

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Martin Kavčič

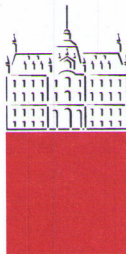
**Aplikacija za izdelavo in uporabo  
spletnih anket s pomočjo tehnologije  
WCF**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Franc Solina

Ljubljana 2012



Št. naloge: 00243/2012

Datum: 02.04.2012

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

Kandidat: **MARTIN KAVČIČ**

Naslov: **APLIKACIJA ZA IZDELAVO IN UPORABO SPLETNIH ANKET S  
POMOČJO TEHNOLOGIJE WCF  
APPLICATION FOR CREATING AND CONSUMING WEB SURVEYS  
USING WCF TECHNOLOGY**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Preučite problematiko anketiranja preko interneta oziroma svetovnega spleta.  
Izdelajte aplikacijo, ki bo omogočila njihovo čim lažjo izdelavo ter uporabo na internetu.

Mentor:

prof. dr. Franc Solina

Dekan:

prof. dr. Nikolaj Zimic



Rezultat 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.<sup>1</sup>

*Besedilo je oblikovano z urejevalnikom besedil  $\LaTeX$ .*

---

<sup>1</sup>V dogovorju z mentorjem lahko kandidat diplomsko delo s pripadajočo izvorno kodo izda tudi pod katero izmed alternativnih licenc, ki ponuja določen del pravic vsem: npr. Creative Commons, GNU GPL. V tem primeru na to mesto vstavite opis licence, na primer tekst [?]

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! Glej tudi sam konec Poglavlja ?? na strani ??.

## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Martin Kavčič, z vpisno številko **63020227**, sem avtor diplomskega dela z naslovom:

*Aplikacija za izdelavo in uporabo spletnih anket s pomočjo tehnologije WCF*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Franca Soline,
- 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 14. junija 2012

Podpis avtorja:

*Zahvaljujem se mentorju za pomoč pri izdelavi naloge, Anji za potrpežljivost med mojim delom, predvsem pa staršema, ki sta mi omogočila študij in verjela vame.*

# Kazalo

Povzetek

Abstract

Seznam Kratic

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Opis tehnologij</b>	<b>3</b>
2.1	Ogrodje .NET . . . . .	3
2.2	Microsoft Visual Studio 2008 . . . . .	5
2.3	Servisi . . . . .	6
2.4	Asp.NET . . . . .	6
2.5	Microsoft SQL Server 2008 Express . . . . .	7
2.6	JavaScript . . . . .	7
2.7	jQuery . . . . .	8
2.8	AnjLab SQLProfiler . . . . .	8
2.9	WCF (Windows Communication Foundation) . . . . .	9
<b>3</b>	<b>Ankete in njihova uporaba</b>	<b>11</b>
3.1	Pravila anketiranja . . . . .	11
3.2	Tipi anket . . . . .	12
3.3	Spletne ankete . . . . .	13

<b>4</b>	<b>Razvoj spletne aplikacije Ankete</b>	<b>15</b>
4.1	Razvoj informacijske rešitve po metodologiji RUP . . . . .	15
4.2	Analiza zahtev in načrtovanje . . . . .	18
4.2.1	Funkcionalne zahteve . . . . .	18
4.2.2	Nefunkcionalne zahteve . . . . .	20
4.2.3	Podatkovni model sistema . . . . .	21
4.3	Razvoj spletne aplikacije . . . . .	21
4.3.1	Idejna rešitev . . . . .	23
4.3.2	Prijava v sistem . . . . .	25
4.3.3	Registracija . . . . .	25
4.3.4	Pozabljeno geslo . . . . .	26
4.3.5	Izdelava ankete . . . . .	26
4.3.6	Dodajanje vprašanj in možnih odgovorov . . . . .	26
4.3.7	Reševanje anket in prikaz rezultatov . . . . .	28
<b>5</b>	<b>Razvoj servisa WCF in odjemalca WCF</b>	<b>31</b>
5.1	Umestitev servisa v aplikacijo Ankete . . . . .	31
5.2	Programski del servisa WCF . . . . .	31
5.2.1	Podatkovne pogodbe . . . . .	32
5.2.2	Servisne pogodbe . . . . .	32
5.2.3	Implementacija servisa in poslovne logike . . . . .	33
5.3	Gostovanje servisa WCF . . . . .	33
5.4	Končna točka . . . . .	35
5.5	Izmenjava metapodatkov (ang. Metadata Exchange) . . . . .	38
5.6	Upravljanje instanc pri tehnologiji WCF . . . . .	38
5.7	Varnost . . . . .	40
5.8	Razvoj odjemalca . . . . .	40
<b>6</b>	<b>Zaključek</b>	<b>47</b>
	<b>Literatura</b>	<b>49</b>

# Povzetek

Cilj diplomskega dela je izdelati spletno aplikacijo, ki uporabniku omogoča izdelavo ankete po meri. Želimo doseči to, da večina procesov in poslovne logike teče na strežniku, kar je bistvo porazdeljenih sistemov. Spletna aplikacija, poimenovana *Ankete*, bo delovala kot servis, ki registriranim uporabnikom nudi koriščenje shranjenih anket. Te uporabnike, bodo predstavljali spletni razvijalci oziroma ljudje s poznavanjem osnov okolja .NET, ki bi s pomočjo tehnologije WCF narejeno anketo povezali s svojo spletno stranjo. Vključil bi jo v svoj uporabniški vmesnik in ji po potrebi uredil slog ter videz. S tem bi se uporabniki izognili prehajanju med stranmi. Odjemalec bi na tak način svojo spletno stran polepšal. Postala bi bolj interaktivna in uporabna. Z uporabo tehnologije WCF bi zagotovili konsistentno komuniciranje ter prenašanje podatkov med odjemalcem in strežnikom.

KLJUČNE BESEDE: porazdeljeni sistemi, tehnologija WCF, spletne aplikacije, okolje .NET

# Application for creating and consuming web surveys using WCF technology

The main aim of diploma thesis is to create a web application, which enable registered users to create customized web surveys. We want to achieve, that most of the processes and business logic is running on the web server, which is the main point of distributed systems. Web application, named *Ankete*, will work as a service, which offer to registered users to consume already saved surveys. These users will represent web developers or people with basic knowledge of .NET Framework. They will link the survey with their own web site, with the help of Windows Communication Foundation technology. They will include the survey in user interface, change its appearance, dimensions and style. So users will avoid migration between pages. Developer will also improve his web page appearance, utility and interactivity. With use of WCF technology we provide consistent communication and message transferring between client and server.

KEYWORDS: distributed systems, WCF technology, web applications, .NET Framework

# Seznam kratic

**Asp** - Active Server Pages - skriptni jezik

**Asp.NET** - Spletno aplikacijsko okolje na ogrodju .NET

**CLR** - Common Language Runtime - izvajalno okolje programske kode v ogrodju .NET

**C#** - Microsoftov programski jezik

**CSS** - Cascading Style Sheets (predloge za izgled spletnih strani)

**DNS** - Domain Name Server (domenski strežnik)

**HTTP** - HyperText Transfer Protocol (protokol za prenos podatkov)

**HTML** - HyperText Markup language (znakovni jezik za izdelavo spletnih strani)

**IIS** - Internet Information Service (informacijski strežnik)

**IP** - Internet Protocol

**JAVA SCRIPT** - (skriptni jezik za izdelavo interaktivnih vsebin na strani)

**JQUERY** - (JavaScript knjižnica, ki poenostavi pisanje skript)

**RUP** - Rational Unified Process

**SQL** - Structured Query Language (jezik za upravljanje s podatki v relacijskih podatkovnih bazah)

**SOAP** - Simple Object Access Protocol (protokol, ki ga uporabljajo aplikacije za komunikacijo preko HTML-ja)

**URL** - Uniform resource locator (globalni naslov dokumentov)

**UML** - Unified Modeling Language - univerzalni modelni jezik

**XML** - Extensible Markup Language (jezik za zapis dokumentov v formatu, ki je razumljiv računalniku in ljudem)

## *KAZALO*

**WCF** - Windows Communication Foundation (programsko okolje za izdelavo servisno orientiranih aplikacij)

**WS** - Web Service

**WSDL** - Web Service Definition Language (jezik za opis spletnih servisov)

# Poglavje 1

## Uvod

V razvoju računalništva smo priča stalnemu napredku. Operacijski, informacijski in drugi računalniški sistemi so vedno naprednejši in tehnično sposobnejši. Vzporedno se povečuje tudi kapaciteta prenosa podatkov v omrežjih, preko katerih so ti sistemi povezani. Posledica tega je razvoj porazdeljenih sistemov. Dandanes celoten splet deluje po principu odjemalec-strežnik.

Po pogovoru z mentorjem smo se odločili, da razvijemo spetno aplikacijo, ki bi omogočala izdelavo spletnih anket. Anketa je orodje, ki nam pomaga priti do določenih znanj in spoznanj. Večina jih je namenjenih za podporo izboljšanju kvalitete produktov, izboljšanju trženja, za psihološke ali sociološke raziskave, za zabavo ter številna druga področja. V zadnjih časih se anketiranje velikokrat uporablja za raziskovanje človeških odnosov do storitev in produktov v tržnem gospodarstvu. Rezultati in analize teh anket igrajo pomembno vlogo pri poslovnih odločitvah. V našem delu smo se omejili na grafičen prikaz rezultatov.

Na medmrežju obstaja veliko orodij za izdelavo anket. Predvsem so to samostojne spletne aplikacije, ki narejene ankete gostujejo, do njih pa dostopamo preko povezav, ki jih dobimo preko elektronske pošte ali na kakšnem drugem mediju. To je po našem mnenju nepriljučno in le majhen procent ljudi se odloči rešiti anketo.

Mislimo, da bodo uporabniki, kljub začetnim težavam pri vzpostavitvi povezave s servisom WCF, kasneje prihranili veliko časa, katerega bi sicer porabili za vsakokratno razvijanje svoje ankete. Da bi omilili začetne probleme, smo zagotovili tudi pomoč, opise in postopke, ki se nanašajo na tehnologijo WCF.

V prvem delu naloge smo opisali tehnologije, katere smo uporabljali pri razvoju aplikacije. Nadaljevali smo z opisom bistvenega pomena anket in kako jih snujemo, da so učinkovite. Opisali smo tudi načrtovanje in razvoj aplikacije *Ankete* ter razvoj servisa WCF. Na koncu smo opisali še razvoj odjemalca, ki predstavlja računovodski servis. S primeri in opisi smo predstavili, kako se servis izkorišča ter kako se uporablja določeno anketo.

# Poglavje 2

## Opis tehnologij

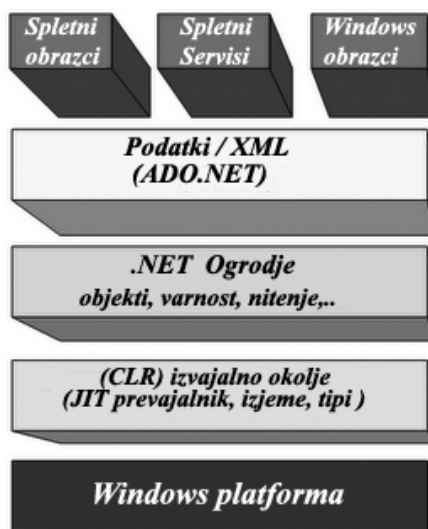
To poglavje je namenjeno opisu tehnologij, ki smo jih izbrali za izdelavo spletne aplikacije *Ankete*.

### 2.1 Ogrodje .NET

Ogrodje .NET je programsko okolje, katero je Microsoft razvil za izgradnjo, razvijanje in poganjanje aplikacij ter servisov, ki temeljijo na tehnologiji .NET. Okolje omogoča razvijalcem hiter razvoj, kot posledico pa dobimo optimalno, zmožljivo, učinkovito usmerjene aplikacije [6]. Veliko poudarka je na interoperabilnosti. Interoperabilnost je sposobnost več programov ali komponent, da med seboj izmenjujeta in uporabljata informacije [9].

Dva glavna modula ogordja .NET sta:

- **Izvajalnik kode CLR** (ang. Common Language Runtime), je osrednja komponenta ogordja .NET, ki predstavlja navidezni računalnik. Programerju se ni potrebno ukvarjati z značilnostmi posameznega procesorja. Programska koda se ne prevaja neposredno v strojno kodo, ampak se prevede v kodo vmesnega jezika. Posledica tega je, da se lahko programer pri razvoju projekta poslužuje različnih programskih jezi-



Slika 2.1: arhitektura .NET ogradja

kov, ki so nameščeni v ogradju .NET. CLR je tako imenovano izvajalno okolje za različne programske jezike, ki nadzoruje izvajanje in upravlja zahteve .NET programske kode [7]. To okolje zagotavlja servise, kot so upravljanje s spominom in nadzor nad izjemami (ang. exception) in napakami. Omogoča tudi varnostne mehanizme in razhroščevanje (ang. debugging).

