UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Platební brána

Tomáš Reinert

Bakalářská práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2012/2013**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Tomáš Reinert**
Osobní číslo:	**I09252**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Platební brána**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

V teoretické části bakalářské práce budou představeny technologie sloužící k tvorbě webových aplikací. Dále budou představeny a zhodnoceny současné platebních systémů na trhu (PayPal, PaySec, ?). Důraz rešerše bude kladen na rozebrání bezpečnostních rizik spojených s prováděním platebních transakcí v otevřeném prostředí internetu. V aplikační části práce bude vytvořena objektová knihovna usnadňující práci s více druhy komerčních platebních bran prostřednictvím co možná nejvyšší míry abstrakce konkrétních řešení. Ověření funkčnosti knihovny bude provedeno nasazením pro potřeby konkrétní aplikace.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

Gilmore, Jason. Velká kniha PHP a MySQL 5. Praha : Zoner Prass, 2007. ISBN: 80-86815-53-6.
Schafer, Steven. HTML, XHTML a CSS. Praha : Grada, 2009. ISBN: 978-80-247-2850-6.

Vedoucí bakalářské práce: **Ing. Lukáš Čegan, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **21. prosince 2012**
Termín odevzdání bakalářské práce: **10. května 2013**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 29. března 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10. 05. 2013 Tomáš Reinert

Poděkování

Děkuji vedoucímu bakalářské práce panu Ing. Lukáši Čeganovi, Ph.D. za cenné rady a připomínky, které mi poskytl při tvorbě bakalářské práce.

Dále bych chtěl poděkovat své rodině za podporu poskytnutou během studia a paní MUDr. Gabriele Puškárové a ostatním zaměstnancům Thomayerovi nemocnice.

Anotace

Bakalářská práce popisuje návrh a implementaci objektové knihovny umožňující práci s více druhy komerčních platebních bran. Knihovna je použita pro aplikaci univerzálního platebního tlačítka, které řeší získávání financí pro neziskové organizace.

Teoretická část popisuje technologie sloužící k tvorbě webových aplikací. Dále se zabývá bezpečností elektronických platebních systémů. Nakonec představuje a zhodnocuje současné platební systémy na českém trhu.

Klíčová slova

platební brána, platební systém, transakce, objektová knihovna, PHP

Title

Payment Gateway

Annotation

The Bachelor’s thesis describes the design and the implementation of an object library which enables to work with more types of commercial payment gateways. The library is used for the application of a universal payment button which solves fundraising for non-profit-making organizations.

The theoretical part describes technologies used for creation of web applications. Further it deals with the security of the electronic payment systems. Finally it presents and assesses the current payment systems in the Czech market.

Keywords

payment gateway, payment system, transaction, object library, PHP

Obsah

Seznam zkratek

ISO	International Organization for Standardisation
WWW	World Wide Web
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IIS	Internet Information Services
HTML	HyperText Markup Language
XML	Extensible Markup Language
XHTML	Extensible HyperText Markup Language
CSS	Cascading Style Sheets
W3C	World Wide Web Consortium
PHP	Hypertext Preprocessor
MySQL	My Structured Query Language
EPS	Elektronické platební systémy
DNS	Domain Name Systém
AES	Advanced Encryption Standard
VPN	Virtual Private Network
SET	Secure Electronic Transaction
TCP	Transmission Control Protocol
IP	Internet Protocol
APEK	Asociace pro elektronickou komerci
UML	Unified Modeling Language
API	Application Programming Interface
SOAP	Simple Object Access Protocol

Seznam obrázků

Seznam tabulek

Seznam grafů

Úvod

S rozvojem nových technologií, zajišťující bezpečnou a rychlou komunikaci, se internet stal zajímavější pro obchodní využití. Rychlý rozvoj přilákal řadu finančních i komerčních společností, které začaly nabízet své služby obchodníkům.

Bakalářská práce je zaměřena na elektronické platební systémy a jejich využití. Platební systémy poskytují elegantní řešení on-line plateb za zboží a služby v prostředí internetu. Na rozdíl od plateb pomocí bankovních převodů, trvá jejich provedení několik sekund, a proto jsou stále více mezi zákazníky i obchodníky oblíbenější.

V teoretické části bakalářské práce budou představeny technologie sloužící k tvorbě webových aplikací. Čtenáře by měla uvést do základní problematiky návrhu moderních webových dokumentů. Dále bude popsán úvod do elektronických platebních systémů, zaměřený na jejich bezpečnost. Na závěr kapitoly budou popsány a zhodnoceny současné komerční platební systémy na českém trhu.

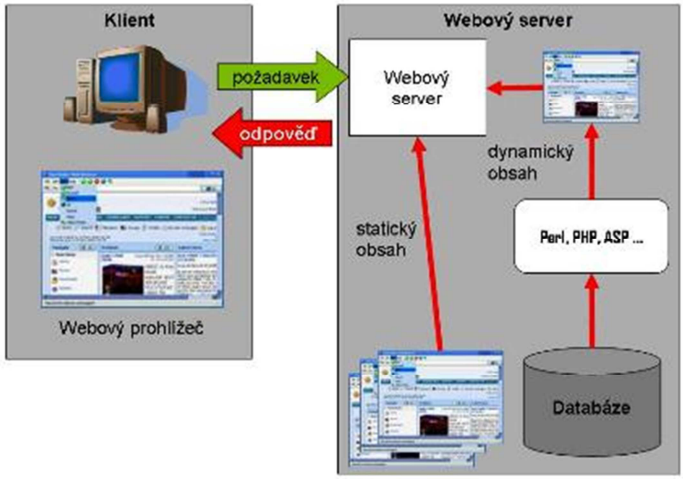
Praktická část se bude zabývat návrhem objektové knihovny zahrnující podporu několika komerčních platebních systémů. Řešení knihovny bude implementováno na aplikaci představující univerzální darovací tlačítko, které bude určeno pro neziskové organizace. Aplikace bude poskytovat řešení získávání financí pro jejich aktivity.

1 Webové technologie a elektronické platební systémy

V kapitole je nastíněna základní komunikace v prostředí Word Wide Web a přehled jednotlivých technologií pro přenos a tvorbu webových dokumentů. Lehký úvod do elektronických platebních systémů zaměřený především na jejich zabezpečení. Na závěr kapitoly jsou představeny a porovnány vybrané platební systémy vyskytující se na českém trhu.

1.1 WWW

WWW je nejrozšířenější internetová služba pro vyhledávání, prohlížení a sdílení webových dokumentů. Komunikace probíhá mezi klientem a webovým server, kteří spolu komunikují podle definovaných pravidel označovaných jako protokoly. Pro přenos dat se využívá HTTP protokolu, kdy klient pomocí webového prohlížeče pošle požadavek na server. Odpověď webového serveru je požadovaný dokument. [1], [2]



Obrázek 1 – Komunikace mezi klientem a serverem [2]

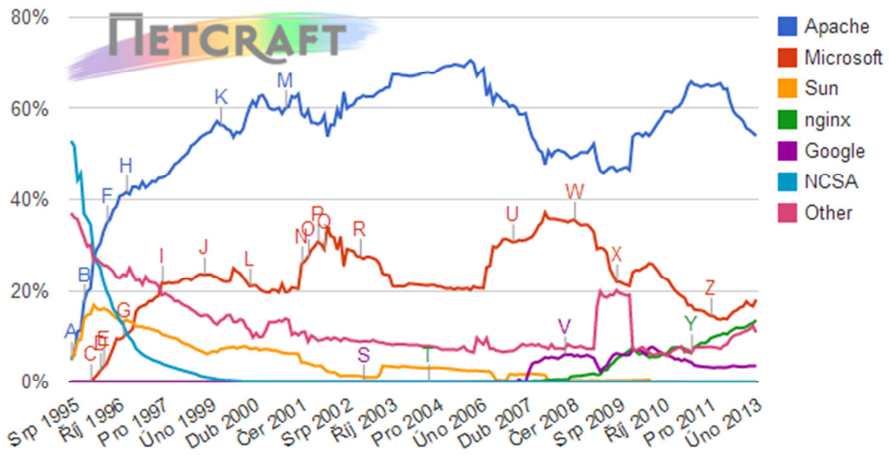
HTTP

HTTP je protokol aplikační vrstvy OSI/ISO modelu. Funguje na způsobu požadavek a odpověď. Teoreticky dokáže přenášet data jakéhokoliv druhu. Protokol HTTP je bezstavový (nedokáže navázat dlouhodobou relaci), to znamená, že klient pošle v krátkém časovém sledu více požadavků, ale server nerozezná, že jde o stejného klienta. Přenášená data nejsou nijak zabezpečena. O to se stará jeho nadstavba HTTPS. Pracuje implicitně na portu 80. HTTP 1.1 je zatím poslední verzí protokolu. [2]

Apache HTTP server

Apache HTTP server je v současnosti nejrozšířenější softwarový webový server (Obrázek 2). Jeho největší výhoda spočívá v otevřeném zdrojovém kódu podporující mnoho operačních systémů: např. unixové systémy, Microsoft Windows, Solaris a další. Apache podporuje velké množství funkcí, které jsou implementovány jako moduly rozšiřující jádro. Kon-

figurace se provádí pomocí textových souborů, a proto je velice jednoduché dynamicky přidávat nebo odebírat jednotlivé moduly. Výsledná funkčnost Apache je dána moduly, které jsou v okamžiku spuštění připojeny. Nejnovější verze Apache byla vydána v únoru roku 2013. [3]



Obrázek 2 – Využití webových serverů na trhu [4]

Internet Information Services

Softwarový webový server od společnosti Microsoft pro operační systém Windows. Dnes je druhým nejpoužívanějším webovým serverem. Na rozdíl od Apache je zpoplatněn a je součástí produktové řady Windows Server. První verze IIS obsahovaly mnoho chyb. Teprve od verze IIS 6.0 je webový server považován za bezpečný. IIS má od verze 7.0 (stejně jako Apache) modulární strukturu. Využívá modulů rozšiřující jádro systému, které přidávají nové funkce. Administrátor jednoduše konfiguruje funkčnost serveru bez nutnosti celý systém přeinstalovat. Nejnovější verze nese označení IIS 8.0 a je součástí systémů Windows Server 2012 a Windows 8. [5]

Nginx

Nginx (čteno „engin-x“) je softwarový webový server s otevřeným zdrojovým kódem. Jeho výhoda spočívá v rychlosti zpracování požadavků. Byl navržen tak, aby dokázal obsloužit 10 000 požadavků současně s minimální náročností na paměť. Nginx je běžně součástí linuxových distribucí, ale existuje i verze pro jiné operační systémy. Nginx poskytuje řadu modulů, které rozšiřují funkčnost serveru. Na rozdíl od Apache nelze moduly jednoduše přidávat nebo odebírat. Moduly musejí být zakompilovány v distribuci operačního systému. Jinak je nutné moduly překompilovat ručně, což není snadný proces. [6], [7]

1.2 Jazyky pro tvorbu webových aplikací

Webový dokument má svoji statickou část zobrazenou pomocí značkovacích jazyků a dynamickou část, u kterých využíváme jazyky skriptovací. Celý vzhled webového dokumentu potom obstarává formátovací jazyk. [8]

1.2.1 Značkovací jazyky

Značkovací jazyk je systematizovaná a standardizovaná sada značkovacích instrukcí. Při jejich vytváření musí být splněno několik požadavků:

- instrukce se řídí přísnými pravidly,
- instrukce je obsažena v textovém dokumentu,
- instrukce je pro koncového uživatele neviditelná,
- instrukce ukazuje zobrazovacímu zařízení (např. webovému prohlížeči), kde má začít, kde skončit a jaký formát bude použit. [8]

HTML

HTML je hypertextový značkovací jazyk používaný při tvorbě webových dokumentů. V roce 1990 byla vyvinuta první verze HTML spolu s protokolem HTTP a prvním webovým prohlížečem. S rychlým rozvojem webu bylo nutné stanovit standarty pro HTML, které vydává mezinárodní společenství W3C. Verze HMTL 4.01 měla být původně poslední a postupně se plánovalo přejít na XHTML. Kvůli nespokojenosti některých společností s XHTML byla v roce 2007 založena nová pracovní skupina, jejímž cílem je vývoj nové verze označené jako HTML 5, na které se v současnosti stále ještě pracuje. Již dnes webové prohlížeče podporují mnoho rozšíření HTML 5. [8], [9]

XML

EXtensible Markup Language (XML) je značkovací jazyk podobný HTML. Standart XML byl vyvinut společenstvím W3C. Slouží k vytváření dokumentů a výměně dat mezi aplikacemi, u kterých popisuje strukturu dat - nezabývá se jejich vzhledem. Z tohoto důvodu nemůže být XML označováno jako náhrada za HTML. XML nemá pevně stanovené atributy a elementy jako HTML. Uživatel tak vytváří svůj vlastní značkovací jazyk. Syntaxe XML je daleko přísnější než HTML. XML existuje ve dvou verzích a to XML 1.0 a XML 1.1. [8], [10]

XHTML

Původně mělo jít o nástupce staršího jazyka HTML vyhovující podmínkám tvorby XML dokumentů. Při tvorbě webových dokumentů se vývojáři dopouští řady chyb, které ovšem jazyk HTML toleruje. XHTML byl vyvinut s cílem stanovení přísných pravidel a minimalizováním chyb, které uživatel ve svém prohlížeči nevidí, ale z programátorského hlediska jsou špatně. Například všechny atributy a elementy musejí být psány malými písmeny,

protože XHTML rozlišuje velká a malá písmena. XHTML představuje spíše zúženou verzi HTML než jeho nástupce. [8], [11]

Verze XHTML:

- **XHTML 1.0 Strict** - Čistý strukturovaný dokument bez formátovacích značek, využívá se společně s CSS.
- **XHTML 1.0 Transitional** - Povoluje formátování textu a odkazů pomocí atributů v elementu body.
- **XHTML 1.0 Frameset** - Umožňuje používat zastaralé značky a podporuje rámce.

