

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Jakopič

Implementacija e-računov v spletno bančništvo

DIPLOMSKO DELO NA VISOKOŠOLSLEM STROKOVNEM
ŠTUDIJU

Mentor: prof. dr. Miha Mraz

Somentor: doc. dr. Iztok Lebar Bajec

Ljubljana, 2011



Št. naloge: 00126/2011

Datum: 16.05.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANDREJ JAKOPIČ**

Naslov: **IMPLEMENTACIJA E-RAČUNOV V SPLETNO BANČNIŠTVO
IMPLEMENTING E-INVOICES INTO ONLINE BANKING**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj v začetku svojega dela predstavi problematiko vpeljave e-računov v že obstoječe spletno bančništvo. V nadaljevanju dela naj kandidat predstavi razvojna orodja, s katerimi je možno implementirati e-račune v obstoječe rešitve in predstavi konkreten primer rešitve, ki je bila izdelana za eno od bank, ki posluje v Sloveniji.

Mentor:

prof. dr. Miha Mrz

Dekan:

prof. dr. Nikolaj Zimic

Somentor:

doc. dr. Iztok Lebar Bajec



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Andrej Jakopič,
z vpisno številko 63040055,

sem avtor diplomskega dela z naslovom:

Implementacija e-računov v spletno bančništvo

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
izr. prof. dr. Mihe Mraza
in somentorstvom
doc. dr. Iztoka Lebarja Bajca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.)
ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 2.5.2011

Podpis avtorja:

Zahvala

Rad bi se zahvalil mentorjuizr. prof. dr. Mihi Mrazu in somentorju doc. dr. Iztoku Lebarju Bajcu za pomoč pri izdelavi diplomskega dela, celotnemu kolektivu S&T Slovenija d.d. za izkazano zaupanje in družini ter prijateljem za podporo v času študija.

Kazalo

POVZETEK	1
ABSTRACT	3
1 UVOD	5
2 ELEKTRONSKO BANČNIŠTVO	7
2.1 ELEKTRONSKO POSLOVANJE	7
2.2 ELEKTRONSKO BANČNIŠTVO	12
2.3 SPLETNO BANČNIŠTVO	13
2.4 E-RAČUNI	16
2.4.1 <i>Sistem E-račun</i>	17
2.4.1.1 Vključitev prejelnika v sistem	20
2.4.1.2 Izdajanje računov	20
2.4.1.3 Izmenjavanje povratnih informacij	21
2.4.2 <i>Prednosti in slabosti udeležencev v sistemu e-račun</i>	21
3 UPORABLJENE TEHNOLOGIJE	23
3.1 HTML	23
3.1.1 <i>Razvoj</i>	23
3.1.2 <i>Sintaksa</i>	24
3.1.3 <i>Cascading Style Sheets (CSS)</i>	26
3.1.4 <i>JavaScript</i>	27
3.2 VISUAL BASIC	28
3.2.1 <i>Visual Basic 6</i>	29
3.2.2 <i>WebClasses in IIS aplikacije</i>	30
3.3 EXTENSIBLE MARKUP LANGUAGE (XML)	33
3.4 XML PODPIS	36
3.5 SPLETNE STORITVE	38

3.5.1	<i>Web Service Definition (Description) Language (WSDL)</i>	39
3.5.2	<i>Simple Object Access Protocol (SOAP)</i>	41
4	IMPLEMENTACIJA	43
4.1	NAROČILO NA E-RAČUNE	44
4.1.1	<i>Oddaja vloge za e-prijavo ali e-odjavo</i>	44
4.1.2	<i>Seznam izdajateljev</i>	51
4.1.3	<i>Pregled oddanih vlog</i>	51
4.2	PLAČILO E-RAČUNOV	52
4.2.1	<i>Plačilo e-računov</i>	52
4.2.2	<i>Plačani in ostali e-računi</i>	54
4.2.3	<i>Iskanje e-računov</i>	55
5	ZAKLJUČEK	57
	LITERATURA IN VIRI	59

Seznam kratic

- ASP** - skriptni jezik in tehnologija, ki omogoča razvoj razvoj dinamičnih spletnih strani na strani strežnika (angl. *Active Server Pages*)
- CSS** - prekrivni slogi (angl. *Cascading Style Sheets*)
- DOM** - način procesiranja XML dokumenta, ki objektni model dokumenta naloži v pomnilnik v drevesni obliki (angl. *Document Object Model*)
- HTML** - označevalni jezik za oblikovanje večpredstavnostnih dokumentov (angl. *Hyper Text Markup Language*)
- HTTP** - protokol za komunikacijo med odjemalcem in strežnikom preko mreže (angl. *Hyper Text Transfer Protocol*)
- IIS** - Microsoftov spletni strežnik (angl. *Internet Information Server/Services*)
- RIP** - računalniško izmenjavanje podatkov (angl. *Electronic Data Interchange - EDI*)
- SAX** - način procesiranja XML dokumentov, od začetka do konca po en element naenkrat (angl. *Simple API for XML*)
- SOAP** - protokol za kontaktiranje spletnih storitev (angl. *Simple Object Access Protocol*)
- VB** - programski jezik in razvojno okolje (angl. *Visual Basic*)
- WSDL** - standard za opisovanje spletnih storitev (angl. *Web Service Definition/Description Language*)
- XML** - razširljivi označevalni jezik (angl. *Extensible Markup Language*)
- XSLT** - deklarativni XML jezik uporabljen za transformacije XML dokumentov (angl. *Extensible Stylesheet Language Transformations*)

Povzetek

Cilj diplomskega dela je predstaviti e-račune in njihovo implementacijo v že obstoječe spletno bančništvo. E-računi uporabniku spletnega bančništva omogočijo boljšo preglednost prejetih računov in enostavnejše poravnavanje obveznosti, medtem ko izdajateljem zagotavljajo znižanje stroškov izdajanja računov, hitrejšo posredovanje računov in boljšo sledljivost le teh. Za doseg cilja smo vzpostavili povezavo s spletnimi storitvami ponujenimi s strani podjetja Halcom d.d. in tako v spletnem bančništvu za fizične osebe podprli nove funkcionalnosti, kot so prejem, pregled in plačevanje e-računov ter prijava in odjava na e-račune. Za uspešno komunikacijo s spletnimi storitvami smo oblikovali SOAP zahteve, ki so ustrezali WSDL predlogam, ki so opisovale spletne storitve. Podatke pridobljene s pomočjo spletnih storitev smo uporabniku primerno prikazali in mu omogočili dober pregled nad e-računi ter plačilo brez potrebe po ročnem izpolnjevanju plačilnih nalogov. Poleg plačila računov smo uporabnikom z avtomatskim vnosom potrebnih podatkov olajšali tudi prijavo za prejem e-računov.

Ključne besede:

e-računi, spletno bančništvo, spletne storitve

Abstract

In our thesis we focused on representing e-invoices and their implementation in to existing online banking system. E-invoices enable the user of online banking better overview of received invoices and easier payment. While for the issuer they lower costs of issuing an invoice, faster delivery and better overview of issued invoices. For successful implementation we used web services offered by Halcom d.d.. We offered users new functionalities like subscription to e-invoices and receipt, overview and payment of e-invoices. For successful communication with web services we designed SOAP requests that met the requirements given in WSDL templates that described the web services. We displayed the obtained data on user friendly forms for better overview and easier use. With automatic insertion of necessary data in forms we made subscription to e-invoices and their settlement easier for the users

Key words:

e-invoices, online banking, web services

1 Uvod

Elektronsko poslovanje se je z uveljavitvijo interneta in posledično zmanjšanjem stroškov za implementacijo le tega, razširilo na razna področja kot so trgovina, borzništvo, zavarovalništvo, storitve na zahtevo, ipd.

Eno izmed pglavitnih področij v elektronskem poslovanju je zagotovo elektronsko bančništvo (e-bančništvo). Slednje se je razvilo že mnogo pred internetom, saj so banke postavljala svoja omrežja in računalniške sisteme, da so poslovalnice lažje komunicirale med seboj. Elektronsko bančništvo se je za uporabnike razvilo nekoliko kasneje. Sprva so se pojavili bančni avtomati, nato telefonsko bančništvo, kateremu sta sledila spletno in mobilno bančništvo, ki sta trenutno najhitreje razvijajoča se sistema elektronskega bančništva.

Spletno bančništvo je doživelo hiter razvoj in razcvet. Tako je sedaj možno opraviti preko spleta že marsikatero bančno storitev in to kadarkoli in kjerkoli. K temu je pripomogel predvsem razvoj pametnih telefonov in dostopnost mobilnega interneta.

Banke so v spletno bančništvo integrirale skoraj vse storitve, ki jih ponujajo. Za izboljšanje spletnega bančništva so se morale povezati tudi z ostalimi gospodarskimi subjekti. Tako so sedaj zadnji hit e-bančništva tako imenovani e-računi, ki nadgrajujejo prejšnje brez-papirno izdajanje računov strankam podjetij. Prve variacije e-računov so bile poslana na elektronsko pošto uporabnika. Uporabnik je lahko potem poravnal račun preko spletne banke z ročnim vnosom ali celo preko plačilnega naloga na pošti ali bančnem okencu. Nova verzija elektronskega poslovanja na področju poravnanja stroškov usmeri račun v samo spletno bančništvo, od koder ga poravnamo z nekaj kliki. Ob tem nam ni potrebno vnesti nobenega polja na plačilnem obrazcu, tako da ne more priti do človeške napake.

E-računi so tako nov standard znotraj spletnega bančništva, ki ga je določilo Združenje Bank Slovenije (ZBS). Samo izvedbo omogočajo spletne storitve, ki jih je razvila družba Halcom d.d. in se imenujejo Hal E-invoices. Za uvedbo e-računov se je odločila tudi stranka podjetja S&T Slovenije d.d., Banka Sparkasse d.d., zato smo morali v tem podjetju v že obstoječe spletno bančništvo vzpostaviti podporo e-računom. Cilj je bil omogočiti uporabniku vse prednosti e-računov. Za doseg te ciljev smo morali implementirati nove funkcionalnosti kot so:

- pregled izdajateljjev e-računov,
- prijavo in odjavo na e-račune ter pregled prijav oz. odjav in
- prejemanje, pregled in plačilo e-računov.

Pri tem pa smo morali zagotoviti tudi preprosto uporabo, ki je glavni cilj spletnih aplikacij kot je e-bančništvo, saj je namenjeno široki populaciji uporabnikov spleta.

Za uvedbo e-računov smo oblikovali forme za prikaz podatkov pridobljenih s pomočjo ustreznih SOAP zahtevkov, ki smo jih poslali spletnim storitvam. Uporabnikom spletnega bančništva smo z avtomatskim vnosom obrazcev za plačevanje e-računov (obrazec BN 02 in posebna položnica) in prijavo na e-račune olajšali uporabo nove storitve.

V prvem poglavju diplomskega dela bomo spoznali elektronsko poslovanje in njegov razvoj z razširitvijo uporabe interneta in uveljavitev elektronskega poslovanja v bančništvu. Predstavili bomo tudi e-račune in sistem E-račun, ki omogoča poenotenje poslovanja z e-računi in prednosti njegove uporabe.

Diplomsko delo v nadaljevanju opisuje tehnologije uporabljene za uspešno implementacijo e-računov. V naslednjem poglavju predstavi potek implementacije in njene rezultate.

V samem zaključku diplomskega dela spoznamo probleme s katerimi smo se srečevali ob izdelovanju dopolnitve, njen doprinos in možnost izboljšav.

2 Elektronsko bančništvo

V tem poglavju bomo spoznali elektronsko poslovanje na splošno in kako se je le to razvilo, zakaj je prišlo do razvoja elektronskega bančništva in razloge za uvedbo spletnega bančništva in prednosti le tega.

2.1 Elektronsko poslovanje

Elektronsko poslovanje je z vse večjo uporabo interneta vedno bolj razširjena besedna zveza tudi v Sloveniji. V angleščini [1] se uporabljata dva različna izraza za elektronsko poslovanje in sicer

- e-commerce,
- e-business.

Pojem e-commerce se je pojavil prvi in lahko predpostavimo, da je e-business le njegova nadgradnja. E-commerce tako vsebuje predvsem e-trgovino, kjer so edini podatki, ki se izmenjajo o nabavi in prodaji blaga. E-business trgovino blaga nadgradi, tako da vpeljuje še druge storitve, ki so možne preko spleta, kot je npr. pomoč in svetovanje uporabnikom [2].

Slovenski izraz elektronsko poslovanje bi lahko definirali, kot vse poslovanje, ki se opravi v elektronski obliki z uporabo informacijske in komunikacijske tehnologije [3].

Razvoj elektronskega poslovanja se je začel že pred razširitvijo in popularizacijo interneta. Začetki segajo v sedemdeseta in osemdeseta leta, kot posledica sprotnega zagotavljanja proizvodnih virov (angl. *Just in Time*) in namenom povezovanja organizacij z dobavitelji. Tako se je razvilo pri nas imenovano računalniško izmenjavanje podatkov - RIP (angl. *Electronic Data Interchange - EDI*), ki je bilo namenjeno neposredni elektronski komunikaciji podjetij, ki so si tako izmenjavala naročila, račune in prometne dokumente (dobavnice, prevzemnice, izdajnice ...). V ta namen so morala podjetja razviti svoja omrežja (angl. *Value-Added Networks - VANs*), v katera so povezovala svoje računalniške sisteme. Problem takega poslovanja so bili seveda stroški izgradnje omrežja, razvoja programske opreme in pojav mnogih standardov komuniciranja, ki so se razlikovali od podjetja do podjetja. Zaradi teh problemov so si lahko RIP privoščila le velika podjetja oz. korporacije,

medtem ko so ostala manjša podjetja ostala na uveljavljenih izmenjavah podatkov (faks, pošta, telefon). To so tudi razlogi, da RIP verjetno ne bo nikoli popolnoma zaživel, kljub temu, da se po notranjih omrežjih pretakajo poslovne transakcije v vrednosti več kot 150 milijard ameriških dolarjev na leto. S pojavom interneta so tudi manjša podjetja dobila medij za elektronsko poslovanje, ki je bil finančno dostopen. Posledično so leta 2002 v raziskavah, ki jih je opravil IDC (angl. *International Data Corporation*) ocenili, da bo promet poslovnih transakcij opravljenih s pomočjo elektronskega poslovanja v nekaj letih dosegel vrednost 800 milijard ameriških dolarjev [3].

Internet se je izkazal za izredno primeren medij za e-poslovanje predvsem zaradi dostopnosti. Vsi uporabniki imajo enako tehnologijo, s katero se med seboj povezujejo. Posledično je trženje na internetu dobesedno sinonim za globalni prosti trg. Internet je tako zabrisal tradicionalne prostorske in časovne meje v poslovnem svetu, saj se podjetja vedno bolj približujejo potrebam uporabnikov oz. potrošnikov, saj imajo sedaj konkurenčnega ponudnika le nekaj klikov stran. Posledično lahko rečemo, da je e-business za potrošnika v resnici »*me-business*«, saj lahko sam izbere čas, prostor in vrsto storitve [1]. Prednosti uporabe interneta pred zasebno zgrajenimi omrežji so prikazane v tabeli 1.

Tabela 1: Razlika med RIP in poslovanjem preko interneta [3].

RIP	Internet
Znani poslovni partnerji	Neznani poslovni partnerji
Zaprta trg	Odprta trg
Majhen obseg transakcij	Velik obseg transakcij
Visoka vrednost transakcij	Majhna vrednost transakcij
Visoka stopnja varnosti	Spremenljiva stopnja varnosti
Togost	Prilagodljivost

Z razvojem elektronskega poslovanja je nastalo veliko oblik le tega, kot so:

- elektronsko trgovanje,
- elektronsko bančništvo,
- elektronski finančni prenosi,
- elektronsko borzništvo,

- storitve na zahtevo,
- informacijski portali,
- elektronsko založništvo,
- elektronsko zavarovalništvo,
- delo na daljavo,
- e-poslovanje državnih in javnih služb,
- e-izobraževanje.

