

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleš Perkovič

**Spletna aplikacija za pripravo in
izvedbo longitudinalnih anket
prilagojenih skupinam anketirancev**

DIPLOMSKO DELO
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Damjan Vavpotič

Ljubljana 2013

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

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.



Št. naloge: 00369/2013

Datum: 01.03.2013

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

Kandidat: **ALEŠ PERKOVIČ**

Naslov: **SPLETNA APLIKACIJA ZA PRIPRAVO IN IZVEDBO
LONGITUDINALNIH ANKET PRILAGOJENIH SKUPINAM
ANKETIRANCEV**

**WEB APPLICATION FOR PREPARATION AND IMPLEMENTATION OF
LONGITUDINAL SURVEYS TAILED TO GROUPS OF
RESPONDENTS**

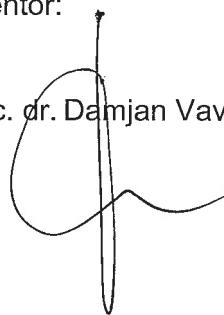
Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V okviru diplomskega dela izdelajte spletno aplikacijo, ki bo namenjena podpori izvedbe anket. Aplikacija mora podpreti ponavljajoče izvajanje anket ob prednastavljenih časovnih terminih, anketiranje po delih in prilaganje skupinam anketirancev. Anketiranje po delih pomeni, da posamezne skupine anketirancev ocenjujejo le vnaprej določene dele objekta ocenjevanja. Aplikacija mora omogočati pripravo poljubnih vprašalnikov, ki vključujejo tako odprte kot zaprte tipe vprašanj. Del aplikacije namenjen izvedbi anket naj omogoča ocenjevanje tako neprijavljenim (anonimno) kot prijavljenim anketirancem. V teoretičnem delu naloge predstavite ključne tehnologije, ki ste jih uporabili pri izdelavi aplikacije, okvirno zgradbo aplikacije in njene ključne funkcije.

Mentor:

doc. dr. Damjan Vavpotič



Dekan:

prof. dr. Nikolaj Žimic



IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Aleš Perkovič, z vpisno številko **63060323**, sem avtor diplomskega dela z naslovom:

Spletna aplikacija za pripravo in izvedbo longitudinalnih anket prilagojenih skupinam anketirancev

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Damjana Vavpotiča,
- 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 3. junija 2013

Podpis avtorja:

Zahvala

Za vso strokovno pomoč in nasvete pri izdelavi se zahvaljujem mentorju doc. dr. Damjanu Vavpotiču. Prav tako se zahvaljujem vsem, ki so kakorkoli pripomogli k nastanku te naloge. Še posebej bi se rad zahvalil Urši za potrežljivost, moji Emi in seveda staršem, ki so mi študij omogočili.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Vrste anket in anketiranje	3
2.1	Vrste anket	4
2.1.1	Osebno anketiranje ali intervju	4
2.1.2	Poštno anketiranje	4
2.1.3	Telefonsko anketiranje	4
2.1.4	Spletno anketiranje	5
2.2	Vrste vprašanj	6
3	Uporabljene tehnologije	7
3.1	Ogrodje .NET	7
3.2	ASP.NET	8
3.3	Visual Studio 2010	9
3.4	PowerDesigner	9
3.5	HTML	9
3.6	CSS	10
3.7	Javascript	10
4	Planiranje, analiza, načrtovanje in razvoj aplikacije	11
4.1	Strukturni pristop k razvoju aplikacije	11

KAZALO

4.2	Namen aplikacije	12
4.3	Funkcionalne in nefunkcionalne zahteve	13
4.3.1	Funkcionalne zahteve	13
4.3.2	Nefunkcionalne zahteve	14
4.4	Diagram primerov uporabe	15
4.5	Diagram funkcionalne razgradnje	16
4.6	Podatkovni model	17
4.7	Opis podatkovne baze	18
4.8	3-nivojska arhitektura	20
4.9	Razvoj aplikacije	21
4.9.1	Povezava s podatkovno bazo	22
4.9.2	Uporabniški vmesniki aplikacije	23
4.10	Testiranje	25
4.11	Opis delovanja končne aplikacije	26
4.11.1	Prijava v sistem	26
4.11.2	Prijava kot administrator	28
	Urejanje obstoječih anket	28
	Pregled in urejanje uporabnikov	32
	Izdelava ankete	33
	Dodajanje ali izbris elementa vrednotenja	36
	Izvoz podatkov	37
4.11.3	Prijava kot uporabnik	37
	Urejanje računa	38
	Pregled anket	38
4.11.4	Prijava kot gost	39
4.11.5	Reševanje ankete	39
5	Sklepne ugotovitve	41

Seznam uporabljenih kratic

CSS – (angl. Cascading Style Sheets) stilski jezik za urejanje videza html dokumenta

SQL – (angl. Structured Query Language) strukturni jezik, namenjen pisanju poizvedb

.NET – Microsoftovo ogrodje za razvoj programske opreme

HTML – (angl. HyperText Markup Language) označevalni jezik, namenjen razvoju spletnih strani

C# – objektni programski jezik, ki je na voljo v okolju .NET

CSV – (angl. Comma Separated Values) vrednosti, ločene z vejico

Povzetek

Namen diplomskega dela je izdelava spletnne aplikacije za prilagojeno anketiranje glede na različne tipe anketirancev, ki omogoča možnost longitudinalnega anketiranja. Aplikacijo je možno vključiti v spletno stran, kjer jo lahko uporabljajo gostje in interni uporabniki (npr. zaposleni v podjetju). Uporabnost longitudinalnega anketiranja se kaže pri proučevanju dolgoročno pridobljenih podatkov, ki jih dobimo z izvajanjem enakih anket na istih tipih anketirancev v več časovnih intervalih.

V delu so predstavljeni tehnologije in orodja, uporabljena pri izdelavi aplikacije, ki je bila razvita v ogrodju .NET s programskim jezikom C#. Prikazana sta analiza in načrtovanje aplikacije, opisani so njen namen, funkcionalne in nefunkcionalne zahteve ter podatkovni model aplikacije. Jedro naloge predstavlja opis razvoja aplikacije, ki vključuje opis podatkovne baze in razvoja sestavnih delov ter natančen opis delovanja končne aplikacije.

Ključne besede: spletnne aplikacije, ogrodje .NET, C#, razvoj, anketiranje

Abstract

The purpose of the thesis is creation of a web application for customized survey regarding different types of respondents, which enables longitudinal research. The application can be integrated in any website (e.g. of a company), where it can be used by guests (as well as employees). The aspect of longitudinal research is reflected in data obtained in a long-term study, which we get by conducting the same survey on the same types of respondents in several intervals.

The technologies and tools that I used to create the application are described in the thesis. The application was developed in .NET framework with C# programming language. Analysis and planing of the application are presented which includes the purpose, functional and non-functional requirements as well as data model of the application. The main goal of the thesis is the description of developed application that includes database plan, development of the components and the exact description how the finished application operates.

Keywords: web application, .NET framework, C#, development, survey

Poglavlje 1

Uvod

V svetu računalništva smo priča neprestanemu razvoju. Z veliko hitrostjo se razvijajo svetovni splet in vse tehnologije, ki ga podpirajo in sestavljajo. Splet je postal nepogrešljiv del vsakdanjega življenja in tudi najhitrejši način za dostopanje do najrazličnejših podatkov. Prav tako se splet vse bolj uporablja v tržne in oglaševalne namene, saj skoraj ne najdemo uspešnega podjetja, ki ne bi imelo svoje spletnih strani. Na tak način se lahko podjetje predstavi svojim morebitnim strankam, hkrati pa želi od njih pridobiti čim več povratnih informacij, kar lahko naredi s pomočjo spletnih anket. Te so danes postale sestavni del mnogih spletnih strani, saj lahko na primer podjetje (ali pa kakšen drugi anketar) na relativno enostaven in hiter način pride do podatkov večje skupine ljudi.

Zato sem v okviru diplomskega dela pripravil aplikacijo za anketiranje, ki se bo lahko dodala v neko obstoječo spletno stran. Njena funkcionalnost bo predvsem osredotočena na longitudinalno anketiranje, torej se bo lahko neka anketa izvajala v več ciklih, ki si bodo sledili v določenem časovnem intervalu. Tako bo lahko anketar (naročnik aplikacije) prišel do dolgoročnih ugotovitev. Praksa anketiranja je pokazala, da je anketirance dobro med seboj ločiti, še posebej pri longitudinalnem anketiranju, saj od različnih tipov anketirancev po navadi želimo pridobiti različne informacije. Aplikacija bo

zato omogočala anketiranje uporabnikov z anketami, ki bodo prilagojene za določen tip anketiranca.

Poglavlje 2

Vrste anket in anketiranje

Anketa je najbolj poznan in največkrat uporabljen postopek zbiranja podatkov in mnenj. Pomaga nam, da na enostaven način in relativno hitro pridemo do želenih podatkov in rezultatov v ciljni skupini anketirancev. [7]

Najstarejša oblika anketnega zbiranja informacij je terensko anketiranje, kjer anketar po navadi na domu obišče anketiranca in dobljene odgovore beleži. Zaradi uspešnosti takšnega načina anketiranja se je začelo iskati cenejše načine, kako priti do enakih rezultatov. Za cenejše se je izkazalo samoanketiranje, sprva predvsem z anketami po pošti, a te niso prinesle želenih rezultatov. Tehnološki razvoj, predvsem dostopnost telefonov, je omogočil, da se je začelo vse bolj uveljavljati telefonsko anketiranje. Tovrstno anketiranje je še dodatno poenostavila tehnologija računalniško podprtga telefonskega anketiranja. Poleg računalniško podprtga telefonskega anketiranja poznamo računalniško podprto osebno anketiranje in računalniško podprto samoanketiranje. Razvoj interneta, tehnologij za prenos podatkov in brskalnikov ter združevanje različnih tehnologij sta pripeljala do razvoja anketiranja po internetu, sprva z anketiranjem po elektronski pošti, kasneje pa so nastale prve spletne ankete. Te so bile najprej statične, a je njihov razvoj šel v smer interaktivnih spletnih anket. [9]

2.1 Vrste anket

Razvoj anketiranja je skozi zgodovino potekal vzporedno z razvojem tehnologije, zato ločimo več tipov anket glede na način anketiranja.

2.1.1 Osebno anketiranje ali intervju

Osebno anketiranje je najstarejši način zbiranja podatkov, kjer anketar neposredno izprašuje anketiranca in odgovore beleži. Če se vprašalnik izvaja na prenosnem računalniku z neposrednim vnašanjem odgovorov, temu rečemo računalniško podprt osebno anketiranje. Osebno anketiranje omogoča daljše in kompleksnejše anketiranje, odgovori pa so zaradi možnosti obrazložitve vprašanja veliko bolj natančni in popolni. Vendar način izvajanja anketiranja na terenu prinese velike stroške in zahteva veliko časa.

