

UNIVERZA V NOVI GORICI
POSLOVNO-TEHNIŠKA FAKULTETA

**RAZVOJ SPLETNE PROGRAMSKE OPREME ZA
SPREMLJANJE RAČUNALNIŠKIH PROJEKTOV**

DIPLOMSKO DELO

Sašo Špacapan

Mentor: doc. dr. Bogdan Filipič

Nova Gorica, 2008

ZAHVALA

Za posredovano znanje in koristne informacije pri nastajanju diplomskega dela se zahvaljujem sodelavcem v podjetju Gold Club d.o.o. iz Sežane. Posebej pa se zahvaljujem mentorju doc. dr. Bogdanu Filipiču za spodbudo, nasvete in pregled celotnega dela.

IZVLEČEK

V diplomski nalogi sta predstavljena zasnova in razvoj spletne programske opreme za spremljanje računalniških projektov v podjetju Gold Club d.o.o. v Sežani. Delo je obsegalo načrtovanje in razvoj podatkovne zbirke na podatkovnem strežniku Microsoft SQL Server in implementacijo spletne programske opreme z razvojnim orodjem Microsoft Visual Studio. Razvita programska oprema temelji na spletnem strežniku Microsoft Internet Information Services in tehnologiji ASP.NET, ima pregleden uporabniški vmesnik in omogoča večuporabniško delovanje. Uporabljajo jo projektni vodje in delavci na projektih, z njeno vpeljavo pa sta se izboljšali učinkovitost in preglednost izvajanja projektov.

ABSTRACT

This diploma thesis presents the design and development of a project tracking web-based software for the Gold Club d.o.o. company in Sežana, Slovenia. The work included database design and development on the Microsoft SQL Server database system, and implementation of the web application in the Microsoft Visual Studio environment. The developed software is based on the Microsoft Internet Information Services web server and ASP.NET technology. It features a friendly user interface and a multi-user operation capability. It is used by project managers and project participants, and improves the efficiency and transparency of project execution in the company.

KLJUČNE BESEDE

spletna programska oprema, relacijska podatkovna zbirka, spremljanje računalniških projektov, uporabniški vmesnik, programski jezik C#

KEY WORDS

web-based software, relational database, computer projects tracking, user interface, C# programming language

KAZALO

1. UVOD.....	1
1.1. Problematika spremljanja projektov.....	1
1.2. Izbrana programska orodja.....	2
1.3. Pričakovani rezultati.....	3
1.4. Pregled vsebine.....	3
2. ANALIZA STANJA.....	5
2.1. Podjetje Gold Club d.o.o.	5
2.2. Informacijski sistem podjetja.....	5
3. UPORABLJENA METODOLOGIJA IN PROGRAMSKA ORODJA.....	7
3.1. Relacijske zbirke podatkov.....	7
3.2. Sistem za upravljanje podatkovnih zbirk Microsoft SQL Server.....	8
3.3. Spletni strežnik Microsoft Internet Information Services.....	9
3.4. Razvojno okolje Microsoft Visual Studio.....	10
4. NAČRTOVANJE PODATKOVNE ZBIRKE.....	15
4.1. Diagram toka podatkov.....	16
4.2. Normalizacija podatkov.....	18
4.3. Fizični entitetno-relacijski podatkovni model.....	20
5. RAZVOJ SPLETNE PROGRAMSKE OPREME.....	25
5.1. Podatkovna zbirka.....	25
5.1.1. Nastavitev SQL strežnika.....	25

5.1.2. Razvoj podatkovne zbirke	26
5.2. Spletni strežnik Microsoft Internet Information Services	27
5.2.1. Nastavitev spletnega strežnika.....	27
5.3. Uporabniški vmesnik.....	29
5.3.1. Prijavna stran	29
5.3.2. Glavna stran	31
5.3.3. Meni <i>Projects</i>	33
5.3.4. Meni <i>Tasks</i>	34
5.3.5. Meni <i>WorkTime</i>	35
5.3.6. Meni <i>Gantt</i>	35
6. VPELJAVA RAZVITE PROGRAMSKE OPREME V UPORABO.....	37
6.1. Testiranje programske opreme	37
6.2. Priprava na delo	38
6.3. Učinki programa in izkušnje uporabnikov	38
7. ZAKLJUČEK	41
8. LITERATURA	43

KAZALO SLIK

Slika 1: Delovno okolje SQL Server Management Studia	9
Slika 2: Delovno okolje Microsoft Visual Studio 2005.....	11
Slika 3: Razvojno okolje Microsoft Visual Studio s pogledom na kodo.....	12
Slika 4: Lastnosti programskega gradnika v oknu Properties.....	13
Slika 5: Gradniki v razvojem okolju Microsoft Visual Studio	13
Slika 6: Diagram toka podatkov v sistemu za spremljanje projektov.....	17
Slika 7: Primer normalizacije podatkov.....	19
Slika 8: Fizični entitetno-relacijski model	22
Slika 9: Tabela <i>tasks_users</i> s podatki o delovnem času	23
Slika 10: Okno za definiranje uporabnikov SQL strežnika	26
Slika 11: Primer SQL procedure.....	27
Slika 12: Konfiguracijsko okno ASP.NET	28
Slika 13: Prijavna stran sistema za spremljanje računalniških projektov	30
Slika 14: Stran za definiranje novega uporabnika	30
Slika 15: Stran za pridobitev pozabljenega gesla	31
Slika 16: Glavna stran za vodje projektov	32
Slika 17: Glavna stran za navadne uporabnike.....	33
Slika 18: Meni za upravljanje s projekti	34
Slika 19: Meni za upravljanje z nalogami	35
Slika 20: Meni za upravljanje z delovnim časom.....	36

Slika 21: Zaslonska slika gantograma 36

1. UVOD

Diplomska naloga obsega zasnovo in razvoj spletne programske opreme za spremljanje računalniških projektov v podjetju Gold Club d.o.o. v Sežani. Programska oprema bo služila predvsem projektnim vodjem v podjetju, ki do sedaj niso imeli programske podpore za spremljanje računalniških projektov. Izhajali bomo iz ugotovitve, da sodobno podjetje potrebuje tako programsko rešitev razvito s sodobnimi tehnologijami in orodji. V našem primeru so to spletne tehnologije.

Na tržišču že obstajajo nekatere programske rešitve za spremljanje projektov: Microsoft Project, Easy Projects .NET podjetja Logic Software, Task Tracker podjetja Agito d.o.o. in druge. Za razvoj lastne rešitve smo se odločili predvsem zaradi specifičnih zahtev pri vodenju projektov v podjetju in ker naj bi bila rešitev čim enostavnejša za uporabo.

Ker do sedaj ni obstajala baza podatkov za shranjevanje podatkov o projektih, bo potrebno podatke o njih na novo vnesti v bazo podatkov in jih nato sproti dopolnjevati. V ta namen smo izbrali zmožljiv sistem za upravljanje baz podatkov Microsoft SQL Server, ki omogoča večuporabniško delovanje. Za razvoj spletne programske opreme za spremljanje projektov bomo uporabili razvojno okolje Microsoft Visual Studio 2005 in tehnologijo ASP.NET, programska oprema pa se bo izvajala na spletnem strežniškem sistemu Microsoft IIS (Internet Information Services).

Glavni cilj diplomske naloge je zasnovati in razviti kakovostno programsko podporo za spremljanje računalniških projektov v podjetju Gold Club d.o.o.

1.1. Problematika spremljanja projektov

Kot računalniški projekti se v podjetju Gold Club d.o.o. obravnavajo vsi projekti v razvojnih oddelkih, to je v oddelku strojne opreme in oddelku programske opreme. Posamezen projekt je razdeljen na manjše strukture – naloge, vsak projekt pa lahko izvaja več delavcev.

Glavni projektni vodja je zadolžen za nadzor in spremljanje vseh projektov. Vsak razvojni oddelek ima tudi vodjo oddelka, ki je istočasno tudi projektni vodja in

spremlja projekte, ki se odvijajo v njegovem oddelku. Na vsakemu projektu dela določeno število ljudi, ki so lahko programerji, elektroniki, testerji ipd. Projektni vodja oddelka določi trajanje projekta, naloge, ki spadajo k projektu, koliko časa bo posamezna naloga trajala in kdo oziroma koliko ljudi bo na projektu delalo. S tem, ko tako opredeli projekt, se smatra, da je tudi zadolžen za vodenje projekta. Posameznik ne more izvajati dveh nalog istočasno, lahko pa zaradi dinamike in količine dela več ljudi dela na isti nalogi ali na istem projektu. Vodja lahko spremlja stanje in potek projekta ter nalog na njem. Spremlja lahko zadolžitve posameznika in njegovo obremenjenost z delom. Tako lahko vidi, kdaj je posameznik brez dela (ima čas) in mu posledično lahko dodeli novo zadolžitvev.

1.2. Izbrana programska orodja

Razvita programska oprema bo namenjena uporabi v okolju spletnega strežniškega sistema Microsoft IIS (Internet Information Services), pri njenem razvoju pa bomo uporabili sistem za upravljanje podatkovnih baz Microsoft SQL Server, razvojno okolje Microsoft Visual Studio 2005 in spletno tehnologijo ASP.NET.

Microsoft IIS je zmogljiv in zanesljiv spletni strežnik. Omogoča hiter in enostaven razvoj spletne programske opreme in spletnih strani. Za gostovanje spletne programske opreme na tem strežniku so se v podjetju odločili predvsem zaradi združljivosti z operacijskim sistemom Windows.

Microsoft SQL Server je eden izmed zmogljivejših sistemov za upravljanje z relacijskimi bazami podatkov. Uporablja se v podjetništvu za male in srednje velike zbirke podatkov, v zadnjem obdobju pa so ga prilagodili tudi za velike zbirke podatkov. Njegov primarni jezik in njegova največja prednost je poizvedovalni jezik (query language) T-SQL (Transact-SQL). Za ta sistem so se v podjetju odločili zaradi njegove zmogljivosti in zato, ker ga uporabljajo tudi v drugih programskih rešitvah.

Microsoft Visual Studio je razvojno orodje podjetja Microsoft, namenjeno programerjem. Vsebuje integrirano razvojno okolje, ki omogoča razvoj samostojnih programov, spletnih strani, spletnih aplikacij in storitev, ki se izvajajo na računalnikih z Microsoftovo podporo .NET Framework. V podjetju so ga prav tako

izbrali zaradi zmogljivosti in predvsem zato, ker z njim razvijajo tudi ostalo programsko opremo.

1.3. Pričakovani rezultati

Spletna programska oprema za spremljanje računalniških projektov bo omogočala odpiranje novih projektov, dodelitev nalog in dodelitev delavcev na projekte, vnašanje delovnega časa ter sledenje projektom in nalogam. Ker bo delovala na sistemu za upravljanje z relacijskimi bazami podatkov Microsoft SQL Server in bo to spletna programska oprema na osnovi tehnologije ASP.NET na spletnem strežniku IIS, bo v začetni fazi omogočala večuporabniško delovanje in dostop do spletnega programa kjerkoli v podjetju, kasneje pa tudi kjerkoli na spletu.

1.4. Pregled vsebine

V drugem poglavju bomo predstavili podjetje Gold Club d.o.o. in njegov sedanji informacijski sistem.

V tretjem poglavju bomo opisali, kakšno metodologijo in katera programska orodja smo uporabili pri razvoju programske opreme za spremljanje računalniških projektov. Razložili bomo, kaj je relacijska podatkovna zbirka, spoznali bomo sistem za upravljanje podatkovnih zbirk Microsoft SQL Server, spletni strežnik Microsoft Internet Information Services in razvojno okolje Microsoft Visual Studio.

Načrtovanje podatkovne zbirke bomo opisali v četrtem poglavju. Načrtovanje podatkovne zbirke obsega korake od načrtovanja toka podatkov in normalizacije podatkov do izdelave fizičnega entitetno-relacijskega podatkovnega modela. Z diagramom toka podatkov bomo prikazali proces spremljanja projektov v podjetju. Na izbranem zapisu bomo opisali normalizacijo podatkov. Na osnovi normaliziranih podatkov pa bomo predstavili izdelavo fizičnega entitetno-relacijskega podatkovnega modela.

Peto poglavje je razdeljeno na tri dele. V prvem delu bomo opisali potek nastavitve podatkovnega strežnika Microsoft SQL Server in nato še razvoj podatkovne zbirke. V drugem delu bomo opisali, kako ustrezno nastaviti spletni strežnik Microsoft

Internet Information Services. V tretjem delu pa se bomo posvetili razvoju uporabniškega vmesnika z meniji in gantogrami.

V šestem poglavju bomo navedli nekaj podatkov o razviti programski opremi. Opisali bomo potek testiranja in priprave na delo s programsko opremo. Predstavili bomo tudi njene učinke in prve izkušnje uporabnikov v podjetju Gold Club.

V zaključku bomo povzeli potek razvoja spletne programske opreme za spremljanje računalniških projektov in nakazali njen nadaljnji razvoj.

2. ANALIZA STANJA

Za lažje razumevanje bodočega delovanja spletne programske opreme za spremljanje računalniških projektov in njen enostavnejši razvoj bomo поблиže spoznali podjetje Gold Club d.o.o. in njegov informacijski sistem.