1.2.2 CSS

Organizace W3C navrhla formátovací jazyk CSS, který se používá pro popis vzhledu a formátování dokumentu u značkovacích jazyků HTML, XHTML a XML. Styly určují, jakým způsobem budou jednotlivé prvky dokumentu naformátovány. Hlavní smysl spočívá v tom, že autor textu se nezabývá jeho formátem, ale o to se postará člověk, který definuje styly. Použitím stylů lze na jednom místě změnit definici a změna se projeví u všech prvků, které daný styl používají. Největší nevýhodou CSS je nedostatečná podpora ze strany webových prohlížečů. Nejnovější verze CSS 3 je vyvíjena přímo na míru HTML 5 tak, aby dokázala využít jeho potenciálu.

Kaskádové styly se aplikují na HTML dokumenty třemi způsoby:

- **Přímý styl** – Pravidlo se vztahuje pouze na konkrétní tag.
- **Stylopis v hlavičce HTML** – Styl uzavřen mezi tagy *<style></style>*.
- **Externí soubor –** Připojení externího souboru s koncovkou „.css“. [8], [12]

1.2.3 Skriptovací jazyky

Skript je krátký program napsaný ve skriptovacím jazyce. Skripty se rozdělují na dvě základní skupiny podle toho, kde jsou skripty spouštěny, a to na klientské a serverové. Klientské skripty nijak nezatěžují server a jsou spouštěny klientským softwarem. Vkládají se do HTML dokumentu, odkud se dostávají ke klientovi, který musí podporovat použitý skriptovací jazyk. Serverové skripty jsou spouštěny na straně webového serveru a nijak nezatěžují klienta. Ten nemusí ani zaregistrovat spuštění skriptu. [8]

JavaScript

JavaScript je objektově orientovaný skriptovací jazyk, kdy skript je spouštěn na straně klienta. Není navržen jako samostatný jazyk, a proto je nejčastěji vkládán do HTML dokumentů. JavaScript poskytuje řešení, jak zinteraktivit jednotlivé prvky webové stránky. Používá se k manipulaci s obrázky, k ověření správnosti údajů vyplněných uživatelem ve formuláři nebo vytváření animace. Jeho syntaxe se podobá jazyku Java, ale jinak s ním nemá nic společného. [8]

PHP

Nejrozšířenější skriptovací jazyk k tvorbě dynamických webových dokumentů. Skripty jsou spouštěny na straně serveru. PHP je open-source projekt, kdy uživatelé nejsou omezeni licenčními podmínkami jako u komerčních produktů. Cílem vývoje nebylo navrhnout nový programovací jazyk, ale poskytnou takové řešení k tvorbě aplikací, u kterých uživateli stačí minimum znalostí. Poslední verze PHP 5 byl zlomový předěl v evoluci jazyka. Přidala podporu objektově orientovaného programování, obsluhu výjimek, vylepšenou podporu XML a webových služeb. PHP je rozšířen pro svoji praktičnost a v současné době mají vývojáři k dispozici přes 200 knihoven, které dohromady obsahují přes 1000 funkcí. [13]

1.3 MySQL

MySQL je relační databázový server. První verze byla vydána v roce 1996. Během svého vývoje se stalo velice populárním. Od roku 2009 spadá pod společnost Oracle Corporation. Největší výhoda MySQL je v podpoře celé řady operačním systémů. Poskytuje API pro mnoho programovacích jazyků, např. Java, C, C++ a další. MySQL nabízí svůj software zcela zdarma, ale za podmínek licence GPL (General Public License). Už od prvního vydání klade MySQL důraz především na výkon. Dříve byla nejvíce vytýkána MySQL absence pokročilých schopností. S vývojem MySQL jsou postupně funkce doplňovány a může se tak porovnávat se systémy na úrovni velkých korporací. [13]

1.4 Elektronické platební systémy

EPS je souhrn prostředků a metod zabezpečující převod financí digitálním formou. Laicky řečeno, cílem je přenos finanční částky mezi různými účastníky transakce. Některé EPS jsou elektronickou verzí založených na existujících platebních systémech, jako jsou šeky a kreditní karty nebo jsou založeny na nových technologiích jako např. digitální hotovost. V dnešní době zažívají svůj rozmach, protože poskytují:

- rychlost převodu financí,
- snadná ovladatelnost a srozumitelnost pro uživatele,
- zabezpečení soukromých informací,
- nízké poplatky za finanční transakce. [14]

1.4.1 Rozdělení

Existuje mnoho způsobů dělení elektronických plateb, které se vždy vztahují k námi zvolenému kritériu.

Hlavní rozdělení EPS je:

- elektronické platební systémy **s elektronickými penězi**,
- elektronické platební systémy **bez elektronických peněz**.

Rozdíl spočívá v tom, že platební nástroj (osobní počítač, čipová karta, magnetická karta) EPS **s elektronickými penězi** v sobě nese elektronickou hotovost. Oproti tomu systém **bez elektronických peněz** nenese elektronickou hotovost, ale ten rovnou pracuje s přímým elektronickým modelem mincí a bankovek.

Další rozdělení je podle časového kritéria mezi dobou, kdy entita, která zahájila platbu, považuje tento proces za ukončený a časem, kdy se provede transakce od plátce, rozdělujeme mezi:

- předplacené systémy (pre-paid payment systems),
- aktuálně placené systémy (pay-now payment systems),
- systémy, kdy se provádí platba později (pay-later payment systems).

Pre-paid systémy se označují jako hotovostní. Zákazník musí nejprve zakoupit kredit, který mu slouží k zaplacení nákupu za zboží nebo služby. Po zákazníkovi není vyžadováno mít založený účet v bance. Zákazník využívající systémy **pay-now** nebo **pay-later** musí mít zřízen účet v bance. Placení pomocí debetní karty představuje využití systému pay-now, zatím co u systémů pay-later se platí pomocí kreditní karty.

Během transakce se přenášejí libovolně velké částky financí a podle nich klasifikujeme elektronické platební systémy:

- mikroplatby,
- platby s **malou finanční částkou**,
- platby s **velkou finanční částkou**. [14], [15]

1.4.2 Zabezpečovací mechanismy

Existuje mnoho bezpečnostních problémů, které se týkají použitého hardwaru, softwarového vybavení a komunikačních protokolů. Důsledkem je vznik jednotných bezpečnostních standardů a norem, které mají za cíl udržet bezpečnost platebních systémů na vysoké úrovni. [16]

Základní požadavky

Elektronické platební systémy jsou velice komplexní a složité, a proto je při jejich vytváření nutné dodržovat určitá kritéria. Bezpečnostní požadavky se liší v závislosti na použitém systému. Obecně lze charakterizovat základní vlastnosti, které musí mít každý systém:

- integrita,
- utajenost,
- autorizace,
- dostupnost,
- spolehlivost.

Integrita a **autorizace** zabraňuje převodu peněz od uživatele, který tuto transakci nepotvrdil. Umožnuje odmítnutí přijetí platby bez souhlasu. Autorizace je nejdůležitějším požadavkem na platební systémy – jedná se o získání souhlasu s provedením nějaké operace. Může být prováděna třemi způsoby:

- autorizace třetí stranou,
- heslem,
- podpisem.

Autorizaci v mnoha případech předchází autentizace. Nemusí tomu ve všech případech být, protože některé strany při využívání elektronických plateb požadují nezveřejnění transakce. Informace o transakci (např. jméno plátce, příjemce, celková suma) zůstanou utajeny vůči třetí straně. Tento požadavek se nazývá **utajenost**.

Dostupnost a **spolehlivost** umožňuje platebním systémům provádět platby na libovolném místě. Platební transakce je atomická operace – tzn. transakce je provedena celá, nebo se neprovede žádná transakce. V žádném případě se nesmí stát, že bude přerušena a zůstane v nedokončeném stavu. Je kladen velký důraz na spolehlivost hardwaru a softwaru konkrétních systémů. [15]

Útoky z pohledu uživatele

Říká se, že každý řetěz je tak silný, jak je silný jeho nejslabší článek. Uživatel může být vnímán jako nejslabší článek celého systému, protože vlastní citlivé informace (např. pin kód, certifikát, heslo), které může útočníkovi nevědomě poskytnout.

Nejběžnější útoky při realizaci platebních transakcí:

- phishing,
- pharming,
- spyware.

Nejrozšířenější útok, při kterém útočník získá citlivé informace, je **phishing**. Nejčastěji uživatel obdrží e-mail, kde je vyzván k navštívení stránky např. své banky. Stránka je ovšem napodobeninou originálních stránky banky, kterou lze jen velmi těžko rozlišit. Uživatel v domnění, že komunikuje s důvěryhodnou organizací, poskytne své osobní informace.

Mnohem důmyslnějším útokem je **pharming**, který manipuluje s DNS záznamy. Uživatel je automaticky přesměrováván na útočníkovi vlastní stránky. Uživatel opět nevědomě poskytne útočníkovi osobní informace (např. formou přihlašovacího formuláře).

Spyware má za úkol sbírat informace o činnosti uživatele. Jeho spuštění uživatel nezaregistruje, v čemž spočívá jeho největší nebezpečí. Do počítače se nejčastěji dostane formou trojského koně. Spyware lze velmi jednoduše odstranit z počítače. Známý je útok na Ko-

merční Banku v roce 2006, kdy se podařilo hackerům pomocí trojského koně ukrást desítky přístupových certifikátů a hesel k elektronickému bankovnictví. [16]

Symetrické šifrování

Využívá jeden stejný klíč, který obě strany používají pro šifrování a dešifrování zpráv. Výhodou symetrického šifrování je nízká výpočetní náročnost. Na druhou stranu nevýhoda spočívá ve sdílení tajného klíče bezpečnou cestou, na kterém se musejí obě komunikující strany domluvit. Nejznámější symetrické šifrovací algoritmy jsou např. AES, RC2. [17]

Asymetrické šifrování

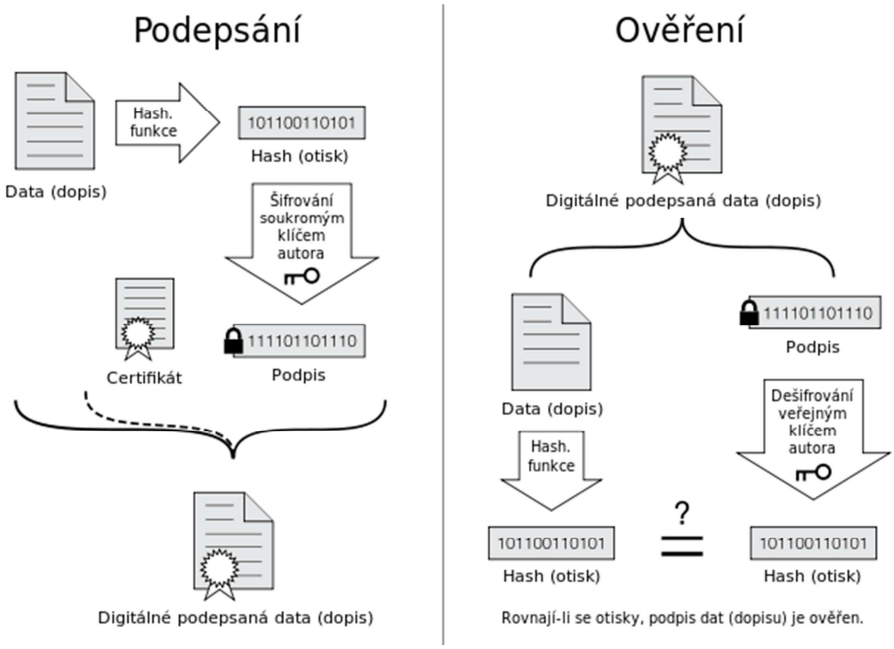
Obsahuje dva klíče – veřejný a soukromý. Pomocí veřejného klíče může kdokoliv zašifrovat zprávu a odeslat ji majiteli veřejného klíče, který zprávu může dešifrovat pouze jeho soukromým klíčem. Jedině majitel zná svůj soukromý klíč. Pro asymetrické šifrování je nejznámější algoritmus označený RSA. [18]

Hašovací funkce

Znemožňuje úmyslné poškození dat. Výstupem hašovací funkce je miniatura, otisk či hash (česky haš), což je posloupnost určité délky. Z vypočítaného haše nelze již dostat zpátky vstupní data. Hašovací funkcí je například MD5. [19]

Elektronický podpis

V informatice plní stejnou roli jako klasický vlastnoruční podpis. Elektronický podpis dokáže ověřit autenticitu, integritu, nepopiratelnost a může obsahovat časové razítko, které uchovává čas a datum podepsání dokumentu. [20]



