

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

GAŠPER RUPNIK

RAZVOJ APLIKACIJ ZA OKOLJE SHAREPOINT

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

Mentor:
izr. prof. dr. Viljan Mahnič

Ljubljana, 2011

Št. naloge: 01774/2011

Datum: 01.09.2011



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

Kandidat: **GAŠPER RUPNIK**

Naslov: **RAZVOJ APLIKACIJ ZA OKOLJE SHAREPOINT
DEVELOPING SHAREPOINT APPLICATIONS**

Vrsta naloge: Diplomsko delo univerzitetnega študija


Tematika naloge:

Proučite značilnosti okolja MS SharePoint in predstavite možnosti, ki jih nudi za razvoj aplikacij. Opišite razpoložljiva programska orodja in z njihovo pomočjo realizirajte nekaj spletnih aplikacij za potrebe korporacije Hidria d.d.: spletno aplikacijo za prijavo in obdelavo izboljšav ter spletno aplikacijo za nadzor terminalov.

Mentor:


prof. dr. Viljan Mahnič

Dekan:


prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a **GAŠPER RUPNIK**,

z vpisno številko **63060281**,

sem avtor/-ica diplomskega dela z naslovom:

RAZVOJ APLIKACIJ ZA OKOLJE SHAREPOINT

S svojim podpisom zagotavljam, da:

sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

IZR. PROF. DR. VILJANA MAHNIČA

in somentorstvom (naziv, ime in priimek)

/

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 **7. 12. 2011**

Podpis avtorja/-ice: _____

ZAHVALA

Ob tej priložnosti bi se rad najprej zahvalil mentorju izr. prof. dr. Viljanu Mahničju za posredovano pomoč in nasvete pri izdelavi diplomskega dela.

Najlepša hvala tudi vsem sodelavcem iz poslovne enote HIDRIA IT znotraj korporacije HIDRIA, ki so mi med izdelavo diplomskega dela kakorkoli pomagali, pa seveda tudi oddelku za inovacijsko kulturo znotraj korporacije za zaupan projekt izdelave aplikacije.

Na koncu pa bi se rad zahvalil tudi družini, ki mi je bila v oporo skozi celoten študij. Posebej pa bi se rad zahvalil Blažki za vso njeno pomoč in oporo.

KAZALO VSEBINE

POVZETEK	1
ABSTRACT	2
1 UVOD	3
2 PLATFORMA SHAREPOINT.....	5
2.1 ZGRADBA PLATFORME SHAREPOINT	6
2.2 ORGANIZACIJA STRANI V SHAREPOINTU	7
2.3 CENTRALNA ADMINISTRACIJA.....	7
3 PROGRAMIRANJE V SHAREPOINTU.....	9
3.1 PROGRAMERSKA ORODJA.....	9
3.1.1 Osnovno programiranje preko brskalnika.....	9
3.1.2 Sharepoint Designer 2010.....	10
3.1.3 Visual Studio 2010.....	11
3.1.4 Expression Blend	12
3.2 USTVARJANJE SHAREPOINTOVE PROGRAMSKE REŠITVE Z ORODJEM VISUAL STUDIO 2010.....	13
3.2.1 Sharepoint predloge v Visual Studiu 2010	13
3.2.2 Sandboxed in farmske programske rešitve	15
3.2.3 Preslikovanje map in datotek.....	15
3.2.4 Pakiranje funkcionalnosti	16
3.3 SHAREPOINTOVI OBJEKTNI MODELI	16
3.3.1 SPContext	18
3.3.2 SPFarm.....	18
3.3.3 SPWebApplication	19
3.3.4 SPSite.....	20
3.3.5 SPWeb	21
3.3.6 SPList	22
3.3.7 SPListItem.....	23
3.3.8 SPFile	24
3.3.9 SPField	25
3.3.10 SPQuery.....	26
3.3.11 SPUser	27
3.3.12 SPException	28

3.3.13 SPUtility	28
3.4 SHAREPOINTOVI ODZIVNIKI NA DOGODKE (EVENT RECEIVERS).....	29
3.4.1 Dogodki na SharePoint seznamih	29
3.4.2 Dogodki na elementih SharePoint seznama.....	30
3.4.3 Dogodki na elektronsko pošto.....	30
3.4.4 Dogodki na strani.....	30
3.4.5 Dogodki na delovne tokove	31
3.5 DELO Z ZUNANJIMI PODATKI (BUSINESS DATA).....	31
3.6 SHAREPOINTOVI DELOVNI TOKOVI (WORKFLOWS)	33
3.7 SHAREPOINTOVE SPLETNE STORITVE (WEB SERVICES).....	35
3.8 VARNOST V SHAREPOINTU	37
3.9 POVEZOVANJE S PAMETNIMI TELEFONI.....	39
4 SHAREPOINT PROJEKTI	41
4.1 APLIKACIJA ZA PRIJAVO IN OBDELAVO IZBOLJŠAV	41
4.1.1 Vzroki za nastanek aplikacije	41
4.1.1.1 Težave skrbnikov izboljšav.....	42
4.1.1.2 Težave predlagateljev.....	43
4.1.1.3 Težave vodij inovativnih jeder	44
4.1.2 Izgradnja aplikacije	45
4.1.2.1 Tehnične in vsebinske zahteve	46
4.1.2.2 Vloge uporabnikov.....	51
4.1.2.3 Organizacija podatkov	52
4.1.3 Delovanje aplikacije.....	54
4.1.3.1 Proces prijave in obdelave izboljšav	60
4.1.3.2 Prijava izboljšave	61
4.1.3.3 Potrditev vodje inovativnega jedra	62
4.1.3.4 Potrditev skrbnika izboljšave	64
4.1.3.5 Spremljanje izvajanja izboljšave	66
4.1.3.6 Izračun prihranka izboljšave	67
4.1.3.7 Izračun nagrade izboljšave	68
4.1.3.8 Zaključevanje izboljšave	69
4.1.4 Težave in možnosti izboljšav	70
4.2 APLIKACIJA ZA NADZOR TERMINALOV	72
4.2.1 Vzroki za nastanek aplikacije	73

4.2.2 Izgradnja aplikacije	74
4.2.2.1 Tehnične in vsebinske zahteve	75
4.2.2.2 Vloge uporabnikov	76
4.2.2.3 Organizacija podatkov	76
4.2.3 Delovanje aplikacije.....	79
4.2.3.1 Podrobnosti posameznega terminala	81
4.2.3.2 Šifranti aplikacije	83
4.2.4 Težave in možnosti izboljšav	84
4.3 DODATEK ZA GRAFIČNI PRIKAZ OPRAVIL	86
4.3.1 Vzroki za nastanek dodatka.....	86
4.3.2 Izgradnja dodatka.....	87
4.3.2.1 Tehnične in vsebinske zahteve.....	87
4.3.2.2 Vloge uporabnikov	88
4.3.2.3 Organizacija podatkov.....	88
4.3.2.4 Delovanje dodatka	89
4.3.3 Težave in možnosti izboljšav	91
4.4 GRADNIK ZA IZBIRO OSEB	92
4.4.1 Vzroki za nastanek gradnika.....	92
4.4.2 Izgradnja gradnika	93
4.4.2.1 Tehnične in vsebinske zahteve.....	93
4.4.2.2 Vloge uporabnikov	94
4.4.2.3 Organizacija podatkov.....	94
4.4.2.4 Delovanje gradnika.....	94
4.4.3 Težave in možnosti izboljšav	96
5 ZAKLJUČEK.....	97
LITERATURA.....	103

POVZETEK

Diplomsko delo obsega raziskavo zmožnosti programiranja platforme SharePoint 2010 ter prikaz izdelkov, ustvarjenih s tem znanjem.

V uvodu je opisano ozadje izbire teme in nastanka diplomskega dela.

V drugem poglavju je predstavljena platforma SharePoint. Ta obsega opis njene zgradbe, delovanja in uporabnosti.

Tretje poglavje se usmeri zgolj v programiranje te platforme. Najprej so predstavljena nekatera izmed najbolj uporabnih programskih orodij za njeno programiranje, v nadaljevanju pa se posveti zgolj najmočnejšemu izmed njih – Microsoft Visual Studiu. Predstavljeni so nekateri izmed najbolj uporabnih objektov objektnega modela SharePointa, delovanje in uporabnost SharePointovih odzivnikov na dogodke (Event Receivers), delo z zunanjimi podatki (Business Data), SharePointovi delovni tokovi (WorkFlows), SharePointove spletne storitve (Web Services), kako je v SharePointu poskrbljeno za varnost ter možnosti povezovanja z aplikacijami na pametnih telefonih.

V četrtem poglavju so predstavljeni izdelki, ustvarjeni za korporacijo HIDRIA, s pridobljenim znanjem iz programiranja platforme SharePoint. Prvi izdelek se nanaša na aplikacijo za prijavo in obdelavo izboljšav, ki jih lahko prijavi vsak zaposleni znotraj korporacije. Drugi izdelek, aplikacija za nadzor terminalov, pa zavzema področje nadzora proizvodnje preko terminalov. Sledita še dodatek za grafični prikaz opravil iz SharePointovega seznama za opravila ter SharePointov gradnik za izbiro oseb.

V zaključku so predstavljene sklepne ugotovitve.

KLJUČNE BESEDE: SharePoint, programiranje za platformo SharePoint, aplikacija za prijavo in obdelavo izboljšav, aplikacija za nadzor terminalov.

ABSTRACT

The thesis includes a research on SharePoint 2010 programming capabilities and a display of products created by this knowledge.

The introduction part includes background information on how the topic was chosen and how the thesis was developed.

The second chapter presents the SharePoint platform, which includes a description of its structure, function and usability.

The third chapter focuses solely on the programming of the platform. First, some of the most useful software tools for its programming are presented and afterwards our interest is centred on the most powerful of them – Microsoft Visual Studio. Some of the main useful objects of SharePoint object model are presented, as well as performance and usability of SharePoint Event Receivers, working with Business Data, SharePoint WorkFlows, SharePoint Web Services, SharePoint 2010 security concerns and possibility to connect with applications on smart phones.

The fourth chapter presents products created for HIDRIA corporation with obtained knowledge of SharePoint development. The first product refers to an application for the registration and improvement processing that may be claimed by each employee within the corporation. The second product is an application for the control of terminals, which deals with the area of production control with the help of terminals. In addition, there is an appendix with graphical representation of SharePoint task list and SharePoint component for the selection of persons.

The concluding arguments are given at the end of the thesis.

KEYWORDS: SharePoint, SharePoint Development, application for the registration and improvement processing, application for the control of terminals.

1 UVOD

Kot štipendist poslovne enote HIDRIA IT znotraj korporacije HIDRIA d.d. sem imel v oddelku elektronskega poslovanja in razvoja možnost opravljati že zelo raznoliko delo. Prav v času nastajanja diplomske naloge smo znotraj korporacije začeli s pospešenim uvajanjem platforme Microsoft SharePoint 2010 za namene intraneta, zato je bila izbira te teme več kot primerna za mojo diplomsko nalogo.

Korporacija HIDRIA d.d. združuje 5 podjetij, med katere štejemo Hidrio Rotomatiko, Hidrio IMP Klimo, Hidrio AET, Hidrio Tomos in Hidrio Perles. Poleg tega združuje še tri inštitute, logistični center in inženiring, vse skupaj pa se razteza v 21 državah. Vse te dele korporacije združuje skupni intranet – spletne strani namenjene zgolj zaposlenim. Iz zgoraj napisanega ni težko razumeti, da imamo tukaj veliko množico zaposlenih, ki so seveda uporabniki intraneta. Številka približno 1500 uporabnikov pove, da zato potrebujemo robusten in zmogljiv ter hkrati odziven in enostaven sistem, za kar naj bi bila platforma SharePoint več kot primerna.

Že takoj sem uvidel, da je platforma SharePoint močna, vendar me je z vgrajenimi funkcionalnostmi pustila hladnokrvnega. Teh je sicer kar lepo število in lahko tako tudi ne-programerski profil uporabnika pripravi iz platforme SharePoint uporabno internetno stran, vendar je to tudi vse. Uporabnost v današnji stopnji in hitrosti razvoja informacijskih tehnologij ni več vse, vedno bolj pomembnejša je enostavnost spletnih aplikacij, kar pa platforma SharePoint za specifične potrebe z vgrajenimi funkcionalnostmi enostavno ne omogoča. Omejitve me kot programerja še kako nervirajo, zato sem se pri spoznavanju platforme SharePoint takoj preusmeril na njegovo prilagajanje – programiranje, kjer pa platforma SharePoint pokaže svojo pravo moč, ki me je v kategoriji spletnih tehnologij tako navdušila, da sem se odločil, da se vanjo poglobim.

Vse to znanje sem nato uporabil pri izdelavi dveh večjih projektov za korporativni intranet – t.i. HIDRIANET. Prvi predstavlja spletno aplikacijo za prijavo in obdelavo izboljšav, drugi pa spletno aplikacijo za nadzor terminalov. Poleg tega sem izdelal tudi dodatek za grafični prikaz SharePoint opravil ter gradnik za izbiro oseb.

Dela sem se lotil postopoma. Poleg najpomembnejšega znanja iz programiranja v platformi SharePoint, sem moral pridobiti še ustrezna znanja s področja inovacijskih kultur pri prvem projektu in znanja s področja zajema in analize podatkov v realnem času iz proizvodnje s pomočjo programske opreme ProInfo¹ preko terminalov pri drugem projektu. Temu je sledilo načrtovanje obeh spletnih aplikacij. Ti dve morata biti uporabnikom dovolj prijazni in enostavni za uporabo, hkrati pa dovolj učinkoviti, da bosta zadostili zahtevam naročnikov, ter fleksibilni ob potrebah po razširitvah ali popravkih.

¹ ProInfo je Manufacturing Execution System (sistem MES), čigar osnovna naloga je spremljanje, zajem in analiza podatkov v realnem času tako v proizvodnji kot tudi v servisni in projektni dejavnosti [8].

Sledila je izdelava obeh projektov. Časovno sem izdelal aplikacijo za prijavo in obdelavo izboljšav pred aplikacijo za nadzor terminalov, zato ju bom tako razvrstil tudi v svojem diplomskem delu. Nato pa bom predstavil še prej omenjeni dodatek in gradnik.

Pričel bom s samo predstavitvijo platforme SharePoint in ugotovitev ter znanja, ki sem jih pridobil s preučevanjem zmožnosti programiranja te platforme.

2 PLATFORMA SHAREPOINT

SharePoint je sodobna spletna tehnologija, katere začetki segajo v leto 2001. Oglaševali so jo kot revolucijo, vendar je zaradi svoje vsestranskosti zmedla marsikoga pri razumevanju njene uporabnosti. Šele različica 2007 je doživela spodoben komercialni uspeh, dandanes pa lahko mirno rečemo, da je portalski strežnik SharePoint eden najuspešnejših Microsoftovih izdelkov.

Opišemo ga lahko kot poslovni portal za komunikacijo znotraj podjetja. Predstavljamo si ga lahko kot orodje za skupinsko delo, dokumentni sistem, iskalnik za podatke v podjetju, spletno alternativo omrežnim diskom, orodje za socialno mreženje, poslovno inteligenco, projektno vodenje, platformo za izdelavo spletnih aplikacij, pa tudi kot odjemalec za baze podatkov. Tesna povezljivost s pisarniškiimi orodji družine Microsoft Office² mu daje še dodatno ceno. Ob vsem tem in dejstvu, da ga je mogoče s programiranjem prilagoditi po želji, ni težko uvideti, kako zelo je uporaben.

Omeniti pa je potrebno, da Microsoft platformo SharePoint z različico 2010 sili tudi izven podjetij v svetovni splet. Tako so nastale različice SharePoint 2010 for Internet Sites, ki so prilagojene za delo z aplikacijami za javnost, kupce, skratka za vse, ki niso za požarnimi zidovi podjetij.

SharePoint z osrednjim izdelkom SharePoint Server 2010 spada med plačljive, vendar obstaja tudi okrnjen paket t.i. SharePoint Foundation 2010, ki je popolnoma brezplačen a zato funkcionalno omejen in služi predvsem kot vaba za uporabnike, ki si želijo nekaj več.

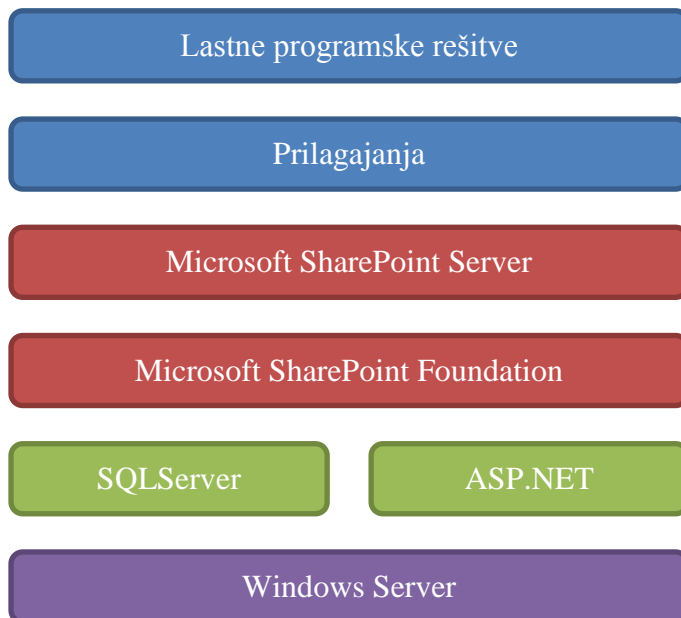
Ob vsem tem ni težko ugotoviti, da je SharePoint Server 2010 več kot primerna platforma za postavitev intranetnih strani v podjetjih. Njegovo moč je mogoče občutiti posebno v tistih podjetjih, ki temeljijo na Microsoftovih tehnologijah, kot so strežniki Microsoft Windows Server, podatkovne baze Microsoft SQL Server, pisarniška programska oprema Microsoft Office, komunikacijska oprema Microsoft Lync, pametni telefoni z Microsoft Windows Phone operacijskim sistemom itd., saj je integracija z njimi v platformi SharePoint na visokem nivoju.

Ob vseh dobrih stvareh je potrebno vedeti, da SharePoint še zdaleč ni popoln izdelek. S stališča administratorjev je še kako občutiti pomanjkanje orodja za nadzor in upravljanje vseh vsebin na portalu ter avtomatizaciji nekaterih pogostih opravil, ko je npr. selitev nekega spletnega mesta, knjižnice na drugo mesto ali strežnik. V spletu je sicer čutiti hitro naraščanje števila takih specializiranih orodij tretjih proizvajalcev, ki pa niso najbolj poceni. Med najbolj priljubljena orodja te vrste spadajo izdelki podjetja Quest Software [27].

² Microsoft Office je zbirka pisarniških programov, ki jih razvija in trži podjetje Microsoft na Windows in Macintosh platformi. Sestavlja jo urejevalnik besedil Word, program za preglednice Excel in program za predstavitve PowerPoint. V dražjih različicah, namenjene poslovnim uporabnikom, pa še program za upravljanje z zbirkami podatkov Microsoft Access ter odjemalec elektronske pošte Outlook [9].

2.1 ZGRADBA PLATFORME SHAREPOINT

Če hočemo uspešno programirati na platformi SharePoint, moramo razumeti njeno delovanje, to pa je še kako odvisno od same zgradbe platforme SharePoint [1].



Slika 1: Zgradba platforme SharePoint

Kot je mogoče videti na zgornji sliki, platformo SharePoint poganja operacijski sistem Windows Server, pri tem pa uporablja spletni strežnik Internet Information Service (IIS)³. Na tem mestu lahko omenim, da je z nekaj modifikacijami namestitvenega procesa mogoče produkt SharePoint namestiti tudi na operacijski sistem Windows 7, kar je še kako uporabno v testne namene, česar sem se posluževal tudi sam.

Platforma SharePoint je odvisna od podatkovnega strežnika Microsoft SQL Server ter ogrodja ASP.NET⁴. Tako se lahko poznavalec tega ogrodja lepo znajde tudi na platformi SharePoint, saj je večina njegovih konceptov programiranja mogoče uporabiti tudi tu.

Nivo Microsoft SharePoint Foundation predstavlja osnovne funkcionalnosti platforme SharePoint, naslednji nivo Microsoft SharePoint Server pa dodatne funkcionalnosti, ki razširjajo verzijo Foundation in prinašajo plačljivost. Sledita še nivoja funkcionalnosti po meri, ki jih lahko kupimo od tretjih proizvajalcev ali pa izdelamo sami.