Vendar pa nikoli ne moremo naštetih vseh oblik, saj dnevno nastajajo nove [1]. V elektronskem poslovanju imamo lahko 3 različne udeležence in sicer:

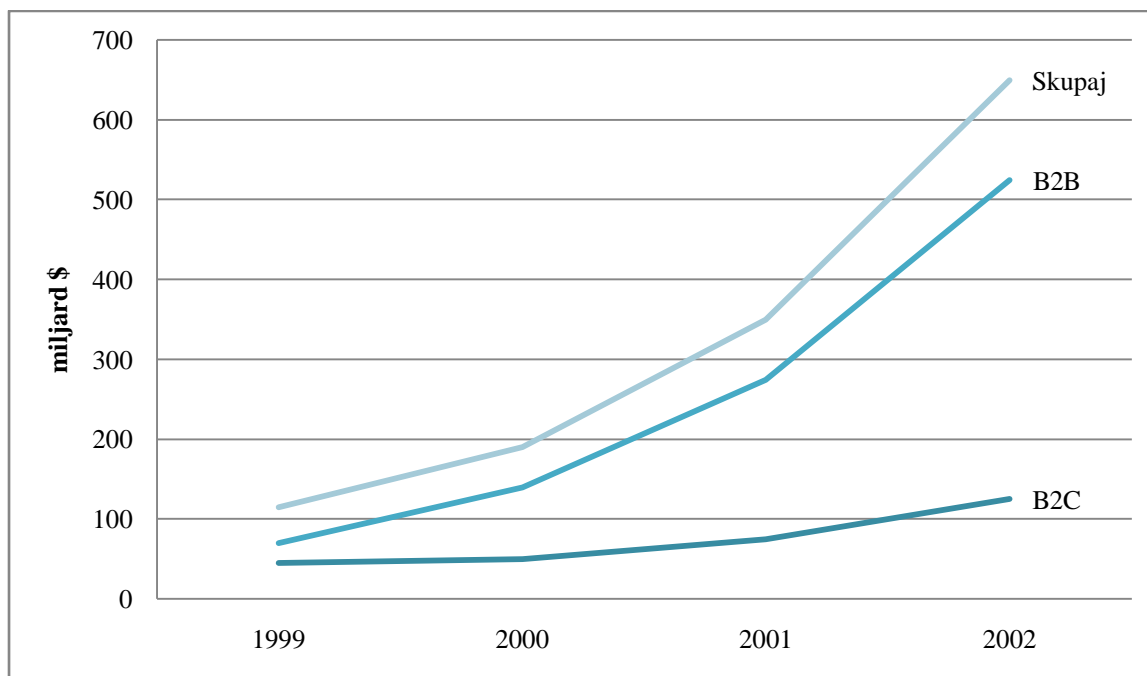
- posameznika, potrošnika ali državljana (angl. *customer, citizen* - C),
- podjetje ali ustanovo (angl. *business* - B),
- državo oz. državno upravo (angl. *government* - G) [1].

Tako lahko glede na interakcijo različnih udeležencev elektronsko poslovanje ločimo na naslednje skupine:

- med podjetji (B2B): elektronsko bančništvo za pravne osebe, povezave med podjetjem in njegovimi dobavitelji oz. podjetjem in njegovimi naročniki, sodelovanje v virtualnih organizacijah,
- med podjetji in potrošniki (B2C): elektronsko bančništvo za fizične osebe, elektronsko trgovanje, izobraževanje na daljavo, itd.,
- med potrošniki (C2C): forumi, dražbe, itd.,
- med podjetji in javno oz. državno upravo (B2G): izvajanje javnih razpisov, oddajanje napovedi za odmero DDV, itd.,
- med državljani in javno oz. državno upravo (C2G): oddaja napovedi za odmero dohodnine, izdaja izpiska iz rojstne matične knjige, itd.,
- znotraj javne oz. državne uprave (G2G): spremljanje podatkov o realizaciji proračuna Republike Slovenije [1], itd.

Zmotno je prepričanje, da je B2C najpomembnejša vrsta e-poslovanja. To prepričanje izvira iz vsakdanjega srečevanja z njim in posledično s poznavanjem le tega. Če pogledamo s strani

vrednosti transakcij (glej sliko 1) je najpomembnejša oblika B2B, ki je tako kot B2C dobil dodaten zagon zaradi razširitve interneta, saj so se lahko v elektronsko poslovanje vpletla tudi srednje velika in majhna podjetja.



Slika 1: Vrednost elektronskega poslovanja v svetovnem merilu [3].

Glavni razlogi za uvedbo e-poslovanja ne glede na obliko poslovanja ali udeležence so naslednji:

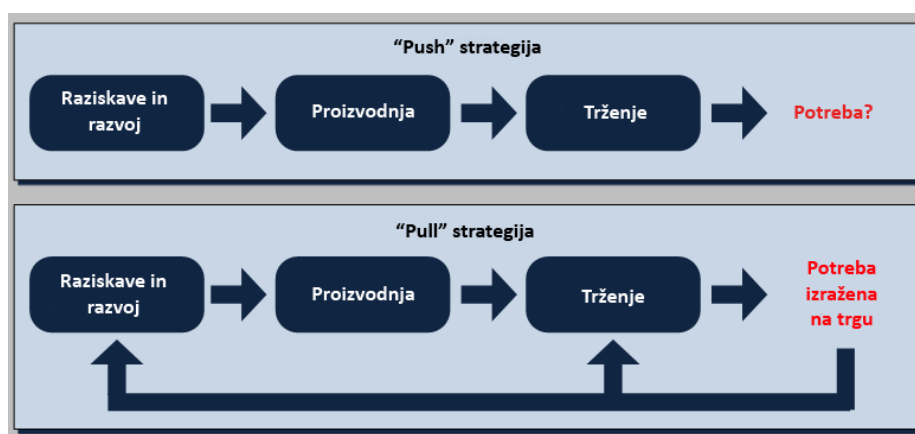
- zniževanje stroškov nakupa: gre predvsem za zmanjšanje stroškov na področju izvajanja procesa ter izdelave in pošiljanja; ocene kažejo, da so podjetja privarčevala 5-10% ob uporabi RIP sistema z uvedbo in uporabo interneta privarčevala še dodatnih 10%.);
- zniževanje obsega zalog: omogoča nam znižanje stroškov skladiščenja;
- skrajševanje poslovnega cikla: le to je posledica lažjega komuniciranja s stranko in hitrejšega dostavljanja dokumentov;
- razvoj učinkovitejše pomoči uporabnikom ter povezovanje z odjemalci,
- zniževanje stroškov prodaje in trženja;
- ustvarjanje novih tržnih priložnosti oz. majhen začetni kapital [3].

Z elektronskim poslovanjem se je razvil tudi poslovni model, ki je bil prirejen temu načinu poslovanja. Prednosti si lahko ogledamo v tabeli 2.

Tabela 2: Razlike med tradicionalnim in e-poslovnim modelom [3].

Tradicionalni poslovni model	E-poslovni model
Serijska proizvodnja.	Izdelek, prilagojen strankinim zahtevam.
Stranka ima majhno izbiro izdelkov.	Stranka sodeluje pri ustvarjanju izdelka.
Dobavni časi so dolgi.	Kratek dobavni čas.
Netočne dobave.	Velika zanesljivost dobave.
Visoke varnostne zaloge.	Majhen obseg zalog.
Stroški celotne oskrbovalne verige so visoki.	Stroškovno učinkovita oskrbovalna veriga.
Sistem »push« proizvodnje.	Sistem »pull« proizvodnje izhaja iz strankinih zahtev.
Razvoj novega izdelka traja zelo dolgo.	Znatno skrajšanje razvoja izdelkov.
Nezadovoljstvo strank, visoki stroški oskrbovalne verige.	Nižji stroški, zadovoljne stranke.
Cena > vrednost produkta.	Cena = vrednost produkta.

Razlika nastane predvsem zaradi uporabe »push« strategije (glej sliko 2) v tradicionalnem poslovnem modelu, medtem ko v e-poslovnem modelu večinoma uporabljamo »pull« strategijo (glej sliko 2). Seveda pa se lahko v obeh poslovnih modelih uporablja kombinacija obeh strategij, kar se smatra za najbolj učinkovito.



Slika 2: »Push« in »pull« strategija [4].

2.2 Elektronsko bančništvo

Med elektronsko bančništvo lahko štejemo vse bančne storitve, pri katerih komitent uporabi vmesnik in komunikacijsko omrežje in nato banka za potrditev o opravljeni storitvi uporabi svoj vmesnik in pošlje potrdilo preko komunikacijskega omrežja do komitenta (glej sliko 3).

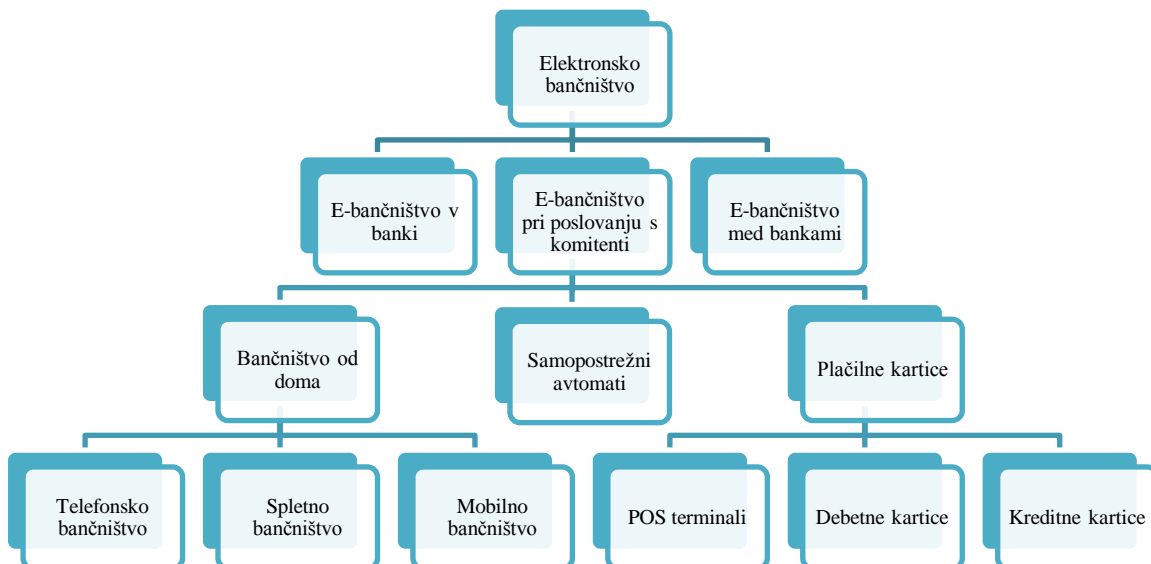


Slika 3: Princip delovanja elektronskega bančništva [5].

Elektronsko bančništvo lahko v grobem razdelimo na:

- elektronsko bančništvo v banki,
- elektronsko bančništvo med bankami in
- elektronsko bančništvo pri poslovanju s komitenti oz. strankami.

Natančnejša razdelitev elektronskega bančništva je prikazana na sliki 4.

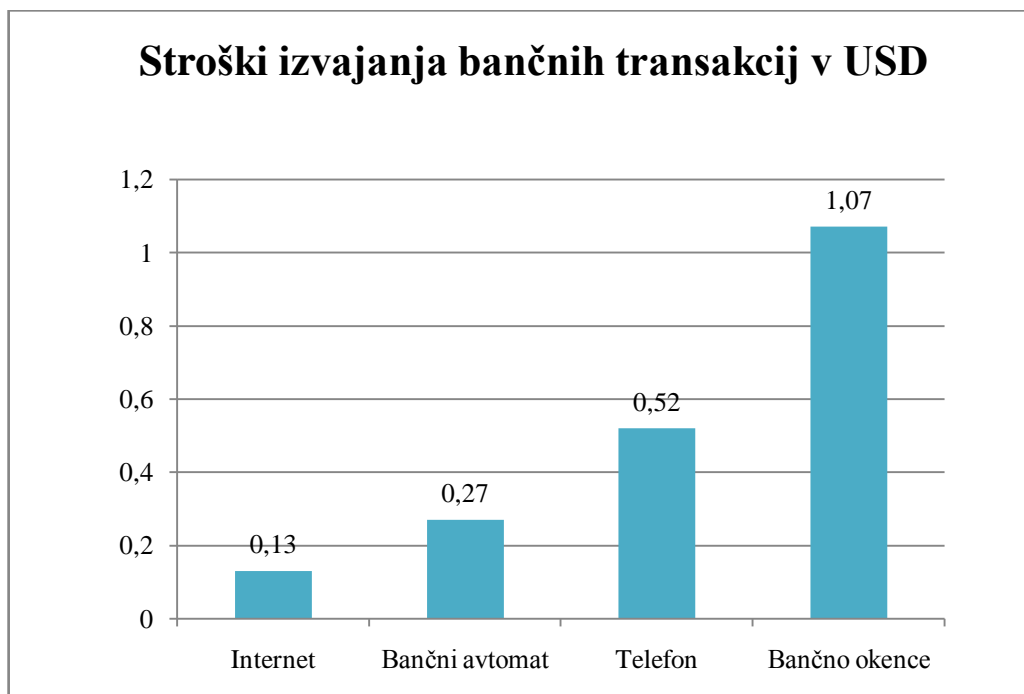


Slika 4: Oblike elektronskega bančništva [5].

Elektronsko bančništvo je bilo s strani bank uvedeno zaradi

- zmanjšanja stroškov (glej sliko 5),

- bančniki se lahko posvetijo zahtevnejšim bančnim storitvam (kreditu itd.),
- lažje dostopnosti strank do banke in posledično večje ciljne skupine.



Slika 5: Stroški banke na transakcijo ob različnih oblikah poslovanja [6].

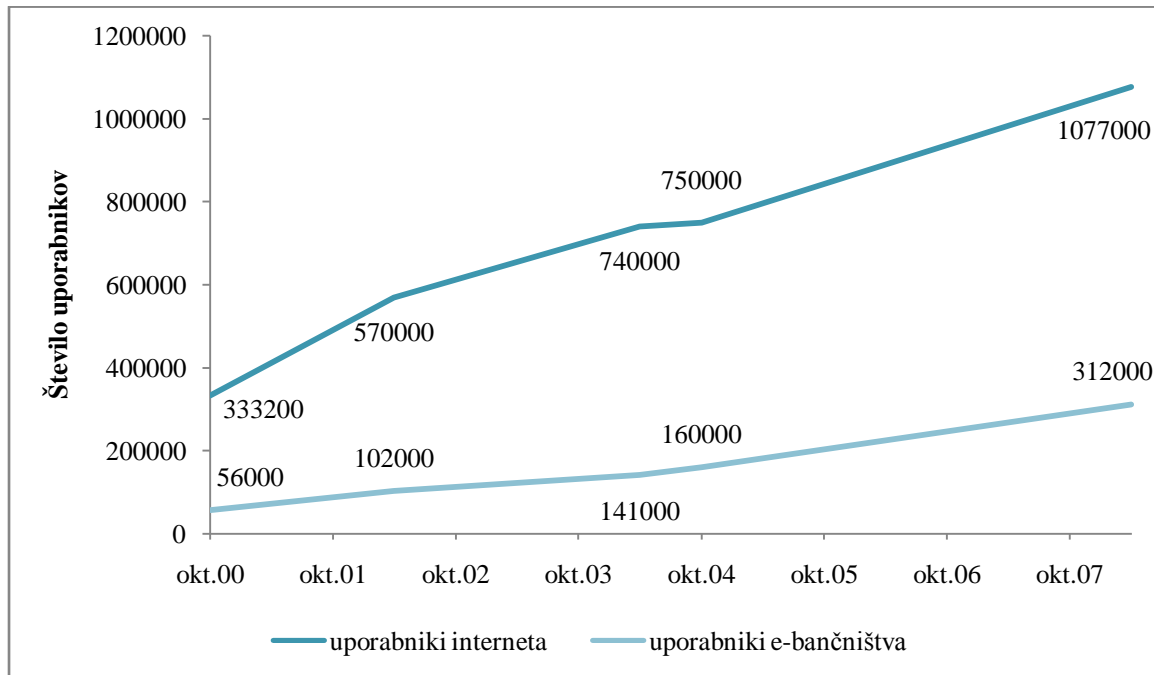
Razvoj elektronskega bančništva se je začel že pred razširitvijo interneta z bankomati in telefoni.

2.3 Spletno bančništvo

Pojma elektronsko in spletno ali internetno bančništvo se velikokrat enačita. Vendar je e-bančništvo širši pojem slednjega, saj vsebuje že prej naštetе vrste (bančni avtomati, plačilne kartice, telefonsko bančništvo in mobilno bančništvo) [7].

Spletno bančništvo pridobiva na priljubljenosti predvsem zaradi dostopa do banke iz domačega naslanjača 365 dni v letu, 24 ur na dan. Prednost ponuja tudi cena same storitve, saj za uporabo ni potrebna posebna programska oprema, ampak le spletni brskalnik in dostop do interneta. Razvoj mobilnih telefonov nam ponuja novo tržno nišo, kombinacijo mobilnega in spletnega bančništva (spletno bančništvo na mobilnih telefonih), tako da imamo banko na dosegu roke kjerkoli smo.