2.1. Podjetje Gold Club d.o.o.

Podjetje Gold Club d.o.o. izdeluje elektronske igralne avtomate, specializiralo pa se je za izdelavo avtomatiziranih inteligentnih elektronskih rulet, ki so jih razvili z lastnim znanjem že leta 1997 in jih od takrat nenehno dograjujejo, med uporabniki pa so vedno bolj priljubljene. Njihovi proizvodi so izdelani iz okolju prijaznih materialov, odstotek okvar je pa je zelo majhen. Gold Club je uspešen tudi na področju prirejanja iger na srečo.

Proizvodno mrežo vzpostavljajo v Rusiji in Južni Ameriki, s čimer so pridobili konkurenčno prednost na trgih Peruja, Kolumbije, Venezuele, Ekvadorja, Mehike in Brazilije. Na domačem trgu prodajo le okoli odstotek svojih izdelkov.

Največjo rast je podjetje doživelo leta 2003, ko je proizvodnjo v primerjavi z letom prej povečalo kar za 70 odstotkov. V letu 2004 so izdelali 735 igralnih avtomatov, kar je za desetino več kot leto prej. Leta 2005 pa so jih zaradi spremenjene zakonodaje v Rusiji naredili 579.

Gold Club zaposluje 88 ljudi, sodelujejo z več kot 60 dobavitelji iz Slovenije in tujine. Največ prodajo v Avstrijo, Ukrajino, Srbijo, Makedonijo, Nemčijo, Češko, Slovaško, Veliko Britanijo, Irsko, Madžarsko in Hrvaško.

2.2. Informacijski sistem podjetja

Podobnega informacijskega sistema, kot je bil razvit v okviru tega diplomskega dela, v podjetju prej niso uporabljali ali imeli, zato je to za podjetje Gold Club d.o.o. novost in velika pridobitev za organizacijo in vodenje projektov ter projektnih nalog. Nova programska oprema prinaša stalno dosegljivost pregleda stanja projektov, česar v podjetju doslej ni bilo. Pričakujemo, da bo z njeno redno uporabo projektno vodenje postalo učinkovitejše in preglednejše.

Ne glede na to, da na področju spremljanja projektov informacijske podpore do sedaj ni bilo, pa na drugih področjih nekatere informacijske rešitve v podjetju že obstajajo. Tako so informacijsko podprte poslovne funkcije kadrovske evidence, računovodstva, potnih nalogov, sprejema blaga, plač, osnovnih sredstev ipd. Zanje podjetje uporablja enovito programsko opremo, kar omogoča enostavnejše in preglednejše izvajanje poslovnih funkcij.

3. UPORABLJENA METODOLOGIJA IN PROGRAMSKA ORODJA

Spletna programska oprema za spremljanje računalniških projektov, ki jo bomo poimenovali GCProjectManager, bo temeljila na spletnem strežniku in relacijskem podatkovnem modelu. Za razvoj spletne programske podpore za spremljanje računalniških projektov smo uporabili spletni strežnik Microsoft IIS (Internet Information Services), sistem za upravljanje podatkovnih zbirk Microsoft SQL Server in razvojno orodje Microsoft Visual Studio. Microsoft IIS je zanesljiv spletni strežnik za gostovanje spletnih aplikacij in spletnih strani, Microsoft SQL Server pa je zmogljiv strežnik za zbirke podatkov. Microsoft Visual Studio je učinkovito razvojno orodje za razvoj programske opreme v okolju Windows. Opis posameznih orodij sledi v nadaljevanju poglavja.

3.1. Relacijske zbirke podatkov

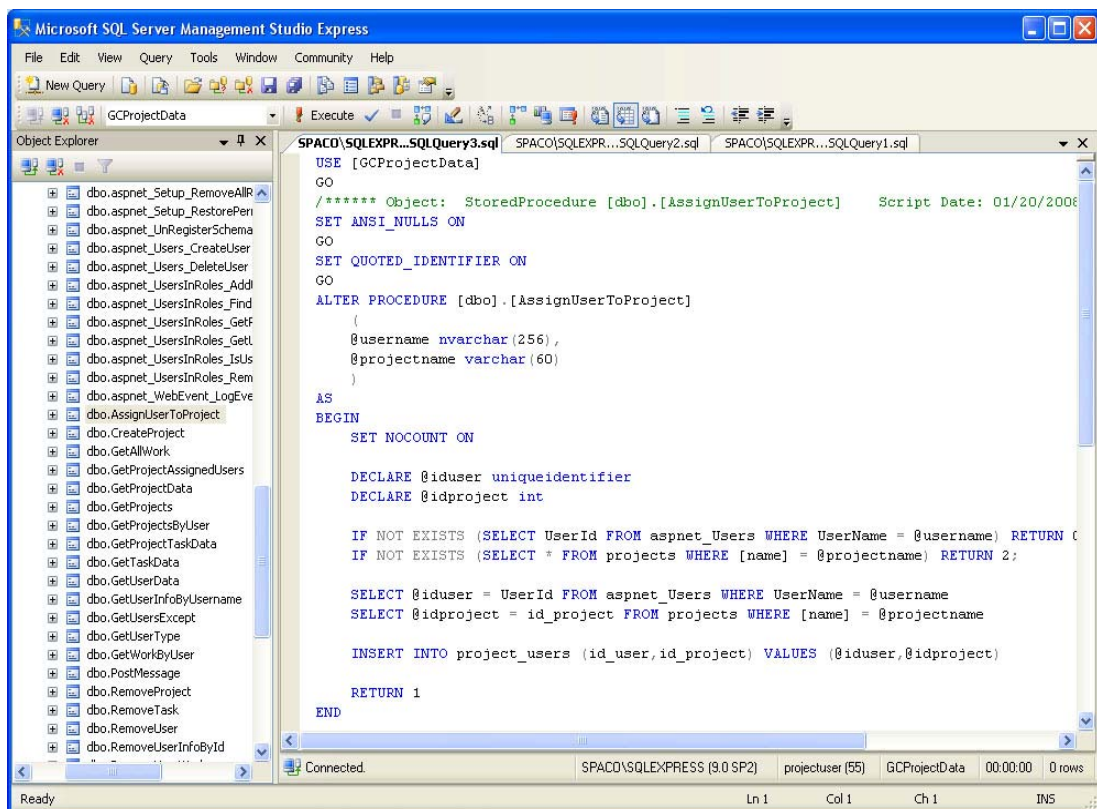
Relacijska zbirka podatkov je vrsta zbirke podatkov, v kateri so podatki predstavljeni v dvodimenzionalnih tabelah. Tabela je osnovna enota v relacijski zbirki in je podobna tabeli, kot jo poznamo iz orodij za delo s preglednicami, vendar s to razliko, da so podatki organizirani po pravilih relacijske teorije, torej v obliki tabel, stolpcev in vrstic (zapisov), opremljenih s ključi, tabele pa so med seboj v odnosih. Zapisi v tabelah so prek posebnih polj med seboj povezani. V relacijskih zbirkah imajo vsi zapisi v posamezni tabeli enaka polja (stolpce). Določena izmed njih so tako imenovana kazalna polja ali ključi (primarni in tuji), s katerimi informacije ene tabele povežemo z informacijami v drugi (Drobne, Podobnikar, 2007).

Vsaka tabela v relacijski zbirki mora imeti primarni ključ ali indeks. To je atribut ali skupina atributov, katerih vrednosti morajo biti edinstvene za vsako vrstico v tabeli. Primarni ključ predstavlja identifikator objekta. Tuji ključ v tabeli pa predstavlja nek primarni ključ v drugi tabeli. Če se glavni atributi (ključi) v različnih tabelah ujemajo, govorimo o relacijskem spajanju podatkovnih tabel. Uporabniki dostopajo do vsebine tabel s poizvedbami v t.i. povpraševalnih jezikih SQL (Oppel, 2004).

3.2. Sistem za upravljanje podatkovnih zbirk Microsoft SQL Server

Microsoft SQL Server je zmogljiv sistem za upravljanje z relacijskimi podatkovnimi zbirkami. Njegova največja prednost je poizvedovalni jezik (query language) SQL. Microsoft SQL Server sicer uporablja izpeljanko jezika T-SQL (Transact-SQL), to je implementacija SQL-92 (tretja izdaja povpraševalnega jezika SQL) z mnogimi razširitvami (Thomas in drugi, 2006). Jezik T-SQL za razliko od ostalih povpraševalnih jezikov vsebuje mnoge razširitve za delo s shranjenimi procedurami in funkcijami. Zadnja izdaja sistema za upravljanje podatkovnih zbirk SQL Server iz leta 2005 podpira tudi možnost povezovanja različnih odjemalcev preko protokola spletnih storitev SAOP (Simple Object Access Protocol), ki omogoča odjemalcem, ki ne uporabljajo okolja Windows, da komunicirajo z SQL strežnikom.

SQL strežnik lahko upravljamo na dva načina. Prvi je preko ukazne vrstice z dodeljenimi ukazi, drugi način pa je z orodjem, ki se imenuje SQL Server Management Studio (slika 1). S tem orodjem lahko urejamo in nadziramo vse komponente in nastavitve strežnika. Lahko urejamo podatke o uporabnikih strežnika in njihovih pravicah, upravljamo s podatkovnimi zbirkami in njihovimi strukturami (tabelami, povezavami itd.), arhiviramo podatke, pišemo proceduralne SQL stavke itd. Orodje vsebuje tudi urejevalnik skript (povpraševalnega jezika SQL) in grafična orodja za delo z različnimi strežniškimi objekti. Ena glavnih značilnosti orodja je Object Explorer (raziskovalec objektov), ki omogoča pregledovanje strežniških objektov in izvajanje različnih akcij nad njimi.



Slika 1: Delovno okolje SQL Server Management Studia

3.3. Spletni strežnik Microsoft Internet Information Services

Spletni strežnik Microsoft IIS (Internet Information Services) je skupina internetnih spletnih strežniških sistemov za okolje Windows. Sestavljajo ga različne vrste strežniških spletnih protokolov. Nekateri pomembnejši so: HTTP strežnik (Hypertext Trasfer Protocol), FTP strežnik (File Transfer Protocol) in SMTP strežnik (Simple Mail Trasfer Protocol) (Internet Information Services, 2007). HTTP je komunikacijski protokol, ki služi za prenos dokumentov na svetovnem spletu. Je protokol vrste zahteva-odziv (request-response) med odjemalci in strežniki. Odjemalec, npr. spletni brskalnik, pošlje določeno HTTP zahtevo strežniku, od katerega nato dobi ustrezen odziv, npr. besedilo, sliko itd. FTP protokol omogoča prenos datotek med računalniki preko interneta ali lokalnega omrežja. Največ se uporablja za prenos in izmenjavo datotek preko omrežij, ki temeljijo na protokolu TCP/IP (internet in intranet). SMTP pa je preprost tekstovni protokol za prenos elektronske pošte po spletu in je dandanes *de facto* standard. Za naše potrebe bomo uporabljali HTTP strežnik, ki bo odgovoren za sprejemanje HTTP zahtev

odjemalcev, v našem primeru spletnih brskalnikov, in zagotavljanje HTTP odgovorov z zahtevanimi podatki. V večini primerov bodo to podatki z SQL strežnika ter razna besedila, slike ipd.

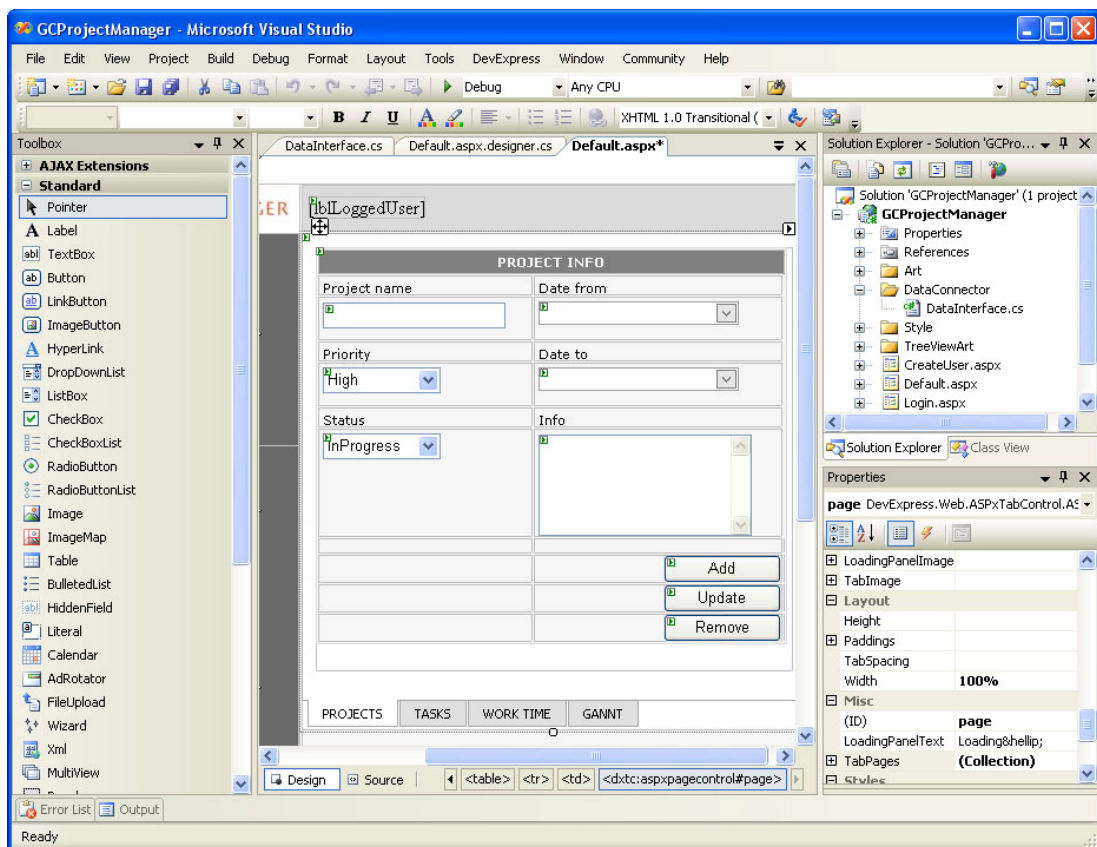
Pomembna prednost strežnika IIS je podpora več vrstam overovitev odjemalcev. To pomeni preverjanja pristnosti odjemalca, ki dostopa do spletne programske opreme ali spletne strani. IIS pozna štiri vrste overovitev: Windows authentication (ki se deli na Basic access authentication, Digest access authentication, Integrated Windows authentication), Forms authentication, .NET passport authentication in Anonymous access authentication (anonimen dostop).

