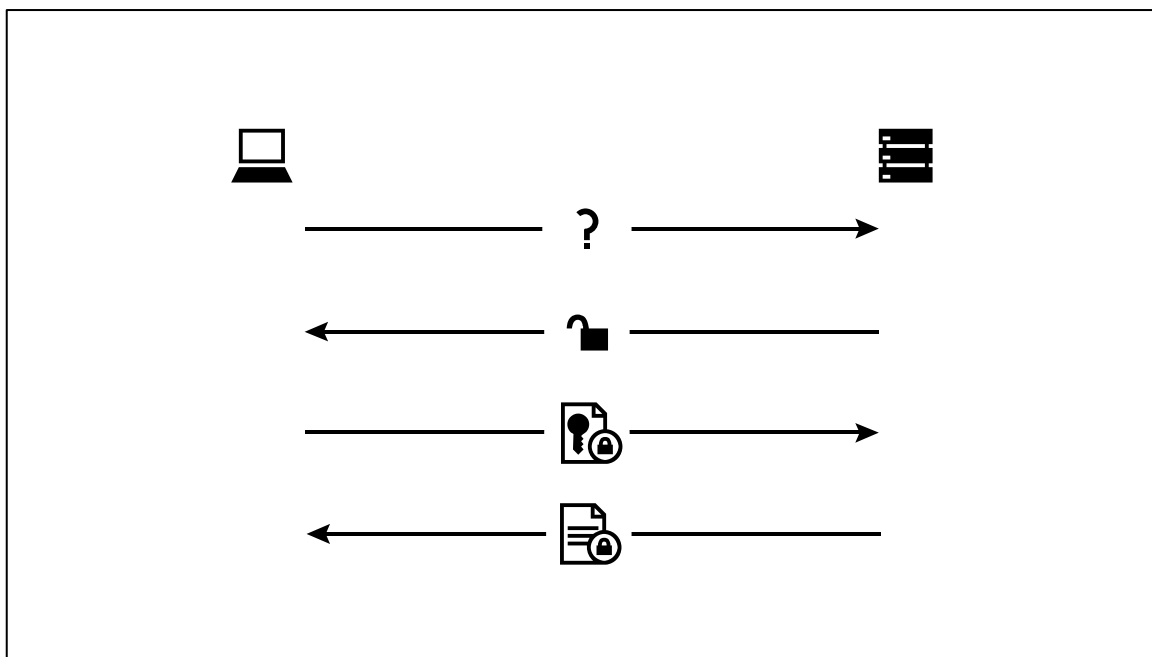
1.3.1 Zabezpečená přenosová vrstva nad HTTP (TLS/SSL)

Ze samotného principu fungování *linkové vrstvy* v síťové architektuře internetu je třeba trasu přenosu dat považovat za nezabezpečenou. Je tedy nutné přijmout opatření na úrovni *aplikační vrstvy*, která brání odposlouchávání a neoprávněným zásahům do přenášených dat.

TLS a jeho předchůdce *SSL* jsou souhrnem kryptografických protokolů a komunikačních procedur fungujících pod aplikačním protokolem, jako je typicky HTTP nebo FTP. (8)

Úkolem TLS je provést několik procedur v určeném pořadí, jak znázorňuje obrázek 2.

1. Klient a server si vymění informaci o podporovaných kryptografických metodách. Server pak rozhodne o jejich použití.
2. V závislosti na zvolených šifrách si pak server i klient předají své *šifrovací certifikáty* obsahující asymetrické veřejné šifrovací klíče.
3. Klient i server pomocí pseudonáhodné funkce zvolí hlavní symetrický šifrovací klíč a předá jej svému protějšku v, jeho veřejným klíčem, asymetricky zašifrované podobě.
4. Posledním krokem je kontrola konzistence předchozí komunikace předáním symetricky zašifrovaných otisků přenesených dat. Veškerá následující komunikace pak probíhá v symetricky šifrované podobě.



Obrázek 2: Zjednodušené schéma předání symetrického šifrovacího klíče

Jak zjednodušeně znázorňuje obrázek 2, klient si vyžádá veřejný klíč od serveru, ten mu jej v nezašifrované podobě předá, klient asymetricky zašifruje symetrický klíč a předá jej serveru, který pak již může komunikovat v symetricky šifrované podobě.

1.3.2 Symetrické šifrování dat

Symetrické šifry se vyznačují použitím jediného privátního klíče, který symetricky šifruje i dešifruje data.

Tento způsob šifrování se v praxi běžně používá pro zabezpečení samotných surových dat na nejnižší úrovni souboru kryptografických procedur.

Hlavní předností symetrického šifrování jsou nízké nároky na výpočetní výkon. Nevýhodou jsou naopak vysoké nároky na ochranu samotného klíče, který musí být zajištěn proti zcizení respektive odposlechnutí. (9)

Ochranu symetrických klíčů typicky plní asymetrické šifrování.

1.3.2.1 DES a AES

DES a jeho nástupce AES je dvojice symetrických šifrovacích algoritmů, které patří mezi dominantní pro zabezpečení webové komunikace a webových datových úložišť.

Jedná se v obou případech o blokovou šifru. Délka klíče je 56 bitů v případě DES a až 256 bitů v případě AES. (9)

Samotný DES není v současné době považován za dostatečně odolný vůči kryptoanalýze, používá se tedy převážně ve tříprůchodové variantě jako *TripleDES* – za což ovšem platí výrazně vyššími výpočetními nároky. (9)

1.3.3 Asymetrické šifrování dat

Asymetrické šifry pracují vždy s dvojicí odlišných jednosměrných klíčů – veřejným šifrovacím klíčem a soukromým dešifrovacím klíčem. Obě strany tak nemusí před zahájením komunikace sdílet žádná tajná data.

Jak už plyne z názvu, veřejný klíč je možné předat druhé straně v nezabezpečené podobě.

Analogickou kryptografickou metodu lze použít i pro *podepisování dat*, tedy ověření, že data na straně příjemce odpovídají datům na straně odesílatele. V takovém případě se role klíčů obrátí a šifrovací klíč je považován za privátní. Příjemce pak může pomocí veřejného dešifrovacího klíče ověřit, že podpis odpovídá přijatým datům.

1.3.3.1 RSA

Dominantním představitelem asymetrického šifrování je *RSA*.

Bezpečnost této šifry je založena na výrazné asymetrii matematické obtížnosti násobení prvočísel a jejich zpětné faktorizace. Velikost a míra náhodnosti těchto prvočísel je pak klíčovým faktorem síly šifry. (9)

Vzhledem k tomu, že šifrovaná data, v případě RSA, nemohou být delší, než je délka veřejného klíče, omezuje se v praxi použití této metody především na zabezpečení přenosu symetrických klíčů pro následnou již symetricky šifrovanou komunikaci. (9)

1.3.4 Asymetrický otisk dat

Otisk dat je produktem algoritmu, označovaným jako *hash* (anglicky *sekané maso*), který z libovolně velkého bloku dat vytvoří výběr o konstantní velikosti.

Samotný hash algoritmus má některé důležité vlastnosti. (9)

- Je jednosměrný. Nelze reálně nalézt vstup podle výstupu.
- Náročnost výpočtu je nízká i pro velký objem vstupních dat.
- Nelze reálně nalézt dva kolizní vstupy pro jeden výstup.
- I drobná změna vstupu vyvolá rozsáhlou změnu výstupu.

V praxi jsou využívány především dvě klíčové vlastnosti otisku dat. Jeho bezkoliznost a lavinovitá změna výstupu podle vstupu pro *kontrolu konzistence dat* a jeho jednosměrnost pro bezpečné uložení dat, která není třeba číst, ale jen porovnávat – typicky *hesla*.

Dlužno dodat, že zmiňovaná nenáročnost výpočtu otisku je výhodná pro kontrolu konzistence velkého množství dat, ale značně problematická pro ochranu krátkých hesel – usnadňuje totiž použití hrubé síly pro nalezení vstupu funkce.

1.3.4.1 MD5 a SHA

MD5 a *SHA-1* patří mezi nejpoužívanější hash algoritmy v prostředí webových aplikací. Produkují otisk o délce 128 respektive 160 bitů. U obou však byla zpochybněna jejich bezkoliznost – v případě *SHA-1* teoreticky, v případě *MD5* již i včetně praktického ověření. (9)

Pro ochranu důležitých dat je tedy vhodné použít variantu *SHA-2* produkující až 256bitové otisky.

2 HLAVNÍ BEZPEČNOSTNÍ RIZIKA INTERNETOVÝCH APLIKACÍ

Pro výběr hlavních bezpečnostních rizik aplikace předpokládejme, že oblasti, které můžeme při návrhu ovlivnit, se omezují jen aplikaci samotnou a nikoliv na ostatní systémy, nad kterými bude fungovat.

2.1 Chybné zpracování nedůvěryhodných dat

Jedním z nejčastěji zneužívaných bezpečnostních rizik webových aplikací je chybné zpracování nedůvěryhodných dat. To pramení ze samotného modelu aplikace, která je složena ze dvou oddělených částí – klientské a serverové – spojených prostřednictvím HTTP komunikace.

Z podstaty věcí neexistuje nástroj, který by serverové části zaručil integritu klientské části aplikace. Veškerá data přijatá od klienta je tedy nutné apriorně považovat za nedůvěryhodná a na straně serveru prověřit jejich:

- Odpovídající datový typ,
- hodnotu uvnitř povoleného rozsahu,
- přítomnost řídicích znaků.

2.1.1 Injektáž kódu

Na základě přijatých dat je často třeba sestavit dokument, dotaz nebo direktivu pro další interpretaci. Je kriticky důležité zajistit, aby surová data nemohla obsahovat řídicí sekvence znaků, které by interpretaci mohly ovlivnit.

Nejčastěji je taková chyba zneužívána při sestavování SQL řetězců a samotných HTML dokumentů.

2.2 Kompromitace osobních údajů

Jednou z prvořadých povinností provozovatele webové aplikace je zajistit bezpečnost osobních údajů uživatelů. Obzvláště to platí u aplikací zprostředkovávajících obchodní styk a manipulujících tak s reálnými platebními prostředky.

Za mimořádně citlivé údaje, které mají pro útočníky reálnou cenu, lze považovat především čísla a ověřovací kódy kreditních a debetních karet a čísla účtů. Vzhledem

k enormní výši potenciální škody vzniklé kompromitací těchto dat, je nezbytně nutné přijmout některá bezpečnostní opatření nad běžný rámec ochrany.

