Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00510/2010
Datum: 05.04.2010

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

Kandidat: **TOMAŽ KARER**

Naslov: **RAZVOJ INFORMACIJSKEGA SISTEMA ZA ELEKTRONSKI PRISTOP K BANČNIM STORITVAM**

DEVELOPMENT OF INFORMATION SYSTEM FOR ONLINE BANKING SERVICES

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Opišite značilnosti elektronskega poslovanja v bančnem sektorju in najpomembnejše izzive na tem področju. Predstavite rešitve, ki jih uporablja Abanka Vipa in opišite razvoj aplikacije za elektronski pristop h krovnemu skladu Abančne DZU. Iz opisa naj bodo razvidne uporabniške zahteve in omejitve, orodja, uporabljena pri izdelavi, postopek izdelave, testiranje in predaja v uporabo.

Mentor:
prof. dr. Viljan Mahnič

Dekan:
prof. dr. Franc Solina

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Tomaž Karer

Razvoj informacijskega sistema za elektronski pristop k bančnim storitvam

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: izr. prof. dr. Viljan Mahnič

Ljubljana, 2010

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Tomaž Karer,

z vpisno številko 63020226,

sem avtor/-ica diplomskega dela z naslovom:

Razvoj informacijskega sistema za elektronski pristop k bančnim storitvam z opisom primera na e-pristopu h krovnemu skladu Abančne DZU

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom izr. prof. dr. Viljana Mahniča,
- 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 1.7.2010 Podpis avtorja/-ice:

Zahvala

Zahvaljujem se mentorju, izr. prof. dr. Viljanu Mahniču, za vso pomoč in nasvete pri izdelavi diplomske naloge. Prav tako se zahvaljujem vsem sodelavcem v podjetju Abanka Vipa, d. d., ki so pripomogli k nastanku naloge.

Zahvaljujem se staršem za možnost in podporo ob študiju ter potrpežljivost. Nenazadnje se zahvaljujem Tjaši za vso spodbudo.

Kazalo

Povzetek **1**

Abstract **2**

1 Uvod **3**
1.1 Predstavitev problema 3
1.2 Predstavitev ciljev 4

2 Predstavitev bančnega okolja **5**
2.1 Predstavitev banke 5
2.1.1 Slovenski bančni prostor 5
2.1.2 Abanka 6
2.1.2.1 Služba razvoja informatike 6
2.1.3 ADZU - Abančna družba za upravljanje s skladi 6
2.2 Uvajanje elektronskega poslovanja 7
2.2.1 Elektronsko poslovanje v Sloveniji 7
2.2.2 Elektronsko poslovanje z Abanko 8
2.2.2.1 Bankomati 8
2.2.2.2 AbaSMS 8
2.2.2.3 Abanet in E-račun 8

3 Opis obstoječe aplikacije, informacijskega sistema **9**
3.1 Opis informacijskega sistema Abanke 9
3.1.1 Informacijski sistem Abanke 9
3.2 Opis obstoječe aplikacije 11
3.2.1 Zaledni sistem 11
3.2.1.1 Dokument XML pogodbe 12
3.2.1.2 Konceptualni model 12
3.2.2 Zunanji moduli 17
3.2.2.1 Podpisovanje 17

3.2.2.2 Arhiviranje 17
3.2.3 Centralni pregled naročil 17

4 Uporabniške in tehnološke zahteve, omejitve, pravni vidik 19
4.1 Zahteve 19
4.1.1 Uporabniške zahteve 19
4.1.1.1 Postopek obdelave naročila 20
4.1.2 Tehnološke zahteve 20
4.1.3 Prenova sistema Arhiviranje 21
4.1.4 Diagram bodoče postavitve 21
4.2 Varnost 22
4.2.1 Varnost na strani uporabnika 23
4.2.2 Kvalificirano digitalno potrdilo, podpisna komponenta uporabnika 23
4.3 Pravni vidik 24
4.3.1 Pravna podlaga, zakoni 24
4.3.2 Sledljivost 25

5 Predstavitev uporabljenih tehnologij, orodij 26
5.1 Oracle 26
5.1.1 Sistem za upravljanje z bazami podatkov Oracle 26
5.1.2 PL/SQL 27
5.2 TOAD 27
5.3 Delphi 27
5.3.0.1 Delphi IDE 27
5.3.0.2 Delphi Pascal 28
5.4 XML 28
5.4.1 Standard XML 28
5.4.2 Vizualizacija XML 29
5.5 WebServices 30

6 Izdelava aplikacije 33
6.1 Izdelava 34
6.1.1 Implementacija sprememb 34
6.1.1.1 Oracle, PL/SQL 34
6.1.1.2 Delphi 35
6.2 Testiranje 37
6.2.1 Razvojno testiranje posameznih modulov, funkcij 37
6.2.2 Testiranje pred predajo v uporabo 38

6.3 Predaja v uporabo 39
6.3.1 Objava na produkcijskih strežnikih 39

7 Delovanje aplikacije **41**
7.1 Delovanje skozi oči uporabnika 42
7.2 Nadzorovanje delovanja 46
7.2.1 Nadzorovanje delovanja skozi nadzorne module banke . . 46
7.2.1.1 Centralni pregled naročil 46

8 Zaključne misli in ugotovitve **49**
8.1 Nadaljnji razvoj 50

Seznam slik **51**
Literatura **53**

Povzetek

V diplomski nalogi je opisan postopek razvoja in nadgradnje informacijskega sistema za elektronski pristop k storitvam Abanke. Najprej je predstavljena obstoječa aplikacija, ki skrbi za prejemanje in obdelavo naročil bančnih storitev prek interneta, nato pa njena nadgradnja z naročilom pristopa h krovnemu skladu Abančne DZU.

Na kratko je predstavljeno podjetje Abanka kot naročnik in uporabnik sistema. Predstavljene so uporabniške zahteve ter tehnološke, pravne in poslovne omejitve pri izdelavi.

Predstavljen je informacijski sistem banke, na katerega se aplikacija navezuje, in orodja ter tehnologije, uporabljene pri izdelavi.

V nadaljevanju je opisan sam postopek izdelave aplikacije, testiranje in predaja v uporabo.

Na koncu je prikazano delovanje aplikacije, kot jo vidijo uporabniki, težave pri razvoju in ideje za nadaljnji razvoj.

Ključne besede:

elektronsko poslovanje, elektronsko podpisovanje, razvoj informacijskega sistema

Abstract

This diploma paper describes the process of developing and upgrading the information system for electronic provision of services by Abanka. First the existing application is presented which is designed for receiving and processing banking service orders placed via the internet, and then its upgrade involving orders for entering the umbrella fund managed by Abančna DZU is introduced.

A short presentation is given of the company Abanka acting as both the client and user of the system. User requirements as well as technological, legal and commercial limitations encountered in the process of development are specified.

Particulars of the bank's information system to which the application concerned is linked as well as the tools and technologies used in its development are given.
Further on, the very procedure of developing the application, its testing and putting into operation are described.

By way of conclusion, operation of the application from the users' point of view is illustrated, the problems encountered during the phase of development and some ideas for further development are given.

Key words:

electronic commerce, electronic signatures, information system development

Poglavje 1

Uvod

1.1 Predstavitev problema

Bliskovit razvoj elektronskih načinov komuniciranja je v poslovanje bank prinesel velike spremembe. Od bankomatov, prek telefonskega bančništva do internetnega bančništva so bile pregovorno konservativne banke primorane vlagati sredstva v razvoj novih načinov poslovanja tako znotraj banke same kot z drugimi bankami in s svojimi strankami. Možnost komunikacije s svojo banko iz udobja svojega naslanjača v današnjem času tudi v slovenskem okolju razumemo kot nekaj samoumevnega. Tradicionalne slovenske banke te možnosti ponujajo že kar nekaj časa, v zadnjem času pa smo priče tudi tako imenovanim "virtualnim bankam", ki imajo svoje poslovalnice izključno na internetnih straneh.

V večini bank, ki delujejo na slovenskem območju, je prek interneta mogoče opravljati najrazličnejše storitve, vendar je stopnja avtomatizacije izvedbe storitev različna. Naprednejše storitve, kot so naročilo izrednega limita, vezava depozita, pristop k storitvam mobilnega bančništva ipd., še vedno zahtevajo ročno izvedbo in niso povsem avtomatizirane.

Tak način obdelovanja se pokaže kot nezanesljiv in počasen. Naročila, še vedno vezana na papir in podvržena človeški napaki, se izvršijo z napako, v najslabšem primeru se celo izgubijo, kar velikokrat pripelje do nezadovoljstva strank. Tako v veliki meri ostaja neizkoriščen potencial sodobnih načinov poslovanja.

1.2 Predstavitev ciljev

V diplomski nalogi bom pokazal, kako smo se s tem problemom spoprijeli v Abanki. V nalogi je predstavljen razvoj informacijskega sistema za v karseda veliki meri avtomatizirano obdelovanje naročil, oddanih prek spletne banke (sistem naročil).

Pokazal bom, kako je potekal postopek razvoja in nadgradnje informacijskega sistema, ki skrbi za obdelavo in izvršitev naročil naprednejših storitev. Ker je sistem podvržen najrazličnejšim zahtevam in omejitvam, pravnim, poslovnim in tehnološkim, so te okoliščine opisane.

Diplomska naloga je sestavljena iz treh delov. V prvem, ki zajema drugo in tretje poglavje, je opisano okolje - slovenski bančni prostor in Abanka sama -, v katerem informacijski sistem nastaja. V drugem delu, ki zajema četrto in peto poglavje, so predstavljene uporabniške in tehnološke zahteve, opisane so pravne zahteve in omejitve, zajeta je tudi pomembna komponenta poslovanja na daljavo - varnost. V tem delu so predstavljene tudi tehnologije in orodja, ki so bile uporabljene pri izdelavi. V zadnjem, tretjem delu, ki obsega šesto, sedmo in osmo poglavje, pa je zajeta izdelava sama, priprave nanjo in različne ugotovitve po zaključku projekta.

Poglavje 2

Predstavitev bančnega okolja

2.1 Predstavitev banke

Abanka Vipa, d. d., je svoje delovanje začela leta 1955 kot podružnica Jugoslovanske banke za zunanjo trgovino. Leta 1977 se je preimenovala v Jugobanko - Temeljno banko Ljubljana, decembra 1989 pa se je z izstopom iz sistema Jugobanke preoblikovala v Abanko, delniško družbo.

Povsem samostojno je začela poslovati leta 1990, leta 2002 pa je prišlo do združitve z Banko Vipa, d. d., in posledično do preimenovanja v Abanko Vipo, d. d. Pod tem imenom in v tej obliki tudi danes deluje kot ena največjih bank v Sloveniji [1].

2.1.1 Slovenski bančni prostor

Kratka, a pestra zgodovina samostojne Slovenije je za slovenske banke pripravila veliko izzivov. Sprememba politično-ekonomskega sistema je pripeljala prvi val sprememb: lastna valuta, privatiziranje družbene lastnine ... Vstop v Evropsko in Monetarno unijo pa je prinesel drugi val: uveljavljanje nove, skupne evropske valute, posledično širitev trgov in prihod tujih bank z agresivnih pristopom ... Banke so se spremembam prilagajale na različne, bolj ali manj uspešne načine. Leta 2008 je v Sloveniji delovalo 23 bank, združenih v ZBS - Združenje bank Slovenije [2].

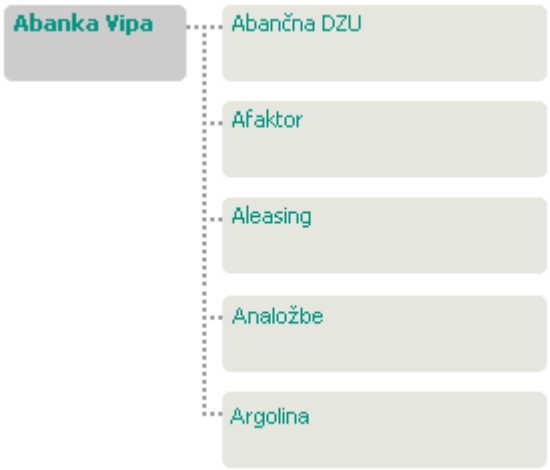
2.1.2 Abanka

2.1.2.1 Služba razvoja informatike

Oddelek, v katerem delam, Služba razvoja informatike, je organizacijsko uvrščen v Področje informatike. Znotraj širšega oddelka Področja informatike skrbimo za notranji razvoj in vzdrževanje informacijskih sistemov. Za razvoj sistemov, razvitih zunaj banke, skrbi Služba skrbništva informacijskih sistemov, s katero sem pri razvoju sistema naročil tesno sodeloval.