Windows authentication je način overovitve, kjer spletni strežnik preverja pristnost odjemalca glede na njegov okenski uporabniški račun. Forms authentication je način overovitve, kjer spletni strežnik preko spletne strani pridobi informacije o odjemalcu. Zato pa je treba posebej napisati programsko kodo za preverjanje pristnosti odjemalca. Način .NET passport authentication preverja pristnost odjemalca glede na uporabniški račun v okviru storitve Microsoft Passport. Odjemalec si mora definirati svoj račun na ustrezni spletni strani.

V naši spletni programski opremi smo uporabili tip overovitve Forms authentication, zato se bomo v nalogi osredotočili le nanj.

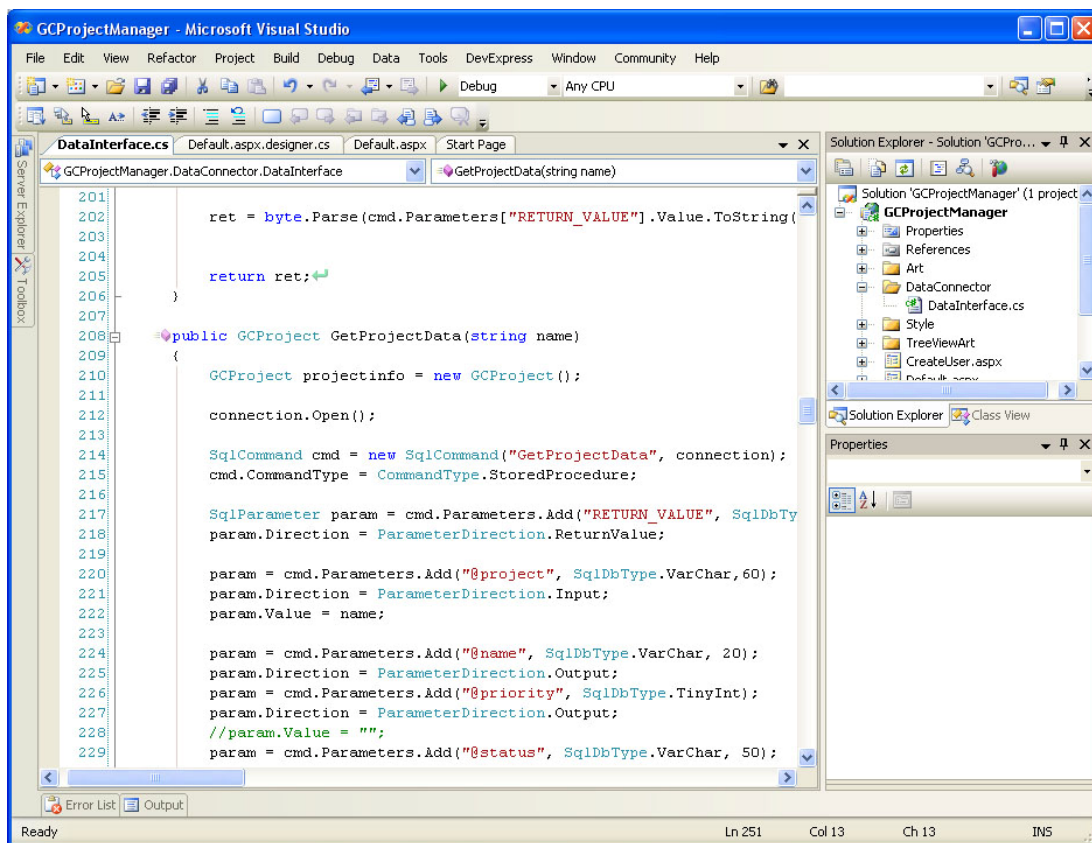
3.4. Razvojno okolje Microsoft Visual Studio

Microsoft Visual Studio je integrirano razvojno orodje (slika 2), ki programerjem omogoča razvoj samostojne programske opreme za operacijske sisteme Windows, ASP.NET spletnih strani in ASP.NET spletne programske opreme, ki se lahko izvaja na kateremkoli računalniku z okoljem .NET Framework, kar je tudi glavna prednost tega razvojnega orodja (Microsoft Visual Studio, 2007). V večini primerov je to okolje Windows. Večina sodobne programske opreme, ki deluje v okolju Windows, je razvita z razvojnim orodjem Microsoft Visual Studio.



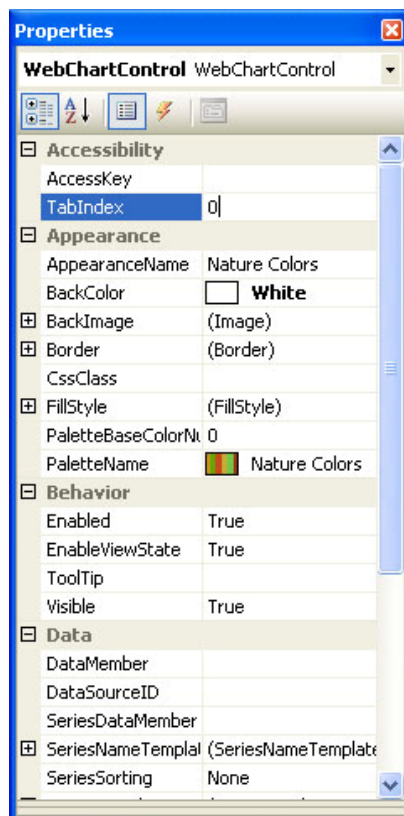
Slika 2: Delovno okolje Microsoft Visual Studio 2005

Kot smo že omenili, je Visual Studio integrirano razvojno okolje, ki omogoča programiranje v več programskih jeziki. To so Visual Basic, Visual C++, Visual C#, Visual J# in ASP.NET. Za razvoj naše spletne programske opreme smo uporabili programski jezik Visual C# in spletno tehnologijo ASP.NET (slika 3). Visual C# je objektno orientiran programski jezik, ki ga je razvilo podjetje Microsoft kot del okolja .NET Framework. Sintaksa jezika je proceduralna, objektno orientirana in temelji na programskem jeziku C++. Vključuje tudi nekatere lastnosti drugih programskih jezikov (največ Delphi in Java) s poudarkom na enostavnosti jezika. ASP.NET pa je spletna tehnologija, ki programerjem omogoča razvijanje dinamičnih spletnih strani, spletne programske opreme in spletnih storitev. Tako kot Visual C# je tudi ASP.NET del okolja .NET Framework.

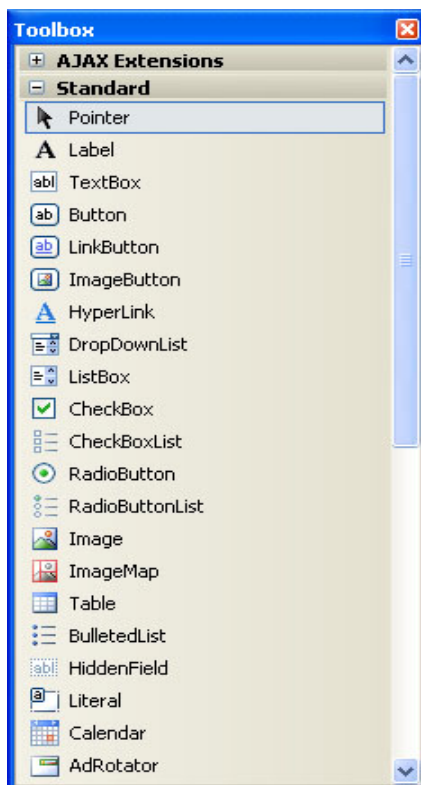


Slika 3: Razvojno okolje Microsoft Visual Studio s pogledom na kodo

Razvojno okolje omogoča tudi vizualno postavljanje in premikanje programskih gradnikov kot tudi spreminjanje njihovih lastnosti. To omogoča okno *Properties* (slika 4). Številni gradniki so že vgrajeni (slika 5), lahko pa razvijemo lastne in dodamo številne druge že razvite. Teh je res veliko. Gradniki se delijo na spletne gradnike ASP.NET in gradnike .NET Windows. Gradniki ASP.NET so uporabni za spletne strani in spletno programsko opremo. Gradniki .NET (dot NET) so uporabni za razvoj samostojne programske opreme v okolju Windows. Obstajajo pa tudi gradniki, ki jih lahko uporabljamo za obe vrsti programske opreme. Za naše potrebe smo uporabljali spletne gradnike ASP.NET.



Slika 4: Lastnosti programskega gradnika v oknu Properties



Slika 5: Gradniki v razvojnem okolju Microsoft Visual Studio

Za dostop do baze podatkov smo uporabili tehnologijo ADO.NET, ki je tudi del okolja .NET Framework. To je skupek knjižnic za upravljanje s podatki, shranjenimi v relacijskih bazah podatkov in dostopanje do njih. Gre za zbirko podatkov, ki jo upravljamo s sistemom SQL Server.

4. NAČRTOVANJE PODATKOVNE ZBIRKE

Načrtovanje podatkovne zbirke je zelo pomembna faza, ki obsega več korakov. Pri načrtovanju moramo vedeti, katere podatke bomo potrebovali in kakšne povezave bodo obstajale med temi podatki.

Skupaj z vodji projektov smo dorekli, katere podatke potrebujemo in kakšne so povezave med njimi, saj oni najboljše poznajo področje vodenja projektov. S tem smo pridobili na kakovosti načrtovanja podatkovne baze. Kakovostno načrtovanje podatkovne baze nas namreč pripelje do usklajenih, popolnejših in natančnejših podatkov v podatkovni zbirki. Če je podatkovna baza površno načrtovana, je iz nje težje pridobiti določene povezave med podatki in obstaja nevarnost, da bo iskanje dalo netočne informacije. Netočni podatki so najpogostejša posledica neustreznega načrtovanja podatkovne zbirke (Hernandez, 2003).

Čas, ki ga uporabimo za načrtovanje podatkovne zbirke, je dobro investiran čas. Kakovostno načrtovanje nam prihrani čas na dolgi rok, ker nam ni treba stalno krpati na hitro in slabo načrtovane strukture. Rezultati kakovostnega načrtovanja podatkovne zbirke so (Powell, 2006):

- Strukturo podatkovne zbirke lahko brez večjih težav spreminjamo, ne da bi škodljivo vplivali na ostale podatke.
- Podatke z lahkoto spreminjamo. Spremembe vrednosti, ki jih vnašamo v izbrano polje v tabeli, ne vplivajo na vrednosti polj iz drugih tabel.
- Informacije iz podatkovne baze pridobimo z lahkoto. Povpraševalne stavke napišemo z lahkoto, ker so tabele dobro zgrajene in relacije med njimi pravilne.
- Uporabniško programsko opremo je enostavneje razvijati, ker se lahko bolj posvetimo programiranju, saj se ni treba ukvarjati s težavami, ki izvirajo iz slabo načrtovane podatkovne zbirke.

4.1. Diagram toka podatkov

»Diagrami toka podatkov so mrežno-plastna (nivojska) predstavitev podatkovnih tokov v sistemu in njihovih transformacij« (Strukturna analiza, 2007). Z drugimi besedami: diagram toka podatkov je grafična ponazoritev toka podatkov skozi informacijski sistem. Lahko se uporablja tudi za ponazoritev procesiranja podatkov.