2.1.2 Poštno anketiranje

Pri tej tehniki anketiranja gre pravzaprav za samoanketiranje, kjer anketiranc prejme anketo po navadni pošti. Po navadi je anketi priložena izpolnjena ovojnica s plačano poštnino, tako da anketiranec nima nobenih stroškov. Anketiranec lahko anketo prejme tudi v elektronski obliki preko elektronske pošte in jo v elektronski obliki odpošlje nazaj anketarju. S tem načinom anketiranja ni toliko stroškov kot z drugimi načini, njegova slabost pa je predvsem majhen odstotek realiziranih anket. Ker anketar ni prisoten, se lahko zgodi, da anketiranec navodil ne razume ali jih razume napačno.

2.1.3 Telefonsko anketiranje

Zaradi več razlogov je to najpogostejši način anketiranja. Od osebnega anketiranja se razlikuje le v tem, da anketar po telefonu pokliče anketiranca. Pri tem tipu anketiranja je računalniška podpora prisotna dlje časa kot pri osebnem anketiranju. Računalniško podprt telefonsko anketiranje omogoča enostavnnejše shranjevanje odgovorov in tudi optimizacijo telefonskih klicev.

Pri tem je pomembno, da so anketarji ustrezeno usposobljeni za tovrstno anketiranje, zato je strošek nekoliko večji, a to vsekakor pripomore k večji količini pridobljenih podatkov. [8]

2.1.4 Spletne ankete

Spletne ankete so novejše področje anketne metodologije, ki je neposredno odvisna od računalniške tehnologije. Najpogosteje uporabljeni metodi za pridobivanje podatkov preko interneta sta anketiranje z vprašalniki, ki jih pošljemo po elektronski pošti, in spletna anketa. Pri slednji anketiranec dostopa do anketnega vprašalnika preko standardiziranih spletnih brskalnikov in odgovarja na vprašanja brez pomoči anketarja (samoanketa).

Prve spletne ankete so bile statične, toda razvoj različnih spletnih tehnologij je omogočil, da so postale vedno bolj interaktivne. Danes se s tovrstnimi anketami srečujemo na različnih spletnih straneh, kjer želijo ponudniki različnih storitev izvedeti čim več o morebitnih uporabnikih ali kupcih, torej o obiskovalcih svojih spletnih strani. Tako lahko spletne ankete glede na njihovo izvajanje razdelimo v tri skupine: [9]

- spletne ankete s splošnimi vabili, kjer seznam anketirancev ne obstaja;
- spletne ankete z individualnimi vabili, kjer seznam anketirancev obstaja;
- prestrezne ankete, kjer se seznam generira med samim anketiranjem.

Prednosti spletne ankete so predvsem:

- precej nižji stroški, predvsem v primerjavi z osebnim anketiranjem in anketiranjem po telefonu;
- zbiranje podatkov je kvalitetnejše in zanesljivejše;
- v kratkem času pridemo do velike količine podatkov;

- več je iskrenih odgovorov zaradi anonimnosti, predvsem pri občutljivih temah.

Kot vse ankete ima tudi spletno anketiranje svoje slabosti, na primer nemotiviranost anketirancev za izpolnitev vprašalnika v celoti. Kot pri vsakem samoanketiranju tudi pri tem ne moremo biti povsem prepričani, da bo anketiranec popolnoma razumel vsa vprašanja, zato morajo biti ta napisana čim bolj razumljivo.

2.2 Vrste vprašanj

Anketna vprašanja lahko razdelimo na tri skupine: odprta, zaprta in kombinirana vprašanja. [7] Na odprta vprašanja anketiranec odgovarja s svojimi besedami, zato so lahko ti odgovori bolj individualni ter podrobnejši in posledično zahtevnejši za obdelavo, poleg tega pa je večja možnost trivialnih odgovorov. Če k vprašanju ponudimo možne odgovore, je to zaprto vprašanje. Takšne odgovore je lažje obdelovati in so primernejši za večino anketirancev. Za tovrstna vprašanja lahko uporabimo več tipov vprašanj, kot so:

- izbirno vprašanje z eno izbiro, kjer anketiranec izbere le en možni odgovor, ki mu najbolj odgovarja;
- izbirno vprašanje z več izbirami, kjer lahko anketiranec izbere več možnih odgovorov;
- lestvično vprašanje, kjer anketiranec izbere, kako zelo se z neko trditvijo strinja (po navadi je razpon lestvice od 1 do 5 ali od 1 do 7);
- matrično vprašanje, ki je sestavljenlo iz več lestvičnih vprašanj oziroma trditev.

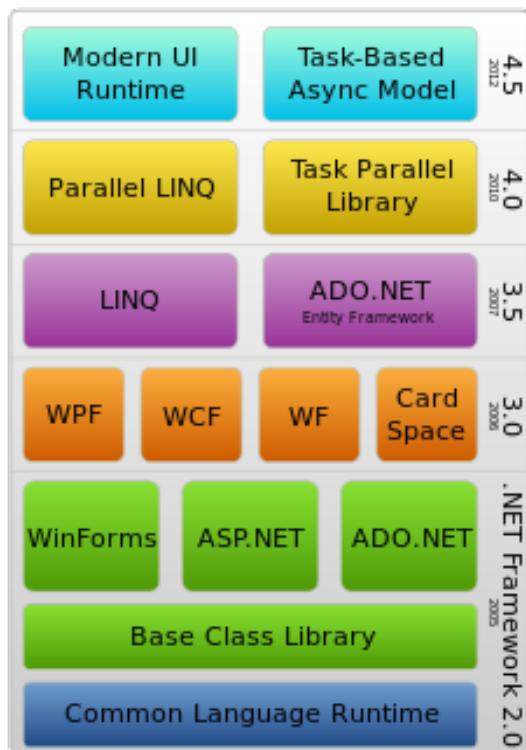
Pri kombiniranih vprašanjih navedemo nekaj predvidenih odgovorov, zraven pa dodamo možnost svojega odgovora, če anketiranu ne ustreza noben podan odgovor.

Poglavlje 3

Uporabljene tehnologije

3.1 Ogrodje .NET

.NET je Microsoftovo programsko ogrodje, ki deluje predvsem na operacijskih sistemih Windows in služi za razvoj aplikacij ter servisov, ki temelji na .NET tehnologiji. Zaradi svoje uporabnosti razvijalcem omogoča hiter razvoj. Ogrodje je sestavljeno iz dveh glavnih komponent (slika 3.1). S programsko kodo, napisano v okolju .NET, upravlja tako imenovani izvajalnik kode skupnega jezika (angl. Common Language Runtime, s kratico CLR). To je nekakšna virtualna naprava, ki vsebuje servise, ki skrbijo za varnost, upravljajo s spominom, napakami in podobnimi rečmi. Programska koda se namesto v strojno kodo prevede v tako imenovano vmesno kodo. To pa omogoča souporabo več programskih jezikov hkrati, kar pomeni, da lahko razvijalec uporablja sebi najljubši programski jezik, ki ga ogrodje podpira. Prav tako lahko v svojo kodo vključuje metode, napisane v drugih jezikih. Druga komponenta je osnovna knjižnica (angl. Base Class Library), ki vsebuje ogromno razredov z najrazličnejšimi funkcionalnostmi. Zajema različna področja programskih zahtev, kot so dostop do podatkov, povezava s podatkovno bazo, izdelava uporabniških vmesnikov, kriptografija, numerični algoritmi in omrežna komunikacija. Ti razredi so organizirani po imenskih prostorih, imenovanih ‘namespaces’. [1]



Slika 3.1: Zgradba ogrodja .NET.

3.2 ASP.NET

ASP.NET je tehnologija, ki jo je razvilo podjetje Microsoft in služi za izdelavo sodobnih dinamičnih spletnih strani in servisov. Je sestavni del ogrodja .NET (slika 3.1) in zaradi svoje arhitekturne zasnove omogoča pisanje kode v kateremkoli programskem jeziku, ki ga podpira ogrodje. V tem se tudi najbolj razlikuje od tehnologije Active Server Page (ASP), ki velja za predhodnico ASP.NET-a in ki je bila usmerjena zgolj v skriptno usmerjene jezike. [2] Najpogostejsa jezika tehnologije ASP.NET pa sta VisualBasic in C#. Sam sem pri izdelavi svoje aplikacije uporabil zadnjega, torej C#.

3.3 Visual Studio 2010

Microsoft Visual Studio je najbolj razširjeno orodje za pisanje kode v .NET ogrodju. Omogoča nam razvijanje spletnih strani, spletnih storitev, uporabniških vmesnikov in *Windows Forms* aplikacij v različnih programskih jezikih. Najbolj znani so C#, VisualBasic, C++ in Javascript, mnoge pa je mogoče namestiti v orodje. Vsebuje uporabniku prijazen urejevalnik kode in orodja za delo s podatkovno bazo, nepogrešljiv pa je tudi tako imenovani razhroščevalnik (angl. debugger), ki nam je v veliko pomoč predvsem pri iskanju in odpravljanju napak. [3]

3.4 PowerDesigner

PowerDesigner je programsko orodje za modeliranje, ki ga je razvilo podjetje Sybase. Uporablja se za risanje različnih diagramov in je zaradi enostavne uporabe zelo priljubljen. Z njim je mogoče ustvarjati različne modele, kot so konceptualni podatkovni model, fizični podatkovni model in poslovni procesni model. Uporabil sem ga pri izdelavi konceptualnega podatkovnega modela in diagrama primerov uporabe. Z njim je mogoče izdelati še mnoge druge dijagrame, kot so diagrami zaporedja, objektni, razredni, komponentni diagrami in diagrami prehajanja stanj. [4]

3.5 HTML

HTML (Hyper Text Markup Language) je označevalni jezik, ki se uporablja za izdelavo spletnih strani in aplikacij. Ukaze imenujemo tudi značke, ki se vedno nahajajo med znakoma < in > (na primer
). Poznamo dve vrsti značk: [1]

- samostojne značke (na primer
), ki ne potrebujejo zaključka, kot je nova vrstica, in

- začetne ter končne značke (na primer <tag> in </tag>), ki nam povedo, od kod do kod sega učinek značke, ki jih lahko nato znotraj para poljubno gnezdim.

3.6 CSS

CSS (Cascading Style Sheets) omogoča, da predpišemo obliko posameznih elementov ločeno od vsebine HTML-kode, kar močno poveča preglednost napisane kode. Z njim lahko določimo vrsto oblikovnih lastnosti, kot so poravnave, obrobe, pozicije, ozadja, razmiki, pisave, na enostaven način. Najnovejša različica je CSS 3, ki razvijalcem omogoča veliko novih dodatkov za vizualno izpopolnitve spletnih strani. Med temi novostmi bi izpostavil nove atribute, ki so postali zelo priljubljeni: *border-radius*, ki ustvari zaokrožene robove, ter *box-shadow* in *text-shadow* za senčenje. [6]

3.7 Javascript

JavaScript je objektni skriptni programski jezik, ki so ga razvili v podjetju Netscape. Izvaja se pri odjemalcu (internetnem brskalniku) in lahko sodeluje s HTML-kodo, zato je postal med spletnimi programerji eden izmed najbolj priljubljenih orodij za izdelavo dinamičnih spletnih strani.