- **Knjižnice razredov ogradja** (ang. Framework Class Library), je skupno ime za tisoče razredov, ki sestavljajo ogradje .NET in zagotavljajo bogat nabor vnaprej definiranih funkcionalnosti. Knjižnice so organizirane po imenskih prostorih (ang. namespaces) [8].

Glavne knjižnice ogradja .NET, ki so uporabljene v diplomskem delu, so:

- **System**, vsebuje bistvene razrede, ki predstavljajo najpogosteje uporabljene vrednosti, podatkovni tipi, vmesniki, atributi, procesne napake, itd.,

- **System.Collections**, glavni razredi za upravljanje s podatkovnimi zbirkami objektov,
- **System.Data**, večinoma vsebuje razrede, ki predstavljajo arhitekturo ADO.NET,
- **System.Diagnostics**, določa sposobnost diagnoze aplikacij (event logging, tracing..),
- **System.Net**, zagotavlja vmesnik za večino protokolov, ki jih internet danes uporablja,
- **System.Security**, razredi, ki omogočajo kriptografijo in zaščito,
- **System.Web.Services**, oskrbuje z razredi, ki omogočajo izdelavo XML spletnih strežnikov in odjemalcev, ki lahko uporabljajo HTTP, XML, XSD, SOAP, in protokole WSDL .

## 2.2 Microsoft Visual Studio 2008

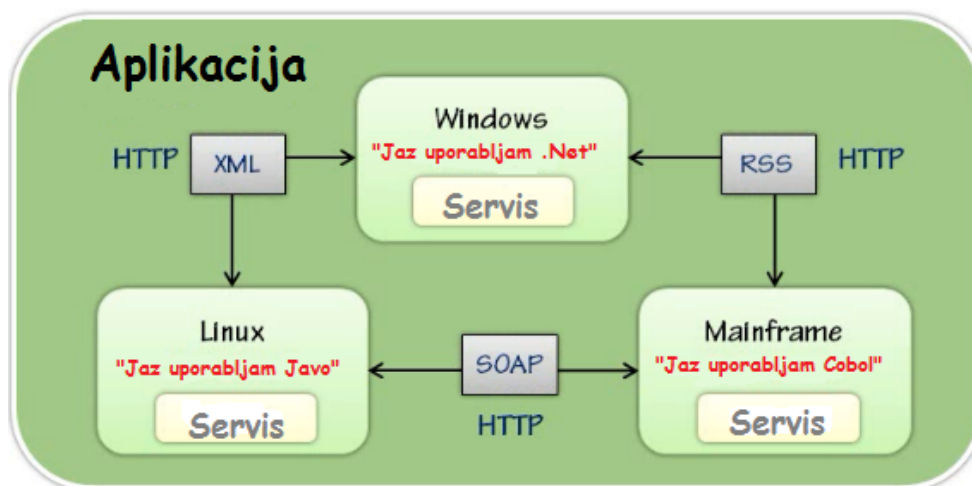
Microsoft Visual Studio je vodilno razvojno okolje za razvoj aplikacij na Microsoft ogrodju .NET. Vsebuje napredna razvojna orodja, razhroščevalne funkcije, orodja za delo z bazo in inovativna orodja za hiter razvoj jutrišnjih modernih aplikacij na različnih platformah [10].

Visual studio 2008 vsebuje razširitve, kot je grafični urejevalnik (ang. visual designer) za hitrejše razvijanje v okolju .NET Framework 3.5. Vsebuje tudi dodatne izboljšave pri orodjih za razvoj spletnih aplikacij in izboljšave podatkovnih tipov v programskem jeziku, kar omogoča hiter razvoj aplikacij.

Razvojno okolje Visual Studia 2008 je namenjeno profesionalnim razvijalcem. Podpira razvoj aplikacij na najnovejših platformah, ki vključuje splet (ASP.NET, AJAX), Windows Vista, Windows Server, Microsoft Office 2007, SQL Server 2008, igre (pc, xbox) in naprave Windows Mobile.

## 2.3 Servisi

Servis je funkcionalna enota, ki je na voljo svetu. S tega stališča je to nek napredek iz funkcije v objekt, iz objekta v komponento in iz komponente v servis. Servisna orientiranost (ang. Service orientation) je abstraktna množica principov in najboljših praks za gradnjo servisno orientiranih aplikacij. Servisno orientirana aplikacija združi servise, s katerimi operira, v samostojno logično aplikacijo, kot kaže Slika 2.2. Pri tehnologiji WCF so vsa sporočila med odjemalcem in servisom sporočila SOAP.

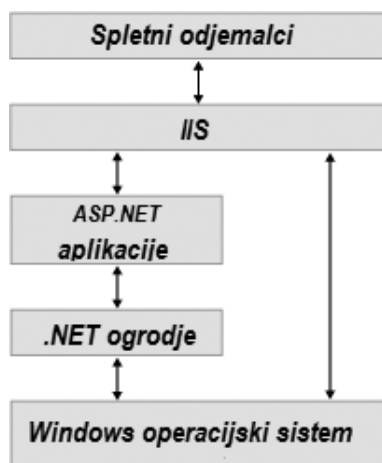


Slika 2.2: Servisi

## 2.4 Asp.NET

Asp.NET je Microsoftova tehnologija, ki omogoča izdelavo dinamičnih spletnih strani in spletnih servisov. Je del ogrodja .NET. Razvojno okolje nudi dobre pogoje za delo z HTML, CSS, JavaScript, jQuery in drugimi tehnologijami. Najpogostejša programska jezika za izdelavo Asp.NET spletnih aplikacij ter strani sta VB.NET in C#. Slednji je bil predstavljen vzporedno z tehnologijo Asp.NET, zato ga nekateri kličejo kar .NET jezik [13].

Bistvena razlika s predhodno tehnologijo asp je ta, da razvoj Asp.NET spletne strani stremi k objektno usmerjenemu načinu izvedbe. To pomeni, da je vsak element na taki strani obravnavan kot objekt, ki se izvaja na strežniku.



Slika 2.3: arhitektura Asp.NET

## 2.5 Microsoft SQL Server 2008 Express

Na začetku smo se soočili s problemom razvoja podatkovnega modela, za čigar izdelavo smo izbrali Microsoft SQL Server 2008 Express. To je Microsoftovo orodje za delo z relacijskimi podatkovnimi bazami (ang. Relational Database Management System). Ta verzija orodja je brezplačna, enostavna za uporabo, vsebuje orodje SQL Server Management Studio, namenjeno grafičnemu urejanju. SQL Server 2008 Express, zagotavlja bogate funkcije, varnost podatkov in veliko hitrost. Idealno je za majhne strežniške aplikacije in za lokalno hrambo podatkov.

## 2.6 JavaScript

JavaScript je najpopularnejši objektni skriptni jezik. Sintaksa tega jezika je podobna C++ in Javi, vendar je od niju popolnoma neodvisen. Jezik je raz-

vilo podjetje Netscape. Deluje na vseh glavnih brskalnikih, kot so Internet Explorer, Firefox, Chrome, Opera in Safari. Razvit je bil z namenom, da bi dodal interaktivnost spletnim stranem. Najpogosteje je vključen v HTML kodo na straneh in tako opravlja naloge, katere niso izvedljive statično ter izračunava in rokuje s predmeti v brskalniku. Primer je kontrola vnesenih vrednosti, prehod med stranmi, odpiranje oken, ipd. Je neodvisen, kar pomeni, da se skripta izvede brez predhodne pretvorbe (ang. compile).

## 2.7 jQuery

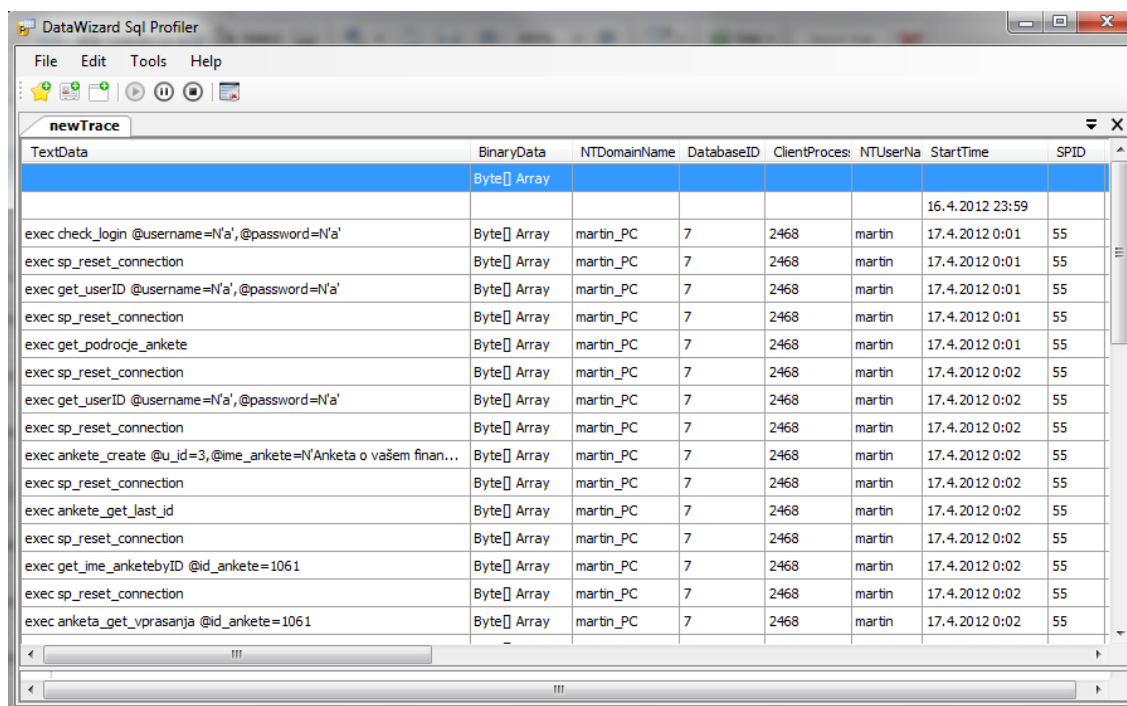
jQuery je JavaScript knjižnica, zasnovana za poenostavitev pisanja html skript in hitrejši razvoj spletnih aplikacij [11]. Primeren je za razvoj širokega spektra skript, poizvedb, vizualnih učinkov, itd. Omogoča nam reševanje težav v zvezi s spletnimi brskalniki, ker poenoti implementacije pri različnih brskalnikih.

Največkrat se uporablja za:

- nadzor nad dogodki,
- aplikacije Ajax,
- dodajanje animacij, vizualnih učinkov,
- podporo pri izpolnjevanju spletnih obrazcev.

## 2.8 AnjLab SQLProfiler

AnjLab Sql Profiler je brezplačno orodje za nadzor nad dogodki na strežniku Microsoft SQL Express. Na voljo je možnost izbire filtrov in dogodkov. Imamo pregled nad poizvedbami SQL, ki se izvajajo na strežniku. Pri vsaki poizvedbi je v realnem času razvidno, na kateri bazi se izvaja, kdo jo izvaja, porabi resursov itd. Na Sliki 2.4 vidimo grafični vmesnik programa.



TextData	BinaryData	NTDomainName	DatabaseID	ClientProces	NTUserNa	StartTime	SPID
	Byte[] Array					16.4.2012 23:59	
exec check_login @username='N'a',@password='N'a'	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:01	55
exec sp_reset_connection	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:01	55
exec get_userID @username='N'a',@password='N'a'	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:01	55
exec sp_reset_connection	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:01	55
exec get_podrocje_ankete	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:01	55
exec sp_reset_connection	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55
exec get_userID @username='N'a',@password='N'a'	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55
exec sp_reset_connection	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55
exec ankete_create @u_id=3,@ime_ankete='N'Anketa o vašem finan...	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55
exec sp_reset_connection	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55
exec ankete_get_last_id	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55
exec sp_reset_connection	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55
exec get_ime_anketebyID @id_ankete=1061	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55
exec sp_reset_connection	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55
exec anketa_get_yprasanja @id_ankete=1061	Byte[] Array	martin_PC	7	2468	martin	17.4.2012 0:02	55

Slika 2.4: Anjlab SqlProfiler

## 2.9 WCF (Windows Communication Foundation)

Windows Communication Foundation je razvojno računalniško okolje za izdelavo porazdeljenih sistemov na Microsoftovi platformi [1]. Aplikacije WCF izpostavlja svoje funkcionalnosti s pomočjo servisov.