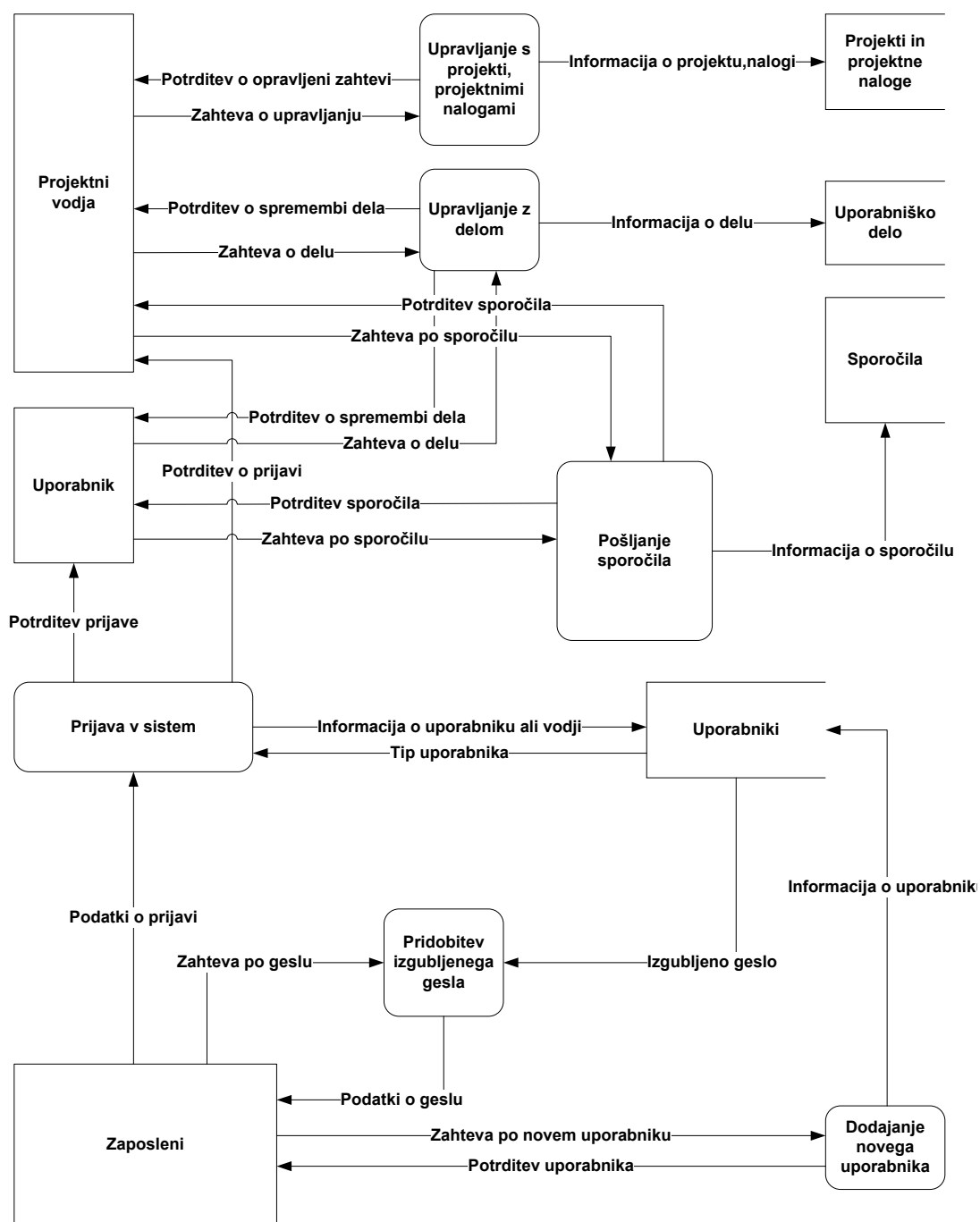
Na začetku poglavja smo omenili, da moramo za načrtovanje podatkovne zbirke poznati podatke, ki jih bomo potrebovali. Omenili smo tudi, da smo te podatke določili v razgovorih z vodji projektov. To smo naredili tako, da smo skupaj analizirali proces spremljanja projektov in tako opredelili vse podatke, ki jih potrebujemo. Na podlagi analize in definiranih podatkov smo izdelali diagram toka podatkov (slika 6).

Diagram toka podatkov predstavlja množico procesov, zunanjih entitet in podatkovnih shramb ter podatkovnih tokov kot povezav med njimi. Glavna procesa sta proces prijave uporabnikov in proces spremljanja projektov. Proces prijave se deli na tri podprocese: prijavo v sistem, registracijo uporabnika in pridobitev izgubljenega gesla. Proces spremljanja projektov se tudi deli na tri podprocese: upravljanje projektov in projektnih nalog, upravljanje z delom ter pošiljanje sporočil.

Vsi zaposleni v podjetju imajo uporabniško ime in geslo, s katerima se prijavljajo v sistem za spremljanje projektov. Obstajata dva tipa uporabniških imen: prvi tip uporabljajo vodje projektov, drugega pa navadni uporabniki. Vsak zaposleni se lahko prijavi kot navadni uporabnik. Če pozabi geslo, ga lahko pridobi preko procesa pridobitve gesla.

Ob vsaki prijavi sistem ugotovi, kateri uporabniški profil ima zaposleni, ki se je prijavil. Navadnemu uporabniku sistem omogoča samo upravljanje s podatki, ki se nanašajo na njegovo lastno delo. To pomeni, da lahko uporabnik vnaša podatke ali evidentira podatke le projektih, na katere je bil dodeljen. Če se prijavi zaposleni s profilom vodje projektov, mu sistem omogoča spremljanje vseh projektov in nalog za vse uporabnike. Lahko dodaja, spreminja in odstranjuje projekte in naloge ter dodeljuje navadne uporabnike na projekte. Ostane pa mu tudi pravica za upravljanje z lastnim delovnim časom, tako da ugotovimo, da imata vodja projektov in navadni

uporabnik skupen proces upravljanja z delom. To je tudi logično, saj vsak zaposleni ne glede na svoj profil lahko upravlja z lastnim delovnim časom. Vsi zaposleni (vodje projektov in navadni uporabniki) pa lahko pišejo sporočila, ki jih nato pošljejo na poljubno nalogo. Sporočila lahko berejo vsi sodelujoči na projektu, ki vsebuje nalogo, kateri je bilo sporočilo poslano.



Slika 6: Diagram toka podatkov v sistemu za spremljanje projektov

4.2. Normalizacija podatkov

Normalizacija podatkov je proces, s katerim učinkovito organiziramo podatke v podatkovni zbirki. Z normalizacijo želimo doseči dvoje. Izločiti želimo odvečne in ponavljajoče se podatke (npr. hranjenje enakih podatkov v več kot eni tabeli) ter zagotoviti, da so relacije med podatki logično zastavljene. Oboje pa nam zagotavlja, da bodo podatki zavzeli manj prostora v podatkovni zbirki in bodo logično urejeni (Chapple, 2007). Relacije, ki vsebujejo odvečne podatke, lahko povzročijo anomalije pri spreminjanju podatkov. Poznamo anomalije pri dodajanju, brisanju in popravljanju podatkov (Sevčnikar, 2007).

V okviru normalizacijskega postopka poznamo šest t.i. normalnih oblik tabel oziroma relacij. V večini primerov je zadosti, da normalizacijo izvedemo le do tretje normalne oblike, ker ta običajno ne potrebuje več modifikacij, da bi ustrezala višjim normalnim oblikam. Zaželeno je, da že v prvem poskusu načrtujemo tabelo oziroma relacijo v tretji normalni obliki. Tabela oziroma relacija je v prvi normalni obliki, če nima ponavljajočih atributov, ima definiran primarni ključ in določene funkcionalne odvisnosti. V drugi normalni obliki je relacija, ki je v prvi normalni obliki in noben atribut, ki ni del ključa, ni funkcionalno odvisen le od dela primarnega ključa, temveč tudi od celotnega ključa. Relacija je v tretji normalni obliki, če je v drugi normalni obliki in če med atributi, ki niso del primarnega ključa, ni funkcionalnih odvisnosti (Database normalization, 2007).

V našem primeru smo se normalizacije lotili tako, da smo podatkovno bazo že na začetku poskušali načrtovati v tretji normalni obliki, kar nam je v dobršni meri tudi uspelo. S podatki iz diagrama toka podatkov smo si pomagali pri načrtovanju podatkovne zbirke. Na sliki 7 vidimo normalizirane podatke zapisov projektov, projektnih nalog in uporabnikov.

Nenormalizirani podatki

ZAPIS
Project
Task1
Name
Priority
Status
Start
Finish
Info
Task2
Type
date_from
date_to
task_user
Date
time_from
time_to
info

Normalizirani podatki

PROJECT
id_project
name
priority
status
start
finish
info

TASK
id_task
id_project
name
status
priority
type
date_from
date_to
Info

TASK_USERS
id_task_users
id_task
id_user
name
date
time_from
time_to
info

Slika 7: Primer normalizacije podatkov

Pri spremljanju računalniških projektov je projekt osnovni objekt. Projekt se odvija v nekem določenem času. Za izpeljavo projekta so potrebne naloge, ki jih uporabniki oz. zaposleni opravljajo v času projekta. Za vsako nalogo zaposleni delavec porabi določen čas.

4.3. Fizični entitetno-relacijski podatkovni model

Fizični entitetno-relacijski model izhaja in logičnega entitetno-relacijskega modela. Medtem ko pri logičnem modelu govorimo le o pridobivanju uporabnikovih podatkovnih zahtev in spreminjanju teh zahtev v nek model, pri fizičnem modelu govorimo o načrtovanju podatkovne baze. Nanaša se na fizične podatke in vsebuje podatke o povezavah tabel, indeksih, podatkovnih tipih ipd. Je specifičen glede na podatkovni strežnik. To pomeni, da se objekti, definirani med fizičnim modeliranjem, razlikujejo glede na uporabljen podatkovni strežniki. Na primer, večina relacijskih podatkovnih sistemov (strežnikov) ima svoje specifične lastnosti, kot sta na primer prikaz podatkovnih tipov in način hranjenja podatkov, čeprav so osnovni podatkovni tipi v različnih izvedbah podatkovnih strežnikov večinoma enaki (Logical Versus Physical Database Modeling, 2007).

Pri načrtovanju fizičnega entitetno-relacijskega modela je potrebno opraviti naslednje naloge:

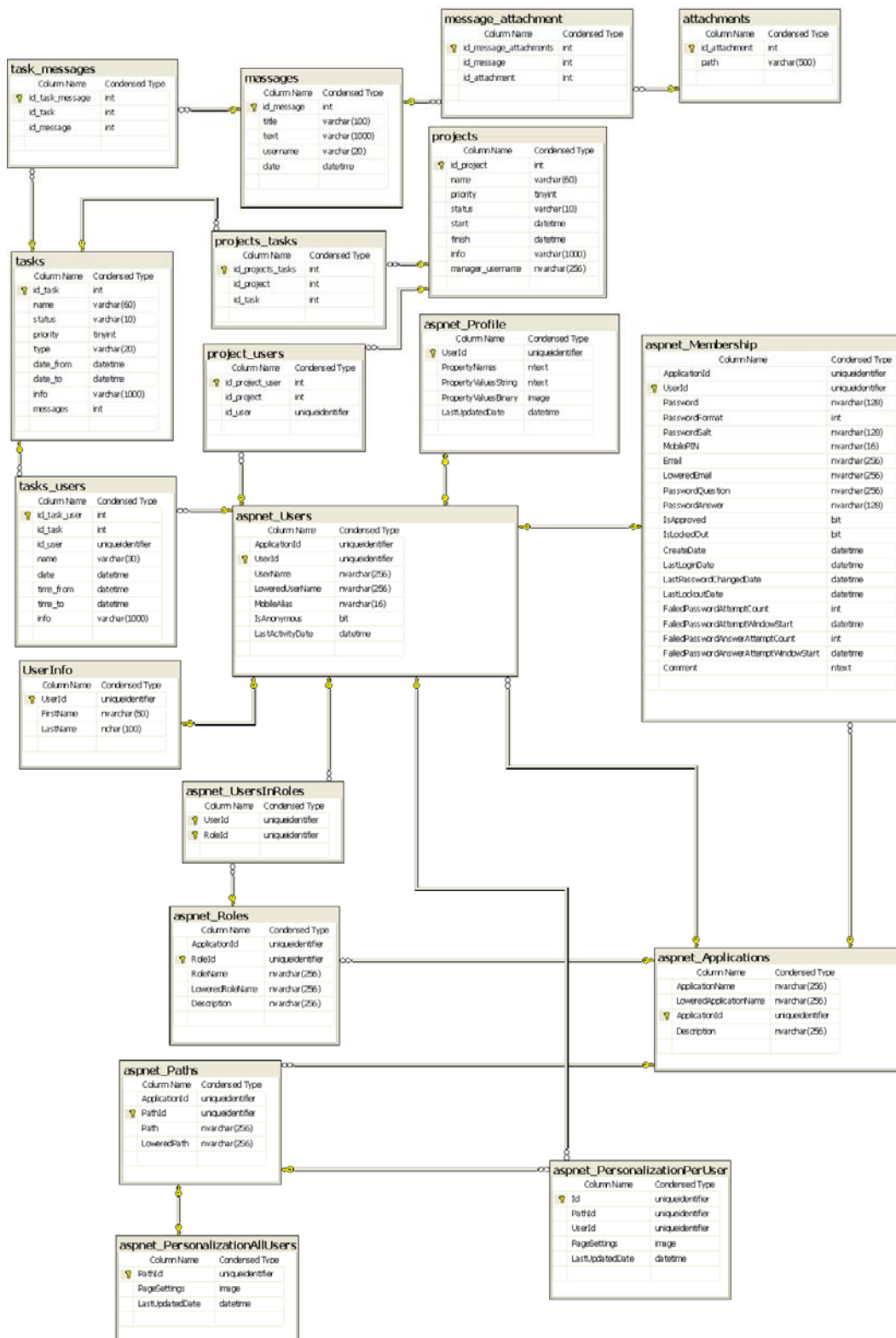
- določitev tabel,
- normalizacijo tabel,
- določitev stolpcev-atributov tabel,
- določitev SQL procedur, ki bodo upravljale s podatki,
- določitev relacij med tabelami,
- dodelitev ključev (primarnih in zunanjih).