Poglavlje 4

Planiranje, analiza, načrtovanje in razvoj aplikacije

4.1 Strukturni pristop k razvoju aplikacije

Za razvoj aplikacije sem izbral strukturni pristop, ki je eden izmed starejsih, a se še vedno pogosto uporablja. Ključna za to odločitev je bila pomembnost podatkovne baze, ki pri strukturnem pristopu predstavlja osnovo, okoli katere se kasneje razvijajo deli sistema. Že na samem začetku sem se odločil, da bom aplikacijo napisal v programskem jeziku C#, ki je objektni programski jezik in tako v osnovi primernejši za objektni pristop k razvoju informacijskih sistemov. Hkrati sem pri razvoju uporabil tudi agilnost, saj se nisem strogo držal navodil strukturnega pristopa, ampak sem dal večji poudarek na delujajočo kodo.

Pred dobrim desetletjem je agilen pristop k razvoju informacijskih rešitev pridobil veliko podporo med razvijalci. Z namenom, da bi definirali smernice in določili glavna načela, se je februarja 2001 sestalo 17 metodologov z znanji iz različnih področij, ki so napisali Manifest agilnosti, iz katerega so izpeljali štiri osnovna načela [12]:

- **posamezniki in njihovo sodelovanje** so pomembnejši od procesov

in orodij;

- **delujoči program** je pomembnejši od obsežne dokumentacije;
- **sodelovanje z naročnikom** je pomembnejše od pogajanja, ki se nanaša na pogodbo;
- **reagiranje na želene spremembe** je pomembnejše od sledenja načrtu.

Pomembno pri tem je, da pri razvoju upoštevamo tudi desni del načela, vendar imajo levi deli (označeni s krepkim tiskom) pred njimi prednost.

Dva izmed osnovnih principov agilnih metod, ki ju zagovarja Ambler [12], sta uporaba diagramov iz različnih pristopov razvoja ter ustvarjanje ravno toliko modelov in dokumentacije, kot je potrebno. V razvoju sem tako zajel bistvene postopke strukturnega pristopa. V zajemu in specifikaciji zahtev sem na kratko opisal namen aplikacije in vse funkcionalne ter nefunkcionalne zahteve in dodal diagram primerov uporabe. Kot najpomembnejši izdelek faze analize sem izdelal podatkovni model in diagram funkcionalne razgradnje. Rezultat načrtovanja je načrt podatkovne baze, ki je služil kot osnova za izdelovanje moje aplikacije, ki sem jo sproti testiral.

4.2 Namén aplikacie

V svoji aplikaciji sem želel zagotoviti pripravo in izdelavo anket, ki so prilagojene določeni skupini anketirancev in omogočajo možnost za longitudinalno anketiranje. Po navadi se ankete, namenjene različnim ciljnim skupinam, vsaj delno razlikujejo, saj nas pri različnih področjih raziskovanja zanimajo različne stvari. Poleg izdelave prilagojene ankete je namen aplikacije, da se registrirani uporabniki razvrstijo v različne ciljne skupine anketirancev. Tako lahko vsako ciljno skupino anketiramo z njej prilagojeno anketo. Na ta način rešimo težavo s prilagajanjem anket več ciljnim skupinam anketirancev. Aplikacija prav tako omogoča večkratno izvajanje ene ankete pri isti skupini

anketirancev, s čimer lahko pridemo do longitudinalnih raziskav na želenem področju. Aplikacija omogoča izvoz podatkov v CSV-obliku (angl. Comma Separated Value), ki je standardna za večino orodij za nadaljnjo obdelovanje in analizo podatkov.

4.3 Funkcionalne in nefunkcionalne zahteve

V praksi se večkrat zgodi, da so prve faze razvoja slabo izvedene, kar se pozna pri nadaljnji izvedbi projekta. Zato je za uspešnost razvoja neke aplikacije pomembno, da so funkcionalne in nefunkcionalne zahteve natančno navedene. Te zahteve so po navadi predstavljene s strani naročnika, ponudnik pa jih mora pri izvedbi projekta v celoti upoštevati. V svojem primeru sem naročnik in razvijalec, zato lahko vse funkcionalnosti aplikacije določim sam.

4.3.1 Funkcionalne zahteve

Zahteve, ki opisujejo delovanje programske opreme in se nanašajo na želeno funkcionalnost, se imenujejo funkcionalne zahteve. Aplikacija bo ločila med tremi tipi uporabnikov, zato je najbolje, da funkcionalne zahteve porazdelim med te akterje. Največ funkcionalnih zahtev bo imel administrator, ki ga bo aplikacija ob prijavi v sistem preusmerila na administratorski uporabniški vmesnik, ta pa bo administratorju omogočal:

- izdelavo prilagojenih anket, sestavljenih iz različnih sklopov vprašanj (vprašanja bodo tekstovna, izbirna, lestvična in matrična);
- urejanje registriranih uporabnikov in dodeljevanje pravic;
- možnost izbrisca obstoječih anket in dodeljevanja ciklov;
- dodajanje ali izbris elementa vrednotenja;
- možnost izvoza podatkov v CSV-obliku.

Tako administratorja kot registriranega uporabnika aplikacija ob prijavi preusmeri na ustrezen uporabniški vmesnik, ki uporabniku omogoča, da:

- spreminja svoje uporabniške podatke;
- ima možnost izbire ankete za reševanje, ki so mu na voljo;
- lahko izbrano anketo reši.

Oba, administrator in registriran uporabnik, bosta imela možnost, da se bo sta lahko iz sistema odjavila. Gost se bo v sistem lahko prijavil kot gost, na voljo pa mu bodo (podobno kot registriranemu uporabniku) izbira in reševanje ankete, vendar ne bo imel možnosti spremeniti svojih uporabniških podatkov, saj v sistemu ne bo registriran. Za lažje razumevanje je v nadaljevanju prikazan model primerov uporabe (slika 4.1).

4.3.2 Nefunkcionalne zahteve

Za razliko od funkcionalnih zahtev, ki se lahko od sistema do sistema zelo razlikujejo, so si nefunkcionalne zahteve med seboj zelo podobne. Te zahteve opisujejo osnovne kvalitete, ki zajemajo specifične uporabniške scenarije. V večini primerov so to zahteve, ki jih mora sistem podpirati ves čas svojega delovanja. [10]

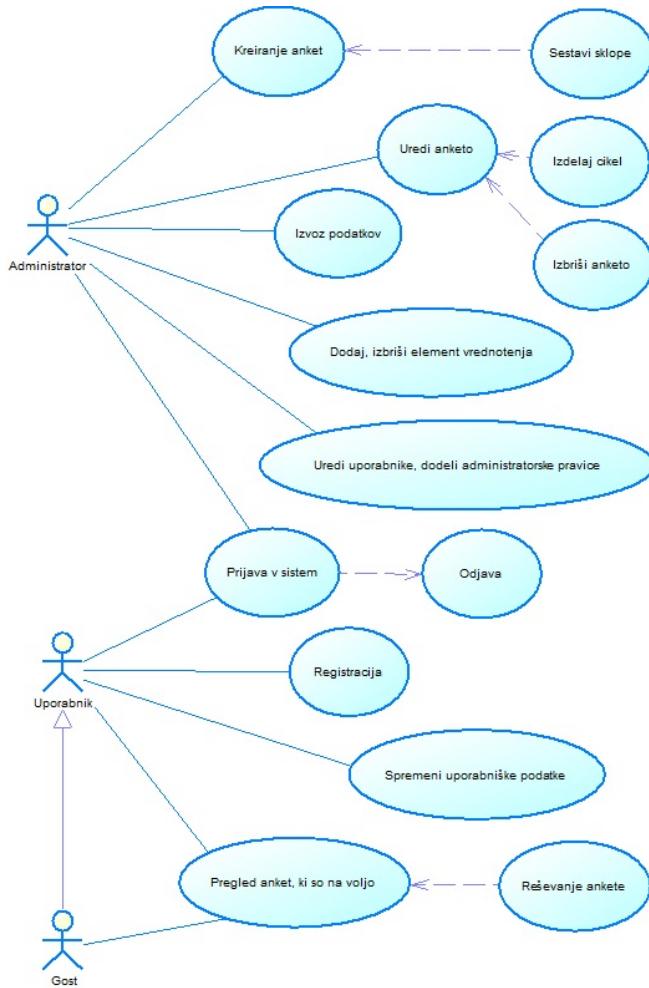
Najpomembnejše nefunkcionalne zahteve moje aplikacije so naslednje:

- razpoložljivost – sistem mora biti ves čas razpoložljiv;
- podatkovna baza – podatki se shranjujejo v ustrezeno bazo podatkov, ki se lahko izvaja na SQL Serverju 2008;
- varnost podatkov – podatkovna baza mora zagotavljati ustrezeno varnost podatkov;
- zanesljivost – sistem mora delovati brezhibno in brez napak;

- uporabnost – sistem mora biti enostaven za uporabo, vmesniki pa morajo biti prijazni do uporabnika.

4.4 Diagram primerov uporabe

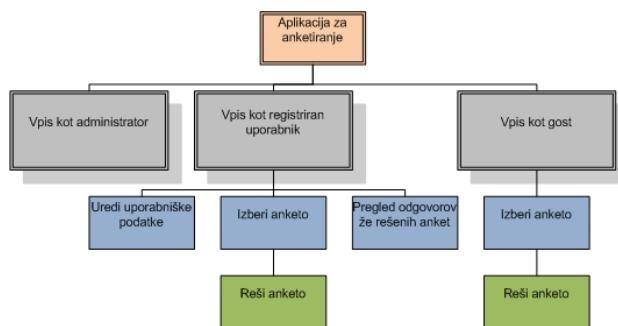
Diagram primerov uporabe predstavlja vizualen prikaz komunikacije med akterji in sistemom. Moj model prikazuje primere uporabe za celotno aplikacijo (slika 4.1) in dodatno pripomore k razumevanju funkcionalnih zahtev aplikacije.



Slika 4.1: Primeri uporabe za mojo aplikacijo.

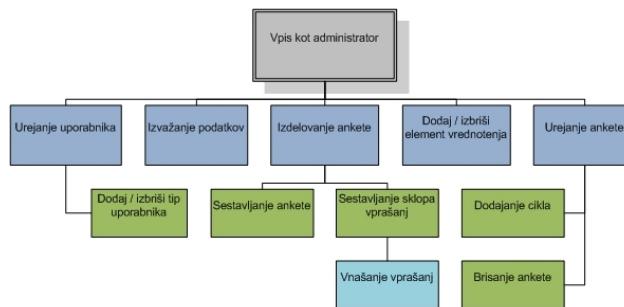
4.5 Diagram funkcionalne razgradnje

Diagram funkcionalne razgradnje nam prikazuje hierarhijo funkcij, ki jih bo aplikacija podpirala. Za uporabo tega diagrama sem se odločil, da tako še nazorneje predstavim strukturo funkcij, ki bodo uporabnikom na voljo v posameznem programskem sklopu. Čeprav je to delno že razvidno iz diagrama primerov uporabe, ta tehnika nazorneje prikazuje hierarhično odvisnost med funkcijami aplikacije. Kot je razvidno iz slike 4.2, bosta imela registriran uporabnik in gost podobne funkcije s to razliko, da gost ne bo mogel spremenjati svojih podatkov in dostopati do podatkov že rešene ankete.