³ Internet Information Service (IIS) predstavlja Microsoftov spletni strežniški sistem, ki je namenjen ogrodju ASP.NET oz. staremu ASP.

⁴ Microsoft ASP.NET je ogrodje za razvijanje spletnih aplikacij za Microsoftov spletni strežnik Internet Information Services (IIS).

2.2 ORGANIZACIJA STRANI V SHAREPOINTU

Na platformi SharePoint so strani organizirane po naslednji hierarhiji:

- **Spletna aplikacija (Web Application)**, ki živi na spletnem strežniku IIS.
- **Zbirka strani (Site Collection)**, ki predstavlja glavno SharePoint stran. Teh je lahko znotraj ene spletne aplikacije tudi več.
- **Posamezne strani (Sites)**, ki živijo znotraj zbirk strani.



Slika 2: Organizacija strani na platformi SharePoint

Platforma SharePoint s tem prinaša dodatno urejenost, ki še kako pride v poštev v primerih, ko imamo znotraj spletne aplikacije ogromno število strani, kar je npr. za intranet tudi značilno.

V posamezno stran lahko vstavimo dele strani (Web Parts), ki delujejo samostojno, neodvisno od drugih elementov, ki se nahajajo na tej strani.

2.3 CENTRALNA ADMINISTRACIJA

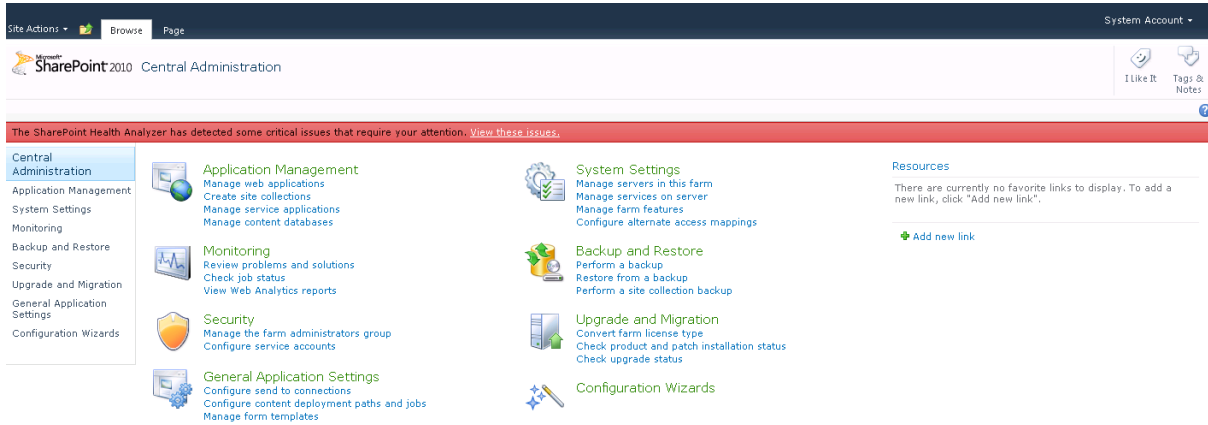
Ker moja diplomska naloga zajema programiranje oz. prilagajanje platforme SharePoint in ne njegovo administracijo, sem to podpoglavje vključil zgolj informativno in mu ne bom posvetil prevelike pozornosti.

Ob namestitvi platforme SharePoint imamo privzeto ustvarjeni dve spletni aplikaciji. Prva predstavlja predlogo strani, ki jo uporabimo za izdelavo spletne aplikacije (v našem primeru intraneta). Druga pa predstavlja administrativne strani platforme SharePoint – t.i. centralna administracija.

V centralni administraciji lahko administrator izvaja lepo število aktivnosti, ki so razdeljene na 8 različnih skupin:

- **urejanje in nadzor aplikacije** (urejanje spletnih aplikacij, urejanje zbirk strani itd.);
- **spremljanje sistema** (problemi in rešitve, ogled poročil itd.);
- **varnost** (urejanje dovoljenj dostopanja, pravic oseb in skupin itd.);
- **splošne nastavitve aplikacije** (možnosti pošiljanja elektronske pošte itd.);

- **systemske nastavitve** (urejanje strežnikov, storitev, funkcij itd.);
- **varnostno kopiranje in obnavljanje**;
- **posodobitve in migracija** (sprememba licenčnega tipa izdelka, posodabljanje itd.);
- **nastavitveni čarovnik** (vodeno spreminjanje nastavitvev platforme SharePoint).



Slika 3: Centralna administracija v SharePointu

3 PROGRAMIRANJE V SHAREPOINTU

Programiranje v platformi SharePoint je zanimiv hibrid različnih Microsoftovih tehnologij in načinov razvijanja programske opreme. Ko programiramo del strani (Web Part) ali stran v platformi SharePoint, je občutek zelo podoben, kot če bi programirali del strani ali stran v ASP.NET. Tu je tudi vedno univerzalni LINQ⁵ za povezavo do podatkovnih baz SQL [4], ki je med programerji še kako priljubljen, pa deklarativno programiranje, programiranje potekov dela (Workflow Programming) in še bi lahko našteval.

Vendar platforma SharePoint ponuja nekaj več. Ima veliko množico storitev, programskih vmesnikov (API) in objektov, ki so pri spletnem programiranju še kako uporabni, kar naredi razvijalsko izkušnjo bogatejšo. To je tudi pglavitna stvar, ki mene pri platformi SharePoint tako navdušuje, da mi ASP.NET, ASP.NET MVC⁶ in ostale spletne tehnologije niso več tako zanimive.

Programska rešitev, ki jo programiramo za platformo SharePoint, je dejansko .NET aplikacija⁷, spisana v programskem jeziku C# ali Visual Basic. Sam sem pristaš programskega jezika C#, zato so vsi projekti v tej diplomski nalogi spisani v tem jeziku, prav tako priloženi primeri izvirne kode.

SharePoint s svojo ogromno množico različnih tehnologij ob prvem stiku ustvari pri programerju zmedo, vendar ob sistematičnem spoznavanju to kmalu preide v navdušenje.

3.1 PROGRAMERSKA ORODJA

Platforma SharePoint omogoča programerju uporabljati kar nekaj programerskih orodij. Mednje spadajo:

- osnovno programiranje preko brskalnika (Browser-Based Development),
- SharePoint Designer 2010,
- Visual Studio 2010 in
- Expression Blend.

3.1.1 Osnovno programiranje preko brskalnika

Za to vejo programerskih orodij platforme SharePoint lahko rečemo, da je bolj kaj drugega kot pa programerska, pa tudi ne gre ravno za pravo orodje, saj lahko z nekaj kliki v centralni

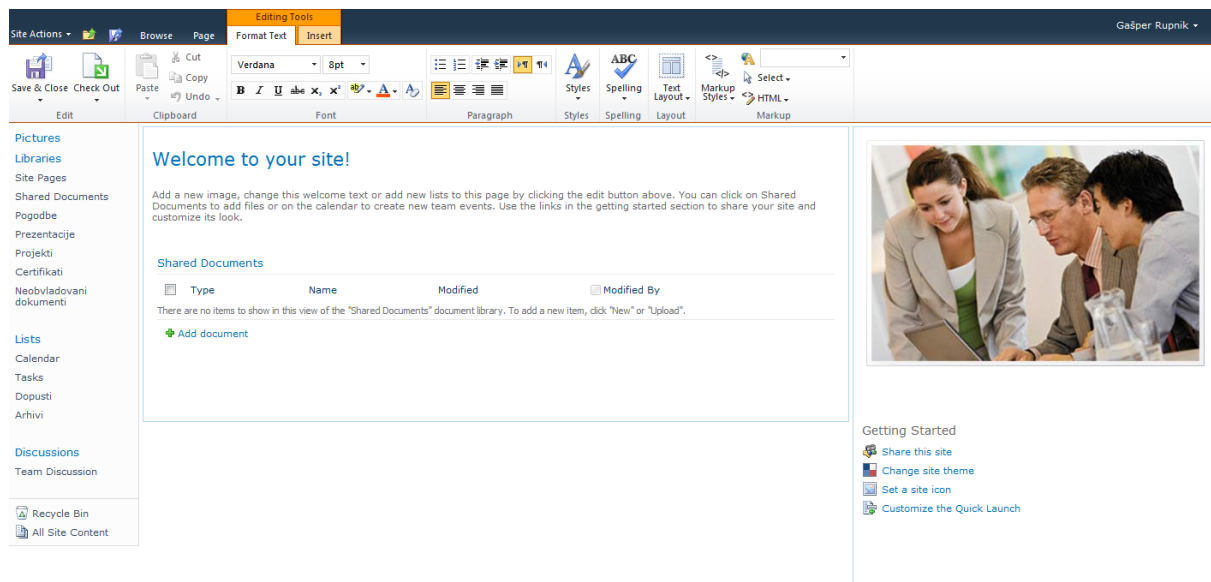
⁵ Language Integrated Query (LINQ) je tehnologija za lažjo povezavo do podatkovnih baz. Ta pretvori jezikovne konstrukte SQL v jezikovne konstrukte LINQ in obratno, kar omogoča objektno delo s podatkovno bazo [10].

⁶ Model-View-Controller (MVC) je programska arhitektura, ki se uporablja v razvoju programske opreme. Razdeljena je na model (Model), pogled (View) in upravljalca (Controller) in tako ločuje metode za dodajanje, odstranjevanje in posodabljanje informacij v bazi podatkov od pogleda za pridobitev podatkov iz modela ter upravljalca, ki je odgovoren za uporabnikova dejanja [11].

⁷ .NET aplikacija je programska oprema, narejena za ogrodje .NET (.NET framework). Aplikacije, narejene za to ogrodje, so neodvisne od okolja (operacijskega sistema, tipa računalnika itd.), v katerem živijo [12].

administraciji ali pa kar sproti v spletni aplikaciji preko brskalnika enostavno ustvarimo novo stran, oblikujemo njeno vsebino na podoben način kot v urejevalniku besedil, gradimo hierarhične menije in podmenije, dodajamo ali zamenjavamo vsebino, slike, tabele, administriramo uporabnike itd.

Vse to spominja na t.i. sisteme CMS⁸. Platforma SharePoint vsekakor spada mednje. Glede na njene značilnosti nekateri platformo SharePoint poimenujejo tudi kar Intranet CMS sistem.



Slika 4: Osnovno programiranje preko brskalnika

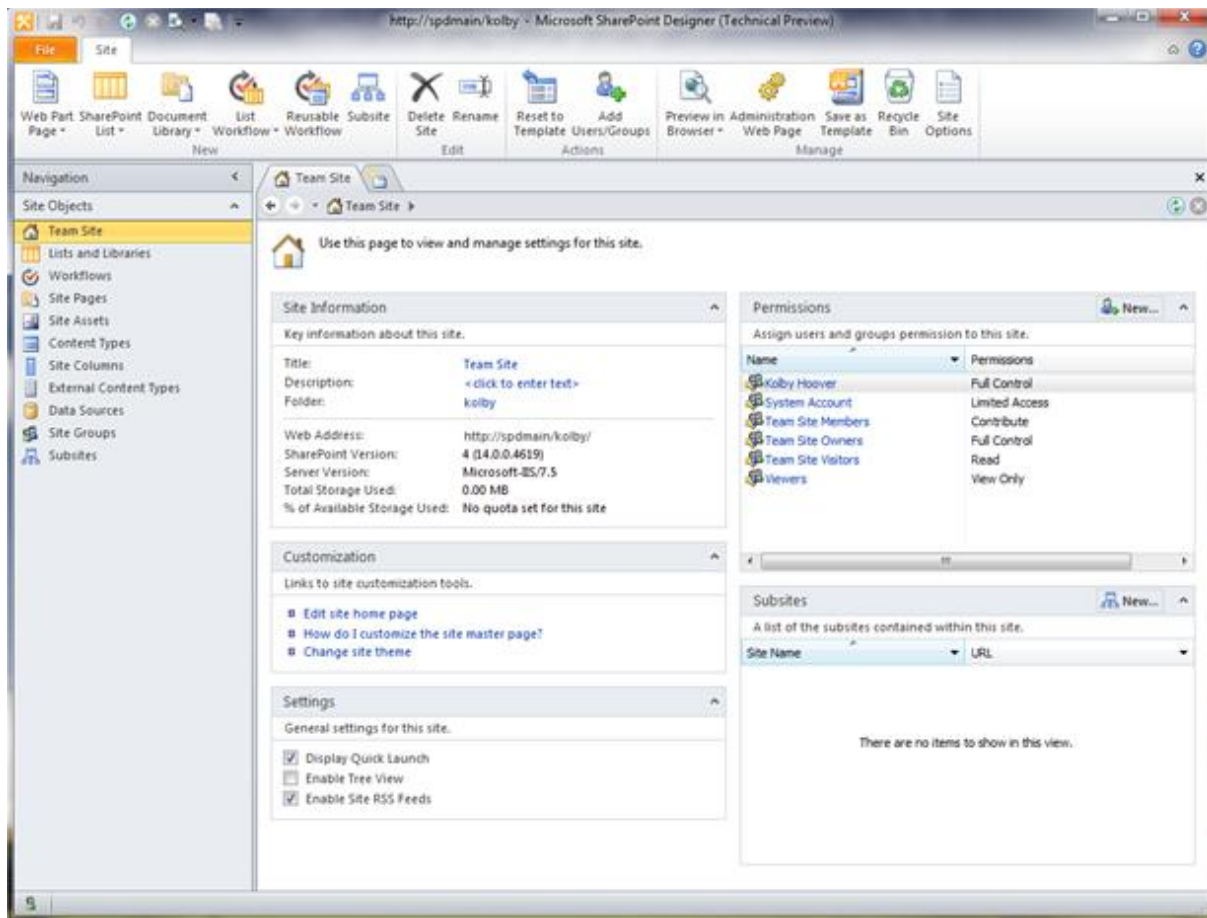
3.1.2 Sharepoint Designer 2010

Programsko orodje SharePoint Designer 2010 lahko za razliko od prejšnjega šteje med samostojno programsko opremo, ki je namenjena zgolj programiranju platforme SharePoint. Omogoča vse to, kar omogoča tudi prej omenjeno osnovno programiranje preko brskalnika z nekaj dodatnimi možnostmi nastavljanja. Vse sicer poteka podobno kot zgoraj s kliki in čarovniki, deloma pa omogoča tudi pisanje izvorne kode, zato lahko to orodje uvrščamo nekje med osnovnim programiranjem preko brskalnika ter programskim orodjem Visual Studio 2010.

Sam se sicer SharePoint Designerju 2010 pri svojem delu izogibam, saj mi ne ponuja nič več kot osnovno programiranje preko brskalnika, za ostale stvari pa raje uporabljam Visual Studio 2010, ki mi nudi več svobode pri ustvarjanju. Poleg tega deluje SharePoint Designer 2010 nestabilno. Sesutje programskega orodja ni redko, sploh če je povezava med SharePoint Designerjem in SharePoint Serverjem slaba.

⁸ Content Management System (CMS) je sistem, ki omogoča urejanje in vzdrževanje vsebine spletnih strani brez znanja označevalnega jezika HTML. Urednik spletne strani tako lahko samostojno spreminja besedila, slike in druge elemente spletne strani brez pomoči podjetja ali osebe, ki je stran izdelalo [13].

Izpostavil pa bi še eno dobro stvar SharePoint Designerja 2010, in sicer da ga ni potrebno namestiti na isti sistem oz. strežnik kot SharePoint Server, kar pa ne velja za programsko opremo Visual Studio 2010.

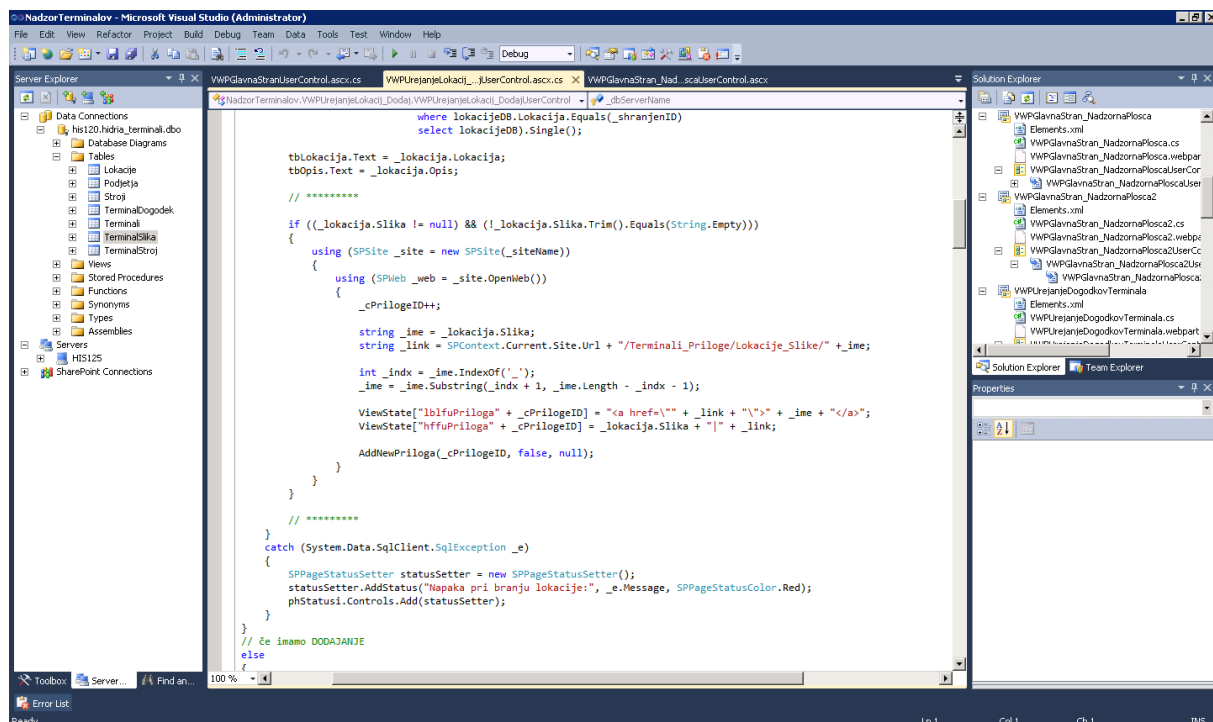


Slika 5: Programsko orodje SharePoint Designer 2010

3.1.3 Visual Studio 2010

Programsko orodje Visual Studio 2010 lahko prav tako uvrščamo med samostojno programsko opremo, le da ta ni namenjena zgolj programiranju platforme SharePoint ampak vseh Microsoftovih tehnologij. Ne samo da omogoča vse, kar omogočata zgornji dve orodji, ampak omogoča za platformo SharePoint dejansko programirati vse, kar se sploh lahko programira. Tako je programer brez kakršnih koli omejitev, kaj lahko naredi in kaj ne, kar mi osebno veliko pomeni.

Vse stvari, ki sem jih do sedaj sprogrimiral za platformo SharePoint, so tako nastale s tem programskim orodjem, zato se bodo tudi v nadaljevanju moje diplomske naloge vsi praktični primeri ter oba projekta nanašala nanj.



Slika 6: Visual Studio 2010

3.1.4 Expression Blend

Prav tako kot Visual Studio 2010 je tudi Expression Blend programsko orodje, ki ni namenjeno zgolj programiranju platforme SharePoint. Za razliko od ostalih programskih orodij za SharePoint pa ta predstavlja odličen pripomoček za izdelavo aplikacij, obogateneh z dinamičnim in bolj kompleksnim oblikovanjem uporabniškega vmesnika (animacije, obnašanja, gradienti itd.). Vse to omogoča Microsoftova tehnologija Silverlight⁹. Dejstvo, da je vgrajena v platformo SharePoint, je več kot dobrodošlo, po drugi strani pa to ni nič čudnega, saj SharePoint temelji na ogrodju ASP.NET, kjer je tehnologija Silverlight prisotna že kar nekaj časa.

Sam se s programskim orodjem Expression Blend nisem ukvarjal, saj predstavlja zgolj nek lepotni dodatek, kar pa ni cilj moje diplomske naloge.

⁹ Tehnologija Silverlight predstavlja ogrodje, namenjeno pisanju in poganjanju obogateneh spletnih aplikacij, kjer že vrsto let deluje tudi Adobe Flash. Na voljo je kot vtič za spletne brskalnike.