Največji problem spletnega bančništva, predvsem začetnikov na spletu, je nepoznavanje tehnologije uporabnika in posledično nezaupanje v varnost. Vendar z vse boljšo izobraženostjo uporabnikov tudi na področju spletnega poslovanja narašča zaupanje v varnost elektronskega poslovanja. Tako se odraža tudi naraščanje uporabnikov spletnega bančništva, saj je v letu 2007 po podatkih RIS-a (projekt Raba interneta v Sloveniji) spletno bančništvo uporabljalo že skoraj 30% vseh aktivnih uporabnikov interneta (glej sliko 6).



Slika 6: Rast uporabe e-bančništva [8].

Spletno bančništvo v Sloveniji ponujajo že skoraj vse banke. Zaradi vse večje konkurence in majhnih stroškov ob transakcijah se banke odločajo za vedno večji razvoj spletnega bančništva. Z razvojem je sedaj preko spleta možno opraviti že marsikatero bančno storitev. Tipični primeri tovrstnih storitev so:

- pregled prometa na osebnih računih in pregled izpiskov,
- različne bančne transakcije (nakazila v državi, možnost plačila z valuto vnaprej, naročilo deviznega nakazila, prenos sredstev med podračuni osebnega računa, čezmejno plačilo itd.),
- odprtje, zaprtje in spremembe trajnih nalogov ter direktnih obremenitev,
- naročilo za spremembo limita na osebnem računu ali kreditni kartici,
- odprtje depozita ali varčevalnega računa,

- vpogled v odprte depozite, kredite in varčevalne račune ter spreminjanje lastnosti teh poslov (npr. podaljšanje depozita),
- plačevanje spletnih nakupov in
- prejemanje e-računov.

Naslednje trendi v izdelavi spletnih storitev bodo vplivali tudi na e-bančništvo:

- »customization« oz. prilagajanje: gre predvsem za prilagodljivost uporabniku in njegovim željam ter potrebam; prilagodljive imamo lahko le grafične vmesnike oz. spletne strani, lahko pa tudi same spletne aplikacije;
- »security« oz. varnost: poudarek na varnem poslovanju in posledično občutek varnosti uporabnika;
- »social media« oz. trženje preko dobrega glasu: izkoriščanje ponudbe tehnologije Web 2.0.

Razvoj e-bančništva bo šel v smeri prilagodljivosti stranki. Tako bo potrebna raznovrstnost storitev, ki bodo usmerjene k različnim ciljnim skupinam. Te storitve je potrebno prilagajati ciljni skupini, ki jih najpogosteje uporablja. Šele nato v razvoju teh storitev gledamo na bodoče potencialne uporabnike [8].

Uvedba spletnega bančništva prinese banki veliko prednosti, kot so:

- nižji stroški bančnih transakcij,
- prihranek pri obvestilu o stanju na računih,
- več potencialnih strank,
- manjše število potrebnih poslovalnic,
- razbremenitev bančnih uslužbencev,
- večjo kontrolo poslovanja in preglednost,
- večjo kakovost poslovanja in kakovostnejše storitve,
- večjo ažurnost in informiranost o poslovanju komitentov,
- spremljanje uporabe,
- povečanje ponudbe ostalih finančnih storitev.

Prednosti, ki jih pridobijo komitenti z uporabo spletnega bančništva, so sledeče:

- izognitev gneči in dolgim čakalnim vrstam,
- možnost dostopa iz službe in doma,
- prihranek časa,
- možnost opravljanja storitev ob katerem koli času,
- nižja cena bančnih storitev,
- večja informiranost,
- večja preglednost nad stanjem in izdatki,
- enostavnost uporabe,
- zasebnost.

Vendar se ob številnih prednostnih pojavljajo tudi slabosti in problemi za banke, ki ponujajo spletno bančništvo. Ti so sledeči:

- visoki stroški uvedbe,
- neprimernost spletnega bančništva za vse komitente,
- možnost tehničnih napak in nezanesljivosti sistema,
- varovanje podatkov.

S strani komitenta je največji problem v nepoznavanju spletnih tehnologij in posledično neprimerni uporabi spletnega bančništva, kjer se nato pojavlja problem varnosti poslovanja na spletu [5].

2.4 E-računi

E-račun je elektronska oblika računa, ki ga izdajatelj računa izda svojemu dolžniku oz. prejemniku računa. Namen izdaje e-računa je doseči prihranek pri manipulativnih in poštnih stroških ter poenostavitev obstoječih procesov izdajanja in prejemanja računov v podjetjih in prejema pri fizičnih osebah. Po svetu so vpeljani različni poslovni modeli e-računov, ki so dosegli pričakovano kritično maso uporabnikov in prometa. Značilnost vseh uspešnih sistemov e-računov je, da so nosilno vlogo prevzele banke oz. njihova infrastruktura, tako pri razvoju kot tudi pri procesiranju [9].

Osnova za razvoj poslovnega modela e-računa so standardi in ustrezna zakonodaja, ki izenačuje veljavnost klasičnih in elektronskih dokumentov. Pravne osnove za razvoj sistemov za procesiranje e-računov so definirane v okviru posameznih držav, vendar je na ravni EU napovedan intenziven razvoj učinkovitega poslovnega modela za e-račune v okviru projekta SEPA (Single Euro Payments Area). Znotraj EU se razvija tudi EEI (European Electronic Invoicing) direktiva, ki bo oblikovala koncept e-računov in bi predvidoma podjetjem prihranila 64,5 milijarde EUR na leto [10].

Z namenom širše uveljavitve e-poslovanja so se v okvirju projekta e-SLOG pri Gospodarski zbornici Slovenije (GZS) oblikovali standardni elektronski poslovni dokumenti, med katerimi je tudi e-račun. Uporaba e-računov je bila prisotna in omejena na večje družbe, ki so se elektronsko povezovale s svojimi partnerji. Podatke so si izmenjevala podjetja s pomočjo različnih tehnoloških rešitev, ki so bile integrirane znotraj informacijskih sistemov. Zaradi želje po razširitvi baze uporabnikov, tako na manjše pravne subjekte, kot tudi na fizične osebe, so se banke preko Združenja Bank Slovenije organizirale in ponudile novo storitev upravljanja z e-računi. Tako se je znotraj Združenja Bank Slovenije oblikovala delovna skupina, katere namen je bil vzpostaviti enotno skupno infrastrukturo (standard, varnost, pravila) za procesiranje e-računov [9].

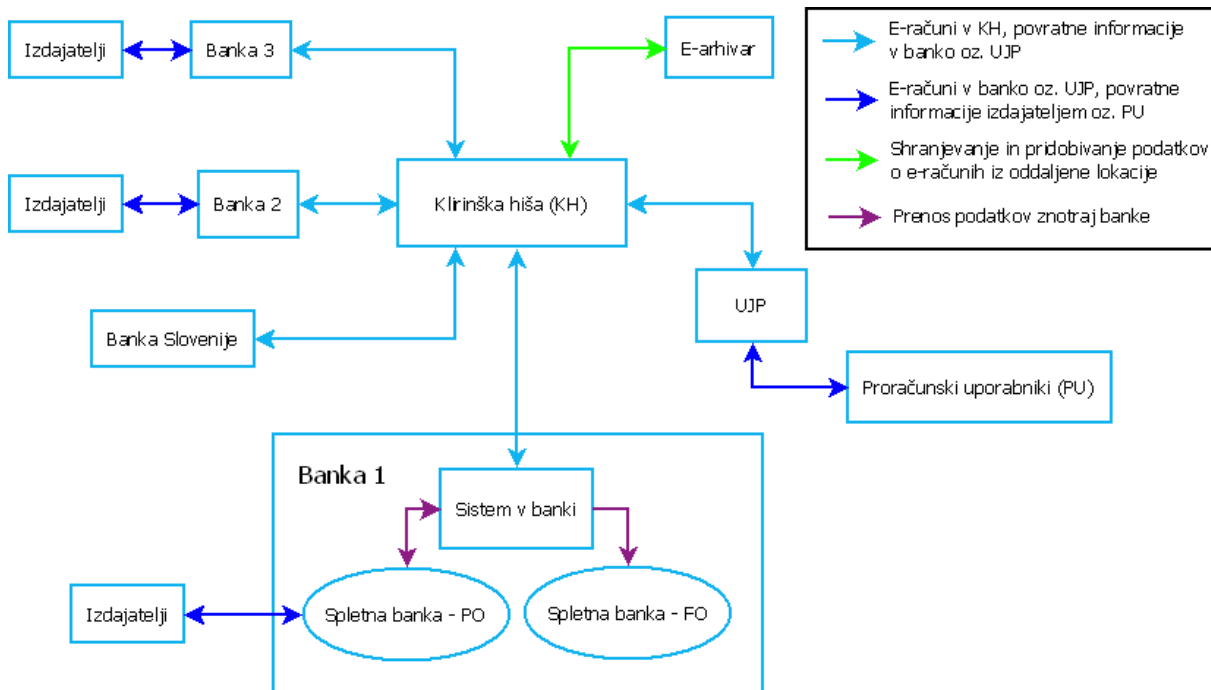
2.4.1 Sistem E-račun

Cilj uvedbe e-računov je znižanje stroškov (ni poštne, ni ročnega dela, krajši čas za izdajo in prejemanje računov) in poenostavitev (ni napak pri plačilih, hitrejša izvedba plačil, avtomatsko zapiranje računov) izmenjave računov med upniki in dolžniki oz. med izdajatelji in plačniki.

Na sliki 7 je predstavljena shema sistema E-račun, ki je sestavljen iz naslednjih udeležencev:

- banke izdajatelja,
- banke prejemnika,
- centralnega posrednika - procesor (klirinška hiša),
- izdajatelja: kot izdajatelji nastopajo pravne osebe, tako mala kot velika podjetja in zasebniki,
- prejemnika e-računa, pravne in fizične osebe,

- e-arhivar-ja,
- Banke Slovenije,
- Uprave Republike Slovenije za javna plačila,
- proračunskega uporabnika.



Slika 7: Shema sistema E-račun [9].

Združenje bank Slovenije pa je predvidevalo naslednje korake za uspešno delovanje sistema e-računov:

- sklenitev medbančnega dogovora,
- pristopna izjava za izmenjavo e-računov v sistemu E-račun,
- vključitev banke v sistem E-račun,
- vključitev izdajatelja v sistem,
- vključitev prejemnika (pravna oseba) v sistem,
- vključitev prejemnika (fizična oseba) v sistem,
- izdajanje e-računov,
- izmenjava povratnih informacij,
- zavrnitev e-računa in

- arhiviranje e-računov.

Tabela 3: Udeleženci v posameznem koraku v sistemu E-račun [9].

Korak v procesu/ SODELUJOČI UDELEŽENCI	BANKA IZDAJATELJA	BANKA PREJEMNIKA	CENTRALNI POSREDNIK - PROCESSOR	IZDAJAT ELJ	PREJEMNIK E-RAČUNA (PRAVNA OSEBA)	PREJEMNIK E-RAČUNA (FIZIČNA OSEBA)	E- ARHIVAR
sklenitev medbančnega dogovora	x	x					
pristopna izjava za izmenjavo e- računov v sistem	x	x	x				
vklučitev banke v sistem			x				
vklučitev izdajatelja v sistem	x	x	x	x			
vklučitev prejemnika (PO) v sistem		x	x	x	x		
vklučitev prejemnika (FO) v sistem		x		x		x	
izdjanje e- računov	x	x	x	x			
izmenjavanje povratnih informacij	x	x	x	x			
zavrnitev e- računa	x	x	x	x	x	x	
arhiviranje e- računov	x	x	x	x	x		x

Tabela 3 prikazuje kateri udeleženci sistema E-račun sodelujejo v posameznem koraku v procesu sistema E-račun.

Vsaka banka, ki se vključi v sistem E-račun, lahko postane aktivna ali pasivna članica sistema. Komitenti aktivne banke lahko izdajajo in prejemajo e-račune, medtem ko lahko komitenti pasivne banke le prejemajo e-račune [9].

V diplomski nalogi se bomo osredotočili na procese izdajanja e-računov, izmenjavanja povratnih informacij in vključitev prejemnika v sistem, ki jih bomo implementirali v spletno bančništvo.

2.4.1.1 Vključitev prejemnika v sistem

Pri procesu vključitve prejemnika v sistem sodelujejo:

- *prejemnik e-računa*, ki je pravna ali fizična oseba; slednji mora z oddajo vloge izraziti željo za prejemanje e-računov; prejemnik to lahko stori pri samem izdajatelju ali pa preko elektronske banke; če je vloga oddana preko elektronskega kanala, mora biti le-ta tudi ustrezno elektronsko podpisana;
- *banka prejemnika e-računov*, ki bodočemu prejemniku omogoča oddajo vloge za prijavo na e-račune, ne glede na to, ali je banka aktivna, ali pasivna članica;
- *centralni posrednik - procesor*, ki omogoči posredovanje pravih e-vlog med obema bankama;
- *banka izdajatelja e-računov*, ki omogoči izdajatelju e-računa sprejem e-vloge;
- *izdajatelj e-računa*, ki sprejema prijave preko svoje e-banke, svojih spletnih portalov, ali v svojih poslovalnicah in jih zapisuje v svoj interni register prejemnikov e-računov.

Enak postopek, kot ob vključitvi (prijavi) prejemnika v sistem, velja tudi ob izključitvi (odjavi) prejemnika iz sistema.

2.4.1.2 Izdajanje računov

V samem procesu izdajanja e-računov sodelujejo:

- *izdajatelj e-računov*, ki oblikuje standardizirano ovojnico in e-račun, ki ga posreduje svoji banki;
- *banka izdajatelja*, ki shrani vse izdane e-račune v svoj sistem in jih nato ustrezno obdela glede na IBAN prejemnika e-računa; če je le ta znotraj banke izdajatelja, banka posreduje račun prejemniku in izdajatelja obvesti o statusu izdanega računa; če je prejemnikov račun izven banke, le-ta posreduje e-račun centralnemu posredniku-procesorju;
- *centralni posrednik-procesor*, prevzame e-račun od banke izdajatelja in ga posreduje banki prejemnika; ob tem CPP preveri tudi obstoj bank v registru bank in ob neobstoju banke prejemnika zavrne e-račun;
- *banka prejemnika e-računa* prejme e-račun in ga ustrezno predstavi; ob tem mora izdajatelju e-računa posredovati status dostave;

- *prejemnik e-računov* prejme, pregleda in plača e-račun; neustrezne račune lahko tudi zavrne preko alternativnih poti (telefon, elektronska pošta, poslovalnica); v prihodnosti bi to funkcijo vgradili v same e-banke.

2.4.1.3 Izmenjavanje povratnih informacij

V procesu izmenjave povratnih informacij sodelujejo:

- *banka prejemnika e-računa*: povratno informacijo obvezno pošlje ob neuspešni dostavi e-računa, opcijsko pa lahko pošlje povratno informacijo tudi ob uspešni dostavi; povratno informacijo banka pošlje direktno izdajatelju, če je le ta njen komitent, drugače pa centralnemu posredniku-procesorju;
- *centralni posrednik-procesor*: posreduje prejete povratne informacije;
- *banka izdajatelja e-računa*: posreduje povratno informacijo od centralnega posrednika-procesorja do samega izdajalca e-računa.

Povratna informacija mora biti oblikovana po vnaprej dogovorjenem standardu, tako da je možno iz nje razbrati, kje je prišlo do napake. Povratne informacije lahko grupiramo v tri večje skupine:

- *negativna vsebinska povratna informacija* se pošlje, ko e-račun ni bil dostavljen zaradi vsebinskih napak;
- *negativna sistemska povratna informacija* se pošlje, ko pride do napake v sistemu; informacija pove v katerem podsistemu je prišlo do napake in številko napake;
- *pozitivna povratna informacija* se pošlje na posebno zahtevo izdajatelja e-računa s strani banke prejemnika e-računa ob uspešni dostavi računa.

2.4.2 Prednosti in slabosti udeležencev v sistemu e-račun

Prednosti izdajateljev ob uvedbi sistema e-račun so:

- znižanje stroškov izdaje papirnega računa,
- hitrejšo posredovanje računov,
- lažja sledljivost računov,

- boljše poravnavanje obveznosti prejemnikov,
- avtomatizacija procesov administriranja izdanih/prejetih računov.

Prejemniki računov, ki se odločijo za uporabo e-računa, imajo naslednje prednosti:

- boljša preglednost prejetih računov,
- avtomatizacija celotnega sprejema in računovodskega evidentiranja računov,
- znižanje stroškov zaradi avtomatizacije vnosa podatkov za računovodsko evidentiranje,
- enostavnejše poravnavanje obveznosti.

Poslovne banke z uvedbo sistema e-račun pridobijo:

- nov posel,
- nov prihodek,
- povečanje tržnega deleža,
- širitev dejavnosti.