Slika 4.2: Diagram funkcionalne razgradnje za aplikacijo.

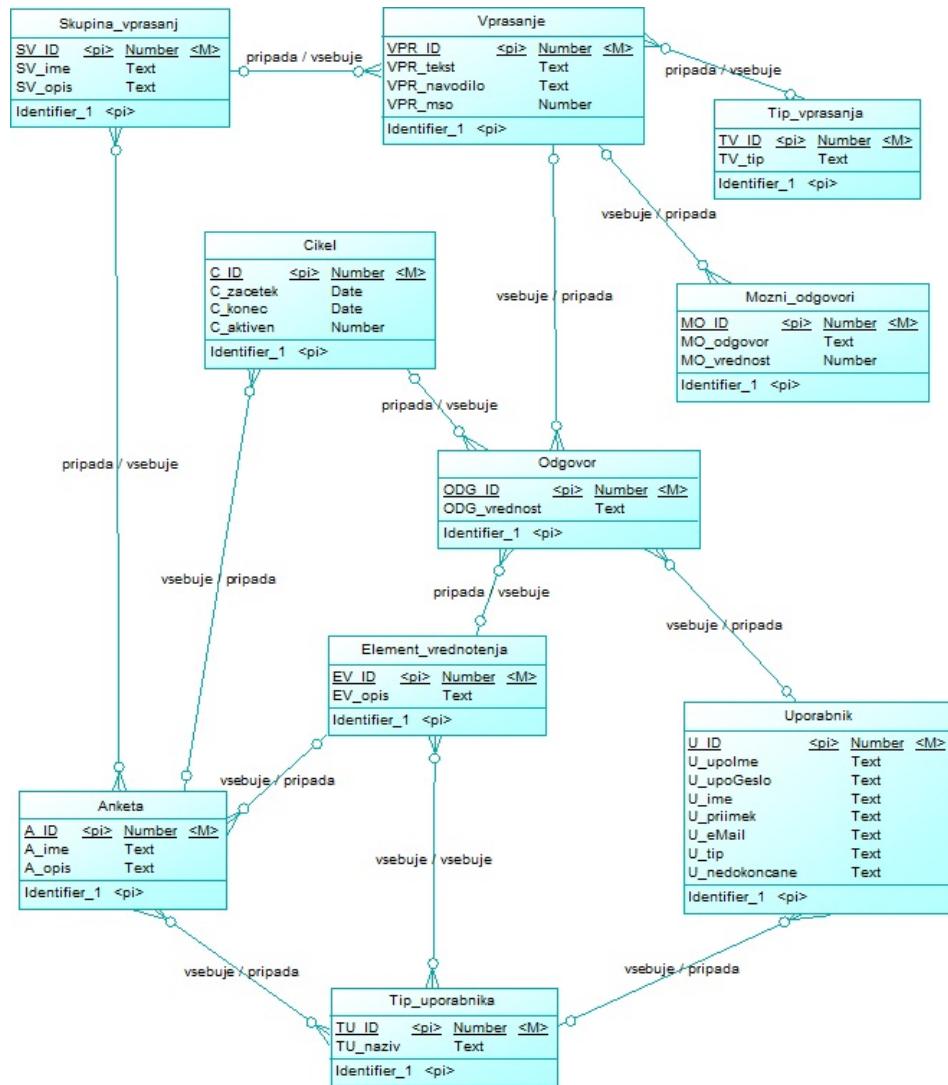
Največ funkcij pa bo pripadalo administratorju aplikacije, kot je razvidno iz navpične hierarhije funkcij na sliki 4.3.



Slika 4.3: Diagram funkcionalne razgradnje za prijavo kot administrator.

4.6 Podatkovni model

Na podlagi zahtev in pričakovane funkcionalnosti aplikacije sem zgradil ustrezen podatkovni model. Pri izdelavi modela sem temeljil na modelu, prikazanem v doktorski disertaciji doc. dr. Damjana Vavpotiča [13]. V fazi analize je to eden izmed najpomembnejših izdelkov. Kot je razvidno na sliki 4.4, je sestavljen iz 10 entitet.



Slika 4.4: Podatkovni model aplikacije.

4.7 Opis podatkovne baze

Že pred začetkom pisanja kode sem s pomočjo podatkovnega modela izdelal podatkovno bazo s potrebnimi entitetami. Ker program Visual Studio 2010 omogoča ustvarjanje podatkovne baze, sem jo izdelal kar v njem. Za nemoteno delovanje aplikacije sem izdelal naslednje tabele:

- **Vprasanje:**

Ta tabela vsebuje splošne podatke nekega vprašanja. Zagotovo je to nepogrešljiva tabela, ki se bo polnila ob vsakem vnosu nekega vprašanja. Tabela vsebuje polje VPR_tekst, ki je namenjeno vprašanju v tekstovni obliki, in polje VPR_navodilo, prav tako tekstovne oblike, ki je namenjeno navodilu za reševanje trenutnega vprašanja. Polje VPR_mso vsebuje nastavitev določenega vprašanja, primarni ključ pa je identifikacijska številka, poimenovana VPR_ID. Tabela vsebuje tudi dva tuja ključa, TV_ID številskega tipa, s katerim se povezuje s tabelo Tip_vprasanja, in SV_ID, ki je prav tako tipa int, s katerim dostopa do tabele Skupina_vprasanj.

- **Tip_vprasanja:**

Ta tabela vsebuje podatke o tipu vprašanja. Identifikacijska številka je primarni ključ (TV_ID), označuje pa določen tip vprašanja, ki je v tekstovni obliku in shranjen v koloni TV_tip.

- **Mozni_odgovor:**

V tej tabeli so shranjeni vsi možni odgovori, ki pripadajo določenemu vprašanju, zato vsebuje tuj ključ VPR_ID, s katerim se povezuje s tabelo Vprasanje. Identifikator je MO_ID, zraven pa vsebuje koloni MO_odgovor, kjer je shranjen možni odgovor v tekstovni obliku, in MO_vrednost, ki vsebuje vrednost tega odgovora v številski obliku.

- **Skupina_vprasanj:**

Ta tabela zajema podatke o nekem sklopu vprašanj, ki je sestavljen iz več vprašanj, zato je njegov ključ hkrati tuj ključ tabele Vprasanje.

Vsebuje dva podatka o skupini, in sicer ime skupine vprašanj, poimenovano SV_ime, ki je tekstovne oblike, in opis skupine vprašanj SV_opis, ki je prav tako tekstovne oblike. Skupine vprašanj se med seboj ločijo z identifikacijsko številko SV_ID, ki je tudi primarni ključ.

- **Anketa:**

Ta tabela vsebuje podatke o neki anketi, to sta ime ankete (A_ime) in opis ankete (A_opis), ki sta oba tekstovne oblike. Tabela prav tako vsebuje identifikacijsko kolonico A_ID številskega tipa, za dostop do tabele Element_vrednotenja pa služi tuj ključ EV_ID.

- **Tip_uporabnika:**

Ta tabela hrani določeno število tipov anketiranih uporabnikov, ki so v tekstovni obliki shranjeni v koloni TU_naziv. Identifikator je shranjen v TU_ID.

- **Cikel_vrednotenja:**

Ta tabela določa cikel izvajanja neke ankete, ki se lahko izvaja večkrat. Vsebuje identifikacijsko številko CV_ID, začetek in konec nekega cikla, ki sta datumskega tipa (CV_zacetek in CV_konec), ter polje, ki označuje ali je anketa aktivna ali neaktivna, in ki se imenuje CV_aktivnen. Poleg tega vsebuje tuji ključ A_ID, s katerim dostopa do Ankete.

- **Uporabnik:**

V tej tabeli so shranjeni vsi podatki o registriranih uporabnikih in administratorjih, kot so ime (U_ime), priimek (U_priimek), elektronski naslov (U_eMail), tip uporabnika (U_tip), ključni za prijavo pa sta uporabniško ime (U_upoIme) in uporabniško geslo (U_upoGeslo). Vsebuje tudi polje U_nedokoncane, ki je tekstovnega tipa in služi za hrambo podatkov o predčasno zaključenem reševanju anket. Identifikacijska številka se shranjuje v U_ID, vsebuje pa tudi tuj ključ tabele Tip_uporabnika TU_ID.

- ***Element_vrednotenja:***

Ta tabela vsebuje element vrednotenja, ki nam pove, kaj z določeno anketo ocenjujemo. Vsebuje polje EV_opis, ki hrani ta element vrednotenja, in identifikacijsko številko tega elementa, poimenovano EV_ID.

- ***Odgovor:***

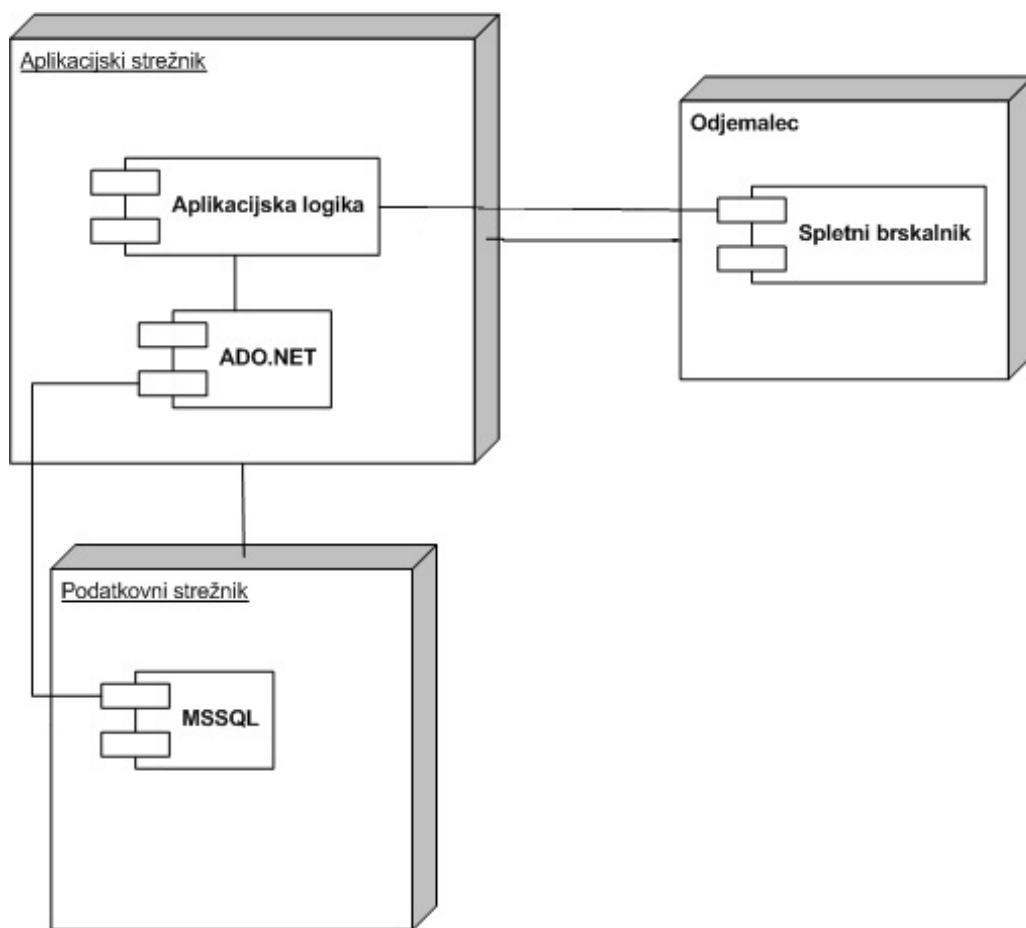
Ta tabela je ena izmed pomembnejših tabel, kamor se shranjujejo vsi odgovori neke rešene ankete. Shranjujejo se v tekstovno polje O_vrednost. Identifikacijska številka odgovora je shranjena v O_ID, vsebuje pa tudi tuje ključe U_ID za povezavo s tabelo Uporabnik, CV_ID za povezavo s Cikel_vrednotenja, V_ID za povezavo s tabelo Vprasanje in EV_ID za povezavo s tabelo Element_vrednotenja.