Slika 7: Expression Blend

3.2 USTVARJANJE SHAREPOINTOVE PROGRAMSKE REŠITVE Z ORODJEM VISUAL STUDIO 2010

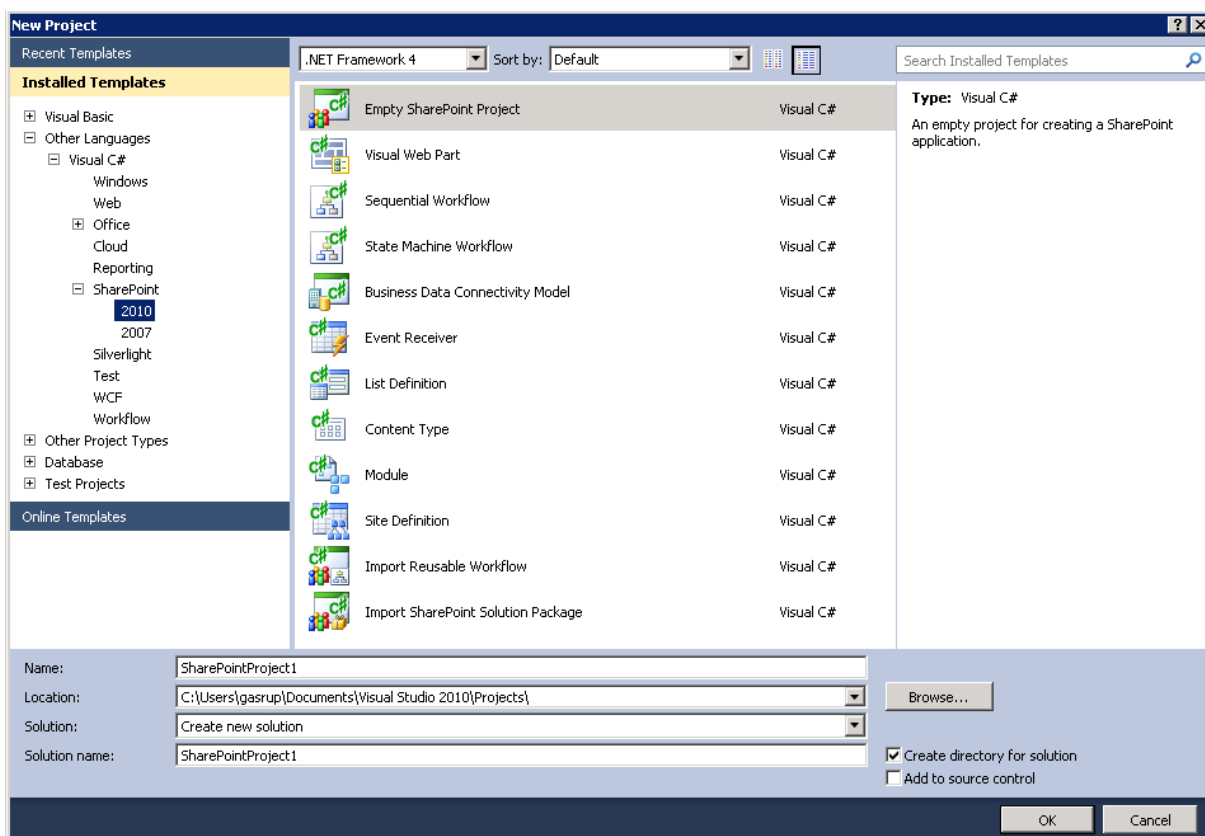
V predhodnih poglavjih sem predstavil samo platformo SharePoint ter orodja, s katerimi lahko na njej programiramo. Od tu naprej pa se bom podrobneje osredotočil samo na orodje Visual Studio 2010, ki je tudi najmočnejše od vseh prej naštetih.

3.2.1 Sharepoint predloge v Visual Studiu 2010

Za delo s platformo SharePoint ponuja Visual Studio programerju 10 predlog:

- **Prazen SharePoint projekt (Empty SharePoint Project):** ustvari se prazen SharePoint projekt do te mere, da ga je mogoče pripeti na platformo SharePoint oz. iz nje odstraniti – torej samo ogrodje brez kakršnekoli funkcionalnosti.
- **Del strani (Web Part):** ustvari se Sharepointov projekt, v katerega je že vključena predpriprava za izdelavo dela strani.
- **Zaporedni potek dela (Sequential Workflow):** ustvari se SharePoint projekt, ki je predpripravljen za izdelavo zaporednega poteka dela. To pomeni, da pišemo zaporedne akcije, ki jih mora platforma SharePoint izvesti po točno določenem vrstnem redu.

- **Potek dela s stanji (State Machine Workflow):** ustvari se SharePoint projekt, ki je predpripravljen za izdelavo poteka dela s stanji. To pomeni, da določimo stanja in povezave med stanji ter dogodke, ki prožijo prehode med stanji. Platforma SharePoint se sprehaja med temi stanji glede na dogodke, ki se v nekem trenutku dogodijo.
- **Delo z zunanjimi podatki (Business Data Connectivity Model):** ustvari se SharePoint projekt, ki omogoča integracijo zunanjih podatkov v SharePoint. To pomeni, da lahko tabele neke zunanje podatkovne baze pretvorimo v SharePoint sezname.
- **Odzivnik na dogodke (Event Receiver):** ustvari se SharePoint projekt, ki omogoča definiranje nalog, katere naj SharePoint platforma izvede, ko pride do nekega dogodka na SharePoint seznamu oz. na nekem drugem gradniku platforme SharePoint.
- **Definicija seznama (List Definition):** ustvari se SharePoint projekt, s katerim lahko definiramo delovanje nekega SharePoint seznama (kakšen je njegov prikaz, kako lahko ažuriramo podatke v njem itd.).
- **Tip vsebine (Content Type):** ustvari se SharePoint projekt, s katerim lahko definiramo vsebino nekega SharePoint seznama (kakšne podatke lahko vanj shranimo).
- **Modul (Module):** ustvari se SharePoint projekt, s katerim lahko definiramo nek zaključen del, ki vsebuje eno ali več datotek, ki bi jih radi dodali neki SharePoint strani.
- **Definicija strani (Site Definition):** ustvari se SharePoint projekt, s katerim lahko definiramo predlogo za neko stran. S tem definiramo začetno vsebino strani, ki se prikaže, ko ustvarimo novo SharePoint stran s to definicijo strani.



Slika 8: Visual Studio predloge za platformo SharePoint

3.2.2 Sandboxed in farmske programske rešitve

Vsako od prej naštetih 10 predlog se lahko ustvari kot t.i. Sandboxed programsko rešitev ali pa kot programsko rešitev celotne farme strežnikov, ki jih pod seboj združuje ena platforma SharePoint. SharePoint projekt ustvarimo kot programsko rešitev celotne farme strežnikov takrat, ko bi radi, da se ta projekt nahaja na nivoju spletne aplikacije (Web Application), ker potrebujemo komunikacijo med več zbirkami strani (Site Collection). Vse ostale SharePoint projekte ustvarimo kot Sandboxed programsko rešitev, kar pomeni, da se tak projekt nahaja na nivoju zbirke strani (Site Collection).

3.2.3 Preslikovanje map in datotek

Preslikovanje map in datotek v Visual Studio projektu za platformo SharePoint nam daje možnost dostopa do sredstev, kjer se le te morajo v datotečni strukturi platforme SharePoint¹⁰ nahajati. Kot primer sredstva vzemimo slike. V platformi SharePoint je za odlaganje slik namenjena posebna mapa IMAGES. Vanjo bi v nekem SharePoint projektu radi dodali nekaj slik, ki bi jih potrebovali tudi v nekem drugem projektu. V ta namen za vsak projekt ustvarimo navidezno mapo, katere vsebina se preslika v to posebno mapo IMAGES. S tem se izognemo zmešnjavi, ki bi nastala zaradi različnih mest, na katere bi shranjevali slike vsakega projekta posebej.

¹⁰ Datotečni strukturi platforme SharePoint strokovno pravijo panj (Hive) in jo je na SharePoint strežniku možno najti na lokaciji »...\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14«.

3.2.4 Pakiranje funkcionalnosti

Znotraj nekega Visual Studio projekta za platformo SharePoint lahko izberemo, katere funkcionalnosti tega projekta želimo dodati v platformo SharePoint in katere ne. To določimo v datoteki *Elements.xml*, ki jo vsebuje vsak projekt platforme SharePoint. Temu postopku pravimo pakiranje funkcionalnosti.

Kot primer si zamislimo projekt, v katerem smo izdelali več delov strani (Web Parts) za platformo SharePoint. Ker enega izmed teh delov strani nismo dokončali, ga nočemo prenesti na platformo SharePoint. To lahko omejimo s pakiranjem funkcionalnosti.

3.3 SHAREPOINTOVI OBJEKTNI MODELI

SharePointovi objektni modeli so ogrodje za programiranje platforme SharePoint. Platforma SharePoint vsebuje štiri take objektivne modele. Pri tem, ko programiramo v platformi SharePoint, tako rekoč neprestano uporabljamo objekte iz teh objektivnih modelov. Lahko bi rekli, da je torej programiranje v SharePointu popolnoma objektivno usmerjeno, kar mi je všeč.

Največji in s tem tudi prvi objektivni model se imenuje objektivni model SharePoint Server. Ta objektivni model se uporablja za vso SharePoint kodo, ki teče na strežniku SharePoint. Ostali trije objektivni modeli pa se uporabljajo za vso kodo, ki teče na odjemalčevi strani – torej v odjemalčevem spletnem brskalniku ali v odjemalčevi aplikaciji, ki komunicira s platformo SharePoint [1]. Odjemalčevi objektivni modeli so več ali manj preslikava strežniških, zato je dovolj že sama obravnava objektivnega modela SharePoint Server.

Spodnja slika prikazuje soodvisnost med posameznimi objekti znotraj objektnega modela platforme SharePoint.



Slika 9: Prikaz soodvisnosti med nekaterimi objekti objektnega modela SharePointa

Objektni model SharePointa je ogromen, zato bom v nadaljevanju na kratko predstavil zgolj nekatere njegove najbolj uporabljene tipe objektov. Za vsakega bom podal nekaj najbolj uporabljenih atributov in metod ter primer njihove uporabe v obliki kode.

3.3.1 SPContext

Objekt tipa SPContext je na voljo na vseh SharePoint straneh in uporablja vsebino zahteve HTTP, da omogoča dostop do objektov objektnega modela SharePoint.

ATRIBUT ALI METODA	OPIS
Web	Objekt tipa SPWeb, ki predstavlja stran. Nanaša se na vsebino zahteve HTTP.
Site	Objekt tipa SPSite, ki predstavlja zbirko strani. Nanaša se na vsebino zahteve HTTP.
Current	Objekt tipa SPContext, ki predstavlja vsebino trenutne zahteve HTTP.

Tabela 1: Najbolj uporabljeni atributi oz. metode objekta tipa SPContext

```
public void Page_Load(object sender, EventArgs e)
{
    // Pridobi vsebino trenutne HTTP zahteve
    SPContext _context = SPContext.Current;

    // Iz vsebine izpiši URL zbirke strani ter URL strani
    StringBuilder _izpis = new StringBuilder();
    _izpis.AppendFormat("URL zbirke strani: {0}\n", _context.Site.Url);
    _izpis.AppendFormat("URL strani: {0}\n", _context.Web.Url);
}
```

3.3.2 SPFarm

Objekt tipa SPFarm predstavlja SharePoint farmo strežnikov in vsebuje množico atributov in metod, ki omogočajo prilagajanje nastavitve platforme SharePoint na nivoju farme.

ATRIBUT ALI METODA	OPIS
Services	Objekt tipa SPServiceCollection, v katerem se nahaja seznam vseh storitev, ki na platformi SharePoint tečejo na nivoju farme.
Local	Objekt tipa SPFarm za lokalni strežnik.
Products	Vrne seznam vseh GUID ¹¹ produktov, ki so nameščeni na farmi.
Solutions	Objekt tipa SPSolutionCollection, ki vsebuje seznam vseh programskih rešitev, ki so na voljo na farmi.
Servers	Objekt tipa SPServerCollection, v katerem se nahaja seznam vseh fizičnih računalnikov (strežnikov), ki so vsebovani v farmi.
Create(...)	Funkcija za avtomatsko kreiranje nove farme. Vrne objekt tipa SPFarm.
Open(...)	Funkcija za zagon neke farme. Vrne objekt tipa SPFarm.
Join(...)	Funkcija za pripenjanje novega računalnika (strežnika) farmi.

Tabela 2: Najbolj uporabljeni atributi oz. metode objekta tipa SPFarm

¹¹ Globally Unique Identifier (GUID) je unikatna referenčna številka, namenjena identificiranju v računalniških programih. Predstavljena je z nizom dolžine 32 znakov [15]. Primer: {21EC2020-3AEA-1069-A2DD-08002B30309D}

```

public void Page_Load(object sender, EventArgs e)
{
    // Pridobi SharePoint farmo, kateri pripada lokalni strežnik
    SPFarm _farm = SPFarm.Local;

    StringBuilder _izpis = new StringBuilder();

    // izpis seznama vseh strežnikov iz te SharePoint farme
    SPServerCollection _servers = _farm.Servers;
    Foreach (SPServer _server in _servers)
    {
        _izpis.AppendFormat("Ime strežnika: {0}\n", _server.Name);
    }
}

```

3.3.3 SPWebApplication

Objekt tipa SPWebApplication omogoča dostop do nastavitev in stanja spletne aplikacije IIS, v kateri se nahaja zbirka strani, iz katere preko kode uporabljamo ta tip objekta.

ATRIBUT ALI METODA	OPIS
Sites	Objekt tipa SPSiteCollection, v katerem se nahaja seznam vseh SPSite objektov, ki predstavljajo vse zbirke strani znotraj te SharePoint spletne aplikacije IIS.
Features	Objekt tipa SPFeatureCollection, ki vsebuje seznam vseh SPFeature objektov, ki predstavljajo vse funkcije, ki so aktivne v spletni aplikaciji IIS.
Delete()	Funkcija za izbris spletne aplikacije IIS.
Lookup(...)	Funkcija vzame naslov URL ¹² kot parameter in vrne SPWebApplication objekt za dani naslov URL.
DisplayName	Ime spletne aplikacije.
ContentDatabases	Objekt tipa SPContentDatabaseCollection, ki vsebuje seznam vseh podatkovnih baz, ki jih ta spletna aplikacija IIS uporablja.

Tabela 3: Najbolj uporabljeni atributi oz. metode objekta tipa SPWebApplication

```

public void Page_Load(object sender, EventArgs e)
{
    // Pridobi spletno aplikacijo IIS, ki je dosegljiva na spodnjem
    // naslovu
    SPWebApplication _webApplication = SPWebApplication.Lookup(new
        Uri("http://his125.hidria.local"));

    // Vrni ime spletne aplikacije
    StringBuilder _izpis = new StringBuilder();
    _izpis.AppendFormat("Ime spletne aplikacije: {0}\n",
        _webApplication.DisplayName);
}

```

¹² Uniform Resource Locator (URL) je naslov neke spletne strani v svetovnem spletu. Vsaka spletna stran ima edinstven naslov, ki jo enolično določa [16].

3.3.4 SPSite

Objekt tipa SPSite predstavlja SharePoint zbirko strani, znotraj katere se lahko nahaja ena ali več strani.

ATRIBUT ALI METODA	OPIS
Url	Naslov URL glavne spletne strani v zbirki strani.
ID	GUID, ki unikatno predstavlja zbirko strani.
OpenWeb(...)	Funkcija, ki vrne objekt tipa SPWeb določen s parametrom GUID ali URL.
Features	Objekt tipa SPFeatureCollection, v katerem se nahaja seznam vseh objektov tipa SPFeature, ki predstavljajo vse funkcije, ki so nameščene in aktivirane v zbirki strani.
Solutions	Objekt tipa SPUserSolutionCollection, v katerem se nahaja seznam vseh objektov tipa SPUserSolution, ki predstavljajo vse rešitve, ki so nameščene in aktivirane v zbirki strani.
Usage	Objekt tipa UsageInfo, ki daje informacije o uporabi zbirke strani (število obiskov, število strani itd.).
Delete(...)	Funkcija za izbris zbirke strani.
Exists(Uri)	Funkcija, ki vrne pritrilno vrednost, če zbirka strani s podanim naslovom URL obstaja.
RootWeb	Objekt tipa SPWeb, ki predstavlja glavno spletno stran v zbirki strani.
AllWebs	Objekt tipa SPWebCollection, v katerem se nahaja seznam vseh objektov tipa SPWeb, ki predstavljajo vse spletne strani znotraj zbirke strani.
Owner	Objekt tipa SPUser, ki predstavlja lastnika zbirke strani.
Port	Številka vrat, ki jih uporablja zbirka strani (privzeto so to vrata 80).
Zone	Objekt tipa SPUrlZone, ki predstavlja katero cono uporablja zbirka strani (intranet, internet itd.).
ServerRelativeUrl	Relativni URL glavne spletne strani RootWeb v tej zbirki strani.
ContentDatabase	Objekt tipa SPContentDatabase, ki predstavlja podatkovno bazo te zbirke strani.
EventReceivers	Objekt tipa SPEventReceiverDefinitionCollection, v katerem se nahaja seznam vseh objektov tipa SPEventReceiverDefinition, ki predstavljajo registrirane odzivnike na dogodke iz te zbirke strani.
HostName	Ime strežnika, na katerem gostuje zbirka strani.

Tabela 4: Najbolj uporabljeni atributi oz. metode objekta tipa SPSite

```
public void Page_Load(object sender, EventArgs e)
{
    // poveži se na zbirko strani, katere ime je vpisano v
    // spremenljivki _siteName
    using (SPSite _site = new SPSite(_siteName))
    {
        // izpiši URL in ID zbirke strani
        StringBuilder _izpis = new StringBuilder();
    }
}
```

```

        _izpis.AppendFormat("URL zbirke strani: {0}\n",
            _site.Url);
        _izpis.AppendFormat("ID zbirke strani: {0}\n",
            _site.ID);
    }
}

```

3.3.5 SPWeb

Objekt tipa SPWeb predstavlja eno izmed strani v SharePoint zbirki strani.

ATRIBUT ALI METODA	OPIS
ID	GUID, ki unikatno predstavlja stran.
Url	Naslov URL strani.
Lists	Objekt tipa SPListCollection, v katerem se nahaja seznam vseh objektov tipa SPList, ki predstavlja vse SharePoint sezname znotraj strani.
Name	Ime strani.
Title	Naslov strani.
Site	Objekt tipa SPSite, ki predstavlja zbirko strani, znotraj katere se nahaja ta stran.
Created	Datum in čas ustvarjanja te strani.
Navigation	Objekt tipa SPNavigation, ki omogoča dostop do območja s hitrimi povezavami in zgornje navigacijske vrstice strani.
Webs	Objekt tipa SPWebCollection, v katerem se nahaja seznam vseh SPWeb objektov – torej zbirko vseh strani, ki so otrok te strani.
Workflows	Objekt tipa SPWorkflowCollection, v katerem se nahaja seznam vseh SPWorkflow objektov, ki predstavljajo vse poteke dela znotraj te strani.
Modules	Objekt tipa SPModuleCollection, v katerem se nahaja seznam vseh objektov tipa SPModule, ki predstavljajo vse module znotraj te strani.
CurrentUser	Objekt tipa SPUser, ki predstavlja uporabnika, ki trenutno pregleduje to stran.
EnsureUser(string)	Funkcija, ki vzame niz sestavljen iz uporabniškega imena uporabnika, ki ga hočemo dodati kot uporabnika na to stran. Vrne objekt tipa SPUser, ki predstavlja tega uporabnika.

Tabela 5: Najbolj uporabljeni atributi oz. metode objekta tipa SPWeb

```

public void Page_Load(object sender, EventArgs e)
{
    // poveži se na zbirko strani, katere ime je vpisano v
    // spremenljivki _siteName
    using (SPSite _site = new SPSite(_siteName))
    {
        // Poveži se s trenutno odprto stran
        using (SPWeb _web = _site.OpenWeb())
        {
            // Pridobi trenutnega uporabnika

```

```

SPUser _user = _web.CurrentUser;

// Izpiši podatke strani
StringBuilder _izpis = new StringBuilder();
_izpis.AppendFormat("URL strani: {0}\n",
    _web.Url);
_izpis.AppendFormat("Število seznamov: {0}\n",
    _web.Lists.Count);
    }
}
}

```

3.3.6 SPList

Objekt tipa SPList predstavlja SharePointov seznam, ki je namenjen shranjevanju podatkov.