Kljub številnim prednostim sistema e-račun se predvsem na začetku pojavljajo slabosti sistema. Slabosti se kažejo predvsem na strani izdajatelja in bank, ki imajo visoke stroške ob vpeljavi novega načina poslovanja v svoje informacijske sisteme in izobraževanja kadra za uporabo le tega. Vendar gre pri začetni investiciji predvsem za visok enkratni izdatek, ki se skozi kasnejše prihranke ob dolgoročni uporabi lahko nadomesti.

3 Uporabljene tehnologije

Sledijo predstavitve tehnologij, ki smo jih uporabili za implementacijo e-računov v že obstoječe spletno bančništvo.

3.1 HTML

HTML ali HyperText Markup Language je označevalni jezik, ki je trenutno dominanten na področju spletnih strani [11]. HTML dokument sestavljajo HTML elementi (v nadaljevanju elementi), ki jih opisujejo oznake med lomljenimi oklepaji (npr. `<html>`). Slednji povedo spletnemu brskalniku kako naj interpretira vsebino med njimi, samih oznak pa spletni brskalniki ne prikazujejo.

Ker se je HTML kmalu po njegovi predstavitvi hitro prijel kot vodilni spletni jezik, se je na začetku razvijal brez smernic in posledično se je jeziku, ki je bil sprva zamišljen kot semantični jezik, dodalo veliko predstavitvenih elementov, ki na začetku niso bili predvideni. To problematiko je začel reševati World Wide Web Consortium (W3C), ki je želel HTML popeljati na njegove začetke. Posledično je za predstavitveno plat razvil dva nova programska jezika CSS (angl. *Cascading Style Sheets*) in XSL (angl. *Extensible Stylesheet Language*), ki pa je namenjen le predstavitvi XML dokumentov.

Poleg ločitve pomenske in predstavitvene plasti se je začela pojavljati tudi potreba po dinamičnosti spletnih strani. Zaradi te težnje so se razvili skriptni jeziki kot sta JavaScript in VBScript, ki jih je možno implementirati v HTML dokumente.

3.1.1 Razvoj

Začetki HTML-ja segajo v začetek 90. let prejšnjega stoletja. Njegov izumitelj je bil Tim Berners-Lee, ki je prvi javno objavil 20 HTML oznak, ki so bile zelo podobne takratnemu SGMLguidu. SGMLguid je bil razvit v CERNU (angl. *European Organization for Nuclear Research*), kot interni SGML (angl. *Standard Generalized Markup Language*) za oblikovanje dokumentov [11].

V letu 1995 je delovna skupina za HTML, ki jo je sestavil IETF (angl. *Internet Engineering*

Task Force), predstavila prvo uradno specifikacijo HTML-ja, ki se je imenovala HTML 2.0 in je bila sestavljena iz obeh prejšnjih predlogov »Hypertext Markup Language (HTML)«, ki sta jo napisala Berners-Lee in Dan Connolly ter »HTML+ (Hypertext Markup Format)«, ki ga je sestavil Dave Raggett. Nadaljni razvoj je bil zavlečen vse do leta 1996 zaradi nasprotnih interesov znotraj IETF. Od leta 1996 je razvoj in vzdrževanje specifikacij prevzel World Wide Web Consortium (W3C), v letu 2000 pa je HTML postal tudi mednarodni standard (ISO/IEC 15445:2000) [11].

Trenutno zadnja objavljena specifikacija s strani W3C je »HTML 4.01 Recommendation«, ki je bila objavljena konec leta 1999, ki jo poznamo v treh različnih verzijah:

- »Strict«, kjer so elementi, ki so bili izločeni iz prejšnjih verzij prepovedani,
- »Transitional«, kjer so elementi, ki so bili izločeni iz prejšnjih verzij dovoljeni (razen elementa `frame`),
- »Frameset«, kjer sta še vedno dovoljena elementa `frameset` in `frame`.

Trenutno je v razvoju HTML 5, ki je prva verzija HTML-ja, ki se bo oddaljila od svojega predhodnika SGML-ja glede procesiranja kode. HTML 5 bo predstavil nove elemente kot so `article`, `aside`, `audio`, `canvas`, `command`, `datalist`, `details`, `embed`, `figcaption`, `figure`, `footer`, `header`, `hgroup`, `keygen`, `mark`, `meter`, `nav`, `output`, `progress`, `rp`, `rt`, `ruby`, `section`, `source`, `summary`, `time`, `video` in `wbr`, nova vnosna polja oz. kontrole na maski za datume, e-mail naslove, url naslove, iskanja, barve in nove attribute `charset` (pri elementu `meta`), `async` (pri elementu `script`) ter nove globalne attribute, ki bodo lahko aplicirani na vsak element (`id`, `tabindex`, `hidden`, `data`).

Od leta 2000 naprej pa se razvija še jezik XHTML, ki je kombinacija jezikov HTML 4.01 in XML 1.0.

3.1.2 Sintaksa

Kot smo že omenili je HTML sestavljen iz elementov. Primer programa »Hello World!« si lahko ogledamo na izpisu 1.

```
<!doctype html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World! </p>
  </body>
</html>
```

Izpis 1: HTML sintaksa.

Za HTML oznake veljajo naslednja pravila:

- oznaka je vedno znotraj lomljenih oklepajev (< in >);
- poznamo začetno (<head>) in končno oznako, ki se od začetne razlikuje po poševnici pred imenom (</head>);
- ime oznake ni občutljivo na velike in male znake; posledično so <head>, <HEAD> in <hEaD> enake [12].

HTML elemente lahko ločimo na:

- prazne elemente, ki vsebujejo le začetno oznako in atribute; takšna sta elementa
 in <link>

```
<link rel=stylesheet href="style.css" media="screen" > ;
```
- normalne elemente, ki imajo začetno in končno oznako in poljubno število atributov, znotraj oznak pa vsebujejo tekst in ostale elemente;
- tekstovne elemente, ki imajo začetno in končno oznako, a za razliko od normalnih elementov ne vsebujejo ostalih elementov, tako da je celoten tekst znotraj oznak vsebina [11].

Vsak HTML element ima lahko tudi attribute, ki so vsebovani v začetni oznaki elementa.

Sestavljeni so iz imena in vrednosti, ki sta ločena med sabo z enačajem, vrednost pa je praviloma znotraj enojnih ali dvojnih narekovajev. V primeru

```
<span id="anId" class="jargon" style="color:purple;" title="Hypertext Markup Language">HTML</span>
```

so atributi id, class, style in title, njihove vrednosti pa so an-id, jargon, color:purple in Hypertext Markup Language.

3.1.3 Cascading Style Sheets (CSS)

CSS je slogovni jezik, s katerim obvladujemo izgled in formatiranje vsakega dokumenta, ki je napisan v označevalnem jeziku [13]. Najpogostejša uporaba pa je sigurno oblikovanje spletnih strani oz. dodajanje predstavitvene plasti HTML-ju in XHTML-ju. Prav ločevanje vsebine dokumenta od vizualizacije dokumenta pa je glavni namen CSS-ja. Ločitev omogoča:

- lahko ponovno uporabo sloga,
- prekrivanje (angl. *cascading*) - več različnih stilov lahko uporabimo naenkrat,
- izbiro stila glede na način prikazovanja oz. medij prikazovanja,
- uporabniku lahko ponudimo različna prikazovanja strani,
- izboljšavo učinkovitosti (slogovna datoteka se prenese le enkrat in se shrani v brskalnikov predpomnilnik).

Sintaksa (glej izpis 2) je preprosta in je pripomogla pri uveljavitvi CSS-ja.

```
selector [, selector2, ...][:pseudo-class] {  
  property: value;  
  [property2: value2;  
  ...]  
}
```

Izpis 2: CSS sintaksa [13].

Slogovna datoteka je sestavljena iz pravil. Vsako pravilo je sestavljeno iz enega ali več selektorjev in deklaracijskega bloka, ki je znotraj zavutih oklepajev.

Deklaracijski blok je sestavljen iz lastnosti (angl. *property*), ki jo želimo pripisati elementu in njene vrednosti (angl. *value*), ločeni pa sta z dvopičjem. Če je lastnosti več, jih ločimo s podpičjem, če pa ima ena lastnost več vrednosti, te ločimo z vejico.

Selektor (angl. *selector*) določa, kateri izmed elementov bo podedoval stil, ki je določen v deklaracijskem bloku.

Poleg selektorja se lahko navede še psevdo razred (angl. *pseudo-class*) ali psevdo element, ki ga ločimo od selektorja z dvopičjem in omogoča posebne stile ob določenih dogodkih na spletni strani [14].

CSS lahko v HTML vključimo na dva različna načina:

- CSS kodo zapišemo v svojo datoteko in jo potem uvozimo v samo HTML datoteko;

to storimo z elementom `<link>` npr.:

```
<link rel=stylesheet href="style.css" media="screen"> ;
```

takega načina se poslužujemo, ko je CSS kode zelo veliko oz. jo nameravamo ponovno uporabiti;

- CSS kodo zapišemo kar znotraj samega HTML dokumenta znotraj `<style>` elementa npr.:

```
<style media=print>
  .print {
    font-family: Tahoma, Arial, Verdana, Helvetica;
    font-size: 20px;
  }
  .noprint {display:none;}
</style> ;
```

tak način uporabimo le, ko je stil specifičen za trenutno stran in ga ne nameravamo uporabljati drugje.

3.1.4 JavaScript

JavaScript je implementacija ECMAScript standarda. Razvil ga je Brendan Eich leta 1995 za podjetje Netscape, ki je nato tudi prvo implementiralo podporo za JavaScript v svoj brskalnik Netscape Navigator. Na JavaScript je zelo vplival C, kar je najbolj razvidno iz same sintakse. Ostali jeziki, ki so vplivali na razvoj pa so bili Scheme, Java, Perl in Python [15].

JavaScript je skriptni jezik, ki je dinamično in šibko tipiziran in omogoča prototipno objektno programiranje [15].

JavaScript je bil sprva razvit in se še vedno najpogosteje uporablja za integracijo dinamične vsebine v spletne strani oz. v HTML dokumente. Navadno se za ta namen uporablja client-side JavaScript, ki se interpretira v odjemalčevem brskalniku brez predhodnega prevajanja v strojno kodo.

Osnovne funkcije JavaScripta znotraj HTML dokumentov so:

- oblikovalcem omogoča preprosto programiranje,

- omogoča uporabo dinamične vsebine,
- reagira na dogodke, kot so kliki (`onclick`) ali končanje nalaganja HTML datoteke (`onload`),
- branje in pisanje vrednosti v HTML elemente, kar nam omogoča spreminjanje vrednosti elementov,
- validacija vnesenih podatkov pred oddajanjem,
- detekcija vrste in verzije brskalnika, kar nam omogoča prirejanje strani za posamezne brskalnike,
- kreiranje piškotkov [16].

V HTML JavaScript vgradimo z HTML elementom `<script>`. Preprost primer JavaScripta znotraj HTML dokumenta bi tako izgledal tako:

```
<script type="text/javascript">
  document.write("Hello world!");
</script>
```

Tako kodo lahko umestimo znotraj `<body>` elementa, koda pa bi se izvedla ob naložitvi le tega elementa. Če pa bi v kodi ustvarili funkcijo

```
<script>
function funkcija() {
  document.write("Hello world!");
}
</script>
```

in jo namestili v `<head>` ali v `<body>`, bi jo lahko preprosto uporabili s klicem

```
<script>funkcija();</script>
```

Za lažjo ponovno uporabo JavaScripta lahko v sam `<script>` element dodamo atribut `src`

```
<script src=".\\CreaSignWeb\\creasignweb.js"></script>
```

kar nam omogoča uporabo vseh funkcij znotraj datoteke, ki je navedena v vrednosti atributa.

3.2 Visual Basic

Visual Basic se je razvil iz programskega jezika BASIC pod okriljem Microsofta. Cilj Microsofta je bil združiti preprostost programiranja v BASICu in lahko oblikovanje form. Najnovejše verzije (od leta 2003 oz. verzije 7.1 naprej) uporabljajo Microsoftovo .NET platformo. Vendar je obstoječe spletno bančništvo izvedeno v Visual Basic 6, ki uporablja COM objekte, zato se bomo v nadaljevanju posvetili le temu.

3.2.1 Visual Basic 6

Visual Basic (v nadaljevanju VB) je visokonivojski računalniški programski jezik, ki se uporablja za dogodkovno programiranje in razvojno okolje, ki ga je razvil Microsoft za njihov COM (angl. *Component Object Model*) model [17]. Razvil se je iz jezika BASIC in je bil tako kot njegov predhodnik razvit za uporabo začetniških programerjev, kljub temu pa je mogoče z VB razviti kompleksnejše programe. Predvsem preprostost razvijalskega okolja je bila velika prednost VB-a pred ostalimi programskimi jeziki kot so C++, C# ali Java, saj kontrole s katerimi želimo delati preprosto nanesimo na formo s tehniko »drag and drop«.

Z VB je možno narediti izvršljive datoteke (.exe), ActiveX kontrole in DLL datoteke. Primarno je uporabljen za razvijanje aplikacij v Windows okolju in dodajanje vmesnikov podatkovnim bazam. Alternativno se lahko VB uporablja tudi brez vmesnika (vizualne komponente) in le oskrbuje druge programe z ActiveX objekti preko COM-a, kar omogoča procesiranje na strani strežnika. S prihodom VB 6 pa se je VB razširil še na področje spletnih aplikacij (DHTML in IIS aplikacije).

VB je na začetku kodo prevedel v P-kodo, ki se interpretira znotraj VB virtualnega stroja. Prednost P-kode je prenosljivost. S prihodom verzije 5 in 6 pa lahko VB kodo prevede tudi v strojno kodo, ki je optimalnejša za izvajanje predvsem večjih aplikacij [17].

Posebnost VB-a so tudi spremenljivke, ki so splošno definirane kot podatkovni tip `variant`, ki lahko sprejme številko, datum ali niz. To je tudi ena izmed pomanjkljivosti VB-ja, saj `variant` podatkovni tip zahteva več sistemskih virov, kot ostali podatkovni tipi, seveda pa se lahko z dobrim programiranjem podatkovnemu tipu `variant` tudi izognemo. Pri tem nam lahko pomaga tudi možnost »Explicit«, s katero zahtevamo eksplicitno definiranje tipa spremenljivke. Poleg `variant` podatkovnega tipa poznamo še vse ostale klasične podatkovne tipe, ki so prikazani v tabeli 4.

Tabela 4: Podatkovni tipi v okolju Visual Basic 6 [18].

Podatkovni tip	Vrednosti
Byte	Cela števila od 0 do 255
Integer	Cela števila od -32.768 do +32.767
Long	Cela števila od -2.147.483.468 do +2.147.483.468
Single	Decimalna števila od $-3,4 \times 10^{-38}$ do $+3,4 \times 10^{38}$
Double	Decimalna števila, ki presegajo natančnost Single podatkovnega tipa.
Currency	Denarne vrednosti (4 števke desno od decimalne vejice in 15 števk levo od decimalne vejice).
String	Alfa-numerične vrednosti.
Date	Datumske vrednosti od 01.01.0100 do 31.12.9999
Boolean	Vrednosti True in False.
Variant	Poljubne vrednosti.

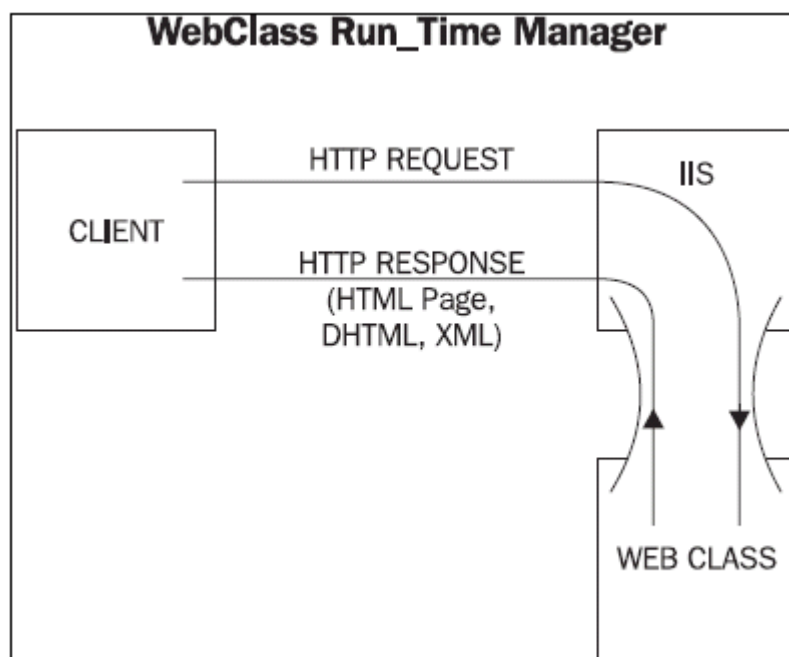
Boolean shranjuje vrednosti True in False numerično. True se zapiše v spremenljivko kot vrednost -1, False pa kot 0. Kot True se interpretirajo vsa števila, ki so različna od 0.