Pri povezavah med entitetami mnogo na mnogo so nastale vmesne tabele, ki služijo zgolj za povezavo med tabelama, ki sta v povezavi. V našem primeru so nastale tri takšne tabele:

- ***Anketa_Skupina_vprasanj***, ki povezuje tabeli Skupina_vprasanj in Anketa, zato vsebuje njuna primarna ključa;
- ***Anketa_Tip_uporabnika***, ki povezuje tabeli Tip_uporabnika in Anketa;
- ***Element_vrednotenja_Tip_uporabnika***, ki povezuje tabeli Element_vrednotenja in Tip_uporabnika.

4.8 3-nivojska arhitektura

Aplikacija se bo izvajala v trinivojski arhitekturi, kar pomeni, da bosta na strežniku tako podatkovna baza kot tudi aplikacija, uporabniki pa bodo do nje dostopali preko spletnega brskalnika (slika 4.5). Glavni razlog za to odločitev je, da se v tem primeru na strežniku izvajata celotno procesiranje in nadzor nad vsemi dostopi do podatkov. Posledica tega je, da se pri odjemalcu izvaja zelo malo procesiranja.



Slika 4.5: 3-nivojska arhitektura aplikacije.

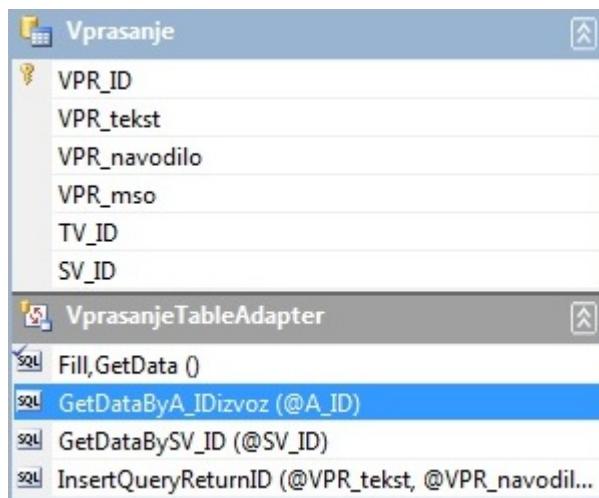
4.9 Razvoj aplikacije

Za izdelavo svoje spletnne aplikacije za anketiranje sem uporabil razvijalno orodje Microsoft Visual Studio 2010, v katerem sem izdelal povsem nov projekt spletnne aplikacije (ASP.NET Empty Web Application). Za razvojni jezik sem izbral C#, ki je trenutno eden izmed najpogosteje uporabljenih jezikov za delo z Microsoftovo .NET platformo. Ta nam omogoča hiter in lažji razvoj ter izvajanje obsežnih in zahtevnih ter majhnih in enostavnih projektov.

Na samem začetku izdelave sestavnih delov aplikacije sem tem določil prioritete. Za nadaljne delo je to pomemben korak, saj se ti deli med seboj prepletajo, za sprotno testiranje aplikacije pa je ključnega pomena vrstni red izdelave teh delov. Zaradi pomembnosti podatkovne baze sem se najprej lotil njene izdelave, nato pa začel implementirati svojo aplikacijo, pri kateri sem najprej začel z administratorskim uporabniškim vmesnikom za izdelavo anket. Nadaljeval sem z urejanjem anket in se nato lotil uporabniškega vmesnika za registrirane uporabnike ter ga v večini dokončal. Zatem sem dokončal administratorski uporabniški vmesnik in izdelal uporabniški vmesnik za goste, ki je zelo podoben vmesniku za registrirane uporabnike.

4.9.1 Povezava s podatkovno bazo

Za dostop do baze podatkov sem uporabil razred, imenovan DataSet. To je ena izmed bistvenih komponent ADO.NET-arhitekture in je nekakšna spominska shramba podatkov, pridobljenih iz podatkovnega vira. Vsebuje lahko več TableAdapterjev, ki omogočajo komunikacijo med našo aplikacijo in podatkovno bazo. Za vsako pomembnejšo tabelo sem zato izdelal poseben TableAdapter, v katerega sem po potrebi vnašal poizvedbene stavke, na katere sem se kasneje v programski kodi skliceval. Vsak poizvedbeni stavek vsebuje ustrezeno SQL-poizvedbo. Zaradi velikega števila poizvedb je treba te vnešene poizvedbene stavke poimenovati logično, da v nadaljevanju izdelave aplikacije ne prihaja do neželenih zmešnjav. Primer TableAdapterja za tabelo Vprasanje prikazuje slika 4.6.



Slika 4.6: TableAdapter tabele Vprasanje.

Poizvedbeni stavek *GetDataByA_IDizvoz*, ki zahteva en atribut, vsebuje naslednji SQL-stavek:

```
SELECT Vprasanje.VPR_ID, Vprasanje.VPR_tkst, Tip_vprasanja.TV_tip
FROM Anketa INNER JOIN Anketa_Skopina_vprasanj ON Anketa.A_ID =
    Anketa_Skopina_vprasanj.A_ID
    INNER JOIN Skopina_vprasanj ON Anketa_Skopina_vprasanj.SV_ID =
        Skopina_vprasanj.SV_ID
    INNER JOIN Vprasanje ON Skopina_vprasanj.SV_ID = Vprasanje.SV_ID
    INNER JOIN Tip_vprasanja ON Vprasanje.TV_ID = Tip_vprasanja.TV_ID
WHERE (Anketa.A_ID = @A_ID)
```

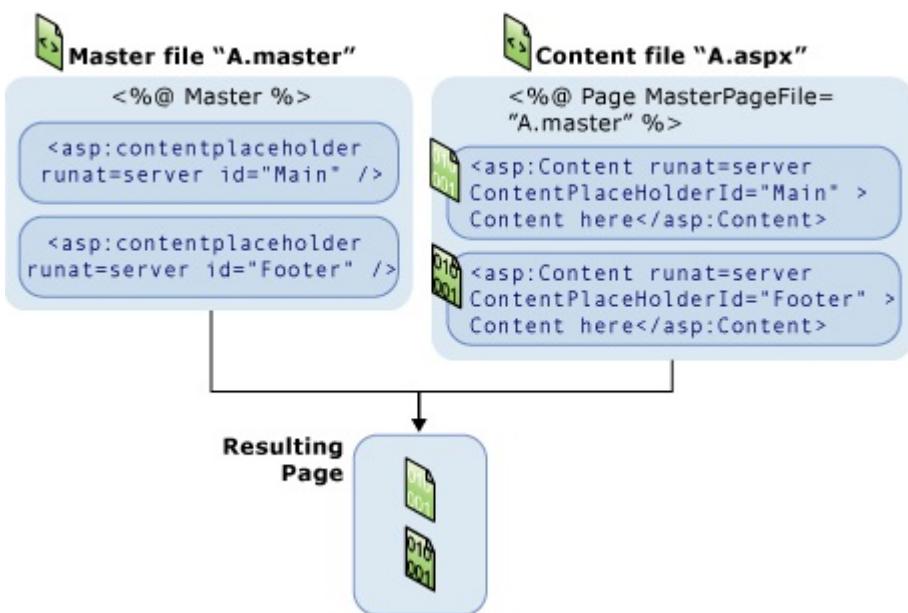
V kodi ga enostavno kličemo in shranimo v DataTable z ukazom:

```
DataTable dt = new
EAnkTableAdapters.VprasanjeTableAdapter().GetDataByA_IDizvoz(idAnkete);
```

4.9.2 Uporabniški vmesniki aplikacije

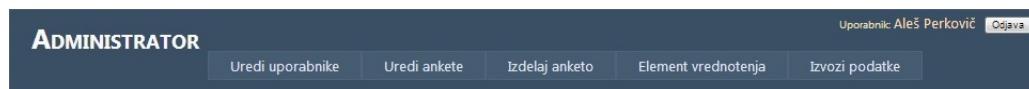
Aplikacijo sestavlja trije večji uporabniški vmesniki. Te sem ločil tako, da sem vsakemu dodelil svojo glavno stran (angl. Master Page). To je nekakšna osnovna predloga strani, ki nam omogoča, da ustvarimo enoten videz in obnašanje celotne skupine strani v aplikaciji, obenem pa priskrbi

predloge za druge strani s skupno postavitvijo in funkcionalnostjo. Vsebuje prostornike (angl. placeholders) za vsebino, ki se prepišejo iz vsebinskih strani. Končen rezultat je kombinacija glavne strani in vsebinske strani, pri čemer vsebinska stran vsebuje vsebino, ki je prikazana v prostorniku na glavni strani (slika 4.7).



Slika 4.7: Prikaz delovanja glavne in vsebinske strani.

Vsek uporabniški vmesnik ima točno določeno funkcionalnost. Med vsemi tremi je najobsežnejši administratorski uporabniški vmesnik (slika 4.8), ki bo omogočal uporabniku z administratorskimi pravicami izdelavo ankete, urejanje anket, urejanje uporabnikov, dodajanje ter brisanje elementov vrednoteњa in izvoz podatkov. Druga dva uporabniška vmesnika pa sta namenjena registriranim uporabnikom in gostom. Uporabniški vmesnik za registrirane uporabnike (slika 4.9) bo tem omogočal spremiњanje osebnih podatkov in izbiro ankete za reševanje. Navadni gosti bodo imeli le možnost izbire ankete za reševanje (slika 4.10).



Slika 4.8: Glavna stran za administratorja.



Slika 4.9: Glavna stran za registriranega uporabnika.



Slika 4.10: Glavna stran za gosta.

4.10 Testiranje

Testiranje aplikacije se je izvajalo vzporedno z razvojem vsakega modula posebej v razvojnem okolju Visual Studio 2010, ki omogoča sprotno razhroščevanje. Vsak končani del aplikacije sem sproti preizkusil in odpravil večino napak, ki so nastale. Med testiranjem sem dal večji poudarek na tiste dele aplikacije, kjer se podatki vpisujejo v podatkovno bazo. Predvsem je pomembno, da so tipi podatkov ustrezeni in da v nasprotnem primeru aplikacija javi ustrezeno opozorilo. Poleg tega sem sproti preverjal delovanje SQL-poizvedb. Za polnjenje odgovorov sem izdelal manjšo funkcijo, ki je avtomatsko polnila naključne odgovore, kar mi je v nadaljevanju pomagalo pri preizkusu izvoza podatkov.