Obrázek 3 – Elektronický podpis [20]

Digitální certifikát

Umožnuje identifikaci druhé strany, která chce navázat zabezpečené spojení (HTTPS, VPN). Certifikáty vydává certifikační autorita, a proto lze důvěřovat neznámým certifikátům, pod kterou je podepsána certifikační autorita. Každý vydaný certifikát má omezenou dobu platnosti a to z důvodu zamezení jejich zneužívání. Běžné certifikáty mají dobu platnosti jeden rok. [21]

Digitální obálka

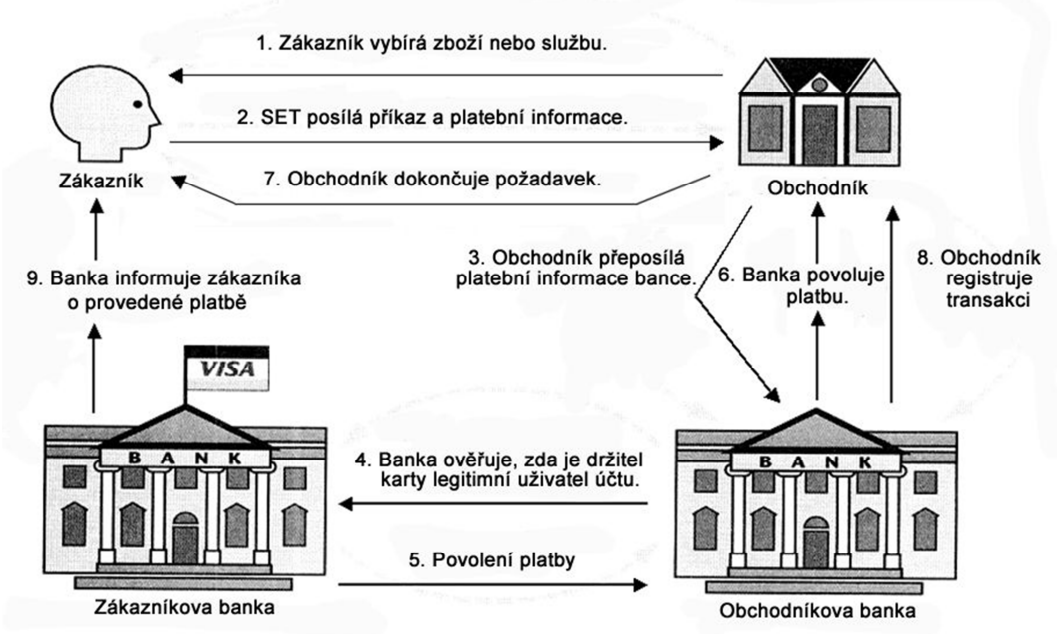
Zajišťuje bezpečný přenos symetrického klíče od vysílače k příjemci. Vygenerovaný symetrický klíč K je zašifrován veřejným klíčem příjemce Vk-p. Zašifrovaný symetrický klíč E(K)Vk-p je odeslán příjemci. Pouze příjemce ho může dešifrovat svým soukromým klíčem, tím dostane symetrický klíč, který používá v následné komunikaci. [22]

Secure Electronic Transaction

Secure Electronic Transaction je komplexní bezpečnostní protokol. Byl vyvinut společnostmi VISA a MasterCard k ochraně transakcí kreditních karet. Hlavním cílem bylo naplnit obchodní požadavky na platební transakce:

- důvěrnost (šifrování komunikace),
- integrita dat,
- autorizace (ověření účastníků platební transakce),
- soukromí (utajení informací transakce před třetí stranou).

Před provedením transakce, musí každý s účastníků vlastnit certifikát k prokázání své identity. Průběh transakce je popsán na obrázku 4.



Obrázek 4 – Průběh SET transakce [23]

Do SET protokolu byla implementována řada bezpečnostních mechanismů:

- symetrické šifrování,
- asymetrické šifrování,
- hašovací funkce,
- elektronický podpis,
- dvojí podpis (Dual Signature).

Významnou inovací v SET je použití **dvojího podpisu**. Účelem je zajistit integritu a utajenost dat. Metoda dvojího podpisu bezpečně spojuje dvě zprávy určené pro různé příjemce:

- **Informace o nákupu** (OI) – Poslané od zákazníka k obchodníkovi.
- **Platební informace** (PI) – Poslané od zákazníka k bance.

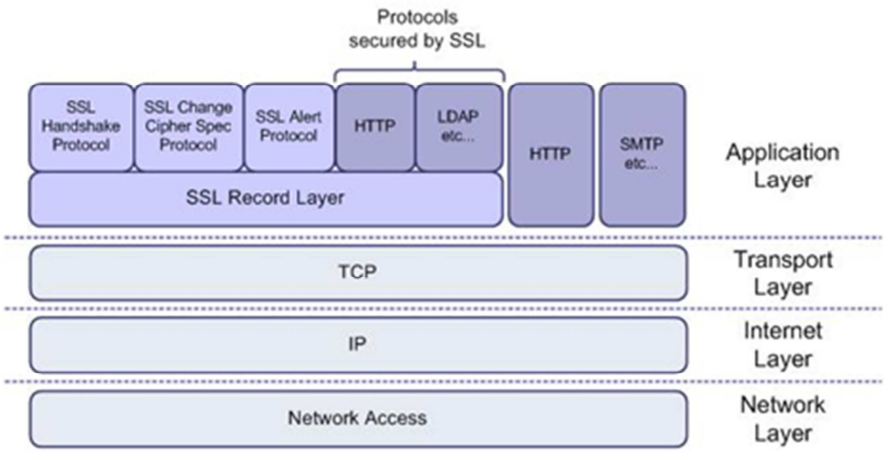
Cílem je poskytovat jen ty informace, které určitá strana vyžaduje k provedení transakce např. obchodník nepotřebuje znát číslo kreditní karty zákazníka, banka nepotřebuje znát podrobnosti o nákupu zákazníka. [23]

Secure Sockets Layer

SSL je komunikační protokol, který zabezpečuje komunikaci síťových služeb využívající TCP/IP protokol. Z pohledu TCP/IP se jedná o další vrstvu, která se vkládá mezi transportní vrstvu TCP a vrstvu aplikačního protokolu (Obrázek 5). Poskytuje tak bezpečné a ověřené spojení mezi dvěma body v síti. Nejčastěji se používá pro komunikaci HTTP serveru a klientské aplikace.

SSL protokol zajišťuje:

- šifrování dat,
- autentizaci serveru,
- autentizaci klienta,
- integritu dat.



Obrázek 5 – Umístění SSL v TCP/IP modelu [24]

Dnes se nejčastěji navazuje zabezpečená komunikace pomocí SSL protokolu při přihlašování do aplikace EPS přes webového klienta např. uživatel využívající internetové bankovnictví z domova, navazuje SSL spojení s bankovním serverem. [24]

3-D Secure

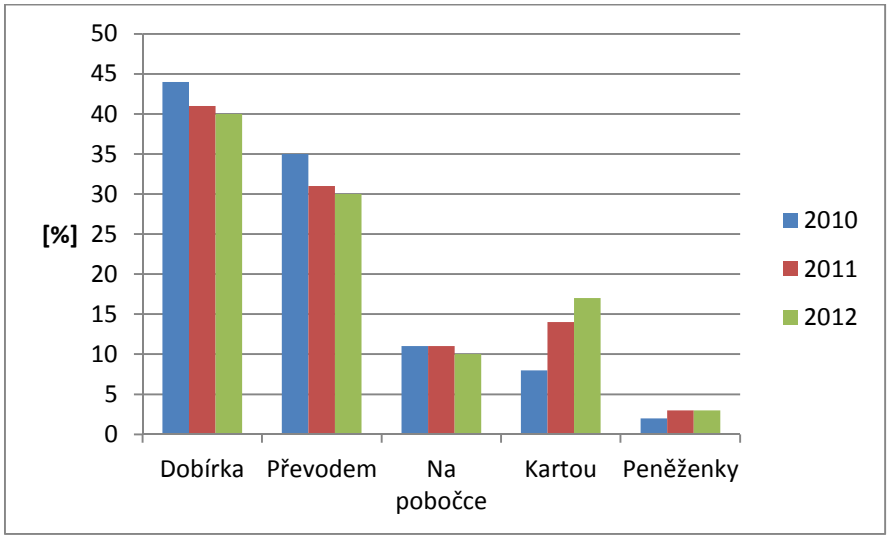
V roce 2001 představila společnost VISA a MasterCard nový autentizační standart pro zabezpečení on-line plateb. Společnosti představily 3-D Secure protokol. VISA označuje svůj systém Verified by VISA, naopak MasterCard používá název SecureCode. Oproti SET protokolu uživatelé pro ověření své identity používají uživatelské jméno a pin kód. 3-D Secure představuje tří doménový model, který odděluje odpovědnost různých stran v rámci transakce.

Domény zapojené během transakce:

- **Issuer domain** – Doména vydavatele banky zákazníka.
- **Acquirer domain** – Doména zpracovatelské banky obchodníka.
- **Interoperability domain** – Doména kartových asociací VISA, MasterCard. [25]

1.5 Platební metody na českém internetu

Podle tiskové zprávy Asociace pro elektronickou komerci (APEK), vydané v červnu 2012, je stále nejoblíbenější platební metodou na českém internetu dobírka neboli platba při osobním odběru. Dobírka nemá s EPS nic společného, ale její existenci nelze přehlížet, protože několik let po sobě je nejpoužívanější platební metodou. [26]



Graf 1 - Oblíbenost platebních metod [26]

Naneštěstí obchodníci začínají nabízet řadu dalších platebních možností, jejichž zavádění zvedá obchodníkům tržby. Do popředí se dostávají nové platební prostředky.

Platební metody na českém internetu:

- platební karta,
- platební brána,
- platební agregátor,
- na splátky,
- premium Rate SMS,
- m-platba,
- elektronická peněženka,
- převod na účet,
- dobírka,
- hotovost.

APEK očekává pozvolný přechod od hotovostních plateb k dalším bezhotovostním nástrojům během několika let. Podle odborníků by budoucnost mohla patřit platebním branám a platebním agregátorům. [27]

1.5.1 Platební brána

Platební brána představuje specifickou službu provázanou s platebním systémem. Jedná se o zjednodušenou formu zadávání platebního příkazu, který má již předvyplněné údaje. Obchodníkům dovoluje přijmout platbu od zákazníka, který má zřízen účet s internetovým bankovnictvím u banky provozující platební bránu.

Průběh transakce je jednoduchý:

1. Zákazník zvolí platbu pomocí platební brány a následně je přesměrován do prostředí platebního systému.
2. Zákazník je vyzván k vyplnění přihlašovacích údajů.
3. Platební příkaz je vyplněn údaji o platbě a zboží a zákazník jej pouze potvrdí.
4. Obchodník je okamžitě informován o provedení platby.

V České republice dnes poskytují platební brány téměř všechny banky. Česká spořitelna, mBank, Komerční banka, ČSOB, Raiffeisenbank a Poštovní spořitelna poskytují své platební brány. Fio banka, GE Money Bank a Volksbank využívají služby poskytované společností PayU. [27]

Tabulka 1 – Výhody a nevýhody platební brány [27]

Výhody	Nevýhody
Předvyplněný platební příkaz	Zákazník musí mít zřízen účet u banky poskytující konkrétní platební bránu
Zákazník je přesměrován do systému vlastní banky, který zná	Obchodník bez účtu nemůže využívat konkrétní platební bránu banky
Obchodník obdrží peníze ihned po provedení platby	Menší obchodníci nemohou nabídnout více než jednu platební bránu
Výše platební částky není nijak omezená	

1.5.2 Platební agregátor

Platební agregátor je služba, která sjednocuje více platebních metod. Je to logický krok v situaci, kdy na trhu existuje více platebních bran od různých bank. Zákazník tak není omezen tím, u které banky vlastní účet. Na českém trhu v současnosti existuje několik poskytovatelů služeb implementující více platebních metod např. PayU, GoPay a další. [27]

Tabulka 2 – Výhody a nevýhody platebního agregátoru [27]

Výhody	Nevýhody
Univerzálnost systému, který spojuje více platebních bran	Výše poplatků se liší od každého obchodníka
Široké portfolio finančních institucí	Malé rozšíření u českých obchodníků
Využití u velkých i malých obchodníků	
Bezproblémová reklamace	

PayU

Provozovatelem platebního systému PayU byla společnost Aukro s.r.o., ale od roku 2011 spadá pod mezinárodní skupinu PayU Group. PayU poskytuje více platebních metod v rámci jednoho systému.

V současnosti zprostředkovává platební metody:

- ePlatba (Raiffeisenbank),
- Mojeplatba (Komerční banka),
- mPeníze (mBank),
- on-line převod od GE Money Bank,
- on-line převod od Volksbank,
- on-line převod Fio banky,
- hotovostní platba na terminálech Sazka a pobočkách České pošty,
- platební karty VISA, MasterCard.