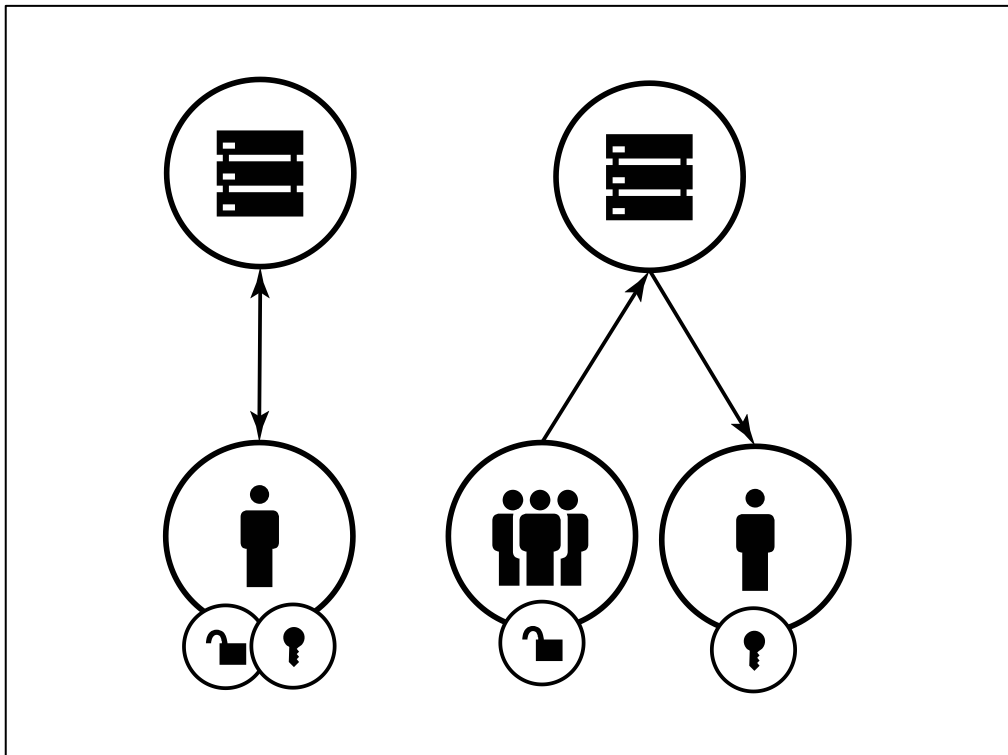
- Integrita infrastruktury webového serveru už není implicitně předpokládána. Data úložiště je třeba šifrovat.
- Také přenos dat mezi klientem a serverem musí být šifrován. To platí i pro jiné komunikační kanály – například e-mail.
- Přístup k privátnímu klíči šifrovaných dat musí být omezen a kontrolován. Ideální je jeho umístění na externí paměťové médium v symetricky šifrované podobě.

2.2.1 Zabezpečení dat úložiště

Kriticky citlivá data úložiště je třeba při volbě šifrovací metody rozdělit do dvou oblastí.

1. Oblast dat, kterou vytváří i spravuje subjekt se shodnou úrovní pravomocí. V praxi se jedná typicky o interní informace, se kterými pracují správci aplikace. Taková data je možné šifrovat symetricky.
2. Oblast dat, kterou vytváří veřejný subjekt a spravuje subjekt s vysokou úrovní pravomocí, správce. Zde je nutné využít asymetrické kryptografie.

Jak již bylo zmiňováno dříve a jak ukazuje obrázek 3, pokud pracujeme s předpokladem nezajištěné integrity webového serveru, šifrování i dešifrování musí provádět vždy klient, server působí jen jako prostředník.



Obrázek 3: Symetrické a asymetrické šifrování dat úložiště

2.2.2 Phishing a Pharming

Termín *Phishing* a *Pharming*, označuje soubor sociálně zaměřených metod útoku na uživatele aplikace. Obrana proti těmto technikám je problematická, protože místo chyb zabezpečení aplikace využívají neznalosti a nepozornosti uživatelů za účelem vylákání jejich osobních údajů.

Z hlediska návrhu aplikace je možné některé metody útoků ztížit nebo i úplně potlačit.

- Pokud provozovatel komunikuje s uživateli nezávislým kanálem, typicky pomocí e-mailu, měl by přijmout striktní politiku obsahu této komunikace, se kterou by měl také uživatele seznámit. Běžně by to měla být například politika nevyžadovat jakékoliv osobní informace.
- Šifrovací certifikáty webového serveru by měly používat *rozšířené ověření*, které garantuje a u většiny webových prohlížečů i viditelně zobrazuje také provozovatele serveru. Zcela analogicky by i e-maily měly obsahovat elektronický podpis garantující odesílatele a integritu obsahu korespondence.

- Doména serveru by měla být zajištěna proti neoprávněné manipulaci s DNS záznamy certifikátem *DNSSEC*.

2.3 Neoprávněná autorizace

Webovou aplikaci běžně obsluhuje několik skupin uživatelů s různou úrovní pravomocí. Neoprávněné získání autorizace k vyšším pravomocem použití aplikace je jedním ze základních bezpečnostních rizik.

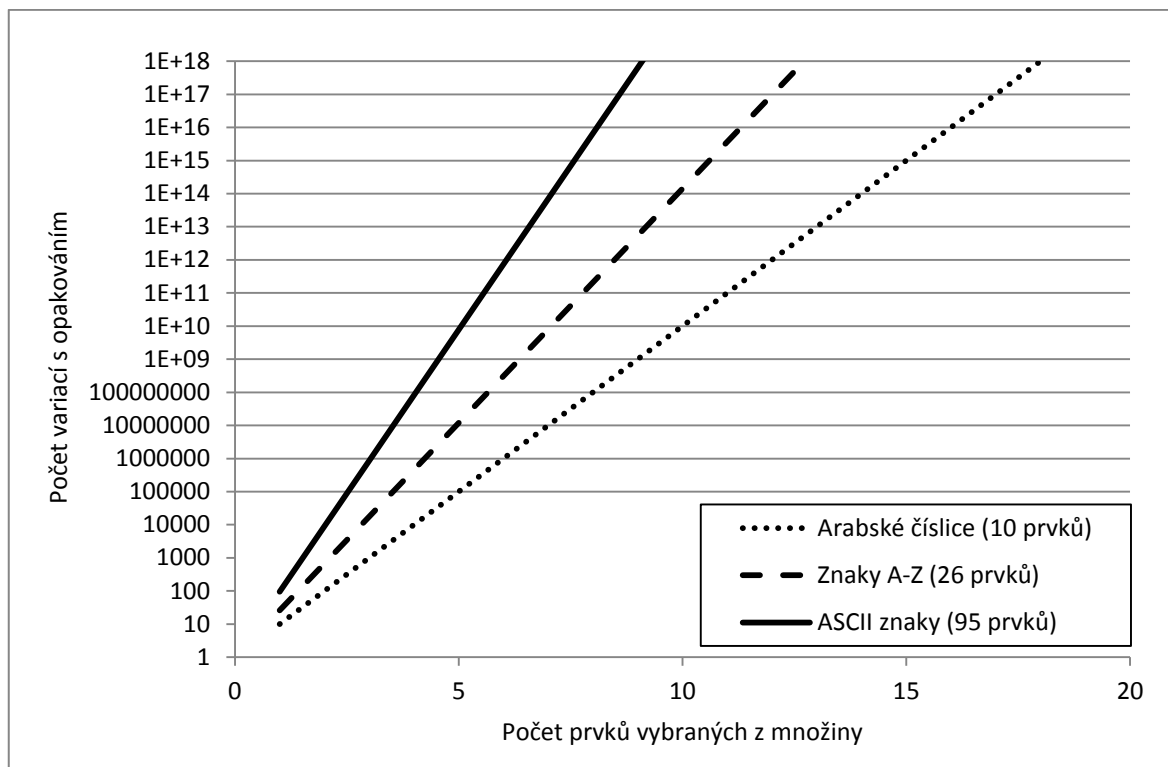
Nejběžnější metoda autorizace uživatele je vyžádání a ověření tajného klíče – *hesla*.

2.3.1 Hrubá síla

Klasickou metodou překonání autorizace heslem je pokus jej uhádnout. V případě, že je heslo shodné s nějakým běžným výrazem, je možné s úspěchem využít *slovníkový útok*, kdy je hodnota hesla nalezena v řádech sekund.

V ostatních případech je třeba vyzkoušet všechny variace množiny znaků o požadované délce. Zde je třeba zdůraznit, že počet variací roste parabolicky s délkou výrazu a exponenciálně s velikostí množiny. Dostatečná délka hesla tedy může ospravedlnit malou množinu znaků, což bohužel provozovatelé běžně ignorují a nutí uživatele volit obtížně zapamatovatelné výrazy.

Jak ukazuje obrázek 4, u sedmimístného alfanumerického hesla lze zvýšit míru entropie stejnou měrou zapojením nestandardních ASCII znaků, jako rozšířením délky na deset znaků.



Obrázek 4: Srovnání vlivu délky výrazu a velikosti množiny prvků na sílu hesla

Typickou ochranou proti použití hrubé síly při autentizaci je umělé snížení počtu pokusů ověření, které mohou za určenou časovou jednotku proběhnout.

2.3.2 Zabezpečení hesel

Kompromitace hesla je jedním ze základních způsobů neoprávněné autorizace.

Vzhledem k tomu, že se ověření u tohoto typu autorizace omezuje jen na ověření shodnosti hesla, je možné pro porovnání použít jen jeho asymetrický otisk.

Při volbě vhodného algoritmu pro otisk hesla je třeba brát v úvahu jeden z rysů běžně používaných hash funkcí – vysoká rychlost výpočtu, která v tomto případě usnadňuje použití hrubé síly, jejíž použití už nemůže být, v případě krádeže dat, kontrolováno.

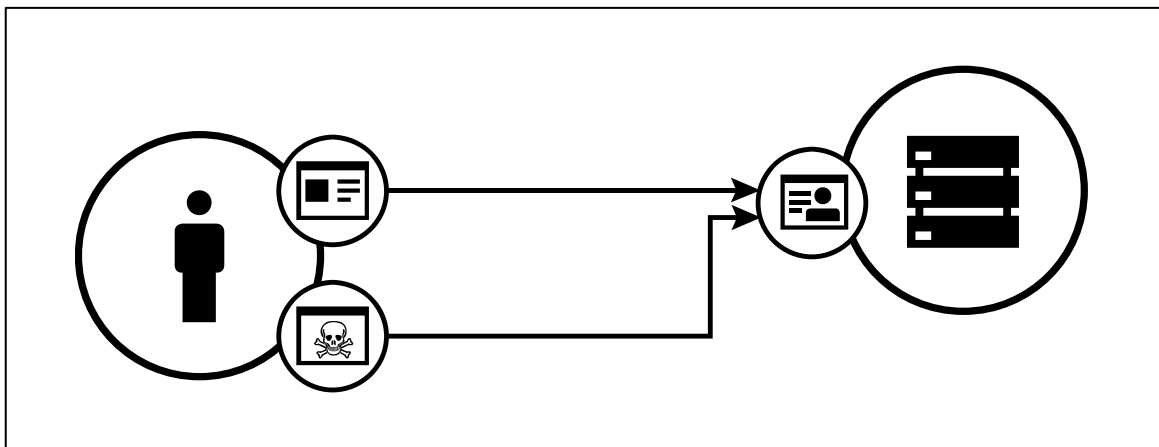
Kritická je tedy úroveň entropie, dosažená délkou nebo složitostí, výrazu.

Další velmi podstatnou ochranou otisku před použitím hrubé síly je rozšíření zdrojového výrazu o libovolně zvolenou deterministickou hodnotu – takzvaný *salt* (*sůl*). Připojení *saltu* znemožní použití předgenerovaných tabulek otisků pro získání zdrojového výrazu – takzvaných *Rainbow tables*.

2.3.3 Podvržení požadavků klienta (XSRF)

Nestavový charakter HTTP komunikace a hypertextová provázatelnost webu otevírá možnost donutit klienta nevědomky sestavit a odeslat HTTP požadavek na jinou webovou aplikaci.

Jak zjednodušeně ukazuje obrázek 5, v praxi útočící strana zneužívá nejčastěji v tu chvíli platné a autentizované relace klienta na cílovém webu a z nelegitimního zdroje se pokouší provést neautorizovanou akci.



Obrázek 5: Zneužití autentizované relace požadavkem z nelegitimního zdroje

Vzhledem k tomu, že protokol HTTP neposkytuje *spolehlivou* metodu, jak určit zdroj požadavku, pro bezpečnostně kritické akce je třeba zajistit, aby požadavek obsahoval unikátní kontrolní parametr, který útočící strana není schopna pro jeho validní sestavení získat.

2.4 Narušení funkčnosti (Denial of Service)

Narušení funkčnosti služby lze považovat za méně nebezpečný druh útoku na aplikaci. Většinou jde o nějakou formu úmyslného vyčerpání systémových prostředků, která brání ostatním uživatelům v jejím plnohodnotném použití.