3.2.2 WebClasses in IIS aplikacije

IIS (angl. *Internet Information Server*) aplikacije so alternativa klasičnemu ASP-ju (angl. *Active Server Pages*). Njihova prednost je predvsem integracija VB sintakse, ki je omogočila VB programerjem razvijanje spletnih aplikacij brez učenja dodatne sintakse (ASP) [19].

IIS aplikacije so strežniške aplikacije, ki gostujejo na Microsoftovem IIS-u, samo aplikacijo pa lahko uporabnik uporablja kar preko spletnega brskalnika. Sama tehnologija je zelo podobna delovanju prejšnjih tehnologij, kot je ASP, vendar z možnostjo izrabe vseh funkcij VB in tudi implementacijo prejšnjih rešitev napisanih v VB.

VB IIS aplikacije so variacija standardnim ActiveX DLL projektom z razliko, da se sklicujejo (imajo referenco) na Microsoft WebClass Library (mswcrun.dll), ki omogoča uporabo jedra vsake IIS aplikacije - WebClass objekta. WebClass objekt sodeluje še z eno drugo COM komponento, ki se imenuje WebClass Run-time Manager COM objekt. Le ta omogoča WebClass objektu, da postane prejemnik HTTP zahtevkov (glej sliko 8).

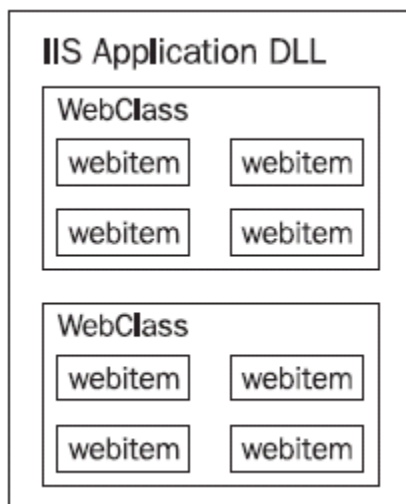


Slika 8: Delovanje IIS aplikacije [19].

WebClass objekt je specializirana COM komponenta znotraj vsake IIS aplikacije, ki omogoča proženje ustrezne VB kode ob prejemu HTTP zahtevka in posledično generiranje ustreznega HTTP odgovora. HTTP zahtevke proži sam uporabnik spletne aplikacije s pomočjo interaktivnih HTML elementov.

WebClass je ovit v ActiveX DLL, ki je na serverju, kar nam omogoča instanciranje objekta na vsak zahtevek odjemalca. Za samo kreiranje instance WebClass objekta poskrbi ASP stran, ki jo avtomatično generira VB. Ob instanciranju se WebClass naloži v pomnilnik serverja, instanca pa se uniči ob uspešnem procesiranju HTTP zahteve oz. ko se sproži ustrezna VB koda.

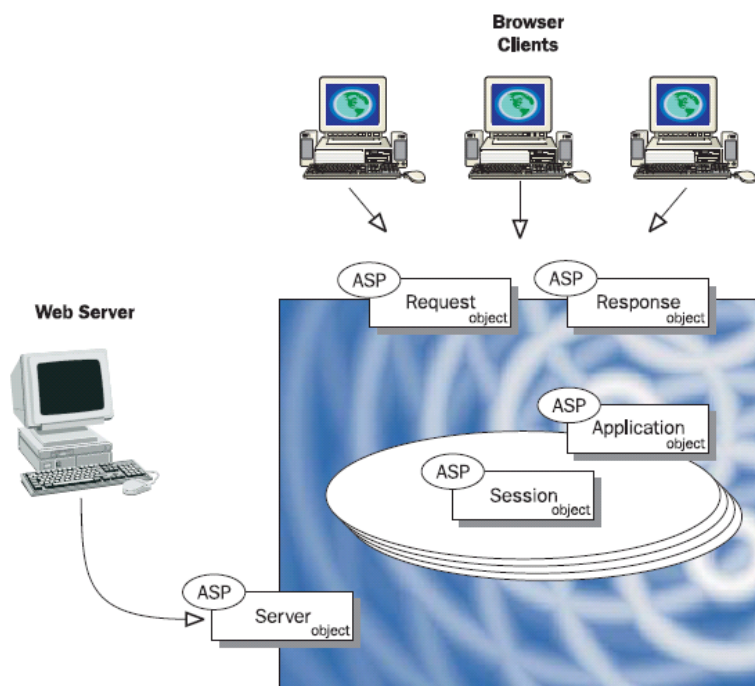
Vsaka IIS aplikacija je sestavljena iz več WebClassov, za vsak WebClass pa je avtomatično kreirana tudi ASP stran, ki omogoča njegovo instanciranje. WebClass lahko vsebuje več elementov (angl. *webitem*) s katerimi odgovori na HTTP zahteve. Takšna zgradba spletne aplikacije je prikazana na sliki 9.



Slika 9: Sestava spletne aplikacije [19].

Elementi so HTML predloge, ki jih dokončno oblikujemo s pomočjo VB kode. HTML predloge nam omogočajo, da ločeno razvijamo vizualni in poslovni del aplikacije. Prednosti [19] uporabe WebClassov in IIS aplikacij so:

- *cena namestitve*: z uporabo lahkega odjemalca (angl. *thin client*) zmanjšamo stroške namestitve, saj nam ni potrebno razvijati in nameščati dodatne programske opreme;
- *razvijalno okolje*: poznavanje standardnega VB okolja za razvoj omogoča hitrejši razvoj;
- *objektni model*: uporaba Active Server Pages objektnega modela (glej sliko 10) nam omogoča lahko upravljanje z viri IIS-a;
- *obseg uporabnikov*: preprosta in univerzalna programska oprema na strani odjemalca (spletni brskalnik) nam omogoča povečanje populacije uporabnikov, ne glede na platformo ali operacijski sistem oz. nadaljni razvoj računalništva;
- *ponovna uporaba*: z generiranim DLL-jem lahko hitro prenesemo aplikacijo na nov IIS ali pa lahko kodo uporabimo iz več različnih poti;
- *ločitev kode*: od samega oblikovanja omogoča razdelitev vlog znotraj razvojnih skupin in tako lažje, hitrejšo in učinkovitejšo razvijanje;
- *upravljanje stanj*: ki je samoumevno znotraj namiznih aplikacij, saj s pravilnim deklariranjem spremenljivke le to lahko vidimo znotraj vsake forme, medtem, ko tukaj z uporabo ASP objektov, kot so `Session` in `Application` prenašamo spremenljivko oz. informacije o stanju skozi celotno aplikacijo in med različnimi WebClassi.



Slika 10: Active Server Pages objektni model [19].

3.3 Extensible Markup Language (XML)

XML je tako kot HTML označevalni jezik, ki za razliko HTML-ja nima točno določenega namena uporabe in se ga posledično da prilagoditi glede na uporabo. XML predstavlja za razliko od HTML-ja le vsebino in ne tudi vizualne predstavitve poleg tega nima točno določenih imen elementov in jih tako lahko poimenujemo sami, kar nam omogoča lažje razumevanje XML dokumenta.

XML nam omogoča pošiljanje strukturiranih podatkov med aplikacijami, ki lahko delujejo na različnih sistemih in platformah [20].

Sestava XML dokumenta (glej izpis 3) je podobna HTML dokumentu, saj vsebuje na začetku XML dokumenta XML deklaracijo, ki definira da gre za XML dokument, sledijo pa ji lahko ostali elementi XML dokumenta, kot so oznake, elementi in atributi.

Zaradi razširljivosti in posledično možnosti enakega poimenovanja elementov, ki imajo različen pomen, so se v XML-u uveljavili imenski prostori (angl. *namespaces*), ki točno določijo za kateri element gre.

```

<?xml version="1.0"?>
<InvoiceSubscription xmlns:bs-sibex="http://BankaSparkasse.si/sibex/V1.00">
  <bs-sibex:MsgID DocID="20110408125236980">
    <DocTime>2011-04-08T12:52:36</DocTime>
    <Version>1.0</Version>
  </bs-sibex:MsgID>
  <RcpDtls>
    <RcpIBAN>SI56340001001313613</RcpIBAN>
    <RcpTaxID>50026319</RcpTaxID>
    <RcpType>FO</RcpType>
    <RcpName>MAEKO Madonca</RcpName>
    <RcpAddr>Mivka</RcpAddr>
  </RcpDtls>
</InvoiceSubscription>

```

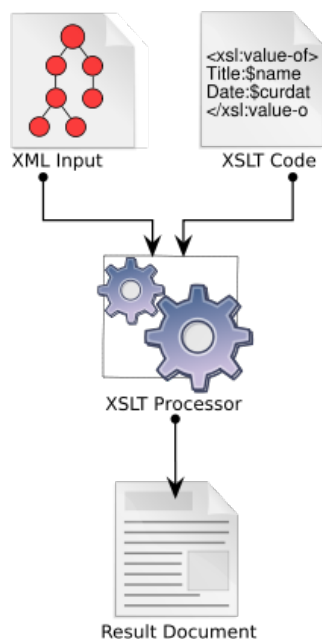
Izpis 3: Sestava XML dokumenta.

Vsak dobro formuliran XML dokument (angl. *well-formed XML*) mora biti pravilno zgrajen glede na pravila, ki jih določa XML specifikacija. Poleg tega pa lahko zgradbo XML dokumenta primerjamo z XML shemo. Če se XML dokument ujema s shemo, mu pravimo veljaven XML dokument (angl. *valid XML*). Za validacijo se uporablja različne sheme:

- *Document Type Definitions (DTD)*: prva izmed vseh shem;
- *XML Schema Definitions (XSD)*: W3C standard, ki je nadgradil DTD; prednosti XML sheme so uporaba XML sintakse, popolna podpora imenskih prostorov, validacija tekstovnih polj glede na podatkovne tipe, ki so že vgrajeni ali pa definirani s strani uporabnika, lažja izgradnja in ponovna uporaba vsebinskih modelov; vsebuje osnovne principe programiranja kot so dedovanje in transformacija tipov;
- *Regular Language for XML Next Generation (RELAX NG)*: združuje prednosti XSD in DTD ter odpravlja njune napake; kljub temu je še vedno bolj razširjen XSD, ker ima večjo podporo;

Ker so v XML-u glavni podatki in se sam standard ne ukvarja s prikazovanjem le teh, so se razvili tudi standardi, ki omogočajo oblikovanje in prikazovanje podatkov. Za oblikovanje prikaza podatkov se večinoma uporabljata XSLT (angl. *Extensible Stylesheet Language Transformations*) in CSS. XSLT je sicer namenjen transformiranju XML dokumentov. Možna je transformacija XML dokumenta v XML dokument z drugačno strukturo, vendar se lahko uporabi tudi za transformacijo v drug format, kot so HTML, XHTML, PDF in druge formate, ki se uporabljajo za prikazovanje vsebine na uporabniku prijazen način. XSLT za transformiranje uporabi XSLT procesor, ki za parametre uporabi XML dokument in XSLT

predlogo in po opravljenih transformacijah vrne nov dokument (glej sliko 11).



Slika 11: Delovanje XSLT transformacije [21].

CSS se lahko uporablja samostojno znotraj XML dokumenta ali pa skupaj s pomočjo XSLT, tako da najprej XML dokument pretvorimo v HTML ali XHTML dokument, ki ga nato oblikujemo s pomočjo CSS.

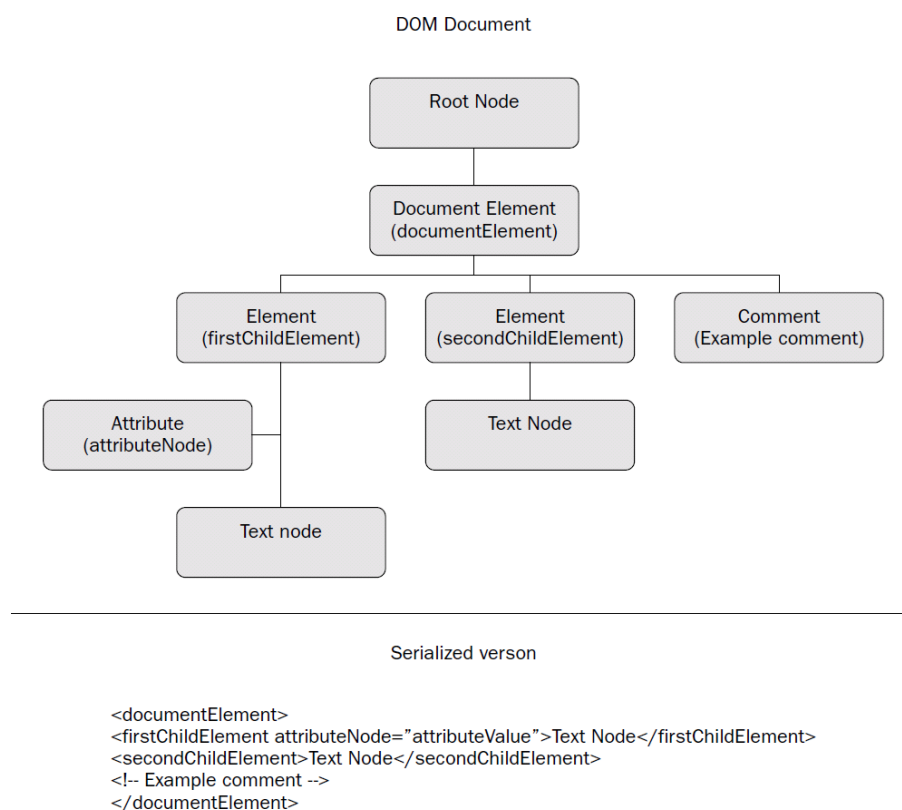
XSLT uporablja še eno XML tehnologijo imenovano XPath (angl. *XML Path Language*), brez katere ne more funkcionirati. XSLT implementira zanke in pogojne stavke, medtem ko XPath omogoči izbiranje določenih elementov znotraj XML dokumenta. Na izpisu 4 so vse vrednosti atributov `select` v XPath formatu.

```
<xsl:for-each select="catalog/cd">
  <tr>
    <td><xsl:value-of select="title"/></td>
    <td><xsl:value-of select="artist"/></td>
  </tr>
</xsl:for-each>
```

Izpis 4: Izsek iz XSLT transformacije.

Za potrebe programerjev za obdelavo XML dokumentov sta se razvila dva programska vmesnika XML DOM (angl. *Document Object Model*) in SAX (angl. *Simple API for XML*). Razlika med njima je v načinu procesiranja. Medtem ko DOM ustvari v pomnilniku drevesno

strukturo XML dokumenta (glej sliko 12) in tako lahko dostopamo, brišemo ali spreminjamo kateri koli element v dokumentu, ga SAX procesira po en element naenkrat od začetka do konca. Ob procesiranju SAX proži dogodke (začetek dokumenta, začetek elementa, vsebina elementa, konec elementa, konec dokumenta itd.). Tako je SAX boljši, ko imamo opravka z velikimi XML dokumenti oz. imamo na voljo premajhen pomnilnik [20].



Slika 12: DOM struktura in pripadajoči XML dokument [20].

3.4 XML podpis

XML podpis (angl. *XML signature*) je specifična XML sintaksa, ki je uporabljena za predstavitev digitalnega podpisa na kakršni koli digitalni vsebini. Digitalni podpisi se uporabljajo za zagotavljanje integritete oz. pristnost podatkov in avtentikacijo oz. identifikacijo sporočila in podpisnika [22].

XML podpis se je razvil zaradi potrebe po poenostavitvi digitalnega podpisovanja. To je dosegel s svojo strukturirano zgradbo (glej izpis 5) in tako doprinesel k enostavnejšemu razumevanju in zmanjšanju poslanih podatkov potrebnih za digitalni podpis.

```
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
    (<Reference URI? >
      (<Transforms>)?
      <DigestMethod>
      <DigestValue>
    </Reference>)+
  </SignedInfo>
  <SignatureValue>
    (<KeyInfo>)?
    (<Object ID?>)*
</Signature>
```

Izpis 5: Struktura XML podpisa.