Pogosto je v navadi, da pred fizičnim modelom izdelamo tudi logični entitetno-relacijski model, ki se od fizičnega razlikuje po tem, da ne vsebuje podatkov o fizičnih lastnostih atributov, vendar v našem primeru tega nismo naredili. To pa zato, ker smo podatkovni strežnik in njegove lastnosti poznali že od vsega začetka in smo se lahko pri izdelavi fizičnega modela osredotočili na posebnosti podatkovnega strežnika SQL Server.

Fizični entitetno-relacijski model podatkovne baze za spremljanje računalniških projektov smo izdelali kar z orodjem Microsoft SQL Management Studio, kar ima tudi veliko prednost. Ta je v tem, da se pri izdelavi fizično entitetno-relacijskega modela samodejno izdela tudi celotna baza podatkov (brez SQL procedur in funkcij) z vsemi tabelami, atributi in povezavami, ki jih določimo v modelu. Na ta način si močno olajšamo razvoj podatkovne baze.

V fizičnem entitetno-relacijskem modelu (slika 8) je razvidno, da celotna podatkovna zbirka sloni na tabeli *aspnet_users*, ki združuje podatke o uporabnikih sistema za spremljanje računalniških projektov. Tabela je preko tabele *project_users* povezana s tabelo *projects*, ki vsebuje podatke o projektih, preko tabele *tasks_users* pa je povezana s tabelo *tasks*, v kateri so podatki o projektnih nalogah. Tabela *tasks_users* ne služi le kot povezovalna tabela, ampak so v njej podatki o delovnem času (slika 9).

Na sliki 9 v prvem stolpcu tabele *tasks_users* vidimo ključ, kar pomeni, da je polje z imenom *id_task_user* primarni ključ tabele. Drugi stolpec označuje ime polja v tabeli, tretji pa podatkovni tip polja.



Slika 8: Fizični entitetno-relacijski model

tasks_users	
Column Name	Condensed Type
 id_task_user	int
id_task	int
id_user	uniqueidentifier
name	varchar(30)
date	datetime
time_from	datetime
time_to	datetime
info	varchar(1000)

Slika 9: Tabela *tasks_users* s podatki o delovnem času

5. RAZVOJ SPLETNE PROGRAMSKE OPREME

Za potrebe razvoja spletne programske opreme za spremljanje računalniških projektov GCProjectManager smo najprej ustrezno nastavili SQL strežnik. Podatkovno zbirko, tabele in relacije med njimi smo dobili že s tem, ko smo izdelali fizični entitetno-relacijski model. Naslednji korak je bil razvoj SQL procedur za upravljanje s podatkovno zbirko. To je jedro naše spletne programske opreme, ki skrbi za upravljanje s podatki o podatkovni zbirki. Upravljanje obsega dodajanje, brisanje in popravljanje podatkov. V naslednjem koraku smo ustrezno nastavili spletni strežnik, da smo tako lahko začeli razvijati spletno programsko opremo. Sledila je izgradnja uporabniškega vmesnika in njegova povezava z jedrom. V celotnem procesu razvoja smo uporabili metodologijo in programska orodja, opisana v tretjem poglavju. V nadaljevanju podrobneje predstavljamo korake razvoja.

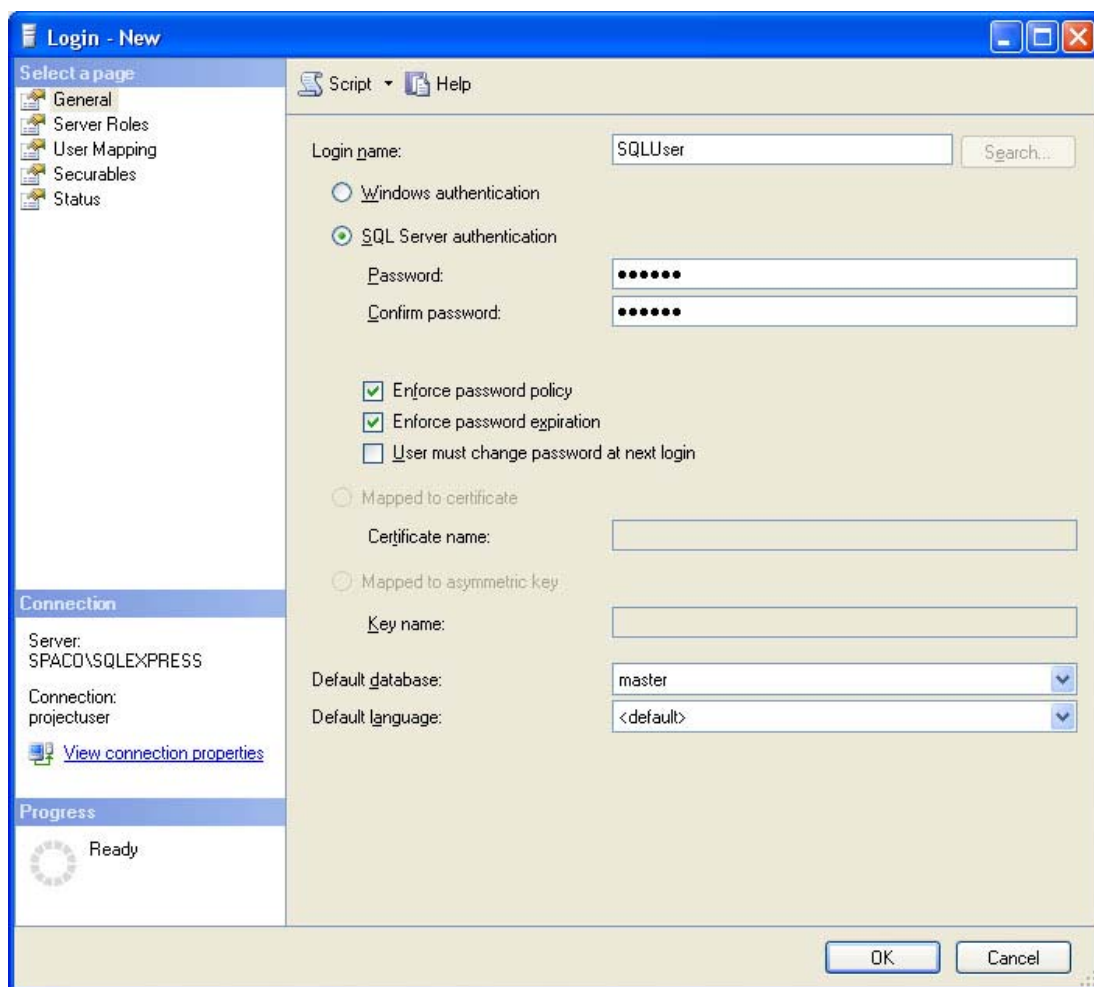
5.1. Podatkovna zbirka

Logično za razvoj podatkovne zbirke bi bilo, da bi najprej načrtovali entitetno-relacijski model in šele nato izdelali podatkovno zbirko, vendar smo v našem primeru naredili obratno. Pred izdelavo fizičnega entitetno-relacijskega modela smo izdelali prazno podatkovno zbirko in nastavili SQL strežnik, saj drugače ne bi bila mogoča avtomatska izdelava tabel in povezav med njimi. V tej nalogi je sicer vrstni red razvoja opisan v logični smeri, zaradi lažjega razumevanja problematike.

5.1.1. Nastavitev SQL strežnika

Z orodjem SQL Management Studio smo definirali uporabnika, s katerim bomo dostopali do podatkovne zbirke za upravljanje na strežniku (slika 10). Uporabniku smo določili uporabniško ime, geslo in pravice, ki jih ima pri dostopu do podatkovne zbirke. Zato bo pri prijavi v SQL strežnik potekala identifikacija uporabnika preko uporabniškega imena in gesla. Obstaja še način, kjer za dostop do SQL strežnika preverjamo uporabnika, s katerim dostopamo do mrežne domene (Windows domain authentication), vendar smo v našem primeru izbrali način preverjanja uporabniškega imena in gesla (SQL server authentication), ker bomo do programske opreme dostopali tudi izven lokalnega omrežja. Tako definirani uporabnik bo lahko dostopal le do zbirke podatkov za spremljanje računalniških projektov. Pravic za dostop do

drugih zbirke podatkov mu nismo dodelili, ima pa vse pravice za upravljanje z zbirko podatkov za spremljanje z računalniških projektov.



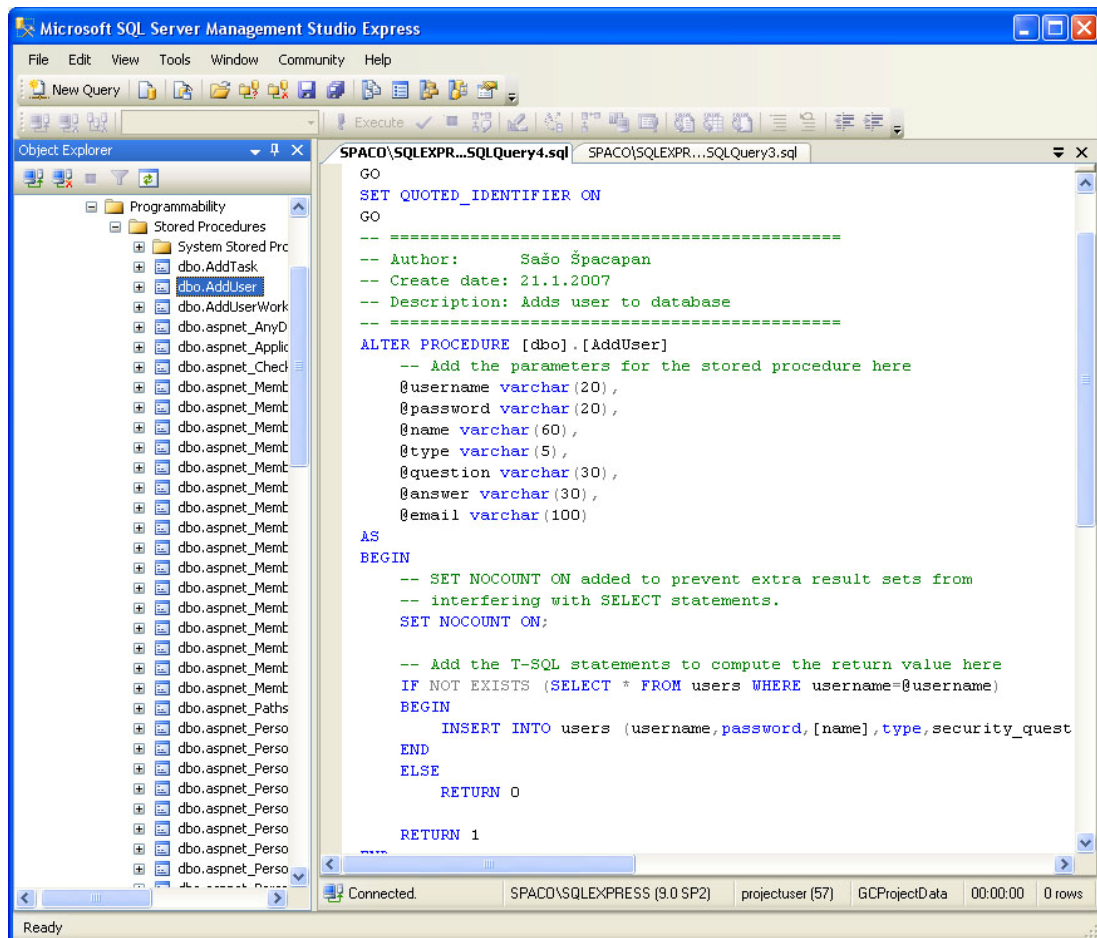
Slika 10: Okno za definiranje uporabnikov SQL strežnika

5.1.2. Razvoj podatkovne zbirke

Velik del razvoja podatkovne zbirke za spremljanje računalniških projektov smo opravili že s samo izdelavo fizičnega entitetno-relacijskega modela z orodjem SQL Management Studio, ker smo istočasno z njegovo izdelavo avtomatsko izdelali tudi vse potrebne tabele, njihove attribute in povezave med tabelami.

Za upravljanje s podatki v podatkovni zbirki smo v nadaljevanju razvili skupek SQL procedur, s katerimi lahko izvajamo popoln nadzor nad podatki v podatkovni zbirki (slika 11). Z njimi lahko pregledujemo, dodajamo, brišemo in popravljamo podatke,

ki se nanašajo na projekte, projektne naloge, delovni čas, uporabnike, projektne vodje, sporočila in priponke.