Pri snovanju porazdeljenih aplikacij poznamo številne pristope. Pomembnejši so:

- XML spletni servisi,
- delo z oddaljenimi objekti (ang. .NET Remoting),
- sporočilna vrsta (ang. Message Queuing (MSMQ)),
- COM+ (ang. Component Object Model).

Vsaka izmed teh tehnologij ima svoj tip servisnega vmesnika. Zaradi različnosti tehnologije zahtevajo posebno znanje za njihovo kodiranje. To je samo po sebi napeljevalo k razvoju nove tehnologije. Tehnologija WCF je na tem področju naredila velik korak naprej, saj nadomesti zgoraj naštete. Lahko rečemo, da vztrajno prihaja v širšo uporabo, saj razvijalcem močno olajša delo in pripomore k interoperabilnosti med programi oziroma sistemi.

Tehnologija WCF, katero so na začetku razvoja poimenovali z imenom »Indigo«, je prvič vključena v namestitvi ogrodja .NET 3.0, za svoje delovanje pa zahteva vsaj ogrodje .NET 2.0. WCF predstavlja nov nabor komunikacijskih orodij pri razvoju in uporabi servisov. Ta orodja omogočajo povezavo več programov in implementacijo storitvene usmerjenosti v rešitvah, ki tak način komunikacije potrebujejo.

## Poglavje 3

# Ankete in njihova uporaba

Anketa je orodje, ki nam pomaga priti do določenih znanj in spoznanj. Večina jih je namenjenih za podporo izboljšanju kvalitete produktov, izboljšanju trženja, za psihološke ali sociološke raziskave, za zabavo ter številna druga področja.

Razvoj anketnega raziskovanja temelji na namenih uporabe. Najprej so se anket posluževale vladne, statistične, vojaške in druge službe. Uporabljajo se za razumevanje družbenih problemov, v zadnjih časih pa se velikokrat uporabljajo za raziskave človeških odnosov do storitev in produktov v tržnem gospodarstvu.

Področju, ki se ukvarja z načini anketiranja, pravimo anketna metodologija [5]. Ta stremi k pridobitvi čim kvalitetnejših podatkov ob čim manjši porabi stroškovnih in drugih virov.

### 3.1 Pravila anketiranja

Pri izdelavi anket moramo upoštevati naslednja pravila:

1. Vprašanja razporejamo:
  - od enostavnih k zapletenim vprašanjem. S tem anketiranca privabimo v reševanje in težje prekine anketiranje. Vzbudimo tudi zaupanje,

- od zanimivih k manj zanimivim. Vzbudimo interes,
- od splošnih k specifičnim.

2. Vprašanja v anketi morajo biti:

- specifična. Če vprašanje zahteva mnenje o videzu in kvaliteti nekega predmeta, ti dve lastnosti ločimo v posamezni vprašanji,
- nedvoumna. Anketirancu mora biti popolnoma jasno, kaj se od njega pričakuje,
- nesugestivna. Izogibamo se vprašanjem, ki so podana tako, da spodbujajo k izražanju mnenj. Sugestivna so tudi vprašanja, ki so čustveno nabita. Primer: Drugače se sliši tat, kot oseba, ki je izvršila kaznivo dejanje,
- vprašanja ne smejo vzbujati ogroženosti. Ne dotikamo se človeških vrednot, na katere so ljudje ponavadi občutljivi,
- jezik prilagajamo ciljni množici anketirancev.

## 3.2 Tipi anket

Tipi anket oziroma vprašalnikov, ki so najpogosteje v uporabi, so:

- **intervju.** Pri tem načinu pridemo do podatkov neposredno. Največkrat preko pogovora med anketirancem in anketarjem. Intervju terja veliko časa, saj se izvajajo po različnih lokacijah, kar doprinese k večjim stroškom. Prednost tega tipa, je da dobimo relativno podrobne in nedvoumne odgovore, saj ima anketiranec možnost pojasniti,
- **telefonske ankete.** Podatki se zbirajo preko telefonskega pogovora. Anketarji morajo imeti vsa potrebna znanja za tovrstno anketiranje. To veliko stane, saj je treba anketarje usposobiti. Prednost je v tem da lahko v razmeroma kratkem času pridemo do veliko podatkov,

- **ankete preko pošte ali e-pošte.** Podatki se zbirajo preko pošiljanja vprašalnika po pošti na naslove anketirancev. Pri tej obliki anketiranja je slabost to, da moramo imeti na razpolago vse naslove. Slabost je tudi, da je tovrstno anketiranje dolgotrajno in precej manj učinkovito, ker anketiranci nimajo dodatnih pojasnil kako reševati anketo,
- **spletne ankete.**

### 3.3 Spletne ankete

Ena največjih prednosti spletnih anket je ta, da je zbiranje in obdelava podatkov avtomatizirana, kar uporabniku prihrani veliko zamudnega dela pri raziskovanju. Avtomatiziran je tudi prikaz rezultatov s pomočjo tabel, grafov, razpredelnic.

Količina podatkov na internetu je v današnjem času vedno večja. Postavlja se vprašanje, kakšna je količina informacije, katero ti podatki predstavljajo. Se pravi, koliko izmed teh podatkov je verodostojnih in koliko ne. Večina ljudi iz vseh demografskih skupin se kot splošnega informacijskega medija največkrat poslužuje interneta. Njegova vloga postaja vse pomembnejša, tradicionalni načini komunikacije pa izgubljajo svoj pomen. Z razvojem interneta se spreminja način zbiranja javnomnenjskih raziskav in anketiranja, saj poznamo vedno več metod, podprtih z informacijsko tehnologijo, kar je vse skupaj cenovno pocenilo, poenostavilo in pohitrilo.

Pozitivne lastnosti spletnih anket, so:

- nižja cena (v primerjavi z anketami preko e-pošte, telefona, intervjuji),
- z meta podatki dobimo natančnejši vpogled v vzorce respondentov (npr. koliko časa so reševali posamezno vprašanje oz. celotno anketo),
- hitrost in učinkovitost zbiranja podatkov,
- možnost vključitve večpredstavnosti v ankete (slika, video, animacije),

- odzivnost ljudi je večja, predvsem zaradi anonimnosti,
- anketiranec ima več časa za premislek in je precej bolj iskren,
- zmanjševanje napak pri vnosu v podatkovno bazo,
- presežene so časovne in geografske omejitve,
- elektronska oblika podatkov (kakovostnejši podatki).

Nekatera področja, na katerih se spletne ankete uporabljajo, so:

- merjenje stopnje zadovoljstva kupcev in potrošnikov,
- raziskovanje uporabe interneta, ocenjevanje spletnih strani,
- poskusi razumevanja družbenih problemov in psiholoških eksperimentov,
- za raziskave javnega mnenja.

# Poglavje 4

## Razvoj spletne aplikacije Ankete

V tem poglavju je predstavljen razvoj aplikacije od analize zahtev do končnega videza narejene aplikacije.

### 4.1 Razvoj po metodologiji RUP

Po premisleku, kako bo potekal razvoj aplikacije, kakšne bodo njene glavne funkcionalnosti, njena arhitektura, videz, smo se odločili za vpeljavo procesa RUP (Rational Unified Process [3] )

Metodologija RUP obsega vse, kar počnemo, da bi dosegli želen rezultat, torej izdelek ali storitev, ki je cilj našega dela. Daje nam smernice za učinkovit razvoj kakovostne programske opreme. Zmanjšuje tveganje in povečuje predvidljivost. RUP velja za težko metodologijo, saj zajema in podrobno opisuje številne elemente. Kljub temu se je izkazala kot primerna izbira. RUP za vizualizacijo uporablja jezik UML [2]. Glavni namen te metodologije je uspešen razvoj programske opreme.

Veliko vlogo pri izbiri te metodologije razvoja so imele tudi naše preferenčne

želje, objektno usmerjen razvoj ter naše znanje in izkušnje s področja razvoja informacijskih sistemov. Svoj doprinos k tej izbiri je dodalo tudi to, da tehnologija podpira iterativni razvoj. Uporabili smo tudi diagrame primerov uporabe, ki so pri jeziku UML nepogrešljivi.

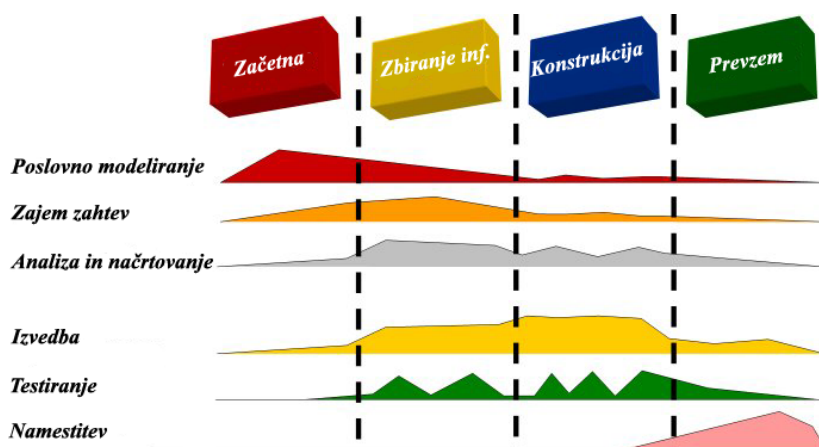
Metodologija RUP je baza znanja, ki je nastala kot rezultat prizadevanja za bolj učinkovito objektno usmerjen razvoj programske opreme na ravni organizacije. Opisuje kako učinkovito uporabiti šest najboljših izkušenj s področja razvoja programske opreme [4]:

- Iterativni razvoj. Temelji na izvajanju več iteracij oziroma ponovitev, kjer v vsaki iteraciji razvijemo del funkcionalnosti sistema.
- Obvladovanje zahtev. Tukaj gre za sledenje zahtev od začetka, preko sprememb, do zaključka.
- Uporabo komponente arhitekture. Razbijanje razvoja na več delov, kar zmanjša zapletenost sistema.
- Vizualno modeliranje. Prikaz modela sistema z uporabo jezika UML.
- Preverjanje kakovosti. Gre za ocenjevanje kakovosti izdelka z vidika funkcionalnosti, zanesljivosti in zmogljivosti sistema.
- Nadzorovanje sprememb. Tukaj gre za redno spremljanje in poročanje. Pri nadzoru pomagajo struktura načrtov in vnaprej določena poročila.

RUP vključuje štiri faze življenjskega cikla razvoja:

- začetna faza: vzpostavitev projekta, opredelitev okvirjev področja, ki ga obravnavamo, načrtovanje virov,
- zbiranje informacij: načrtovanje arhitekture, specifikacija značilnosti, zbiranje informacij o obravnavanem področju,
- konstrukcija izdelka: zajema kodiranje in oblikovanje izdelka,
- prevzem: namestitev in predaja izdelka v uporabo končnemu uporabniku.

Pri metodologiji RUP je značilno, da razvoj poteka iterativno. Iteracija je specifično zaporedje aktivnosti, izvedenih na osnovi načrta in z določenim kriterijem vrednotenja, ki se konča z izdajo izdelka. Aktivnosti v iteracijah tečejo vzporedno Slika 4.1. V vseh iteracijah imamo opravka predvsem s poslovnim modeliranjem in zajemom zahtev. V naslednjih iteracijah prevladuje aktivnost analize in načrtovanja. Testiranje je prisotno skozi vse štiri razvojne iteracije.



Slika 4.1: Vzporeden tok aktivnosti

## 4.2 Analiza zahtev in načrtovanje

Analiza in načrtovanje sta najpomembnejša dejavnika za uspešno izvedbo razvoja aplikacije. Če je izvedba teh dveh dejavnikov primerna, lažje dosežemo ustrezno razumevanje sistema, ga poenostavimo ter zagotovimo ponovno uporabo. S tem korakom ugotovimo informacijske zahteve, kaj sistem potrebuje da bo učinkovito deloval, katere podatke potrebujemo in kateri postopki te podatke spreminjajo.

V tem poglavju bomo opredelili:

- funkcionalne zahteve (diagram primerov uporabe, podatkovni model),
- nefunkcionalne zahteve.