Signature element je koren samega XML podpisa znotraj katerega se definirajo vse lastnosti le tega. SignedInfo element vsebuje vse informacije o podatkih, ki smo jih podpisali. Ime samega elementa je okrajšava za angleško zvezo »signed information«. CanonicalizationMethod element definira metodo normalizacije XML dokumenta, medtem ko element SignatureMethod določa algoritem uporabljen za sam podpis. Vrednosti atributov v CanonicalizationMethod in SignatureMethod vplivata na vrednost v elementu SignatureValue. Element Reference določa podatke, ki smo jih podpisali, njegov sin Transforms vsebuje elemente Transform, ki določajo transformacije (normalizacija, kodiranje, validacija, XSLT ...) opravljene nad vhodnimi podatki pred samo obdelavo. DigestMethod določa zgoščevalno funkcijo s katero dobimo vrednost DigestValue po opravljenih transformacijah. DigestValue je tako zgoščevalna vrednost podpisanih podatkov. SignatureValue element vsebuje niz bitov, ki so kodirani z Base64 standardom in predstavljajo elektronski podpis podatkov elementa SignedInfo. KeyInfo element definira certifikat oz. ključ s katerim prejemnik dokumenta lahko preveri pristnost podpisa. Element Object vsebuje originalni dokument oz. podatke, ki smo jih podpisali in je prisoten le ob »enveloping« podpisih.

Poznamo tri različne tipe XML podpisa [22], ki se ločijo glede na končno strukturo XML dokumenta (glej izpis 6). Možne strukture končnega dokumenta so:

- element Signature postane sin originalnega dokumenta (angl. *enveloped*);
- element Signature postane starš podpisane vsebine (angl. *enveloping*);

- element `Signature` je v svojem dokumentu ločen od vsebine na katero se sklicuje v elementu `Reference` (angl. *detached*).

```
<!-- Enveloped -->
<OriginalniXMLdokument>
  <Signature>...</Signature>
</OriginalniXMLdokument>
<!-- Enveloping -->
<Signature>
  <OriginalniXMLdokument></OriginalniXMLdokument>
</Signature>
<!-- Detached -->
<Signature>...</Signature>
```

Izpis 6: Različni formati XML podpisa.

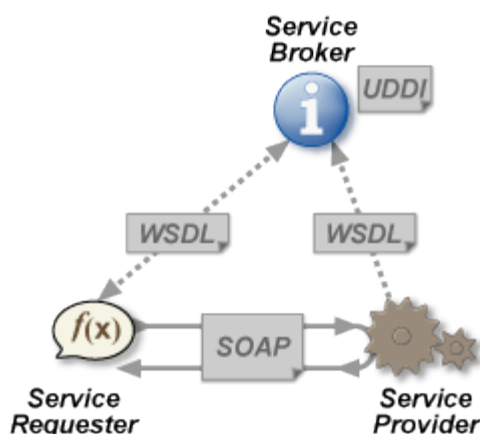
3.5 Spletne storitve

Spletna storitev je način komuniciranja dveh naprav preko omrežja. S pomočjo spletnih storitev lahko ponudimo funkcionalnost svojih aplikacij širšemu krogu uporabnikov ali pa le ostalim svojim aplikacijam [23].

Spletne storitve so se začele razvijati s pospešitvijo interneta, saj na samem začetku niso bile dovolj hitre, da bi zadovoljile potrebo po hitri izmenjavi informacij.

Glavni namen spletnih storitev je medobratovalnost oz. možnost komuniciranja različnih sistemov in platform med seboj, zato spletne storitve bazirajo na uporabi XML standarda in HTTP protokola. Z uporabo le teh so se razvili dodatni standardi in protokoli, ki so prikazani na sliki 13, za uporabo s spletnimi storitvami [24]:

- *Simple Object Access Protocol (SOAP)*: protokol, ki omogoča komunikacijo spletnih storitev;
- *Web Service Definition Language (WSDL)*: jezik za opisovanje spletnih storitev in njihovih funkcij;
- *Universal Description, Discovery and Integration (UDDI)*: register spletnih storitev.



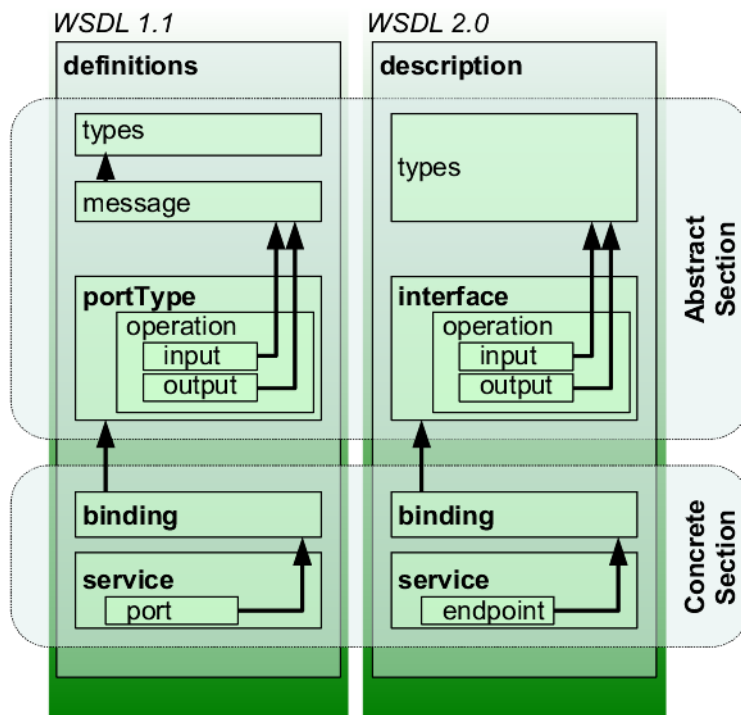
Slika 13: Prikaz delovanja spletnih storitev [23].

Spletne storitve imajo tri pogoste načine uporabe:

- **RPC** (angl. *Remote Procedure Call*) spletne storitve predstavljajo vmesnike klicev distribuiranih funkcij. Uporabljale so se predvsem na začetku razvoja spletnih storitev in večinoma niso bile interoperabilne in so se naslanjale na določeno tehnologijo. Podobni pristopi kot RPC so bili še DCOM (angl. *Distributed Component Object Model*), RMI (angl. *Remote Method Invocation*), CORBA (angl. *Common Object Model Broker Architecture*) in OMG (angl. *Object Management Group*).
- **SOA** (angl. *Service-oriented architecture*) je arhitektura, ki temelji na sporazumevanju aplikacij s pomočjo spletnih storitev. Temelj komunikacije je tokrat sporočilo in ne klic funkcije. Ker temelji na WSDL in SOAP standardih, so lahko aplikacije za različne platforme.
- **REST** (angl. *Representational state transfer*) ali RESTful spletne storitve uporabljajo za svoje delovanje metode HTTP protokola (npr. GET, POST, PUT in DELETE). Za klic funkcije tako uporabimo primerno HTTP metodo in XML sporočilo.

3.5.1 Web Service Definition (Description) Language (WSDL)

WSDL je jezik za opisovanje spletnih storitev, ki temelji na XML-u [25]. Trenutno sta v uporabi dve različni verziji WSDL-ja. Stara verzija 1.1 je v večinski uporabi zaradi prirejenosti orodij na to verzijo, medtem ko verzija 1.2 (kasneje preimenovana v 2.0) ponuja izrabljanje vseh vrst HTTP zahtevkov (ne le GET in POST) in boljšo podporo RESTful spletnim storitvam [25].



Slika 14: Sestava WSDL dokumenta [25].

Na sliki 14 je prikazana sestava WSDL dokumenta. Njegovi deli so elementi: definitions, types, message, portType, binding in service [26].

Element service predstavlja množico sistemskih funkcij, ki so na voljo mrežnim protokolom. Element port oz. endpoint znotraj service elementa predstavlja končno točko oz. naslov spletne storitve. Element binding definira obliko oz. format in podrobnosti protokola za vsako operacijo in sporočilo, ki je definirana znotraj portType elementa. Element portType opisuje samo spletno storitev, definira abstraktne operacije in sporočila, ki so v njej vsebovana. PortType pozna štiri različne vrste operacij:

- *One way* - končna točka dobi sporočilo,
- *Request-response* - končna točka prejme sporočilo in nanj odgovori,
- *Solicit-response* - končna točka pošlje sporočilo in sprejme odgovor,
- *Notification* - končna točka pošlje sporočilo.

Primerljiv je z funkcijsko knjižnico, razredom ali modulom v navadnem programskem jeziku. Element message vsebuje enega ali več elementov part, ki določajo podatkovni tip operacije. Lahko bi jih primerjali s parametri funkcije v navadnem programskem jeziku. Element types vsebuje definicije podatkovnih tipov, ki so uporabljeni v spletni storitvi. Za definiranje

podatkovnih tipov uporabimo XML shemo (XSD) .

3.5.2 Simple Object Access Protocol (SOAP)

SOAP je komunikacijski protokol, ki je uporabljen za komunikacijo med aplikacijami preko interneta oz. preko HTTP protokola. SOAP bazira na XML jeziku in je neodvisen od platforme in programskega jezika [27]. SOAP sporočilo (glej izpis 7) je sestavljeno iz treh delov oz. XML elementov:

- `Envelope` je koren XML dokumenta in ga definira kot SOAP dokument;
- `Header` je opsijski element in vsebuje elemente o SOAP zahtevku, ki lahko pomagajo pri avtentikaciji, plačilu ipd. Elementi znotraj glave imajo lahko tri specifične attribute:
 - `mustUnderstand`: ko je nastavljen na vrednost 1 določa, da mora prejemnik prepoznati ta element, drugače bo prišlo do napake pri procesiranju glave in posledično tudi SOAP sporočila;
 - `actor`: SOAP sporočilo lahko potuje do svojega naslovnika preko različnih končnih točk (angl. *endpoints*); s tem atributom lahko definiramo vmesne točke;
 - `encodingStyle` definira podatkovne tipe znotraj sporočila;
- `Body` vsebuje dejansko vsebino SOAP sporočila in opsijsko lahko vsebuje tudi element `Fault`, ki vsebuje podatke o napakah.

SOAP nam omogoča uporabo XML podpisa za avtentikacijo. Prav tako nam omogoča uporabo priponk, tako da je mogoče preko SOAP sporočila pošiljati različne datoteke in ne le XML dokumente [20]. Največja prednost SOAP protokola pa je predvsem prenos podatkov o HTTP protokolu, saj je le ta veliko varnejši kot RPC protokol, ki se je uporabljal pred tem.

```
<!-- SOAP zahtevek -->
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope">
...
<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPrice>
    <m:StockName>IBM</m:StockName>
  </m:GetStockPrice>
</soap:Body>
</soap:Envelope>

<!-- SOAP odgovor -->
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope">
...
<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPriceResponse>
    <m:Price>34.5</m:Price>
  </m:GetStockPriceResponse>
</soap:Body>
</soap:Envelope>
```

Izpis 7: SOAP zahtevek in odgovor [27].

4 Implementacija

Za našo stranko Banko Sparkasse d.o.o. (BSK) smo morali v podjetju S&T Slovenija d.o.o. (S&T) po naročilu izdelati dopolnitev za obstoječe spletno bančništvo, tako da bo le to omogočalo prejem e-računov. Ker je spletno bančništvo namenjeno predvsem fizičnim osebam, ni bilo potrebe po vgraditvi podpore izdaji e-računov.

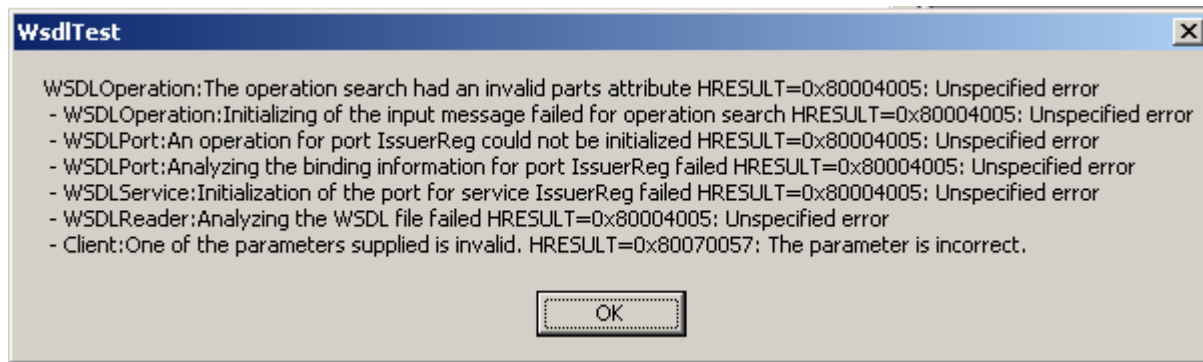
Za prejemanje e-računov smo morali omogočiti nove funkcionalnosti kot so:

- naročilo na e-račun, ki je vsebovalo elemente:
 - prijava in odjava na e-račun,
 - pregled prijav in odjav ter
 - pregled seznama izdajateljjev;
- plačilo e-računov, ki je vsebovalo naslednje podfunkcije:
 - pregled e-računov pripravljenih na plačilo,
 - pregled plačanih e-računov in
 - iskalnik e-računov.

Sama rešitev je morala uporabljati spletne storitve, ki jih je pred tem izdelalo podjetje Halcom d.d., brez shranjevanja podatkov v lokalno podatkovno bazo.

Programski jezik Visual Basic 6, v katerem je napisano spletno bančništvo, ne omogoča samostojne komunikacije s spletnimi storitvami. Zaradi tega smo za implementacijo uporabili Soap Toolkit 3.0, ki omogoča uporabo knjižnic namenjenih komunikaciji s spletnimi storitvami.

Zaradi težav (glej sliko 15) z uporabo predloženih WSDL datotek, ki so opisovale Halcomove spletne servise, in HTTPS protokolom v samem Soap Toolkitu, smo se odločili, da naredimo vmesno (angl. *proxy*) spletno storitev. Uporabo Soap Toolkita smo želeli omejiti tudi zaradi iztekle podpore s strani Microsofta, ki je trajala do leta 2005 (angl. *standard support*) oz. 2008 (angl. *extended support*). Zaradi nepodprtosti s strani Microsofta ima Soap Toolkit omejeno podporo na našem testnem in produkcijskem strežniku, ki ima nameščen operacijski sistem Windows Server 2003.



Slika 15: Napaka ob inicializaciji objekta SoapClient.

Vmesno spletno storitev smo izdelali v VB .NET, ki s pomočjo WSDL datoteke omogoča avtomatično generiranje razredov potrebnih za komunikacijo s spletno storitvijo.

Ob migraciji aplikacije iz razvojnega okolja (Windows XP in IIS 5.0) v testno okolje (Windows Server 2003 in IIS 6.0), smo imeli težave s komunikacijo z vmesno spletno storitvijo. Na napako smo naleteli ob poskusu inicializacije objekta SoapClient z WSDL datoteko, ki je bila referencirana preko URL naslova. Napaka se je glasila:

```
WSDLReader:XML Parser failed at linenumber 1, lineposition 0, reason is:
System error: -2146697209.
  HRESULT=0x1: Incorrect function.
  - WSDLReader:Loading of the WSDL file failed HRESULT=0x80070057: The
parameter is incorrect.
  - Client:One of the parameters supplied is invalid. HRESULT=0x80070057:
The parameter is incorrect.
```

To smo odpravili z nastavitvijo na odjemalčevi strani:

```
clnt.ClientProperty("ServerHTTPRequest") = True
```

Napaka se je pojavljala, ker se WSDL datoteke naložijo v objekt WSDLReader v obliko DOM z uporabo HTTP vmesnika iz MSXML knjižnice.

4.1 Naročilo na e-račune

Meni naročilo na e-račune vsebuje tri funkcije in sicer prijavo oz. odjavo na e-račune, seznam izdajateljev in pregled oddanih vlog.

4.1.1 Oddaja vloge za e-prijavo ali e-odjavo

Oddaja vloge za e-prijavo (glej sliko 16) je prvi korak vsakega prejemnika e-računov. Vlogo

je možno oddati pri samem izdajatelju e-računa, poslovni banki ali preko spletnega bančništva.

E-RAČUNI: Oddaja vloge za e-prijavo/e-odjavo Pomoč

Prijava Odjava

Podatki o prejemniku e-računa

Tip prejemnika: fizična oseba
 pravna oseba
 samostojni podjetnik

IBAN številka transakcijskega računa: SI56 3400 0100 1313 613

Davčna številka: **A0026319**

Ime in priimek/Naziv: **MAEKO Madonca**

Ulica in hišna številka: **Mivka**

Kraj: **Škofljica**

Kontaktne podatki

E-Mail:

Telefon: +386 41 742544

Podatki o izdajatelju e-računa

Naziv: Išči

Davčna številka:

Ulična in hišna številka:

Poštna številka in kraj:

Referenčna številka:

Splošni pogoji banke
 Sprejemam Splošne pogoje poslovanja z elektronskim bančništvom, ki veljajo za storitve e-računi.

Opomba:

Nazaj Potrdite

©2002-2011, BANKA SPARKASSE d.d. Pravica do napak in sprememb pridržana.

Slika 16: Oddaja vloge za e-prijavo ali e-odjavo.