2.1.3 ADZU - Abančna družba za upravljanje s skladi

Abančna družba za upravljanje s skladi (v nadaljevanju ADZU) je hčerinsko podjetje Abanke. Organizacijsko je uvrščena med ostale odvisne družbe Abanke (Slika 2.1). Glavno poslanstvo družbe je upravljanje investicijskih skladov [7]. Svoje delovanje je začela leta 1994, ko je pridobila dovoljenje za opravljanje dejavnosti upravljanja investicijskih skladov.



Slika 2.1: Abančne odvisne družbe

2.2 Uvajanje elektronskega poslovanja

2.2.1 Elektronsko poslovanje v Sloveniji

V tujini so različni načini elektronskega poslovanja s strankami v uporabi že dolgo časa. Prvi bankomati so se po razvitem zahodnem svetu začeli pojavljati v 60. letih prejšnjega stoletja. Banke so nato svoje storitve elektronskega poslovanja razširile najprej s telefonskim poslovanjem in v zadnjem času še s poslovanjem prek računalnika, sprva prek telefonske linije in kasneje prek interneta. V Sloveniji tako praktično vse banke svojim strankam nudijo možnost poslovanja prek interneta.

Po podatkih projekta Raba interneta v Sloveniji [3] je leta 2009 elektronsko bančništvo uporabljalo 22,9 % populacije starih 10-74 let, kar je 3,1 odstotne točke več kot leta 2008. V primerjavi s preostalo Evropo, državami EU27 (27 držav Evropske unije), kjer je v zadnjih mesecih leta 2008 elektronsko bančništvo uporabljalo 27 % populacije, je ta odstotek znatno nižji. Delež uporabnikov v različnih državah sicer močno variira, a vseeno lahko rečemo, da je v Sloveniji elektronsko bančništvo razmeroma nerazvito.

Kaj je vzrok tega? V raziskavi, opravljeni v Združenih državah Amerike [4], ki z 58 % populacije uporabnikov elektronske banke sodi med najrazvitejše, so faktorji, ki vplivajo na odločitev za spletno banko, naslednji:

- dosegljivost - 72 %,
- osveženost spletne strani z ažurnimi informacijami - 71 %,
- urejenost in preglednost spletne strani - 65 %,
- hitrost spletne strani - 65 %,
- zanesljivost - 65 %,
- jasna uporaba jezika - 52 %,
- pomoč in funkcionalnost - 44 %.

Žal že kratek obisk nekaterih spletnih bank pokaže, da se marsikateri od zgornjih faktorjev zanemarja, kar vsekakor lahko razumemo kot enega od vzrokov za neuveljavljenost spletnih bank v slovenskem prostoru.

2.2.2 Elektronsko poslovanje z Abanko

Abanka svojim strankam ponuja več načinov za elektronsko poslovanje. Poleg najstarejšega in najosnovnejšega, bankomatov, ponuja še mobilno storitev AbaSMS, spletno banko Abanet in v povezavi s slednjo še E-račun.

2.2.2.1 Bankomati

V današnjem času vsem poznani in vsepovsod prisotni bankomati ne potrebujejo podrobnejšega opisa. Kljub razmeroma stari tehnologiji pa se njihov razvoj ni ustavil. V začetku njihovega uvajanja so nudili samo najosnovnejši storitvi - dvig gotovine in informacijo o stanju. Danes pa nudijo tudi naprednejše storitve, kot so izpis prometa, plačilo posebnih položnic, polog gotovine s takojšnjim knjiženjem na račun, zamenjava osebne številke PIN (ang. personal identification number), nakup vrednostne kartice za mobilne telefone ...

2.2.2.2 AbaSMS

AbaSMS je storitev, s katero ima uporabnik mobilnega telefona prek SMS (ang. short message service) sporočil stalen pregled nad stanjem in dogajanjem na svojem osebnem računu.

2.2.2.3 Abanet in E-račun

Spletna banka Abanet strankam nudi možnosti poslovanja prek spletnega brskalnika. Glavna prednost je možnost opravljanja različnih storitev brez obiska fizične poslovalnice, za večino storitev pa ponuja nižje stroške provizije. Tako nudi pregled vseh podatkov o osebnih računih, limitih, plačilih, plačilnih karticah, kreditih, varčevanjih in naložbah. Z uporabo kvalificiranega digitalnega potrdila (certifikata), ki nudi možnost zakonsko veljavnega elektronskega podpisa, pa ponuja tudi možnost oddaje naročil za različne storitve, kot so izredni limit, varčevalni račun in tudi naložbe v podsklade ADZU.

V povezavi s spletno banko je uporabnikom na voljo storitev E-račun, ki v standardni elektronski obliki zamenjuje račun na papirju. Tehnologija poenostavlja tako izdajanje kot plačevanje računov in tako omogoča časovne in stroškovne prihranke pri obeh udeleženih straneh pri transakciji.

Poglavje 3

Opis obstoječe aplikacije, informacijskega sistema

3.1 Opis informacijskega sistema Abanke

Informacijski sistem Abanke je v večjem delu plod notranjega razvoja. Zaradi občutljivosti podatkov, potrebe po nadzoru razvoja, obvladovanju stroškov in drugih razlogov, obstaja močna težnja k razvoju karseda velikega dela programske opreme znotraj banke same. Marsikateri sistemi so zaradi različnih razlogov seveda prepuščeni v izdelavo zunanjim sodelavcem, a kljub temu glavnina sistemov ostaja razvitih interno.

3.1.1 Informacijski sistem Abanke

Podatkovna baza, kot osrednji del informacijskega sistema v Abanki, je Oracle, različica 10g. Podatki so glede na vsebino razdeljeni na različne sheme, ki hranijo tudi programsko kodo v obliki shranjenih procedur (ang. stored procedures). Podatkovne baze so pravzaprav tri - razdeljene na razvojno, testno in produkcijsko okolje. Produkcijsko okolje vsebuje aktualne podatke in aplikacije, ki so trenutno v praktični uporabi v banki. Testno okolje je prirejena kopija produkcijskega, z zakamufliranimi občutljivimi podatki po količini in obliki podatkov karseda enako produkcijskemu. Razvojno okolje je po količini podatkov manjše, uporabno za razvijalce, ki si ga lahko poljubno polnijo s podatki, primernimi za razvoj aplikacij. V sistemu, kjer je količina produkcijskih podatkov zelo velika, kot na primer število denarnih transakcij, in kjer zakonska določila zahtevajo sledljivost in večletni arhiv, je obstoj treh baz nujnost.

Majhno in hitro razvojno okolje razvijalcem omogoča hiter razvoj in preizkušanje aplikacije. Kljub testiranju v razvojnem okolju pa lahko marsikatere napake zaradi večjega števila različnih, resnično življenjskih podatkov ugotovimo šele v testnem okolju. Na testnem okolju lahko prav tako izvajamo performančne optimizacije, kar v razvojnem ni možno. Testno okolje, kot kopija produkcijskega, služi tudi kot testni poligon, na katerem se pokažejo napake, ki so posledica dolgotrajnega delovanja avtomatiziranih aplikacij - po možnosti še preden se taka napaka pojavi v produkcijskem okolju.

Do podatkov nenehno dostopa množica povsem avtomatiziranih aplikacij kot tudi aplikacij, ki predstavljajo vmesnik za uporabnika. Razvijalci z orodji, kot je npr. TOAD, lahko do razvojne in testne baze dostopamo neposredno. Operacijski sistem, ki teče na večini računalnikov, je Microsoft Windows XP SP3. Ker je standardno okolje za razvoj aplikacij v banki Delphi za operacijski sistem Windows, je večina aplikacij razvita samo za to platformo. Za osveževanje verzij in pravilno distribucijo tistih aplikacij, pri katerih je osveževanje nujno, skrbi poseben avtomatiziran sistem ATUPDATE. Večina uporabnikov uporablja aplikacijo ATBO (Abančno transakcijsko bančno okence), ki predstavlja osrčje bančnega informacijskega sistema in je osrednje mesto za večino operacij, potrebnih v banki. ATBO je zasnovan modularno, eden od teh modulov je tudi sistem za nadzor naročil - Centralni pregled naročil.

Za izvajanje avtomatiziranih procesov se uporablja notranje razvita aplikacija ATServer (Abančni transakcijski strežnik). Za nadzor delovanja se uporablja aplikacija ATClient (Abančni transakcijski odjemalec), ki omogoča enostaven nadzor in upravljanje vseh avtomatiziranih procesov.

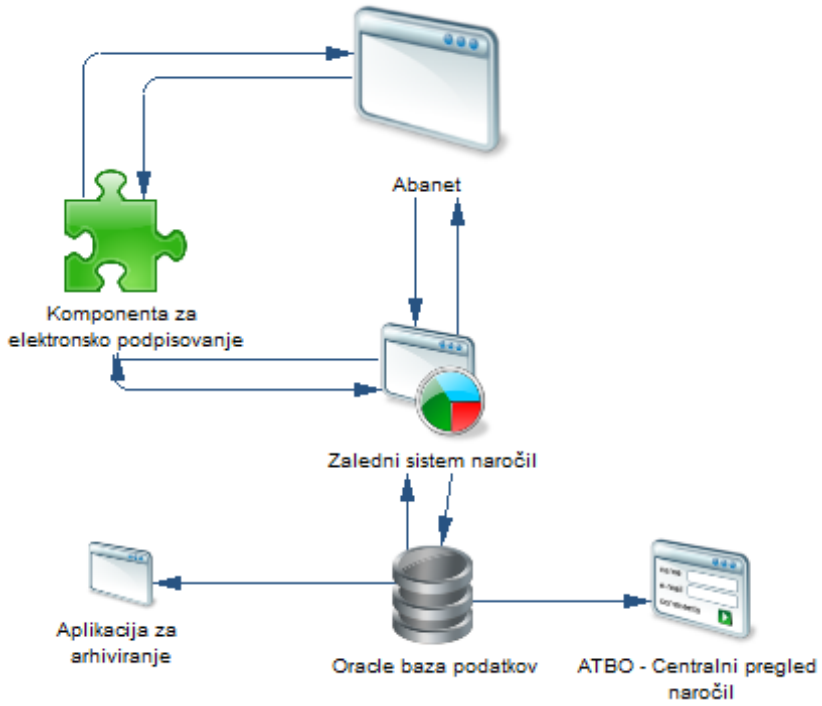
Ker do občutljivih podatkov dostopajo tudi aplikacije, ki jih izdelujejo zunanji izvajalci, se za dostop uporabljajo posebne metode. Takšne aplikacije tako nimajo neposrednega dostopa do podatkov. Do teh na omejen način dostopajo prek baznih funkcij in posebnih pogledov (ang. view), katerih pravice za dostop so specificirane na točno določene zunanje aplikacije. Primer tega je Abanet, izdelan s sodelovanjem Abanke in podjeta SRC sistemske integracije, d. o. o. Ta za razvoj aplikacije uporablja posebne funkcije in poglede, omejene izključno na podatke, ki so tem pogledom avtorizirani. Tak način zagotavlja nemoten razvoj in delovanje sistemov, hkrati pa zagotavlja tajnost podatkov.

3.2 Opis obstoječe aplikacije

Obstoječa aplikacija za obdelovanje naročil prek Abaneta je sestavljena iz več med seboj ločenih, a sodelujočih modulov. Prvi in za uporabnika edini viden je Abanet. Abanet v sodelovanju z zalednim sistemom naročil in uporabniškim vnosom zgenerira pogodbo v obliki dokumenta XML (ang. eXtensible markup language). Ta je vsebinsko identična papirnati pogodbi, dodatno pa vsebuje še vse tehnične podatke, potrebne za izvršitev naročila. Zaledni sistem naročil nato obdela naročilo, s pomočjo zunanjih modulov se pogodba podpiše in izvrši.

3.2.1 Zaledni sistem

Zaledni sistem predstavlja osrčje sistema naročil. V celoti implementiran v jeziku PL/SQL (ang. Procedural Language/Structured Query Language) skrbi za posredovanje podatkov Abanetu, preverjanje uporabniškega vnosa, obdelavo pogodbe in izvršitev. S pomočjo zunanjih modulov skrbi za podpisovanje in arhiviranje dokumentov ter nadzor celotnega procesa (Slika 3.1).