### 4.2.1 Funkcionalne zahteve

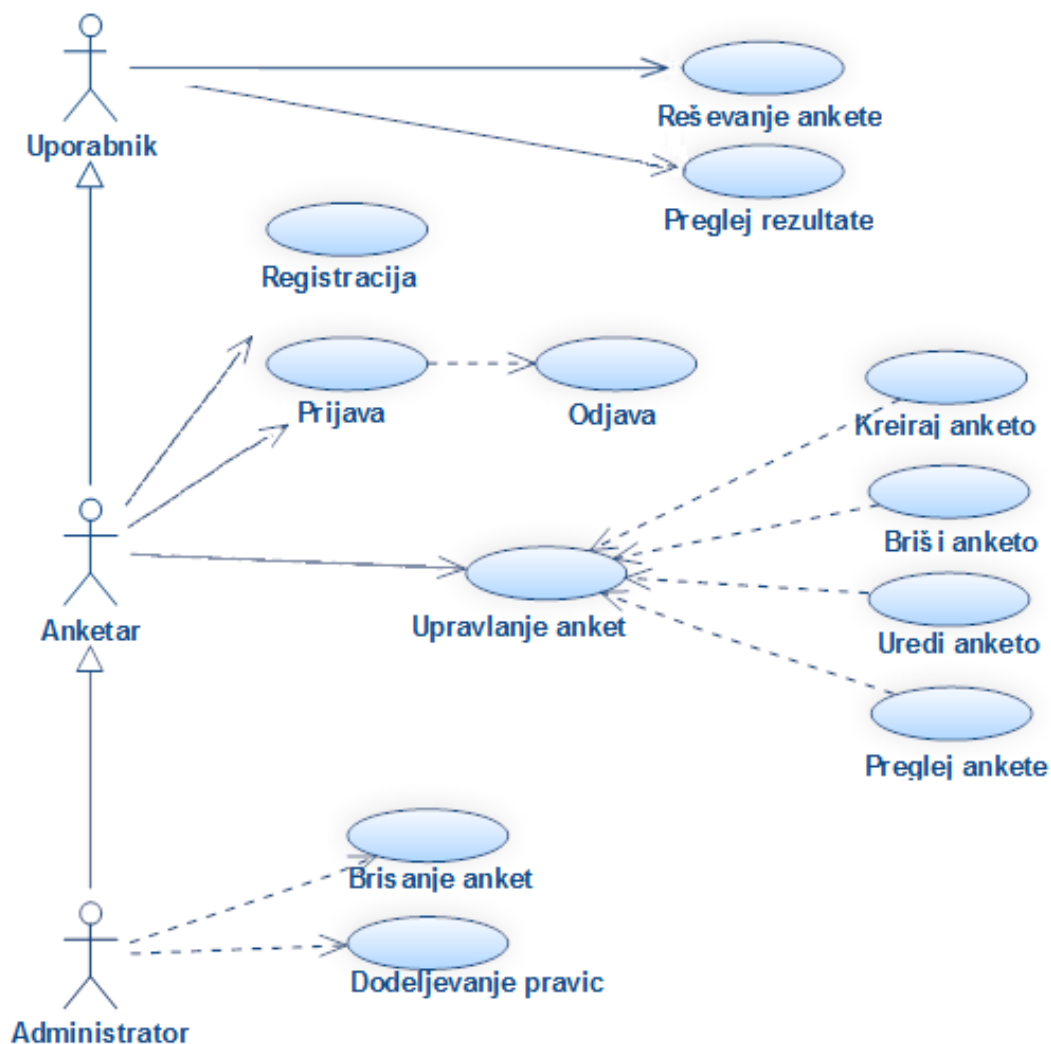
Funkcionalne zahteve so zahteve, ki se nanašajo na želeno funkcionalnost sistema. Predstavili jih bomo z diagramom primerov uporabe. Podroben opis in pogoje bomo opisali tekstovno. Ponavadi te funkcionalnosti sistema poda naročnik sistema. Ker smo si diplomsko delo sami zamislili, smo v tem primeru naročnik mi.

Sodelujoči akterji v sistemu so:

- uporabnik spletne strani odjemalca (neprijavljen uporabnik),
- anketar v spletni aplikaciji *Ankete* (prijavljen uporabnik),
- administrator v spletni aplikaciji *Ankete*.

Funkcionalne zahteve za neprijavljenega uporabnika veljajo tudi za vse druge akterje, prav tako veljajo funkcionalne zahteve anketarja tudi za administratorja. To pomeni, da velja generalizacija po prej navedenem vrstnem redu akterjev.

Tekstoven opis funkcionalnih zahtev:

Slika 4.2: Diagram primerov uporabe aplikacije *Ankete*

1. Uporabnik lahko rešuje ankete, ki so jih naredili prijavljeni uporabniki – anketarji
2. Uporabnik lahko pregleduje rezultate že rešenih anket.
3. Anketar ima na voljo registracijo. S tem dobi pravico do ustvarjanja ene ali več anket po meri. Podati mora ime, priimek, uporabniško ime,

geslo in elektronski naslov.

4. Po prijavi anketar dobi na elektronski naslov sporočilo z informacijami o njegovi registraciji. S temi podatki se lahko anketar prijavi v sistem.
5. Anketar in administrator imata možnost odjave iz sistema.
6. Anketarju in administratorju se po prijavi izriše njegov uporabniški vmesnik. Na podlagi pravic, ki jih ima, se omogočajo gumbi za urejanje ankete.
7. Administrator ima na voljo svoj modul, kjer upravlja z anketami.
8. Anketar lahko naredi novo anketo. Med izdelavo ima omogočeno brisanje in urejanje vprašanj.

Za lažje razumevanje, smo izdelali diagram primerov uporabe Slika 4.2.

#### 4.2.2 Nefunkcionalne zahteve

Nefunkcionalne zahteve se nanašajo na tehnične in druge nevsebinske zahteve sistema.

- Dostopnost:  
Ker mora biti sistem dostopen širši množici uporabnikov, je smotrno uporabiti spletno tehnologijo. Aplikacija mora biti ves čas dostopna na strežniku. Kar se tiče programske opreme, potrebujemo spletni strežnik IIS, Asp.NET programsko okolje, Microsoft .NET ogrodje verzija 3.5 z vgrajeno tehnologijo WCF, Microsoft Server 2008 Express.
- Performanse:  
Omogočeno mora biti tekoče delovanje sistema. Zaradi časovnih omejitev pri komunikaciji WCF je pomembna hitrost internetne povezave.
- Upravljanje podatkov:  
Sistem upravlja z večjo količino podatkov, zato mora biti komunikacija

z relacijsko podatkovno bazo učinkovita (skrbniške funkcije, nadzor nad transakcijami).

- Varnost:

Sistem bo hranil vse podatke v podatkovni bazi, kar samo po sebi zagotavlja zadostno stopnjo varnosti, zato ne potrebujemo posebnih varnostnih mehanizmov.

- Uporabniški vmesnik:

Učinkovit uporabniški vmesnik bistveno vpliva na uporabnost orodja. Mora biti zanimiv za pogled, mirnih barv, in prilagojen za hitro upravljanje z aplikacijo.

### 4.2.3 Podatkovni model sistema

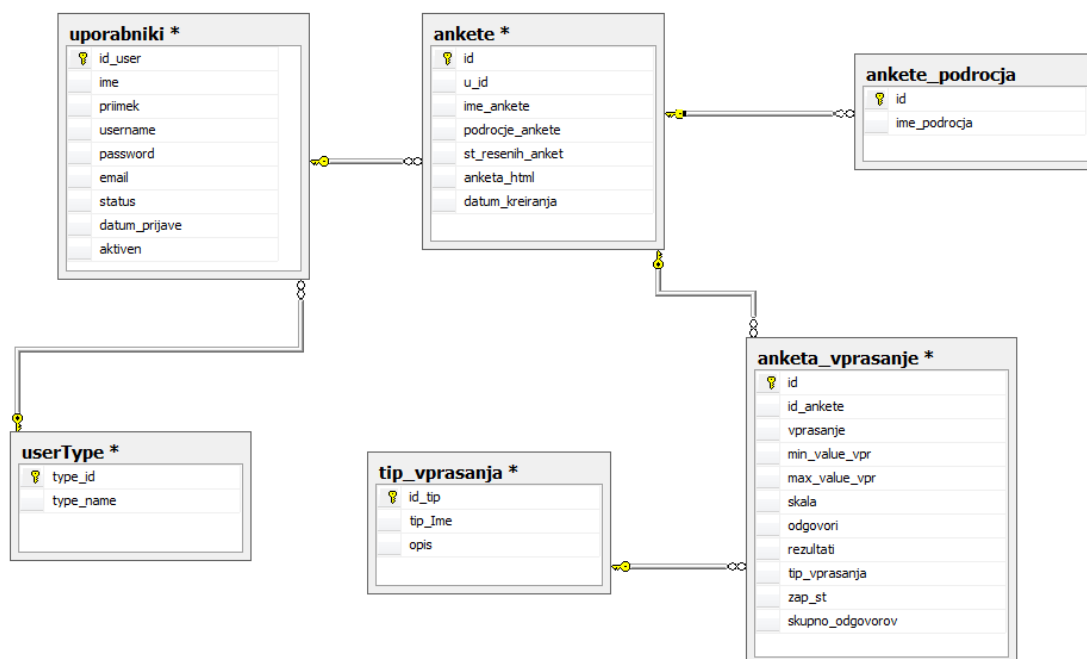
Za lažjo predstavitev sistema prikažemo relacijski podatkovni model s pomočjo diagrama Slika 4.3.

Celoten podatkovni model zajema šest entitet in več kot štirideset shranjenih procedur. Iz tega sklepamo, da ima vsaka entiteta približno 6 procedur, ki omogočajo primarne operacije nad podatki (branje, dodajanje, urejanje, brisanje).

Jedro podatkovnega sistema je entiteta *anketa\_vprasanje*, ki hrani glavne podatke o rešenih anketah.

## 4.3 Razvoj spletne aplikacije

Potrebno je bilo razviti rešitev, ki bo prijavljenim uporabnikom spletne aplikacije omogočala izdelavo anket. Te ankete, bodo zatem na voljo razvijalcem, da jih preko servisa WCF prenesejo na lastne spletne strani in jih tam

Slika 4.3: Relacijski podatkovni model sistema *Ankete*

vključijo v uporabniško okolje. Zatem bodo naključni uporabniki ankete reševali, rezultati pa se bodo preko servisov WCF vračali v spletno aplikacijo Ankete v obdelavo. Ankete bodo na teh straneh vključene na poljubnih lokacijah ali okvirjih.

Po analizi zahtev sistema smo se osredotočili na načrtovanje in izdelavo podatkovne baze. S pomočjo diagrama primerov uporabe smo prišli do ugotovitve, da za hrambo podatkov in za konsistentno delovanje sistema, potrebujemo naslednje entitete.

- *anketa\_vprasanje* – hrani podatke o posameznem vprašanju, kot je tekst vprašanja, tip vprašanja, možni odgovori, rezultati na te odgovore in druge potrebne podatke. Vsako vprašanje ima enolični identifikator id in je vezano na določeno anketo. Ena anketa ima enega ali več vprašanj.
- *ankete* – hrani podatke o posamezni anketi, kot so ime ankete, id anke-

tarja, področje ankete, število rešenih anket. Hrani tudi celotno html kodo ankete, katera hrani vse kontrole, ki so bile dodane na anketo. Ta html koda nam služi za prenašanje preko servisa WCF na odjemalca.

- *uporabniki* – hrani podatke o registriranih uporabnikih. Vsebuje primarni ključ `id_user`, ime, priimek, uporabniško ime, geslo, elektronski naslov in status.
- *ankete\_podrocja* – hrani interesna področja anket. Uporablja se za lažje sortiranje in iskanje anket.
- *tip\_vprasanja* – hrani tipe vprašanj. To so stopenjsko, več možnosti, da/ne/neopredeljeno, matrično in tekstovno.
- *userType* - hrani tip uporabnika. Ta je lahko administrator ali pa navaden anketar.