On-line platby jsou ze strany bank garantované, tzn. po ověření zůstatku na účtu klienta, banka ručí za okamžité odeslání peněz. [27]

1.5.3 Elektronická peněženka

Elektronická peněženka představuje virtuální účet nebo předplacenou kartu, kterou musí uživatel nejprve nabít penězi a následně potom může využívat k placení. U elektronických peněženek nedochází k zneužití platební karty, protože při transakci se využívá přihlašovací jméno a heslo do systému. Oproti vyspělejším zemím nejsou v ČR elektronické peněženky tolik rozšířeny. Světovou špičku platebních systému představuje PayPal. Z tuzemských systémů je nejznámější PaySec. [28]

1.5.4 Porovnání platebních systémů

Z důvodů využití a rozšíření mezi uživateli byli vybrány systémy PayPal, MoneyBookers, PaySec a GoPay.

PayPal

Systém PayPal je nejrozšířenější platební systém na světě. Dnes ho využívá řada internetových obchodníků k přijímání a posílání plateb přes internet. Nejznámější je aukční síň eBay, která PayPal vlastní a využívá ho jako primární platební prostředek. V minulých letech nebyl PayPal v ČR tolik rozšířen, ale vše se změnilo českou lokalizací eBay. Do budoucna je ze strany společnosti slíbena i česká lokalizace PayPal.

Předností PayPal je jednoduchá registrace. Při vytváření nového účtu není po uživateli požadováno zadání čísla účtu ani platební karty.

Při registraci si uživatel může vybrat ze dvou účtů:

- **PayPal for you** - Standartní účet vhodný pro uživatele, kteří se registrovali s cílem nakupovat v internetových obchodech.
- **PayPal for business** - Určen pro obchodníky. Součástí účtu je podpora přístupu více uživatelů.

Peníze lze vložit na účet několika možnostmi. Nejjednodušší řešení je spárování PayPal účtu s platební kartou. Platba je potom velice jednoduchá, protože k provedení transakce postačují pouze přihlašovací údaje k PayPal účtu. Další varianta je převedení peněz z běžného bankovního účtu na PayPal účet. Poslední možností je převedení peněz mezi dvěma účty PayPal. [29]

Tabulka 3 – Poplatky PayPal [30]

Nakupování		**Osobní převod**	
Odeslání platby	Příjem platby	Odeslání platby	Příjem platby
zdarma	1,9 % - 3,4 % + 10,00 CZK	zdarma mezi PayPal účty 3,4 % + 10,00 CZK z platební karty	

Tabulka 4 - Výhody a nevýhody platebního systému PayPal [29]

Výhody	**Nevýhody**
Počet uživatelů	Nepodporuje českou lokalizaci
Provázání účtu s platební kartou	Malé rozšíření u českých obchodníků
Jednoduchost	
Bezproblémová reklamace	

Skrill (MoneyBookers)

Skrill (Moneybookers) je dalším zahraničním platebním systémem. Na rozdíl od PayPal je přeložen do češtiny (nabízí 12 jazykových verzí), ale ani česká lokalizace mu nepomohla prosadit se na českém trhu. Vytvoření nového účtu je velice jednoduché a zabere krátký čas. Při registraci uživatel nevyplňuje číslo účtu ani platební karty.

Skrill (Moneybookers) nabízí dva druhy účtů:

- osobní účet,
- podnikatelský účet.

Oba účty jsou svojí funkčností velice podobné účtům na PayPal. Uživatel může převést peníze na svůj účet pomocí bankovního převodu nebo platební kartou. Skrill (MoneyBookers) nabízí možnost převádět peníze do ciziny za místní poplatky a zpětně ze zahraničí na bankovní účet v ČR. Novinkou pro uživatele je i seznam podporovaných internetových obchodů. [29]

Tabulka 5 – Poplatky za osobní účet Skrill (MoneyBookers) [31]

Odeslání platby	Příjem platby	Vklad na účet	Výběr z účtu
1 % (až 0,50 EUR)	zdarma	zdarma	1,80 EUR bank. převod

Tabulka 6 – Poplatky za podnikatelský účet Skrill (MoneyBookers) [31]

Odeslání platby	Příjem platby	Výběr z účtu
1%	1,90 % - 2,90 % + 0,25 EUR	1,80 % EUR bankovní převod a platební karta

Tabulka 7 – Výhody a nevýhody platebního systému Skrill (MoneyBookers) [29]

Výhody	**Nevýhody**
Počet uživatelů	Malé rozšíření u uživatelů a obchodníků v ČR
Provázání účtu s platební kartou	Vyšší poplatky
Česká lokalizace	
Seznam podporovaných obchodů	

PaySec

PaySec je český zástupce mezi platebními systémy a byl představen v roce 2008 finančními institucemi ČSOB a Poštovní spořitelna. Nabízí uživatelům jednoduché a rychlé řešení platby za zboží a služby bez využití platební karty. PaySec umožňuje platit v internetových

obchodech nebo využívat dárcovská tlačítka. Jedná se čistě o elektronickou peněženku. PaySec nemůže funkčností konkurovat platebním systémům PayPal a Skrill (MoneyBookers), ale i tak získal podporu obchodníků na českém trhu.

PaySec nabízí výběr ze dvou typů účtů:

- Konto PaySec,
- Konto pro obchodníky.

Účet lze nabít pomocí běžného účtu, platební kartou nebo převodem z jiného účtu PaySec a vybít na běžný účet. Oproti PayPal neumožňuje propojit účet s platební kartou. [32]

Tabulka 8 – Poplatky platebního systému PaySec [33]

	Konto PaySec	**Konto pro obchodníka**
Platba na jiný PaySec účet	zdarma	zdarma
Platba za nákup	zdarma	zdarma
Přijetí platby od obchodníka	zdarma	-
Přijetí platby od PaySec Konta	-	individuálně
Přijetí platby z darovacího tlačítka	1 CZK	individuálně
Nabití z běžného účtu	zdarma	zdarma
Nabití platební kartou	2 % z nabité částky	2 % z nabité částky
Vybití na běžný účet ČSOB/ Poštovní spořitelna	zdarma	zdarma
Vybití na běžný účet v jiné bance	2 CZK	3 CZK

Tabulka 9 – Výhody a nevýhody platebního systému PaySec [32]

Výhody	**Nevýhody**
Podpora českých obchodníků	Nelze provázat účet s platební kartou
Jednoduchost	
Snadná integrace	
Český platební systém	
Čitelná cenová politika a nízké poplatky	

GoPay

GoPay je český platební systém. Jedná se o jeden z nejmladších poskytovatelů služeb elektronické peněženky. GoPay je rychle se rozvíjející služba, která v roce 2012 ohlásila přes

2000 uzavřených obchodních vztahů. GoPay nabízí uživatelům možnost elektronické platby, posílání a přijímání peněz na internetu a samozřejmě i služby darovacího tlačítka.

Zákazník má možnost výběru ze tří účtů:

- **Kasička** - Dovoluje zákazníkovi provádět platby pouze v rámci systému GoPay a maximální zůstatek na účtu je omezen na 25 000 CZK.
- **Peněženka** - Standartní účet platebních systémů, který umožňuje zákazníkovi provádět on-line platby včetně bankovních převodů. Stejně jako systém PaySec ani GoPay neumožnuje provázání platební karty s účtem.
- **Obchodní účet –** Určen pro obchodníky. Umožňuje příjem on-line plateb za zboží nebo služby. Poskytuje přístup k rozhraní a funkcím platební brány.

Zákazník k dobití účtu elektronickými penězi má několik možností. Nejznámější je převod z běžného účtu, kdy variabilní symbol značí identifikační kód uživatele GoID. GoPay disponuje více bankovními účty, takže si zákazník může vybrat dle typu své banky. GoPay podporuje dobíjení SuperCASH pomocí terminálů Sazky, speciálních GoKuponů a prémiové SMS. Nezvyklou službou je zasílání zpráv mezi zákazníky a obchodníky. [32]

Tabulka 10 – Poplatky platebního systému GoPay [34]

	Kasička	**Peněženka**
Odeslání platby na jiný GoPay účet	zdarma	zdarma
Odeslání platby na bankovní účet	nelze používat	10 CZK
Přijetí platby z jinýho GoPay účetu	zdarma	zdarma
Přijetí platby bankovním převodem	zdarma	zdarma
Přijetí platby z darovacího tlačítka	0,9 % + 1 CZK	0,9 % + 1 CZK
Přijetí platby platební kartou	1,9 % + 1 CZK	1,9 % + 1 CZK
Přijetí platby SuperCASH	1,9 % + 1 CZK	1,9 % + 1 CZK

Tabulka 11 – Výhody a nevýhody platebního systému GoPay [32]

Výhody	**Nevýhody**
Český platební systém	Nelze provázat účet s platební kartou
Jednoduchost	Vyšší poplatky
Podrobný návod integrace	
Čitelná cenová politika	

2 Knihovna MyGateway

V této kapitole se budu zabývat návrhem a analýzou objektové knihovny umožňující platby pomocí několika komerčních platebních bran.

2.1 Stanovení cílů

Cílem projektu je vytvoření objektové knihovny usnadňující práci s více druhy komerčních platebních bran. Pracovní název knihovny je MyGateway a v základu bude zahrnovat platební systémy PayPal, PaySec a GoPay. Každá platební brána vybraného systému požaduje jiné vstupní parametry k sestavení platebního příkazu. Knihovna bude představovat abstraktní vrstvu nezávislou na platebních branách, umožňující realizaci platebního příkazu a ověření stavu realizované platby. Struktura knihovny bude navržena tak, aby bylo možné implementovat podporu dalších platebních systémů.

Knihovna MyGateway bude určena pro systémy, které hledají řešení podpory více platebních bran, aby umožnily svým uživatelům jejich výběr. Její implementací do systému knihovna poskytne funkce pro realizaci a ověření jednorázové platby za zboží nebo služby. Důležitou funkcí knihovny bude ověření stavu platby.

K nastavení vlastností knihovny bude sloužit konfigurační soubor, kde bude potřeba vyplnit požadované informace pro jednotlivé platební brány. Uživatel nemusí vlastnit obchodní účty všech podporovaných systémů, a proto bude mít možnost vybrat pouze ty, které chce mít v rámci své aplikace aktivní.

2.2 Technologie použité při vývoji knihovny

Pro splnění cíle návrhu objektové knihovny byl jako hlavní programovací jazyk zvolen PHP 5. Bylo přihlíženo k dosavadním zkušenostem s tvorbou webových aplikací a výběr jazyka ovlivnila podpora objektově orientovaného programování od verze PHP 5.

Jako vývojové prostředí byl použit nástroj NetBeans 7.3 a pro kreslení UML diagramů posloužily nástroje Visual Paradigm for UML 10.0 a Microsoft Visio Professional 2013.

Při integraci platebních systémů PayPal, PaySec a GoPay byly použity integrační návody jednotlivých systémů, které jsou veřejně přístupné na jejich domovských stránkách.

2.3 Analýza knihovny MyGateway

Use Case diagram

Knihovna MyGateway rozděluje účastníky na tři skupiny – správce aplikace, webová aplikace a platební systémy. Každý účastník má svoji specifickou roli v rámci systému.

- **Správce aplikace –** Pomocí konfiguračního souboru správce nastavuje vlastnosti knihovny. Nastaví podporované platební systémy a vyplní k nim požadované informace (např. číslo obchodníkova účtu). Informace se liší v závislosti na plateb-

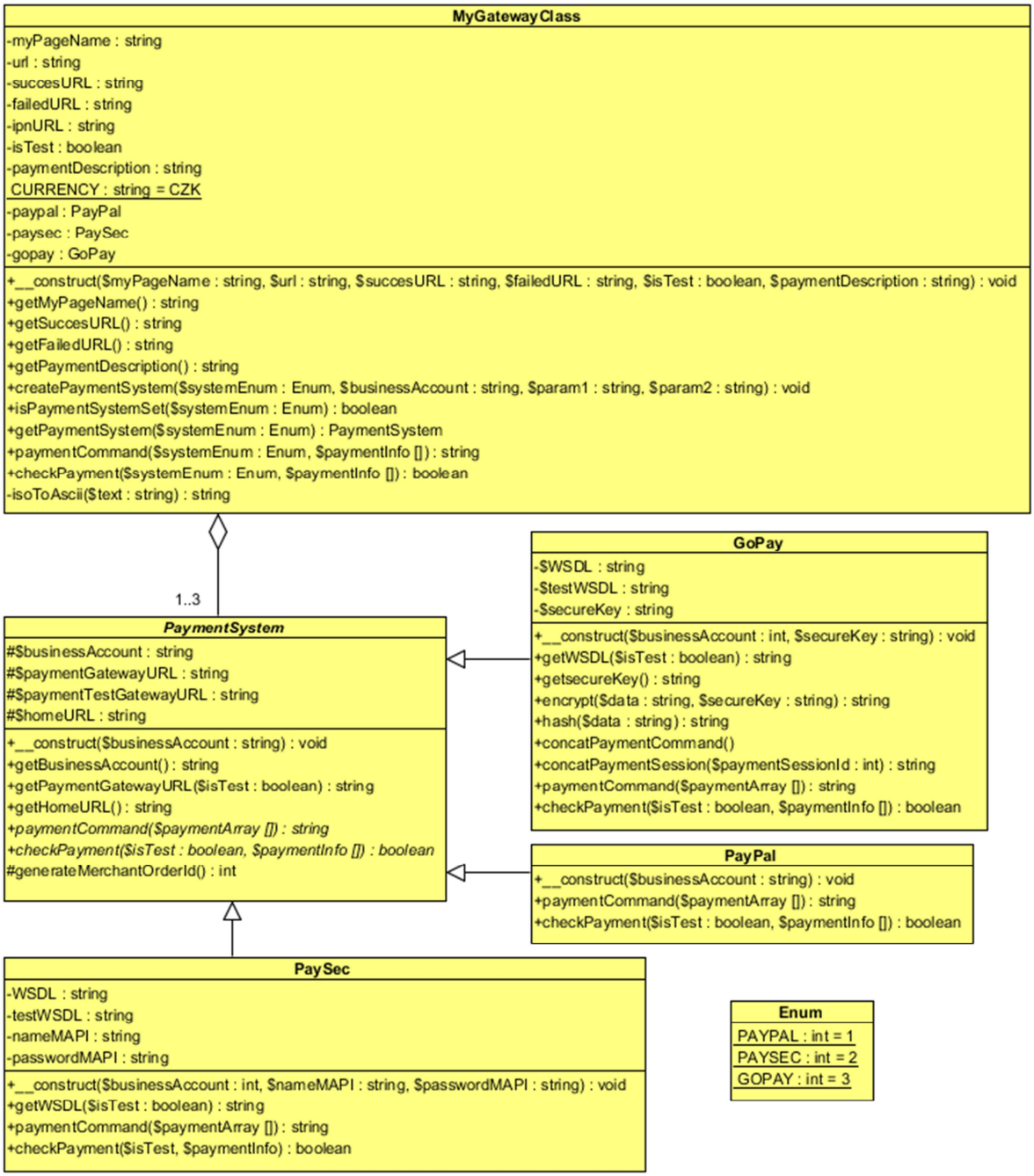
ním systému. Správce má možnost podpory všech nebo jenom jednoho systému. V rámci konfigurace všech platebních systému má možnost volby mezi produkčním nebo testovacím prostředím. Správce v nastavení stanový počet a výší příspěvků.