Slika 3.1: Diagram postavitve sistema

3.2.1.1 Dokument XML pogodbe

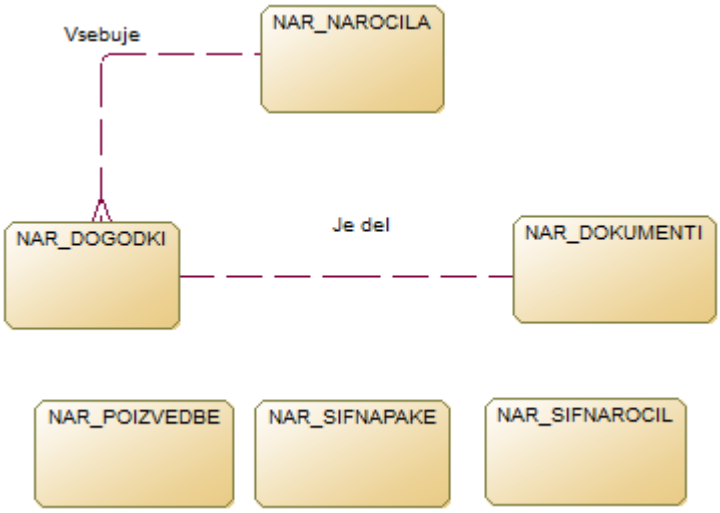
Dokument XML, ki predstavlja pogodbo, je zasnovan tako, da je enakovreden papirnati pogodbi. Skozi celoten proces delovanja aplikacije se dopolnjuje. Tako se najprej vhodnim podatkom uporabnika (element XML <NAROCILO>) dodajo podatki, potrebni za izvršitev (<POGODBA>) in vizualizacijo pogodbe (<TRANSFORMATION>). Po uporabnikovem podpisu (<SIGNATURES><SIGNATURE id="sign1">) se po ročni potrditvi ali zavrnitvi (<PrijavaData><STATUS>) podpiše še s strani ADZU-ja (<SIGNATURES><SIGNATURE id = "sign2">).

```
<?xml version="1.0" encoding="WINDOWS-1250"
  standalone="no"?>
<?xml-stylesheet type="text/xml" href="#trans1"?>
<Document>
 <Data>
  <PogodbaData id="data1">
   <Narocilo>
   %Podatki uporabnika
   </Narocilo>
   <Pogodba>
   %Podatki za izvrsitev
   </Pogodba>
  </PogodbaData>
 </Data>
 <Transformation>
 %Podatki o vizualizaciji
 </Transformation>
 <Signatures>
 %Elektronski podpisi
 </ds:Signature>
</Document>
```

3.2.1.2 Konceptualni model

Zaledni sistem je v grobem sestavljen iz treh delov: iz podatkovnih tabel, paketa s funkcijami ter vhodno-izhodnih funkcij in pogledov, namenjenih komunikaciji z zunanjimi moduli.

Podatkovni model je sestavljen iz šestih tabel (Slika 3.2).



Slika 3.2: Podatkovni model

Pripravljenih je pet pogledov, ki prikazujejo naročila vsak v svojem statusu procesa:

- NARV_EBNCAKVRSTA
- NARV_EBNIZVRSENI
- NARV_EBNPOTEKLI
- NARV_EBNPREKLICANI
- NARV_EBNZAVRNJENI

z enotno strukturo podatkov:

```
IDNAROCILA          VARCHAR2(16)
NAROCILOVRSTA       VARCHAR2(2)
IDKOM               VARCHAR2(16)
IDNAPAKE            VARCHAR2(10)
NAPAKA              VARCHAR2(200)
DATNAR              DATE
```

```
STATUS                  VARCHAR2 (1)
FAZA                    VARCHAR2 (2)
DATSPR                  DATE
VSEBINA                 XMLTYPE
DATARH                  DATE
```

Vhodne funkcije oziroma procedure, ki služijo kot ovojnice za nekatere funkcije v paketu NAROCILA_PCK, so naslednje:

- NAR_EBNVPISINAROCILO
- NAR_EBNNAROCILO
- NAR_EBNPREKLICINAROCILO
- NARF_EBNPREVERINAROCILO

Prva izvede vpis naročila v sistem naročil. Ob uspešno izvedenem vpisu ustvari enolično identifikacijsko številko naročila ter vrne podatke za pogodbo v podpis naročniku, ob neuspešnem pa identifikacijsko številko napake in sporočilo o napaki.

```
PROCEDURE EBNNAR_VPISINAROCILO (
  aNAROCILO             IN  XMLTYPE,
  aIDNAROCILA           OUT VARCHAR2,
  aPOGODBA              OUT XMLTYPE,
  aIDNAPAKE             OUT VARCHAR2,
  aNAPAKA               OUT VARCHAR2
  )
```

Naslednja, ki kot vhodni parameter dobi pogodbo, podpisano s strani uporabnika, pogodbo podpiše še s strani banke in poskrbi za uspešno izvedbo naročila.

```
PROCEDURE EBNNAR_NAROCILO (
  aIDNAROCILA           IN  VARCHAR2,
  aNAROCILO             IN  XMLTYPE,
  aPOGODBA              OUT XMLTYPE,
  aIDNAPAKE             OUT VARCHAR2,
  aNAPAKA               OUT VARCHAR2
  )
```

Procedura NAR_EBNPREKLICINAROCILO v vseh fazah obdelave ponuja možnost prekinitve in preklica izvajanja naročila.

```
PROCEDURE NAR_EBNPREKLICINAROCILO (
  aIDNAROCILA          IN  VARCHAR2,
  aIDNAPAKE            OUT VARCHAR2,
  aNAPAKA              OUT VARCHAR2
  )
```

Zadnja, pomožna funkcija se uporablja za vračanje podatkov ob poizvedovanju Abaneta. Poizvedbe, imenovane preverjanja, se uporabljajo, kadar je še pred oddajo naročila potrebno preverjati različne kriterije oziroma dostopati do dodatnih podatkov.

```
FUNCTION NARF_EBNPREVERINAROCILO (
  aIDKOM               IN   VARCHAR2, /* identifikacijska
  stevilka uporabnika*/
  aNarociloVrsta       IN   VARCHAR2,
  aPreverjanjeVrsta    IN   VARCHAR2,
  aParametri           IN   XMLTYPE,
)RETURN XMLTYPE
```

Funkcija vrača rezultat v obliki XML in v nekaterih primerih zahteva dodatne parametre, prav tako v obliki XML. Primer preverjanja, ki za podano številko transakcijskega računa:

```
<Parametri>
    <TRK>051000001234567</TRK>
</Parametri>
```

vrne banko računa in njeno kodo:

```
<?xml version="1.0" encoding="windows-1250" ?>
<NARFPreveriNarocilo>
  <NazivBanke>
    <BANKA>
      <NAZIV>ABANKA VIPA D.D. LJUBLJANA</NAZIV>
      <KODA>05</KODA>
    </BANKA>
  </NazivBanke>
  <Napaka>
```

```
    <IDNapake>0</IDNapake>
  </Napaka>
</NARFPreveriNarocilo>
```

Paket NAROCILA_PCK je skupek vseh funkcij in procedur, ki poskrbijo za pravilno obdelavo, podpisovanje in izvršitev ali zavrnitev naročila. Primer je procedura NAR_PRETECENANAROCILA, ki periodično, enkrat dnevno, poskrbi, da se naročila, ki v določenem času (do konca dneva) niso podpisana s strani uporabnika in posledično izvršena, zavrnejo. Za njeno izvajanje je kreiran Oraclov posel (ang. job) z naslednjo skripto, ki zagotovi, da se izbrana procedura izvede enkrat dnevno, ob polnoči:

```
DECLARE
  X NUMBER;
BEGIN
  SYS.DBMS_JOB.SUBMIT
  ( job       => X
   ,what      => 'NAR_PRETECENANAROCILA;'
   ,next_date => to_date('04.06.2010␣00:00:00',
     'dd/mm/yyyy␣hh24:mi:ss')
   ,interval  => 'TRUNC(SYSDATE+1)'
   ,no_parse  => FALSE
  );
COMMIT;
END;
```

Paket NAROCILA_PCK za svoje delovanje uporablja različne interne funkcije informacijskega sistema banke, ki so glede na pripadnost določeni vrsti naročila združene v pakete. Tako so npr. funkcije in procedure za izvedbo naročila izrednega limita združene v paket POGLIMIT_PCK. Taka strukturiranost olajšuje vzdrževanje in nadaljnji razvoj po načelih večnivojskega programiranja.

Vse napake pri izvajanju kakor tudi vse poizvedbe, ki se ne izvedejo uspešno, se hranijo v sistemu poizvedb. S klicem funkcije NARF_NAPAKAPOIZVEDBE, ki je dodan na vseh pomembnih mestih v sistemu, se beležijo vse kritične napake. Te so nadzornikom sistema pregledno na voljo v centralnem pregledu naročil in so v veliko pomoč tako pri reševanju ugovorov strank kakor tudi pri odpravljanju programskih napak.

3.2.2 Zunanji moduli

Zaledni sistem naročil za pravilno delovanje sodeluje s štirimi zunanjimi moduli: Abanet, modul za elektronsko podpisovanje, modul za arhiviranje podatkov in nadzorni modul, Centralni pregled naročil.

3.2.2.1 Podpisovanje

Modul za podpisovanje predstavlja posebna komponenta za elektronsko podpisovanje dokumentov proXSign®, podjetja SETCCE, d. o. o. Deluje s pomočjo sistema javnih in zasebnih ključev in omogoča podpisovanje dokumentov XML in njihovo verifikacijo. Prav tako omogoča tudi časovni žig. S pomočjo posebnega vtičnika omogoča podpisovanje neposredno s spletnega brskalnika. Zaledni sistem naročil do modula dostopa s tehnologijo spletnih vmesnikov.

Ker se do komponente dostopa iz okolja PL/SQL, komunikacija prek spletnih vmesnikov ni tako preprosta kot je npr. v okolju Delphi. Tako je najprej potrebna izdelava izhodnih podatkov v obliki XML, njihova ograditev z ovojnico SOAP in posredovanje s pomočjo Oraclovega paketa UTL_HTTP.

3.2.2.2 Arhiviranje

Arhiviranje pogodb je realizirano s sistemom Arhiviraj.si podjetja Mikrocop. Dokumenti izvršenih naročil se redno pošiljajo v sistem Arhiviraj.si, kjer se hranijo in so na voljo za hiter dostop. V primerjavi s hranjenjem papirnatih dokumentov tak način omogoča hitrejši in cenejši način hranjenja. Prav tako zagotavlja varno in zakonsko skladno hrambo dokumentov.

Za izvajanje arhiviranja se uporablja samostojna aplikacija, izdelana v okolju Delphi.NET. Do sistema zunanjega izvajalca dostopa s tehnologijo spletnih vmesnikov in omogoča dnevno arhiviranje vseh dokumentov na ta dan izvršenih naročil. Pregledovanje arhiviranih dokumentov je na voljo prek spletne aplikacije podjetja Mikrocop.

3.2.3 Centralni pregled naročil

Centralni pregled naročil nadzornikom omogoča nadzorovanje delovanja sistema naročil. Umeščen je v aplikacijo ATBO. V njem je možno spremljati naročila prek različnih faz od začetka do izvršitve oziroma zavrnitve, možno je

pregledovati podrobnosti posameznega naročila in njegov dokument XML ter ugotavljati vzroke za neuspešno izvrševanje naročila prek sistema poizvedb.

Poglavje 4

Uporabniške in tehnološke zahteve, omejitve, pravni vidik

4.1 Zahteve

4.1.1 Uporabniške zahteve

V projektu nadgradnje sistema naročil z možnostjo elektronskega pristopa k podskladom ADZU se kot uporabnik pojmuje ADZU. Njihove glavne zahteve so bile naslednje:

- uvrstitev pristopa k podskladom ADZU v obstoječo aplikacijo Abanet,
- pristop k ADZU se uvrsti v področje Nebančne storitve,
- področje Nebančne storitve mora biti v prikazu ločeno od storitev Abanke,
- stranka se mora strinjati z izjavo o seznanjenosti s storitvijo in izjavo o tveganju,
- v primeru, da stranka ni znan vlagatelj ADZU (še nima odprte nobene pristopne izjave, torej ni bila opravljena osebna identifikacija), se naročilo izvrši šele po ročni potrditvi.

Zadnja zahteva je z možnostjo ročne potrditve predstavljala odmik od trenutnega zaporedja izvajanja. Zaradi tega je bilo potrebno diagram delovanja sistema prilagoditi.