### 4.3.1 Idejna rešitev

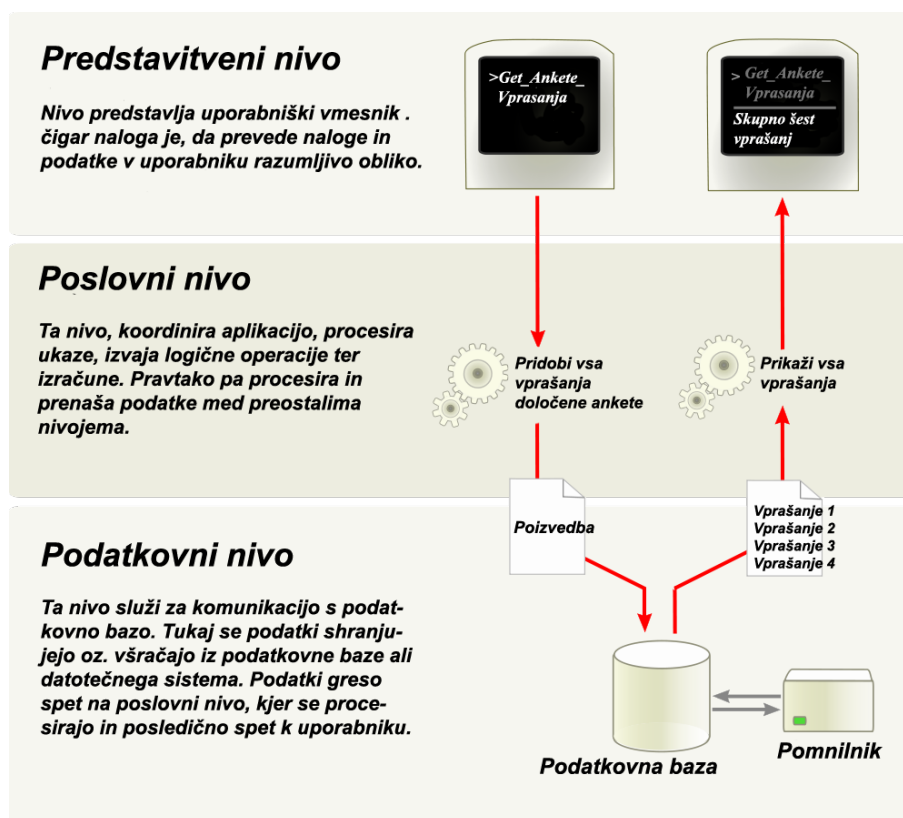
Aplikacijo *Ankete* sestavljata dva ločena modula. Prvi modul je namenjen anketarjem za izdelavo ankete, recimo mu *anketarjev modul*. Drugi modul je implementiran kot servis WCF, namenjen odjemalcem.

*Anketarjev modul* ni preveč tehnološko zahteven, saj je nabor podatkov relativno preprost in ne pretirano obsežen. Predstavlja večinski del celotne aplikacije. Po shranjevanju rešene ankete imamo na voljo vpogled v grafičen prikaz rezultatov. To smo implementirali s pomočjo kontrole `ChartControl` za izris grafov. Na voljo je več vrst grafičnih prikazov (Tortni graf, histogram,..).

Poudarek smo dali tudi na navodila in pomoči, ki se nanašajo na uporabo naše aplikacije preko tehnologije WCF. Sem spadajo splošna navodila in opis, kaj sploh servisi WCF so, kako so implementirani, kako se do njih dostopa, katere funkcionalnosti so omogočene ter še veliko drugih stvari.

Aplikacijo smo razvijali v tehnologiji Asp.NET, njen podatkovni del pa je razvit s pomočjo orodja SQL Server in tehnologije za podatkovne transakcije T-SQL. Največjo vlogo pri izbiri te tehnologije so imele naše izkušnje s temi orodji. Odgovarjala nam je tudi izdelava spletnih strani po principu glavnavebinska stran (ang. master-content page)

Asp.NET tehnologija nam omogoča strukturiran razvoj, zato smo aplikacijo zasnovali po principu tri-nivojske arhitekture. Pri tej arhitekturi imamo predstavitveni, poslovni in podatkovni del ločen med sabo Slika 4.4.



Slika 4.4: Shema trinovojske arhitekture aplikacije *Ankete*

### 4.3.2 Prijava v sistem

Kot smo že povedali, lahko spletno aplikacijo ankete polno uporabljajo le prijavljeni uporabniki. Ko želimo ustvariti anketo, nas v primeru, da nismo prijavljeni, avtomatično preusmeri na obrazec za prijavo. Pri prijavi moramo podati svoje uporabniško ime in geslo. Neprijavljeni uporabniki imajo možnost reševanja le že narejenih anket ter pregledovanja rezultatov. Prijava uspe v primeru, da v podatkovni bazi obstajata ime in njemu pripadajoče geslo Slika 4.5. V nasprotnem primeru imamo možnost registracije. Seja prijave je aktivna dvajset minut. V primeru, da uporabnik ni aktiven več kot dvajset minut, se seja prekine.

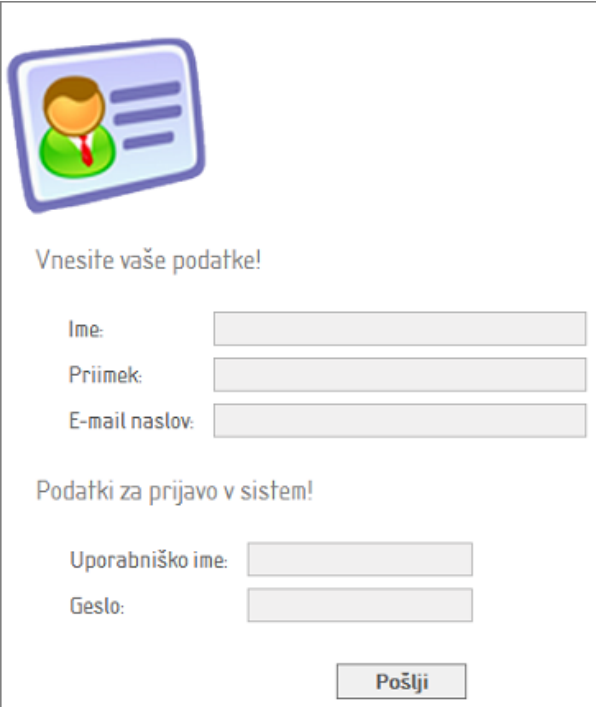


The image shows a login interface. At the top left is a 3D graphic of a person holding a key. Below it, the word 'PRIJAVA:' is centered. There are two input fields: 'Upor. ime:' and 'Upor. geslo:'. To the right of the second field is a 'Prijavi' button. At the bottom left is a link 'Registriraj se!' and at the bottom right is a link 'Pozabljeno geslo'.

Slika 4.5: Vnosna polja pri prijavi v sistem

### 4.3.3 Registracija

Pri registraciji moramo vnesti svoje ime, priimek, uporabniško ime, geslo in elektronski naslov, na katerega nam aplikacija posreduje elektronsko sporočilo z našimi podatki. Vsa polja so obvezna. V primeru napake oziroma praznega polja, nas aplikacija opozori z napisom »napačen vnos«, ki se pojavi poleg polja Slika 4.7.

The image shows a registration form with a header icon of a person. Below the icon, the text 'Vnesite vaše podatke!' is displayed. There are three input fields for 'Ime:', 'Priimek:', and 'E-mail naslov:'. Below these, the text 'Podatki za prijavo v sistem!' is shown, followed by two input fields for 'Uporabniško ime:' and 'Geslo:'. A 'Pošlji' button is located at the bottom right of the form area.

Vnesite vaše podatke!

Ime:

Priimek:

E-mail naslov:

Podatki za prijavo v sistem!

Uporabniško ime:

Geslo:

Pošlji

Slika 4.6: Vnosna polja pri registraciji

#### 4.3.4 Pozabljeno geslo

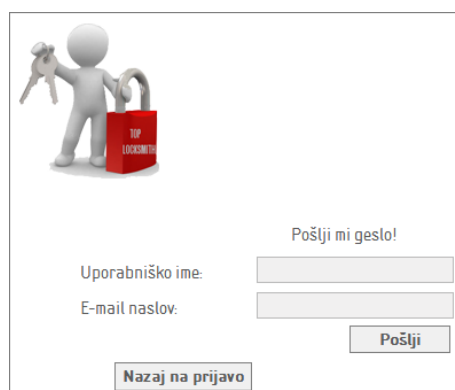
V primeru da uporabnik pozabi ime ali geslo, aplikacija avtomatično pošlje ime in geslo na uporabnikov elektronski naslov.

#### 4.3.5 Izdelava ankete

V meniju na levi izberemo razdelek »ustvari anketo«. Po uspešni prijavi se pojavi obrazec za izdelavo nove ankete. Prvo izberemo ime ankete, nato v spustnem polju izberemo njeno interesno področje Slika 4.8.

#### 4.3.6 Dodajanje vprašanj in možnih odgovorov

V tem delu aplikacije lahko dodajamo vprašanja in oblikujemo možne odgovore. Na voljo imamo pet tipov vprašanj. V kontroli z zavihki (ang. tab



Slika 4.7: Obrazec za obnovo up. imena in gesla



Slika 4.8: Izbira imena ankete

control) izberemo zavihek s tipom vprašanja, ki nam ustreza in vnesemo attribute, katere naj vprašanje vsebuje. Ob kliku na gumb »Dodaj vprašanje«, se vprašanja dodajajo eno pod drugo na stran Slika 4.9.

Na spodnji sliki Slika 4.10, lahko vidimo primer vseh tipov vprašanj v eni anketi.

- Stopenjsko – sami izberemo lestvico oziroma stopnje, ki naj jih ima odgovor na izbiro. Na izbiro imamo 2, 3, 5 ali 8 stopenj.
- Več možnosti – pri tem tipu, prvo izberemo število možnih odgovorov. Na voljo imamo 2, 3, 4 in 5. Za tem nam aplikacija dinamično doda vnosna polja za vsak možen odgovor.

\*Polje za vprašanja. S klikom na gumb 'Dodaj vprašanja', se bodo eno pod drugim dodajala na stran.

Ali vam na počitnicah odgovarjajo ribe?

\*Izberite tip vprašanja

Stopenjsko   **Več možnosti**   Da/Ne/Neopredeljeno   Matrično   Tekstovno

Najmanjša vrednost: ne

Največja vrednost: da

Lestvica od 1 do -> 5   primer: malo ○○○○○ veliko

Dodaj vprašanje

Slika 4.9: Kontrola z zavihki za izbiro tipa vprašanja

- Da Ne Neopredeljeno – pri tem tipu vprašanj je na voljo odgovor z tremi možnostmi. Te so da, ne ali neopredeljeno.
- Matrično – pri tem tipu prvo izberemo število kriterijev v odgovoru, ki jih zatem z gumbom »Dodaj«, dodamo. Te kriterije oziroma odgovore nato ocenjujemo z ocenami zelo slabo, slabo, srednje, dobro ali odlično.
- Tekstovno – pri tem tipu vprašanja se zahteva tekstovni odgovor. Največje število znakov je omejeno na 250.

### 4.3.7 Reševanje anket in prikaz rezultatov

Ko smo anketo uspešno izdelali in jo shranili, je na voljo v razdelku »reši anketo« v meniju na levi. Ankete so sortirane po interesnih področjih, ki jih izberemo v spustnem polju. V razpredelnici lahko sortiramo ankete po času, ko so bile izdelane. Anketo aktiviramo za reševanje s klikom na gumb »reši« Slika 4.11.

Na voljo imamo tudi vpogled v rezultate rešenih anket, ki so prikazani grafično. Rezultati procentualno prikazujejo povprečne vrednosti odgovorov Slika 4.12. Na sliki je prikazan primer za dve vprašanji.

## S kakšno hrano se prehranjujete, ko ste na morju?

**1. Ali vam na počitnicah odgovarjajo ribe?**

ne       da Briši

**2.kafera hrana vam najbolj odgovarja na počitnicah?**

Meso  
 Zelenjava  
 Škrob  
 Maščobe Briši

**3.Se boste na naslednjem dopustu ukvarjali s športom?**

da  ne  neopredeljeno  Briši

**4.Oceni najprimernejšo posodo za pripravo hrane na počitnicah!**

z.slabo slabo srednje dobro z.dobro

električni Pizza Pan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Briši
Žar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Ponev	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Lonec	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Mikrovalovka	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

**5.Opišite vašo najljubšo počitniško hrano!**

Briši

Še  znakov!