Pokud zůstaneme na aplikační úrovni, máme jen málo nástrojů, jak zranitelnost ovlivnit, velká část zodpovědnosti leží totiž na samotné obsluze spojení, frontových disciplín a systémových procesů, kterou řídí HTTP server potažmo operační systém.

Během tvorby aplikace se tedy soustředíme především na omezení programových procedur, které by vyčerpání systémových prostředků snadno umožňovaly. Jedná se především o:

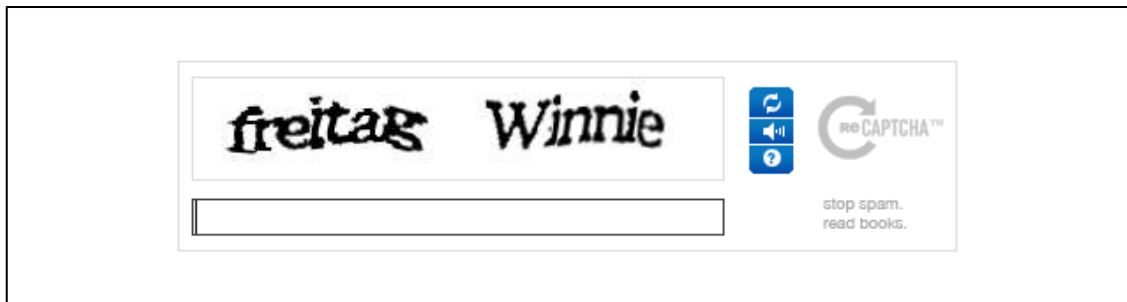
- Kontrolu vstupních parametrů, jestli kromě formální správnosti mají i logický smysl. To se týká například nepřírodně dlouhých výrazů pro vyhledávání, nebo nepřírodně velkých rozměrů pro generování náhledů obrázků.
- Omezení časově náročných operací nebo jejich přesunutí paralelně mimo vliv HTTP serveru a jeho klientů.

2.4.1 Strojové zneužití služby

Některé společné rysy s narušením funkčnosti aplikace má strojové zneužití jejich služeb. To si neklade za cíl vyčerpání její systémové prostředky, ale zneužít ji napodobením běžné lidské interakce.

Typickým příkladem je zneužití komentářů nebo e-mailových terminálů k publikaci nevyžádaných sdělení – charakteristicky v enormním množství za krátký čas.

Obrázek 6 ukazuje typickou ochranu proti takovým aktivitám – *plně automatický Turingův test k rozlišení počítačů od lidí (CAPTCHA)*.



Obrázek 6: Ukázka projektu *reCAPTCHA* (zdroj recaptcha.net)

II. PRAKTICKÁ ČÁST

3 VOLBA VHODNÝCH TECHNOLOGIÍ, NÁSTROJŮ A PODOBY WEBOVÉ APLIKACE

Aplikace pro elektronické obchodování je bezesporu jedním z nejrozmanitějších programových celků. Pramení to pochopitelně z potřeb provozovatelů, jenž se liší podle obchodní oblasti, které bude aplikace sloužit.

V našem případě není rozhodně cílem obsáhnout všechny možnosti a činnosti, které by potenciální klient chtěl nebo mohl vykonávat. Praktická část se raději soustředí na nejužší funkční základ a bude mít za cíl demonstrovat primární operace především z pohledu bezpečnosti, výkonu a návrhové vhodnosti.

3.1 Serverový programový interpret, databázový systém a aplikační rámce

Při volbě se přidržíme hlavního proudu v praxi používaných a licenčními poplatky nezátížených technologií.

To tedy na serverové straně znamená využití interpreta PHP a databázového systému *MySQL*. Co se samotného HTTP serveru týče, aplikace neklade žádné zvláštní požadavky a je otestována na serveru *Apache* i *Microsoft IIS*.

3.1.1 Nette Framework

Nette Framework je open-source aplikační rámec, napsaný v PHP, který usnadňuje některé základní činnosti a řeší některé běžné úkoly webové aplikace v prostředí PHP.

V našem případě je to především zjednodušení:

- Rozdělení aplikace na část prezenční, obslužnou a datovou,
- udržování cest k obslužným objektům podle adres požadavků,
- zpracování dat z požadavků a sestavování odpovědí z šablon,
- komunikace s databází a databázová abstrakce,
- udržování relací.

3.1.2 Apache Flex

Apache Flex je open-source aplikační rámec respektive vývojová sada pro tvorbu *Flash* aplikací. V našem případě bude využita pro vytvoření asymetrického šifrovacího modulu.

3.1.3 JQuery

JQuery je JavaScriptová knihovna objektů zjednodušující a sjednocující běžné operace JavaScriptu napříč odlišnými implementacemi jazyka v různých verzích a typech webových prohlížečů.

3.2 Funkční rozsah aplikace

Pro demonstraci základních operací elektronického obchodu vytvoříme funkční aplikaci, která bude vykonávat především následující činnosti.

1. Přehledné zobrazení položek k prodeji s viditelnou cenou, dostupností a obrazovým náhledem, seřazených podle uživatelem zvoleného kritéria.
2. Detail položky se všemi náhledy, podrobným popisem, historií vývoje ceny, popisem dostupnosti a možností položku přidat do košíku.
3. Samotná správa virtuálního košíku s možností měnit počty položek a sledovat výslednou cenu.
4. Objednání vybraných položek, volbu dopravy a platby.
5. Odeslání objednávky spojenou s automatickou registrací.
6. Demonstraci platby pomocí šifrovaného vložení údajů kreditní či debetní karty.
7. Sledování stavu objednávky po přihlášení ke svému účtu.
8. Jednoduchou administraci obsahu pro správce aplikace.
9. Administraci objednávek a dešifrování platebních údajů pro správce aplikace.

4 MODEL DATABÁZOVÉHO ÚLOŽIŠTĚ

Pro tvorbu modelu MySQL databáze a vykreslení *EER* diagramu jsme zvolili volně dostupný nástroj *MySQL Workbench*.

Při návrhu byly sledovány především následující cíle.

- **Minimální redundance dat**

Především u časově proměnných hodnot, jako jsou typicky ceny, to znamená jejich separace a zachování pro potřeby časově závislých výpočtu, jako jsou například přehledy starých objednávek.

- **Snadná rozšiřitelnost**

Veškeré textové data jsou odděleny do jedné tabulky a označeny identifikátorem druhu a jazyka. Budoucí rozšíření počtu jazyků nebo textových hodnot se tak obejde bez zásahu do struktury databáze.

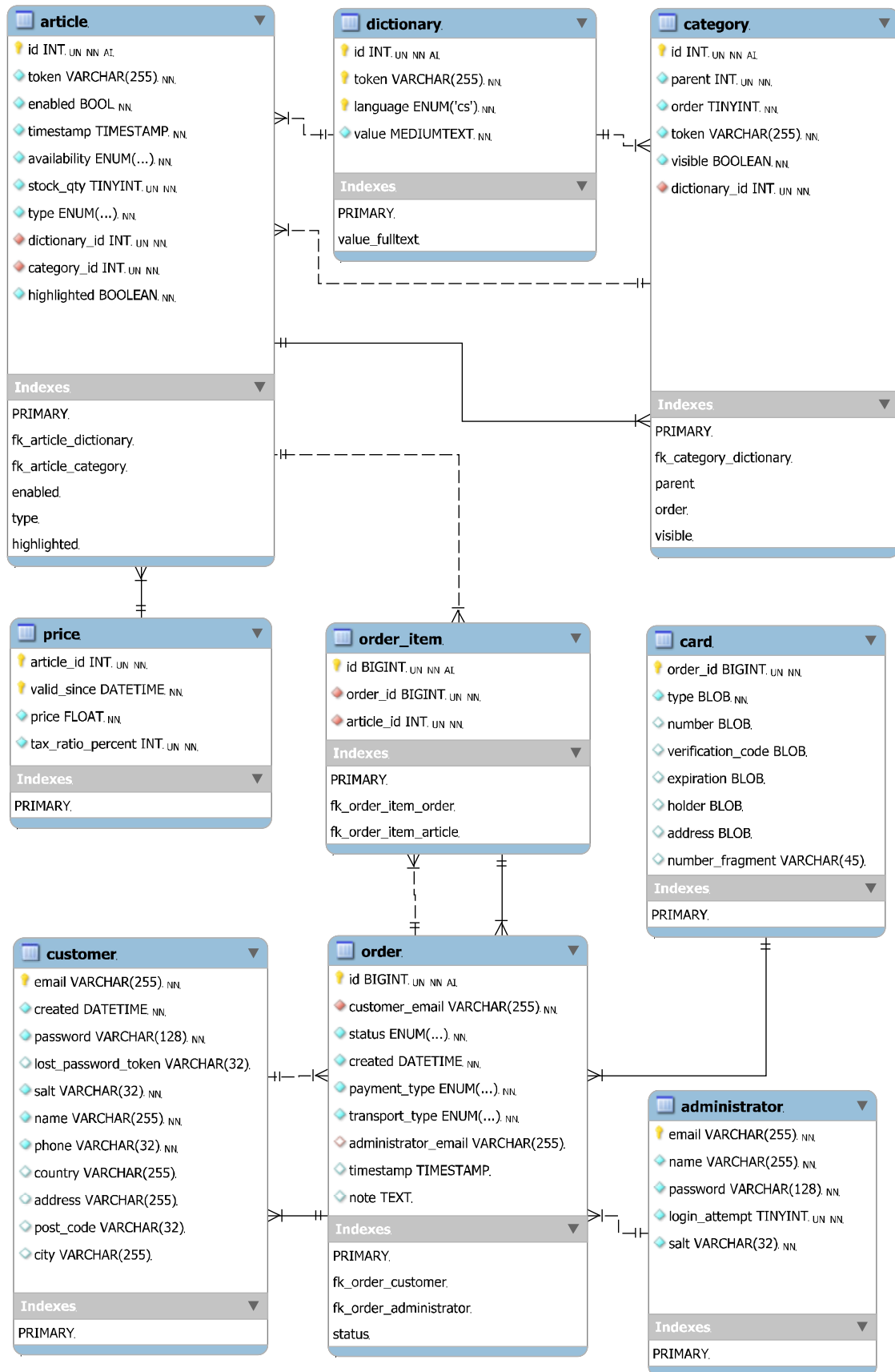
- **Nízké nároky na výkon databázového systému**

Samozřejmostí je využití indexů pro selekci cizích klíčů a fulltextové vyhledávání.

Jak názorně ukazuje obrázek 7, klíčové datové objekty aplikace reprezentují tabulky databáze.

- Prodejní položku reprezentuje tabulka *article*, která je zařazena do kategorie v tabulce *category*.
- Tabulka *dictionary* plní roli centralizovaného úložiště textových hodnot položek i kategorií.
- Položka má stanoveny ceny v tabulce *price*. Ty jsou časově závislé a kumulují se pro potřeby časově závislých výpočtů.
- Objednávky reprezentuje tabulka *order* označující zákazníka v tabulce *customer*, kolekci vybraných položek v tabulce *order_item*, odpovědného administrátora v tabulce *administrator* a případně také údaje o platební kartě v tabulce *card*.

Některé z těchto objektů si podrobněji popíšeme dále.



Obrázek 7: EER diagram databázového úložiště

4.1 Produkt

Nabízený prodejní produkt charakterizuje tabulka *article* a trojice pomocných tabulek, jak detailně ukazuje obrázek 8. Každý záznam v tabulce produktů je identifikován číselnou hodnotou *id* a ASCII řetězcem *token* – ten slouží jen pro tvorbu čitelnějších a lépe vyhledatelných URL adres.