ATRIBUT ALI METODA	OPIS
Update(...)	Funkcija, ki posodobi podatkovno bazo z vsemi spremembami, ki smo jih naredili z uporabo SPList metod in lastnosti.
Items	Objekt tipa SPListItemCollection, v katerem se nahaja seznam vseh objektov tipa SPListItem, ki predstavljajo vse vrstice v seznamu.
Title	Naslov/ime SharePoint seznama.
ID	GUID, ki unikatno določa SharePoint seznam.
Lists	Objekt tipa SPListCollection, v katerem se nahaja zbirka vseh SharePoint seznamov, med njimi tudi trenutni seznam.
Fields	Objekt tipa SPFieldCollection, v katerem se nahaja seznam vseh objektov tipa SPField, ki predstavljajo vse stolpce v seznamu.
Forms	Objekt tipa SPFormCollection, v katerem se nahaja seznam vseh objektov tipa SPForm, ki predstavljajo vse obrazce tega seznama (obrazec za kreiranje, popravljanje, prikaz vrstice seznama).
AddItem(...)	Funkcija, ki doda nov element seznama v SharePoint seznam.
GetItemById(int)	Funkcija, ki vrne element seznama, določen z ID številko.
EventReceivers	Objekt tipa SPEventReceiverDefinitionCollection, v katerem se nahaja seznam vseh objektov tipa SPEventReceiverDefinition, ki predstavljajo registrirane odzivnike na dogodke za ta SharePoint seznam.
Ordered	Urejanje vrstnega reda elementov seznama.

Tabela 6: Najbolj uporabljeni atributi oz. metode objekta tipa SPList

```

public void Page_Load(object sender, EventArgs e)
{
    // poveži se na zbirko strani, katere ime je vpisano v
    // spremenljivki _siteName
    using (SPSite _site = new SPSite(_siteName))
    {
        // Poveži se s trenutno odprto stran
        using (SPWeb _web = _site.OpenWeb())
        {
            // Poveži se na SharePoint seznam z imenom Izboljsave

```

```

        SPList _list = _web.Lists["Izboljsave"];

        // Izpiši nekatere podatke SharePoint seznama
        StringBuilder _izpis = new StringBuilder();
        _izpis.AppendFormat("Ime seznama: {0}\n",
            _list.Title);
        _izpis.AppendFormat("Datum generiranja: {0}\n",
            _list.Created);
    }
}
}

```

3.3.7 SPListItem

Objekt tipa SPListItem predstavlja element SharePoint seznama.

ATRIBUT ALI METODA	OPIS
Web	Objekt tipa SPWeb, ki predstavlja stran, na kateri se nahaja element seznama.
Item	Vrednost v elementu seznama, ki ga izberemo z identifikacijsko številko.
Name	Ime elementa seznama.
Xml	Podatki elementa seznama kot niz v formatu XML ¹³ .
ID	Številka, ki unikatno določa element seznama.
Fields	Objekt tipa SPFieldCollection, v katerem se nahaja seznam vseh objektov tipa SPField, ki predstavljajo vse stolpce seznama, iz katerega izhaja element seznama.
Copy(...)	Funkcija, ki kopira element seznama iz enega SharePoint seznama v drugega.
Attachments	Objekt tipa SPAttachmentsCollection, v katerem se nahaja seznam vseh objektov tipa SPAttachment, ki predstavljajo vse priloge nekemu elementu SharePoint seznama.

Tabela 7: Najbolj uporabljeni atributi oz. metode objekta tipa SPListItem

```

public void Page_Load(object sender, EventArgs e)
{
    // poveži se na zbirko strani, katere ime je vpisano v
    // spremenljivki _siteName
    using (SPSite _site = new SPSite(_siteName))
    {
        // Poveži se s trenutno odprto stran
        using (SPWeb _web = _site.OpenWeb())
        {
            // Poveži se na SharePoint seznam z imenom Izboljsave
            SPList _list = _web.Lists["Izboljsave"];

            // Iz seznama poberi prvi element

```

¹³ Extensible Markup Language (XML) je razširljiv označevalni jezik. Podoben je jeziku HTML. Omogoča nam format za opisovanje strukturiranih podatkov [17].

```

        SPListItem _item = _list.Items[0];

        // Izpiši podatke iz elementa seznama
        StringBuilder _izpis = new StringBuilder();
        _izpis.AppendFormat("Ime: {0}\n",
            _item.Name);
        _izpis.AppendFormat("URL: {0}\n",
            _item.Url);
    }
}
}

```

3.3.8 SPFile

Objekt tipa SPFile predstavlja neko datoteko v SharePointu. Kot datoteka je lahko tu mišljen tako del strani (Web Part) kot datoteka v SharePoint knjižnici itd.

ATRIBUT ALI METODA	OPIS
Web	Objekt tipa SPWeb, ki predstavlja stran, na kateri se nahaja ta datoteka.
Name	Ime datoteke, vključno z njeno končnico ¹⁴ .
Url	Relativni naslov URL datoteke.
Length	Dolžina datoteke v bajtih ¹⁵ .
Convert()	Funkcija za pretvarjanje datoteke iz enega formata v drugega.
Exists	Pritrdilna vrednost, ali datoteka obstaja.
Delete()	Funkcija za brisanje datoteke.
Author	Objekt tipa SPUser, ki predstavlja uporabnika, ki je kreiral to datoteko.
CheckOut(...)	Funkcija, ki datoteko vzame iz SharePoint knjižnice na ta način, da jo drugi ne vidijo več.
CheckIn(...)	Funkcija, ki prikaže datoteko nazaj v SharePoint knjižnici.
MoveTo(...)	Funkcija za premik neke datoteke iz ene lokacije na drugo.

Tabela 8: Najbolj uporabljeni atributi oz. metode objekta tipa SPFile

```

public void Page_Load(object sender, EventArgs e)
{
    // poveži se na zbirko strani, katere ime je vpisano v
    // spremenljivki _siteName
    using (SPSite _site = new SPSite(_siteName))
    {
        // Poveži se s trenutno odprto stran
        using (SPWeb _web = _site.OpenWeb())
        {
            // Poveži se na SharePoint seznam z imenom Izboljsave

```

¹⁴ Vsaka datoteka ima svojo končnico. Končnica predstavlja tip datoteke, iz česar lahko predvidevamo njeno vsebino in program, s katero jo lahko preberemo ali poženemo. Tipično je sestavljena iz treh črk. Skupaj z imenom datoteke pa tvori niz v naslednji obliki: {ime_datoteke}.{koncnica_datoteke}

¹⁵ Byte (bajt) je v računalništvu enota za merjenje količine podatkov oz. velikost pomnilnika [18].

```

        SPList _list = _web.Lists["Izboljsave"];

        // Iz seznama poberi prvi element
        SPListItem _item = _list.Items[0];

        // Poberi datoteko, ki jo vsebuje element seznama
        SPFile _file = _item.File;

        Izpiši podatke datoteke
        StringBuilder _izpis = new StringBuilder();
        _izpis.AppendFormat("Author: {0}\n",
            _file.Author);
        _izpis.AppendFormat("URL: {0}\n",
            _file.Url);
    }
}
}

```

3.3.9 SPField

Objekt SPField predstavlja stolpec SharePoint seznama.

ATRIBUT ALI METODA	OPIS
Type	Objekt tipa SPFieldType, s katerim lahko določimo ali preverimo tip podatka, ki ga določa stolpec.
Title	Ime stolpca.
Required	Ali je podatek v stolpcu obvezen ali ne.
DisplaySize	Dolžina prikaza vsebine stolpca.
DefaultValue	Privzeta vrednost stolpca.

Tabela 9: Najbolj uporabljeni atributi oz. metode objekta tipa SPField

```

public void Page_Load(object sender, EventArgs e)
{
    // poveži se na zbirko strani, katere ime je vpisano v
    // spremenljivki _siteName
    using (SPSite _site = new SPSite(_siteName))
    {
        // Poveži se s trenutno odprto stran
        using (SPWeb _web = _site.OpenWeb())
        {
            // Poveži se na SharePoint seznam z imenom Izboljsave
            SPList _list = _web.Lists["Izboljsave"];

            StringBuilder _izpis = new StringBuilder();

            // Pojdi čez vse stolpce SharePoint seznama in izpiši
            // njihove podatke
            foreach (SPField _field in _list.Fields)
            {
                _izpis.AppendFormat("Ime stolpca: {0}\n",
                    _field.Title);
            }
        }
    }
}

```

```

        _izpis.AppendFormat("Obveznost podatka: {0}\n",
            _field.Required);
    }
}
}
}
}

```

3.3.10 SPQuery

Objekt tipa SPQuery omogoča konstruirati poizvedbo, s katero lahko določimo, kateri element seznama bi radi dobili iz nekega SharePoint seznama.

ATRIBUT ALI METODA	OPIS
Query	Poizvedba v jeziku CAML. Več o pisanju poizvedb v jeziku CAML bom predstavil v nadaljevanju.
Folder	Poizvedba nad datotekami v SharePoint knjižnicah.
ViewFields	Stolpci, ki jih naj vrne poizvedba v jeziku CAML.
RowLimit	Maksimalno število vrstic, ki jih vrne poizvedba v jeziku CAML.
ViewAttributes	Drugi parametri poizvedbe v jeziku CAML, kot so npr. ali delamo poizvedbo samo na enem nivoju datotek in map, ali naj se poizvedba izvede tudi rekurzivno v podmapah.

Tabela 10: Najbolj uporabljeni atributi oz. metode objekta tipa SPQuery

Platforma SharePoint za poizvedovanje po SharePoint seznamih uporablja poizvedovalni jezik **Collaborative Application Markup Language (CAML)**. Poizvedbe v jeziku CAML bom predstavil zgolj na kratko, saj to ni poglobljena tema moje diplomske naloge, poleg tega pa ga s SharePointom verzije 2010 počasi zamenjuje popularni LINQ.

CAML uporablja strukturo XML. Če npr. hočemo v nekem SharePoint seznamu poiskati vse elemente seznama, kjer je stolpec Naslov enak »Test«, bomo uporabili spodnji blok poizvedbe CAML:

```

<OrderBy>
  <FieldRef Name='Naslov' />
</OrderBy>
<Where>
  <Contains>
    <FieldRef Name='Naslov' />
    <Value Type='Text'>Test</Value>
  </Contains>
</Where>

```

XML element *<Contains>* predstavlja operator, s katerim SharePointu naročimo, naj nam poišče tiste elemente seznama, ki imajo v stolpcu *Naslov* (*<FieldRef Name="Naslov" />*) tekstovno vrednost »Test« (*<Value Type="Text">Test</Value>*).

Če želimo pri poizvedbi uporabiti več pogojev, lahko to storimo tako, da v poizvedbi tipa CAML vrinemo operatorje *<EQ>* za enako, *<NEQ>* za neenako, *<GT>* za več kot, *<GEQ>*

za več ali enako, `<LT>` za manj kot, `<IsNull>` če je null in `<BeginsWith>` če se začne z nekim željenim nizom.

```
public void Page_Load(object sender, EventArgs e)
{
    SPQuery _allQuery = new SPQuery();
    _allQuery.Query =
        "<OrderBy>" +
        "<FieldRef Name='Vodja' Ascending='True' />" +
        "</OrderBy>" +
        "<Where>" +
        "<And>" +
        "<Eq>" +
        "<FieldRef Name='Vidno' />" +
        "<Value Type='Boolean'>1</Value>" +
        "</Eq>" +
        "<Neq>" +
        "<FieldRef Name='Zaklenjen' />" +
        "<Value Type='Boolean'>1</Value>" +
        "</Neq>" +
        "</And>" +
        "</Where>";
}
```

3.3.11 SPUser

Objekt tipa SPUser predstavlja nekega SharePoint uporabnika.

ATRIBUT ALI METODA	OPIS
Name	Ime uporabnika.
Email	Elektronski naslov uporabnika.
ID	Identifikacijska številka uporabnika.
Xml	Niz, ki predstavlja podatke uporabnika v strukturi XML.
Groups	Objekt tipa SPGroupCollection, v katerem se nahaja seznam objektov tipa SPGroup objektov, ki predstavljajo vse uporabniške skupine, v katerih se nahaja uporabnik.
LoginName	Uporabniško ime uporabnika.
Alerts	Objekt tipa SPAlertCollection, v katerem se nahaja seznam objektov tipa SPAlert, ki pripadajo določenemu uporabniku.
SID	Unikatna ID, s katero je uporabnik predstavljen v imeniku AD.

Tabela 11: Najbolj uporabljeni atributi oz. metode objekta tipa SPUser

```
public void Page_Load(object sender, EventArgs e)
{
    // poveži se na zbirko strani, katere ime je vpisano v
    // spremenljivki _siteName
    using (SPSite _site = new SPSite(_siteName))
    {
        // Poveži se s trenutno odprto stran
        using (SPWeb _web = _site.OpenWeb())
```

```

    {
        // Pridobi trenutnega uporabnika
        SPUser _user = _web.CurrentUser;

        // Izpiši attribute uporabnika
        StringBuilder _izpis = new StringBuilder();
        _izpis.AppendFormat("ID uporabnika: {0}\n", _user.ID);
        _izpis.AppendFormat("Število skupin: {0}\n",
            _user.Groups.Count);
    }
}

```

3.3.12 SPException

Objekt tipa SPException predstavlja izjeme, ki jih je sprožil SharePoint. Z njimi lahko ugotovljamo napake med samim izvajanjem naše kode.

ATRIBUT ALI METODA	OPIS
ErrorCode	Unikatna številka, ki predstavlja vrsto napake.
NativeErrorMessage	Sporočilo o napaki.
NativeStackTrace	Skladovna sled, s katere je mogoče ugotoviti izvor napake.

Tabela 12: Najbolj uporabljeni atributi oz. metode objekta tipa SPException

3.3.13 SPUtility

Objekt tipa SPUtility predstavlja ogromno množico uporabnih metod, ki jih poleg vseh zgornjih še lahko uporabljamo v platformi SharePoint.

ATRIBUT ALI METODA	OPIS
Redirect(...)	Preusmerimo se na drugo stran.
SendEmail(...)	Pošljemo elektronsko pošto iz SharePoint strani.
FormatDate(...)	Funkcija, ki vrne formiran datum.
ParseDate(...)	Pretvori formiran datum v objekt tipa DateTime.

Tabela 13: Najbolj uporabljeni atributi oz. metode objekta tipa SPUtility

3.4 SHAREPOINTOVI ODZIVNIKI NA DOGODKE (EVENT RECEIVERS)

Odzivnike na dogodke se lahko uporablja ob izvedbi nekega dogodka na SharePoint seznamu, elementu seznama, strani ali delovnem toku [1]. Kot primer uporabe vzemimo SharePoint knjižnico, v katero naložimo nek dokument. Ob tem hočemo, da je vsakič, ko nekdo naloži dokument v to knjižnico, ta dokument pregledan od določene osebe. Ko je to storjeno, se dokument prenese v neko drugo SharePoint knjižnico.

Odzivnike v SharePointu delimo na:

- **SINHRONI:** sprožijo se, preden se nek dogodek izvede (ItemAdding, ItemDeleting ...);
- **ASINHRONI:** sprožijo se potem, ko se je nek dogodek že izvedel (ItemAdded, ItemDeleted ...).

ATRIBUT	OPIS
Cancel	Določa, ali je bil nek dogodek preklican.
ErrorMessage	Vpišemo lahko sporočilo, ki se prikaže, ko je bil dogodek preklican.
EventType	Vrednost, ki določa tip dogodka (ItemAdding, ItemDeleting itd.).
ReceiverData	Niz, ki vsebuje podatke odzivnika.
RedirectUrl	Določa preusmeritveni naslov URL v primeru preklica dogodka.
Status	Ima vrednost "Continue", če dogodek ni preklican. Ima pa lahko še vrednosti "CancelWithError" in "CancelWithRedirectUrl".

Tabela 14: Najbolj uporabljeni atributi, ki veljajo za vse odzivnike na dogodke

3.4.1 Dogodki na SharePoint seznamih

Na SharePoint seznamih lahko uporabljamo **SPListEventReceiver** odzivnik na naslednje dogodke:

DOGODEK	OPIS
FieldAdded	Dodaj nov stolpec.
FieldAdding	Dodaja se nov stolpec.
FieldDeleted	Stolpec izbrisan.
FieldDeleting	Stolpec se briše.
FieldUpdated	Stolpec posodobljen.
FieldUpdating	Stolpec se posodablja.
ListAdded	SharePoint seznam dodan.
ListAdding	SharePoint seznam se dodaja.
ListDeleted	SharePoint seznam izbrisan
ListDeleting	SharePoint seznam se briše

Tabela 15: Dogodki na SharePoint seznamih

3.4.2 Dogodki na elementih SharePoint seznama

Odzivnike na dogodke lahko uporabljamo tudi na samih elementih SharePoint seznamov. Ti dogodki so lahko naslednji:

DOGODEK	OPIS
ItemAdded	Dodan nov element.
ItemAdding	Dodaja se nov element.
ItemDeleted	Element izbrisan.
ItemDeleting	Element se briše.
ItemCheckedIn	Element je postal viden ostalim.
ItemCheckingIn	Element se pripravlja na to, da bo viden ostalim.
ItemCheckedOut	Element je postal neviden ostalim.
ItemCheckingOut	Element se pripravlja na to, da bo neviden ostalim.
ItemUpdated	Element seznama posodobljen.
ItemUpdating	Element seznama se posodablja.
ItemAttachmentAdded	Elementu seznama je dodana priloga.
ItemAttachmentAdding	Elementu seznama se dodaja priloga.
ItemAttachmentDeleted	Elementu seznama je izbrisana priloga.
ItemAttachmentDeleting	Elementu seznama se briše priloga.

Tabela 16: Dogodki na elemente SharePoint seznama

3.4.3 Dogodki na elektronsko pošto

Odzivnike na dogodke pri samih SharePoint seznamih lahko uporabljamo še na način, ko za dogodek vzamemo prejem elektronskega sporočila v nek SharePoint seznam. Ta dogodek imenujemo **EmailReceived**.

```
public override void EmailReceived(SPList list, SPEmailMessage
    emailMessage, string receiverData)
{
    Base.EmailReceived(list, emailMessage, receiverData);
}
```

SPEmailMessage je tip objekta, ki predstavlja elektronsko pošto, ki je bila poslana SharePointovemu seznamu. Preko njega se lahko dobi dele elektronskega sporočila, bere podatke pošiljatelja, priloge itd.

3.4.4 Dogodki na strani

Odzivnike na dogodke lahko uporabljamo tudi na SharePoint straneh in zbirkah strani. Ti dogodki so lahko naslednji:

DOGODEK	OPIS
SiteDeleted	Zbirka strani je bila izbrisana.
SiteDeleting	Zbirka strani se briše.
WebAdding	Stran se dodaja.
WebDeleted	Stran je bila izbrisana.

WebDeleting	Stran se briše.
WebMoved	Stran je bila premaknjena.
WebMoving	Stran se premika.

Tabela 17: Dogodki na elektronsko pošto

3.4.5 Dogodki na delovne tokove

Odzivnike na dogodke pri samih SharePoint seznamih lahko uporabljamo še na en način, ko za izvor dogodkov vzamemo SharePointove delovne tokove. Tu imamo naslednje dogodke na delovnih tokovih:

DOGODEK	OPIS
WorkflowCompleted	Delovni tok se je končal.
WorkflowStarted	Delovni tok se je začel.
WorkflowStarting	Delovni tok se bo začel.
WorkflowPostponed	Delovni tok preložen.

Tabela 18: Dogodki na delovnih tokovih

3.5 DELO Z ZUNANJIMI PODATKI (BUSINESS DATA)

Pridemo lahko do situacije, ko se nam pri delu s platformo SharePoint pojavi želja po tem, da bi radi v njej uporabljali tudi nekatere podatke, ki so shranjeni izven nje. Kot primer takih podatkov v nekem podjetju lahko vzamemo podatke o kupcih v sistemu CRM¹⁶. Ker lahko platformo SharePoint uporabljamo kot intranet, bi bilo lepo, če bi lahko nek zaposleni znotraj podjetja kar preko nekega SharePoint seznama ustvarjal, bral, posodabljal in brisal podatke v sistemu CRM. Vse to je v SharePointu mogoče preko t.i. Business Data Connectivity oz. storitve BCS [2].

Ko v Visual Studiu ustvarimo nov projekt BCS, dejansko ustvarjamo storitev, kjer določimo, kako se bodo tabele iz neke zunanje podatkovne baze preslikovale v SharePointove sezname (kaj je ključ tabele, kakšni so atributi itd.). To nam omogoča, da lahko podatke iz zunanjih tabel enostavno urejamo v SharePointu kar preko SharePoint seznamov, ki jih BCS ustvari.