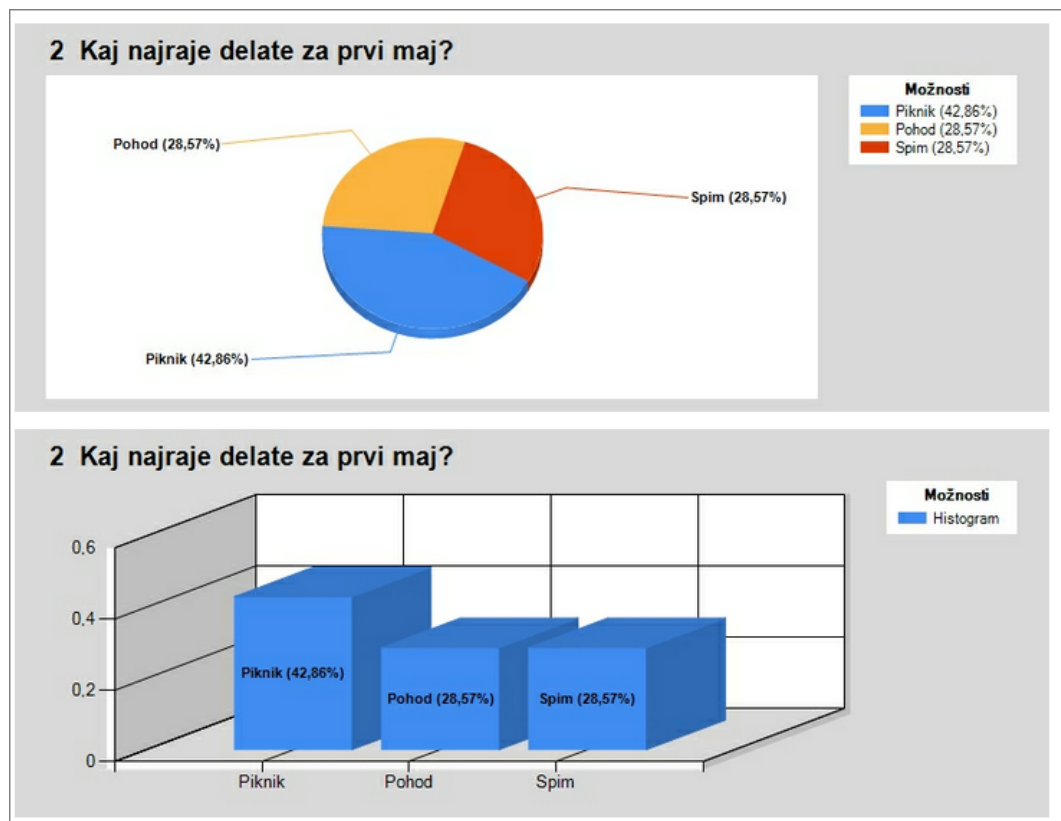
**Shrani ankefo za reševanje!**

Slika 4.10: Primer ankete z vsemi petimi tipi vprašanj

Področje:

	Anketa	ID Ankete	Ustvarjena ^ v
<b>Reši</b>	S kakšno hrano se prehranjujete, ko ste na morju?	1084	12.06.12
<b>Reši</b>	Anketa o prvomajskih praznikih!	1062	26.04.12
<b>Reši</b>	Anketa o politični opredelitvi!	969	07.02.12

Slika 4.11: Obrazec, kjer izberemo anketo za reševanje



Slika 4.12: Prikaz rezultatov

## Poglavje 5

# Razvoj servisa WCF in odjemalca WCF

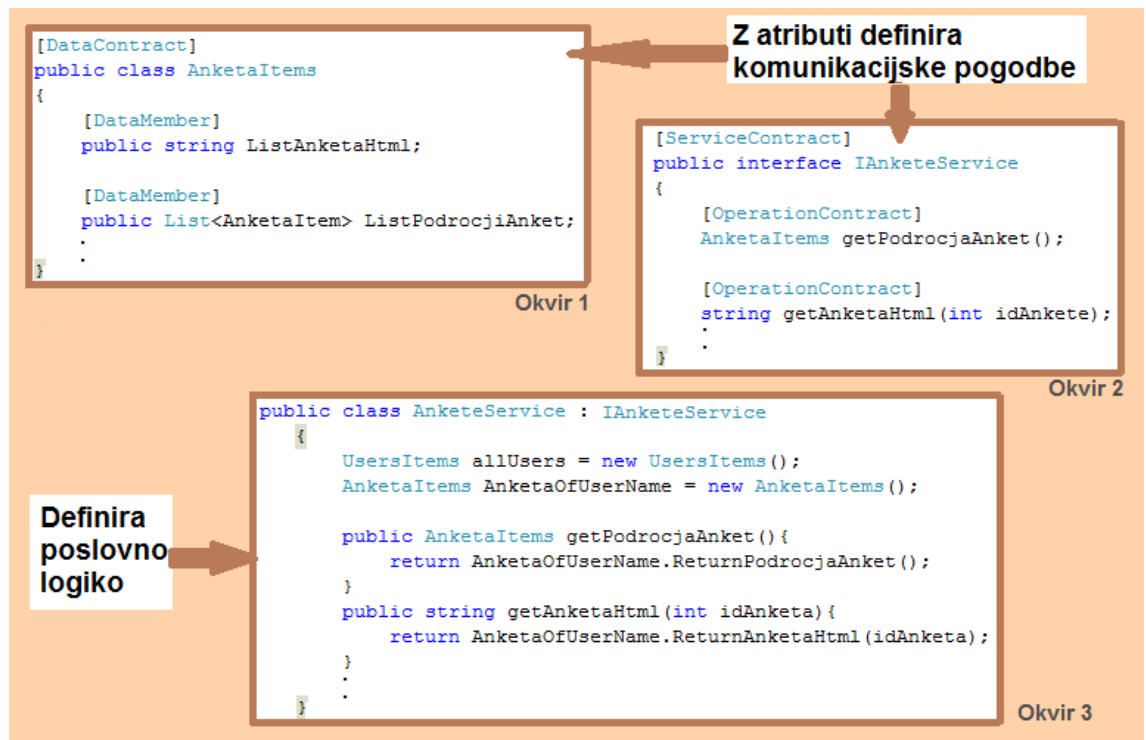
V tem poglavju je opisan razvoj in uporaba servisa WCF. Osredotočili smo se na glavne korake, ki jih moramo izvesti, da dobimo stabilen in uporaben servis.

### 5.1 Umestitev servisa v aplikacijo Ankete

Ko smo imeli spletno aplikacijo ankete okvirno narejeno, smo začeli z izdelavo spletnega servisa WCF. Danes se od večine porazdeljenih aplikacij pričakuje, da zagotavljajo nek servisni vmesnik (ang. service interface), ki je na voljo drugim aplikacijam oziroma odjemalcem. Ta vmesnik ima nalogo, da odjemalcem zagotavlja dosegljivost spletnih anket, ki so jih z aplikacijo *Ankete* naredili.

### 5.2 Programski del servisa WCF

Aplikacije WCF omogočajo funkcionalnost s pomočjo servisov. Servis je CLR tip, ki vsebuje poslovno logiko in množico metod, ki so dostopne odjemalcem. CLR tip deluje kot servis samo v primeru, da implementira servisno pogodbo.



Slika 5.1: primer kode pri servisu WCF

### 5.2.1 Podatkovne pogodbe

Razred *AnketaItems* predstavlja sporočilo, ki ga bo ta servis uporabljal. To sporočilo bo vsebovalo vse postavke, katere anketa potrebuje. Ko razred označimo z atributom [DataContract], omogočimo njegovo serializacijo. Eksplisitno z atributom [DataMember] moramo označiti, katere postavke hočemo imeti v našem sporočilu Slika 5.1 - (okvir 1). S tem dosežemo to, da se objekti .NET serializirajo v sporočila in obratno. Če postavki atributa [DataContract] ne dodamo, ta ni vključena v sporočilo.

### 5.2.2 Servisne pogodbe

Servisna pogodba je definirana takrat, ko vmesniku *IAnketeService* dodamo atribut [ServiceContract]. Vsak razred, ki ta vmesnik implementira, postane

servisni tip. Ko definiramo vmesnik (ang. interface), moramo z atributi [OperationContract] eksplicitno označiti, katere metode oziroma funkcionalnosti želimo, da naš servis vsebuje Slika 5.1 - (okvir 2). Tako označene metode postanejo del servisne pogodbe in so na voljo odjemalcem. Pri vsaki izmed metod servisa lahko atributu [OperationContract] dodamo pravila, ki opisujejo, kako se sporočila izmenjujejo. Prav tako lahko pravila dodamo tudi atributu [ServiceContract]. Na tak način postane servis bolj sofisticiran in omejen, s tem pa tudi robustnejši. Attribute [ServiceContract] in [DataContract] najdemo v knjižnici *System.ServiceModel*.

### 5.2.3 Implementacija servisa in poslovne logike

Ko imamo servisne in podatkovne pogodbe urejene, lahko servis implementiramo. To naredimo tako, da servisni razred *AnketaService* izpeljemo iz vmesnika *IAnketaService* Slika 5.1 - (okvir 3). Zatem implementiramo vse metode, ki definirajo poslovno logiko servisa. Na tej točki lahko servisu in njegovim metodam dodamo še attribute obnašanja (ang. behavior). Z atributoma [ServiceBehavior] in [OperationBehavior] določamo, kako se servis instancira, nitenje, ali so določeni podatki obvezni ali ne, ter mnoge druge parametre.

## 5.3 Gostovanje servisa WCF

Vsak servis WCF mora nekje gostovati, če hočemo, da obstaja. Vsak tak servis živi v procesu, kateremu pravimo gostiteljski proces (ang. Host process). Kot smo že povedali, servisi WCF temeljijo na servisno-orientirani arhitekturi, zato je robustnost servisa zelo pomembna, kar je močno odvisno od tega, kje in kako naš servis gostuje.

Pri izbiri načina gostovanja, se moramo vprašati:

- kakšne so zahteve pri načinu dostopa do servisa,
- kako naj bo servis dosegljiv in kako se aktivira,

- kakšen naj bo način postavitve v okolje,
- ali je zahtevana podpora starejšim verzijam servisov,
- kakšno podporo varnostnim mehanizmom želimo?

Poznamo štiri glavne načine gostovanja servisov WCF.

- Samo gostovanje (ang. self hosting) – za razliko od klasičnih spletnih servisov, lahko servis WCF gostuje v katerikoli aplikaciji Windows, ki podpira izvajanje kode .NET. Pri takem načinu mora razvijalec sam skrbeti nad življenjskim ciklom gostiteljskega procesa.
- Windows servis – servis WCF lahko gostujemo tudi v navadnem servisu Windows (ang. Windows service). Pri tem primeru se servis WCF avtomatsko zažene. Življenje procesa lahko kontroliramo z nadzornim upravljavcem (ang. Service Control Manager).
- gostovanje IIS - gostiteljski proces se avtomatično zažene, ko pride do prve odjemalčeve zahteve. Temu pravimo aktivacija na zahtevo (on-demand activation) [12]. Pri tem se koristijo vse funkcije IIS-ja kot so recikliranje procesov, zaustavitev ob nedejavnosti, nadzor nad stanjem procesa. Glavna slabost IIS-ja je, da podpira samo protokol HTTP.
- WAS - Windows aktivacijski servis je sistem, ki je na voljo pri operacijskih sistemih Windows Vista, Windows Server 2008 in novejših. Na voljo je z verzijo IIS 7.0. Za razliko od IIS 6.0 in starejših, IIS 7.0 podpira poleg protokola HTTP tudi protokol TCP, imenski cevovod,.. Ta način gostovanja je najbolj primeren, saj omogoča izkoriščanje vseh prednosti servisov WCF.

V našem primeru smo se odločili za IIS gostovanje na domačem strežniku, z nameščenim Windows 7 operacijskim sistemom in IIS-jem 7.5. Pri tem načinu gostovanja nam ni treba skrbeti za življenjski cikel procesa. Ta se aktivira med prvo prihajajočo zahtevo. Servis WCF je dosegljiv na URL-ju:

`http://ankete.sytes.net/Ankete/anketeservice.svc`

V datoteki `anketeservice.svc` so podatki, kako je naš servis WCF povezan z arhitekturo IIS in našo aplikacijo Ankete. Lahko bi rekli, da je v tej datoteki referenca iz aplikacije na WCF servis. Ko pride uporabnikova zahteva na končno točko servisa, tehnologija Asp.NET zahtevo prestreže in izdela `serviceHost` proces, kateri gosti razred `AnketaService`. Dodati moramo še referenco na zbirko servisa v bin mapo naše spletne aplikacije. Dodana mora biti tudi referenca na knjižnico `System.ServiceModel.dll`, katero dobimo z namestitvijo ogrodja .NET 3.0 in novejšega. Ta knjižnica zajema večino funkcij, ki jih servisi WCF potrebujejo.

Vsebina `anketeservice.svc` datoteke zgleda takole:

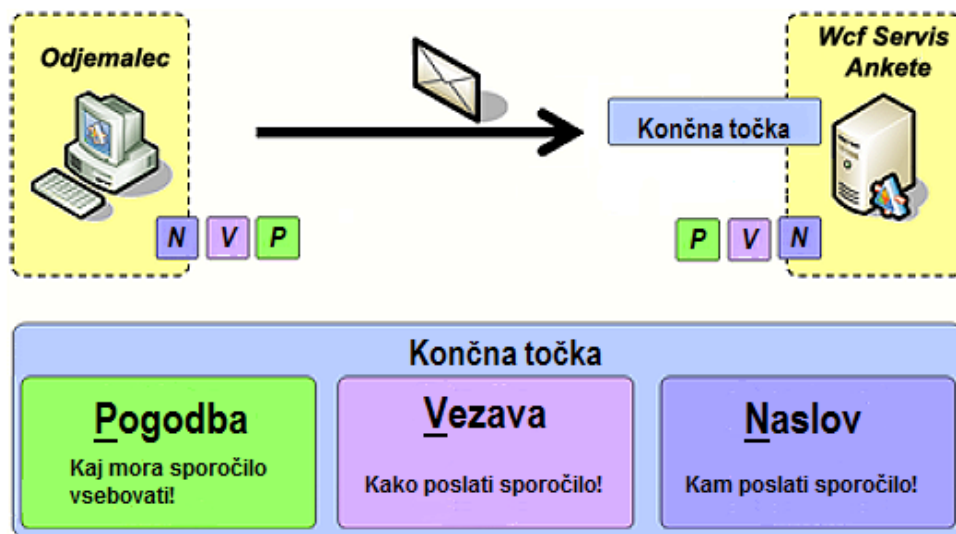
```
<%@ ServiceHost Service="AnketaService.AnketaService" %>
```

Vsaka spletna aplikacija ima `web.config` konfiguracijsko datoteko. Kadar servise WCF gostujemo v strežniku IIS, jih konfiguriramo v tej datoteki. V njej določimo končne točke, kjer je servis dosegljiv, obnašanje servisa ter druge nastavitvene parametre.