4.1.1.1 Postopek obdelave naročila

Po strankini izbiri pristopa k ADZU Abanet preveri, ali so tehnične zahteve za izvršitev naročila izpolnjene (prisotnot podpisne komponente). S sistemom preverjanj (klici funkcije NARF_EBNPREVERINAROCILO) za prikaz v Abanetu pridobi podatke o stranki in skladih ADZU. Po strankini odločitvi za pristop k sistemu naročil posreduje vhodni dokument XML (procedura EBNNAR_VPISINAROCILO). Po preverjanju pravilnosti vhodnih podatkov sistem naročil v informacijski sistem ADZU (sistem Alpha) posreduje podatke o naročilu, generira pogodbo XML in jo posreduje Abanetu. Ta poskrbi za vizualizacijo pogodbe in jo ponudi stranki v podpis. Ko stranka pogodbo s podpisno komponento podpiše, se zopet posreduje v sistem naročil (procedura EBNNAR_NAROCILO), nato se v informacijskem sistemu ADZU (sistem Alpha) preveri, ali gre za njim znano stranko. Če gre, se pogodba podpiše še s strani ADZU, v sistemu Alpha in sistemu naročil pa se naročilo zabeleži kot izvršeno. V primeru, da gre za neznano stranko, se naročilo zaustavi v statusu "v čakanju", stranki pa se posreduje sporočilo, da je za izvršitev naročila potrebno ADZU-ju posredovati fotokopijo osebnega dokumenta. Po prejemu in potrditvi pristnosti dokumenta ter ročni označitvi naročila v sistemu Alpha kot "potrjeno" se naročilo podpiše s strani ADZU in izvrši v obeh sistemih. V primeru, da stranka po določenem času dokumenta ni posredovala, se naročilo v sistemu Alpha označi kot "zavrnjeno" in zavrne tudi v sistemu naročil.

4.1.2 Tehnološke zahteve

Zaradi zapisa vseh potrebnih podatkov v prilagodljivi obliki XML sprememba podatkovnega modela v sistemu naročil ob dodajanju novih vrst naročila načeloma ni potrebna. Tako se je izkazalo tudi pri naročilu ADZU. Potrebno pa je bilo prilagoditi zaporedje izvajanja naročila. Za ta namen je bil uveden nov status izvajanja "v čakanju". Ker se naročilo ni več avtomatično izvršilo (oziroma zavrnilo) je bilo potrebno dodajanje programske logike za preverjanje statusa naročila v sistemu Alpha. Ker je v sistemu že izvedeno dnevno preverjanje podpisanosti naročil, se je tudi preverjanje statusa umestilo v ta sklop (procedura NAR_PRETECENANAROCILA).

Za potrebe preverjanj je bil narejen paket TRKEADZU_PCK, ki vsebuje potrebno logiko za vračanje potrebnih informacij, kot je na primer preverjanje stranke.

Za komunikacijo s sistemom Alpha je bil narejen paket NARADZUALPHA_PCK. Vsebuje logiko za vračanje informacij in funkcije za vpis, aktivacijo oziroma preklic novega naročila. Posebna funkcija se uporablja tudi za preverjanje statusa naročila. Ker je način komunikacije tehnologija spletnih vmesnikov, je način programiranja in težave z njim enak kot je opisan v oddelku 3.2.2.1 Podpisovanje.

V paket NAROCILA_PCK je bilo potrebno vključiti vso programsko logiko, potrebno za vračanje podatkov o preverjanjih, izvršitev in podpisovanje novega naročila. Klic komponente za podpisovanje je bilo potrebno prilagoditi za možnost izbire certifikata, ki se pri naročilu ADZU razlikuje od ostalih naročil, kjer se uporablja Abančni certifikat.

4.1.3 Prenova sistema Arhiviranje

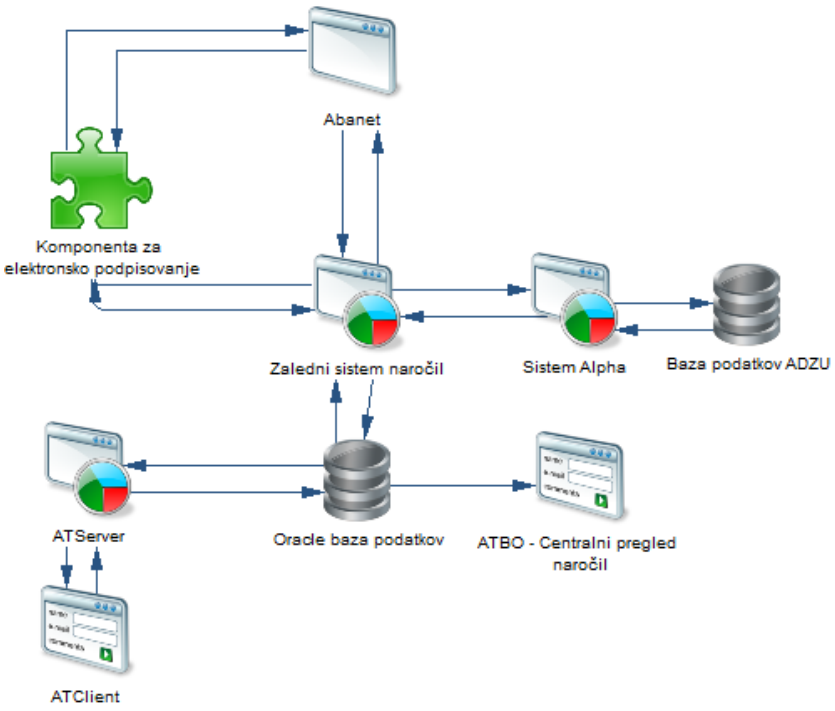
Kot je navedeno v oddelku 3.2.2.2 Arhiviranje, se je za arhiviranje uporabljala namenska samostojna aplikacija, izdelana s tehnologijo Delphi.NET. Vzdrževanje aplikacije se je zaradi nestabilnosti Delphi.NET tehnologije izkazalo kot precej težavno. Prav tako je nadzor delovanja zaradi nezdružljivosti tehnologije z obstoječimi nadzornimi sistemi banke otežen, včasih celo nemogoč.

Zato je bila sprejeta odločitev, da se aplikacija ukine, njena funkcionalnost pa prenese v obstoječ bančni sistem ATServer (poglavje 3.1.1).

Zaradi specifike naročila ADZU je bil sprejet sklep, da se dokument ne arhivira samo enkrat, v statusu "izvršeno", ampak tudi v statusu "v čakanju", zato je bila potrebna sprememba logike arhiviranja. Med zahteve za spremembo aplikacije je bila uvrščena možnost večkratnega arhiviranja dokumenta v poljubnih statusih.

4.1.4 Diagram bodoče postavitve

Celotno postavitev sistema ponazarja naslednja slika (Slika 4.1).



Slika 4.1: Diagram postavitve sistema

4.2 Varnost

Internet kot nova oblika virtualnega prostora, v katerem je oblika anonimnosti in svobode razmeroma velika, predstavlja nove izzive na področju varnosti. Zaradi svoje zasnove ga je nemogoče v celoti obvladovati, zato je na vsakem udeležencu samem odgovornost za zagotavljanje varnosti in zasebnosti podatkov, ki jih oddaja oziroma prejema. Žal z razvojem metod za varen prenos in zaščito podatkov poteka tudi vzporeden razvoj metod za prestrezanje in poneverbo podatkov, slednji marsikdaj prehiteva prvega. Spletno bančništvo z občutljivimi finančnimi transakcijami predstavlja še poseben izziv pri zaščiti.

Spletna banka Abanet uporablja različne metode za zagotavljanje varnosti poslovanja. Vendar pa zaradi varnostne politike banke ne smejo biti tema diplomske naloge, zato bom več pozornosti posvetil varnostnim ukrepom s strani uporabnika.

4.2.1 Varnost na strani uporabnika

Večina napadov na spletno bančništvo danes temelji na zavajanju uporabnika, ki mu želijo odtujiti podatke, potrebne za prijavo v spletno banko. Nekatere od metod, s katerimi poizkušajo, so:

- spletno ribarjenje (ang. phishing),
- zvabljanje (ang. pharming),
- napad XSS (ang. cross-site scripting),
- beleženje (ang. keystroke logging),
- trojanski konj (ang. trojan horse).

Tem napadom je skupno, da z različnimi metodami uporabnika prevarajo, da zlonamernemu programu posreduje svoje podatke, potrebne za prijavo v spletno banko. Ta nato te podatke zlorabi za vstop v banko in preusmerjanje denarja na račune lastnika programa.
Uporabnik se proti večini napadov lahko zavaruje z uporabo varnega operacijskega sistema in spletnega brskalnika ter skrbnega varovanja svojih podatkov, potrebnih za prijavo. Metode za zagotavljanje varnosti so osnovne in preproste: izbira varnega operacijskega sistema in spletnega brskalnika, redno posodabljanje programske opreme, pazljivost pri prejemanju in izvajanju nepoznane programske opreme, uporaba varnega protivirusnega programa in programa za požarni zid.

Kljub vsem naprednim varnostnim mehanizmom se v praksi izkaže, da je najšibkejši člen v varnostni verigi uporabnikova neizkušenost in naivnost.

4.2.2 Kvalificirano digitalno potrdilo, podpisna komponenta uporabnika

Pri spletni banki Abanet se, poleg gesla, za identifikacijo uporabnika uporablja kvalificirano digitalno potrdilo (včasih imenovano tudi certifikat), KDP, javnega overitelja, ki je za izdajo certifikata pooblaščen. KDP enolično predstavlja uporabnika in je nujno potreben za začetek vstopa v spletno banko.

Za oddajo naročil pa je poleg KDP-ja nujno potreben tudi poseben vtičnik za spletni brskalnik, podpisna komponenta. Skupaj s KDP-jem omogočata storitev elektronskega podpisovanja.

4.3 Pravni vidik

4.3.1 Pravna podlaga, zakoni

Na vprašanje varnosti se tesno navezuje identifikacija uporabnika in s tem povezana pravna veljavnost sklepanja pogodb prek elektronskih medijev. Za ta namen se v sodobnem poslovanju uporabljata v prejšnjem poglavju omenjena kvalificirano digitalno potrdilo in podpisna komponenta.

V slovenskem pravnem redu je bilo področje elektronskega podpisovanja prvič urejeno leta 2000 [8]. S temeljnim aktom, Zakonom o elektronskem poslovanju in elektronskem podpisu (ZEPEP-UPB1, Ur. l. RS, št. 98/2004) [9], in pripadajočo Uredbo o pogojih za elektronsko poslovanje in elektronsko podpisovanje (Ur. l. RS, št. 77/2000, 2/2001, 86/2006) priznava enako veljavnost in dokazno vrednost elektronskega in lastnoročnega podpisa. Skupaj z Zakonom o elektronskem poslovanju na trgu (ZEPT, Ur. l. RS, št. 61/2006) [10], ki se neposredno sicer ne nanaša na varnost elektronskega podpisovanja, a določa pogoje za sklepanje pogodb v elektronski obliki, predstavljata pravne okvire za delovanje sistema naročil.

Varen elektronski podpis izpolnjuje naslednje zahteve:

- povezan je izključno s podpisnikom,
- iz njega je mogoče zanesljivo ugotoviti podpisnika,
- ustvarjen je s sredstvi za varno elektronsko podpisovanje, ki so izključno pod podpisnikovim nadzorom,
- povezan je s podatki, na katere se nanaša, tako da je opazna vsaka kasnejša sprememba podatkov ali povezave z njimi.

Za potrebe elektronskega podpisovanja banka priznava KDP-je naslednjih slovenskih overiteljev:

- SIGEN-CA,
- Pošta-CA,
- Halcom-CA,
- AC NLB.

Ker je pogodba v obliki XML, je bil kot standard podpisovanja izbran standard XMLDsig, definiran pri organizaciji W3C (ang. World Wide Web Consortium), ki predpisuje načine elektronskega podpisovanja elektronskih dokumentov v obliki XML. Ta standard izpolnjuje zahteve, zapisane zgoraj: integriteto in pristnost podatkov ter podpisnika.

Kot je omenjeno že v oddelku 3.2.2.1 Podpisovanje se za podpisovanje s strani banke uporablja komponenta proXSign®. Za podpisovanje s strani stranke se uporablja komponenta, ki nudi enako funkcionalnost, le da je izvedena v obliki vtičnika za spletni brskalnik. Za brskalnik Firefox je to komponenta SETCCE proXSign® XML Firefox.