V BCS za posamezno tabelo zunanje podatkovne baze ustvarimo preslikano tabelo, ki predstavlja SharePoint seznam. Tej moramo ustvariti attribute (stolpce tabele), kot jih ima zunanja tabela, ter določiti, kateri izmed njih je oz. so ključi zunanje tabele in torej tudi te, preslikane tabele.

Nato imamo na voljo pet metod, preko katerih določimo, kako in kateri atributi iz zunanje tabele se bodo prepisali v attribute te preslikane tabele:

¹⁶ Customer Relationship Management (CRM) je strategija v podjetju, ki pozornost osredotoča na odnose s strankami in ne toliko na artikle oz. produkte. Tako se strategija osredotoča na strankine kupne navade, podatke o stranki, komunikacije s stranko itd. Upravljanje odnosov s strankami se v večini že podpira tudi s programskimi rešitvami t.i. CRM sistemi, ki omogočajo še lažje in hitrejše delo [19].

METODE	OPIS
Finder	Vrne seznam vseh elementov tabele. <i>IEnumerable<EntityType> GetEntities()</i>
Specific Finder	Vrne element tabele. <i>EntityType GetEntityById(IdType1 id1, etc ...)</i>
Creator	Ustvari nov element tabele. <i>void CreateEntity(EntityDataType fields)</i>
Updater	Posodobi obstoječi element tabele. <i>void UpdateEntity(EntityDataType fields)</i>
Deleter	Izbriše obstoječi element tabele. <i>void DeleteEntity(idType id1, etc ...)</i>

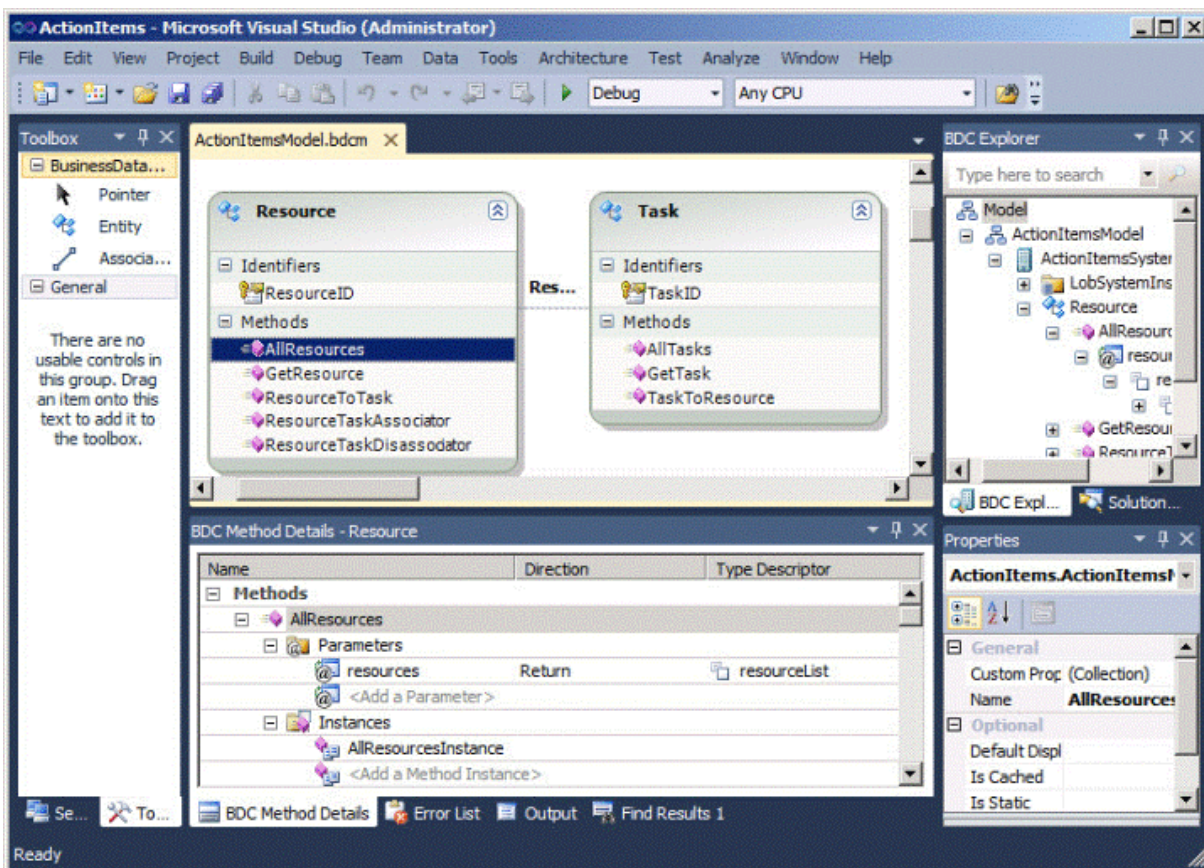
Tabela 19: Metode za preslikavo parametrov iz zunanje tabele v preslikano tabelo

Dejansko s projektom BCS samo ustvarimo preslikane tabele, ki predstavljajo SharePoint sezname, in določimo, kako se atributi zunanjih tabel preslikujejo vanj. Za dejansko povezavo projekta BCS z zunanjo podatkovno bazo pa potrebujemo eno izmed možnih funkcionalnosti, ki nam jih Visual Studio omogoča. Sam sem tu uporabil že omenjeno komponento LINQ.

Za lažje razumevanje spodnja slika še grafično pojasnjuje samo delovanje povezave platforme SharePoint z zunanjimi podatki v navezi s komponento LINQ.



Slika 10: Povezava platforme SharePoint z zunanjo podatkovno bazo v navezi s komponento LINQ

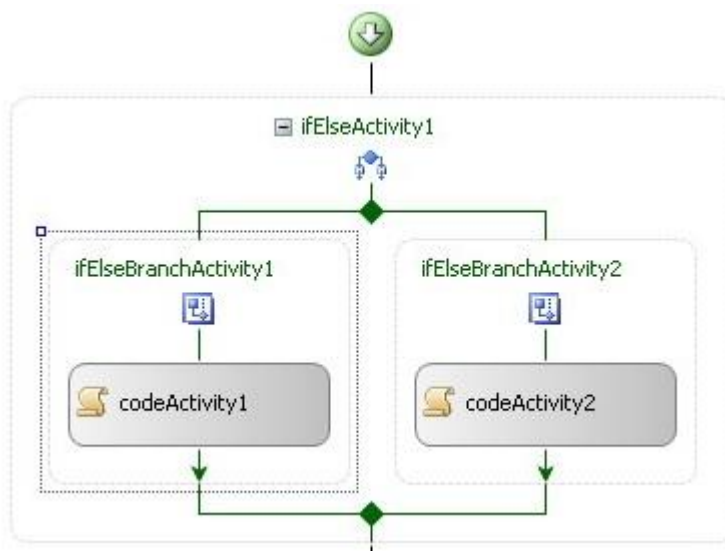


Slika 11: BCS projekt v programskem orodju Visual Studio 2010

3.6 SHAREPOINTOVI DELOVNI TOKOVI (WORKFLOWS)

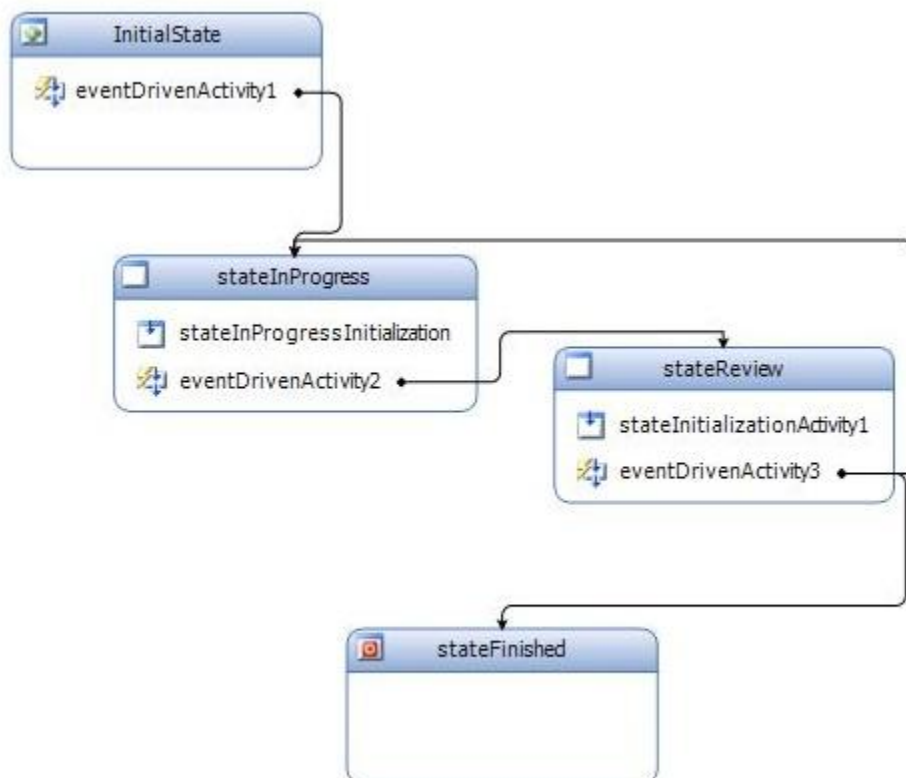
Sharepointovi delovni tokovi omogočajo izvajanje procesov z več koraki ali pa procesov z več stanji. Delovni tok je po definiciji nek proces, ki je dolgo trajajoč, čakajoč na človeka, zunanji sistem ali nek drug proces, da opravi neke naloge. Ta vsebuje eno ali več aktivnosti, ki se morajo izvesti v nekem vrstnem redu, da se s tem izvede njegova funkcija [1].

Delovni tok si lahko predstavljamo kot naš delovni dan. Zjutraj se zbudimo, popijemo kavo ali čaj, se peljemo v službo, gremo na kosilo itd. Če te aktivnosti vzamemo kot zaporedje nalog, ki se morajo v takšnem vrstnem redu tudi zgoditi, imamo opravka s **SEKVENČNIM DELOVNIM TOKOM**. Ima veliko podobnost z algoritmi.



Slika 12: Prikaz sekvenčnega delovnega toka

Delovni tok si lahko organiziramo kot stanja, med katerimi imamo povezave, ki določajo, kako se lahko med temi stanji premikamo, premik iz enega stanja v drugega pa sproži nek dogodek. Imamo začetno in končno stanje. Takemu delovnemu toku pravimo **DELOVNI TOK S STANJI**. Uporablja se jih v bolj kompleksnih poslovnih procesih, kjer bi proces težko določili z zaporedjem nalog in to lažje storimo s stanji in povezavami med njimi.



Slika 13: Prikaz delovnega toka s stanji

Aktivnosti v nekem delovnem toku lahko razdelimo v tri skupine:

- **KONTROLNE AKTIVNOSTI (CONTROL FLOW ACTIVITIES):** uporabljajo se za kontroliranje/usmerjanje toka izvajanja aktivnosti. Mednje štejemo aktivnosti While, IfElse itd;
- **VSEBOVANE AKTIVNOSTI (CONTAINER ACTIVITIES):** predstavljajo skupek več aktivnosti;
- **STANDARDNE AKTIVNOSTI (STANDARD ACTIVITIES):** vsebujejo neke naloge. To je dejansko neka izvorna koda.

Najbolj uporabljene aktivnosti delovnih tokov iz vseh treh skupin so:

METODE	OPIS
Delay	Aktivnost, ki zaustavi delovanje delovnega tokova za določen čas.
IfElse	Aktivnost s popolnoma enakim obnašanjem kot <i>if then else</i> stavek v drugih programskih jezikih.
Listen	Aktivnost, namenjena poslušanju zunanjih aktivnosti. To lahko torej uporabljamo za čakanje, da se neka druga aktivnost izvede.
Parallel	Aktivnost se uporablja za izvajanje dveh ali več vej hkrati. Ko se ena izmed njih izvede, mora ta počakati še vse ostale, da dokončajo svoje delo.
Terminate	Aktivnost se uporablja za zaustavitev delovnega tokova.
While	Aktivnost s popolnoma enakim obnašanjem kot while stavek v drugih programskih jezikih.
CompleteTask	Aktivnost, ki konča neko opravilo v SharePointovemu seznamu za opravila.
CreateTask	Aktivnost, ki kreira novo opravilo.
SendEmail	Aktivnost, ki pošlje elektronsko pošto.
CreateItemActivity	Aktivnost, ki kreira nov element SharePoint seznama.
DeleteItemActivity	Aktivnost, ki izbriše že obstoječi element seznama.

Tabela 20: Najbolj uporabljene aktivnosti delovnih tokov v SharePointu

3.7 SHAREPOINTOVE SPLETNE STORITVE (WEB SERVICES)

Ker je SharePoint dejansko platforma, omogoča tudi delo s spletnimi storitvami. BCS, o katerem sem že govoril, je primer SharePointove spletne storitve. Spletna storitev seveda predstavlja aplikacijo, ki nima uporabniškega vmesnika, za svoje delo ne potrebuje uporabnika – torej je samostojna, se lahko sama zažene, izvede nalogo in zaključi.

V SharePointu lahko spletne storitve uporabljamo na dva načina. Prvi je ta, da spletno storitev omejimo samo na delovanje znotraj omrežja podjetja in omogoča uporabo funkcionalnosti platforme SharePoint preko drugih aplikacij, ki jim spletna storitev posreduje podatke, oz. jih za njih shranjuje v platformo SharePoint. Drugi način pa ustvari most tudi v svetovni splet,

kjer lahko spletna storitev pošilja zunanjemu svetu podatke, ki niso strogo zaupni – npr. novice korporacije, ki so lahko javno dostopne.

Spletna storitev lahko za komunikacijo med platformo SharePoint in neko zunanjo aplikacijo oz. odjemalcem uporablja več različnih poti, po katerih lahko komunicira, kot so Hypertext Transfer Protokol (HTTP), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) itd.

V SharePointu najdemo že vgrajene storitve, lahko pa jih spišemo sami. SharePoint temelji na ASP.NET ogrodju, zato lahko uporabljamo njegove storitve. Sicer pa podpira še storitve Windows Communication Foundation (WCF), storitve RESTful in storitve za računalništvo v oblaku [1].

Do vgrajenih storitev ASP.NET SharePointa lahko dostopamo z naslednjo obliko naslova URL:

http://<site>/_vti_bin/<Web Service Name>.asmx

Kot primer navedimo dostop do vgrajenih storitev za delo s SharePoint seznamami, kjer je naslov URL takšen:

http://<site>/_vti_bin/lists.asmx

Lists

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [AddAttachment](#)
- [AddDiscussionBoardItem](#)
- [AddList](#)
- [AddListFromFeature](#)
- [AddWikiPage](#)
- [ApplyContentTypeToList](#)
- [CheckInFile](#)
- [CheckOutFile](#)
- [CreateContentType](#)
- [DeleteAttachment](#)
- [DeleteContentType](#)
- [DeleteContentTypeXmlDocument](#)
- [DeleteList](#)
- [GetAttachmentCollection](#)
- [GetList](#)

Slika 14: Prikaz nekaterih vgrajenih storitev za delo s SharePoint seznamami

V Visual Studiu do funkcionalnosti teh vgrajenih storitev dostopamo tako, da v projekt dodamo storitveno referenco na njih. Lahko pa seveda spišemo tudi svojo storitev ASP.NET.

3.8 VARNOST V SHAREPOINTU

SharePoint 2010 ima varnostno infrastrukturo urejeno na visokem nivoju in podpira mnogo tehnologij kot so Active Directory (AD), Secure Store Service (SSS), Claims-Based Authentication (CBA), Forms-Based Authentication (FBA), Kerberos itd. [3].

Če hočemo določiti pravice posameznim ljudem ali skupinam ljudi znotraj organizacije je tukaj seveda najboljša odločitev uporaba imenika AD¹⁷. V primeru, da hočemo določiti pravice še uporabnikom nekega drugega zunanjsega sistema, uporabimo tehnologijo SSS ali CBA.

Secure Store Service (SSS) je SharePointova storitev, ki omogoča preslikavo uporabnikov iz nekega zunanjsega sistema v platformo SharePoint. Kot primer vzemimo, da imamo uporabnika Janeza Novaka, ki ima svoj uporabniški račun v imeniku AD, poleg tega pa ima še uporabniški račun v nekem drugem ločenem sistemu. SSS v tem primeru omogoča povezavo teh dveh uporabniških računov.

Forms-Based Authentication (FBA) je najbolj znana še iz ogrodja ASP.NET. Omogoča uporabnikom dostop preko neke prijavnne strani. V ozadju uporablja podatkovno bazo z uporabniki in vlogami, s čimer je določeno, kdo se lahko prijavi in kakšne pravice ima. Preverjanje istovetnosti poteka s pomočjo piškotkov.

Claims-Based Authentication (CBA) predstavlja močno in fleksibilno tehnologijo za preverjanje istovetnosti uporabnikov, saj deluje z velikim številom identifikacijskih sistemov, ki so lahko v neki organizaciji, kot je npr. imenik AD, Lightweight Directory Access Protocol (protokol LDAP), modeli Web-centric, kot so Microsoftov LiveID itd. Pri tem uporablja protokol Security Assertion Markup Language (SAML), ki temelji na osnovi XML in je standard za izmenjavo prijavnih podatkov med dvema sistemoma za identifikacijo.

¹⁷ Active Directory (imenik AD) oz. aktivni imenik je skupek vseh uporabnikov v Windows okolju. S takim imenikom pridobimo centralno avtentikacijo za vse računalnike z operacijskim sistemom Windows. Prav tako omogoča administratorjem dodeljevanje pravic in nastavljanje pravil, nameščanje programske opreme in kritičnih nadgradenj ter popravkov celotnemu podjetju. Vse te podatke hrani v centralni podatkovni bazi [20].

Create New Web Application

Authentication

Select the authentication for this web application.

[Learn about authentication.](#)

Claims Based Authentication

Classic Mode Authentication

IIS Web Site

Choose between using an existing IIS web site or create a new one to serve the Microsoft SharePoint Foundation application.

If you select an existing IIS web site, that web site must exist on all servers in the farm and have the same name, or this action will not succeed.

If you opt to create a new IIS web site, it will be automatically created on all servers in the farm. If an IIS setting that you wish to change is not shown here, you can use this option to create the basic site, then update it using the standard IIS tools.

Use an existing IIS web site

Default Web Site

Create a new IIS web site

Name

SharePoint - 11376

Port

11376

Host Header

Path

C:\inetpub\wwwroot\wss\VirtualDirecto

Security Configuration

If you choose to use Secure Sockets Layer (SSL), you must add the certificate on each server using the IIS administration tools. Until this is done, the web application will be inaccessible from this IIS web site.

Allow Anonymous

Yes

No

Use Secure Sockets Layer (SSL)

Yes

No

Claims Authentication Types

Choose the type of authentication you want to use for this zone.

Negotiate (Kerberos) is the recommended security configuration to use with Windows authentication. If this option is selected and Kerberos is not configured, NTLM will be used. For Kerberos, the application pool account needs

Enable Windows Authentication

Integrated Windows authentication

NTLM

Basic authentication (credentials are sent in clear text)

Slika 15: Izbira varnostnih tehnologij v Centralni administraciji SharePointa

3.9 POVEZOVANJE S PAMETNIMI TELEFONI

Pri povezovanju platforme SharePoint s pametnimi telefoni sem se osredotočil zgolj na pametne telefone z operacijskim sistemom Windows Phone 7 (WP7), saj imam tak telefon tudi sam. V prihodnosti pa bom raziskal še povezovanje z drugimi operacijskimi sistemi pametnih telefonov, saj je to področje, ki me zelo zanima.