## 5.4 Končna točka

Servis WCF je program, ki ponuja množico končnih točk (ang. *endpoint*). Vsaka končna točka je vmesnik, preko katerega storitve servisa komunicirajo z zunanjim svetom. Vsaka storitev mora imeti vsaj eno končno točko, lahko jih ima tudi več. Z njimi lahko dosežemo uporabo različnih načinov komunikacije za različne odjemalce. Bistvo pri tem je, da je poslovna logika, ki jo aplikacija uporablja, pri vseh končnih točkah enaka.

Končno točko sestavljajo tri komponente Slika 5.2. Naslov, vezava in pogodba. Velikokrat jih poimenujemo ABC komponente (ang. *Address, Binding, Contract*). Končne točke lahko v Visual Studiu dodamo programsko ali pa ročno v konfiguracijski datoteki.



Slika 5.2: Končna točka

Spodnji primer prikazuje, kako je končna točka definirana v konfiguracijski datoteki web.config.

```
<endpoint
  address="http://ankete.sytes.net/Ankete/AnketeService.svc"
  binding="wsHttpBinding
  contract="AnketeService.IAnketeService">
</endpoint>
```

- **Naslov** (ang. Address) Naslov je v bistvu naslov URL, ki določa, kje gostuje servis WCF. Odjemalec bo ta naslov uporabljal za povezavo s servisom. V tem naslovu je že podan transportni protokol HTTP in naslov DNS. Vsem podatkom v naslovu rečemo tudi transportna shema. Primer:
- **Vezava** (ang. Binding) Vezava opisuje na kakšen način in preko katerega protokola odjemalci komunicirajo s servisom. Pri tehnologiji WCF imamo na voljo več različnih protokolov. To nam omogoča, da si izberemo najprimernejšega.

Vezava ima naslednje karakteristike, vključno z naslednjimi.

1. Prenos (ang. Transport) - Definira osnoven transportni protokol, ki se uporablja. Nekateri izmed njih so (HTTP , TCP, MSMQ, Named Pipes).
2. Kodiranje (ang. Encoding) – Na voljo so trije tipi. To je tekstovni, binarni in optimiziran mehanizem za prenos sporočil MTOM (ang. Message Transmission Optimization Mechanism).
3. Protokol (ang. Protocol) – Vsebuje podatke o varnosti in varnem prenosu podatkov.

Nekaj izmed najpogosteje uporabljenih vezav:

1. *wsHttpBinding* - Zanesljiva, varna, interoperabilna vezava, ki podpira html protokol.
  2. *BasicHttpBinding* – Osnovna komunikacija med spletnimi servisi. Brez prevzete varnosti.
  3. *WsDualHttpBinding* – Dvosmerna komunikacija.
  4. *NetNamedPipeBinding* – Komunikacija med aplikacijami WCF na istem računalniku.
  5. *NetTcpBinding* – Komunikacija med računalniki s podporo dvo-smerni komunikaciji.
- Pogodba (ang. Contract)

Pri tehnologiji WCF vsak servis ponuja tako imenovane pogodbe. Pogodba je način za opis delovanja servisa. WCF določa 4 tipe pogodb.

1. Servisna pogodba (ang. service contract) – opisuje, katere operacije ponuja servis na voljo odjemalcu.
2. Podatkovna pogodba (ang. data contract) – definira, katere podatkovni tipe si odjemalec in servis izmenjujeta

3. Sporočilna pogodba (ang. message contract) – dopušča servisu, da komunicira direktno s sporočili. Ta tip pogodbe uporabljamo v primeru, da želimo imeti več kontrole nad sporočili SOAP.
4. Pogodba o napakah (ang. fault contract) – definira, katere napake lahko servis sproži in kako potem te napake obvladuje in poroča odjemalcu o njih.

## 5.5 Izmenjava metapodatkov (ang. Metadata Exchange)

Ker je servis WCF za odjemalce, ki dostopajo do njega precejšnja neznanka, ponavadi ponuja metapodatke. Dostop do metapodatkov moramo eksplicitno omogočiti. Objavljeni so v tehnološko nevtralni tehnologiji WSDL (ang. Windows Service Description Language). Ti podatki opisujejo funkcionalnost in komunikacije s servisom, ki so na voljo. S pomočjo teh podatkov lahko odjemalec dostopa do razredov CLR na servisu. Metapodatke omogočimo s tem, ko na servisu v konfiguracijski datoteki omogočimo atribut *serviceMetadata*.

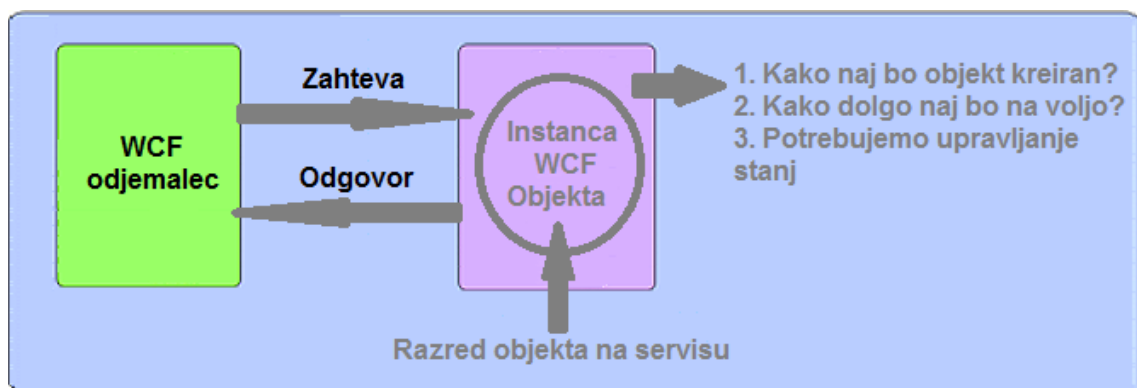
```
<serviceBehaviors>
  <behavior name="Metadata">
    <serviceMetadata httpGetEnabled="true"/>
  </behavior>
</serviceBehaviors>
```

## 5.6 Upravljanje instanc pri tehnologiji WCF

Ko uporabljamo spletne servise, nas velikokrat zanima, kako je objekt, do katerega dostopamo, kreiran in koliko časa bo na servisu na voljo.

Zahteve in odgovori pri komunikaciji WCF sprožijo naslednje zaporedje akcij:

1. odjemalec sproži zahtevo po objektu na servisu WCF,
2. instanca objekta WCF se kreira,
3. instanca WCF objekta ustreže zahtevi in pošlje odgovor WCF odjemalcu.



Slika 5.3: Tok kreiranja instanc

Kako se servis WCF instancira določimo z atributom [ServiceBehavior] in parametrom InstanceContextMode, kateri zajema tri načine:

- **za vsak klic** (ang. per call). Nova instanca servisa se ustvari za vsak odjemalčev klic storitve in se po končani rabi uniči,
- **za vsako sejo** (ang. per session). Vsakemu odjemalcu se dodeli svoja instanca servisa. Zaradi potrebe po obstoju stanja med klici metod, je instanca servisa na voljo, kolikor traja določena seja,
- **ena instanca za vse klice** (ang. single instance mode). Pri tem načinu ena instanca servisa WCF ravna z vsemi klici. Pri tem morajo vsi odjemalčevi klici čakati, da se predhodni klici izvršijo do konca.

## 5.7 Varnost

Pri vseh porazdeljenih sistemih moramo zagotoviti določeno stopnjo varnosti. Predvsem je pomembna medsebojna overitev pristnosti odjemalca in strežnika. V kolikšni meri bomo komunikacijo zavarovali, je odvisno tudi od okolja in problemskega področja, ki ga naša storitev rešuje.

Da to dosežemo, WCF zagotavlja vse pomembne funkcije za:

- **overjanje.** Z overjanjem odjemalca in strežnika preprečimo, da se tretja oseba vmeša v komunikacijo,
- **integriteto sporočil.** Pri tem načinu so sporočila digitalno podpisana, kar zagotovi da nismo bila po poti spremenjena,
- **zaupnost sporočil.** Sporočila morajo biti šifrirana,
- **pooblastitev** (ang. authorization). S tem določimo kakšne funkcije se lahko nad sporočilom izvajajo.

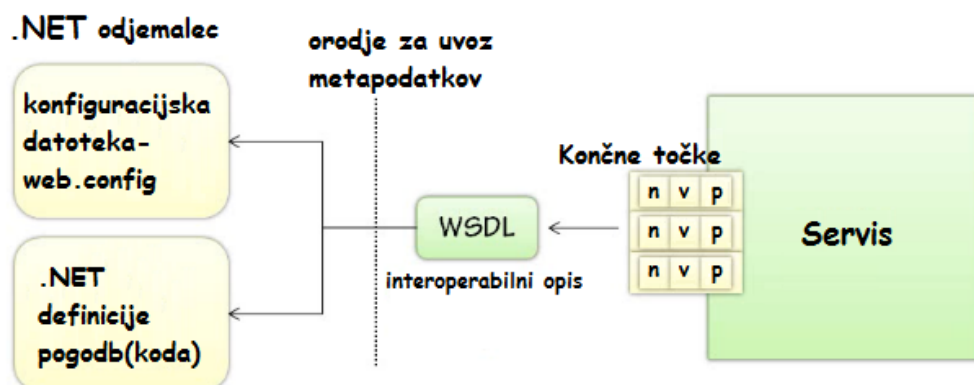
Vse te funkcije določimo z izbiro načina vezave (ang. binding). Ko na servisu ustvarimo končno točko, izberemo način vezave, ki določa ustrezen komunikacijski protokol in kako bodo sporočila šifrirana.

## 5.8 Razvoj odjemalca

Za prikaz delovanja in uporabe servisa WCF smo izdelali preprostega odjemalca. Odjemalci so v našem primeru spletne strani. Razvijalci le teh želijo imeti spletne ankete, vključene na njih. V diplomskem delu smo se omejili na odjemalce na platformi Microsoft, z nameščenim ogrodjem Microsoft .NET 3.0 ali novejšim. Servise WCF lahko uporabljamo oziroma izkoriščamo tudi z drugimi jeziki, ki temeljijo na drugih platformah, saj tehnologija WCF temelji na izmenjavi sporočil in standardu SOAP.

Prvo se posvetimo arhitekturi WCF na odjemalčevi strani. Tako kot servis WCF, tudi odjemalec potrebuje končne točke, da lahko komunikacija

steče. Te se avtomatično zgenerirajo, ko na odjemalcu dodamo referenco na servis WCF. V Visual Studiu 2008 dodamo referenco projektu s pomočjo menijske funkcije "dodaj servisno referenco" (ang. Add Service Reference). Podati moramo URL naslov, kjer servis gostuje. V našem primeru je naslov: `http://ankete.sytes.net/Ankete/anketeservice.svc`



Slika 5.4: Pridobivanje definicij in pogodb

S tem se zgradi *programski kanal* oziroma *posredniški razred* (ang. proxy class), ki odjemalcu zagotavlja komunikacijo s strežniškimi metodami. Poglejmo primer, kako na odjemalcu definiramo kanal, kateri zagotavlja metode, ki so definirane na servisu WCF. Vidimo primer klica metode `getAnketaHtml`, z parametrom `idAnkete`. Ta parameter predstavlja anketin enolični identifikator. Po klicu metode se iz servisa na odjemalčevo stran prenese HTML koda v tekstovni obliki. S pomočjo funkcije `.InnerHtml` vpišemo HTML kodo v prazen div okvir, tabelo, ...