4.3.2 Sledljivost

Poleg zagotavljanja pravilnosti nastajanja pravno zavezujočih elektronskih dokumentov je pri elektronskem poslovanju potrebno poskrbeti tudi za sledljivost dokumentov oziroma njihovo hrambo. V Sloveniji način hrambe in zagotavljanja celovitosti in avtentičnosti dokumentarnega gradiva ureja Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih (ZVDAGA, Ur. l. RS, št. 30/2006) [11] skupaj s pripadajočo Uredbo o varstvu dokumentarnega in arhivskega gradiva (Ur. l. RS, št. 86/2006).

V sistemu naročil se dokumenti hranijo:

- lokalno v informacijskem sistemu banke kot posamezne verzije dokumenta, ki nastajajo v postopku naročila,
- pri zunanjem izvajalcu, kjer se hrani končna verzija dokumenta.

Postopek hranjenja pri zunanjem izvajalcu je že opisan v oddelku 3.2.2.2 Arhiviranje. Tako pri arhiviranju kot pri elektronskem podpisovanju poleg banke same tudi zunanji izvajalec jamči za zakonsko pravilnost izvedenega postopka.

Poglavje 5

Predstavitev uporabljenih tehnologij, orodij

5.1 Oracle

5.1.1 Sistem za upravljanje z bazami podatkov Oracle

Za zapisovanje in hranjenje podatkov se v Abanki uporablja sistem Oracle 10g [5]. Baza podatkov Oracle oziroma natančneje sistem za upravljanje relacijskih podatkovnih baz Oracle je eden izmed množice produktov, ki, kot pove že ime, ponuja upravljanje z relacijsko bazo podatkov. Veliko število občutljivih podatkov in transakcij med njimi, hitrost izvajanja operacij in potreba po čim večji zanesljivost in odzivnosti so bili nekateri od faktorjev, ki so v množici konkurenčnih produktov, kot so npr. zaprtokodni IBM DB2, Microsoft SQL Server in odprtokodni PostgresSQL, Firebird in MySQL, pripeljali do odločitve za Oracle. Za vnašanje in pridobivanje podatkov se uporablja jezik SQL za delo s podatkovnimi bazami.

Ena od prednosti Oracla je tudi podpora tehnologiji XML, kar zagotavlja od verzije 9 naprej. Sistemu naročil, kjer je pogodba s stranko predstavljena v formatu XML, omogoča upravljanje in manipuliranje z dokumentom.

Uporabljane so tudi shranjene procedure in funkcije (ang. stored procedures), ki so shranjene in izvajane na Oraclu samem. V Oraclu podprta programska jezika za implementacijo takih objektov sta v trenutno uporabljani verziji Java in PL/SQL, slednjega v Abanki tudi aktivno uporabljamo.

5.1.2 PL/SQL

PL/SQL je Oraclova proceduralna razširitev jezika SQL. Kljub dejstvu, da je SQL popolni jezik po Turingu[6], je za zahtevnejše programiranje neprimeren. PL/SQL to pomanjkljivost odpravlja. Uvaja spremenljivke, IF stavke, zanke in izjeme. Združenost z bazo olajša programiranje dostopa do podatkov in poveča hitrost izvajanja. Slabosti pa so omejenost pri vhodno-izhodnih operacijah in zanemarjanje principov večnivojske arhitekture, ki predpisujejo, da poslovna logika programa ni del baze.

Oraclove funkcije in procedure se lahko vsebinsko združujejo v pakete (ang. packages). Tak način programiranja izboljšuje preglednost in učinkovitost.

5.2 TOAD

Za delo s podatkovno bazo in programiranje v PL/SQL se uporablja orodje TOAD (Tool for Oracle Application Developers). Kljub temu da s podatkovno bazo lahko v celoti upravljamo iz ukazne vrstice, so za kompleksnejše projekte primernejše aplikacije, ki olajšajo delo. Za dotični projekt je bil izbran TOAD, ki omogoča lažje in preglednejše kreiranje tabel, prožilcev in podatkovnih struktur, prav tako močno izboljša učinkovitost pri programiranju v PL/SQL in omogoča razhroščevanje. Pri sistemu naročil, aplikaciji, pretežno razviti v PL/SQL, je uporaba takšnega okolja pravzaprav nujna.

5.3 Delphi

5.3.0.1 Delphi IDE

Delphi je integrirano razvojno okolje, zasnovano okoli programskega jezika Delphi Pascal, ki je različica objektnega Pascala. Njegove prve različice so orale ledino na razvoju med okolji za hitro razvijanje programov, še danes pa novejše verzije, kljub temu da po razširjenosti zaostajajo, stojijo ob boku sodobnim programskim okoljem. Največje prednosti Delphija dandanes so njegovo razvojno okolje, ki vključuje obsežno knjižnico vizualnih komponent VLC (ang. Visual Component Library). To omogoča hitro razvijanje programskih sistemov, še posebej prikaznih mask, prav tako je enostavno delo z bazami podatkov. Še ena prednost je velika internetna skupnost na Usenet forumih in spletnih forumih.

V projektu se uporablja predvsem za izdelavo uporabniškega vmesnika. Ker je Delphi standardno okolje za razvoj v Službi razvoja informatike in zaradi dejstva, da je vmesnik umeščen v že obstoječo aplikacijo Abanke ATBO, narejeno v Delphiju, je bil pravzaprav edina izbira.

5.3.0.2 Delphi Pascal

Delphi Pascal je različica programskega jezika Pascal, uporabljena v razvojnem okolju Delphi. Pascal je imperativen in proceduralen programski jezik z močnim tipiziranjem vseh objektov. Narejen je bil z namenom učenja strukturiranega programiranja. Na njegov razvoj so vplivali jeziki, kot so Algol 60, COBOL, Simula 67. Sprva čisti proceduralni jezik je bil kasneje nadgrajen s koncepti objektno usmerjenega programiranja in nastal je Objektni Pascal. Danes morda ne najbolj popularen, vendar zaradi svoje tradicije še vedno eden od najbolj uporabljanih jezikov.

5.4 XML

5.4.1 Standard XML

XML, razširljiv označevalni jezik, je spisek pravil za elektronsko zapisovanje dokumentov. Cilji ob njegovem nastajanju so bili preprostost, splošnost in uporabnost na medmrežju. Kljub temu da je zapis tekstovni in da je primarno usmerjen na dokumente, je široko uporabljen za zapisovanje različnih podatkovnih struktur. Zato je uporaben tako za računalniško obdelavo kot, zaradi svoje relativno preproste berljivosti, tudi za človeško oko. Poseben poudarek je tudi na njegovi splošnosti in univerzalnosti - zaradi podpore Unicode formatu je uporaben v vseh svetovnih jezikih.

```
<?xml version="1.0" encoding="WINDOWS-1250"
  standalone="no"?>
<PristopneIzjave>
 <Podsklad>
  <PODSKLADID>105</PODSKLADID>
  <PODSKLADIME>ABANCNA DZU PASIVNI AFRIKA</PODSKLADIME>
  <PODSKLADTRR>051008012993627</PODSKLADTRR>
  <PRISTOPNICAID>2</PRISTOPNICAID>
  <PristopneSheme>
   <Shema>
```

```
    <SHEMASKLIC>03 23000-26380129</SHEMASKLIC>
   </Shema>
  </PristopneSheme>
 </Podsklad>
</PristopneIzjave>
```

V projektu je bil izbran zaradi svoje primernosti za elektronsko poslovanje. Dokument na papirju je preprosto predstavljen kot dokument XML. Podatkovna baza Oracle zanesljivo podpira delo s to tehnologijo, primeren pa je za elektronsko podpisovanje in verificiranje.

5.4.2 Vizualizacija XML

Kljub svoji berljivosti za oko programerja pa dokument XML za navadnega uporabnika ni primeren. Za prikaz dokumenta XML na ekranu uporabnika je bila zato uporabljena preslikava XLST (ang. Extensible Stylesheet Language Transformation). XLST je tehnologija, uporabna za transformiranje dokumentov XML v druge dokumente XML oziroma drugačne formate, kot npr. HTML (ang. hyper text markup language). V sistemu naročil je izvorni dokument XML s procesorjem predlog XSLT na podlagi slogovnega modula XSLT preveden v izhodni dokument HTML, ki ga uporabnik vidi na ekranu kot pogodbo (Slika 5.1). Slogovni modul XLST je del izvornega dokumenta XML [12].

```
<?xml version="1.0" encoding="WINDOWS-1250"
standalone="no"?>
<?xml-stylesheet type="text/xml" href="#trans1"?>
<Document>
...
</Document>
<xsl:stylesheet
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  id="trans1" version="1.0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
 <xsl:output encoding="windows-1250" indent="no"
    method="html"
  version="1.0"></xsl:output>
 <xsl:decimal-format decimal-separator=","
    grouping-separator="."
  name="dec"></xsl:decimal-format>
 <xsl:template match="Data">
```

Št. 100618NA06001778

PRISTOPNA IZJAVA k pravilom upravljanja Krovnega sklada Abančna DZU za naslednje podsklade:

Naziv sklada	Shema	Številka pristopne izjave
Abančna DZU URAVNOTEŽENI	SHEMA B	2425

Slika 5.1: Prikaz pogodbe XML v brskalniku Firefox

5.5 WebServices

Tehnologija spletnih storitev (ang. Web Services) je vmesnik za programiranje aplikacij, ki so dosegljive prek različnih protokolov in izvršene na oddaljenem računalniku. W3C definira spletne vmesnike kot programski sistem, narejen za podporo interoperabilnih interakcij računalnika z računalnikom prek omrežja. Razmeroma nova tehnologija je olajšala kompleksnejšo komunikacijo z oddaljenimi sistemi in je v zadnjem času nekakšen "modni hit" prenosa informacij, kjer je potrebna interoperabilnost.
Sistem naročil pri komunikaciji s sistemom za arhiviranje tako uporablja spletne storitve, ki uporabljajo sporočila XML, narejena po protokolu SOAP (ang. Simple Object Acess Protocol) prek protokola HTTP (ang. hyper text transfer protocol). Spletni strežnik za arhiviranje nudi opis spletnih storitev v formatu WSDL (ang. Web Services Description Language). Ta služi za avtomatično generiranje programske kode odjemalca.

Kar se programerju odjemalca navzven kaže kot kreiranje objekta in klic funkcije

```
var
  wsArhiviraj: InDocArchiveWebServiceSoap;
begin
  wsArhiviraj := GetInDocArchiveWebServiceSoap;
  wsArhiviraj.InitSaveDocument(auiUserInfo, orgID,
     jobID, sSessionToken, AResult);
end;
```

je pravzaprav kreiranje dokumenta SOAP, pošiljanje po protokolu HTTP:

```
POST /abanka/webservice2/InDocArchiveWS.asmx HTTP/1.1
Host: hosting.arhiviraj.si
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <InitSaveDocument
       xmlns="http://www.mikrocop.com/InDoc/Archive">
      <UserInfo>
        <UserName>uporabnisko\_ime</UserName>
        <Password>geslo</Password>
        <IntegratedSecurity>true</IntegratedSecurity>
        <InstanceName>ime_instance</InstanceName>
      </UserInfo>
      <OrgID>0</OrgID>
      <JobID>0</JobID>
    </InitSaveDocument>
  </soap12:Body>
</soap12:Envelope>
```

in procesiranje odgovora:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <InitSaveDocumentResponse
       xmlns="http://www.mikrocop.com/InDoc/Archive">
      <InitSaveDocumentResult>string
        </InitSaveDocumentResult>
      <AResult>
        <ErrorCode>59</ErrorCode>
```