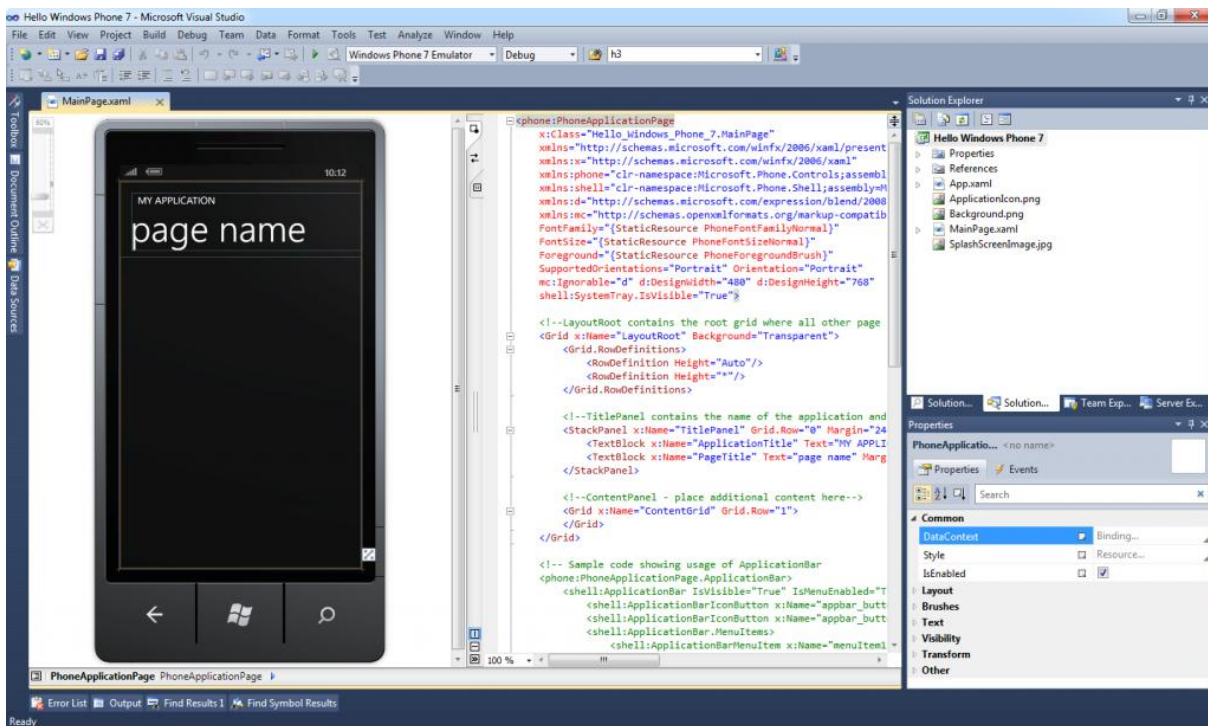
Operacijski sistem Windows Phone 7 sicer omogoča integracijo s platformo SharePoint preko programske opreme Office Hub.



Slika 16: Windows Phone 7 Office Hub

Ampak Office Hub ponuja zgolj ogled SharePoint seznamov v seznamih operacijskega sistema Windows Phone 7. To nam sicer omogoča, da lahko prenašamo datoteke in beremo podatke, ki so shranjeni v SharePoint seznamih, vendar je to tudi vse. Za kaj več je seveda potrebno narediti svojo aplikacijo. Z njo bi tako lahko npr. v prilagojenem uporabniškem vmesniku prikazovali novice z intraneta, prenašali dokumente, upravljali SharePoint aplikacije itd.

Aplikacije za operacijski sistem Windows Phone 7 lahko, tako kot vse druge Microsoftove tehnologije, programiramo s programskim orodjem Visual Studio.



Slika 17: Programiranje aplikacij za OS Windows Phone 7 s programskim orodjem Visual Studio 2010

Prvi problem pri povezovanju WP7 aplikacije s platformo SharePoint predstavlja avtentikacija. Ker operacijski sistem WP7 za avtentikacijo ne podpira protokola NT LAN Manager (NTLM), moramo na platformi SharePoint omogočiti Forms-Based Authentication (avtentikacijo FBA), ki sem jo omenil v prejšnjem poglavju. Tako bomo lahko v WP7 aplikaciji za preverjanje istovetnosti uporabili uporabniška imena in gesla, ki jih bomo generirali za FBA.

Za integracijo SharePointa in WP7 aplikacije lahko uporabimo storitve SharePointa. Za avtentikacijo uporabimo storitev "http://<site>/_vti_bin/authentication.asmx", za dostop do SharePointovih seznamov pa že omenjeno storitev "http://<site>/_vti_bin/lists.asmx". Ker uporabljamo avtentikacijo FBA, je potrebno v WP7 aplikaciji omogočiti uporabo piškotkov [6].

Pri integraciji SharePointa s WP7 aplikacijo lahko uporabimo tudi t.i. potisna sporočila, ki jih nudi Microsoft Push Notification Service. To nam omogoča, da dobi uporabnik te aplikacije na pametnem telefonu obvestilo, da je bila neka vsebina na SharePointu dodana, spremenjena ali odstranjena, takoj ko je do tega tudi prišlo. To daje le še dodaten plus na integracijo SharePointa s pametnimi telefoni, saj lahko na tak način pohitrimo, poenostavimo in mobiliziramo marsikatero nalogo v poslovnih procesih podjetja.

4 SHAREPOINT PROJEKTI

V tem poglavju želim predstaviti vse svoje projekte, ki so nastali kot plod mojih raziskav na področju programiranja v platformi SharePoint in vsebujejo programske rešitve za procese, ki se izvajajo v korporaciji HIDRIA.

Več o vzrokih, problematiki, poteku izdelave ter samem videzu vseh projektov pa v nadaljevanju; vsak projekt v svojem ločenem podpoglavju. Začel bom s spletno aplikacijo, ki je namenjena prijavi in obdelavi izboljšav, nadaljeval bom z aplikacijo za nadzor terminalov ter zaključil z dodatkom za grafični prikaz SharePoint opravil ter gradnikom za izbiro oseb za platformo SharePoint.

4.1 APLIKACIJA ZA PRIJAVO IN OBDELAVO IZBOLJŠAV

Naročnik, natančneje oddelek za inovacijsko kulturo znotraj korporacije HIDRIA, je želel vzpostaviti sistem za vsa podjetja znotraj korporacije, preko katerega bo lahko kdorkoli izmed vseh zaposlenih prijavil neko izboljšavo.

Izboljšava je vsak izveden koristen predlog v delovnem okolju, procesu, izdelku ali tehnologiji, ki prinaša materialno ali nematerialno korist. Ne smemo je enačiti z inovacijo. Neka izboljšava je inovacija samo v primeru, ko imamo nek bistveno spremenjen izdelek, tehnologijo ali proces, ki je novost na trgu, prinaša večjo prodajo, možnost patentiranja, konkurenčno prednost na trgu ali nižje stroške in optimizirano poslovanje.

4.1.1 Vzroki za nastanek aplikacije

Na HIDRII razvijamo unikaten, fleksibilen model, ki spodbuja razvoj drznih, tržno uspešnih rešitev na vseh področjih delovanja. Korak na poti do tega predstavlja tudi ta aplikacija za prijavo in obdelavo izboljšav.

Aplikacija obsega 7 modulov, ki so med seboj povezani na različne načine v procesu prijave in obdelave izboljšave:

- prijave izboljšave,
- potrditev vodje inovativnega jedra,
- potrditev skrbnika izboljšav,
- spremljanje izvajanja izboljšave,
- izračun prihranka izboljšave,
- izračun nagrade izboljšave in
- zaključevanje izboljšave.

Prijaviteljev oz. predlagateljev neke izboljšave je lahko več. To dodatno zakomplicira sistem, saj moramo tako pri posamezni prijavi izboljšave spremljati in obdelati več oseb, ki so navedene med predlagatelji.

Potrditev izboljšav se porazdeli na inovativna jedra. Inovativno jedro je smiselno zaključena celota, v kateri poteka podoben ali med seboj povezan delovni proces. Vsako inovativno jedro ima svojega vodja – to je oseba, ki je neposredno odgovorna za načrtovanje in doseganje ciljev na področju izboljšav posameznega inovativnega jedra. Skrbi za motivacijo podrejenih, organizacijo dela in s tem tudi za zbiranje in izvedbo pobud in predlogov izboljšav. Vsak prijavitelj ima svojega vodjo inovativnega jedra. V korporaciji HIDRIA imamo trenutno približno 140 inovativnih jeder, torej 140 vodij inovativnih jeder, ki uporabljajo ta sistem in s tem potrjujejo posamezno izboljšavo, ki spada v njihovo inovativno jedro. Število inovativnih jeder se še povečuje, saj se jih trenutno še oblikuje.

Tu imamo še množico skrbnikov izboljšav. Skrbnik izboljšave je član oddelka za inovacijsko kulturo znotraj korporacije HIDRIA, ki je posebej zadolžen za strokovno delo na področju evidentiranja, ocenjevanja in nagrajevanja izboljšav, motiviranje vodij in zaposlenih ter za njihovo stalno izobraževanje in usposabljanje. Vsak od skrbnikov pokriva več inovativnih jeder oz. poslovnih enot.

Če ob vsem tem pomislimo, da je znotraj korporacije HIDRIA trenutno zaposlenih približno 2600 oseb, lahko ugotovimo, da je populacija uporabnikov te aplikacije ogromna. To le podpira mojo odločitev glede tega, da mora biti enostavnost uporabe ter preglednost te aplikacije na visokem nivoju in da tu z vgrajenimi univerzalnimi funkcionalnostmi SharePointa ni kaj početi.

S koncem leta 2011 je predvidena opustitev dosedanje rešitve prijave in obdelave izboljšav, ki jo bo nadomestila aplikacija na intranetu korporacije HIDRIA – t.i. HIDRIANET, ki temelji na SharePoint platformi. Dosedanja rešitev prijave in obdelave izboljšav temelji na množici papirnatih obrazcev, ki se jih obdeluje ročno s pomočjo Microsoftovega pisarniškega programa Excel¹⁸. Tak način spremljanja izboljšav pa predstavlja nemalo problemov.

Težave, s katerimi so se srečevali, bi lahko razdelili v tri ločene sklope:

- **težave skrbnikov izboljšav,**
- **težave predlagateljev in**
- **težave vodij inovativnih jeder.**

4.1.1.1 Težave skrbnikov izboljšav

Težave skrbnikov izboljšav so definitivno največje od vseh treh zgoraj naštetih sklopov, saj imajo v procesu prijave in obdelave izboljšav tudi največjo vlogo in s tem posledično največ dela.

¹⁸ Microsoft Excel je računalniški program za obdelavo razpredelnic. Trži ga podjetje Microsoft, ki ga je tudi razvilo.

Vsa korespondenca s predlagatelji in vodji inovativnih jeder je morala biti izvedena ročno, in sicer v obliki fizičnega stika z njimi ali pa preko prenosnih medijev, kot so elektronska pošta, telefonski klic, navadna pošta itd. Nikjer ni bilo nobene avtomatizacije, kot jo je moč najti v aplikaciji sedaj; vse je bilo potrebno narediti ročno.

Vsi zapisi ter beležke, ki so nastajale med obravnavo izboljšave so seveda v večini papirnate oblike, te pa je moral skrbnik trajno arhivirati in hraniti. Ni potrebno posebej poudarjati, da tak način dela prinaša nepreglednost in zmešnjavo.

V HIDRII nagrajujemo dobre pobude za izboljšave. Nagrada se določi glede na izračunano gospodarsko korist z upoštevanjem korekcijskih faktorjev. Tako gospodarsko korist kot nagrado je bilo potrebno izračunati ročno, za vse prijavitelje neke izboljšave, posebej glede na njihov procent soudeležbe. Tudi tu ni potrebno posebej poudarjati, da človek za to porabi več časa, saj vse računa ročno, ob tem pa lahko prihaja do napak v izračunih, saj ljudje nismo nezmotljivi.

Evidence prijavljenih, zavrnjenih, izvedenih in zaključenih izboljšav so seveda potekale na papirju, to pa prav tako prinaša nepreglednost in zmešnjavo.

Zgoraj navedene težave bi sicer lahko rešili tudi s pisarniško programsko opremo Microsoft Excel, česar so se naročniki v določeni meri tudi posluževali. Vendar prav tako s takim načinom dela ne dosežemo avtomatizacije na nivoju, kot bi si jo želeli in kot nam jo lahko ponuja specializirana aplikacija.

4.1.1.2 Težave predlagateljev

Predlagatelj oz. predlagatelji lahko svoje izboljšave prijavljajo s kartončkom »5 minut za izboljšavo«, na katerem opišejo težavo, ki so jo zaznali, in predlagajo rešitev. Njihove težave ob tem sicer niso tako velike kot to velja za zgoraj opisan sklop uporabnikov, vsekakor pa se v določeni meri pojavljajo tudi tukaj.

Izpolni predlagatelj						
5 minut za izboljšavo	Problem:					
	Rešitev:					
Podatki predlagatelja	os./mat. št.	priimek, ime	% soudeležbe	vodja inovativnega jedra	datum prijave	podpis predlagatelja

Slika 18: Sprednja stran kartončka "5 minut za izboljšavo"

Predlagatelj mora kartonček s prijavo izboljšave oddati svojemu vodji inovativnega jedra oz. vodji inovativnega jedra prvega predlagatelja, če je predlagateljev več. To seveda zahteva fizični stik z njimi, za kar se porabi več časa kot v primeru avtomatizacije procesa v sklopu aplikacije. Pojavi se lahko tudi problem, da se prijava izboljšave tukaj konča, ker ima predlagatelj določen strah pred vodjem ali pa je vzrok preprosto v tem, da se mu do vodje ne da pristopiti in ga to odvrne že od same želje, da bi izboljšavo sploh prijavil.

Pojavljajo se lahko tudi take težave, da bi si predlagatelji želeli prijaviti izboljšavo, vendar kartončka iz takih ali drugačnih vzrokov ne najdejo oz. ne dobijo. Pri tej aplikaciji pa so predlagatelji deležni enotne vstopne točke za prijavo izboljšav, ki je preko internih strani (intraneta) vedno na voljo.

Ker mora predlagatelj izpolniti kartonček, se lahko pojavijo problemi, ko nekega obveznega polja ne izpolni iz kakršnegakoli razloga – ali je nanj enostavno pozabil ali pa se mu ga ni ljubilo. To pri aplikaciji ni več mogoče, saj mu ta ne pusti oddati prijave, dokler ne izpolni vseh obveznih podatkov. Prav tako aplikacija pri določenih poljih preverja pravilnost vnosa in s tem zmanjša možnost napak zaradi napačnih podatkov.

4.1.1.3 Težave vodij inovativnih jeder

Težave vodij inovativnih jeder tudi niso tako velike kot težave skrbnikov izboljšav, vendar zato niso nič manj pomembne. Vodja inovativnega jedra dobi od predlagatelja, ki spada v njegovo inovativno jedro, prijavljeno izboljšavo na kartončku »5 minut za izboljšavo«. Za to prijavljeno izboljšavo mora nato opredeliti njeno ustreznost in glede na to izboljšavo sprejeti ali zavrniti. To označi na kartončku, ob tem pa mora na njem dopisati še svoj komentar na izboljšavo, označiti, ali je bila izboljšava že izvedena in ali je izračun prihranka in nagrade sploh potreben. Kartonček nato odda skrbniku izboljšav, ki pokriva tudi njegovo inovativno jedro.

izpolni vodja

Naslov izboljšave: _____ **št.:** _____ / _____
št. leto

NAGRADA - bruto zneski! (obkroži)

Stopnja izboljšave	Nagrada	Komentar vodje: _____
predlagatelj je izboljšavo sam izvedel	●	_____
predlagatelj je podal rešitev za realizacijo pobude (kako je treba narediti)	●	_____
predlagatelj je podal pobudo / predlog za izboljšavo (kaj je treba narediti)	●	_____

izboljšava je že izvedena

potreben izračun prihranka in nagrade

Priloge: DA NE (ustrezno označi)

Datum: _____

Vodja inovativnega jedra: _____
ime in priimek, podpis

5 minut za izboljšavo
5 minut za izboljšavo

Slika 19: Zadnja stran kartončka "5 minut za izboljšavo"

Prav tako se tudi tu pojavljajo težave pri korespondenci tako s predlagateljem oz. predlagatelji kot s skrbnikom izboljšav za njegovo inovativno jedro in jih je potrebno kolikor se le da tudi omiliti.

Tako kot predlagatelj tudi vodja inovativnega jedra izpolnjuje svoj del kartončka »5 minut za izboljšavo«. Veljajo podobne težave z obveznimi polji kot pri predlagatelju, zato je to potrebno reševati tudi tu.

4.1.2 Izgradnja aplikacije

Cilj naročnika je spodbuditi zaposlene znotraj korporacije HIDRIA, da bi ti predlagali čim več izboljšav. Postavili smo si cilj več 1000 prijavljenih izboljšav letno. Ker je to večja številka, kot je bilo prijavljenih izboljšav do sedaj, je za to potrebno zagotoviti ustrezno aplikacijo, ki bi jo lahko uporabniki zadovoljno, predvsem pa voljno uporabljali. Po pregledu težav, s katerimi so se soočali, smo ugotovili, da je nekaj potrebno ukreniti. Tako se je pojavila ideja o izgradnji aplikacije za prijavo in obdelavo izboljšav, ki bi te težave tudi rešila.

V nadaljevanju bom predstavil proces nastajanja same aplikacije. Tekom samega procesa smo izvedli tri sestanke, četrti in s tem zadnji v fazi nastajanja aplikacije pa še pride pred koncem leta 2011, ko bo izveden zadnji pregled aplikacije ter načrti za uvedbo oz. zagon aplikacije in s tem povezana izobraževanja odgovornih oseb, ki bodo s to aplikacijo tudi delale.

Na prvem sestanku sem se tako sploh prvič spoznal z naročniki, saj sem trenutno v korporaciji HIDRIA kot štipendist zgolj 6 let in v tem času še nisem imel možnosti spoznati vseh zaposlenih. Dorekli smo tehnične in vsebinske zahteve aplikacije ter sam proces prijave in obdelave izboljšav v obsegu, primernemu za enomesečno delo. Dobil sem tudi pravilnik inovacijske dejavnosti v HIDRII [5], s katerim sem si lahko pomagal pri razumevanju procesa prijave in obdelave izboljšav. Pri tehničnih zahtevah aplikacije so mi pustili dokaj proste roke in s tem odgovornost, kar je pomenilo, da sem moral vsebinske zahteve ter proces prijave in obdelave izboljšav dobro naštudirati. Le tako je lahko aplikacija dovolj fleksibilna in nadgradljiva ob spremembah obstoječih funkcionalnosti ter dodajanju novih, poleg tega pa ob upoštevanju vseh zahtev tudi enostavna za uporabo. Ker je bila to sploh moja prva aplikacija, narejena za platformo SharePoint, sem zato moral biti pripravljen na to, da bom za izdelavo potreboval malce več časa kot bi ga sicer. Delo sem tako v celoti opravil v dogovorjenem času, kar mi je vneslo še dodaten zagon, ki sem ga že tako ali tako imel zaradi navdušenja ob spoznavanju platforme SharePoint.

V dogovorjenem roku, mesec dni od prvega sestanka, smo organizirali drugi sestanek, kjer so lahko naročniki prvič tudi videli sam koncept delovanja in videz aplikacije, seveda v takem obsegu, kot smo ga določili na prvem sestanku. Ta obseg je pokrival dva od sedmih modulov, ki sem jih definiriral že pri vzrokih za nastanek aplikacije, in sicer:

- prijava izboljšave in
- potrditev vodje inovativnega jedra.

Tako so prišli prvi popravki že obstoječih funkcionalnosti, saj je bilo sedaj lažje te tudi najti ob samem pregledu in testiranju že narejenega dela projekta. Prav tako smo dorekli zahteve ter sam proces delovanja aplikacije za naprej v dogovorjenem obsegu.

V roku dobrega meseca dni je prišel tako še tretji sestanek, ko so bili dokončani še vsi preostali od sedmih modulov, in sicer:

- potrditev skrbnika izboljšav,
- spremljanje izvajanja izboljšave,
- izračun prihranka izboljšave,
- izračun nagrade izboljšave in
- zaključevanje izboljšave.

Seveda so tudi tu bili določeni popravki, vendar v dosti manjšem obsegu, saj smo že kar lep čas delali skupaj na tem projektu in tako tudi našli skupni jezik.

V nadaljevanju bom predstavil tehnične in vsebinske zahteve, vloge uporabnikov, organizacijo podatkov ter delovanje aplikacije.

4.1.2.1 Tehnične in vsebinske zahteve

Vsekakor je bila želja vseh nas, da pripravimo enotno vstopno točko za vse zaposlene, ki bi radi prijaviли neko izboljšavo. Ta enotna vstopna točka naj jim delo čim bolj olajša in s tem pripomore k povečanju prijavljenih izboljšav, prav tako pa naj pripomore k skrajšanju potrebnega obračalnega časa¹⁹ za obdelavo ene zahteve. Tudi ta čas pripomore k povečanju prijavljenih izboljšav, zato je še kako pomemben.