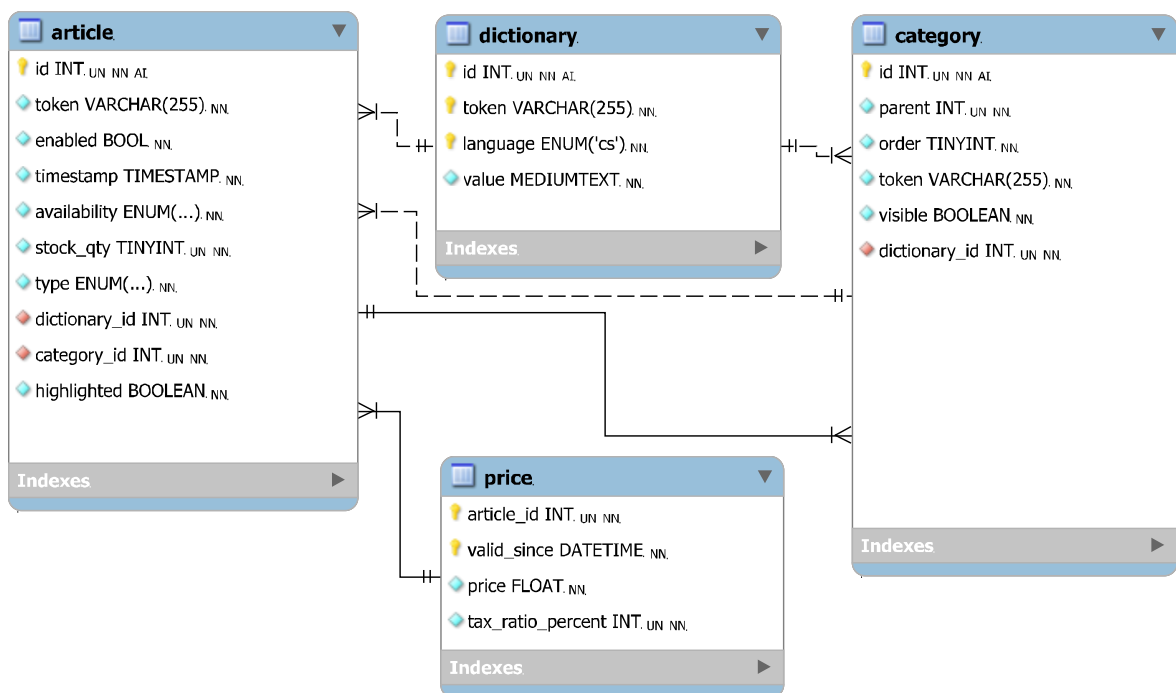
Booleovské hodnoty *enabled* a *highlighted* určují, zda je položka zobrazena respektive zvýrazněna.

Kombinace hodnot *availability* a *stock_qty* stanovuje aktuální dostupnost produktu respektive jeho množství na skladě.

Kromě zboží může tabulka obsahovat také služby. Ty jsou pochopitelně například nevyčerpatelné. Toto rozlišení stanovuje hodnota *type*.

Položky jsou zařazeny do kategorií definovaných tabulkou *category*, která má podobu běžného jednosměrného stromu. Podobně jako u položek nabídky jsou ke kategoriím připojeny textové informace a ASCII *token* pro tvorbu adresy.

Vývoj ceny položky v čase definuje tabulka *price*.



Obrázek 8: Produktová část diagramu úložiště

4.2 Objednávka

Objednávky jsou sdruženy v tabulce *order*, jejich obsah pak definuje tabulka *order_item* výběrem řádků tabulky *article*. Obrázek 9 naznačuje také vztah k dalším datovým celkům, které jsou vysvětleny dále.

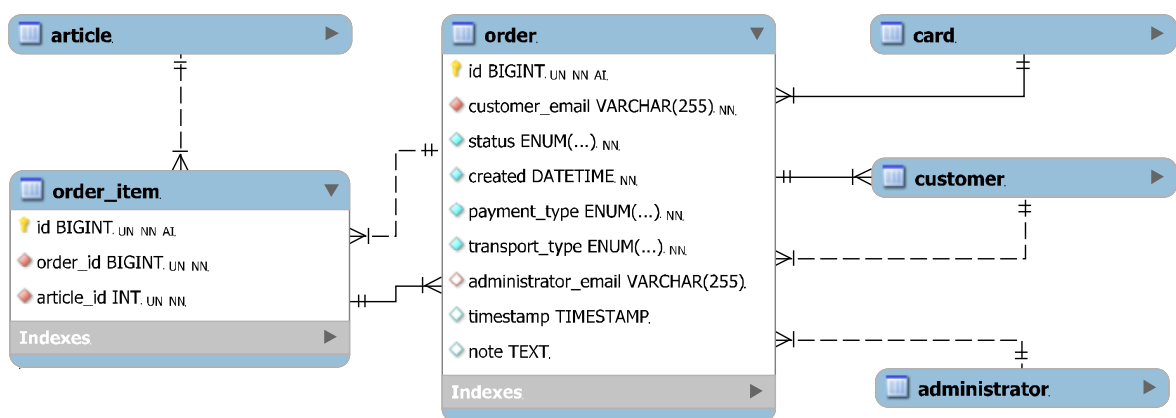
Nakupující zákazník a zodpovědný administrátor je připojen tabulkou *customer* respektive *administrator*.

V případě přímé platby kartou jsou údaje šifrovaně vloženy do tabulky *card*.

Důležitý údaj objednávky je čas jejího vzniku, tedy sloupec *created*, který má vliv na výpočet aktuálních cen položek objednávky.

Jsou připojeny i některé další údaje, jako typ platby a dopravy ve sloupcích *payment_type* a *transport_type*.

Stav vyřízení objednávky sleduje hodnota *status*.



Obrázek 9: Objednávková část diagramu úložiště

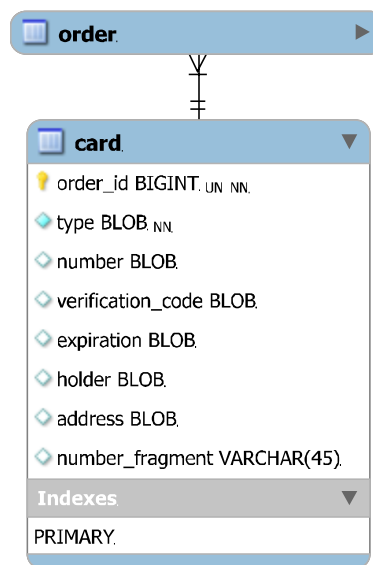
4.3 Údaje o platebních kartách

V případě, že zákazník zvolí platbu zadáním údajů o své platební kartě, jsou tyto zašifrovány a uloženy v tabulce *card*. Její strukturu ukazuje obrázek 10.

Všechny údaje jsou šifrovány, včetně označení typu karty.

V nezašifrované podobě zůstane jen takzvaný fragment čísla karty. Typicky obsahuje první a poslední cifry čísla karty. Tento údaj slouží pouze pro orientaci ve výpisech objednávek.

Často opomíjeným bezpečnostním faktorem je možnost nalezení shodných šifrovaných záznamů. U krátkých a opakujících se hodnot, typicky data expirace nebo typu karty, je vhodné připojit i uživatelův *salt* a shodnost potlačit.



Obrázek 10: Uložení údajů o platební kartě

4.4 Zákazníci a administrátoři

Zákaznické a administrátorské uživatelské účty jsou identifikovány e-mailovou adresou a zabezpečeny otiskem hesla. Heslo je doplněno o náhodný řetězec ze sloupce *salt*.

Jako hash funkce je použit algoritmus SHA512 produkující otisk o délce 128 znaků.

Počet neúspěšných pokusů o přihlášení administrátorů je omezen a následně je zapojena ochrana proti použití hrubé síly ve formě obrazového testu CAPTCHA.

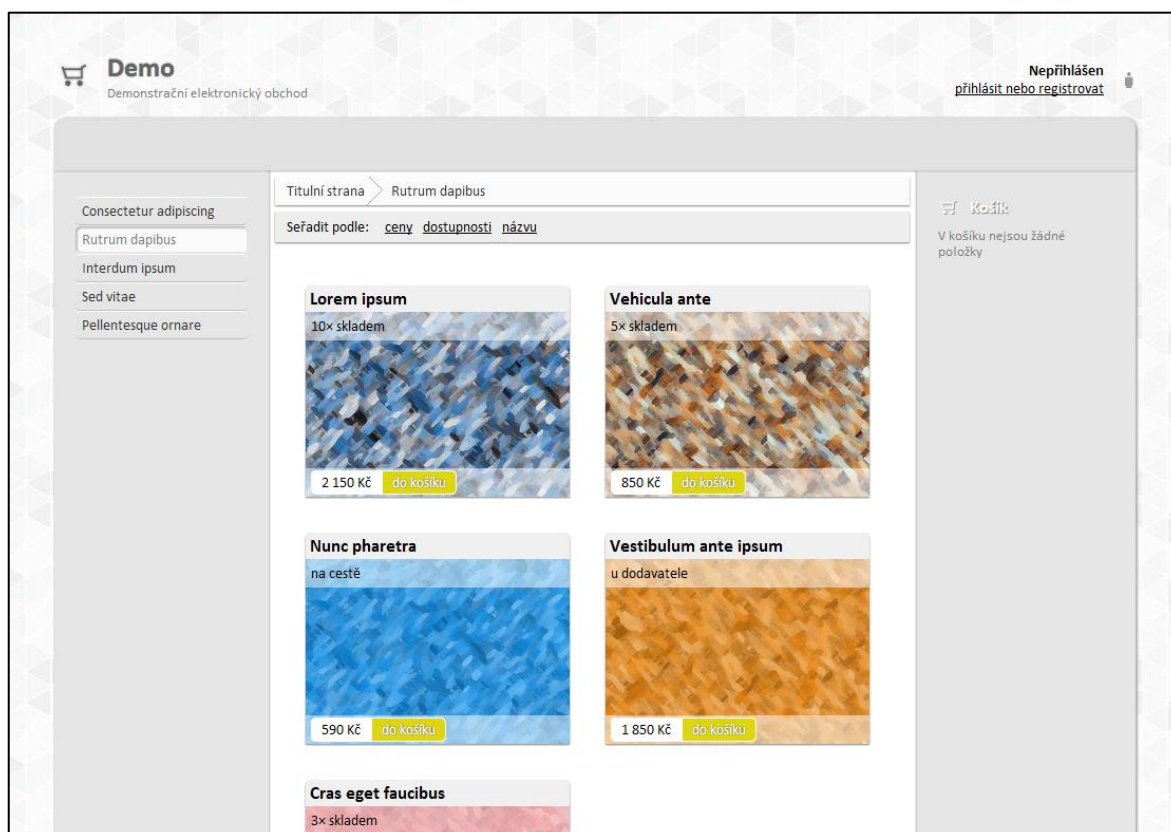


Obrázek 11: Tabulky zákazníků a administrátorů

Jak ukazuje obrázek 11, záznam zákazníka obsahuje také některé kontaktní a fakturační údaje, které pak v případě další objednávky nebude muset opět zadávat.

5 DŮLEŽITÉ ČÁSTI UŽIVATELSKÝCH ROZHRAŇÍ

Hlavním cílem při návrhu klíčových uživatelských rozhraní byla jednoduchá a srozumitelná ergonomie. Při popisu jednotlivých částí rozhraní se budeme držet některých důležitých prvků, které by mohly snadno uniknout pozornosti návrháře aplikace.