4.11 Opis delovanja končne aplikacije

4.11.1 Prijava v sistem

Čeprav lahko aplikacijo za reševanje anket uporabljam tudi navadni gosti, za katere ni nujno, da so v sistemu registrirani, se mora v longitudinalnih raziskovah ciljna skupina uporabnikov, v našem primeru zaposleni v nekem podjetju, ki uporablja to aplikacijo, v sistem registrirati. Po uspešni registraciji se lahko v sistem prijavimo z uporabniškim imenom in uporabniškim geslom (slika 4.11), ki se preverita v podatkovni bazi. Če tam obstajata, sledi uspešna prijava v sistem.

The image shows a user interface for system login. It consists of three horizontal sections. The top section is light blue and contains fields for 'Uporabniško ime:' (username) and 'Geslo:' (password), each with an input field. Below these are two buttons: 'Vstopi' (Login) and 'ali' (or) 'Vstopi kot gost' (Login as guest). The middle section is dark grey and contains a 'Pozabljeno geslo:' (Forgot password) link, a 'Pošlji geslo na e-mail uporabnika:' (Send password to user's email) link, and fields for 'Uporabniško ime:' (username) and 'E-mail:' (email). The bottom section is light blue and contains an 'e-mail' input field and a 'Pošlji' (Send) button. At the very bottom, there is a link 'Ustvari novega uporabnika.' (Create new user).

Slika 4.11: Uporabniški vmesnik za prijavo.

Za registracijo je treba klikniti povezavo *Ustvari novega uporabnika*, ki nas

preusmeri na stran za registriranje novega uporabnika (slika 4.12). Pri registraciji je treba izpolniti vsa polja, saj nas v primeru praznih polj aplikacija opozori z obveznim poljem, tako da zraven polja pripisuje rdečo zvezdico. Prav tako nas sproti opozarja, če je vnos nepravilen (na primer nepravilen vnos elektronskega naslova), kar sem rešil z ustreznim regularnim izrazom: "\w+([-+.]\w+)*@\w+([-.\w+]*\.\w+([-.\w+])*".

The screenshot shows a user registration form with two sections: 'Uporabniško ime in geslo:' and 'Osebni podatki:'. In the 'Uporabniško ime in geslo:' section, the 'Uporabniško ime' field contains 'ales', the 'Geslo' field contains '*****', and the 'Potrdi geslo' field also contains '*****'. A red error message 'Gesli se ne ujemata!' (Passwords do not match!) is displayed next to the 'Potrdi geslo' field. In the 'Osebni podatki:' section, the 'Ime' field contains 'Ales', the 'Priimek' field is empty and has a red asterisk indicating it's required, the 'Leto rojstva' field contains '1987', the 'Elektronski naslov' field contains 'narobe' (which is highlighted in red with the message 'Nepravilen elektronski naslov!'), and the 'Delovno mesto' dropdown menu is set to 'Programer'. At the bottom are two buttons: 'Ustvari' (Create) and 'Prekliči' (Cancel).

Slika 4.12: Uporabniški vmesnik za registracijo.

Če pozabimo geslo, moramo izpolniti polja v sklopu *Pozabljeno geslo*, kamor vnesemo svoje uporabniško ime in svoj elektronski naslov. Aplikacija preveri, ali v podatkovni bazi obstaja uporabnik s takšnim uporabniškim imenom in elektronskim naslovom, in ob uspešni poižvedbi pošlje na elektronski naslov vse podatke uporabnika, vključno z uporabniškim gesлом.

4.11.2 Prijava kot administrator

Aplikacija omogoča uporabnikom, ki so prijavljeni kot administrator, povsem drugačne stvari kot navadnim uporabnikom ali gostom. Administratorji imajo možnost izdelave nove ankete in novih sklopov vprašanj, urejanja anket, upravljanja z uporabniki, dodajanja ali brisanja elementa vrednotenja in izvoza podatkov.

Urejanje obstoječih anket

Tukaj se upravlja z že izdelanimi anketami, ki se prikažejo v tabeli. Tako lahko določeni anketi dodelimo aktivnost, kar pomeni, da s klikom na gumb *Prični* oziroma *Končaj* aplikaciji povemo, pri kateri anketi naj se začne oziroma zaključi cikel vrednotenja. Zraven prikaže tudi datum, ki nam pove, kdaj se je neki cikel začel, in če anketa ni aktivna, nam sporoči tudi, kdaj je bil zadnji cikel te ankete končan (slika 4.13). Želeno anketo lahko tudi pogledamo s klikom na gumb *Poglej*, prav tako pa imamo možnost izbrisala ankete, ki je v prihodnje ne bomo več potrebovali.

ID	Ime ankete	EIV	Aktivnost	Začetek cikla	Konec cikla	Predogled	Izbriši
32	Anketa za počutje	Test	Končaj	12.5.2013	null	Poglej	Izbriši
33	Anketa o okolju	Test	Prični	6.5.2013	10.5.2013	Poglej	Izbriši
34	Zdrava prehrana	Test	Končaj	6.5.2013	null	Poglej	Izbriši
35	Anketa za zaposlene (počutje)	Počutje zaposlenih	Prični	8.5.2013	23.5.2013	Poglej	Izbriši
39	Anketa iz različnih sklopov vprašanj 1	Test	Končaj	14.5.2013	null	Poglej	Izbriši
40	Anketa za trgovsko podjetje	Test	Prični	null	null	Poglej	Izbriši

1

Slika 4.13: Tabela za urejanje obstoječih anket.

Za prikaz takšnih podatkov sem v večini primerov uporabil gradnik Grid-View, ki nam prikaže želene podatke v tabelarični obliki. Enostavno ga lahko napolnimo tudi s podatkovno tabelo (Data Table), kot je vidno v kodi, ki sledi, in česar sem se pri izdelavi aplikacije velikokrat posluževal. Prav tako

ga lahko vizualno uredimo s CSS-razredom, kar nam dopušča velik spekter vizualnih nastavitev. Rezultat taksnega razreda je tabela, ki nam prikazuje izdelane ankete, za katero sem uporabil CSS-razred, poimenovan mGrid, ki je razviden iz kode v nadaljevanju.

Naslednja koda nam prikazuje polnjenje gradnika GridView, ki je viden na sliki 4.13:

```
polniGVAnkete(){
    //pridobimo podatke iz podatkovne baze in polnimo gridView
    DataTable Ankete = new
        EAnkTableAdapters.AnketaTableAdapter().GetDataForAdminEdit();
    GVAnkete.DataSource = Ankete;
    GVAnkete.DataBind();
    for (int i = 0; i < GVAnkete.Rows.Count; i++){
        Button BAktivnost =
            (Button)GVAnkete.Rows[i].Cells[3].Controls[0];
        BAktivnost.CssClass = "btnInGrid";
        //pridobimo podatke, ce je trenutna anketa aktivna oziroma
        //je neaktivna
        DataTable DTA = new
            EAnkTableAdapters.Cikel_vrednotenjaTableAdapter().
            GetDataByAktiven1A_ID(Convert.ToInt32(GVAnkete.Rows[i].
            Cells[0].Text));
        //ce sta aktivna vsaj dva cikla ene ankete, javi napako
        if(DTA.Rows.Count > 1){
            LNapaka.Text="Aktivna sta vsaj dva cikla iste
                ankete.";
            LNapaka.Visible = true;
        }
        //ce je trenutna anketa aktivna, dodamo gumbu ustrezno ime
        //ukaza, ga dodamo v gridView in polnimo se nenapolnjeni
        //celici za zacetek in konec cikla
        if (DTA.Rows.Count == 1){
            BAktivnost.Text = "Koncaj";
            BAktivnost.CommandName = "zakljuciCikel";
            GVAnkete.Rows[i].Cells[3].Controls.Add(BAktivnost);
            GVAnkete.Rows[i].Cells[4].Text =
                Convert.ToDateTime(DTA.Rows[0]["CV_zacetek"]).
                ToString().Date.ToString("d");
            GVAnkete.Rows[i].Cells[5].Text = "null";
        }
        //ce je trenutna anketa neaktivna, dodamo gumbu ustrezno
        //ime ukaza, ga dodamo v gridView in ustrezno polnimo se
        //nenapolnjeni celici
```

```

        else if (DTA.Rows.Count == 0){
            BAktivnost.Text = "Princni";
            BAktivnost.CommandName = "zaciCikel";
            GVAankete.Rows[i].Cells[3].Controls.Add(BAktivnost);
            //pridobimo podatke o vseh ciklih za trenutno anketo
            DataTable DTCV = new
                EAnkTableAdapters.Cikel_vrednotenjaTableAdapter().
                GetDataByA_IDaktiven0(Convert.ToInt32(GVAankete.
                    Rows[i].Cells[0].Text));
            //napolnimo ustrezni celici z zacetkom in koncem
            zadnjega izvedenega cikla
            if (DTCV.Rows.Count > 0){
                GVAankete.Rows[i].Cells[4].Text =
                    Convert.ToDateTime(DTCV.Rows[0]
                        ["CV_zacetek"].ToString()).Date.
                        ToString("d");
                GVAankete.Rows[i].Cells[5].Text =
                    Convert.ToDateTime(DTCV.Rows[0]
                        ["CV_konec"].ToString()).Date.
                        ToString("d");
            }
            //ce se ni cikla za trenutno anketo, napolnimo
            celici za zacetek in konec cikla z "null"
            else if (DTCV.Rows.Count == 0){
                GVAankete.Rows[i].Cells[4].Text = "null";
                GVAankete.Rows[i].Cells[5].Text = "null";
            }
        }
        //gumb za pregled ankete z ustreznim imenom ukaza dodamo v
        //gridview
        Button BPregled =
            (Button)GVAankete.Rows[i].Cells[6].Controls[0];
        BPregled.CssClass = "btnInGrid";
        BPregled.CommandName = "poglejAnketo";
        GVAankete.Rows[i].Cells[6].Controls.Add(BPregled);
        //gumb za izbris ankete z ustreznim imenom ukaza dodamo v
        //gridview
        Button BIzbrisAnketo =
            (Button)GVAankete.Rows[i].Cells[7].Controls[0];
        BIzbrisAnketo.CssClass = "btnInGrid";
        BIzbrisAnketo.CommandName = "izbrisAnketo";
        GVAankete.Rows[i].Cells[7].Controls.Add(BIzbrisAnketo);
    }
}