Sistem mora zadovoljiti tudi vse potrebe po interakciji z vsemi osebkami, ki so kadarkoli kakorkoli udeleženi v tej aplikaciji. Mednje torej štejemo tako prijavitelje kot vodje inovativnih jeder in skrbnike izboljšav pa seveda tudi vodstvo znotraj korporacije.

Na podlagi vsega tega je nastal seznam zahtev za vsakega izmed sedmih modulov:

- **PRIJAVA IZBOLJŠAVE:**
 - o Možnost enostavne prijave izboljšave v lastnem imenu (izboljšavo prijavlja tisti, ki je obrazec za prijavo izboljšave na intranetu tudi odprl) oz. v imenu drugega (izboljšavo prijavlja neka druga oseba, ki ima dostop do obrazca za prijavo izboljšave na intranetu, vendar prijavlja izboljšavo v imenu drugega zaposlenega – ta primer pride v poštev takrat, ko je prijavitelj neka oseba iz proizvodnje in zato nima dostopa do službenih računalnikov in mora prijavo njegove izboljšave izvesti nekdo drug, tipično njegov vodja).
 - o Možnost vnosa obveznih podatkov, kot je naslov izboljšave, opis problema izboljšave ter opis rešitve problema izboljšave.

¹⁹ Povprečen čas od trenutka, ko je izboljšava prijavljena, pa do trenutka, ko je zaključena, naj si bo pozitivno ali negativno (v primeru, ko je izboljšava zavrnjena).

- Možnost vnosa vseh prijaviteljev izboljšave, njihove osebne oz. matične številke zaposlenega, odstotka soudeležbe pri prijavljeni izboljšavi ter njihovega vodje inovativnega jedra.
 - Pri posamezni izboljšavi naj bo možno vnesti največ 4 prijavitelje.
 - Možnost dodajanja prilog prijavljeni izboljšavi – eno ali več prilog.
 - Izboljšavo naj bo možno pred oddajo tudi shraniti, če prijavitelj oz. prijavitelji ne izpolnijo še celotnega prijavnega obrazca ali pa bi ga radi pred oddajo še enkrat pregledali.
 - Posledično zaradi prejšnje točke mora sistem omogočati pregled vseh shranjenih, še ne oddanih prijav izboljšav.
 - Posledično zaradi šeste točke mora sistem omogočati izbris shranjenih, še ne oddanih prijav izboljšav, ko se prijavitelj oz. prijavitelji odločijo, da prijavo izboljšave ne bodo oddali.
 - Sistem naj ob oddaji izboljšave obvesti pravega vodja inovativnega jedra ter pravega skrbnika izboljšav o tem, da je bila oddana nova izboljšava, ter da bo zato potrebno nekaj storiti.
- **POTRDI TEV VODJE INOVATIVNEGA JEDRA:**
- Vodja inovativnega jedra mora biti takoj avtomatsko obveščen, da je bila prijavljena nova izboljšava znotraj njegovega inovativnega jedra. Sistem naj ga obvesti preko službenega elektronskega naslova z avtomatsko generirano elektronsko pošto.
 - Imeti mora možnost vpogleda v podrobnosti vseh prijavljenih izboljšav, ki spadajo v njegovo inovativno jedro.
 - Imeti mora možnost potrditve ali pa zavrnitve neke posamezne prijavitelje izboljšave, ki spada v njegovo inovativno jedro.
 - V obeh primerih (ob potrditvi in ob zavrnitvi) mora vodja obvezno vpisati komentar na prijavitelje izboljšavo.
 - Ob potrditvi izboljšave mora vodja inovativnega jedra oceniti prijavitelje izboljšavo z eno izmed stopenj izboljšav.

STOPNJE	DODATEN OPIS
Predlagatelj je predlog izvedel	Pri tem dobijo prijavitelj oz. prijavitelji največjo vrednost akontacije, ki se pri končnem izračunu odšteje od celotne nagrade.
Predlagatelj je podal predlog za izvedbo	Pri tem dobijo prijavitelj oz. prijavitelji srednjo vrednost akontacije, ki se pri končnem izračunu odšteje od celotne nagrade.
Predlagatelj je zabeležil možnost izboljšave	Pri tem dobijo prijavitelj oz. prijavitelji najmanjšo vrednost akontacije, ki se pri končnem izračunu odšteje od celotne nagrade.

Tabela 21: Stopnje izboljšav

- Ob potrditvi izboljšave mora vodja inovativnega jedra označiti, ali je bila izboljšava že izvedena in ali je potreben izračun prihranka ter nagrade.

- Prav tako mora imeti tudi vodja inovativnega jedra ob potrditvi izboljšave možnost dodajanja ene ali več prilog. Ta možnost pride v poštev predvsem takrat, ko bi rad k prijavljeni izboljšavi kakšno stvar priložil tudi vodja inovativnega jedra oz. takrat, ko predlagatelj izboljšave iz kakršnihkoli vzrokov ni imel te možnosti, da bi lahko prijavljeni izboljšavi priložil nek dokument, sliko oz. karkoli drugega.
- Vodja inovativnega jedra mora imeti možnost spremembe naslova prijavljene izboljšave v primeru da ugotovi, da naslov, ki ga je določil predlagatelj ni primeren oz. pravilen.
- Aplikacija naj po uspešni potrditvi oz. zavrniti prijavljene izboljšave takoj avtomatsko o tem obvesti predlagatelja oz. predlagatelje ter skrbnika izboljšav, ki je zadolžen za inovativno jedro vodje inovativnega jedra.
- Aplikacija naj jasno prikazuje, da je neka izboljšava potrjena oz. zavrnjena tako v seznamu vseh izboljšav kot v podrobnostih posamezne izboljšave.

- **POTRDITEV SKRBNIKA IZBOLJŠAV:**

- Po uspešni potrditvi oz. zavrnitvi prijavljene izboljšave s strani vodje inovativnega jedra, jo mora potrditi oz. zavrniti še skrbnik izboljšave, ki je zadolžen za to izboljšavo.
- Skrbnik izboljšave lahko potrdi neko izboljšavo, kljub temu da jo je vodja inovativnega jedra že prej zavrnil. To pomeni, da izboljšava ni več zavrnjena in se bo njena obravnava odvijala normalno naprej.
- Tako v primeru potrditve kot zavrnitve izboljšave s strani skrbnika izboljšave naj bo polje za vnos komentarja obvezno.
- V primeru, da izboljšava še ni bila izvedena, mora skrbnik izboljšave določiti, kdo bo odgovoren za njeno izvajanje in rok, do kdaj mora biti izvajanje izboljšave zaključeno. Ta rok ne sme presegati leta dni.
- V primeru, da je izboljšava že bila izvedena, mora skrbnik izboljšave določiti datum izvedbe, natančneje – kdaj je bila izvedba izboljšave zaključena. Ta datum ne sme biti večji kot datum potrditve skrbnika izboljšave.
- Prijavljeno izboljšavo mora skrbnik izboljšave potrditi oz. zavrniti najkasneje v 7 dneh od prejema prijavljene izboljšave v aplikacijo.
- Za sprejet predlog izboljšave skrbnik izboljšav po potrebi pridobi strokovno mnenje o izvedljivosti. V oceno predloga vključi tiste strokovnjake, ki so najbolj kompetentni, da podajo mnenje. Kadar je za uvedbo izboljšave potrebna večja investicija, skrbnik izboljšav predlaga vodstvu uvedbo spremembe, o uvedbi odloči vodstvo programa ali družbe. V takih primerih se lahko zgodi, da skrbnik izboljšave ne more izvesti potrditev oz. zavrnitev prijavljene izboljšave najkasneje v 7 dneh, zato mu moramo dati na voljo možnost, da lahko postavi potrditev oz. zavrnitev na čakanje.
- Aplikacija naj po uspešni potrditvi oz. zavrniti prijavljene izboljšave s strani skrbnika izboljšav takoj avtomatsko o tem obvesti predlagatelja oz. predlagatelje ter vodje inovativnih jeder, katerih predlagatelji so navedeni v prijavi izboljšave.

- Aplikacija naj jasno prikazuje, da je neka izboljšava potrjena oz. zavrnjena s strani skrbnika izboljšav tako v seznamu vseh izboljšav kot v podrobnostih posamezne izboljšave.

- **SPREMLJANJE IZVAJANJA IZBOLJŠAVE:**

- Skrbnik izboljšav lahko spremlja izvajanje izboljšav, za katere je zadolžen.
- V primeru, da se izvajanje izboljšave iz kakršnegakoli razloga dejansko ustavi, mora imeti možnost spremembe statusa izboljšave v status »Ustavljena« tudi v aplikaciji.
- V primeru, da se izvajanje izboljšave iz ustavljenega statusa nadaljuje, mora imeti možnost spremembe statusa izboljšave nazaj v status »V izvajanju« tudi v aplikaciji.
- V primeru, da se iz kakršnegakoli razloga izvajanje izboljšave ne samo ustavi, ampak prekine za vedno, je potrebno izboljšavo v aplikaciji zavrniti. Ob tem mora skrbnik izboljšav obvezno pripisati komentar.
- Izboljšava je izvedena, ko je vpeljana, v uporabi in standardizirana. Izvedbo izboljšave preveri in skrbniku izboljšav potrdi vodja inovativnega jedra ali odgovorna oseba za izvedbo, ki je bila določena v modulu »Potrditev skrbnika izboljšav«. V primeru, da je izvajanje izboljšave uspešno zaključeno, mora imeti sistem možnost to tudi zabeležiti. Ob tem mora skrbnik izboljšav vpisati datum zaključka izvedbe. Ta datum ne sme biti večji od datuma, ko skrbnik to vpisuje.
- Aplikacija naj po uspešni spremembi statusa izvajanja izboljšave s strani skrbnika izboljšav takoj avtomatsko o tem obvesti predlagatelja oz. predlagatelje ter vodje inovativnih jeder, katerih predlagatelji so navedeni v prijavi izboljšave.
- Aplikacija naj jasno prikazuje, da se je neki izboljšavi spremenil status izvajanja tako v seznamu vseh izboljšav kot v podrobnostih posamezne izboljšave.

- **IZRAČUN PRIHRANKA IZBOLJŠAVE:**

- Skrbnik izboljšav mora imeti možnost izračuna prihranka izboljšave po zaključku izvajanja izboljšave. Prihranek izboljšave računamo za obdobje enega leta po uvedbi izboljšave.
- Aplikacija mora skrbniku ponuditi dva načina izračuna prihranka izboljšave. Če je dejanska gospodarska korist že znana, lahko izračun prihranka izvede takoj. Izjemoma pa se lahko zgodi, da bo dejanska gospodarska korist izboljšave znana šele eno leto po uvedbi. V tem primeru pa skrbnik izboljšav označi možnost, da bo izračun prihranka preložil na eno leto.
- Aplikacija mora pri izračunu prihranka izboljšave skrbniku izboljšav ponuditi dve obvezni vnosni polji. Prva predstavlja vnos čistih prihodkov, druga pa vnos čistih odhodkov. Iz tega aplikacija izračuna gospodarsko korist in mu to tudi izpiše.

- Če je skrbnik izboljšave izračun prihranka izboljšave preložil na eno leto, ga mora aplikacija po enem letu na to opozoriti.
- Aplikacija naj po uspešnem vnosu izračuna prihranka izboljšave s strani skrbnika izboljšav takoj avtomatsko o tem obvesti predlagatelja oz. predlagateljke ter vodje inovativnih jeder, katerih predlagatelji so navedeni v prijavi izboljšave.
- Aplikacija naj jasno prikazuje, da je bil neki izboljšavi izračunan prihranek izboljšave tako v seznamu vseh izboljšav kot v podrobnostih posamezne izboljšave.

- **IZRAČUN NAGRADE IZBOLJŠAVE:**

- Skrbnik izboljšav mora imeti možnost izračuna nagrade izboljšave po uspešno izračunanem prihranku izboljšave. Nagrado izboljšave izračunamo za vsakega prijavitelja.
- Osnova za izračun nagrade znaša 15 % ugotovljene gospodarske koristi oz. prihranka izboljšave.
- Pri izračunu nagrade nato upoštevamo še korekcijske faktorje, ki jih je potrebno prišteti osnovi za izračun nagrade. Ti korekcijski faktorji so:
 - vpliv na varnost in zdravje pri delu: 2 %,
 - vpliv na ekologijo/okolje: 2 %,
 - timsko delo: 2 %,
 - aplikativnost uvedbe v ostalih družbah: 2 %,
 - gospodarska korist se odraža več kot eno leto: 2 % in
 - samostojnost izvedbe: 20 %.
- Od izračunane nagrade nato odštejemo že izplačano akontacijo, ki je bila določena z oceno izboljšave v potrditvi vodje inovativnega jedra.
- V primeru, ko je izračunana nagrada nižja od že izplačane akontacije, se kot končno izplačilo upošteva že izplačano akontacijo.
- To nagrado, od katere je že odšteta izplačana akontacija, nato razdelimo na prijavitelje glede na njihov procent soudeležbe. Temu pravimo nagrada predlagatelju.
- Po zaključku izračuna nagrade izboljšave mora biti ta potrjena. Če je skupna vrednost nagrade v območju do 1.000 €, je potrditev v pristojnosti vodje programa, vse nagrade, ki pa presegajo 1.000 €, pa potrdijo tudi direktorji družbe. Nagrada je lahko izplačana po uspešni potrditvi.

- **ZAKLJUČEVANJE IZBOLJŠAVE:**

- Ko je nagrada izračunana, uspešno potrjena in je izplačilo nagrade tudi dejansko izvedeno, se lahko prijavo izboljšave tudi zaključi.
- Ob tem mora skrbnik izboljšav obvezno izpolniti še vnosno polje za končen komentar in izboljšava je lahko zaključena.
- Aplikacija naj po uspešnem zaključku izboljšave s strani skrbnika izboljšav takoj avtomatsko o tem obvesti predlagatelja oz. predlagateljke ter vodje inovativnih jeder, katerih predlagatelji so navedeni v prijavi izboljšave.

- Aplikacija naj jasno prikazuje, da je bila neka izboljšava zaključena tako v seznamu vseh izboljšav kot v podrobnostih posamezne izboljšave.

Ostale so samo še zahteve glede samega prikaza seznama vseh izboljšav, do katerega lahko dostopa vsak zaposleni znotraj korporacije HIDRIA:

- Seznam naj prikazuje vse prijavljene izboljšave iz tekočega leta ter samo nezaključene izboljšave iz prejšnjih let.
- Seznam naj o posamezni izboljšavi prikazuje naslednje podatke:
 - zaporedna številka izboljšave (sestavljena naj bo iz zaporedne številke dolžine štirih mest znotraj enega leta ter leta v skrajšani obliki – 0001/11);
 - status izboljšave (predstavljen z barvo);
 - naslov izboljšave;
 - opis problema izboljšave;
 - opis rešitve problema izboljšave;
 - poslovna enota, iz katere izvira izboljšava;
 - datum prijave;
 - število dni do zaključka posameznega statusa.
- Seznam naj bo razdeljen na strani, 50 izboljšav/stran. Omogočeno naj bo premikanje za eno stran naprej/nazaj ter na prvo in zadnjo stran. Možen naj bo tudi skok na določeno stran.
- Status izboljšav naj bo prikazan z barvami (legenda status – barva naj bo vidna ob seznamu).

STATUS	BARVA
Prijavljen	Bela barva
Potrjen	Rumena barva
Ustavljen	Oranžna barva
Izveden	Zelena barva
Zaključen	Siva barva
Zavrtnjen	Rdeča barva

Tabela 22: Statusi izboljšav

- Podrobnosti izboljšave naj bodo videne z izbiro posamezne izboljšave iz seznama. Izbira posamezne izboljšave naj bo možna s klikom kjerkoli v vrstici. Podrobnosti izboljšave si lahko ogleda samo tisti, ki ima za to pravice. Več o pravicah v poglavju »Vloge uporabnikov«, ki sledi.
- Naslov izboljšave, opis problema izboljšave ter opis rešitve izboljšave naj bodo levo poravnani, ostali stolpci pa sredinsko.
- Seznam naj bo mogoče filtrirati po vseh stolpcih.

4.1.2.2 Vloge uporabnikov

Kot v vsaki aplikaciji imamo tudi tu uporabniške vloge. Neka uporabniška vloga predstavlja skupino ljudi, ki imajo enake pravice za delo s to aplikacijo. Vsaka uporabniška vloga ima unikatne pravice glede na drugo uporabniško vlogo.

Po določitvi tehničnih in vsebinskih zahtev je bila naslednja naloga pri izgradnji aplikacije določitev uporabniških vlog. Aplikacija za identifikacijo uporabnika uporablja vgrajeni prijavitni sistem platforme SharePoint. Ta podatke uporabnikov pobira iz imenika AD. Uporabnika prepozna glede na to, katera oseba je prijavljena v računalnik, saj ta preko brskalnika dostopa do platforme SharePoint.

VLOGA UPORABNIKOV	OPIS VLOGE
Vsi zaposleni	Ta uporabniška vloga ima najmanj pravic. Vanjo avtomatično spada vsak zaposleni znotraj korporacije HIDRIA. Tak uporabnik ima pravico prijaviti izboljšavo, si ogledati seznam vseh izboljšav, kjer si lahko podrobnosti posamezne izboljšave ogleda samo v primeru, da je v tej izboljšavi naveden med predlagatelji. Prav tako ima pravico ogleda seznama svojih shranjenih, a še ne oddanih izboljšav.
Vodje inovativnih jeder	Ta uporabniška vloga sledi uporabniški vlogi »Vsi zaposleni«. Poleg vseh njenih pravic ima dodatno še pravico ogleda in potrjevanja prijav, pri katerih nastopa kot vodja inovativnega jedra, v katerega spada glavni predlagatelj.
Skrbniki izboljšav	Ta uporabniška vloga sledi uporabniški vlogi »Vsi zaposleni«. Poleg vseh njenih pravic ima dodatno še pravico ogleda, strokovnega potrjevanja, spremljanje izvajanja izboljšav, izračuna prihranka izboljšav, izračuna nagrad izboljšav in zaključevanje izboljšav, pri katerih nastopa kot skrbnik izboljšave. Nekdo je skrbnik neke izboljšave, ko pokriva inovativno jedro, kateremu pripada glavni predlagatelj.
Administratorji	Ta uporabniška vloga ima največ pravic. Poleg pravic iz vseh sedmih modulov aplikacije ima še pravice dodajanja oseb v posamezne uporabniške vloge in urejanje šifrantov aplikacije.