Na tak način pridemo do naše spletne ankete.

```
public partial class Anketa : System.Web.UI.Page{
    AnketeServiceClient client =
        new AnketeServiceClient("WSHttpBinding_AnketeService");

    int idAnkete = 1072; //Id ankete ki jo želimo na strani.
```

```
protected void Page_Load(object sender, EventArgs e)
{
    this.btnSubmit.Click += new EventHandler(btnSubmit_Click);
    string anketaHtml =
        client.getAnketaHtml(Convert.ToInt32(idAnkete));
    this.anketaHtmlDiv.InnerHtml = Convert.ToString(anketaHtml);
}
protected void btnSubmit_Click(object sender, EventArgs e){
    string rezultatiStopenjskoString = hdnStopenjsko.Value;
    string rezultatiMoznostnoString = hdnVecMozn.Value;
    string rezultatiDaNeString = hdnDaNe.Value;
    string rezultatiMatricnoString = hdnMatricno.Value;
    string rezultatiTekstovnoString = hdnText.Value;

    client.saveAnketa(idAnkete, rezultatiStopenjskoString,
        rezultatiMoznostnoString, rezultatiDaNeString,
        rezultatiMatricnoString, rezultatiTekstovnoString);
}
}
```

Oglejmo si še primer kode, ki se avtomatično doda v nastavitveno datoteko web.config. Povečali smo časovne omejitve (ang. Timeout) in velikosti spremenljivk, ker so nam prednastavljeni delali določene probleme. Na tak način smo dosegli precej robustno in zanesljivo delovanje.

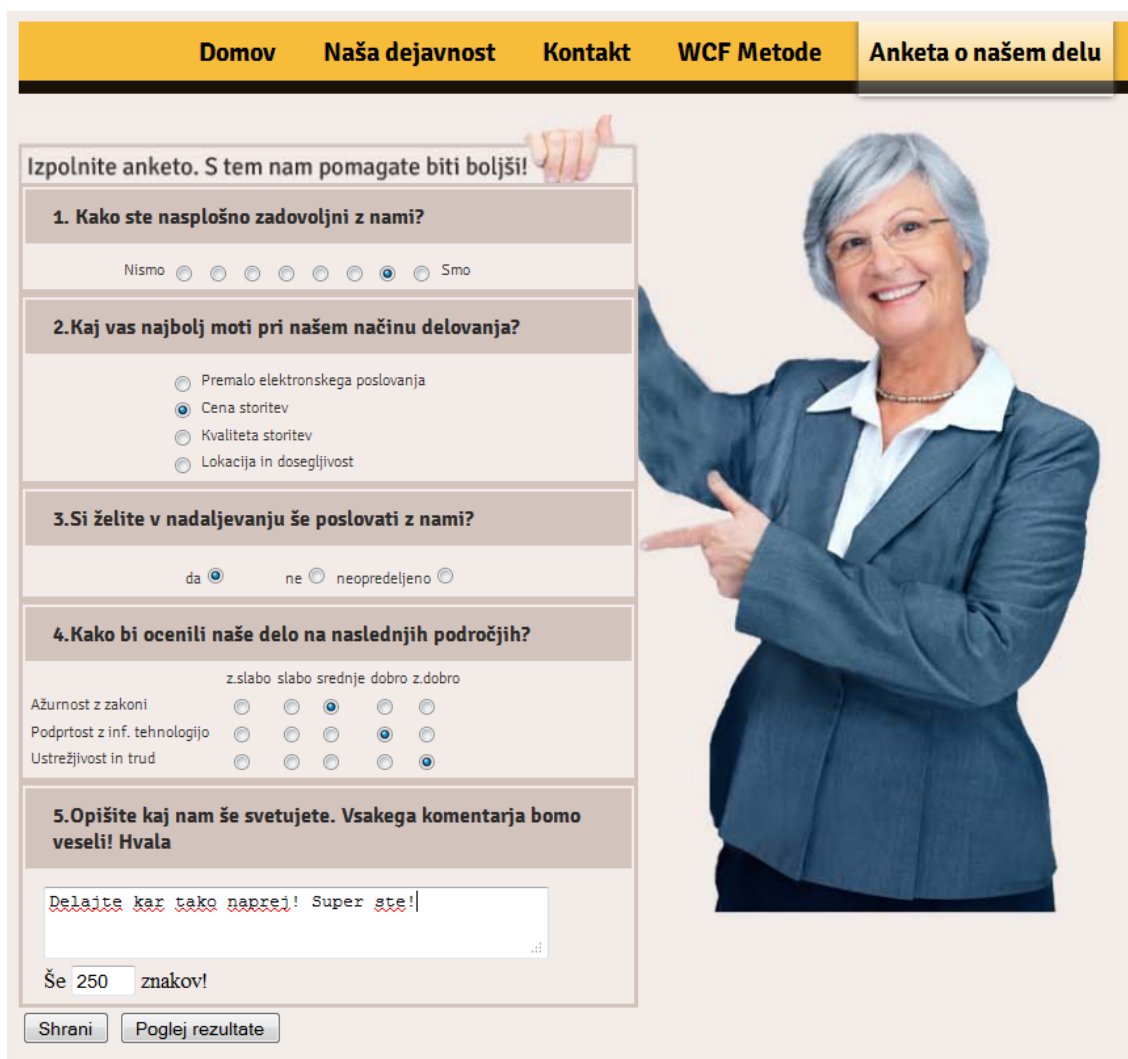
```
<system.serviceModel>
  <bindings>
    <wsHttpBinding>
      <binding name="WSHttpBinding_AnketeService" closeTimeout=
        "00:10:00" openTimeout="00:10:00" receiveTimeout="00:10:00"
        sendTimeout="00:10:00" bypassProxyOnLocal="false">
```

```
transactionFlow="false" hostNameComparisonMode="StrongWildcard"
maxBufferPoolSize="2147483647" maxReceivedMessageSize="2147483647"
messageEncoding="Text" textEncoding="utf-8"
useDefaultWebProxy="true" allowCookies="false">
  <readerQuotas maxDepth="32333" maxStringContentLength="2147483647"
    maxArrayLength="2147483647" maxBytesPerRead="2147483647"
    maxNameTableCharCount="2147483647" />
  <reliableSession ordered="true" inactivityTimeout="00:10:00"
    enabled="true" />
  <security mode="None">
    <transport clientCredentialType="Windows"
      proxyCredentialType="None" realm="" />
    <message clientCredentialType="Windows"
      negotiateServiceCredential="true"
      establishSecurityContext="true" />
  </security>
</binding>
</wsHttpBinding>
</bindings>
<client>
  <endpoint
    address="http://ankete.sytes.net/Ankete/AnketeService.svc"
    binding="wsHttpBinding"
    bindingConfiguration="WSHttpBinding_AnketeService"
    contract="AnketaReference.AnketeService"
    name="WSHttpBinding_AnketeService">
    <identity> <dns value="localhost" /> </identity>
  </endpoint>
</client>
</system.serviceModel>
```

Dotaknimo se še same zasnove in videza posamezne ankete. Vprašanja zgeneriramo in dodamo dinamično, eno za drugim. Na tak način gradimo anketo, ki je sestavljena iz HTML gradnikov. Vsak gradnik smo opremili s CSS stilskimi razredi (ang. Code Style Sheet). Primer HTML kode določene ankete:

```
<table class='tbl_vpr_html_izpis'>
  <tr class='table_row_glava'>
    <th class='vprasanje_glava' colspan='2'>
      Ste zadovoljni z našo ponudbo? </th>
    </tr>
  <tr class='table_row_body'>
    <td class='left_vprasanje_body'>
      <table class='vpr_body_table'>
        <tr class='vpr_body_table_row'>
          <td class='vpr_body_table_col_DaNe'>
            <label class='pisavaAnketa'>da </label>
            <input name='vprDaNe3' id='rbda3' value='1'
              type='radio' class='rbDaNe' /> </td>
          <td class='vpr_body_table_col_DaNe'>
            <label class='pisavaAnketa'>ne </label>
            <input name='vprDaNe3' id='rbne3' value='2'
              type='radio' class='rbDaNe' /></td>
          <td class='vpr_body_table_col_DaNe'>
            <label class='pisavaAnketa'> neopredeljeno </label>
            <input name='vprDaNe3' id='rbneopredeljeno3' value='3'
              type='radio' class='rbDaNe' /> </td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```

Na tak način lahko razvijalec anketo stilsko oblikuje po svojem okusu ter poskrbi, da grafična podoba ankete sovпада z grafično podobo svoje spletne strani Slika 5.5.



**Domov** **Naša dejavnost** **Kontakt** **WCF Metode** **Anketa o našem delu**

Izpolnite anketo. S tem nam pomagajte biti boljši!

**1. Kako ste nasplošno zadovoljni z nami?**

Nismo        Smo

**2. Kaj vas najbolj moti pri našem načinu delovanja?**

Premalo elektronskega poslovanja  
 Cena storitev  
 Kvaliteta storitev  
 Lokacija in dosegljivost

**3. Si želite v nadaljevanju še poslovati z nami?**

da  ne  neopredeljeno

**4. Kako bi ocenili naše delo na naslednjih področjih?**

z.slabo slabo srednje dobro z.dobro

Ažurnost z zakoni

Podprtost z inf. tehnologijo

Ustrežljivost in trud

**5. Opišite kaj nam še svetujete. Vsakega komentarja bomo veselili! Hvala**

Še 250 znakov!

Slika 5.5: Primer umestitve in izkoriščanja ankete na strani odjemalca



# Poglavje 6

## Zaključek

Pri razvoju spletne aplikacije *Ankete* nam je največ težav povzročalo področje implementacije servisa WCF. Ker tehnologija WCF zajema številna tehnološka področja pri spletnem programiranju, je spoznavanje in učenje rahlo zamudno. Istočasno tehnologija WCF uvaja enostavnejše metode implementacije in konfiguracije posameznih podrobnosti. Na ta način lahko damo večji poudarek dejanskemu problemu, ki ga rešujemo, saj ko se enkrat naučimo upravljati s tem orodjem in izkoriščati prednosti, ki jih ponuja, je izdelava varnih odjemalec-strežnik aplikacij precej poenostavljena.

Med ustvarjanjem aplikacije so se nam porodile še dodatne ideje, ki bi jih želeli ob predpostavki, da bi bil za to interes, v prihodnosti razviti. Lahko bi ponudilo širšo paleto možnih vprašanj. Recimo ocenjevanje slik, videa in drugih multimedijskih vsebin. Lahko bi implementirali boljše analiziranje rezultatov, izvoz rezultatov v druge formate (pdf, excel, ..)



# Literatura

- [1] Michele Leroux Bustamante, *Learning WCF*, str. 14, O'Reilly Media, maj 2007
- [2] Marko Bajec, Marjan Krisper. Agilne metodologije razvoja informacijskih sistemov. *Uporabna informatika*, št. 2 (2003), str. 68-76
- [3] KRUCHTEN, P. *The Rational Unified Process – An Introduction*, 2nd ed., Addison-Wesley-Longman, Reading, MA, 2000.
- [4] Razvoj informacijskih sistemov, Visoko strokovni študij, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Doc. dr. Marko Bajec, stran 199, leto 2007
- [5] Anketna metodologija - Opredelitev,  
Dostopno na: [http://www.1ka.si/c/693/Opredelitev/?preid=773?](http://www.1ka.si/c/693/Opredelitev/?preid=773) ,  
maj 2012
- [6] Tehnologija ogrodje .NET,  
Dostopno na: <http://www.c-sharpcorner.com/UploadFile/chandrakantpp/UnderstandingFrameworkatglance11292005013851AM/UnderstandingFrameworkatglance.aspx> , junij 2012
- [7] Izvajalnik kode CLR,  
Dostopno na: <http://dotnetclassnotes.blogspot.com/2010/12/2312-net-framework-architecture.html>, maj 2012

- [8] .NET Framework Class Library, Mark Sapposnek, CS 594,  
Computer Science Department, Boston University, maj 2012
- [9] Interoperability,  
Dostopno na: <http://en.wikipedia.org/wiki/Interoperability#Software> ,  
april 2012
- [10] What's New in Visual Studio 2008,Product Guide7,  
Dostopno na: <http://download.microsoft.com/download/3/B/D/3BDDC5D5-E8CF-4031-A12A-B072ECA27C42/VS2008ProductGuideEMEA.doc>,  
junij 2012
- [11] JQuery,  
Dostopno na: <http://jquery.com/> , maj 2012
- [12] IIS - Internet Information Services,  
Dostopno na: [http://en.wikipedia.org/wiki/Internet\\_Information\\_Services](http://en.wikipedia.org/wiki/Internet_Information_Services),  
maj 2012
- [13] ASP.NET - Tehnologija ASP.NET  
Dostopno na: <http://msdn.microsoft.com/en-us/library/dd566231>, maj  
2012