```

Ob kliku na enega izmed gumbov se kliče metoda RowCommandHandler(Object

sender, GridViewCommandEventArgs e), ki vsebuje naslednjo kodo:

```

protected void RowCommandHandler(Object sender, GridViewCommandEventArgs e)
{
    //klik na gumb "Princni"
    if (e.CommandName == "zaciCikel")
    {
        int row = Convert.ToInt32(e.CommandArgument);
        int id_ankete =
            Convert.ToInt32(GVAnkete.DataKeys[row]["A_ID"]);
        string dt = DateTime.Now.ToString();
        new EAnkTableAdapters.Cikel_vrednotenjaTableAdapter().
            InsertQuery(dt, null, 1, id_ankete);
        polniGVAnkete();
    }
    //klik na gumb "Koncaj"
    else if (e.CommandName == "zakljuciCikel")
    {
        int row = Convert.ToInt32(e.CommandArgument);
        int id_ankete =
            Convert.ToInt32(GVAnkete.DataKeys[row]["A_ID"]);
        string dt = DateTime.Now.ToString();
        new EAnkTableAdapters.Cikel_vrednotenjaTableAdapter().
            UpdateQueryKoncajCikel(dt, id_ankete);
        polniGVAnkete();
    }
    //klik na gumb "Izbrisiti"
    else if (e.CommandName == "izbrisitiAnketo")
    {
        int row = Convert.ToInt32(e.CommandArgument);
        int id_ankete =
            Convert.ToInt32(GVAnkete.DataKeys[row]["A_ID"]);
        new EAnkTableAdapters.AnketaTableAdapter().
            DeleteQuery(id_ankete);
        new EAnkTableAdapters.Anketa_Skupina_vprasanjTableAdapter().
            DeleteQuery(id_ankete);
        new EAnkTableAdapters.Anketa_Tip_UporabnikaTableAdapter().
            DeleteQueryAnketa(id_ankete);
        polniGVAnkete();
    }
    //klik na gumb "Pregled"
    else if (e.CommandName == "poglejAnketo")
    {
        int row = Convert.ToInt32(e.CommandArgument);
        int id_ankete =
            Convert.ToInt32(GVAnkete.DataKeys[row]["A_ID"]);
        this.ClientScript.RegisterStartupScript(this.GetType(),
            "PopUp", "<script language='JavaScript'>" + id_ankete +

```

```

        "&show=anketa" + "', null, 'resizable=0, scroll=1,
        width=840, height=600');" + "</script>");
    }
}

```

CSS razred mGrid, namenjen za vizualno ureditev izdelanega gradnika GridView:

```

.mGrid {
    width: 100%; background-color: #fff; border-bottom-left-radius: 15px;
    border-bottom-right-radius: 15px; box-shadow: 2px 2px 6px
        rgba(0,0,0,0.6);
}

.mGrid td {
    padding: 2px; color: #717171;
}

.mGrid tr:hover{
    background: #e0e0e0;
}

.mGrid th {
    padding: 4px 2px; color: #fff; background: #a1a1a1; font-size: 0.9em;
}

.mGrid .alt {
    background: #f0f0f0;
}

.mGrid .pgr {
    background: #a1a1a1;
}

.mGrid .pgr table{
    margin: 5px 0;
}

.mGrid .pgr td {
    padding: 0 6px; font-weight: bold; color: #fff; line-height: 12px;
    border-radius: 20px;
}

.mGrid .pgr a{
    color: #666; text-decoration: none;
}

.mGrid .pgr a:hover{
    color: #000; text-decoration: none;
}

```

Pregled in urejanje uporabnikov

V tem delu ima administrator dostop do vseh uporabnikov, saj se mu izpišejo potrebni podatki o vseh registriranih uporabnikih. Prav tako lahko te podatke administrator uredi, predvsem pa je pomembno, da lahko dodeli administratorske pravice drugim uporabnikom. To naredi tako, da pod kolono Tip Uporabnika spremeni črko U (uporabnik) v črko A (administrator). Ima pa tudi možnost izbrisala določenega uporabnika. Podobno kot v pregledu obstoječih anket je tudi tukaj gradnik za prikaz podatkov GridView. Na tej strani lahko tudi doda ali izbriše tip anketiranca, kot je razvidno iz slike 4.14.

Uporabniško Ime	Uporabniško Geslo	Ime	Priimek	Leto Rojstva	eMail	Tip Up	Tip anketiranca	
ales	ales1	Aleš	Perkovič	1987	a.p@gmail.com	A	Programer	Edit Delete
ursa	ursa1	Urska	Jeranko	1988	u.j@gmail.com	U	Analitik	Edit Delete
ema	ema1	Ema	Perkovič	2012	e.perkovic@gmail.com	U	Programer	Edit Delete
dejan	dejan1	Dejan	Cvetanoski	1985	d.c@test.si	U	Tester	Edit Delete
matevz	matevz1	Matevž	Premelč	1986	m.p@gmail.com	U	Programer	Edit Delete
spela	spela1	Špela	Lorger	1987	s.l@gmail.com	U	Programer	Edit Delete
rok	rok1	Rok	Jeranko	1996	rok.j@test.si	U	Kandidat	Edit Delete

1 2

Dodaj tip uporabnika:

 Izbriši tip uporabnika:

Slika 4.14: Urejanje uporabnikov.

Izdelava ankete

Administratorju aplikacija dovoljuje izdelavo anket in sklopov vprašanj. To dosežemo s klikom na gumb *Izdelaj anketo*. Za izdelavo ankete je treba predhodno ustvariti nov sklop vprašanj. Tega naredimo tako, da kliknemo *Izdelaj skupino vprašanj*. Ob kliku na ta gumb se aktivira ustrezni gradnik View, ki deluje skupaj z gradnikom MultiView. Skupaj omogočita vklop in izklop določenega odseka ASP.NET-strani. V tem primeru se izvede ukaz `MultiViewAnkVpr.SetActiveView(ViewVpr)`; pri čemer velja, da je gradnik MultiView poimenovan **MultiViewAnkVpr**, gradnik View pa **ViewVpr**. Kot je vidno na sliki 4.15, se nam nato prikažeta dve polji, in sicer:

- ime skupine, kamor vnesemo želeno ime skupine, in
- opis skupine, kjer je prostor za daljši opis te skupine vprašanj.

Slika 4.15: Vnos nove skupine vprašanj.

Obe polji sta obvezni za nadaljevanje vnosa vprašanj, na kar nas opozori rdeča zvezdica v primeru praznega polja. Ob kliku na gumb *Vnesi skupino* se v bazi podatkov ustvari nov sklop vprašanj in prikažejo se možnosti za tipe vprašanj (slika 4.15). V tej aplikaciji so nam na voljo širje večji tipi vprašanj:

- tekstovni tip, kamor vnesemo vprašanje in navodilo za to vprašanje;
- izbirni tip, kamor prav tako vnesemo vprašanje in navodilo, poleg tega pa moramo določiti, koliko možnih odgovorov bo mogoče izbrati (ena izbira ali več izbir) in možne odgovore, ki jih dodajamo s klikom na gumb *Dodaj nov odgovor*;
- lestvični tip, kamor vnesemo vprašanje in navodilo zanj in lestvico ocenjevanja od 1 do 3, 5 ali 7 (1 predstavlja najnižjo oceno);
- matrični tip, kjer moramo poleg vprašanja in navodila za matrično skupino vprašanj dodati tudi podvprašanja oziroma trditve, ki jih ocenjujemo na lestvici od 1 do 3, 4 ali 7, kjer 1 prav tako predstavlja najnižjo oceno.

Slika 4.16: Vnos vprašanj.

Vse odgovore potrdimo s klikom na gumb *Vnesi vprašanje*. Vnešena vprašanja lahko pregledamo s klikom na gumb *Prikaži vprašanja*, kjer lahko morebitna neustrezna vprašanja izbrišemo. Po končanem vnosu vseh vprašanj zaključimo izdelavo trenutnega sklopa vprašanj z klikom na gumb *Končaj*. Zatem imamo ponovno možnost izdelave nove skupine vprašanj. Ker je anketa te aplikacije sestavljena iz vsaj enega ali več skupin vprašanj, moramo za njeno izdelavo klikniti na gumb *Izdelaj anketo*, ki z ukazom `MultiViewAnkVpr .SetActiveView(ViewAnk);` aktivira ustrezen gradnik View, poimenovan **ViewAnk**, kjer se nam prikažeta dve tekstovni polji:

- Ime ankete, kamor vnesemo ime nove ankete, in
- Opis ankete, kamor vnesemo daljši opis te nove ankete.

Prav tako je za izdelavo nove ankete treba izbrati tipe uporabnikov, ki bodo lahko dostopali do te ankete in skupine vprašanj, ki bodo to novo anketo sestavlja. Tukaj imamo možnost pregledati določeno skupino vprašanj, po potrebi pa jo lahko izbrišemo. Obvezno moramo izbrati element vrednotenja

POGLAVJE 4. PLANIRANJE, ANALIZA, NAČRTOVANJE IN RAZVOJ APLIKACIJE

za novo anketo. Anketo dokončamo s klikom na gumb *Potrdi*.

The screenshot shows a user interface for creating a new survey. At the top, there are two buttons: "Izdelaj anketo" (Create survey) and "Izdelaj skupino vprašanj" (Create survey group). Below these are input fields for "Ime ankete:" (Survey name) and "Opis ankete:" (Survey description). Under "Izberite tip/tipe uporabnikov:" (Select participant type), there is a list of checkboxes for "Programer", "Analitik", "Tester", "Gost", and "Kandidat". Next to it is a dropdown menu for "Izberi Element Vrednotenja:" (Select element value) set to "Test". The main area displays a table of existing survey groups:

Ime skupine vprašanj	Opis skupine vprašanj	SV_ID	Predogled	Izbriši
Klimatske spremembe	Tukaj bo vprašalnik osredotočen na klimatske spremembe v vašem kraju. :)	10	[Poglej]	[Izbriši]
Prehrana	Nekaj besed o hrani.	11	[Poglej]	[Izbriši]
Testna Skupina Preizkus	Opis te skupine	25	[Poglej]	[Izbriši]
Počutje na delovnem mestu.	V tej skupini vprašanj bo glavni cilj pridobiti čim več informacij o počutju zaposlenih v nekem podjetju.	39	[Poglej]	[Izbriši]
Testna skupina vprašanj 1	Opis test.	62	[Poglej]	[Izbriši]
Testna skupina vprašanj 2.	Testna skupina	80	[Poglej]	[Izbriši]
Prodaja	Način pristopa k prodaji	81	[Poglej]	[Izbriši]
Zahtevnost strank	Kaj stranke zahtevajo od trgovcev	82	[Poglej]	[Izbriši]
Skladiščenje	Načini in hitrost skladiščenja	83	[Poglej]	[Izbriši]

At the bottom left is a page number "1" and at the bottom right is a "Potrdi" (Confirm) button.

Slika 4.17: Ustvarjanje nove ankete.

Dodajanje ali izbris elementa vrednotenja

Tukaj lahko administrator doda ali izbriše želen element vrednotenja. Obstojеči elementi vrednotenja se ob odprtju strani naložijo v gradnik DropDownList, ki se ob izbrisu nekega elementa znova naloži. Element vrednotenja enostavno dodamo tako, da vpišemo njegovo ime in opis v ustrezno polje ter kliknemo na gumb *Dodaj*, kot prikazuje slika 4.18.

Dodaj Element vrednotenja

Ime EV:

Opis EV:

Dodaj

Izbriši Element vrednotenja

Slika 4.18: Dodajanje ali izbris elementa vrednotenja.