- **Webová aplikace –** Má k dispozici výběr podporovaného platebního systému, se kterým může provádět realizaci a ověření platby.
- **Platební systémy** (PayPal, PaySec, GoPay) – Pomocí předaných parametrů na platební bránu umožňují realizaci platebních příkazů. Dávají na výběr z několika platebních metod. Poskytují rozhraní k ověření stavu platby.

Obrázek 6 – Use Case diagram knihovny MyGateway [vlastní]

Diagram tříd

Pro dosažení požadovaných cílů bylo nutné vytvořit strukturu objektové knihovny MyGateway. Na obrázku 7 je zobrazen statický pohled na strukturu systému. Popisuje atributy a operace jednotlivých knihovních tříd a vztahy mezi nimi.



Obrázek 7 – Diagram tříd knihovny MyGateway [vlastní]

Třída MyGatewayClass uchovává atributy obsahující informace o nastavení knihovny načtených z konfiguračního souboru. Každý atribut má nastaven obor vlastnosti na *private*,

tzn. je přístupný pouze uvnitř třídy, a proto třída poskytuje veřejné metody, které umožní zpřístupnit hodnoty těchto atributů. Definice atributů probíhá při zavolání parametrického konstruktoru. Pouze instance tříd PayPal, PaySec a GoPay jsou vytvářeny při zavolání metody *createPaymentSystem()*, jenž vrací instanci třídy reprezentující platební systém. Metoda obsahuje parametr identifikace obchodního účtu, další dva nepovinné parametry nezávislé na platebním systému a poslední parametr určí, od jaké třídy bude instance vytvořena. Metoda *isPaymentSystemSet()* zjišťuje, jestli instance třídy odpovídající parametru je vytvořena. Hlavní podstatu třídy představují metody *paymentCommand()* a *checkPayment()*. Jak už jejich název napovídá, slouží k realizaci a ověření platby. První zmíněná metoda má vstupní parametr pole informací potřebných pro vytvoření platebního příkazu. Výstupem metody je řetězec s adresou a hodnotami pro vybranou platební bránu. Naproti tomu metoda *checkPayment()* slouží k ověření stavu platby. Vstupním parametrem je pole informací obdržených od platební brány, které je nutné zpracovat a znovu odeslat. Metoda vrací booleovskou proměnou představující úspěšné či neúspěšné provedení platby. Poslední metoda *isoToAscii()* je viditelná pouze v rámci třídy MyGatewayClass. Slouží k převedení řetězce s diakritikou na řetězec bez diakritiky. Uplatňuje se v rámci metody *paymentCommand()* na vstupní informace představující jméno a příjmení uživatele realizující platbu.

Třída PaymentSystem je navržena jako abstraktní třída. Nelze od ní vytvořit instanci a slouží jako šablona, od které se odvozují ostatní třídy. Deklaruje společné atributy pro všechny platební systémy. Atributy jsou zapouzdřeny oborem vlastnosti *protected* a jsou přístupné pouze z odvozených tříd. Třída obsahuje „získávací“ metody, které mají na starost zapouzdření kódu a využívají se při získávání hodnot atributů. Metody *paymentCommand()* a *checkPayment()* jsou abstraktní a deklarují se v rodičovských třídách. Naznačují, že implementace se liší v závislosti na zvolené platební bráně. Metoda *generateMerchantOrderId()* má obor vlastnosti *protected* a jejím výstupem je unikátní identifikační číslo realizované platby.

Příkaz na platební bránu PayPal nevyžaduje žádné specifické parametry, a proto ve stejnojmenné třídě není potřeba deklarovat další atributy. Třída PayPal poskytuje parametrický konstruktor, který zavolá konstruktor rodičovské třídy, a nastaví hodnoty všech atributů. Z rodičovské třídy jsou implementovány metody *paymentCommand()* a *checkPayment()*, které představují příkazy vytvořené platební bránu PayPal.

Třída PaySec oproti předchozí požaduje deklaraci dalších atributů. Je to mu z důvodu implementace metody *checkPayment()*, protože platební brána PaySec poskytuje obchodníkům pro ověření platby rozhraní označované jako MAPI (MerchantAPI). Atributy tak slouží k nastavení parametrů volané funkce MAPI rozhraní.

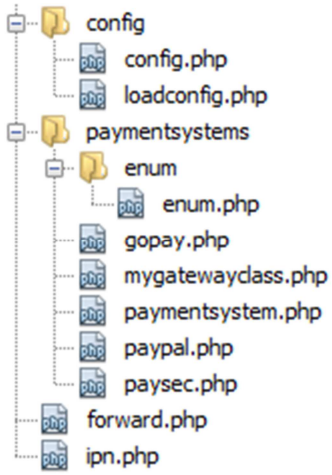
Implementace platební brány GoPay se od ostatních nejvíce liší. Z toho důvodu je třída GoPay rozšířena o nové atributy a metody. Atribut *$secureKey* je šifrovací klíč, obdržený při založení GoPay obchodního účtu. K zašifrování odesílaných dat na platební bránu slouží metoda *encrypt()*. Požaduje dva vstupní parametry - data a šifrovací klíč. Hashovací

funkce *hash()* vrací otisk vstupních dat. Součástí příkazu na platební bránu GoPay je povinný parametr složený ze všech vlastností platebního příkazu. Parametr představuje řetězec, který vrací metody *concatPaymentCommand()* a *concatPaymentSession()*. U první jmenované metody nejsou v diagramu zakresleny všechny vstupní parametry, protože by bylo potom těžké diagram zobrazit v rozumné velikosti. Platební brána GoPay využívá pro ověření platby rozhraní stejně jako platební systém PaySec.

K rozlišení platebního systému slouží třída Enum. V programovacím jazyce PHP chybí podpora výčtového typu, a proto tato třída nahrazuje její funkčnost. Jsou v ní definované konstanty ke každému podporovanému platebnímu systému knihovny MyGateway.

2.4 Adresářová struktura

Knihovna obsahuje adresáře *config*, *paymentsystem*, ve kterých je uložena většina souborů. Adresářová struktura je zobrazena na obrázku 8.



Obrázek 8 – Adresářová struktura knihovny MyGateway [vlastní]

Obsah adresářů:

- **config** – Obsahuje konfigurační soubory *config.php*, nastavený správcem knihovny, a *loadconfig.php*, který automaticky nastaví vlastnosti knihovny na hodnoty z konfiguračního souboru.
- **paymentsystems** – Zde jsou uloženy všechny třídy knihovny.

Knihovna MyGateway dále obsahuje důležité skripty:

- **forward.php** – Na tento skript budou přeposlány informace získané z webové aplikace. Zde se následně zpracují a předají vybrané platební bráně.
- **ipn.php** (informace o platebním oznámení) – Skript zpracuje příchozí informace v odpovědi platební brány na provedení platby.

2.5 Popis vybraných částí kódu

V následující podkapitole budou vybrány a popsány důležité části kódu knihovny MyGateway:

- Metoda vytvoření instance třídy platebního systému.
- Funkce pro zpracování informací předaných z webové aplikace.
- Vytvoření příkazu na platební bránu PayPal
- Ověření stavu platby systému PayPal.
- Ověření platby pomocí rozhraní MAPI platebního systému PaySec.
- Ověření stavu platby pomocí rozhraní platebního systému GoPay

Metoda vytvoření instance třídy platebního systému

Metoda (Obrázek 9) vytváří instanci třídy vybraného platebního systému, jejíž referenci uchovává jako atribut. Vytvořením instance dojde k podpoře a zpřístupnění funkcí konkrétního platebního systému. Hlavička metody obsahuje dva povinné a dva nepovinné vstupní parametry. Počet parametrů je závislý na vybrané třídě, od které se vytváří instance platebního systému. Povinné parametry jsou volba platebního systému a účet obchodníka. Pro systém PaySec představují nepovinné parametry přihlašovací jméno a heslo do rozhraní MAPI. U systému GoPay se zadá pouze jeden nepovinný parametr, který obsahuje šifrovací klíč.

```
public function createPaymentSystem(
        $systemEnum,
        $businessAccount,
        $param1 = NULL,
        $param2 = NULL) {
    if ($systemEnum == Enum::PAYPAL) {
        $this->paypal = new PayPal($businessAccount);
    } elseif ($systemEnum == Enum::PAYSEC) {
        $this->paysec = new PaySec($businessAccount, $param1, $param2);
    } elseif ($systemEnum == Enum::GOPAY) {
        $this->gopay = new GoPay($businessAccount, $param1);
    }
}
```

Obrázek 9 – Metoda createPaymentSystem() [vlastní]

Funkce pro zpracování informací předaných z webové aplikace

Informace předané z webové aplikace je nutné v rámci abstraktní vrstvy nejdříve zpracovat a uložit do pole s takovými klíči, se kterými dokáže pracovat jakákoliv třída reprezentující platební systém. Nejprve je nutné otestovat podmínku, že instance třídy platebního systému, je vytvořena. Pokud se tak dosud nestalo, platební systém není podporován a funkce vrací hodnotu NULL. Jinak se zavolá metoda zpřístupňující referenci na vybraný platební systém. Předané informace jsou zpracovány a uloženy do pole, které představuje vstupní parametr metody volané k vytvoření platebního příkazu zpřístupněné reference.

```php
public function paymentCommand($systemEnum, $paymentInfo) {
    if ($this->isPaymentSystemSet($systemEnum)) {
        $paymentSystem = $this->getPaymentSystem($systemEnum);
        $paymentArray = array(
            "isTest" => $this->isTest,
            "paymentDescription" => $this->paymentDescription,
            "currency" => self::CURRENCY,
            "ipnURL" =>$this->ipnURL,
            "systemEnum" => $systemEnum,
            "failedURL" => $this->failedURL,
            "amount" => $paymentInfo["amount"],
            "name" => $this->isoToAscii($paymentInfo["name"]),
            "surname" => $this->isoToAscii($paymentInfo["surname"]),
            "email" => $paymentInfo["email"]);
        return $paymentSystem->paymentCommand($paymentArray);
    }
    return NULL;
}
```

Obrázek 10 – Metoda paymentCommand() pro zpracování informací [vlastní]

Vytvoření příkazu na platební bránu PayPal

Pro vytvoření platebního příkazu je nutné předat platební bráně požadované vstupní parametry. Sestavení příkazu zajišťuje metoda *paymentCommand()*. Platební příkazy na jednotlivé platební brány jsou si velice podobné, a proto bude uveden pouze příklad pro platební systém PayPal.