Z željo po poenostavitvi oddaje vloge avtomatično vnesemo kar se da veliko podatkov o samem prejemniku e-računov. Avtomatično izpolnjena polja med podatki o imetniku so »IBAN številka transakcijskega računa« (spustni meni), »Davčna številka«, »Ime in priimek/Naziv«, »Ulica in hišna številka«, »Kraj«. Vneseta se tudi polji »E-mail« in »Telefon«, ki ju je možno naknadno tudi spremeniti. Podatki se vnesejo za imetnika transakcijskega računa, ne glede na to kdo izpolnjuje prijavo oz. odjavo. Vsak uporabnik spletnega bančništva dobi na izbiro transakcijske račune, na katerih ima vlogo lastnika, zakonitega zastopnika ali pooblaščen osebe. Podatke pridobimo iz podatkovne baze in jih menjamo glede na izbor transakcijskega računa. Menjavo podatkov realiziramo s preprosto JavaScript skripto.

Za vnos podatkov izdajatelja e-računov smo izdelali gumb in funkcijo »Išči«. Funkcija poišče

izdajatelje po korenu naziva in vrednosti davčne številke, ki so vnesene ob kliku. Iskanje se opravi preko Halcomove spletne storitve. Rezultate prikažemo v tabeli v novem oknu (glej sliko 17). Ob izboru enega izmed rezultatov se vnesejo vsi podatki o izdajatelju in sicer naziv, davčna številka in naslov. Izpolnita se lahko še dve opsijski polji, ki sta drugače skriti in sicer navodila za izpolnitev polja »Referenčna številka« (glej sliko 18) in splošni pogoji izdajatelja (glej sliko 19).

Naziv	Naslov	Kraj	Davčna številka	Račun
PIPCA D.O.O.	TRŽAŠKA 35	VRHNIKA	56465456	SI56 3400 0000 0089 992
TESTNO PODJETJE D.D.	TESTNA 17	1000 LAIBAH	77771111	SI56 3400 0100 0960 339
TESTNO PODJETJE D.D.	TESTNA 17	1000 LAIBAH	77771111	SI56 3400 0100 0961 503
ZAVAROVALNICA TEST D.D.	CELOVŠKA CESTA 6C	1000 LJUBLJANA	12398745	SI56 3400 0100 1371 328

Slika 17: Iskanje izdajatelja.

Prosim vpišite vašo davčno številko! Obvezen podatek!!!

Referenčna številka:

Slika 18: Navodila za izpolnitev polja »Referenčna številka«.

Splošni pogoji banke
Sprejemam Splošne pogoje poslovanja z elektronskim bančništvom, ki veljajo za storitve e-računi.

Splošni pogoji izdajatelja
Prejemnik računov soglašam, da mi izdajatelj do objave, vse račune posreduje v elektronski obliki v elektronsko banko, na račun naveden v tej vlogi. Hkrati izdajatelju jamčim za resničnost vseh podatkov iz te vloge in se zavežujem, da mu bom sam sporočal vsako spremembo podatkov, ki bi vplivala na prejem e-računa.

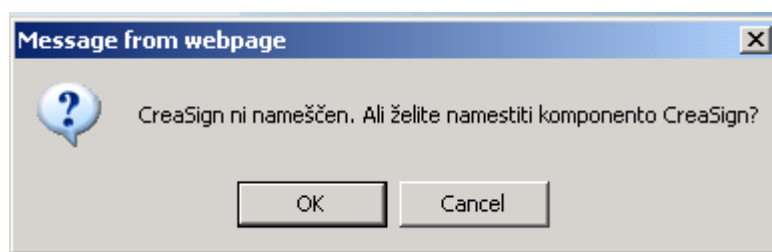
Slika 19: Splošni pogoji izdajatelja.

Po zaključenem vnašanju podatkov in kliku na gumb »Potrdi« smo usmerjeni na drugo stran, kjer lahko preverimo podatke, ki smo jih vnesli, oz. smo obveščeni o napačno izpolnjenih podatkih.

Ko na drugi strani kliknemo gumb »Potrdi«, elektronsko podpišemo vlogo in jo pošljemo v obdelavo preko spletne storitve.

Tabela 5: XML struktura e-prijave oz. e-odjave [28].

Ime taga	Obv.	Opis
<InvoiceSubscription>	M	Vodilni element prijavnice
+<MsgID>	M	Podatki o dokumentu
++<DocID>	M	Id dokumenta
++<DocTime>	M	Čas nastanka dokumenta
++<Version>	M	Različica je namenjena označevanju širitev osnovnega dokumenta. Kot različica se vpisuje 1.0
+<RcpDtls>	M	Podatki o prejemniku
++<RcpIBAN>	M	IBAN številka prejemnika
++<RcpTaxID>	O	Davčna številka prejemnika
++<RcpType>	O	Vrsta prejemnika: PO - Pravna oseba SP - Samostojni podjetnik FO - Fizična oseba
++<RcpName>	M	Naziv prejemnika
++<RcpAddr>	O	Naslov prejemnika
++<RcpCity>	O	Kraj prejemnika
++<RcpEmail>	O	E-poštni naslov prejemnika
++<RcpTel>	O	Telefonska številka prejemnika
+<IssDtls>	M	Podatki o izdajatelju
++<IssIBAN>	M	IBAN številka izdajatelja
++<IssTaxID>	M	Davčna številka izdajatelja računa
++<IssName>	M	Naziv izdajatelja računa
++<IssAddr>	M	Naslov izdajatelja
++<IssCity>	O	Kraj izdajatelja
+<SubDtls>	M	Podrobnosti o prijavi
++<InvRefID>	O	Prejemnikova naročniška številka (npr. ŠTOD, mobilna številka, št. kupca)
++<SubType>	M	Prijava/Odjava
++<BankTrms>	M	Splošni pogoji banke
++<IssuerTrms>	M	Splošni pogoji izdajatelja



Slika 20: Namestitev CreaSignWeb komponente.

Iz veljavnih podatkov najprej kreiramo XML dokument s predpisano strukturo (glej tabelo 5), ki ga kasneje še podpišemo. XML dokument elektronsko podpišemo z elektronskim potrdilom, s katerim je uporabnik prijavljen v spletno bančništvo. Za oblikovanje elektronskega podpisa si pomagamo s programom CreaSign. Sestavljen je iz dveh delov CreaSignWeb in CreaSignServer knjižnice. CreaSignWeb namestimo (glej sliko 20) na odjemalčevo stran ob prihodu na prvo stran, medtem ko je CreaSignServer knjižnica nameščena na strežniku. Če se uporabnik ne strinja z namestitvijo dodatka, ga preusmerimo

na začetno stran spletnega bančništva, saj e-prijave ni mogoče izvesti brez te komponente. XML podpis (glej izpis 8) ustvarimo v treh korakih:

- na strežniku pripravimo predlogo za podpis:

```
' na strežniku pripravimo predlogo za podpis
Set css = CreateObject("CreaSignClientIE.CreaSignServer")
Set rq = CreateObject("CreaSignClientIE.PrepareSignatureRQ")
' podamo certifikat, katerega želimo uporabiti v Base64 formatu
rq.RequiredSigningCertificate = Base64Encode(Request.ClientCertificate("Certificate"))
' izberemo vrsto podpisa - enveloped
rq.SignatureFormat = 2
' določimo dokument, ki ga želimo podpisati
rq.InputDocument = xml
' določimo mesto v xml dokumentu, kamor dodamo Signature element
rq.InsertSignatureAfter = "//SubDtIs"
' pripravimo predlogo podpisa z prej definiranimi lastnostmi
prepareddata = css.PrepareSignature3(rq)
```

- odjemalec ob potrditvi forme odda podpisan dokument:

```
' na odjemalčevi strani podpišemo predpripravljene podatke
var csw= new CreaSignWeb();
var res = csw.CreateSignature(prepareddata);
```

- in na strežniku zaključimo podpis:

```
' na stežniški strani zaključimo podpis
Set css = CreateObject("CreaSignClientIE.CreaSignServer")
css.CompleteSignature3(Request("res"))
```

Po podpisu moramo celoten XML dokument spremeniti v Base64 format. Sam VB6 ne ponuja funkcije za konverzijo v Base64 format. Zato smo v ta namen uporabili lastni funkciji za konverzijo (glej izpis 9). Kreirali smo dve funkciji:

- prva spremeni parameter tipa String v zaporedje UTF-8 bitov s pomočjo objekta Stream iz knjižnice ADODB,
- druga spremeni zaporedje bitov v Base64 format s pomočjo knjižnice MSXML in objektov DOMDocument in IXMLDOMEElement.

Vse vnesene podatke ob uspešni oddaji arhiviramo tudi v interno podatkovno bazo. Vnos uporabimo za uporabo ob možni reviziji in za prikaz podatkov v dnevniku dogodkov (glej sliki 21 in 22).

```

...
</SubDtIs>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
Id="Signature-862D0792-11CC-4C15-A8DC-C146972BFD3B">
  <ds:SignedInfo Id="SignedInfo-F1E9C204-C569-4828-8369-22E77169DE31">
    <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xm1-c14n-20010315"/>
    <ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <ds:Reference URI="">
      <ds:Transforms>
        <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>veAt8Wk4xYkPsG5YwaJInPTN0ro=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue Id="SignatureValue-5673311F-BC7A-4700-8631-A095BEDA0200">
d/k2okn6ub0KRyuzL17T SRjAo94IsLIVR9s091/VRTEfdSLw60b+ccH30IXuhJtV0QRabIGSJK/
ZD4v0LG0ZwP2CQc9gEkjJ6A2tyc2lqRyxp5vxdQghj1Pyp+hSxj5WsmXQfBBfxyDhnqsk51zJtT
G1kkLvEK/Zq/nTcHdzMjg=
  </ds:SignatureValue>
  <ds:KeyInfo>
    <ds:X509Data>
      <ds:X509Certificate>
MIICujCCAiOgAwIBAgIDBSadMA0GC SqGSIB3DQEBBQUA MDcxCzAJBgNVBAYTA1NJMQ8wD
wPEXoMBUGA1UEAwdMOVGVzdG5hLk9zZWJhNDgpcEDAOBgNVBAQTB09zZWJhNDgpcDzANBgNV
1fLfSv0hShVMD09H6ivi zWdic9sOpMK35hnV3UH2Gktgmt8Rl6cB4uz4Fn6ocEiBAA/GGk
GAQQBrjMBAgQwEwYDVR0jBAwwCoA1SK9XZr/SwAlwDgYDVR0PAQH/BAQDAgWGMG0GA1Ud
2b2NhdG1vbmcpc3Q7YmluYXJ5MA0GC SqGSIB3DQEBBQUA A4GBA1IdT29sLhgqdZi7d47n
/FPZu1x7cpNVBzBMUkuQuCTZtgmAHIOWJFA
      </ds:X509Certificate>
    </ds:X509Data>
  </ds:KeyInfo>
</ds:Signature>

```

Izpis 8: Primer pravičnega podpisa.

PREGLEDI: Dnevnik dogodkov

Dogodki od: do:

Šifra dogodka:

Vrsta dogodka:

Status dogodka (za fin. transakcije):

Prikaži

Vrsta dogodka	Datum vnosa	Šifra dogodka	Znesek (za finan. transakcije)	Status (za finan. transakcije)
E-računi prijava	08.04.2011 12:52:43	10000201100000069309		
E-računi prijava	08.04.2011 08:24:40	10000201100000069293		
E-računi prijava	28.03.2011 08:33:56	10000201100000069264		
E-računi prijava	28.03.2011 07:55:58	10000201100000069263		
E-računi prijava	28.03.2011 07:54:13	10000201100000069262		
E-računi prijava	28.03.2011 07:38:28	10000201100000069261		

Slika 21: Dnevnik dogodkov.

```

Private Function EncodeBase64(ByRef arrData() As Byte) As String
    Dim objXML As MSXML2.DOMDocument
    Dim objNode As MSXML2.IXMLDOMElement

    Set objXML = New MSXML2.DOMDocument

    Set objNode = objXML.createElement("b64")
    objNode.DataType = "bin.base64"
    objNode.nodeTypedValue = arrData
    EncodeBase64 = objNode.Text
End Function

Public Function ConvertStringToUtf8Bytes(ByRef strText As String) As Byte()
    Dim objStream As ADOBE.Stream
    Dim data() As Byte

    Set objStream = New ADOBE.Stream
    objStream.Charset = "utf-8"
    objStream.Mode = adModeReadWrite
    objStream.Type = adTypeText
    objStream.Open

    objStream.WriteText strText
    objStream.Flush

    objStream.Position = 0
    objStream.Type = adTypeBinary
    objStream.Read 3
    data = objStream.Read()

    objStream.Close
    ConvertStringToUtf8Bytes = data
End Function

```

Izpis 9: Funkciji potrebni za konverzijo niza v Base64 format.

Dnevnik dogodkov - podrobnosti o dogodku	
Prijava/Odjava:	Prijava
IBAN prejemnika:	SI56 3400 0100 1313 613
Naziv prejemnika:	MAEKO Madonca
Davčna številka prejemnika:	A0026319
Naslov prejemnika:	Mivka
Kraj prejemnika:	Škofljica
IBAN izdajatelja:	SI56 3400 0100 1371 137
Naziv izdajatelja:	ZAVAROVALNICA TEST D.D.
Davčna številka izdajatelja:	12398745
Naslov izdajatelja:	CELOVŠKA CESTA 6C
Kraj izdajatelja:	LJUBLJANA
Opomba:	test
Dogodek izvršil(-a):	Referent 31
Dogodek zahteval(-a) komitent(-ka):	KALE Rukola

Za tisk pritisnite Ctrl+P

Slika 22: Podrobnosti o dogodku.

4.1.2 Seznam izdajateljev

Seznam izdajateljev (glej sliko 23) prikažemo iz podatkov, ki so shranjeni v podatkovni bazi. Podatki se osvežijo enkrat na dan z avtomatičnim klicem na spletno storitev. Le ta ima omejitve števila vrnjenih rezultatov. Zaradi omejitve smo se odločili, da dodamo seznamu še iskalnik, ki ustvari klic na spletno storitev in prikaže ujemajoče rezultate. Iskanje je možno po dveh parametrih in sicer po nazivu izdajatelja in po davčni številki izdajatelja.

Ob kliku na gumb »Prijavi« se nam odpre maska za oddajo vloge za e-prijavo in e-odjavo z izpolnjenimi podatki o izdajatelju.

STANOVANJSKA ZADRUGA MARIBOR Z.O.O.	KREKOVA UL. 18	2000 MARIBOR	40335542	SI56251009714609231	Prijavi
STIGMA - CEVNI SISTEMI D.O.O.	MOTNICA 8	1236 TRZIN	36645419	SI56023000013268257	Prijavi
TEKOM, DODIG SREČKO S.P.	VODNIKOVO NASELJE 7	1000 LJUBLJANA	72118504	SI56020130014583963	Prijavi
TELEKOM SLOVENIJE D.D.	CIGALETOVA 15	1000 LJUBLJANA	98511734	SI56029220013902132	Prijavi
TELEKOM SLOVENIJE D.D.	CIGALETOVA 15	1000 LJUBLJANA	98511734	SI56051008012087356	Prijavi
TELMA TRADE D.O.O.	MOTNICA 13	1236 TRZIN	73961426	SI56023090016925589	Prijavi
TOP SERVICE D.O.O.	CELOVŠKA 172	1000 LJUBLJANA	14574268	SI56020130010176768	Prijavi
VODNIK JOŽE S.P.	PODUTIŠKA 162	1000 LJUBLJANA	61321869	SI56020130256154994	Prijavi
ŽAVBI PRIMOŽ S.P., DOMŽALE	PETROVČEVA 16	1260 DOMŽALE	11838566	SI56023120254888752	Prijavi

Spletni servis omejuje maksimalno število zadetkov. Privzeta omejitev je 100 izdajateljev.

Iskalnik

Naziv izdajatelja:

Davčna številka:

[Išči](#)

[Nazaj](#)

Slika 23: Seznam izdajateljev e-računov.

4.1.3 Pregled oddanih vlog

Funkcija omogoča pregled vseh oddanih vlog (glej sliko 24), tako za e-prijavo kot e-odjavo. Podatke dobimo s klicem na spletni servis. Prikažemo vse potrebne podatke o vlogi, da bi se vloge kar se da dobro razlikovale med seboj. Med podatki je tudi status vloge, ki nam

prikazuje ali je vloga sprejeta s strani izdajatelja in ali je mogoče prišlo do kakšnih zapletov. V prihodnje bo potrebno vgraditi še prikaz obrazložitve zavrnitve vloge.

Ob kliku na tekst smo preusmerjeni na oddajo e-prijave oz. e-odjave. Če kliknemo na podatek o e-prijavi, se nam avtomatsko vnesejo vsi podatki potrebni za odjavo. Če izberemo odjavo se nam prikažejo vsi podatki izbrane odjave.