Obrázek 12: Kostra HTML rozhraní

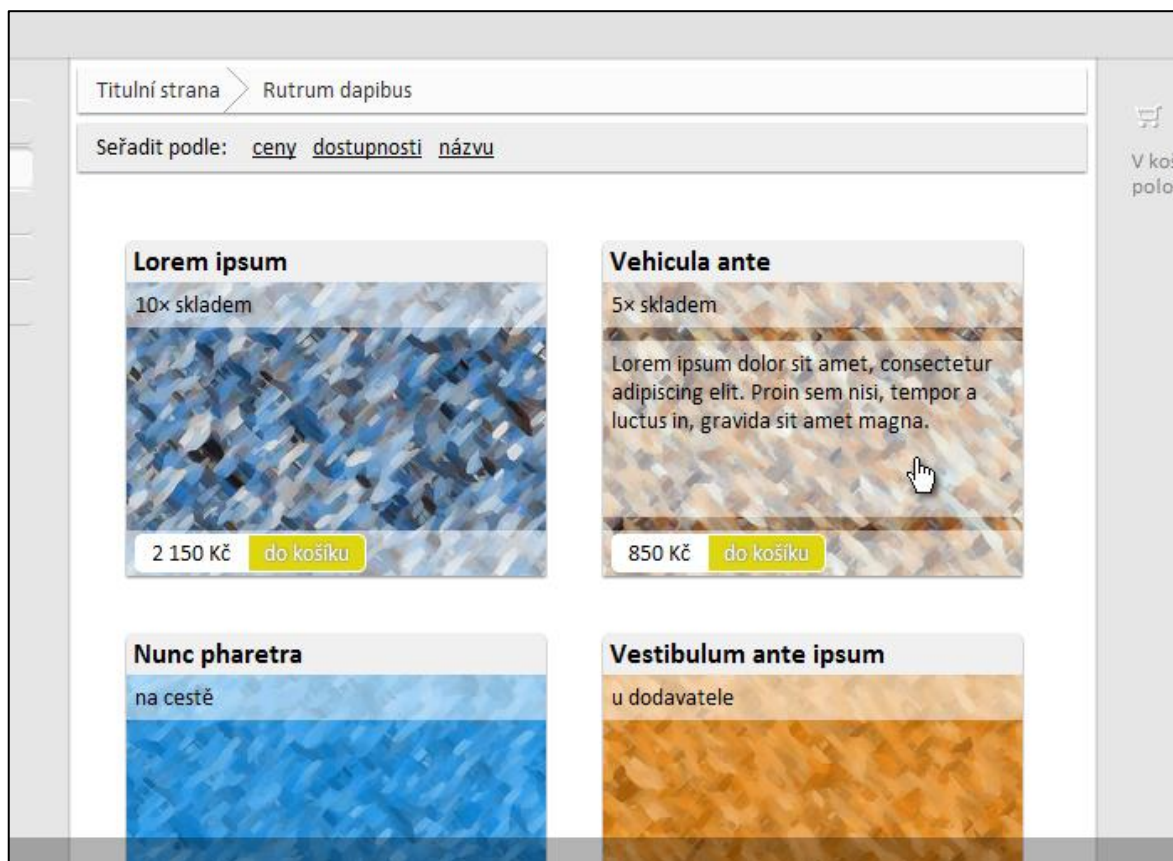
Obrázek 12 ukazuje klasické třísloupcové rozvržení s hlavním menu v levém a nástroji v pravém sloupci.

Dispozice jednotlivých bloků jsou navrženy s ohledem na snadnou parametrickou změnu rozměrů – v praxi ideálně automaticky v závislosti na velikosti uživatelského obrazovacieho prostoru. V našem případě, pro účely snazšího pořizování snímků, fixně zvoleného.

5.1 Seznam položek

Výchozím prvkem rozhraní je seznam položek. Jak bylo již zmíněno v popisu datových objektů úložiště, položky jsou zařazeny do stromu kategorií. Při návrhu jsme sledovali některé důležité cíle, jak ukazuje obrázek 13.

- Zákazník by se neměl ve stromu ztrácet. Je zvýrazněna aktuální položka stromu, aktuální list seznamu a připojena takzvaná *drobečková navigace*.
- Jsou vidět na první pohled klíčové údaje o zboží, jako je cena a dostupnost.
- Položky je možné seřadit nebo filtrovat podle různých kritérií.
- Je zde i možnost rychlého přidání položky do košíku.

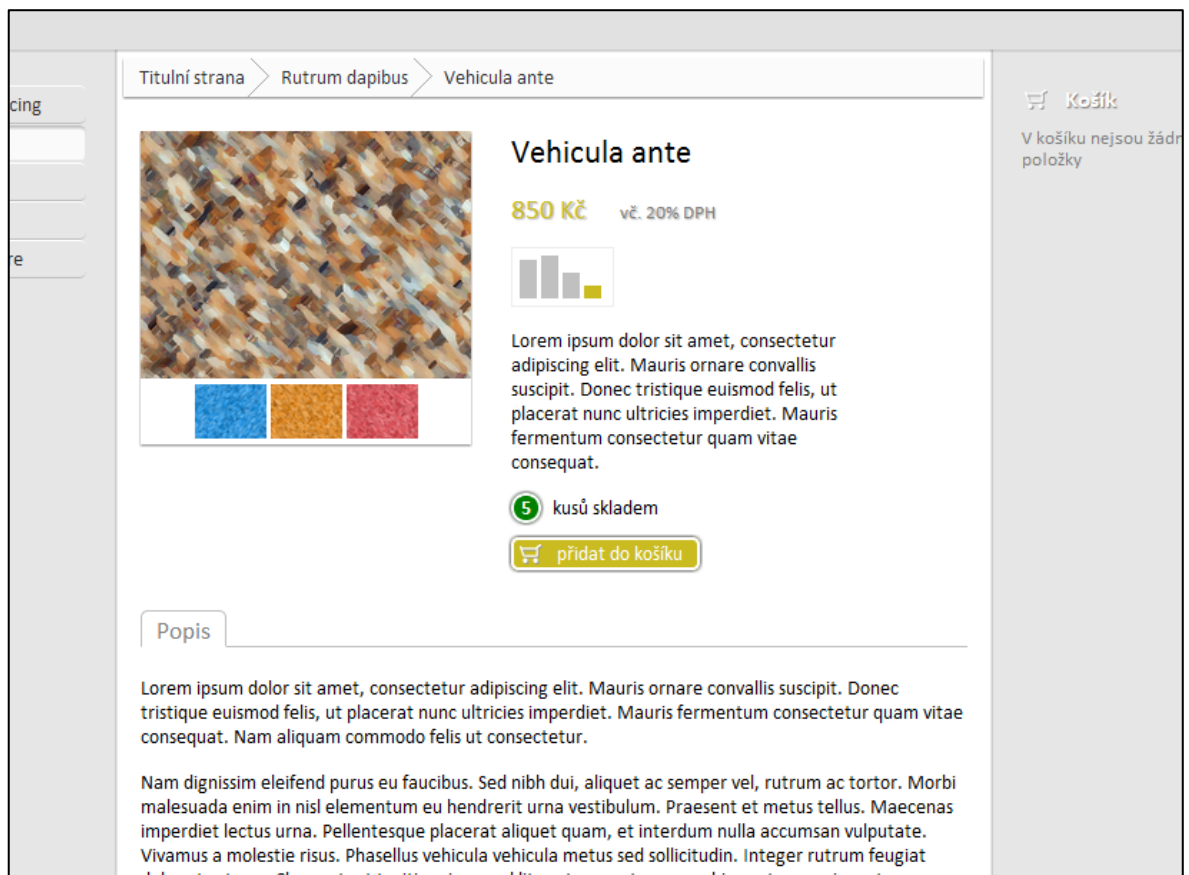


Obrázek 13: Uživatelské rozhraní seznamu nabízených položek

5.2 Detail položky

Obrázek 14 ukazuje detail položky při jejím výběru. Tato část rozhraní patří mezi nejdůležitější a zobrazuje největší množství informací. Klíčovým faktorem při návrhu podoby tohoto rozhraní by měla být přehlednost a srozumitelnost.

- Základem obsahu je detail fotografie, název, dostupnost a cena.
- Vzhledem k tomu, že známe časový vývoj ceny, je možné jej zobrazit.
- Klíčovým ovládacím prvkem je přidání položky do košíku, tento prvek by měl být na první pohled okamžitě rozpoznatelný.
- V praxi je u detailu zboží připojeno celé spektrum doplňujících údajů, většinou technického rázu. V našem případě je to pro účely demonstrace jednoduchý textový popis. Model databáze je nicméně v tomto ohledu neomezeně rozšířitelný.



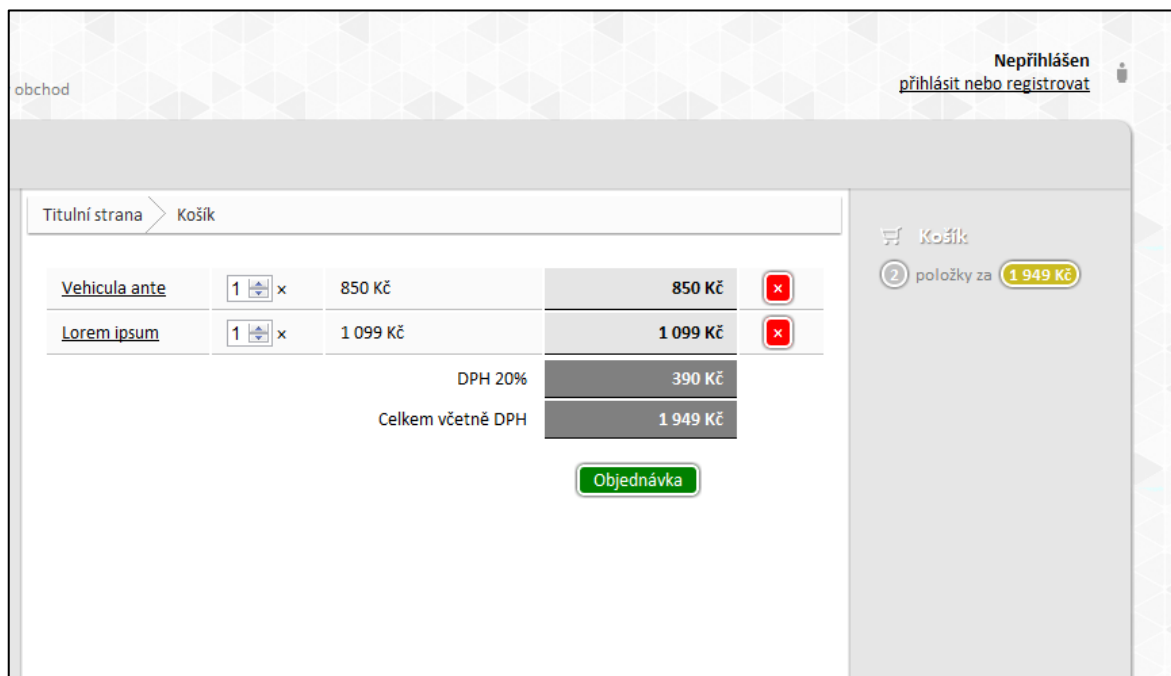
Obrázek 14: Uživatelské rozhraní detailu položky

5.3 Správa košíku

Obrázek 14 s detailem nákupního košíku zobrazuje položky připravené k objednávce.

- Je důležité mít možnost lehce měnit počty kusů položek stejně jako položky zcela odstranit.
- Je zobrazen součet ceny a ideálně také výše daně z přidané hodnoty.

- Zcela klíčovou vlastností košíku by mělo být na první pohled zřetelná možnost přejít k samotné objednávce.



Obrázek 15: Uživatelské rozhraní správy košíku

5.4 Objednávka

Velmi důležitým faktorem úspěšnosti elektronického obchodu je jednoduchost konverze návštěvníka na zákazníka. Sestavení objednávky by tedy mělo být co možná nejméně obtěžující.

Objednávka je proto v našem případě spojena s registrací respektive přihlášením uživatele. Automatická registrace při první objednávce je rozumný kompromis mezi požadavkem obchodníka mít aktuální databázi zákazníků a zákaznickým pohodlím.