- **item_name** – Představuje popis zboží nebo služby.
- **business –** Email obchodního účtu.
- **cmd** – Charakterizuje typ platebního příkazu.
- **currency_code –** Použitá měna.
- **return –** Zobrazená stránka po úspěšném provedení platby.
- **cancel_return –** Zobrazená stránka po neúspěšném provedení platby.
- **rm –** Udává, pomocí jaké metody mají být vráceny data v odpovědi platební brány. Parametr nastavený na hodnotu 2 využívá metodu POST.
- **os0, os1 –** Slouží k doplnění informací o platbě. Uchovávají jméno a příjmení zákazníka.
- **amount –** Hodnota transakce.
- **item_number –** Unikátní číslo transakce. Je nutné zajistit její jedinečnost.

```php
public function paymentCommand($paymentArray) {
    $str = "";
    $str .= $this->getPaymentGatewayURL($paymentArray["isTest"]);
    $str .= '?item_name=' . $paymentArray["paymentDescription"];
    $str .= '&business=' . urldecode($this->businessAccount);
    $str .= '&cmd=_donations';
    $str .= '&currency_code=' . $paymentArray["currency"];
    $str .= '&return=' . $paymentArray["ipnURL"] .
            '?system=' . $paymentArray["systemEnum"];
    $str .= '&cancel_return=' . $paymentArray["failedURL"];
    $str .= '&rm=2';
    $str .= '&on0=Jmeno';
    $str .= '&os0=' . $paymentArray["name"];
    $str .= '&on1=Prijmeni';
    $str .= '&os1=' . $paymentArray["surname"];
    $str .= '&amount=' . $paymentArray["amount"];
    $str .= '&item_number=' . $this->generateMerchantOrderId();
    return $str;
}
```

Obrázek 11 – Vytvoření platebního příkazu PayPal [vlastní]

Ověření stavu platby systému PayPal

Vstupními parametry metody (Obrázek 12) jsou informace o platbě a hodnota, která udává typ platební brány – produkční nebo testovací. Systém PayPal vrací pomocí metody POST informace o provedené platbě. Pro ověření platby je potřeba znovu sestavit platební příkaz z příchozích dat, který bude odeslán na platební bránu. Parametr *cmd* je nutné nastavit na tuto hodnotu „*cmd=_notify-validate*“. Platební brána tak rozpozná, že jde o příkaz ověření platby.

```php
public function checkPayment($isTest, $paymentInfo) {
    $req = 'cmd=_notify-validate';

    foreach ($paymentInfo as $key => $value) {
        $value = urlencode(stripslashes($value));
        $req .= "&$key=$value";
    }

    $url = $this->getPaymentGatewayURL($isTest);
    $res = file_get_contents($url . "?" . $req);

    if (strcmp(trim($res), "VERIFIED") == 0) {
        return TRUE;
    } else if (strcmp(trim($res), "INVALID") == 0) {
        return FALSE;
    }
}
```

Obrázek 12 – Ověření platby systému PayPal [vlastní]

Ověření platby pomocí rozhraní MAPI platebního systému PaySec

Ke komunikaci s rozhraním MAPI je potřeba podpory modulu SOAP ve webovém serveru. Jinak bude zobrazováno chybové hlášení *„Fatal error: Class 'SoapClient' not found“*. Jedná se o nejčastější chybu při navazování komunikace s rozhraním. Bohužel tento požadavek není v integračním manuálu systému PaySec.

K ověření transakce se využívá metoda *VerifyTransactionIsPaid*, která požaduje uživatelské jméno, heslo, identifikační číslo a částku transakce. Návratová hodnota udává stav transakce. Implementace metody *VerifyTransactionIsPaid* je povinná v rámci integrace systému PaySec.

```
public function checkPayment($isTest, $paymentInfo) {
    $paysecMapi = new SoapClient($this->getWSDL($isTest));
    $result = $paysecMapi->VerifyTransactionIsPaid(
            $this->nameMAPI,
            $this->passwordMAPI,
            $paymentInfo["moid"],
            $paymentInfo["value"]);
    if ($result == 0) {
        return TRUE;
    }
    return FALSE;
}
```

Obrázek 13 – Ověření platby systému PaySec [vlastní]

Ověření stavu platby pomocí rozhraní platebního systému GoPay

Platební systém GoPay poskytuje k ověření transakce webové rozhraní. Stejně jako u systému PaySec je nutné mít součástí webového serveru povolený modul SOAP k navázání komunikace.

Funkce webové rozhraní se volá pomocí metody __call(), která má dva parametry – název funkce a potřebná data. V případě ověření transakce představují data pole hodnot, obsahující identifikační číslo obchodníkova účtu, identifikační číslo relace a zašifrovaný otisk podpisu platebního příkazu. Webové rozhraní při úspěšném ověření transakce vrací pole informací o realizované platbě spolu s parametrem „PAID“.

```php
public function checkPayment($isTest, $paymentInfo) {
    $returnedPaymentSessionId = $_GET['paymentSessionId'];

    ini_set("soap.wsdl_cache_enabled", "0");
    $go_client = new SoapClient($this->getWSDL($isTest), array());

    $sessionEncryptedSignature = $this->encrypt(
            $this->hash(
                $this->concatPaymentSession($returnedPaymentSessionId)),
            $this->secureKey);

    $paymentSession = array(
        "targetGoId" => (float) $this->businessAccount,
        "paymentSessionId" => (float) $returnedPaymentSessionId,
        "encryptedSignature" => $sessionEncryptedSignature);

    $paymentStatus = $go_client->__call('paymentStatus',
            array('paymentSessionInfo' => $paymentSession));
    $result = array();
    $result["sessionState"] = $paymentStatus->sessionState;
    $result["sessionSubState"] = $paymentStatus->sessionSubState;
    if ($result["sessionState"] == "PAID") {
        return TRUE;
    } else {
        return FALSE;
    }
}
```

Obrázek 14 – Ověření platby systému GoPay [vlastní]

3 Implementace knihovny MyGateway

Při vývoji objektové knihovny MyGateway projevilo neziskové občanské sdružení Opočnické divadélko o.s. zájem o vytvoření aplikace pro získávání finančních příspěvků na jejich aktivity. Aplikace bude využívat knihovnu MyGateway a v této kapitole bude popsán její návrh.

3.1 Stanovení cílů

Cílem kapitoly je vytvoření aplikace představující univerzální darovací tlačítko, která bude využívat funkce poskytované knihovnou MyGateway. Aplikace bude určena neziskovým organizacím a autorům webových stránek, kteří nic neprodávají, ale nabízejí zajímavý obsah. Pomocí univerzálního darovacího tlačítka budou moci získávat finanční podporu pro jejich aktivity.

Návštěvníkům stránky bude zobrazeno jednoduché platební tlačítko, po jehož stisknutí jim bude předložen formulář, kde si vyberou platební systém a darovanou částku. Po odeslání dat na platební bránu budou přesměrováni do prostředí vybraného platebního systému. Zde k předvyplněnému platebnímu příkazu vyberou platební metodu a dokončí platbu.

Aplikace bude implementována do webové stránky pomocí jednoduchých příkazů, aby byla dostupná i pro uživatele s minimálními znalostmi tvorby webových aplikací. Navržení vzhledu univerzálního darovacího tlačítka, bude umožnovat umístění aplikace kdekoliv ve webové stránce. Aplikace bude poskytovat výpis všech přispívaných plateb.

3.2 Programátorská příručka

Popisuje návrh aplikace z pohledu programátora. Zahrnuje výběr aktuálních technologií, návrh struktury aplikace a podrobnější popis implementace knihovny MyGateway.

3.2.1 Technologie použité při realizaci aplikace

Bylo přihlíženo k použití aktuálních technologií na trhu. Statická část aplikace je kódována v jazyce HTML a je validní se standardem HTML5. Pro návrh grafického designu je využito kaskádových stylů CSS, které jsou validní s CSS3.

Vzhledem k použití knihovny MyGateway je u aplikace použit jako hlavní programovací jazyk PHP 5.

V aplikaci je použita relační databáze MySQL 5.0 a správa databáze probíhala pomocí webové aplikace phpMyAdmin 3.3.7.

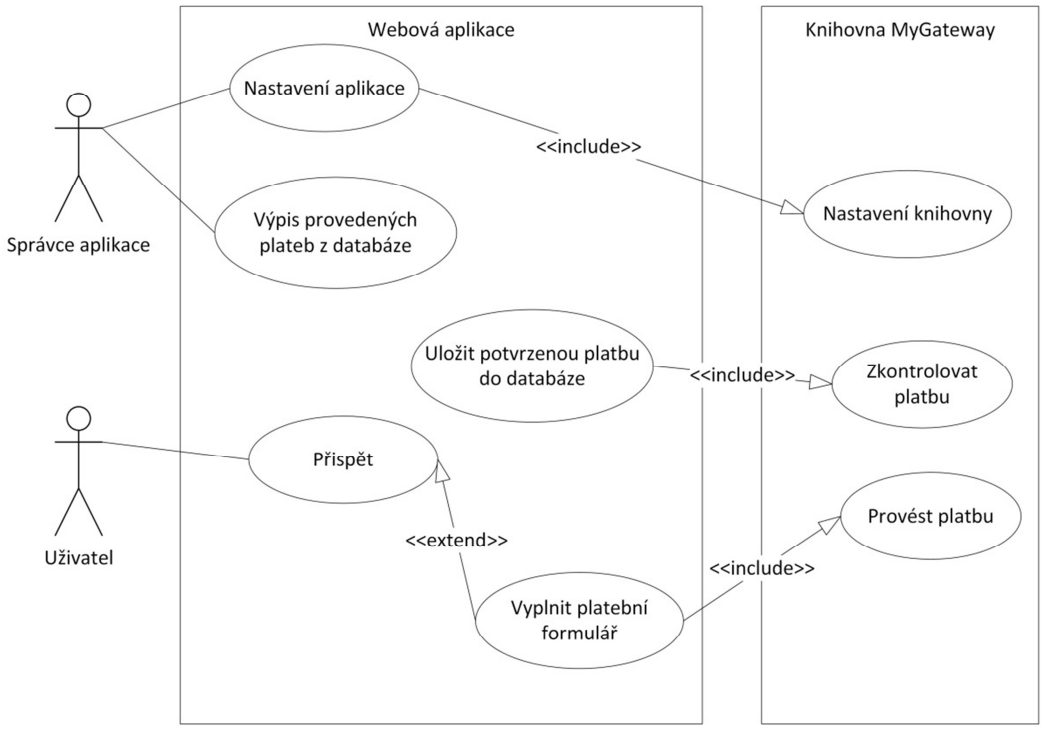
Pro kontrolu formuláře a dosažení uživatelského komfortu byla využita technologie JavaScript. V tomto případě se jedná o knihovnu jQuery, která poskytuje řadu funkcí, umožňující dynamicky měnit obsah webového dokumentu.

Aplikace byla vyvíjena na webovém serveru Apache 2.2 s operačním systémem Debian. Vzhledem ke zkušenostem s vývojovým prostředím byl použit nástroj NetBeans 7.3. Pro návrh UML diagramů byly použity nástroje Visual Paradigm for UML 10.0 a Microsoft Visio Professional 2013.

3.2.2 Analýza aplikace

Use Case diagram

- **Správce** - Má možnost nastavení aplikace. Tato možnost zahrnuje nastavení poskytované knihovnou MyGateway. Dále má správce k dispozici přehled všech provedených plateb, které jsou uloženy v databázi. V přehledu jsou pouze ty platby, které byly ověřeny.
- **Uživatel** - Má možnost přispět darovanou částku. Tuto možnost rozšiřuje platební formulář, který využívá funkce knihovny vybrat platební systém a provést platbu.



Obrázek 15 – UseCase diagram aplikace [vlastní]

Activity diagram

V Příloze A je zachycen aktivity diagram, který popisuje logiku procesu přispění pomocí univerzálního darovacího tlačítka.

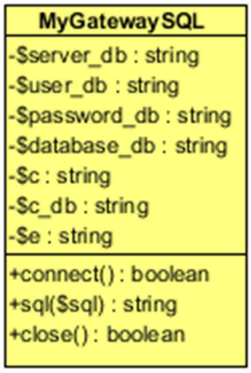
Proces je zahájen stiskem tlačítka přispět ve webové aplikaci. Pokračuje k akci výběru platebního systému a vyplnění povinných údajů formuláře. Následuje kontrola vyplnění formuláře. Pokud je formulář špatně vyplněn, vrací se zpět k vyplnění formuláře. Jestli je formulář správně vyplněn, údaje jsou odeslány na vybraný platební systém. Zde se provede

realizace platby. Postup se liší v závislosti na vybraném platebním systému. Po provedení platby se ověří stav platby. Pokud byla platba úspěšně ověřena, uloží se informace o platbě do databáze systému a je zobrazena stránka pro dokončení platby. Jestliže platba nebyla ověřena, je zobrazena stránka neúspěch platby.

Diagram tříd

Aplikace bude poskytovat výpis úspěšně provedených plateb. Ty budou uloženy v relační databázi, a proto byla navržena třída MyGatewayClass, která s ní značně zjednodušuje práci.

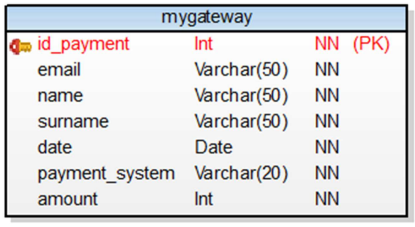
Třída MyGatewayClass (Obrázku 16) obsahuje atributy, obsahující informace potřebné k připojení ke konkrétní databázi. Dále poskytuje metody k navázání a ukončení spojení a provedení SQL dotazu nad vybranou tabulkou.



Obrázek 16 – Diagram tříd aplikace [vlastní]

3.2.3 Návrh databáze

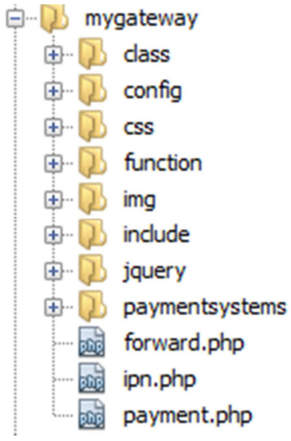
Návrh databáze je velice jednoduchý. Hlavní myšlenkou je uchovávání provedených transakcí, které byly ověřeny. Poskytnout tak správci jednoduchý přehled o tom kdo, kdy a jakou částkou přispěl.