```
        <ErrorMessage>Napacno uporabnisko
           ime/geslo</ErrorMessage>
      </AResult>
    </InitSaveDocumentResponse>
  </soap12:Body>
</soap12:Envelope>
```

Prednosti tehnologije so pri hitrosti razvoja očitne. Vendar pa nova tehnologija XML v starejših razvojnih okoljih ni zadovoljivo implementirana, zato lahko pri programiranju prihaja do težav.

Poglavje 6

Izdelava aplikacije

Izdelava same aplikacije je potekala v več fazah. Prva faza predstavlja izdelavo preverjanj, ki jih za pridobivanje informacij potrebuje Abanet, in njihovo testiranje.

Druga faza je izdelava prilagoditve Abaneta. Ker je ta del prevzel zunanji izvajalec, s stališča banke ta del predstavlja izdelavo karseda natančne dokumentacije. Zaradi čim hitrejšega razvoja pa ta ni dokončna in dopušča nadaljnje spremembe in dopolnitve, če se pokaže potreba.

V tretji fazi se je izvedel notranji razvoj aplikacije. Podlaga za to je uporabniška in tehnološka specifikacija. V dotični aplikaciji je to pomenilo naprej razvoj Oraclovih funkcij za sodelovanje z Abanetom in obdelavo naročila, nato pa še razvoj nadzornega sistema z Delphijem. V ta del je bila vključena tudi prenova sistema arhiviranja in testiranje posameznih modulov in funkcij. Ob samem programiranju je v veliko pomoč literatura, ki se nanaša na dano tehnologijo [13, 14].

V četrti fazi se je izvedlo obširnejše testiranje celotnega sistema z različnimi načini uporabe in predaja aplikacije v uporabo.

6.1 Izdelava

6.1.1 Implementacija sprememb

6.1.1.1 Oracle, PL/SQL

Izdelava se je začela z izdelavo paketov TRKEADZU_PCK in NARADZUALPHA_PCK. Podlaga za programiranje sta dokumenta ABANET_NAR_ADZU-_31_Specifikacija funkcij pristopa k ADZU_v2 in ABANET_NAR_ADZU_31-_Specifikacija funkcij pristopa k ADZUALPHA_v2. V njiju je specifikacija vseh funkcij in procedur, potrebnih za pridobivanje informacij iz bančnih in sistemov ADZU ter napak, ki jih potencialno vračajo. Kot primer navajam del dokumentacije za funkcijo, ki vrne seznam pošt:

```
TRKEADZUSeznamPost (preverjanje 0606)

function TRKEADZUSeznamPost (

  aSeznam out sys_refcursor,
  aErrCode out varchar2,
  aErrMsg out varchar2
)
return number;

--brez vhodnega podatka
--vrne ref cursor aSeznam s seznamom post, (POSTA, KRAJ)
--vrne 0, ce ni napake, sicer  -1;
--v errCode in aErrMsg so kode in opisi napak, loceni z ;

0606 - Napaka pri pridobivanju seznama post.;
```

Sledila je dopolnitev paketa NAROCILA_PCK s preverjanji in klici novih funkcij. Izhodni dokument tega dela je krovni dokument za sistem naročil ABANET-NAR-KROV-Specifikacija zalednega sistema, ki vsebuje vse potrebne podatke za komunikacijo Abaneta s sistemom naročil. Skupaj z dokumentom 20080092_ABA-NAR-eADZU-Tehnološka specifikacija, ki vsebuje potrebne prilagoditve Abaneta, predstavljata vse potrebne informacije za zunanjega izvajalca. Ta del je bil naslednja faza projekta.

Ko je zunanji izvajalec zaključil s svojim delom, je sledila, kar zadeva programiranje zalednega sistema, zadnja faza projekta. V paketu NARO-

CILA_PCK se je sprogramiralo celotno obdelovanje novega naročila. Ta del, časovno in po količini kode obsežnejši, vključuje kontrole za vsebino in strukturno pravilnost vhodnih podatkov, pridobivanje in posredovanje podatkov za pogodbo in kontrolo podpisane pogodbe in njeno nadaljnjo obdelavo.

Do sprememb v primerjavi z drugimi naročili je prišlo v dodajanju nove faze in v fazi podpisovanja. Naročilu, ki je do sedaj prehajalo med statusi: 01 - Oddano, 02 - Pogodba za podpis naročnika, 03 - Naročnik podpisal, 10 - Izvršeno, 80 - Preklicano in 90 - Zavrnjeno, je bil dodan status 04 - V obdelavi. V dnevno preverjanje naročil se je vključil klic funkcije TRKEADZUSTATUS-NAROCILA iz paketa NARADZUALPHA_PCK, ki za naročila v statusu 04 v sistemu Alpha vrne informacijo o ročni potrditvi oziroma zavrnitvi. Del logike za izvršitev oziroma zavrnitev naročila ADZU je bil zato prestavljen v modul dnevnega preverjanja naročil.

Prav tako se je prilagodilo funkcijo za uporabo modula za elektronsko podpisovanje. Ker je bil do sedaj v uporabi le en certifikat, Abančni, je bilo potrebno prilagoditi klic modula z dodatnim parametrom, vrsto certifikata glede na vrsto naročila.

V tej fazi izdelave je bil sistem za oddajo naročil že funkcionalen.

6.1.1.2 Delphi

V okolju Delphi 2007 je potekalo dopolnjevanje centralnega pregleda naročil. Pred projektom je bila funkcionalnost pregleda na osnovni ravni, zato so se med izdelavo samo vključevale nove funkcionalnosti. Dodani sta bili maski za prikaz podrobnosti naročila in za prikaz zgodovine naročila, prenovljen je bil prikaz pogodbe, obstoječe maske pa so bile prilagojene na novo strukturo dokumenta XML naročila.

Prenova sistema arhiviranja je bila zaradi vključitve v ATServer, ki je razvit v tem okolju, izvedena v okolju Deplhi 7. Najprej je bila s pomočjo Delphijevega vgrajenega uvoznika (ang. WSDL importer) in datoteke WSDL ustvarjena datoteka InDocArchiveWS.pas z opisi funkcij in tipov objektov, ki se uporabljajo pri arhiviranju.

Nato je bila znotraj ATServerja na novo zdefinirana nit, namenjena arhiviranju, in sprogramirana logika, ki enkrat dnevno proži delovanje. Z na novo ustvarjenimi funkcijami iz datoteke InDocArchiveWS.pas se nato izvede arhivi-

ranje. Dodatno je bila implementirana tudi možnost večkratnega arhiviranja dokumenta.

Pri razvoju je prišlo do nepredvidenih težav. Eden izmed parametrov funkcije, ki smo jo uporabili za pošiljanje podatkov oddaljenemu strežniku, je tipa TByteDynArray. Delphi 7 ima pri delu s tem tipom veliko dokumentiranih težav. Vrednost spremenljivke, ki predstavlja dejanski parameter, smo zato določili z naslednjo funkcijo:

```
function TxNarArhThread.FileToByteArray( const FileName
   : string ) : TByteDynArray;
const BLOCK_SIZE=1024;
var BytesRead, BytesToWrite, Count : integer;
    F : File of Byte;
    pTemp : Pointer;
begin
  AssignFile( F, FileName );
  Reset(F);
  try
    Count := FileSize( F );
    SetLength(Result, Count );
    pTemp := @Result[0];
    BytesRead := BLOCK_SIZE;
    while (BytesRead = BLOCK_SIZE ) do
    begin
      BytesToWrite := Min(Count, BLOCK_SIZE);
      BlockRead(F, pTemp^, BytesToWrite , BytesRead );
      pTemp := Pointer(LongInt(pTemp) + BLOCK_SIZE);
      Count := Count-BytesRead;
    end;
  finally
    CloseFile( F );
  end;
end;
```

6.2 Testiranje

Testiranje aplikacije lahko razdelimo v dva dela. Prvi del predstavlja testiranje posameznih delov kode še preden je aplikacija končana oziroma še preden ponuja polno funkcionalnost. Drugi del pa je celovito testiranje vseh delov že končane aplikacije.

6.2.1 Razvojno testiranje posameznih modulov, funkcij

Razvojno testiranje posameznih delov kode praviloma opravljamo razvijalci sami in navadno poteka v razvojnem okolju. Po pripravi načrta testiranja se vsaka posamična funkcija oziroma modul kliče z različnimi parametri in preverja odgovore. Tako se npr. preverjanja testirajo z naslednjim klicem:

```
tStart:= dbms_utility.get_time;
RetVal := NAROCILA_pck.NARF_PREVERINAROCILO ( AIDKOM,
   ANAROCILOVRSTA, APREVERJANJEVRSTA, APARAMETRI,
   AEPOT, AATRIBUT1, AATRIBUT2 );
dbms_output.put_line(round((dbms_utility.get_time-
  tStart:= )/100, 2) ||'␣Seconds...' );
DBMS_OUTPUT.Put_Line('RETVAL␣=␣' ||
   RETVAL.GETCLOBVAL());
COMMIT;
```

ki hkrati poda tudi informacijo o časovni zahtevnosti preverjanja.

Pri testiranju elektronskega podpisovanja novega dokumenta je bilo ugotovljeno, da v nekaterih primerih pride do napake pri podpisu. Ugotovljeno je bilo, da do napake pride v vrstici kode, ki skrbi za prejemanje podpisanega dokumenta:

```
sResponse VARCHAR2(32767);
UTL_HTTP.READ_TEXT (r => httpResponse, DATA =>
   sResponse);
```

Težava je bila v dejstvu, da je dolžina dokumenta XML novega naročila skupaj s podpisom včasih presegla 32767 bajtov. V ta namen je bil implementiran naslednji obhod:

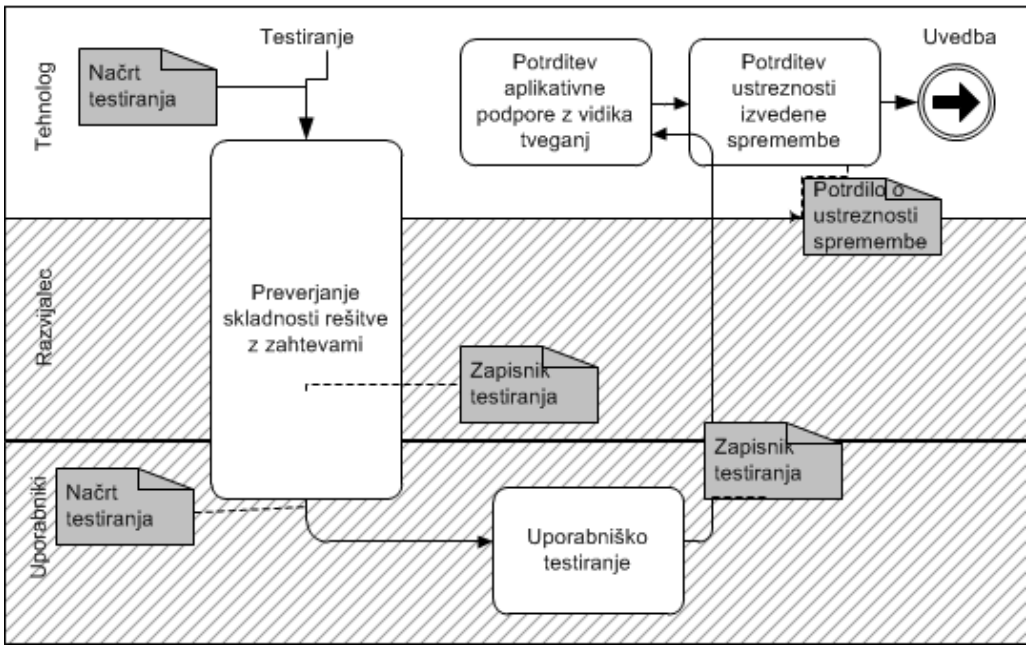
```
sResponse   CLOB:='␣';
buff        VARCHAR2 (32767);
begin
  loop
    UTL_HTTP.READ_TEXT (r => httpResponse, DATA => buff);
    sResponse := sResponse || buff;
  end loop;
exception
  when UTL_HTTP.END_OF_BODY then
    NULL;
end;
```

Podobna logika je bila preventivno implementirana tudi pri pošiljanju podatkov. Dokumenti XML prihodnjih naročil bodo morda še večji, zato smo se na ta način izognili enemu tipu potencialnih napak v podpisovanju dokumentov.

V tej fazi testiranja se navadno odkrije največ hroščev, ki nastanejo pri programiranju. Vendar pa, ker je področje testiranja omejeno, je omejena tudi zmožnost zaznavanja. Kompleksnejše, vsebinske napake, ki nastanejo kot posledica sodelovanja več modulov, v tej fazi v večini primerov ostanejo nezaznane, kar skuša odpraviti naslednja faza.

6.2.2 Testiranje pred predajo v uporabo

Končnega testiranja, za razliko od razvojnega, nikoli ne opravljamo razvijalci sami. Praviloma ga opravljajo tehnologi - testerji v sodelovanju s končnimi uporabniki, če je to možno. Vedno poteka v testnem okolju. Vhodna dokumenta za to fazo sta načrt testiranja in dokument s testnimi scenariji, izhodna pa zapisnik testiranja in potrdilo o ustreznosti spremembe (Slika 6.1).

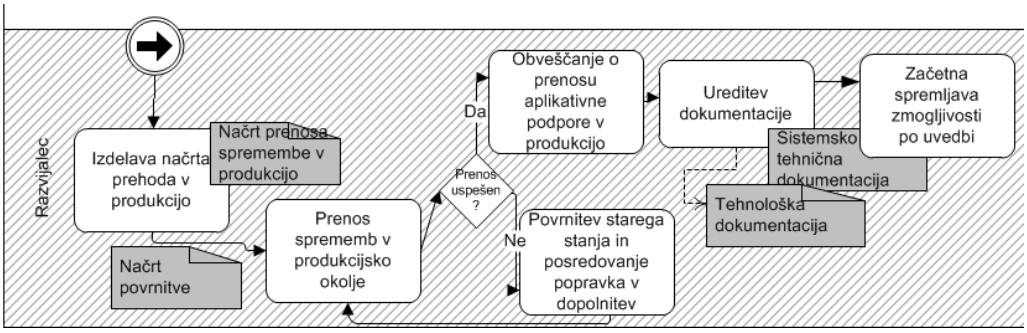


Slika 6.1: Testiranje

6.3 Predaja v uporabo

6.3.1 Objava na produkcijskih strežnikih

Po uspešnem zaključku testiranja in izdaji potrdila o ustreznosti spremembe sledi predaja v uporabo. Začne se z izdelavo načrta prehoda v produkcijo in načrta povrnitve. Po samem prenosu sprememb v produkcijsko okolje se preveri uspešnost prenosa. Če je prišlo do kakršnihkoli napak in prenos ni bil uspešen, se na podlagi načrta povrnitve sistem povrne v stanje pred prenosom. Po ugotovitvi in odpravitvi vzrokov za neuspeh se proces ponovi. Po uspešnem prenosu se o tem obvesti uporabnike in odda vsa potrebna dokumentacija. Prav tako se še nekaj časa spremlja vpliv sprememb na ostale dele informacijskega sistema in na zmogljivost. (Slika 6.2).



Slika 6.2: Prenos v uporabo

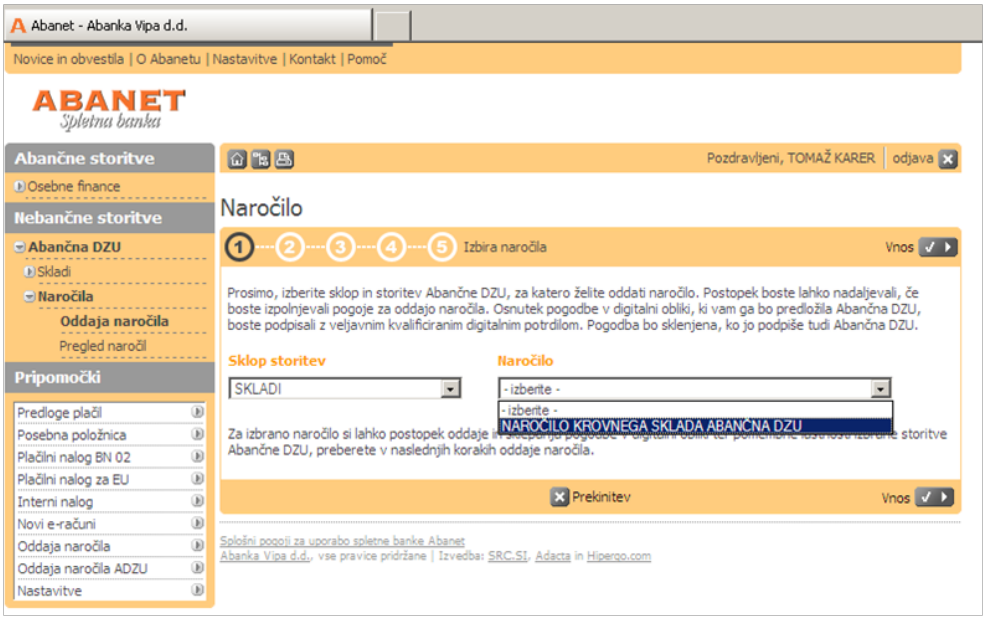
Poglavje 7

Delovanje aplikacije

Delovanje aplikacije lahko vidimo na dva načina. Eden je skozi oči uporabnika, stranke, ki se odloči za pristop k ADZU prek svojega spletnega brskalnika. Drugi način pa je delovanje skozi oči nadzornika, ki spremlja delovanje sistema skozi različne nadzorne module. V nadaljevanju bom predstavil oba vidika, s pomočjo opisa in zaslonskih slik. Vsi osebni podatki so izmišljeni.

7.1 Delovanje skozi oči uporabnika

Po prijavi v Abanet in izbiri Oddaja naročila se stranki ponudi izbira naročila (Slika 7.1):



Slika 7.1: 1. korak naročila

Po izbiri naročila NAROČILO KROVNEGA SKLADA ABANČNA DZU se prikaže drugi korak: strinjanje z izjavo o seznanjenosti s storitvijo in vnos podatkov, potrebnih za pristop (Slika 7.2):

Naročilo Krovnega sklada Abančna DZU

1 2 3 4 5 Vnos naročila – Seznanjenost s storitvijo in podatki vlagatelja Naprej

Na začetku postopka najprej označite polje pred Izjavo o seznanjenosti s storitvijo. Ob tem izberite naslov za prejemanje obvestil (stalni naslov ali drugi naslov) in dejavnost. Pri izbiri podatkov o vplačevanju določite osebni račun, s katerega boste izvedli prvo vplačilo v kasneje izbran/e podsklad/e.

Pri naslovu za prejemanje obvestil lahko drugi naslov spremenite. Opozorilo: V kolikor ga boste spreminjali boste po potrditvi naslova morali ponovno vnesti podatke, katere ste predhodno že izpolnili za postopek pristopa k podskladom Abančne DZU.

Izjava o seznanjenosti s storitvijo

☑ Izjavljam, da sem seznanjen/a z vsemi značilnostmi Krovnega sklada Abančna DZU, Pražakova ulica 8, 1000 Ljubljana, posebej pa z naslednjim: Krovni sklad Abančna DZU ni depozit pri banki in ni vključen v sistem zajamčenih vlog. Izdajatelj krovnega sklada je Abančna DZU, d.o.o., Pražakova ulica 8, 1000 Ljubljana. Abanka pri prodaji investicijskih kuponov nastopa kot prodajni posrednik. Vlagatelj prevzema tveganje, da bo ob unovčitvi investicijskega kupona dobil manj kot je investiral. Abanka za donose in izplačila iz podskladov Abančne DZU ne jamči.

Podatki vlagatelja

ime in priimek:	TOMAŽ KARER
naslov:	POD TOPOLI 55
kraj:	1000 LJUBLJANA
davčna številka:	00000001
davčni urad:	LJUBLJANA - VIČ - RUDNIK

Opozorilo: V primeru, da vaši podatki niso pravilni, vas prosimo, da se oglasite v najbližji poslovalnici Abanke z veljavnim osebnim dokumentom.

elektronski naslov:	TOMAZKARER@GMAIL.COM
telefon:	
telefaks:	

Naslov za prejemanje obvestil

izberite:	stalni naslov
ulica in hišna številka:	POD TOPOLI 55
pošta in kraj:	1000 LJUBLJANA
dejavnost:	Študent

Podatki o vplačevanju

izberite osebni račun (prvo vplačilo):	račun pri nefinančni inštituciji
vnesite številko osebnega računa:	SI56 9100 0000 0000 000
naziv banke / institucije:	Nefinančna inštitucija
razlogi za vlaganje: (varčevanje ali drugo)	

Pred oddajo naročila obvezno preberite Pojasnila za sklepanje pristopne izjave k pravilom upravljanja Krovnega sklada Abančna DZU prek spletne banke Abanet (pdf). Pojasnila so dostopna tudi s klikom na ikono z vprašajem v orodni vrstici.

Izbira Prekinitev Naprej

Slika 7.2: 2. korak naročila

Del drugega koraka je izbira o izpolnjevanju vprašalnika o profilu vlagatelja. Po izpolnjevanju vprašalnika je stranka na tretjem koraku, pregledu naročila (Slika 7.3):

Naročilo Krovnega sklada Abančna DZU

1 2 3 4 5 Pregled naročila Potrditev

Pred oddajo naročila preverite pravilnost vnesenih podatkov. Če želite spremeniti podatke, izberite povezavo "Vnos" levo spodaj. Za oddajo naročila in pripravo osnutka pogodbe v digitalni obliki, izberite povezavo "Potrditev" desno spodaj.

Podatki vlagatelja

ime in priimek:	TOMAŽ KARER
naslov:	POD TOPOLI 55
kraj:	1000 LJUBLJANA
davčna številka:	00000001
davčni urad:	LJUBLJANA - VIČ - RUDNIK
dejavnost:	ŠTUDENT
elektronski naslov:	TOMAZKARER@GMAIL.COM
telefon:	
telefaks:	

Naslov za prejemanje obvestil

naslov:	POD TOPOLI 55
kraj:	1000 LJUBLJANA

Podrobnosti o vplačevanju

osebni račun (prvo vplačilo):	SI56 9100 0000 0000 000
naziv banke / institucije:	NEFINANČNA INŠTITUCIJA
način vplačila:	POLOŽNICA/PLAČILNI NALOG
razlogi za vlaganje:	VARČEVANJE

Izbrani podskladi in sheme

Abančna DZU URAVNOTEŽENI

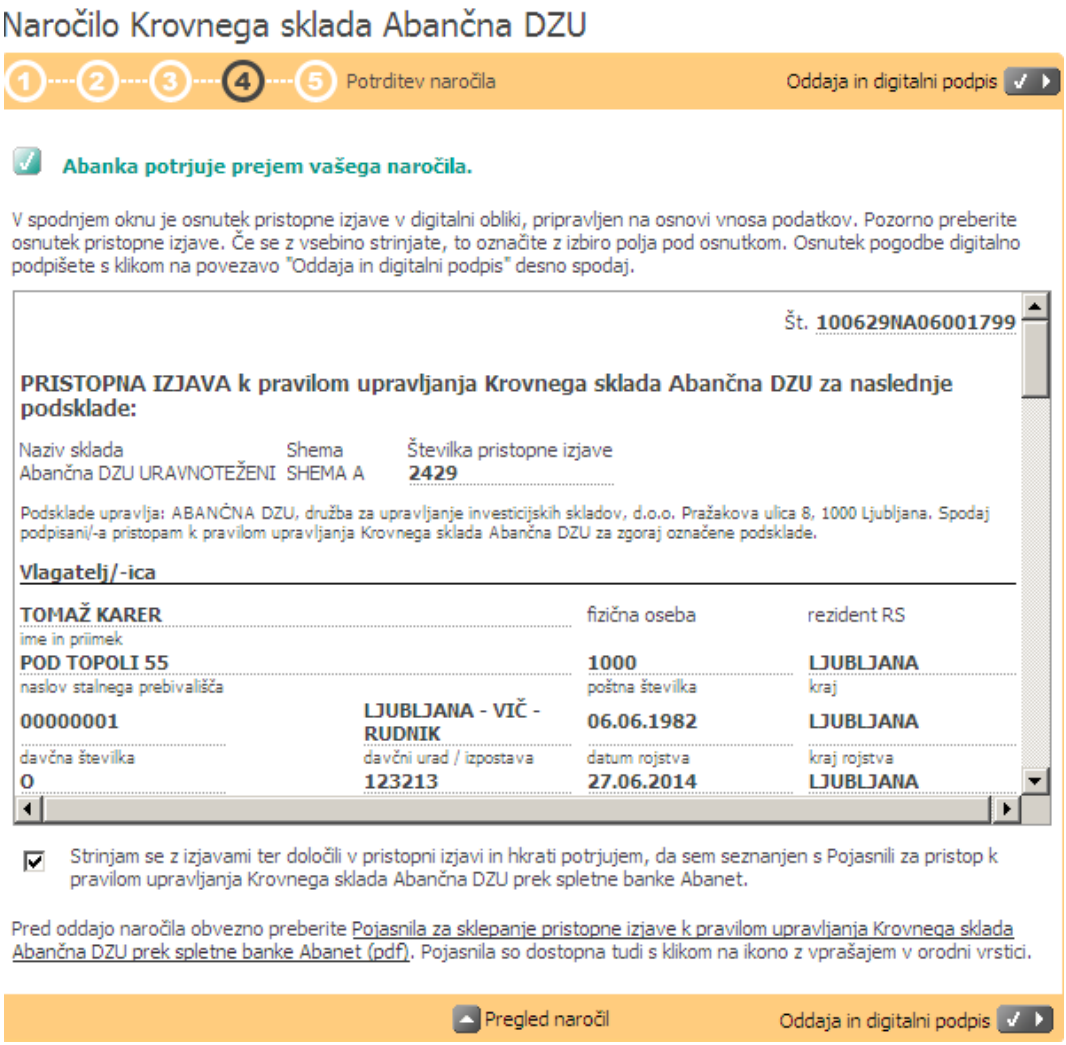
- SHEMA A