Slika 11: Primer SQL procedure

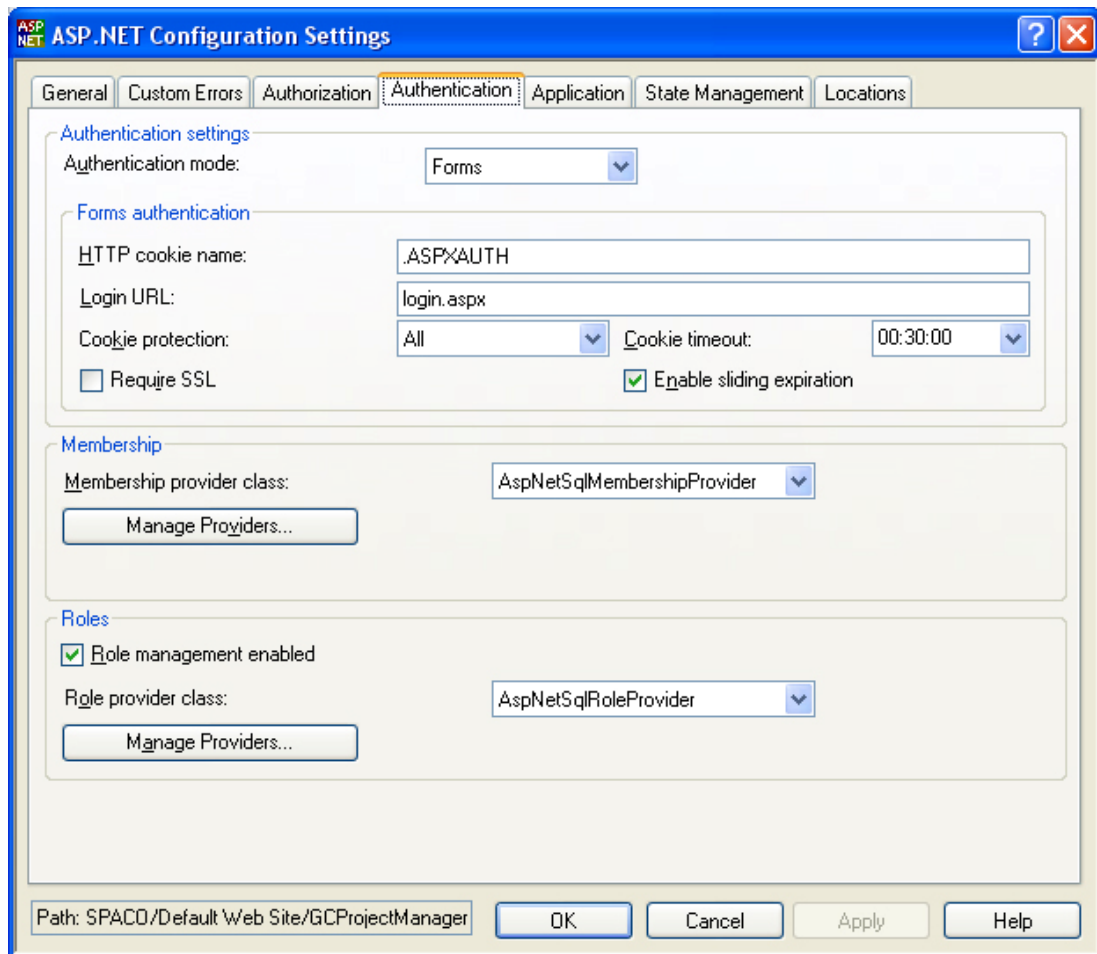
5.2. Spletni strežnik Microsoft Internet Information Services

5.2.1. Nastavitev spletnega strežnika

Ko smo zaključili razvoj podatkovne zbirke, smo lahko začeli z nastavitvami spletnega strežnika. Spletni strežnik nastavimo z okenskim orodjem ASP.NET Configuration Settings (slika 12), ki ga najdemo v nadzorni plošči spletnega strežnika.

Spletni strežnik smo nastavili tako, da imajo do njega dostop samo uporabniki s svojim uporabniškim imenom in geslom preko spletne strani za prijavo v sistem za spremljanje računalniških projektov. Zaradi varnosti smo nastavili tudi omejitve za

navadne uporabnike, tako da lahko samo berejo podatke, ne pa tudi pišejo na strežnik.



Slika 12: Konfiguracijsko okno ASP.NET

Našteli in opisali bomo nekaj pomembnejših nastavitev. Ena od njih je nastavev Authentication mode oziroma način overovitve uporabnikov, ki dostopajo do spletne programske opreme. Nastavili smo jo na Forms, kar pomeni, da se bodo uporabniki v spletno programsko opremo prijavljali preko prijavnih strani, kjer bodo vpisali uporabniško ime in geslo. To pomeni, da bodo lahko dostopali do spletnega strežnika le tisti, ki bodo imeli pravico dostopa. Naslednja nastavev je Login URL, kjer vpišemo ime spletne datoteke, v kateri se nahaja prijavno okno. Nastavili smo tudi parameter, ki vsebuje povezavo do SQL strežnika. SQL strežnik smo povezali s spletnim strežnikom.

5.3. Uporabniški vmesnik

Uporabniški vmesnik smo razvili z orodjem Visual Studio, ki je zmogljivo orodje za vizualno programiranje v okolju Windows. S tem orodjem lahko sproti preverjamo in zaganjamo nastajajoči programski izdelek, kar nam omogoča, da ga lahko tudi sproti testiramo. Uporabniški vmesnik temelji na spletnih tehnologijah HTML in ASP.NET, zato se tudi izvaja v spletnih brskalnikih. Sestavlja ga več spletnih strani, od katerih vsaka služi svojemu namenu. Tako obsega spletne strani za prijavo v sistem, definicijo novega uporabnika, pridobitev izgubljenega gesla in glavno spletno stran, preko katere so dostopne vse možnosti za spremljanje računalniških projektov.

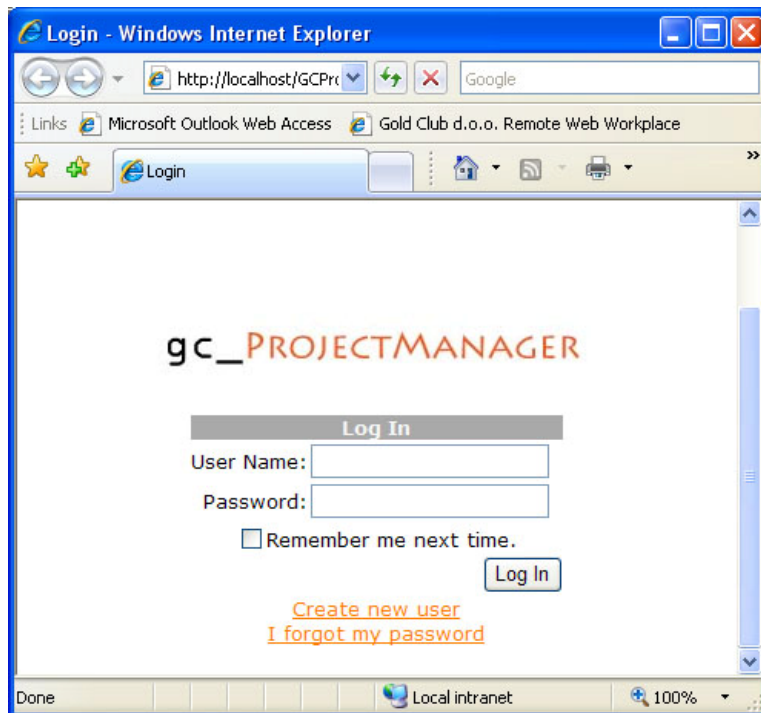
Uporabniški vmesnik smo zasnovali tako, da je za uporabnika pregleden in omogoča hiter in enostaven dostop do želenih podatkov.

5.3.1. Prijavna stran

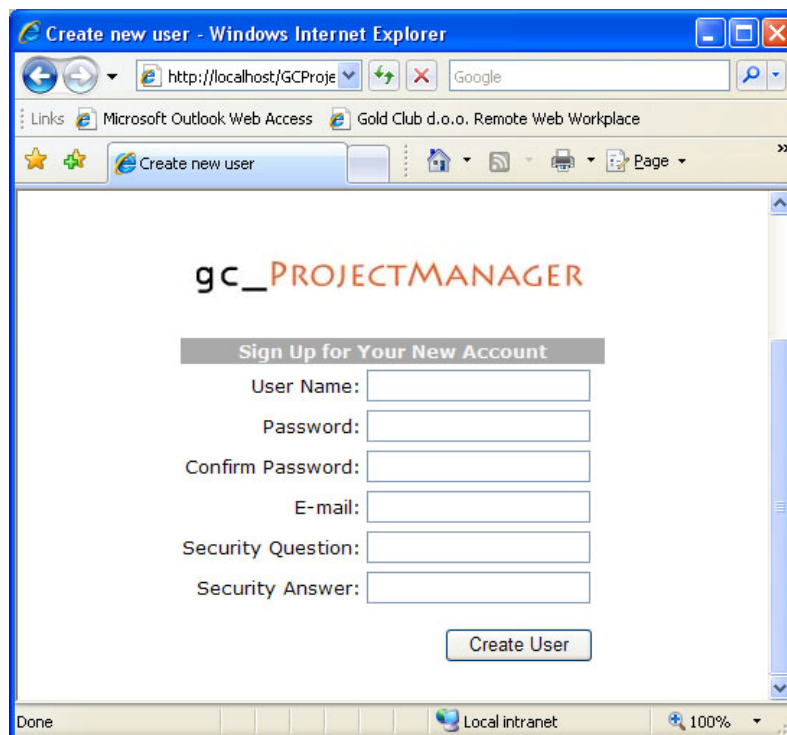
Prijavna stran omogoča prijavo uporabnikov v sistem za spremljanje računalniških projektov. Za prijavo potrebujemo uporabniško ime in geslo (slika 13). Prijavimo se lahko kot vodja projektov (manager) ali kot navadni uporabnik (user). Če napačno vtipkamo uporabniško ime ali geslo, nas sistem na to opozori.

Če uporabniškega imena nimamo, ga za navadne uporabnike lahko določimo preko povezave *Create new user* (slika 13). Podatki, ki jih moramo pri tem vnesti, so poleg uporabniškega imena in gesla še elektronski naslov ter varnostno vprašanje in odgovor (slika 14). Ko je ta postopek končan, se lahko novi uporabnik prijavi v sistem.

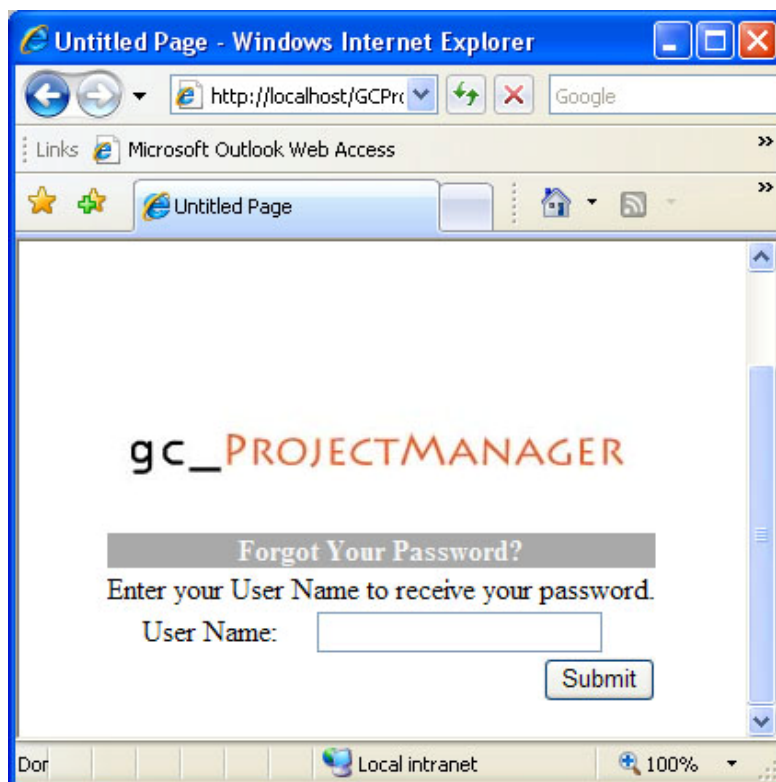
V primeru, da uporabniško ime in geslo že imamo, vendar smo ju pozabili, lahko preko povezave *I forgot my password* pozabljenega gesla pridobimo nazaj. To nam omogoča stran za pridobitev izgubljenih gesel (slika 15). Vedeti moramo uporabniško ime in ko ga vnesemo, geslo prejmemo po elektronski pošti.