- Identifikačním údajem zákazníka je v našem případě jeho e-mailová adresa. Pokud není zákazník přihlášen, nabídneme mu přihlášení zde.
- Před odesláním objednávky je nutná cenová rekapitulace a zahrnutí i nákladů na zvolený druh dopravy. Jak ukazuje obrázek 16, posledním krokem je volba způsobu platby.

obchod Nepřihlášen
[přihlásit](#) nebo [registrovat](#)

Titulní strana > Košík

<u>Vehicula ante</u>	1 ×	850 Kč	850 Kč
<u>Lorem ipsum</u>	1 ×	1 099 Kč	1 099 Kč
DPH 20%			390 Kč
Mezisoučet včetně DPH			1 949 Kč
Obchodní balík ČP			160 Kč
Celkem k úhradě včetně DPH			2 109 Kč

Martin Hozík
+420 123 456 789

Nad Stráněmi 4511
760 05 Zlín
Česká republika

Platba: bankovní převod
 dobírka
 kartou

[Zpět](#) [Odeslat objednávku](#)

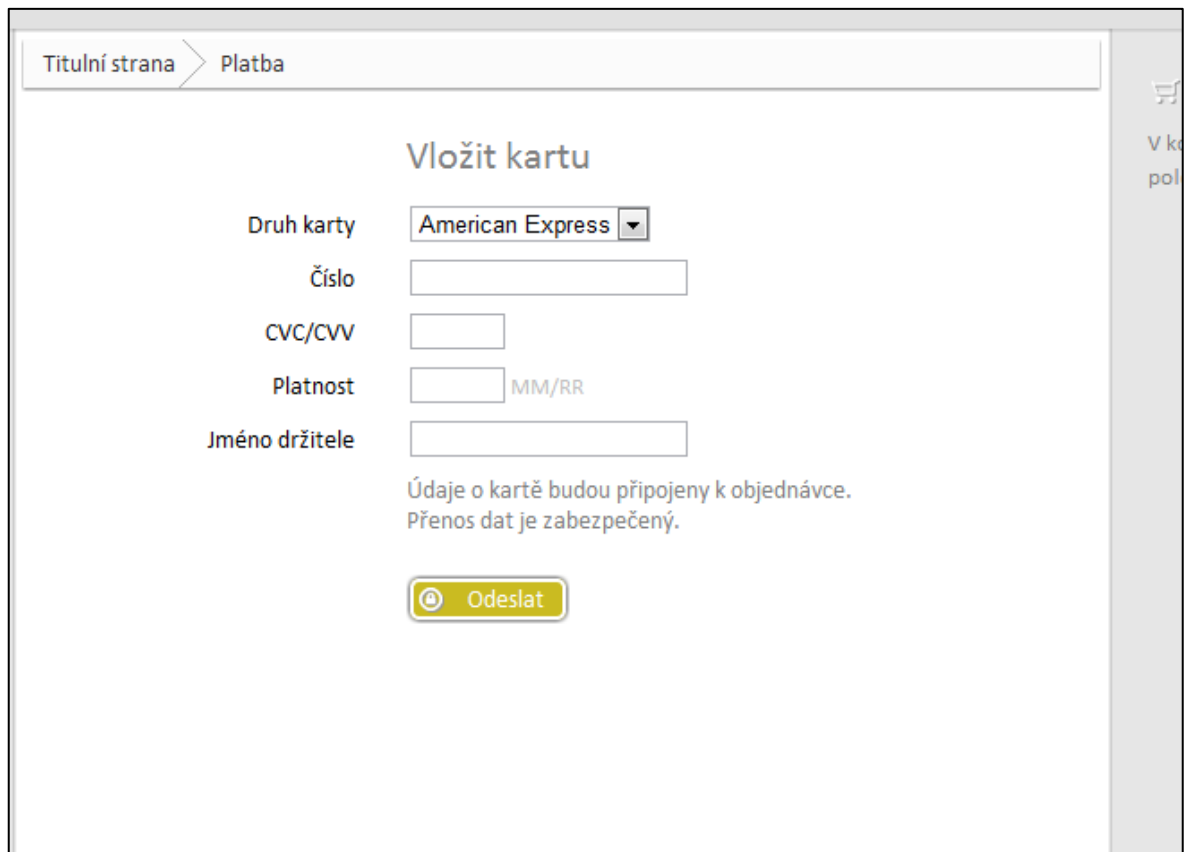
Košík
2 položky za **1 949 Kč**

Obrázek 16: Cenová rekapitulace objednávky

5.4.1 Šifrovací modul údajů o platební kartě

Součástí demonstrační aplikace je i studie zabezpečeného přenosu údajů o platební kartě. V případě zákaznické, veřejné, strany je proces zcela transparentní a mimo běžného vložení údajů o kartě, jak ukazuje obrázek 17, nevyžaduje žádné nestandardní kroky.

Šifrovací modul pak před odesláním dat zašifruje údaje veřejným šifrovacím klíčem.



Titulní strana > Platba

Vložit kartu

Druh karty: American Express ▼

Číslo:

CVC/CVV:

Platnost: MM/RR

Jméno držitele:

Údaje o kartě budou připojeny k objednávce.
Přenos dat je zabezpečený.

Obrázek 17: Formulář pro vložení údajů o platební kartě

5.5 Přihlášení zákazníků a administrátorů

Přihlášení uživatelů je realizováno zadáním uživatelského jména, které v tomto případě reprezentuje e-mailová adresa, a hesla.

Vzhledem k tomu, že kompromitace administrátorského účtu je značným bezpečnostním rizikem, je počet pokusů o přihlášení administrátora omezen na 3 – poté je zapojena ochrana proti nasazení hrubé síly strojově řízenými pokusy o přihlášení. Ochranu plní obrázkový Turingův test, v našem případě, jak ukazuje obrázek 18, formou externí služby reCAPTCHA.

Administrátoři také, na rozdíl od zákazníků, nemají možnost požádat o zaslání nového hesla e-mailem. Ztráta hesla tak vyžaduje zásah některého ze správců aplikace.



Obrázek 18: Ochrana přihlašovacího procesu pomocí obrazového testu

5.6 Administrace

Při návrhu administrační části aplikace bylo sledováno několik cílů.

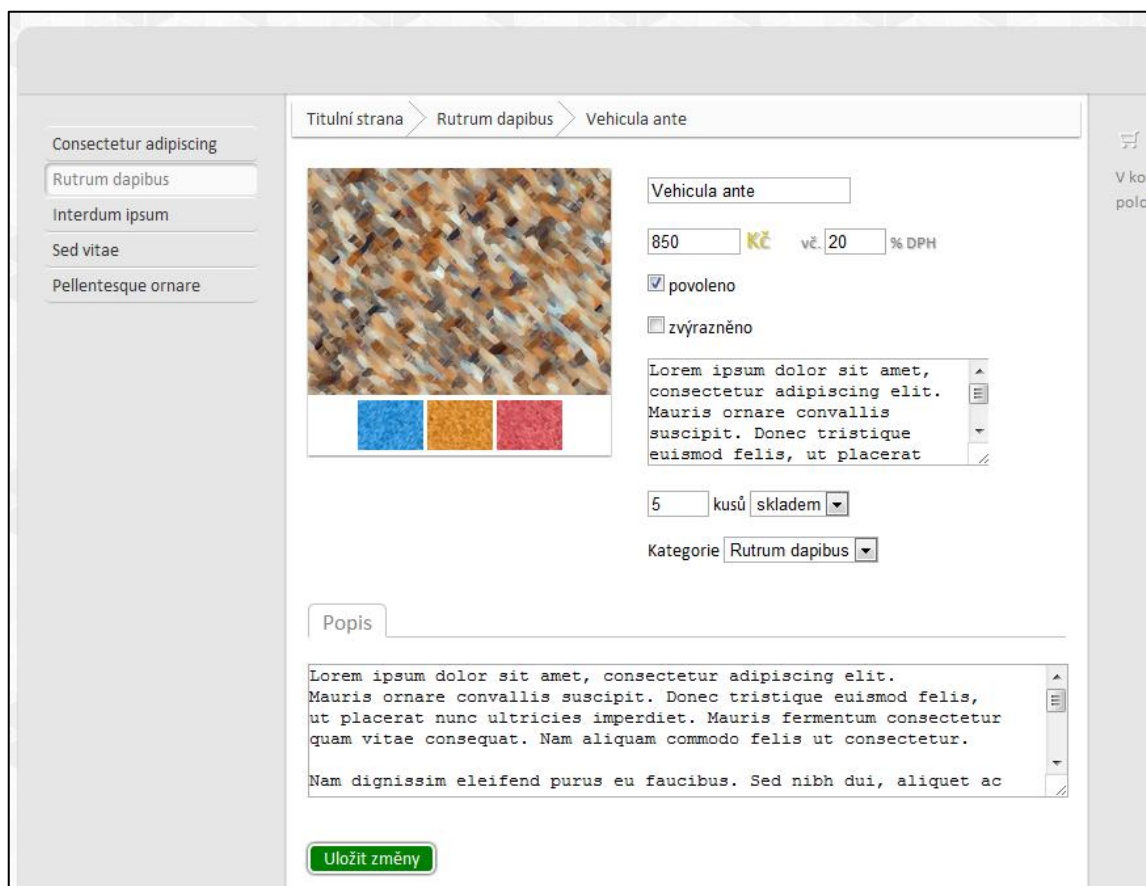
- Administrace je jednoduchá, nevyžaduje žádné odborné znalosti a probíhá v prostředí prohlížeče.
- Orientace v administraci je bezproblémová. Administrační rozhraní má stejnou grafickou podobu a strukturu jako veřejná část webu. Je tak částečně uplatňován princip *WYSIWYG*, kdy je vrstva pro úpravy vizuálně shodná s výslednou obsahovou reprezentací.

5.6.1 Úprava obsahu obchodu

Obrázek 19 ukazuje aktivní vrstvu úpravy obsahu detailu nabízené položky. Je využit princip *WYSIWYG*, pro administrátora je tak velmi jednoduché na první pohled

rozlišit, kterou hodnotu právě upravuje a není třeba žádného dalšího popisu nebo vysvětlení.

Po skončení úprav administrátor potvrdí uložení dat do databáze.



Obrázek 19: Vrstva pro úpravu obsahu detailu položky

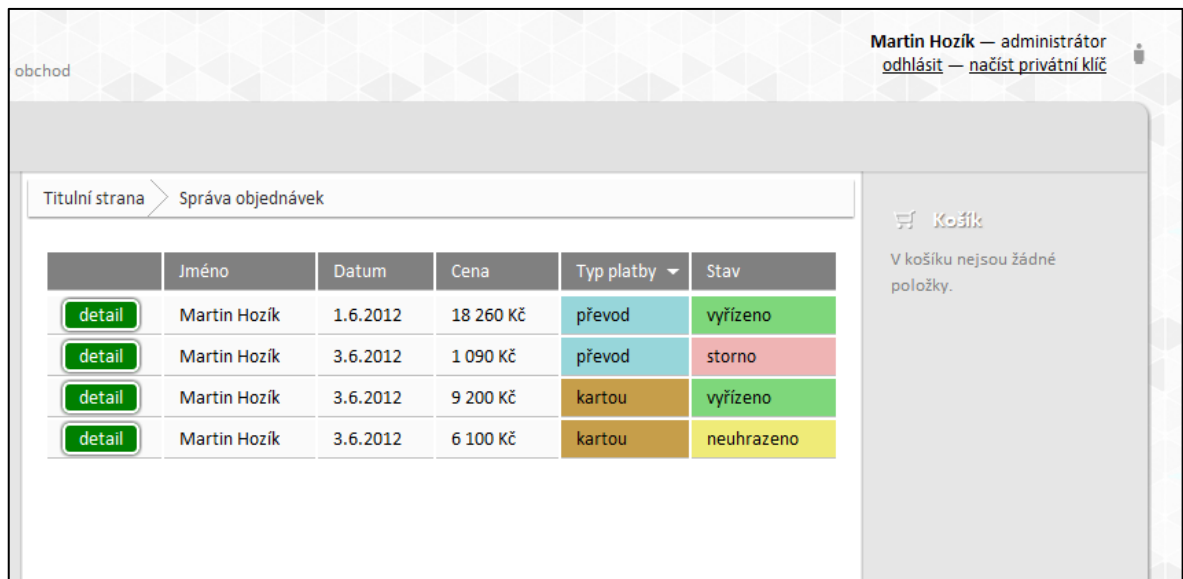
5.6.2 Správa objednávek

Důležitou činností správců aplikace je agenda objednávek. Obrázek 20 ukazuje detail tabulky objednávek přihlášeného správce.