E-RAČUNI: Pregled oddanih vlog				
Datum vloge	Naziv izdajatelja	Status vloge	Referenca	Tip vloge
08.04.2011	ZAVAROVALNICA TEST D.D.	Dostavljena	1231233113	Prijava
08.04.2011	TESTNO PODJETJE D.D.	Dostavljena	123123	Prijava
28.03.2011	ZAVAROVALNICA TEST D.D.	Dostavljena	123123	Prijava
28.03.2011	TESTNO PODJETJE D.D.	Dostavljena	123123	Prijava
28.03.2011	ZAVAROVALNICA TEST D.D.	Dostavljena	1234567	Prijava
29.11.2010	TESTNO PODJETJE D.D.	Dostavljena	99	Odjava
26.11.2010	TESTNO PODJETJE D.D.	Dostavljena	99	Odjava
04.11.2010	TESTNO PODJETJE D.D.	Dostavljena	99	Odjava
26.10.2010	PIPCA D.O.O.	Dostavljena	99	Prijava
21.10.2010	TESTNO PODJETJE D.D.	Dostavljena	00	Prijava
05.10.2010	PIPCA D.O.O.	Vpisana v sistem	12123123123123	Prijava
05.10.2010	PIPCA D.O.O.	Vpisana v sistem	99123123123	Prijava
05.10.2010	PIPCA D.O.O.	Vpisana v sistem	00123123123123123	Odjava
05.10.2010	PIPCA D.O.O.	Vpisana v sistem	001123456468987456	Odjava
05.10.2010	PIPCA D.O.O.	Vpisana v sistem	0012345678	Odjava
05.10.2010	PIPCA D.O.O.	Vpisana v sistem	12	Odjava
04.10.2010	PIPCA D.O.O.	Vpisana v sistem	99	Odjava
04.10.2010	PIPCA D.O.O.	Vpisana v sistem	123213123123	Odjava

Nazaj

© 2002-2011, BANKA SPARKASSE d.d. Pravica do napak in sprememb pridržana.

Slika 24: Pregled oddanih vlog.

4.2 Plačilo e-računov

Meni plačilo e-računov vsebuje tri funkcije in sicer plačilo e-računov, pregled e-računov in iskanje e-računov.

4.2.1 Plačilo e-računov

V funkciji prikažemo vse e-račune, ki so pripravljene za plačilo. To so vsi e-računi, ki še niso bili plačani oz. ne bodo poravnani preko direktne obremenitve. Podatke dobimo s klicem spletne storitve.

Za vsak račun so prikazani najnujnejši podatki in trije gumbi (glej sliko 25) »Podrobno«, »Plačaj« in »Briši«.

Ob kliku na gumb »Podrobno« se nam prikažejo vsi podatki e-računa v novem oknu (glej sliko 26). V podrobnem pregledu e-računa prejmemo tudi sporočilo o veljavnosti digitalnega podpisa, s katerim je podpisan e-račun. Samo okno vsebuje še gumb »Odpri prilogo«, ki odpre PDF ali XML datoteko, ki je priložena e-računu.

Ob kliku na gumb »Plačaj« smo preusmerjeni na obrazec BN 02 ali posebno položnico, če je model reference e-računa enak 12. Ob preusmeritvi se avtomatično vnesejo vsi potrebni podatki za plačilo, kar onemogoča napako pri vnašanju podatkov in posledično napačnemu nakazilu. Po uspešno zaključenem plačilu s klicem na spletno storitev spremenimo status e-računa na plačan. E-račun si lahko sedaj ogledamo v funkciji pregled e-računov oz. plačanih in ostalih računih.

Z gumbom »Briši« preko spletne storitve spremenimo status e-računa na brisan. Brisani e-računi se še vedno hranijo in so na voljo na ogled v funkciji pregled e-računov.

PLAČILA: E-računi za plačilo

[Pomoč](#)

Status e-računa je bil uspešno nastavljen na BRISAN!

Naziv izdajatelja	KEKO - VARICON D.O.O. ŽUŽEMBERK	IBAN izdajatelja	SI56330008075119914
Znesek računa	4,85 EUR	Naslov izdajatelja	TRE
Rok plačila	07.10.2010	Datum izdaje	26.08.2010
Namen plačila	Testni iz Izdajatelj HYPO	Številka e-računa	9069

[Podrobno](#)
[Plačaj](#)
[Briši](#)

Statusi e-računov ne odražajo statusa dejanskega plačila. Prosimo, preverite dejansko plačilo e-računa v prikazu prometa na računu oz. v primeru neizvedbe računa v Seznamu neizvedenih plačil.

[Nazaj](#)

Slika 25: E-računi za plačilo.

Podrobno e-račun			
Digitalni podpis izdajatelja je veljaven!			
Izdajatelj e-računa		Podatki o računu	
Naziv izdajatelja:	PIPCA D.O.O.	Znesek:	4,85
Račun izdajatelja:	SI56 3400 0000 0089 992	Valuta:	EUR
Prejemnik e-računa:	MAEKO Madonca	Rok plačila:	07.10.2010
Račun prejemnika:	SI56 3400 0100 1313 613	Namen plačila:	Testni iz Izdajatelj HYPO
Prejemnik plačila		Podrobnosti	
Naziv prejemnika plačila:	KEKO - VARICON D.O.O. ŽUŽEMBERK	Številka e-računa:	9069
Naslov prejemnika plačila:	TRE	Datum izdaje:	26.08.2010
Račun prejemnika plačila:	SI56 3300 0807 5119 914	Vrsta posla:	A3011
Kraj prejemnika plačila:	TER	Opomba:	?
Nalogodajalec		Model in sklic:	00 906
Naziv nalogodajalca:	EBB LJUBLJANA D.D. PREJEMNIK	Referenca:	?
Naslov nalogodajalca:	SLOVENSKA 1	Status:	Pregledan
Kraj nalogodajalca:	LJUBLJANA	Plačilna metoda:	Račun je potrebno plačati
Odpri prilogo			

Slika 26: Prikaz podrobnih podatkov e-računa.

4.2.2 Plačani in ostali e-računi

Vsi računi, ki so bili že plačani, brisani ali pa bodo plačani preko direktne obremenitve, so prikazani v plačanih in ostalih e-računih oz. krajše pregledu e-računov (glej sliko 27). Gumb »Podrobno« ima enako funkcijo, kot v funkciji plačilo e-računov.

Zaradi sčasoma vedno obširnejšega seznama smo banki že predlagali optimizacijo te funkcije. Možnosti za optimizacijo funkcije so:

- enovrstični prikaz podatkov o e-računu,
- omejitev števila prikazanih e-računov (statična številka ali omejitev na časovno obdobje),
- dodan iskalnik pod prikazom e-računov in
- odstranitev kategorije brisani (možnost prikazovanja e-računa ob iskanju po statusu brisan).

PLAČILA: Plačani in ostali e-računi

[Pomoč](#)

Plačani

Naziv izdajatelja	ZAVAROVALNICA TEST D.D.	IBAN izdajatelja	SI56340001001371328
Znesek računa	1,09 EUR	Naslov izdajatelja	CELOVŠKA CESTA 6C
Rok plačila	17.12.2010	Datum izdaje	26.11.2010
Namen plačila	posebna položnica-da vidimo 1	Številka e-računa	6
Podrobno			

Naziv izdajatelja	ZAVAROVALNICA TEST D.D.	IBAN izdajatelja	SI56340001001371328
Znesek računa	6,50 EUR	Naslov izdajatelja	CELOVŠKA CESTA 6C
Rok plačila	03.12.2010	Datum izdaje	30.11.2010
Namen plačila	posebna ploščnica za LAURO	Številka e-računa	PP-1
Podrobno			

Direktne obremenitve

Naziv izdajatelja	ZAVAROVALNICA TEST D.D.	IBAN izdajatelja	SI56340001001371328
Znesek računa	11.998,80 EUR	Naslov izdajatelja	CELOVŠKA CESTA 6C
Rok plačila	18.11.2010	Datum izdaje	17.11.2010
Namen plačila	Plačilo računa 2	Številka e-računa	2
Podrobno			

Brisani

Naziv izdajatelja	TESTNO PODJETJE D.D.	IBAN izdajatelja	SI56340001000960339
Znesek računa	25,92 EUR	Naslov izdajatelja	TESTNA 17
Rok plačila	17.11.2010	Datum izdaje	15.11.2010
Namen plačila	Plačilo računa 3-3	Številka e-računa	3-3
Podrobno			

Statusi e-računov ne odražajo statusa dejanskega plačila. Prosimo, preverite dejansko plačilo e-računa v prikazu prometa na računu oz. v primeru neizvedbe računa v Seznamu neizvedenih plačil.

[Nazaj](#)

Slika 27: Prikaz plačanih in ostalih e-računov.

4.2.3 Iskanje e-računov

Iskalnik e-računov nam omogoča, da poiščemo vse zelene e-račune v sistemu. Iskanje je možno po vseh parametrih, ki so na sliki 28. Rezultati so prikazani enako, kot v funkciji plačilo e-računov.

Gumb »Išči« nam omogoča iskanje izdajatelja, kot pri oddaji vloge za e-prijavo/e-odjavo.

PLAČILA: Iskanje e-računov

[Pomoč](#)

*Račun prejemnika:

IBAN izdajatelja:

Naziv izdajatelja: [Išči](#)

IBAN prejemnika sredstev pri plačilu:

Naziv prejemnika sredstev pri plačilu:

Datum izdaje od:

Datum izdaje do:

Rok plačila od:

Rok plačila do:

Znesek od:

Znesek do:

Valuta:

Namen plačila:

Plačilna metoda:

Opomba:

Referenca:

Status:

[Nazaj](#) [Potrdi](#)

Slika 28: Prikaz iskalnika za e-račune.

5 Zaključek

V diplomskem delu smo se posvetili elektronskemu poslovanju in njegovemu napredku, specifično spletnemu bančništvu in novi funkciji znotraj le tega, e-računu. Skozi diplomsko delo smo predstavili vsebinsko plat e-računov, njihove prednosti in namen, nato smo opisali tehnologije potrebne za implementacijo e-računov v spletno bančništvo in prikazali končno funkcionalnost.

Za razvoj rešitve je bilo potrebno poznavanje veliko različnih tehnologij. Za nekaj dodatnih tehnologij je bil razlog predvsem zastarelost Visual Basic 6 okolja oz. klasičnega ASP-ja, ki je bil razvit pred uveljavitvijo spletnih storitev in posledično nudenju slabe podpore sodobnim tehnologijam. Za veliko število tehnologij je kriva tudi kompleksnost samih spletnih storitev, ki morajo biti definirane z mnogimi standardi (SOAP, WSDL itd.), da so lahko interoperabilne.

Z uspešno implementacijo smo omogočili uporabnikom spletnega bančništva za fizične osebe uporabo e-računov in tako poenostavljeno poslovanje. Trenutno je rešitev na voljo omejenemu številu uporabnikov oz. je v fazi testne produkcije, vendar se v kratkem seli v produkcijsko okolje za vse uporabnike.

Ob izdelovanju dopolnitve se je predvsem razmišljalo o zmožnostih spletnih storitev, ki jih ponuja Halcom d.d.. Namreč nismo imeli podatka koliko klicev na sekundo lahko sprejme sama spletna storitev, da lahko še vedno pošilja oz. sprejema zahteve. Realne rezultate bomo pridobili šele, ko bo rešitev v polni produkciji in bo več klicev usmerjenih na spletno storitev istočasno. Zaradi pomanjkanja informacij o zmožnostih spletnih storitev se je pojavila ideja, da bi ustvarili lastno bazo s podatki o e-računih, ki bi jo osveževali z avtomatskimi klici na spletne storitve enkrat na dan. Tako bi se izognili klicem spletne storitve v funkcijah za plačilo e-računov (edini klic, ki bi ostal je klic za spremembo statusa e-računa ob plačilu). S tem bi omogočili vsakemu posamezniku hitrejši dostop do podatkov, saj bi lahko izboljšali iskalne pogoje za podatke in posledično optimizirali prikaz in pridobivanje podatkov. Vendar je taka rešitev zahtevnejša, saj zahteva prilagoditev relacijske baze oz. podatkovnega modela. Prav tako je število podatkov veliko, kar pomeni veliko breme za samo relacijsko bazo in njen strežnik. Banka se je posledično odločila za direktno komunikacijo s spletnimi storitvami glede na zahteve samih uporabnikov.

Literatura in viri

- [1] D. Lesjak, T. Klojčnik, B. Lesjak, R. Lampe, V. Sulčič, Pravna informatika - zapiski predavanj, Maribor: Pravna fakulteta Univerze v Mariboru. Dostopno na: http://www.pf.uni-mb.si/pravna_informatika/.
- [2] (2011) Wikipedia: Electronic Bussiness. Dostopno na: http://en.wikipedia.org/wiki/Electronic_business.
- [3] A. Kovačič, A. Groznik, M. Ribič, Temelji elektronskega poslovanja, Ljubljana: Ekonomska fakulteta, 2009.
- [4] (2011) Push-pull strategy - Wikipedia. Dostopno na: http://en.wikipedia.org/wiki/Push%E2%80%93pull_strategy.
- [5] H. Huskič, Spletne banke v Sloveniji s posebnim poudarkom na banki DIBA, diplomsko delo, Trbovlje, 2009.
- [6] (2011) THE INDIAN INTERNET BANKING JOURNEY Dostopno na: <http://www.icmrindia.org/free%20resources/casestudies/banking3.htm>.
- [7] B. Matjašec, Internetno bančništvo prebivalstva na primeru izbranih slovenskih bank, diplomsko delo, Lendava, 2009.
- [8] V. Vehovar, T. Kozic, K. Prevodnik, L. Kogovšek, E-bančništvo 2009/1(#82), Ljubljana, marec 2009. Dostopno na: <http://www.ris.org//uploadi/editor/1270625823E-bancnistvo2009-1.pdf>.
- [9] Združenje bank Slovenije, Poslovni model bank za uvedbo e-računa v Sloveniji projekt e-račun, ver 1.6, februar 2009.
- [10] (2011) European Commission » Internal Market » Payment services» e-Invoicing, Dostopno na: http://ec.europa.eu/internal_market/payments/einvoicing/index_en.htm
- [11] (2011) HTML - Wikipedia. Dostopno na: <http://en.wikipedia.org/wiki/HTML>.
- [12] (2011) HTML 4.01 Specification. Dostopno na: <http://www.w3.org/TR/html401/>.
- [13] (2011) Cascading Style Sheets - Wikipedia. Dostopno na : <http://en.wikipedia.org/wiki/CSS>.
- [14] (2011) CSS Tutorial. Dostopno na: http://www.w3schools.com/css/css_intro.asp.
- [15] (2011) JavaScript - Wikipedia. Dostopno na: <http://en.wikipedia.org/wiki/Javascript>.
- [16] (2011) JavaScript Tutorial. Dostopno na: <http://www.w3schools.com/js/default.asp>.
- [17] (2011) Visual Basic - Wikipedia. Dostopno na:

http://en.wikipedia.org/wiki/Visual_Basic.

- [18] (2011) Visual Basic 6 (VB6) Data Types, Modules and Operators. Dostopno na: <http://visualbasic.freetutes.com/learn-vb6/lesson2.html>.
- [19] J. Ablan, T. Barnaby, P. Boutquin, M. Brown, C. Caison, R. Harrison, J. Hasan, M. Reynolds, D. Slosberg, M. L. Thomas, T. Willis, P. Wilton, Professional Visual Basic 6 Web Programming, Birmingham: Wrox Press, 1999.
- [20] D. Hunter, J. Rafter, J. Fawcett, E. van der Vlist, D. Ayers, J. Duckett, A. Watt, L. McKinnon, Beginning XML 4th Edition, Indianapolis: Wiley Publishing, Inc., 2007.
- [21] (2011) XSLT - Wikipedia. Dostopno na: <http://en.wikipedia.org/wiki/XSLT>.
- [22] B. Dournaee, XML Security, New York: McGraw-Hill, 2002.
- [23] (2011) Web service - Wikipedia. Dostopno na: http://en.wikipedia.org/wiki/Web_service.
- [24] (2011) Web Services Tutorial . Dostopno na: <http://www.w3schools.com/webservices/default.asp>.
- [25] (2011) Web Services Description Language - Wikipedia. Dostopno na: <http://en.wikipedia.org/wiki/WSDL>.
- [26] (2011) Web Service Definition Language (WSDL). Dostopno na: <http://www.w3.org/TR/wsdl>.
- [27] (2011) SOAP Tutorial. Dostopno na: <http://www.w3schools.com/soap/default.asp>.
- [28] (2011) Specifikacija vmesnika za izmenjavo E-računov v elektronski banki Hal E-Bank. Dostopno na: http://www.halcom.si/halcom_exp/UserFiles/File/Specifikacija_za_izdajo_e-racunov.pdf.