Slika 13: Prijavna stran sistema za spremljanje računalniških projektov



Slika 14: Stran za definiranje novega uporabnika



Slika 15: Stran za pridobitev pozabljenega gesla

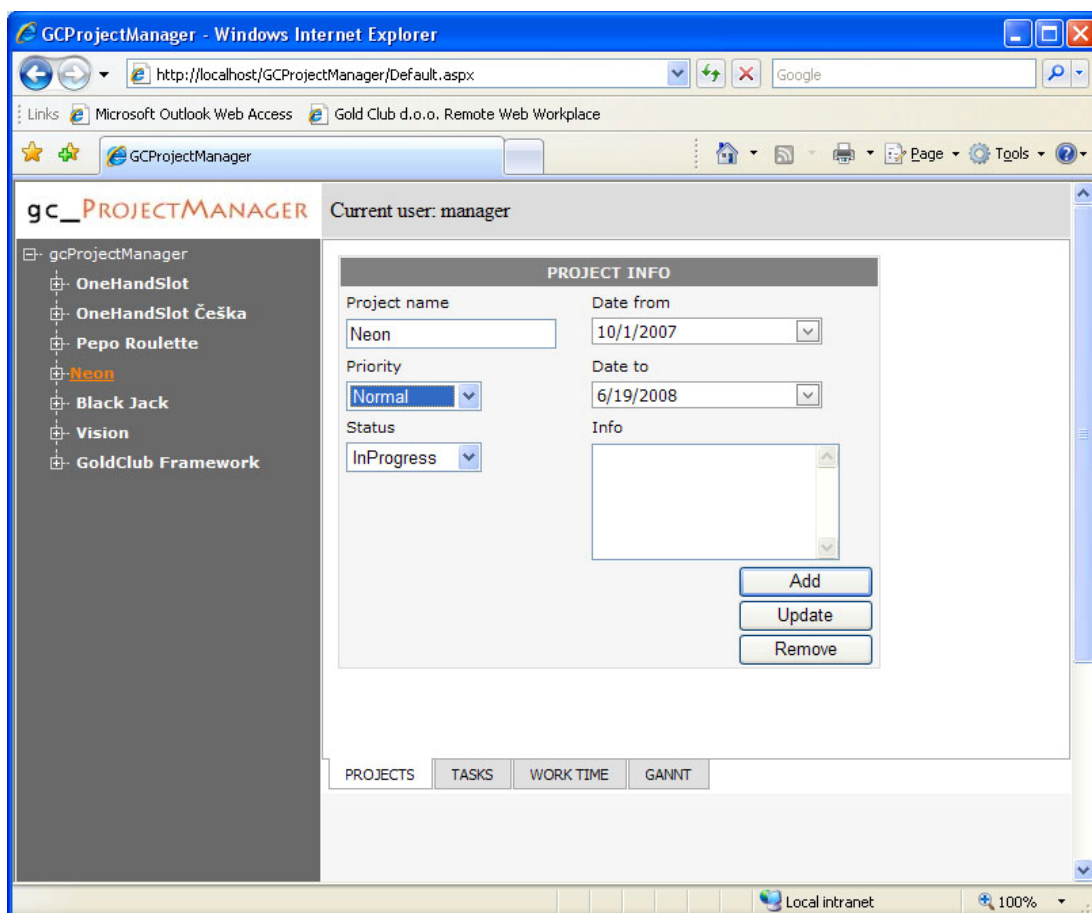
Na opisani način lahko pridobivajo uporabniška imena in pozabljena gesla navadni uporabniki. To so vsi zaposleni, razen vodij projektov. Uporabniško ime tipa vodja projektov lahko definira le glavni vodja projektov.

5.3.2. Glavna stran

Glavno stran sestavljajo drevo na levi strani (slika 16), kjer je seznam projektov in projektnih nalog, logotip in statusna vrstica zgoraj ter osrednji prostor, kjer so našteje glavne možnosti upravljanja s programom.

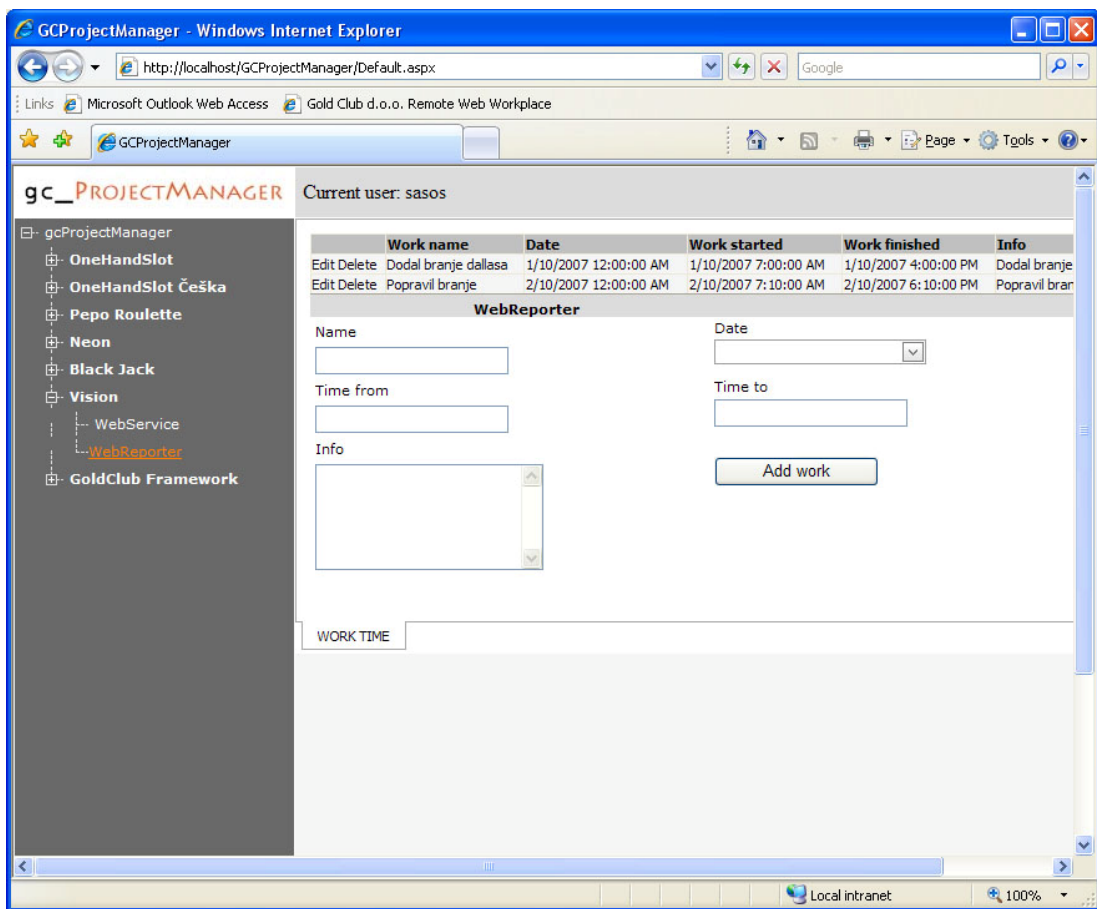
Obstajata dve različici glavne strani. Ena se prikaže ob prijavi vodje projektov (slika 16), druga pa ob prijavi navadnega uporabnika (slika 17). Glavna stran je osrednji nadzorni center sistema za spremljanje računalniških projektov. Razlika med njenima različicama je v številu funkcij, ki jih lahko uporabnik opravlja. Vodja projektov ima popoln nadzor nad svojimi projekti. Lahko dodaja nove, briše in popravlja obstoječe, enako moč ima tudi nad projektnimi nalogami, pregleduje lahko zasedenost delavcev

na projektih in si beleži svoj delovni čas. Navadni uporabnik ima le eno možnost, to je beleženje na projektih porabljenega delovnega časa.



Slika 16: Glavna stran za vodje projektov

Uporabniški vmesnik je deloma odvisen od tipa uporabnika. Funkcije so razdeljene po zavihkih in ti se prikazujejo glede na tip uporabnika. Vodja projektov ima vse možnosti pri spremljanju projektov in zato vidi vse zavihke (slika 16). Ti so: *Projects* (projekti), *Tasks* (naloge), *WorkTime* (delovni čas) in *Gantt* (gantogrami). Medtem ko ima navadni uporabnik vpogled le v zavihek *WorkTime* (slika 17).

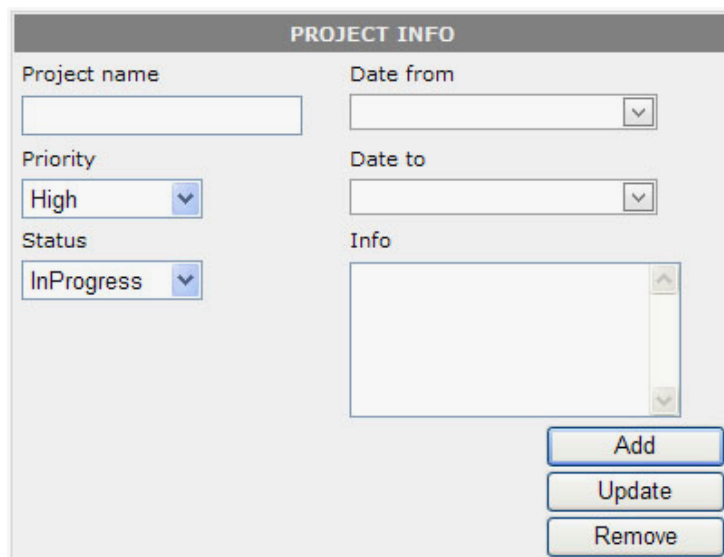


Slika 17: Glavna stran za navadne uporabnike

5.3.3. Meni *Projects*

Do menija *Projects* (projekti) imajo dostop le projektni vodje, ker imajo edini pravico in nalogo upravljati s projekti, kar jim meni tudi omogoča. To pomeni dodajanje novih projektov in brisanje ali popravljanje podatkov o obstoječih projektih (slika 18). Če želi projektni vodja odpreti nov projekt, mora najprej vnesti podatke o projektu. Potrebni podatki so: ime projekta, prioriteta, stanje projekta, datum začetka, datum konca in, če vodja tako želi, še opis. Prioriteta projekta je lahko visoka, normalna ali nizka. Projekti z visoko prioriteto imajo prednost pred vsemi projekti, ki imajo normalno ali nizko prioriteto. Najmanj pomembni projekti imajo nizko prioriteto. Stanje projekta pomeni, v kakšnem stanju je projekt. Lahko je v teku, v mirovanju, zaprt ali opuščen. Za potrditev projekta na koncu vnašanja pritisnemo gumb *Add* (dodaj). Pri popravljanju podatkov o projektu moramo najprej izbrati projekt v drevesu levo (slika 16), po popravljanju pa pritisnemo gumb *Update*

(popravi). Za brisanje projekta izberemo projekt v drevesu in pritisnemo gumb *Remove* (odstrani).



The image shows a software dialog box titled "PROJECT INFO". It has a light gray background and a dark gray header. The dialog contains several input fields and buttons. On the left side, there is a text input field for "Project name", a dropdown menu for "Priority" with "High" selected, and another dropdown menu for "Status" with "InProgress" selected. On the right side, there are two date selection fields: "Date from" and "Date to", both with dropdown arrows. Below these is a large text area labeled "Info" with a vertical scrollbar. At the bottom right of the dialog, there are three stacked buttons: "Add", "Update", and "Remove".

Slika 18: Meni za upravljanje s projekti

5.3.4. Meni *Tasks*

Tako kot meni *Projects* je tudi meni *Tasks* (naloge) dostopen le projektnim vodjem. Podobno kot meni *Projects* tudi meni *Tasks* omogoča projektnim vodjem upravljanje s projektnimi nalogami (slika 19). Lahko jih dodajajo, popravljajo in brišejo. Vsaka projektna naloga pripada enemu projektu. Kadar hoče projektni vodja dodati novo nalogo na projekt, najprej izbere projekt v drevesu ali pa v polju *Project*, ki ga vidimo na sliki 19. Projektni nalogi je treba določiti čas trajanja, ki mora biti v času trajanja celotnega projekta, v nasprotnem primeru nas program opozori na časovno neskladje. Podobno kot projektom lahko tudi nalogam določimo prioriteto in stanje in dodatno še tip naloge. Stanja in prioritete projektnih nalog so enake kot pri projektih. Pomembnejše naloge imajo visoko prioriteto, manj pomembne normalno prioriteto in najmanj pomembne nizko prioriteto. Stanja nalog so: naloga v teku, v mirovanju, zaprta naloga ali opuščena naloga. Tipa naloge sta predlog in navadna naloga. Predlog se v teku naloge lahko spremeni v navadno nalogo.

TASK INFO	
Task name naloga1	Date from 11/27/2007
Priority High	Date to 12/28/2007
Status InProgress	Info test
Type Normal	Project Vision CashBox
<input type="button" value="Add"/> <input type="button" value="Update"/> <input type="button" value="Remove"/>	

Slika 19: Meni za upravljanje z nalogami

5.3.5. Meni *WorkTime*

Meni *WorkTime* (delovni čas) je dostopen obema tipoma uporabnikov, projektnim vodjem in navadnim uporabnikom. Omogoča jim upravljanje z njihovim delovnim časom na izbrani projektni nalogi. Meni sestavljajo: polje s podatki o delovnem času zaposlenega na projektni nalogi in polja za vnos podatkov o novem delovnem času (slika 20). Pri vnosu delovnega časa mora biti datum v mejah trajanja projektne naloge, sicer nam program vnosa ne dovoli in nas na to tudi opozori. Pri vnosu delovnega časa vnesemo še kratek opis dela in trajanje. Pri tem pa obstaja omejitev, da ne moremo opraviti dveh vnosov, ki se jima čas prekriva. To pomeni, da ne moremo opravljati dveh nalog istočasno, ampak le zaporedno eno za drugo.