Pro pohodlnou správu objednávek jsou implementovány některé důležité funkcionality.

- Seznam objednávek je možné řadit podle různých kritérií – typicky je to její stáří a stav vyřízení.
- Administrátor může číst a upravovat parametry objednávky – především stav vyřízení.

- Je možné také připojit interní procesní informace.



Obrázek 20: Správa objednávek obchodu přihlášeným administrátorem

5.6.3 Dešifrování údajů o platební kartě

V případě, že zákazník zvolil platbu předáním údajů o své platební kartě obchodníkovi, jsou tyto zašifrovány a uloženy v databázi.

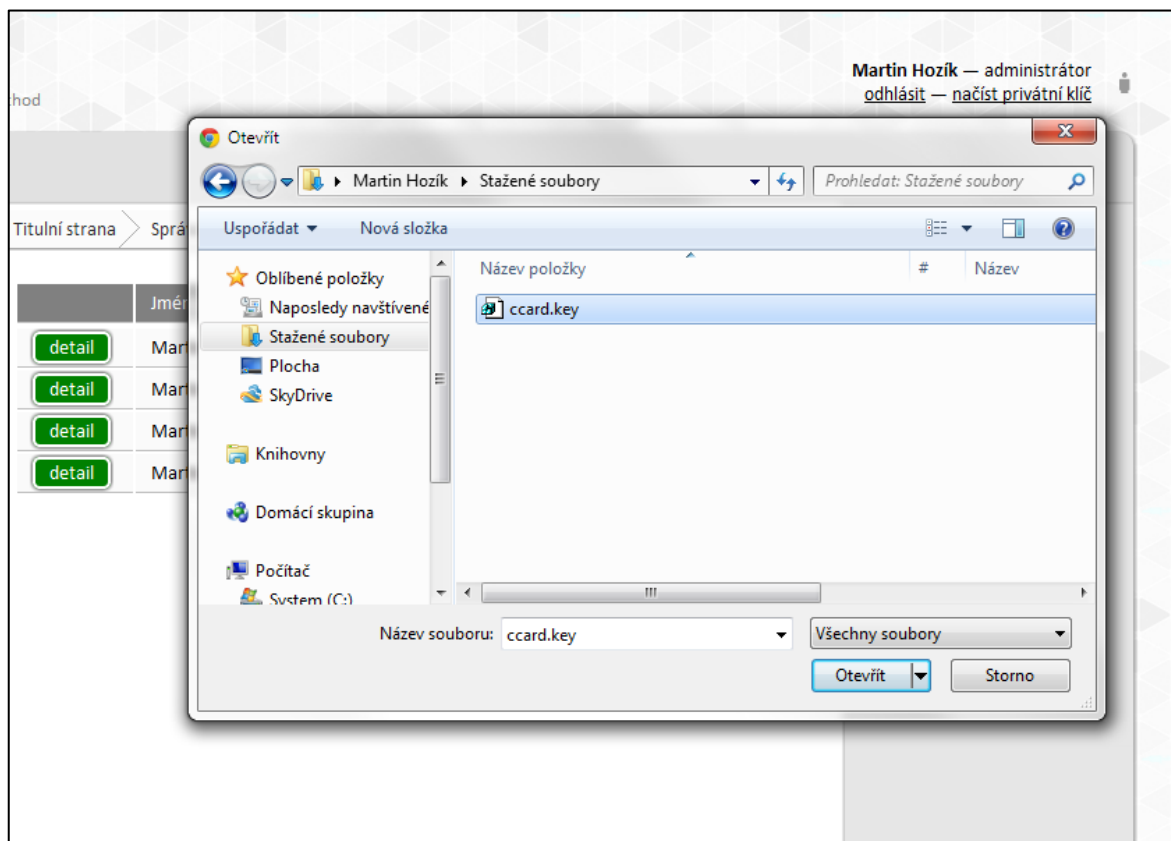
Přístup k těmto datům je pak omezen další úrovní zabezpečení nad běžnou autentizací administrátora.

Pro dešifrování privátních dat je vyžadován privátní dešifrovací klíč, který vlastní pouze správce pověřený k manipulaci s kartami zákazníků. Pro dešifrování je využit objekt *Flash*, do kterého administrátor vloží soubor standardním systémovým dialogem, viz obrázek 21, se svým privátním klíčem a modul pak převede šifrovaná data do čitelné podoby.

Aplikace *Flash* je zde využita, protože poskytuje spolehlivější metody interakce se systémem souborů klienta a má také širší možnosti práce se surovými binárními daty.

Vzhledem k tomu, že návrh aplikace pochopitelně není schopen ovlivnit bezpečnost dat po dešifrování na straně klienta, je vhodné, aby provozovatel přijal některá další opatření na zvýšení bezpečnosti.

- Privátní klíč je vhodné umístit na externí zabezpečené médium.
- Počítač oprávněného administrátora by měl splňovat zpřísněná kritéria ohledně přítomnosti škodlivého nebo nevyžádaného softwaru.



Obrázek 21: Vložení privátního klíče do dešifrovacího modulu administrace

6 ZÁKLADNÍ DRUHY BEZHOTOVOSTNÍCH ON-LINE PLATEB

Využití debetních a kreditních karet pro úhradu elektronických objednávek podstatně zjednodušuje a zrychluje celý nákupní proces.

Pro samotného zákazníka je tento způsob platby výhodný i z hlediska toho, že náklady na transakci typicky hradí příjemce.

Z hlediska bezpečnosti ale s sebou nese využití platebních karet některá bezpečnostní rizika, která vplynula z aplikace tohoto platebního modelu z reálného světa do světa internetového.

- Kartu jako takovou reprezentují jen identifikační a kontrolní čísla.
- Výši čerpání finančních prostředků klient typicky nemůže ovlivnit.

Vzhledem k tomu, že při platbě kartou, v případě elektronického obchodu, nestojíme obchodníkovi tváří v tvář, klade tento model poměrně nadstandardní předpoklady na jeho důvěryhodnost a technickou integritu.

Za těchto okolností se tedy prosadily některé varianty plateb využívajících *důvěryhodného zprostředkovatele*.

6.1 E-Commerce 3D-Secure

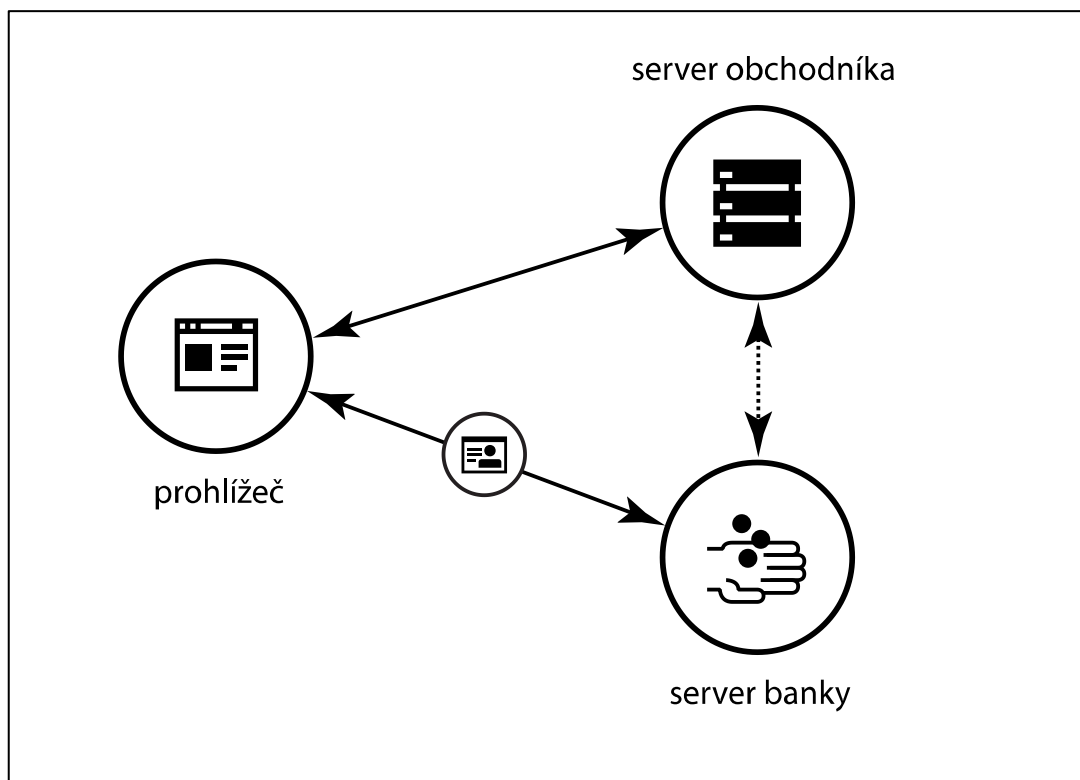
3D-Secure je jedním z nejpoužívanějších modelů zprostředkované platby kartou.

Do standardního procesu platby přidává subjekt, který zákazníci mohou považovat za důvěryhodný. Tuto roli typicky plní zpracovatelská banka, která spravuje obchodníkův bankovní účet pro příjem plateb. (10)

Název 3D v tomto případě značí trojici *bezpečnostních domén*, které spolu komunikují způsobem, který zjednodušeně znázorňuje obrázek 22.

1. Obchodníkův server stanoví detaily platby a přesměruje zákazníkův prohlížeč na terminál zpracovatelské banky.
2. Terminál ověří dotazem na obchodníkův server správnost platebních údajů zákazníka.
3. Údaje o platební kartě jsou zákazníkem předány zpracovatelské bance. Obchodník s nimi nepříjde do styku.
4. Obchodníkův server je informován, že platba proběhla.

5. Zákazník je přesměrován zpět na server obchodníka.



Obrázek 22: Zjednodušené schéma modelu 3D-Secure

Vzhledem k tomu, že obchodník v tomto případě nepracuje s žádnými privátními daty, není nutné používat zabezpečené spojení obchodníka se zákazníkem.

Spojení zákazníka i obchodníka s terminálem banky už pochopitelně zabezpečeno je.

6.2 PayPal Express Checkout

Variací na podobné téma je *Express Checkout* od mezinárodní webové platební společnosti *PayPal*.

PayPal volí mírně odlišný model autorizace a ověřování platebního procesu. (11)

1. Obchodníkuv server stanoví detaily platby a předá je serveru PayPal, který vrátí zpět unikátní identifikátor transakce.
2. Zákazník je přesměrován na terminál PayPal, kam zadá údaje o své kartě nebo o svém PayPal účtu.
3. Zákazník je přesměrován zpět na web obchodníka.

4. Server obchodníka ověří dotazem na server PayPal, zda zákazník autorizoval platbu.
5. Server obchodníka vyzve server PayPal, aby platbu provedl – tomu ještě může předcházet potvrzení zákazníka na webu obchodu.