Obrázek 17 – Návrh databázové tabulky [vlastní]

3.2.4 Adresářová struktura

Celá aplikace spolu s objektovou knihovnou MyGateway je uložena v adresáři *mygateway*. Do podadresářů *class*, *config*, *css*, *function*, *img*, *include*, *jquery*, *paymentsystems* jsou rozděleny ostatní soubory. Adresářová struktura je zobrazena na obrázku 18.



Obrázek 18 – Adresářová struktura aplikace [vlastní]

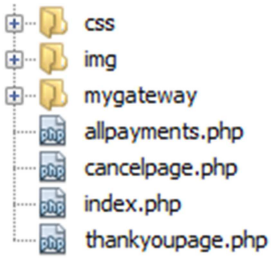
Obsah podadresářů:

- **class** – Obsahuje třídu MyGatewayClass.
- **config** – Část knihovny MyGateway.
- **css** – Obsahuje externí soubor zajišťující vzhled pomocí kaskádových stylů.
- **function –** Zde jsou uloženy PHP funkce využívané v celé aplikaci.
- **img –** Uchovává loga podporovaných platebních systémů.
- **include** – Jsou zde uloženy PHP skripty pro výpis všech ověřených příspěvků a formát vzhledu univerzálního darovacího tlačítka.
- **jquery** – Zde je uložena javascriptová knihovna jQuery.
- **Paymentsystems** – Část knihovny MyGateway.

Adresář *mygateway* dále obsahuje soubory:

- **forward.php -** Část knihovny MyGateway.
- **ipn.php** - Část knihovny MyGateway doplněna o vkládání dat do databáze po ověření platby.
- **payment.php** - Představuje platební formulář, na který je uživatel přesměrován po stisku univerzálního platebního tlačítka.

Pro lepší přehled je na obrázku 19 zobrazena adresářová struktura testovací webové stránky z pohledu hlavního kořenového adresáře.



Obrázek 19 – Adresářová struktura z pohledu kořenového adresáře [vlastní]

Obsah kořenového adresáře:

- **css –** Obsahuje externí soubor kaskádových stylů pro testovací stránku.
- **img –** Obrázky použité pro vzhled testovací stránky.
- **mygateway –** Obsahuje aplikaci, jejíž adresářová struktura je zobrazena na Obrázku 23.
- **allpayments.php –** Testovací stránky pro výpis všech realizovaných transakcí.
- **cancelpage.php –** Testovací stránka zobrazená po neúspěchu transakce.
- **thankyoupage.php –** Testovací stránka zobrazená po úspěšně provedené transakci.
- **index.php –** Testovací stránka pro demonstraci vzhledu univerzálního darovacího tlačítka.

3.2.5 Popis vybrané části kódu

Nejzajímavější a nejvíce problematickou částí aplikace při jejím vývoji byla implementace ověření, zda platba proběhla úspěšně či neúspěšně (soubor *ipn.php*). Knihovní funkce dokáže navázat komunikaci s platební branou a zjistit stav platby, ale problém spočíval ve zpracování příchozích dat od platební brány, která musejí být vložena do databáze, pokud platba proběhla úspěšně. Data jsou uložena v poli a každá platební brána vrací pole s jinými názvy klíčů jednotlivých parametrů.

K vložení dat je použita statická funkce *insertPaymentInfoIntoDB()*, jejíž vstupní parametry představují právě ty hodnoty, které se ukládají do databáze. Na funkci je ukázáno, jak pracuje s třídou MyGatewayClass. Nejdříve je zavolána metoda pro navázání spojení s databází, potom je proveden samotný SQL dotaz, po jehož dokončení je ukončeno spojení s databází.

```
public static function insertPaymentInfoIntoDB($id_payment, $email, $name, $surname, $amount, $system) {
    $db = new MyGatewaySQL();
    $db->connect();
    $db->sql("INSERT INTO mygateway (id_payment, email, name, surname, amount, date, payment_system)
                          VALUES ('" . $id_payment . "', '" . $email . "','" .
            $name . "', '" . $surname . "', '" . $amount . "', CURDATE(), '" . $system . "')");

    $db->close();
}
```

Obrázek 20 – Funkce pro vložení dat do databáze [vlastní]

Před zavoláním samotné funkce *insertPaymentInfoIntoDB()* se zjistí, ze které platební brány přišla příchozí data. Hodnota proměnné $_GET["system"] udává, jaký platební systém bude použit pro ověření platby.

3.3 Uživatelská příručka

Popisuje aplikaci z pohledu uživatele, kterého představuje návštěvník stránky a správce webové aplikace. Aplikace univerzálního darovacího tlačítka je poskytována spolu s testovacími stránkami. Ty slouží jako příklady k porozumění chování aplikace.

3.3.1 Nastavení aplikace

Nastavení aplikace je implementováno v rámci knihovny MyGateway a je uloženo v souboru *mygateway/config/config.php*. Je doporučeno zamezit přístup k souboru, protože uchovává citlivé informace o připojení k rozhraní MAPI platebního systému PaySec. Pomocí těchto údajů se sice útočník nepřipojí do systémů PaySec, takže nemůže manipulovat s účtem, ale i tak může napáchat značné škody. Zabezpečení souboru je ponecháno na řešení konkrétní webové aplikace.

Přehled nastavení:

- **$myPageName –** Název platební stránky uvedený na darovacím tlačítku.
- **$isPaypalSupported –** Nastavení podpory platebního systému PayPal.
- **$isPaySecSupported –** Nastavení podpory platebního systému PaySec.
- **$isGoPaySupported –** Nastavení podpory platebního systému GoPay.
- **$isTest –** V jakém stavu bude aplikace pracovat – produkční nebo testovací.
- **$amounts –** Nastavení počtu a výše příspěvků.
- **$defaultPayment –** Volba před vybraného platebního systému. Návštěvník může změnit v platebním formuláři.
- **$paymentDescription –** Stručný popis transakce (např. darovací tlačítko).
- **$succesURL –** Zobrazená stránka po úspěšné platbě.
- **$failedURL –** Zobrazená stránka po neúspěšné platbě.

Pokud je podporován platební systém PayPal:

- **$myPayPalEmail –** Email obchodního účtu, na které bude transakce směrovat.

Pokud je podporován platební systém PaySec:

- **$microAccountNumber –** Číslo obchodníkova účtu.
- **$nameMAPI –** Uživatelské jméno k přihlášení do rozhraní MAPI.
- **$passwordMAPI –** Uživatelské heslo do rozhraní MAPI.

Pokud je podporován platební systém GoPay:

- **$targetGoID –** Číslo obchodníkova účtu. V systému GoPay označeno jako GoID.
- **$secureKey –** Šifrovací klíč zabezpečující posílaná data na platební bránu. Je obdržen při vytvoření obchodního účtu.

3.3.2 Instalace aplikace

Implementace univerzálního darovacího tlačítka požaduje po webové stránce hostování na serveru, kde je nainstalován HTTP server Apache s podporou modulů PHP a SOAP. Dalším požadavkem je relační databáze MySQL. Pro produkční nebo testovací využití aplikace je požadováno mít vytvořený obchodní účet u každého platebního systému, který bude

nastavený jako aktivní. Webová aplikace musí mít vytvořené stránky pro zobrazení úspěšné či neúspěšné platby, které se nastavují v nastavení knihovny MyGateway.

Postup instalace:

1. Zkopírovat adresář *mygateway* do adresáře, který má webová aplikace nastavený jako Document Root. Tento adresář je nejčastěji pojmenován jako *www*. Adresářová struktura aplikace musí být dodržena, jak byla popsána v kapitole 3.2.4.

2. Pomocí webové aplikace phpMyAdmin se importuje do databáze soubor *databaze/mygateway.sql*, který obsahuje strukturu tabulky pro ukládání dat provedených transakcí.

3. Dalším krokem je nastavení připojení k databázi. Pomocí textového editoru otevřít soubor *mygateway/class/database.php* a na řádcích 5, 6, 7, 8 vyplnit údaje o připojení k databázi. Pokud je databáze na stejném serveru jako Apache, stačí ponechat adresu serveru na hodnotě *localhost*.

4. Nyní se provede nastavení knihovny MyGateway. Opět pomocí textového editoru otevřít soubor *mygateway/config/config.php* a podle uvedených příkladů v komentářích vyplnit jednotlivé parametry. Pokud uživatel nemá založený obchodní účet některého z podporovaných platebních systémů, nastaví ho jako neaktivní. Zvýšená pozornost se věnuje správnému formátu vyplňovaných údajů. Knihovna je defaultně nastavena na testovací prostředí.

5. Přehled výpisu všech realizovaných plateb je uložen ve skriptu *mygateway/include/information.php* a je nutné ho vložit do stránky, kterou má správce aplikace připravenou k přehledu všech transakcí.

6. Posledním krokem je vložení univerzálního darovací tlačítka. V textovém editoru otevít hlavní stránku webové aplikace. Ve většině případů je označena jako *index.php* a do elementu „<head>“ vložit příkaz na obrázku 21. Následně se vloží univerzální darovací tlačítko pomocí příkazu (Obrázek 22) do hlavního nebo postranního panelu webové aplikace.

```
<link rel="stylesheet" href="mygateway/css/styles.css" type="text/css"/>
```

Obrázek 21 – Připojení externího CSS souboru [vlastní]

```
<?php
include 'mygateway/include/mygatewaybutton.php';
?>
```

Obrázek 22 – Vložení univerzálního darovacího tlačítka [vlastní]

Z hlediska aplikace je instalace dokončena a systém je připraven k uvedení do provozu. Součástí Přílohy B a Přílohy C je náhled na zobrazení aplikace.

3.3.3 Ovládání aplikace

Ovládání aplikace je velice jednoduché. Pro nastavení systému stačí správci jednoduchý textový editor, ve kterém jsou uvedeny jednotlivé příklady vyplňovaných údajů.

Návštěvníkovi webové aplikace je po stisknutí univerzálního platebního tlačítka předložen formulář, kde si vybere, jaký platební systém provede transakci. Podle jeho volby vyplní požadované údaje. Rozdíl spočívá při vybrání platebních systémů PaySec a GoPay, ke kterým musí uživatel zadat svoji emailovou adresu. Po odeslání formuláře je přesměrován na vybranou platební bránu k provedení transakce. Po dokončení transakce je navrácen zpět na stránky webové aplikace, kde je informován o stavu platby.

V případě systému PayPal je uživateli zobrazena stránka s přehledem informací o prováděné platbě a je vyzván k přihlášení do systému. Po jeho provedení potvrdí stávající transakci a vybraná částka je připsána na účet obchodníka.

Platební brána PaySec poskytuje volbu mezi pěti způsoby platby. Uživatel jednu vybere a přihlásí se do systému, kde zkontroluje platební příkaz a potvrdí transakci.

Při výběru platebního systému GoPay má uživatel na výběr z patnácti způsobů, jak může svoji transakci uhradit. Pokud souhlasí s údaji v platebním příkazu, přihlásí se do systému a platbu dokončí.

Závěr

Cílem bakalářské práce bylo vytvoření objektové knihovny umožňující on-line platby prostřednictvím několika platebních bran. Pro ověření funkčnosti byla knihovna implementována do aplikace určené pro neziskové organizace.

Podařilo se vytvořit funkční objektovou knihovnu s podporou komerčních platebních systémů PayPal, PaySec a GoPay. Knihovna umožnuje realizaci a ověření transakcí v rámci jednorázových plateb. Při navrhování struktury knihovny byla brána v úvahu možnost případného rozšíření o další platební systém.

Knihovna byla ověřena nasazením pro potřeby konkrétní aplikace představující univerzální darovací tlačítko. Výsledná aplikace poskytuje neziskovým organizacím možnost získávat finanční prostředky na jejich aktivity. Aplikace byla odzkoušena pomocí testovacích účtů a platebních bran systémů PayPal, PaySec a GoPay.

Vývojem aplikace a knihovny se podařilo dosáhnout požadovaných cílů, tím ale jejich vývoj nekončí. Spíše než přidáním dalšího platebního systému, bude vývoj pokračovat rozšířením funkcionality knihovny. Platební brány poskytují další zajímavé funkce. Na příklad výpis pohybu transakcí z obchodního účtu k vymezenému období nebo možnost periodické platby.

Literatura

1. World Wide Web. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/World_Wide_Web.
2. Webový server. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Webov%C3%BD_server.
3. Apache HTTP Server. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Apache_HTTP_Server.
4. November 2012 Web Server Survey. In: *Netcraft* [online]. 2012 [cit. 2013-05-08]. Dostupné z: http://news.netcraft.com/archives/2012/11/01/november-2012-web-server-survey.html.
5. Internet Information Services. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2012, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Internet_Information_Services.
6. Nginx. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2012, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Nginx.
7. DOČEKAL, Michal. Správa linuxového serveru: Webový server Nginx. In: *LinuxEXPRES: opravdový linuxový magazín* [online]. 2011 [cit. 2013-05-08]. ISSN 1214-9608. Dostupné z: http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-webovy-server-nginx.
8. SCHAFER, Steven M. *HTML, XHTML a CSS: bible [pro tvorbu WWW stránek] : 4. vydání*. 1. vyd. Praha: Grada, 2009, 647 s. ISBN 9788024728506.
9. HyperText Markup Language. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2004, 2013 [cit. 2013-05-08]. Dostupné z: https://cs.wikipedia.org/wiki/HyperText_Markup_Language.
10. Introduction to XML. *W3Schools.com* [online]. 1999, 2013 [cit. 2013-05-08]. Dostupné z: http://www.w3schools.com/xml/xml_whatis.asp.
11. Extensible HyperText Markup Language. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2004, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Extensible_HyperText_Markup_Language.
12. Kaskádové styly. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2004, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styly.