5.3.6. Meni *Gantt*

Za projektne vodje je pomemben meni *Gantt* (slika 21). Njegov namen je prikazovanje gantogramov. Gantogram je grafična ponazoritev trajanja opravila v nekem časovnem obdobju. Je uporabno orodje za načrtovanje projektov in je v pomoč pri spremljanju poteka projektov.

	Work name	Date	Work started	Work finished	Info
Edit Delete	Dodal branje dallasa	1/10/2007 12:00:00 AM	1/10/2007 7:00:00 AM	1/10/2007 4:00:00 PM	Dodal branje dallas kjuca
Edit Delete	Popravlil branje	2/10/2007 12:00:00 AM	2/10/2007 7:10:00 AM	2/10/2007 6:10:00 PM	Popravlil branje dallas kjuca

WebReporter

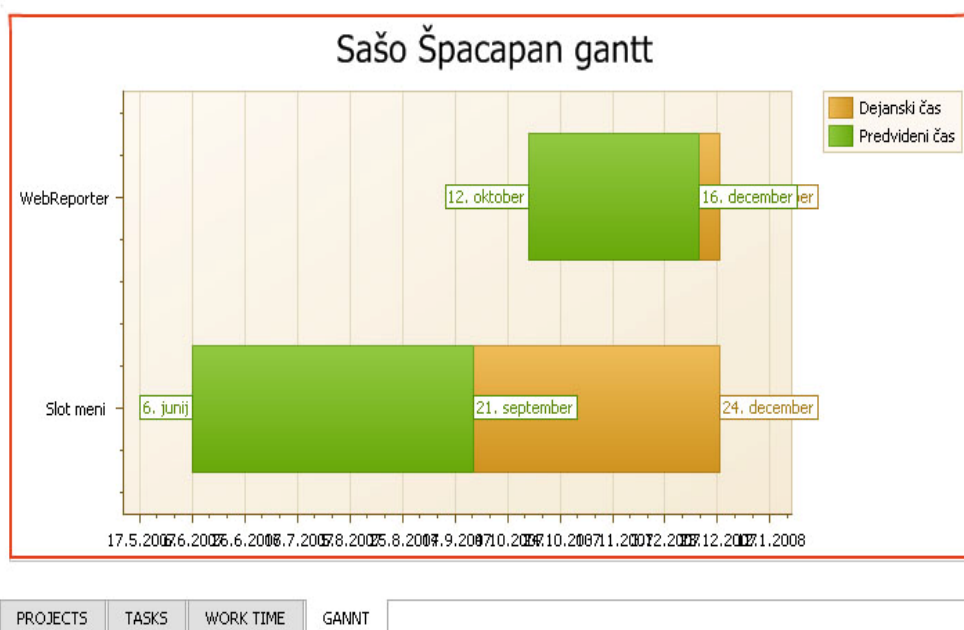
Name Date

Time from Time to

Info

Slika 20: Meni za upravljanje z delovnim časom

Meni *Gantt* omogoča projektnemu vodji, da za vsakega zaposlenega vidi, koliko časa opravlja delo na projektu ali nalogi. Nudi mu vpogled v dejansko trajanje projektov in nalog. Spremlja lahko, za koliko se je prekorščil predvideni čas za posamezne projekte ali naloge.



Slika 21: Zaslonska slika gantograma

6. VPELJAVA RAZVITE PROGRAMSKE OPREME V UPORABO

Spletno programsko opremo GCProjectManager sem približno slab mesec razvijal kot edini razvijalec. Glede na to, da se v času študija nismo poglobljeje spoznali s programerskimi veščinami, sem potrebna znanja za razvoj programske kode pridobil sam, saj je ukvarjanje s programiranjem tudi moj hobi. Razvita programska oprema obsega približno dvajset tisoč vrstic kode v programskem jeziku C#. Velikost programskih datotek je okoli 4 MB.

Razvoju spletne programske opreme za spremljanje računalniških projektov sta sledila njeno testiranje in priprava dokumentacije. Opisali bomo tudi priprave na delo s programsko opremo, izkušnje uporabnikov in učinke nove programske opreme na organizacijo dela v podjetju. Na koncu pa bomo pregledali predloge uporabnikov za izboljšanje programske opreme.

6.1. Testiranje programske opreme

Velik del testiranja je bil opravljen že ob programiranju v okolju Visual Studio. Hiter prevajalnik jezika C# in orodja, kot je razhroščevalnik, so nam omogočili, da smo ključne dele programske opreme testirali že med njenim razvojem. Tako smo lahko sproti preizkušali delovanje uporabniškega vmesnika in vanj vgrajenih funkcij. Na ta način smo odpravili veliko napak.

Ne glede na to, da je bila funkcionalnost programske opreme določena v dogovoru s projektnimi vodji že precej časa pred razvojem, so ti šele nekaj časa po začetku poskusne uporabe programske opreme predlagali nekatere spremembe in izboljšave. To je potrdilo našo domnevo, da šele s praktično uporabo pridemo do nekaterih spoznanj o tem, kaj potrebujemo. Nekaterim predlogom, ki so zahtevali manjše spremembe programske opreme, smo ugodili takoj, tiste, ki zahtevajo večje posege, pa smo prepustili za naslednjo različico programske opreme. Manjše napake, ki so se pokazale pri uporabi, smo sproti popravljali.

Želja sodelavcev z oddelka programske opreme je bila, da bi programska oprema vključevala tudi možnost puščanja sporočil in dodajanja različnih datotek na projekte. Tako bi se vse gradivo nekega projekta zbiralo na enem mestu. To bi zelo

olajšalo spremljanje projektov in tudi njihovo razumevanje. Ta možnost bo vključena v naslednji različici.

6.2. Priprava na delo

Preden so uporabniki začeli uporabljati novo programsko opremo za spremljanje računalniških projektov, smo jim olajšali delo s tem, da smo izdelali kratko dokumentacijo v elektronski obliki o delovanju programske opreme. Dostopna je v mapi za skupno rabo na glavnem strežniku, kjer se že nahaja druga dokumentacija. Dokumentacija je v obliki datoteke PDF in obsega 15 strani.

6.3. Učinki programa in izkušnje uporabnikov

Spletna programska oprema za spremljanje računalniških projektov GCProjectManager je v uporabi od septembra 2007 in deluje na dveh strežnikih. Eden se uporablja kot spletni strežnik, drugi pa kot podatkovni strežnik, to pa zaradi hitrejšega delovanja celotne programske opreme. Programska oprema je trenutno dostopna preko lokalne mreže. Uporabljajo jo projektni vodje in drugi zaposleni, ki delajo v oddelkih programske in strojne opreme. Uporablja jo petnajst zaposlenih, direktor podjetja, trije projektni vodje, dva razvijalca strojne opreme in devet programerjev. S programsko opremo se trenutno spremlja sedem projektov, na katerih sodeluje štirinajst zaposlenih.

Pred vpeljavo programske opreme za spremljanje računalniških projektov v podjetju ni bilo v uporabi nobeno orodje ali pripomoček za spremljanje projektov. Zaradi tega ni bilo natančnega seznama odprtih, zaprtih ali opuščenih projektov. Težko je bilo spremljati projekte in zadolžitve na njih. Največji učinek ima nova programska oprema prav na tem področju. Podatke o projektih in nalogah imamo shranjene v pregledni in hitro dostopni obliki. Poleg tega imamo še dodatne možnosti, kot so spremljanje delovnega časa, porabljenega na določenih projektih in nalogah. Uporabniki so najbolj zadovoljni ravno s tem, da imajo možnost videti, kakšni projekti in naloge se izvajajo in kdo na njih dela.

Uporabniki, ki trenutno uporabljajo programsko opremo za spremljanje računalniških projektov, želijo, da bi bila programska oprema dostopna tudi preko interneta in ne le

v lokalni mreži. S tem bi jim zelo olajšali spremljanje projektov tudi izven podjetja. To jim bomo v prihodnosti tudi omogočili.

7. ZAKLJUČEK

V podjetju Gold Club je zaradi povečane količine dela postal nadzor projektov nepregleden. Velike težave so imeli predvsem vodje projektov, ki jim je nadzor nad projekti in projektnimi nalogami uhajal iz rok, pa tudi sodelavci na projektih, ker niso imeli zadostnega vpogleda v potek dela. Pojavila se je potreba po učinkovitejšem in preglednejšem delu s projekti in projektnimi nalogami.

V pogovoru med razvijalci in projektnimi vodji se je pokazala potreba, da se razvije programska oprema, ki bi podpirala spremljanje projektov in projektnih nalog ter upravljanje z delovnim časom. Po možnosti naj bi to bila spletna programska oprema, dosegljiva kjerkoli in kadarkoli.

S pomočjo zbranih podatkov smo načrtali in razvili podatkovno zbirko. Z orodjem SQL Management Studio smo izdelali fizični entitetno-relacijski model, samodejno pa se je s tem tudi fizično naredila podatkovna zbirka z vsemi tabelami in relacijami. Kasneje smo naredili še skupino SQL procedur za upravljanje s podatki v podatkovni zbirki.

Ker je to spletna programska oprema, smo pred razvojem uporabniškega vmesnika vzpostavili spletni strežnik Microsoft Internet Information Services. Nato smo za interakcijo z uporabniki, ki vključuje vnose, pregledovanje in analize podatkov, razvili uporabniški vmesnik z orodjem Microsoft Visual Studio. Uporabniški vmesnik temelji na tehnologiji ASP.NET.

Pred vpeljavo programske opreme v prakso smo napisali uporabniško dokumentacijo, ki je dostopna vsem uporabnikom. Spletno programsko opremo smo zaradi hitrosti delovanja namestili na dva strežnika. Na enega smo namestili podatkovno bazo, drugega pa smo uporabili kot spletni strežnik, na katerem se nahaja programska oprema. Dostopna je kjerkoli v podjetju vsem zaposlenim, ki imajo uporabniško ime in geslo.

Na začetku so programsko opremo uporabljali le en projektni vodja in trije programerji. Po odpravi nekaterih hroščev so jo začeli uporabljati še ostali. Kot smo omenili v prejšnjem poglavju, jo trenutno uporablja petnajst uporabnikov. Vodje

projektov so zadovoljni s tem, da imajo pregled nad obstoječimi projekti, nalogami in razpoložljivostjo delavcev.

V bližnji prihodnosti bomo na željo uporabnikov omogočili dostop do programske opreme tudi preko spleta, saj je to navsezadnje spletna programska oprema in bi bilo škoda, če bi se uporabljala le v lokalnem omrežju.

Sčasoma se je pokazalo zanimanje za podobno informacijsko podporo tudi na oddelku proizvodnje in servisa. Ker sta podatkovna zbirka in programska oprema zasnovani, da omogočata ne le spremljanje računalniških projektov temveč tudi projektov drugih tipov, bomo računalniško podprto spremljanje projektov postopoma vpeljali tudi na ostalih oddelkih podjetja. Če se bo razvita programska oprema tako izkazala za splošno uporabno, bomo razmišljali tudi o njenem trženju.

8. LITERATURA

Chapple, M. Database normalization basics. Pridobljeno 26. 11. 2007 s svetovnega spleta: <http://databases.about.com/od/specificproducts/a/normalization.htm>

Database normalization. Pridobljeno 29. 11. 2007 s svetovnega spleta: http://en.wikipedia.org/wiki/Database_normalization

Drobne, S., Podobnikar, T. (2007). Osnovni pojmi v geografskih informacijskih sistemih: Relacijska zbirka podatkov. Ljubljana: Fakulteta za gradbeništvo in geodezijo. Pridobljeno 11. 9. 2007 s svetovnega spleta: http://www.fgg.uni-lj.si/~sdrobne/GIS_Pojm/RELACIJSKA%20ZBIRKA%20PODATKOV.htm

Hernandez, J. M. (2003). Database Design for Mere Mortals. 2. izdaja. Boston: Addison Wesley

Internet Information Services. Pridobljeno 21. 8. 2007 s svetovnega spleta: http://en.wikipedia.org/wiki/Internet_Information_Services

Logical Versus Physical Database Modeling. Pridobljeno dne 26. 12. 2007 s svetovnega spleta: <http://www.developer.com/tech/article.php/641521>

Microsoft Visual Studio. Pridobljeno 20. 8. 2007 s svetovnega spleta: http://en.wikipedia.org/wiki/Microsoft_Visual_Studio

Oppel, J., A. (2004). Databases Demystified. New York: McGraw-Hill

Powell, G. (2006). Beginning Database Design. Indianapolis, Indiana: Wiley Publishing

Sevčnikar, A. Normalizacija. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko. Pridobljeno 29. 11. 2007 s svetovnega spleta: <http://lpt.uni-mb.si/lpt/student/pb-tk/files/normalizacija.ppt>

Strukturna analiza. Pridobljeno 6. 12. 2007 s svetovnega spleta: <http://mafalda.uni-mb.si/backup/2002/predstavitve/dtp.html>

Thomas, R., Machanic, A., Skinner, J., Davidson, L., Dewson, R., Narkiewicz, J., Sack, J., Walters, R. (2006). Pro SQL Server 2005. Barkeley: Apress