Izvoz podatkov

Administrator lahko izvozi podatke v CSV-obliku. To naredi tako, da klikne *Izvozi podatke* v glavnem meniju in odpre se mu stran za izvoz podatkov. Tukaj je treba najprej izbrati, za kateri element vrednotenja naj nam aplikacija prikaže ankete. Nato se nam na desni strani izpišejo vsi cikli označene ankete, za katere lahko opravimo izvoz (slika 4.19).

Izberi Element vrednotenja, za katerega vam bo prikazalo ustrezne ankete:

Ankete, ki so na voljo za izvoz:

ID	Naziv ankete	Opis ankete
Select	32 Anketa za počutje	Opisuje počutje, bodisi trenutno ali pa ne.
Select	33 Anketa o okolju	naše okolje
Select	34 Zdrava prehrana	Načela zdravega prehranjevanja
Select	39 Anketa iz različnih sklopov vprašanj 1	Test

Izberi cikel izbrane ankete:

ID	Pričetek	Konec
466	maj, 2013	maj, 2013 <input checked="" type="checkbox"/>
456	maj, 2013	maj, 2013 <input checked="" type="checkbox"/>
446	maj, 2013	maj, 2013 <input type="checkbox"/>
416	maj, 2013	maj, 2013 <input type="checkbox"/>

Slika 4.19: Izvoz podatkov.

4.11.3 Prijava kot uporabnik

Vsak uporabnik, ki je registriran v sistemu, ima možnost, da spremeni svoje podatke in izbere ter reši anketo, ki je na voljo.

Urejanje računa

Tukaj lahko uporabnik spreminja svoje osebne podatke, ki so: ime, priimek, leto rojstva, elektronski naslov in delovno mesto. Za spremenjanje gesla mora vpisati staro geslo in dvakrat novo geslo, pri čemer se morata novi gesli ujemati, sicer aplikacija opozori na napačen vnosu gesla.

Pregled anket

V tem delu uporabnik dostopa do anket, ki so na voljo za njegov tip uporabnika. Razpoložljive ankete se izpišejo v tabeli, kjer sta vidna opis in ime ankete (slika 4.20). S postavitvijo miške na ime ankete se nam prikaže opis določene ankete. Anketo rešimo s klikom na povezavo *Reši*, kar nas preusmeri na stran za reševanje ankete. Če smo anketo že rešili, se nam zraven tabele z anketami, ki so na voljo za reševanje, prikaže nova tabela z rešenimi anketami (slika 4.21).

Ankete, ki so na voljo za reševanje:

ID	Ime	Stanje
1	ime ankete 1	Reši
2	ime ankete 2	Reši
3	ime ankete 3	Reši
24	Testna anketa je to!	Reši
31	Testna anketa 2.6	Reši
33	Anketa o okolju	Reši

Slika 4.20: Pregled anket, ki so na voljo.

Ankete, ki so na voljo za reševanje:			Že rešene ankete:		
ID	Ime	Stanje	ID	Ime	Stanje
1	ime ankete 1	Reši	31	Testna anketa 2.6	Poglej
2	ime ankete 2	Reši			
3	ime ankete 3	Reši			
24	Testna anketa je to!	Reši			
33	Anketa o okolju	Reši			

Slika 4.21: Pregled anket, ki so na voljo, in pregled že rešenih anket.

4.11.4 Prijava kot gost

Če v sistemu nismo registrirani in se tudi ne želimo registrirati (smo na primer le obiskovalec neke spletnne strani, ki uporablja to aplikacijo), nas aplikacija preusmeri na stran, ki je podobna tisti za registrirane uporabnike, le da nimamo možnosti za urejanje profila. Na voljo imamo le ankete, ki so namenjene gostom, in te lahko rešimo na enak način kot uporabniki.

4.11.5 Reševanje ankete

Tukaj se izpiše anketa za reševanje, in sicer se izpiše po skupinah vprašanj. Na vrhu strani se izpiše, kateremu elementu vrednotenja je anketa namenjena in opis trenutnega sklopa vprašanj (slika 4.22). Na koncu vsakega vprašanja je polje [pomoc], ki omogoči, da se ob postavitvi miške nanj prikaže še navodilo za trenutno vprašanje. Po rešenem prvem sklopu vprašanj, kliknemo *Naslednji sklop*, odgovori se shranijo v podatkovno bazo in prikaže se naslednji sklop z vprašanji. Za uspešno nadaljevanje reševanja ankete je potrebno odgovoriti na vsa vprašanja, sicer nas aplikacija opozori na nerešena vprašanja. Če rešujemo zadnji sklop, se prikaže gumb *Zaključi*, ki ob kliku prav tako kliče funkcijo za shranjevanje podatkov v podatkovno bazo. Ob uspešnem polnjenju baze podatkov nas aplikacija preusmeri nazaj na stran za pregled anket, ki so na voljo. Registriran uporabnik ima pri reševanju ankete možnost, da predčasno konča z reševanjem, kar naredi s klikom na

gumb *Nadaljuj naslednjič*. Na strani za pregled anket se mu pri predčasno končani anketi spremeni stanje iz *Reši* v *Nadaljuj* in tako lahko nadaljuje z reševanjem ankete.

Trenutni element vrednotenja: Počutje zaposlenih Ime trenutnega sklopa: Počutje na delovnem mestu.	
Kako dolgo ste že zaposleni v podjetju? [pomoc]	
<input type="radio"/> Od samega začetka <input type="radio"/> 10 - 8 let <input type="radio"/> 7 - 5 let <input type="radio"/> 4 - 2 let <input type="radio"/> 1 leto	
Kako se čutite sprejeti v delovnem kolektivu? [pomoc]	
<input type="radio"/> -2 <input type="radio"/> -1 <input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2	
Kako se razumete z zaposlenimi iz različnih oddelkov. [pomoc]	
Računovodstvo <input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 Kadrovski oddelek <input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 Proizvodnja <input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 Uprava <input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 Razvojni oddelek <input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 Tuji sodelavci <input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6	
Kateri termini delovnika vam odgovarjajo? [pomoc]	
<input type="checkbox"/> 6h - 14h <input type="checkbox"/> 7h - 15h <input type="checkbox"/> 8h - 16h <input type="checkbox"/> 9h - 17h <input type="checkbox"/> 10h - 18h	
Kaj bi na svojem delovnem mestu spremenili? [pomoc]	
<div style="border: 1px solid black; height: 40px; width: 100%;"></div>	
Naslednji sklop -->	
Nadaljuj naslednjič	

Slika 4.22: Reševanje ankete.

Poglavlje 5

Sklepne ugotovitve

Cilj diplomskega dela je bil razvoj in opis razvoja aplikacije, namenjene za longitudinalno anketiranje, prilagojeno tipom anketirancev. Med vsemi fazami razvoja je bila najobsežnejša izvedba, ki mi je vzela največ časa. Uspešno sem razvil aplikacijo z želeno funkcionalnostjo, ki omogoča ustvarjanje anket in njihovo izvajanje. Z izdelavo diplomskega dela sem utrdil svoja znanja, ki sem jih pridobil med študijem, najbolj pa tista, ki se nanašajo na razvoj informacijskih rešitev in programiranja v programskejem jeziku C#.

Pri razvoju aplikacije sem ugotovil, da bi jo bilo možno nadgraditi tako, da bi poganjala tudi kvize. Beležila bi čas, kdaj je neki kandidat začel reševati kviz in kdaj ga je zaključil. S programerskega vidika to ne bi smelo biti pretežko, v podatkovni bazi pa bi bilo treba narediti še eno tabelo, ki bi morala vsebovati vsaj dva atributa za začetek in konec reševanja kviza. V obstoječi tabeli Anketa, bi bilo nato treba dodati še vsaj en atribut, ki bi nam povedal, ali je anketa dejansko anketa ali je kviz. Takšno aplikacijo bi lahko določeno podjetje uporabljalo tudi za testiranje morebitnih novih kadrov, ki bi se prijavili na njihov razpis za neko delovno mesto.

Slike

3.1	Zgradba ogrodja .NET.	8
4.1	Primeri uporabe za mojo aplikacijo.	15
4.2	Diagram funkcionalne razgradnje za aplikacijo.	16
4.3	Diagram funkcionalne razgradnje za prijavo kot administrator.	16
4.4	Podatkovni model aplikacije.	17
4.5	3-nivojska arhitektura aplikacije.	21
4.6	TableAdapter tabele Vprasanje.	23
4.7	Prikaz delovanja glavne in vsebinske strani.	24
4.8	Glavna stran za administratorja.	25
4.9	Glavna stran za registriranega uporabnika.	25
4.10	Glavna stran za gosta.	25
4.11	Uporabniški vmesnik za prijavo.	26
4.12	Uporabniški vmesnik za registracijo.	27
4.13	Tabela za urejanje obstoječih anket.	28
4.14	Urejanje uporabnikov.	33
4.15	Vnos nove skupine vprašanj.	34
4.16	Vnos vprašanj.	35
4.17	Ustvarjanje nove ankete.	36
4.18	Dodajanje ali izbris elementa vrednotenja.	37
4.19	Izvoz podatkov.	37
4.20	Pregled anket, ki so na voljo.	38
4.21	Pregled anket, ki so na voljo, in pregled že rešenih anket.	39

4.22 Reševanje ankete.	40
--------------------------------	----

Literatura

- [1] .NET Framework. Dostopno na:
http://en.wikipedia.org/wiki/.NET_Framework, marec 2013.
- [2] ASP.NET. Dostopno na:
<https://en.wikipedia.org/wiki/ASP.NET>, marec 2013.
- [3] Visual Studio 2010. Dostopno na:
[http://msdn.microsoft.com/en-us/library/52f3sw5c\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/52f3sw5c(v=vs.100).aspx),
marec 2013.
- [4] PowerDesigner. Dostopno na:
<http://www.sybase.com/products/modelingdevelopment/powerdesigner>,
marec 2013.
- [5] HTML. Dostopno na:
<http://sl.wikipedia.org/wiki/HTML>, marec 2013.
- [6] CSS. Dostopno na:
<http://sl.wikipedia.org/wiki/CSS>, marec 2013.
- [7] M. Cencič. *Kako poteka pedagoško raziskovanje*, Ljubljana, 2009, str.
58–64.
- [8] G. Kalton, V. Vehovar. *Vzorčenje v anketah*, Ljubljana, 2001, str.
156–168.
- [9] R. Brežnik, B. Bele, M. Cek, M. Manoilov. *Anketno raziskovanje in analiza podatkov z excelovimi vrtilnimi tabelami*, Ljubljana, 2011.

- [10] G. Melenik, G. Meszaros, J. Bach. *Acceptance test engineering guide*, Microsoft Corporation, 2009, str. 93–97.
- [11] TableAdapter Overview. Dostopno na:
[http://msdn.microsoft.com/enUS/library/bz9tthwx\(v=vs.80\).aspx](http://msdn.microsoft.com/enUS/library/bz9tthwx(v=vs.80).aspx),
Marec 2013
- [12] S. Ambler. *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*, New York, 2002.
- [13] D. Vavpotič. *Opredelitev izhodišč za merjenje in izboljševanje tehnične učinkovitosti in socialne sprejetosti metodologij razvoja programske opreme: doktorska disertacija*, Ljubljana [D. Vavpotič], 2006,
<http://eprints.fri.uni-lj.si/716/>.