Pred oddajo naročila obvezno preberite Pojasnila za sklepanje pristopne izjave k pravilom upravljanja Krovnega sklada Abančna DZU prek spletne banke Abanet (pdf). Pojasnila so dostopna tudi s klikom na ikono z vprašajem v orodni vrstici.

Vnos Prekinitev Potrditev

Slika 7.3: 3. korak naročila

Po potrditvi je naročilo v četrtem koraku. V brskalniku se pojavi sporočilo o prejemu naročila in vizualizirana pogodba z vsemi podatki za podpis. Preklic naročila v tej fazi ni več možen (Slika 7.4):



Slika 7.4: 4. korak naročila

Na petem, zadnjem koraku so prikazani podatki o statusu naročila z opozorilom o nujnosti pošiljanja osebnega dokumenta (Slika 7.5):

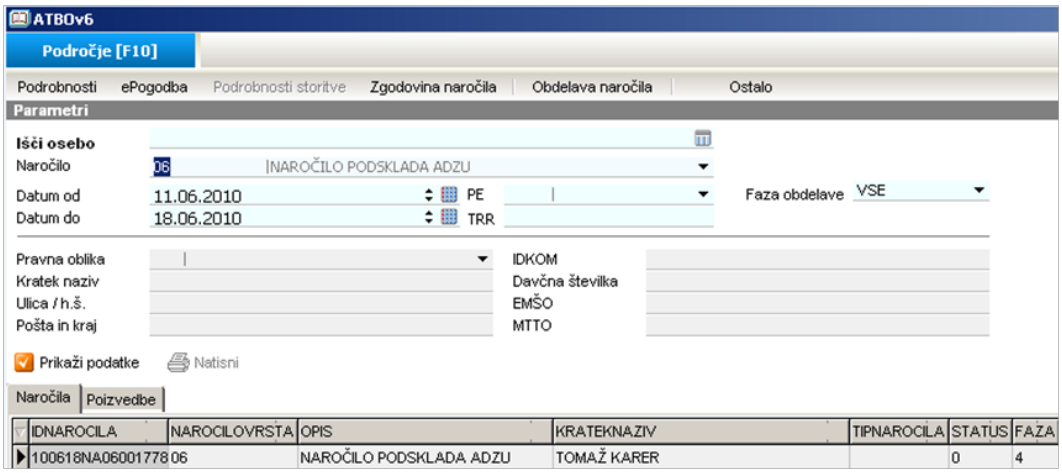
Slika 7.5: 5. korak naročila

7.2 Nadzorovanje delovanja

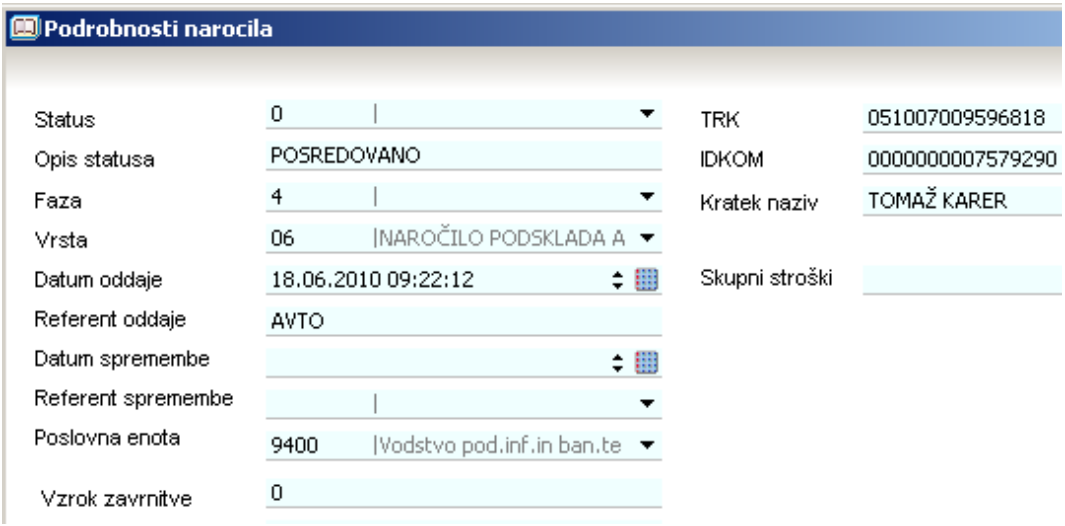
7.2.1 Nadzorovanje delovanja skozi nadzorne module banke

7.2.1.1 Centralni pregled naročil

Centralni pregled naročil na enem mestu nudi celovit pregled nad oddanimi naročili. Nadzornikom je omogočen prikaz oddanih naročil (Slika 7.6), omejen po različnih parametrih. Nudi prikaz podrobnosti naročila (Slika 7.7), pogodbo (Slika 7.8) in zgodovino naročila (Slika 7.9) ter prikaz poizvedb (Slika 7.10).



Slika 7.6: Prikaz naročil



Slika 7.7: Podrobnosti naročila

ePogodba

Shrani XML Natisni

Št. 100629NA06001799

PRISTOPNA IZJAVA k pravilom upravljanja Krovnega sklada Abančna DZU za naslednje podsklade:

Naziv sklada	Shema	Številka pristopne izjave
Abančna DZU URAVNOTEŽENI	SHEMA A	2429

Podsklade upravlja: ABANČNA DZU, družba za upravljanje investicijskih skladov, d.o.o. Pražakova ulica 8, 1000 Ljubljana. Spodaj podpisani/-a pristopam k pravilom upravljanja Krovnega sklada Abančna DZU za zgoraj označene podsklade.

Vlagatelj/-ica

TOMAŽ KARER ime in priimek		fizična oseba	rezident RS
POD TOPOLI 55 naslov stalnega prebivališča		1000 poštna številka	LJUBLJANA kraj
00000001 davčna številka	LJUBLJANA - VIČ - RUDNIK davčni urad / izpostava	06.06.1982 datum rojstva	LJUBLJANA kraj rojstva
0	123213	27.06.2014	LJUBLJANA

Slika 7.8: Pogodba naročila

Zgodovina naročila

IDNAR	VRSTA	DOGODEK	OPIS	DATUMOD	DATUMDO	STATUS	IDNAPAKE
100618NA06001778	06	01	Vpisano naročilo	18.06.2010 09:22:12		0	0
100618NA06001778	06	02	Uporabnik podpisal	18.06.2010 09:22:59		0	0
100618NA06001778	06	03	Posredovano na ADZU v obdelavo	18.06.2010 09:22:59		0	0

Slika 7.9: Zgodovina naročila

Prikaži podatke Natisni

Naročila Poizvedbe

IDKOM	KRATEKNAZIV	VRSTA	OPIS	DATUM	IDNAPAKE	NAPAKA	TRR
000000000579290	TOMAŽ KARER	06	NAROČILO PODSKLADA ADZU	18.06.2010 09:16:47	0603	Uporabnik že ima odprto naročilo za ADZU	0603

Slika 7.10: Poizvedbe

Poglavje 8

Zaključne misli in ugotovitve

Povratne informacije o delovanju informacijskega sistema so večinoma pozitivne. Stranke večjih težav pri pristopanju k skladom prek Abaneta nimajo, prva je pristopila že dan po prenosu sistema v produkcijo. Kljub temu da s potrebo po ročnem potrjevanju še neznanih strank pri pristopu k ADZU naročilo odstopa od načela popolne avtomatizacije procesa, se to dejstvo ne kaže kot preveč moteče. Stranka se še vedno izogne obiskom poslovalnic in čakanju v vrsti, za domačim računalnikom ima več časa in informacij za odločitve, ADZU pa ima popoln nadzor nad strankami, ki želijo pristopiti, in odobritvijo pristopa.

Kljub razmeroma velikem začetnem vložku časa in sredstev za zagon projekta, se pozitivni učinki že kažejo. Vedno več strank se odloča za bančne storitve prek elektronskih poti, prav tako delovanje banke ocenjujejo kot bolj udobno in zanesljivo. S stališča banke pa avtomatizacija sistemov prinaša velike prihranke pri količini dela, potrebnega za izvršitev storitve.

S stališča nadzornikov sistema ni bilo pripomb. Delovanje je pregledno in transparentno, napake se hitro zaznajo in odpravijo. Sistem velja za enega bolj stabilnih in zanesljivih znotraj podjetja.

Zame, kot programerja, je bil izvedeni projekt prvi večji, pri katerem sem sodeloval od same zasnove prek izdelave do zaključka. K njemu sem pristopil kmalu po začetku dela v banki, kar je pomenilo, da sem se hkrati seznanjal s projektom samim kot tudi z novimi tehnologijami, še posebej s tistimi, ki jih v času šolanja nisem spoznal. Novo delovno okolje je prineslo tudi nove načine komunikacije, dokumentiranja in različnih poslovnih praks. Delo je večinoma

potekalo gladko, nekatere težave pa smo sproti uspešno rešili znotraj delovnega kolektiva.

8.1 Nadaljnji razvoj

Zaledni sistem je bil ob svoji izdelavi zasnovan za karseda široko področje delovanja. Že pri načrtovanju in implementaciji prvega naročila, naročila izrednega limita, smo imeli v mislih njegovo širitev z novimi tipi naročil. Zasnovan okoli tehnologije XML omogoča dodajanje novih funkcionalnosti brez drastičnih posegov v podatkovni model. Delo na sistemu se nadaljuje, tako se je praktično takoj po izvedbi projekta pristopa k ADZU začel nov projekt, z dodajanjem novih funkcionalnosti, elektronskega pristopa k storitvi Abasms in naročila potrdila o izvršenem nalogu.

Slike

2.1 Abančne odvisne družbe 6

3.1 Diagram postavitve sistema 11
3.2 Podatkovni model 13

4.1 Diagram postavitve sistema 22

5.1 Prikaz pogodbe XML v brskalniku Firefox 30

6.1 Testiranje 39
6.2 Prenos v uporabo 40

7.1 1. korak naročila 42
7.2 2. korak naročila 43
7.3 3. korak naročila 44
7.4 4. korak naročila 45
7.5 5. korak naročila 46
7.6 Prikaz naročil 47
7.7 Podrobnosti naročila 47
7.8 Pogodba naročila 48
7.9 Zgodovina naročila 48
7.10 Poizvedbe 48

Literatura

[1] (2010) Spletna stran Abanke, Predstavitev. Dostopno na: http://www.abanka.si/sys/cmspage.aspx?MapaId=24998

[2] (2010) Spletna stran Združenja bank Slovenije. Dostopno na: http://www.zbs-giz.si/

[3] (2010) Spletna stran projekta Raba interneta v Sloveniji. Dostopno na: http://www.ris.org/

[4] (2010) Spletna stran Prophis eResearch. Dostopno na: http://www.prophis.com/pr/iBanking4_mar29.doc

[5] Interna gradiva Abanke d.d., 2010

[6] (2010) Turing completeness, Computability theory. Dostopno na: http://en.wikipedia.org/wiki/Turing_completeness#Computability_theory

[7] (2010) Spletna stran Abančne DZU. Dostopno na: http://www.abancna-dzu.si

[8] Pravilnik za uporabo epodpisa v epoteh, Abanka d.d., 2008

[9] (2010) Uradni list RS, št. 98/2004 z dne 9. 9. 2004. Dostopno na: http://www.uradni-list.si/1/objava.jsp?urlid=200498&stevilka=4284

[10] (2010) Uradni list RS, št. 61/2006 z dne 13. 6. 2006. Dostopno na: http://www.uradni-list.si/1/content?id=73791

[11] (2010) Uradni list RS, št. 30/2006 z dne 23. 3. 2006. Dostopno na: http://www.uradni-list.si/1/content?id=72425

[12] D. Tidwell, *XSLT: Mastering XML Transformations*, O'Reilly Media, 2001

[13] B. Chang, M. Scardina, S. Kiritzov, *Oracle9i XML Handbook*, McGraw-Hill, 2001

[14] R. Lischner, *Delphi in a Nutshell*, O'Reilly Media, 2000