13. GILMORE, W. *Velká kniha PHP 5 a MySQL: kompendium znalostí pro začátečníky i profesionály*. Nové, 3. vyd. Překlad Jan Pokorný. Brno: Zoner Press, 2011, 736 s. Encyklopedie Zoner Press. ISBN 978-80-7413-163-9.

14. ŠVADLENKA, Libor a Radovan MADLEŇÁK. *Elektronické obchodování: kompendium znalostí pro začátečníky i profesionály*. Vyd. 1. Překlad Jan Pokorný. Pardubice: Institut Jana Pernera, 2007, 163 s. Encyklopedie Zoner Press. ISBN 978-80-86530-40-6.

15. PIJÁK, Michal. Je elektronické placení bezpečné?. In: *Měšec.cz: váš průvodce finančním světem* [online]. 2003 [cit. 2013-05-08]. ISSN 1213-4414. Dostupné z: http://www.mesec.cz/clanky/je-placeni-bezpecne/.

16. KRHOVJÁK, Jan, Marek KUMPOŠT a Václav MATYÁŠ. Útoky na platební systémy. In: *Zpravodaj ÚVT MU: bulletin pro zájemce o výpočetní techniku na Masarykově univerzitě* [online]. Brno: Ústav výpočení techniky MU, 2007 [cit. 2013-05-08]. ISSN 1212-0901. Dostupné z: http://www.ics.muni.cz/bulletin/articles/562.html.

17. Symetrická kryptografie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2006, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Symetrick%C3%A1_kryptografie.

18. Asymetrická kryptografie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2006, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Asymetrick%C3%A1_kryptografie.

19. Hašovací funkce. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2008, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Ha%C5%A1ovac%C3%AD_funkce#Kryptografick.C3.A1_ha.C5.A1ovac.C3.AD_funkce.

20. Elektronický podpis. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2005, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Elektronick%C3%BD_podpis.

21. Digitální certifikát. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2007, 2013 [cit. 2013-05-08]. Dostupné z: http://cs.wikipedia.org/wiki/Digit%C3%A1ln%C3%AD_certifik%C3%A1t.

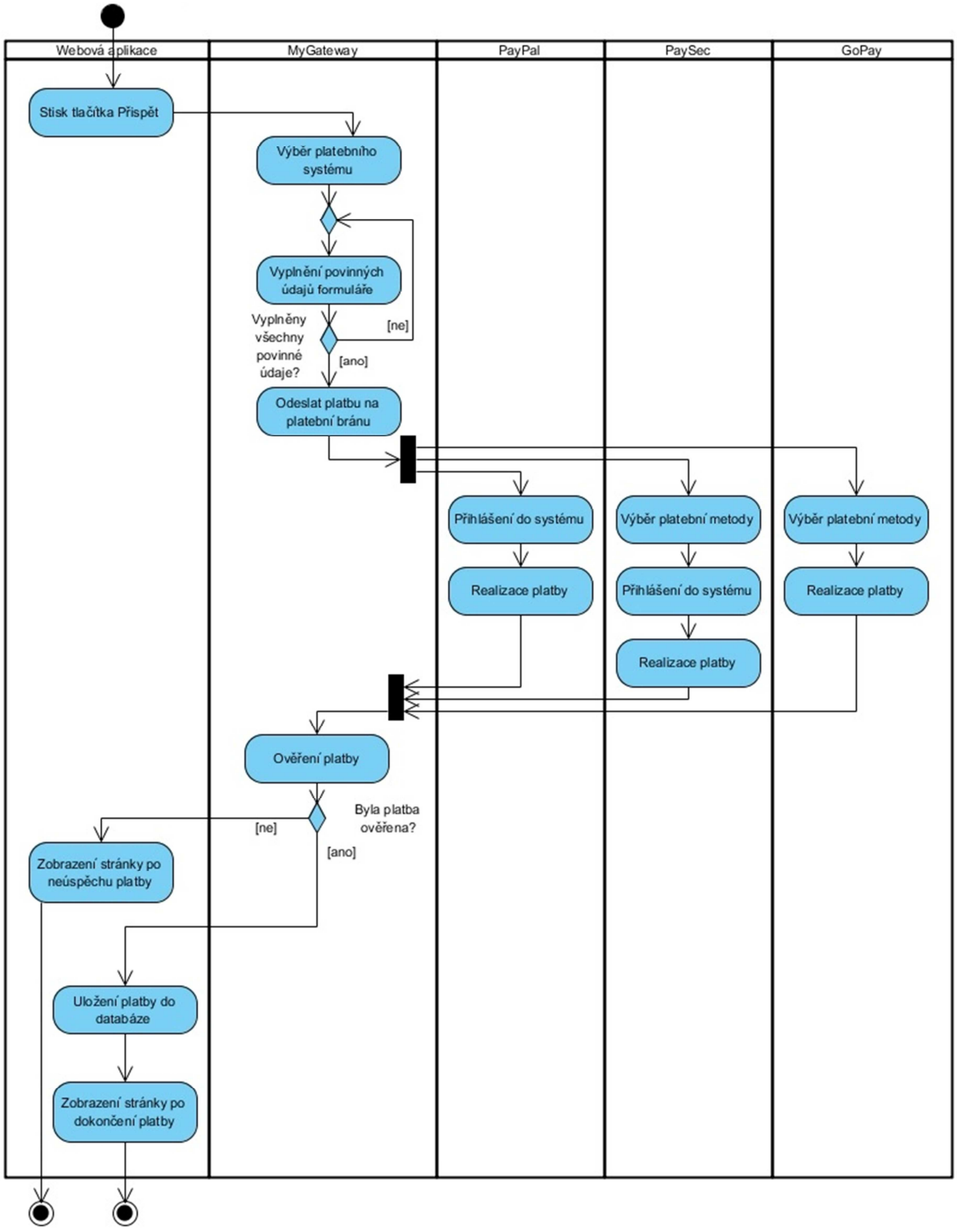
22. Protokoly pro elektronické platební systémy. In: *Security-Portal.cz* [online]. 2007 [cit. 2013-05-08]. Dostupné z: http://www.security-portal.cz/clanky/protokoly-pro-elektronick%25C3%25A9-platebn%25C3%25AD-syst%25C3%25A9my.

23. NAIR, Suku. Secure Electronic Transaction. *Southern Methodist University* [online]. 2005 [cit. 2013-05-08]. Dostupné z: lyle.smu.edu/~nair/courses/7349/SET.ppt.

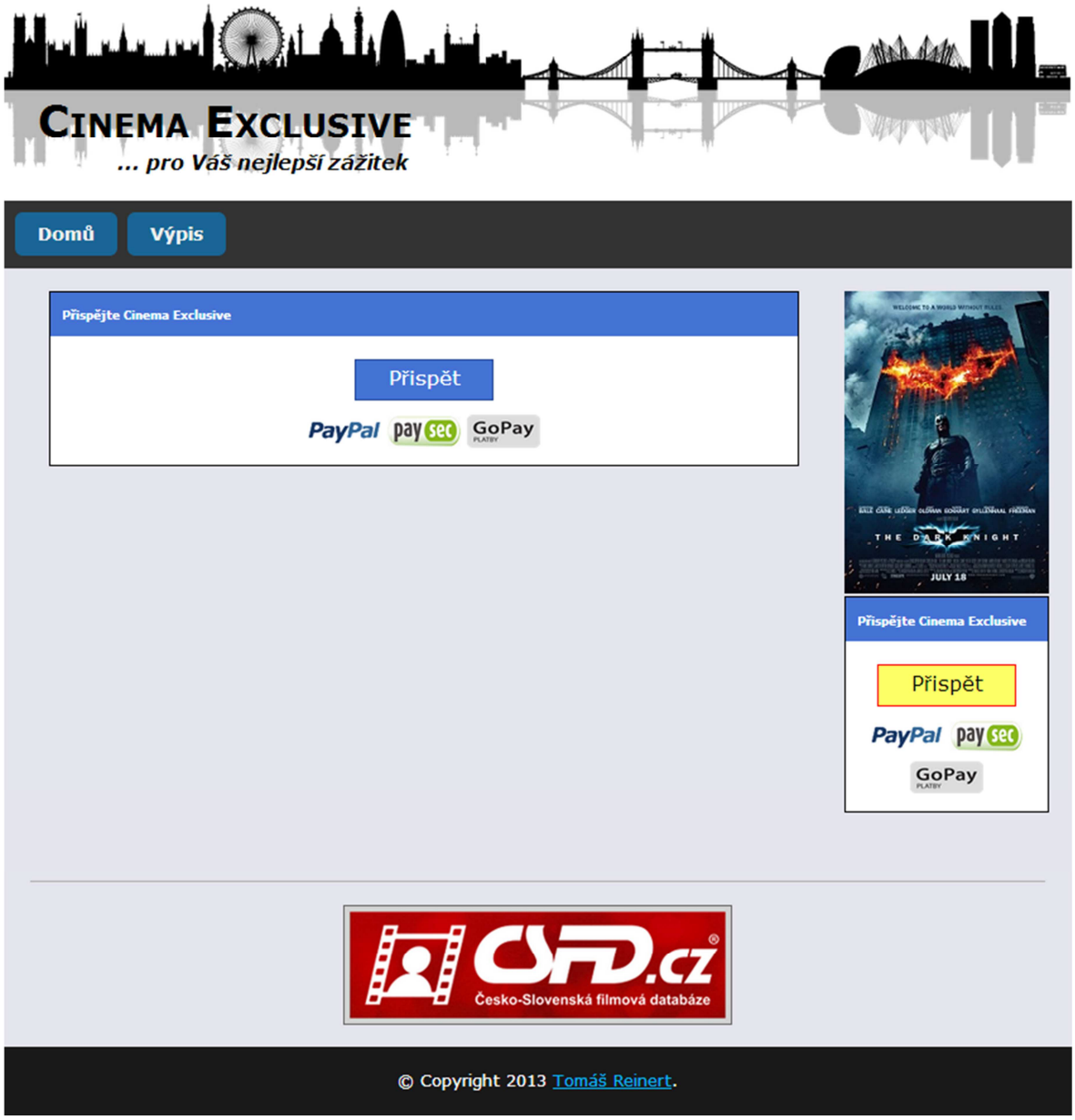
24. ČEGAN, Lukáš. *Správa webserveru*. (přednáška) Pardubice :Univerzita Pardubice, 2012.

25. 3-D Secure. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2013 [cit. 2013-05-08]. Dostupné z: http://en.wikipedia.org/wiki/3-D_Secure.
26. VETYŠKA, Jan. Platební metody na českém internetu. *Svaz obchodu a cestovního ruchu ČR* [online]. 2012 [cit. 2013-05-08]. Dostupné z: http://www.socr.cz/assets/zpravodajstvi/tiskove-zpravy/platebni-metody-na-ceskem-internetu-2012-final.pdf.
27. MORÁVEK, Daniel. Která platební metoda konečně svrhne nadvládu dobírky?. In: *Podnikatel.cz* [online]. 2011 [cit. 2013-05-08]. ISSN 1802-8012. Dostupné z: http://www.podnikatel.cz/clanky/platebni-metody-v-e-shopech/.
28. Platit při nakupování na internetu se dá i jinak než jen dobírkou. *IPodnikatel.cz: Portál pro podnikatele* [online]. 2011 [cit. 2013-05-08]. Dostupné z: http://www.ipodnikatel.cz/Internet/platit-pri-nakupovani-na-internetu-se-da-i-jinak-nez-jen-dobirkou.html.
29. JANU, Stanislav. Platební systémy na internetu – 1.část. In: *NETzin.cz* [online]. 2010 [cit. 2013-05-08]. Dostupné z: http://www.netzin.cz/2010/platebni-systemy-na-internetu-1-cast.php.
30. Fees. *PayPal* [online]. 1999, 2013 [cit. 2013-05-09]. Dostupné z: https://www.paypal.com/cz/fees.
31. Poplatky při dobití a výběru. *Moneybookers.com* [online]. 2012 [cit. 2013-05-09]. Dostupné z: https://www.moneybookers.com/app/help.pl?s=fees&fee_currency=CZK.
32. JANU, Stanislav. Platební systémy na internetu – 2.část. In: *NETzin.cz* [online]. 2010 [cit. 2013-05-08]. Dostupné z: http://www.netzin.cz/2010/platebni-systemy-na-internetu-2-cast.php.
33. Sazebník. *PaySec* [online]. 2011 [cit. 2013-05-09]. Dostupné z: http://www.paysec.cz/cmspage.aspx?id=feelist.
34. Sazebník poplatků. *GoPay* [online]. 2013 [cit. 2013-05-09]. Dostupné z: https://www.gopay.cz/podminky/sazebnik.

Příloha A – Activity diagram



Příloha B – Vzhled univerzálního platebního tlačítka



Příloha C – Platební formulář aplikace