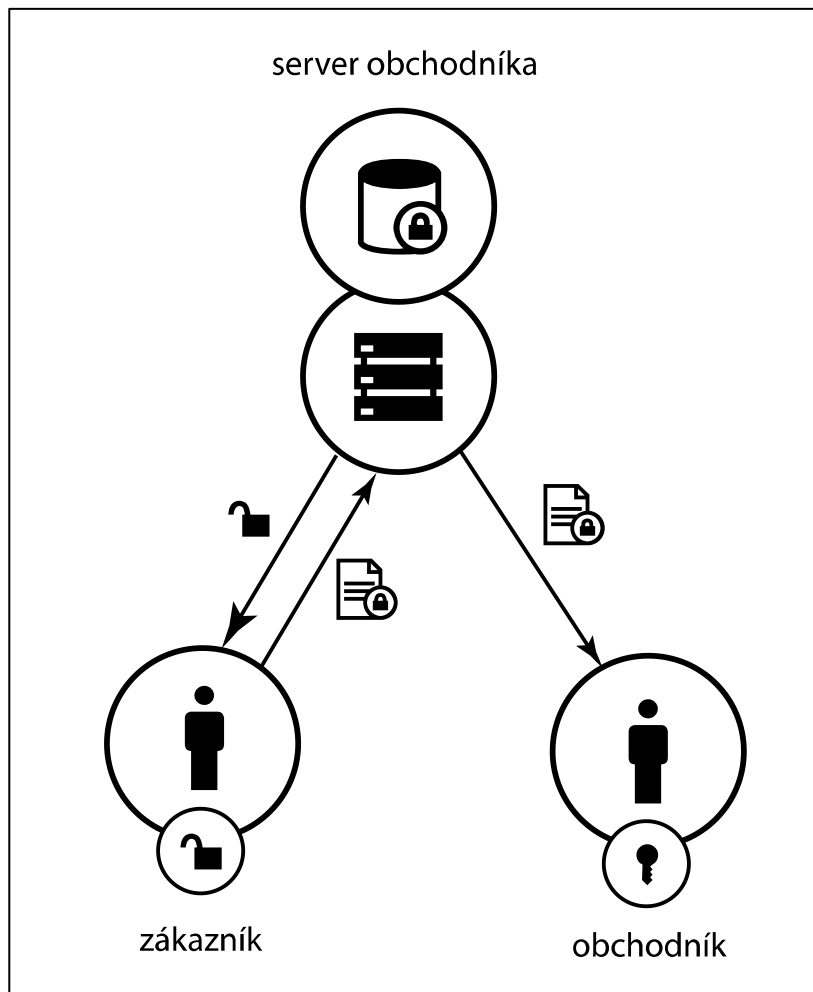
Drobnou výhodou je skutečnost, že, na rozdíl od modelu 3D-Secure, komunikuje obchodníkův server se serverem PayPal vždy z pozice klienta, což zjednodušuje nároky na jeho šifrovací výbavu.

6.3 Předání údajů o kartě obchodníkovi

Předchozí dva zprostředkované platební modely pochopitelně nenechávají žádný prostor pro chybné zpracování privátních platebních údajů – obchodník k nim v žádném z kroků nemá přístup.

Stejně tak i samotná implementace je triviální, využívající hotových skriptů a striktně definovaných komunikačních rozhraní.

V praktické části této práce se proto zaměříme na řešení, při kterém se předpokládá nezprostředkované předání údajů o kartě obchodníkovi. To může být výhodné nebo i nezbytné v případě, že je vyžadována i jiná operace, než obyčejná jednorázová platba – například složení kauce nebo zřízení dlouhodobých inkasních plateb.



Obrázek 23: Zabezpečené předání údajů o kartě obchodníkovi

Jak je zřetelně ukazuje obrázek 23, je pro přenos privátní údajů o platební kartě využita asymetrická kryptografie.

1. Zákazník získá ze serveru veřejný šifrovací klíč.
2. Zákazník zadá privátní údaje ke své platební kartě do formuláře.
3. Šifrovací modul pomocí veřejného šifrovacího klíče před odesláním zašifruje privátní údaje.
4. Šifrované údaje jsou uloženy v databázi na serveru.
5. Přihlášený administrátor získá šifrované údaje o kartě.
6. Načtením svého privátního šifrovacího klíče do dešifrovacího modulu nyní může zobrazit privátní údaje o platební kartě zákazníka.

Přestože tento model komunikace teoreticky nevyžaduje žádné další zabezpečené vrstvy, nijak neřeší problematiku ověřování integrity veřejných klíčů a je tak velmi náchylný pro napadení útokem typu *Man in the middle*. V praxi je tedy vhodné doplnit komunikaci o transparentní TLS/SSL zabezpečenou vrstvou.

ZÁVĚR

Hlavním cílem této bakalářské práce bylo na jednoduchém typu webového elektronického obchodu demonstrovat základní návrhové vzory a bezpečnostní řešení běžné moderní webové aplikace.

Teoretická část se soustředí na popis klíčových technologií a kryptografických metod, které jsou pro vývoj a provoz využity. Vysvětluje také typické bezpečnostní hrozby, které při vývoji a provozu aplikace mohou vzniknout a navrhuje účinná řešení pro jejich potlačení.

Praktická část popisuje návrh demonstračního elektronického obchodu, napsaného v jazyce PHP a sestaveného nad aplikačním rámcem Nette Framework. Ukazuje především vhodnou strukturu a relace datových objektů, se kterými aplikace pracuje.

V praktické části je také věnována pozornost běžným způsobům zajištění a zabezpečení elektronických plateb ve webové aplikaci. Kromě obvyklých řešení třetích stran je navržen vlastní způsob bezpečného přenosu platebních údajů od zákazníka k obchodníkovi, který si udržuje bezpečnostní integritu i v případě kompromitace obsahu serveru a dává tak teoretickou možnost využít tento model komunikace i u serverů s běžnou úrovní zabezpečení.

Přestože ukázková aplikace rozhodně nemá ambice pokrýt všechny případné požadavky reálného použití, byla navržena s ohledem na snadnou a bezpečnou rozšiřitelnost a může sloužit jako koncept skutečného elektronického obchodu.

ZÁVĚR V ANGLIČTINĚ

The main goal of this bachelor thesis was to demonstrate basic design patterns and security solutions of common modern web applications on a simple web e-commerce example.

The theoretical part describes key technologies and cryptographic methods for build and service of web applications. It also describes common security threats in private data processing and storing and shows proper solutions for their prevention.

The practical part demonstrates a PHP-based e-commerce web application as a case study and describes an appropriate scheme for data processing and storing.

It also focuses on common online payment solutions and describes a proprietary communication solution for secure payment detail transfer from customer to merchant even on a low-secured server.

Despite its design simplicity, this application can be used as an extensible and safe concept for real e-commerce web applications.

SEZNAM POUŽITÉ LITERATURY

1. **W3C Technical Architecture Group.** Architecture of the World Wide Web, Volume One. *World Wide Web Consortium*. [Online] 15. prosince 2004. [Citace: 18. května 2012.] <http://www.w3.org/TR/2004/REC-webarch-20041215/>.
2. **The Internet Society.** Hypertext Transfer Protocol -- HTTP/1.1. *The Internet Engineering Task Force*. [Online] Červen 1999. [Citace: 15. května 2012.] <http://tools.ietf.org/html/rfc2616>. RFC 2616.
3. **HTML Working Group.** HTML 4.01 Specification. *World Wide Web Consortium*. [Online] 24. prosince 1999. [Citace: 12. května 2012.] <http://www.w3.org/TR/html4/>.
4. **CSS Working Group.** Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. *World Wide Web Consortium*. [Online] 7. června 2011. [Citace: 12. května 2012.] <http://www.w3.org/TR/CSS2/>.
5. **XML Core Working Group.** Extensible Markup Language (XML) 1.0 (Fifth Edition). *World Wide Web Consortium*. [Online] 26. ledna 2008. [Citace: 12. května 2012.] <http://www.w3.org/TR/xml/>.
6. **Nixon, Robin.** *Learning PHP, MySQL, and JavaScript*. Sebastopol, CA, USA : O'Reilly Media, 2009. ISBN 978-0-596-15713-5.
7. **Bernard, Borek.** *Adobe Flex - Kompletní průvodce tvorbou interaktivních aplikací*. Brno : Computer Press, 2011. ISBN 978-80-251-2765-0.
8. **Network Working Group.** The Transport Layer Security (TLS) Protocol. *The Internet Engineering Task Force*. [Online] Srpen 2008. [Citace: 25. května 2012.] <http://tools.ietf.org/html/rfc5246>.
9. **Schneier, Bruce.** *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C*. Minneapolis, MN, USA : John Wiley & Sons, 1996. ISBN 0-471-12845-7.
10. **Česká spořitelna, a. s.** Manuál pro obchodníky - Přijímání platebních karet prostřednictvím České spořitelny, a.s. *Česká spořitelna*. [Online] Prosinec 2005. [Citace: 1. června 2012.] <http://www.csas.cz/banka/content/inet/internet/cs/ManualObchodnici.zip>.

11. **PayPal, Inc.** PayPal Express Checkout Integration Guide. *PayPal*. [Online] Duben 2012. [Citace: 1. června 2012.]

https://cms.paypal.com/cms_content/US/en_US/files/developer/PP_ExpressCheckout_IntegrationGuide.pdf.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AES	Advanced Encryption Standard
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
DBMS	Database management system
DES	Data Encryption Standard
DNS	Domain Name System
DOM	Document Object Model
EER	Enhanced Entity-Relationship
FTP	File Transfer Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
NPAPI	Netscape Plugin Application Programming Interface
OSI	Open Systems Interconnection
PHP	PHP: Hypertext Preprocessor
SGML	Standard Generalized Markup Language
SHA	Secure Hash Algorithm
SQL	Structured Query Language
TCP	Transmission Control Protocol
TLS/SSL	Transport Layer Security/Secure Sockets Layer
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WYSIWYG	What you see is what you get

WWW	World Wide Web
XML	Extensible Markup Language
XSRF	Cross-site request forgery
MD5	Message-Digest 5

SEZNAM OBRÁZKŮ

Obrázek 1: Model interpretované komunikace klienta a serveru.....	13
Obrázek 2: Zjednodušené schéma předání symetrického šifrovacího klíče	16
Obrázek 3: Symetrické a asymetrické šifrování dat úložiště	21
Obrázek 4: Srovnání vlivu délky výrazu a velikosti množiny prvků na sílu hesla	23
Obrázek 5: Zneužití autentizované relace požadavkem z nelegitimního zdroje	24
Obrázek 6: Ukázka projektu <i>reCAPTCHA</i> (zdroj recaptcha.net)	25
Obrázek 7: EER diagram databázového úložiště	30
Obrázek 8: Produktová část diagramu úložiště.....	31
Obrázek 9: Objednávková část diagramu úložiště	32
Obrázek 10: Uložení údajů o platební kartě	33
Obrázek 11: Tabulky zákazníků a administrátorů	34
Obrázek 12: Kostra HTML rozhraní.....	35
Obrázek 13: Uživatelské rozhraní seznamu nabízených položek	36
Obrázek 14: Uživatelské rozhraní detailu položky	37
Obrázek 15: Uživatelské rozhraní správy košíku	38
Obrázek 16: Cenová rekapitulace objednávky	39
Obrázek 17: Formulář pro vložení údajů o platební kartě.....	40
Obrázek 18: Ochrana přihlašovacího procesu pomocí obrazového testu.....	41
Obrázek 19: Vrstva pro úpravu obsahu detailu položky	42
Obrázek 20: Správa objednávek obchodu přihlášeným administrátorem	43
Obrázek 21: Vložení privátního klíče do dešifrovacího modulu administrace	44
Obrázek 22: Zjednodušené schéma modelu 3D-Secure.....	46
Obrázek 23: Zabezpečené předání údajů o kartě obchodníkovi.....	48

Při tvorbě schémat byla využita knihovna symbolů Picol (picol.org).