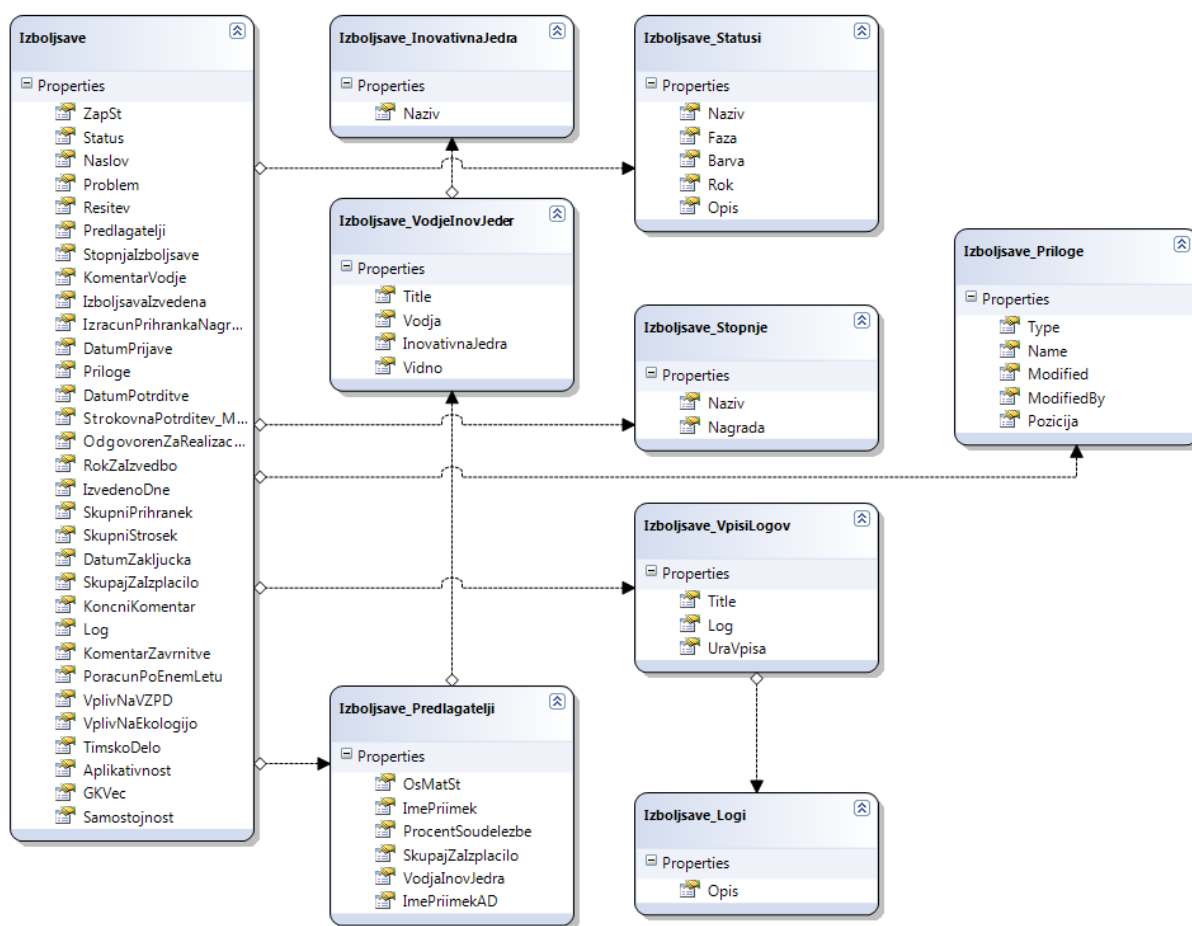
Tabela 23: Vloge uporabnikov pri aplikaciji za prijavo in obdelavo izboljšav

4.1.2.3 Organizacija podatkov

Pomembno vlogo pri izdelavi te aplikacije je imela dobra organizacija podatkov, kljub temu da jo končni uporabniki niti ne vidijo. Dobro premišljena organizacija podatkov mi je tako omogočala lažje programiranje, poleg tega pa sem jo pustil dovolj fleksibilno, da ob morebitnih širitvah funkcionalnosti te aplikacije to ne bi smelo povzročati prevelikih težav.

Podatki so pri tej aplikaciji shranjeni v osmih SharePointovih seznamih in v eni SharePointovi knjižnici²⁰. Spodnja slika prikazuje celotno organizacijo podatkov za to aplikacijo.

²⁰ SharePoint knjižnica je vrsta SharePointovega seznama, prilagojenega za shranjevanje vseh vrst datotek. Predstavlja lahko majhen datotečni sistem, saj se v njej lahko izdelata tudi hierarhija map in podmap.



Slika 20: Organizacija podatkov za aplikacijo za prijavo in obdelavo izboljšav

V organizaciji podatkov je glavni seznam *Izboljsave*. V njem so shranjeni podatki o vseh prijavljenih izboljšavah. Poleg njega lahko med glavne sezname uvrščamo še seznam *Izboljsave_Predlagatelji*, v katerem so shranjeni podatki o vseh predlagateljih za posamezno izboljšavo.

V seznamu *Izboljsave_InovativnaJedra* lahko najdemo šifrant vseh inovativnih jedr v korporaciji HIDRIA, v seznamu *Izboljsave_VodjeInovJeder* pa vodje posameznih inovativnih jedr. Seznama sta med seboj povezana z razmerjem mnogo-mnogo, saj je lahko neka oseba vodja večjemu številu inovativnih jedr, prav tako pa ima lahko eno inovativno jedro več vodij.

V seznamu *Izboljsave_Statusi* se nahaja šifrant vseh statusov, v katerih se lahko nahaja neka izboljšava.

V seznamu *Izboljsave_Stopnje* pa najdemo šifrant vseh stopenj (Tabela 21). Stopnja izboljšave predstavlja oceno predloga, ki jo določi vodja inovativnega jedra. Ta stopnja določa tudi zagotovljeno akontacijo, ki jo dobijo vplačano vsi predlagatelji določene izboljšave.

Tu imamo še seznam *Izboljsave_Logi*, v katerem se nahajajo vsi možni izpisi, ki se lahko pojavijo v arhivu aktivnosti na posamezni izboljšavi. Za dodeljevanje teh izpisov določeni izboljšavi, pa uporablja aplikacija vmesni seznam *Izboljsave_VpisiLogov*.

Poleg vseh teh seznamov uporablja aplikacija še knjižnico *Izboljsave_Priloge*, v kateri so shranjene vse priloge, ki so lahko pripete posamezni izboljšavi. Parameter *Pozicija* določa, kje je bila neka datoteka priložena.

POZICIJA PRILOŽENE DATOTEKE	OPIS
1	Ob prijavi izboljšave
2	Ob potrditvi vodje inovativnega jedra

Tabela 24: Pozicije priloženih datotek

4.1.3 Delovanje aplikacije

Vstopna stran aplikacije predstavlja pregled vseh podanih prijav izboljšav za tekoče leto ter samo nezaključene izboljšave iz prejšnjih let. Za pregled vseh izboljšav iz prejšnjih let se uporablja arhiv izboljšav.

Nadzorna plošča

Prijava izboljšave

Vaše oddane izboljšave

Vaše shranjene izboljšave

Legenda statusov:

■ Prijavljen	■ Izvedena
■ Potrjena	■ Zaključena
■ Ustavljena	■ Zavrnjena

Pregled vseh podanih prijav izboljšav

Zap. št.	Status	Naslov izboljšave	Opis problema	Opis rešitve	Poslovna enota	Datum prijave	Št. dni do zaključka statusa
0014/11	■	Testna izboljšava 5 ■ NEW	Testni opis problema 5	Testni opis rešitve 5	/	21.11.2011	/
0013/11	■	Testna izboljšava 3 ■ NEW	Testni opis problema 3	Testni opis rešitve 3	/	21.11.2011	/
0012/11	■	Testna izboljšava 2 ■ NEW	Testni opis problema 2	Testni opis rešitve 2	/	21.11.2011	9
0011/11	■	Testna izboljšava 1 ■ NEW	Testni opis problema 1	Testni opis rešitve 1	/	21.11.2011	7
0010/11	■	Izdelava nove aplikacije za nadzor terminalov ■ NEW	Stara aplikacija dela težave, ...	Izdelava nove aplikacije, ki b...	/	21.11.2011	30
0009/11	■	Izdelava aplikacije za prijavo in obdelavo izboljšav ■ NEW	Aplikacija za prijavo in obdel...	Rešitev tega problema bi bila ...	/	21.11.2011	7

Stran 1 od 1

Slika 21: Vstopna stran aplikacije za prijavo in obdelavo izboljšav

Status posamezne izboljšave je določen z barvo. Legenda vseh statusov in pripadajočih barv je prikazana na vrhu strani. Če uporabnik pridrži kazalec na barvi statusa, se mu v balončku izpiše tudi opis statusa.

↓	Status	
1	■	Testna izbolj:
1	■	Testna izbolj:
1	■	Testna izbolj:
1	■	Potrjena [V izvajanju]
1	■	Izdelava nov

Slika 22: Prikaz statusa posamezne izboljšave

Naslov izboljšave je omejen po številu znakov. V seznamu vseh izboljšav je prikazan v celoti. To pa ne velja za opis problema ter opis rešitve problema izboljšave. V primeru, da je eden ali drugi daljši od določenega števila znakov, se prikaže samo del besedila, na koncu tega pa so dodane tri pike. Uporabnik lahko celotno besedilo tako opisa problema kot opisa rešitve problema vidi tako, da pridrži kazalec na posameznem skrajšanem besedilu. Ob tem se mu v balončku izpiše celotno besedilo.

	Opis problema	Opis rešitve	Poslovna enota	Datum prijave	Št. dni do zaključka statusa
	Testni opis problema 5	Testni opis rešitve 5	/	21.11.2011	/
	Testni opis problema 3	Testni opis rešitve 3	/	21.11.2011	/
	Testni opis problema 2	Testni opis rešitve 2	/	21.11.2011	9
	Testni opis problema 1	Testni opis rešitve 1	/	21.11.2011	7
NEW	Stara aplikacija dela težave, ...	Izdelava nove aplikacije, ki b...	/	21.11.2011	30
av NEW	Aplikacija za pri...	Stara aplikacija dela težave, ker ob osveževanju strani s seznamom vseh terminalov ponovno naloži celotno stran. To uporabnike zmoti, ker se stran postavi ponovno na vrh, poleg tega pa ob slabi povezavi osveževanje lahko traja tudi več kot sekundo.			

Slika 23: Prikaz daljših opisov v pregledu vseh prijav izboljšav

Enako velja tudi za datum prijave, saj v seznamu ni mogoče videti tudi ure prijave. Za prikaz ure mora uporabnik pridržati kazalec na datumu. Takrat se mu v balončku pokaže tudi ura.

Poslovna enota	Datum prijave	Št. dni do zaključka statusa
/	21.11.2011	/
/	21.11.2011	/
/	21.11.2011	9
/	21.11.2011	7
/	21.11.2011	0
/	21.11.2011	7

Slika 24: Prikaz datuma in ure

Zadnji stolpec predstavlja število dni do zaključka statusa, v katerem se trenutno nahaja neka izboljšava (status določa prej omenjena barva). Koliko časa je lahko izboljšava v določenem statusu, si pogledajmo spodaj:

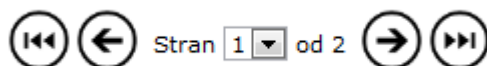
STATUS IZBOLJŠAVE	ŠT. DNI DO ZAKLJUČKA STATUSA
Shranjena	Neomejeno
Prijavljena	7 dni
Potrjena	Omejeno z rokom za izvedbo
Ustavljena	Neomejeno
Izvedena	30 dni
Zaključena	Neomejeno

Tabela 25: Število dni do zaključka posameznega statusa

Ko je število dni do zaključka statusa enako nič, se ta celica obarva z oranžno barvo, ko pa postane število dni manjše od nič, se celica obarva z rdečo barvo.

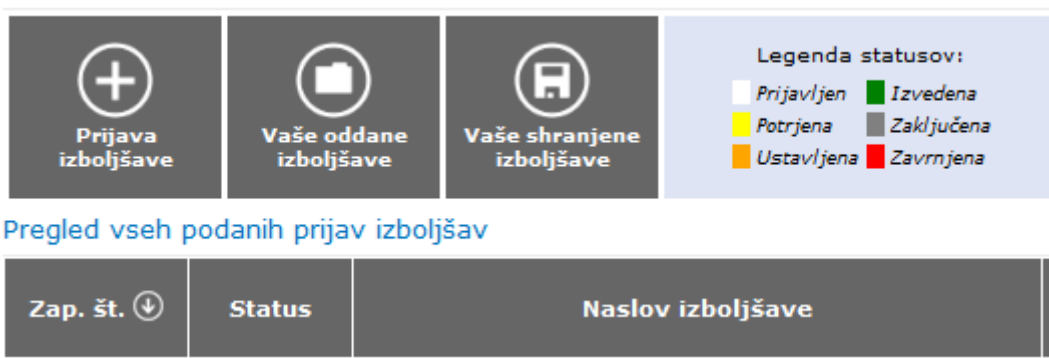
Pod seznamom se nahajajo možnosti prehajanja med stranmi:

- skok na prvo stran,
- skok na prejšnjo stran,
- skok na poljubno stran,
- skok na naslednjo stran in
- skok na zadnjo stran.



Slika 25: Prehajanje med stranmi

Nad seznamom imamo poleg legende statusov še možnost prijavljanja nove izboljšave, pregled oddanih izboljšav in pregled shranjenih (še ne oddanih) izboljšav osebe, ki je trenutno prijavljena v SharePoint. Seznam je možno tudi urejati po vseh atributih, tako padajoče kot naraščajoče.



Slika 26: Prikaz možnih akcij, legende statusov in urejanja po vseh parametrih seznama

S klikom na posamezno izboljšavo v pregledu vseh podanih prijav izboljšav se pokaže stran s podrobnostmi posamezne izboljšave. Podrobnosti posamezne izboljšave zavzemajo pregled obrazca »5 minut za izboljšavo«, pregled prilog izboljšav ter pregled arhiva aktivnosti na izboljšavi. Na vrhu strani je vgrajeni statusni vrstici platforme SharePoint izpisan status izboljšave v določeni barvi. Na desnem delu strani pa se nahaja nadzorna plošča, ki določa, kaj je v nekem trenutku možno z določeno izboljšavo narediti (potrditev izboljšave, izračun prihranka, izračun nagrade itd.).

Pregled izboljšave 0009/11

Page

Status izboljšave: Izboljšava je prijavljena. Trenutno je v fazi strokovne potrditve.

Obrazec 5 minut za izboljšavo

Problem

Aplikacija za prijavo in obdelavo izboljšav bi omogočala lažje in hitrejše prijavljanje izboljšav ter njihovo obdelavo. To bi pripomoglo k temu, da bi ljudje izboljšave raje prijavljali.

Rešitev

Rešitev tega problema bi bila izdelava aplikacije za SharePoint, ki bi vse te pogoje zadostila.

Podatki predlagatelja

Os./mat. št.	Ime in priimek	% soud.	Nadrejeni vodja	Datum
1234	GAŠPER RUPNIK	100 %	Boštjan Tušar	21.11.2011

Stran 1 od 2

Nadzorna plošča

Strokovna potrditev

Priloge izboljšave

Tip priloge	Ime priloge	Oseba
	ASN001.txt	Gašper Rupnik
	F350 - Mesečna evidenca prisotnosti str2.doc	Gašper Rupnik
	Obrazec za obravnavo izboljšave.xlsx	Gašper Rupnik
	IECertError.png	Gašper Rupnik

Stran 1 od 1

Arhiv aktivnosti na izboljšavi

Datum	Opis	Oseba
21.11.2011	Izboljšava je bila potrjena s strani vodje.	Boštjan Tušar
21.11.2011	Izboljšava je bila oddana.	Gašper Rupnik

Stran 1 od 1

Slika 27: Prikaz podrobnosti posamezne izboljšave

Pregled obrazca »5 minut za izboljšavo« je sestavljen iz dveh strani, ki jih je možno preklaplјati z gumbi pod obrazcem. Vsebuje vse podatke, ki so jih prijavitelji navedli ob prijavi izboljšave ter podatke, ki jih je podal vodja inovativnega jedra.

Obrazec 5 minut za izboljšavo

Problem				
Stara aplikacija dela težave, ker ob osveževanju strani s seznamom vseh terminalov ponovno naloži celotno stran. To uporabnike zmoti, ker se stran postavi ponovno na vrh, poleg tega pa ob slabi povezavi osveževanje lahko traja tudi več kot sekundo.				
Rešitev				
Izdelava nove aplikacije, ki bi to rešila s AJAX komponento UpdatePanel. Poleg tega bi omogočala jasnejši prikaz napak na samih terminalih.				
Podatki predlagatelja				
Os./mat. št.	Ime in priimek	% soud.	Nadrejeni vodja	Datum
1234	<input checked="" type="checkbox"/> GAŠPER RUPNIK	50 %	<input checked="" type="checkbox"/> Boštjan Tušar	21.11.2011
2345	<input type="text" value="janez kranjski"/> <input checked="" type="checkbox"/>	50 %	<input checked="" type="checkbox"/> Boštjan Tušar	21.11.2011

Stran 1 od 2

Slika 28: Prva stran obrazca "5 minut za izboljšavo"

Obrazec 5 minut za izboljšavo

Naslov izboljšave:	Izdelava aplikacije za prijavo in obdelavo izboljšav	Zap.št.:	0009/11
Nagrada (bruto zneski)		Komentar vodje	
<input type="radio"/> Predlagatelj je predlog sam izvedel <input checked="" type="radio"/> Predlagatelj je podal rešitev za realizacijo izboljšave (kako je treba narediti) <input type="radio"/> Predlagatelj je podal predlog za izboljšavo (kaj je treba narediti)		Mislim, da je to zelo dobra ideja.	
<input type="checkbox"/> Izboljšava je že izvedena			
<input checked="" type="checkbox"/> Opraviti izračun prihranka in nagrade		Datum:	/
Priloge:	<input checked="" type="radio"/> Da <input type="radio"/> Ne	Vodja:	3

Stran 2 od 2

Slika 29: Druga stran obrazca "5 minut za izboljšavo"

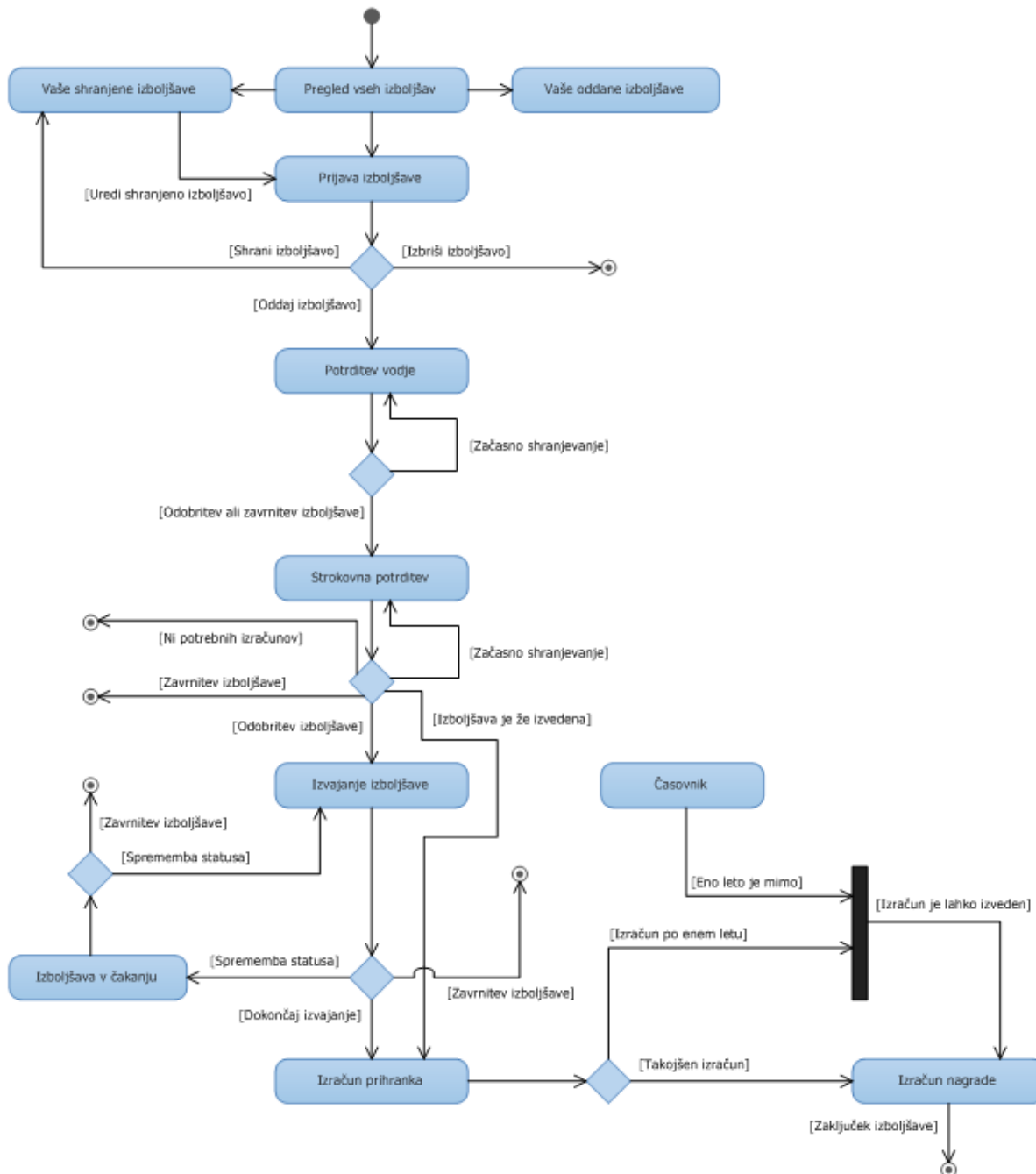
Pregled prilog izboljšave vsebuje seznam vseh prilog, ki jih je kdorkoli priložil izboljšavi (sam predlagatelj oz. predlagatelji, vodja inovativnega jedra ali skrbnik izboljšav). Tip priloge prikazuje ikona, priloga je poimenovana, izpisana pa je tudi oseba, ki je prilogo priložila. Ta oseba je izpisana s statusom in dodatnimi funkcionalnostmi, ki se pokažejo v oblaku, za komuniciranje preko programske opreme Microsoft Lync. Posamezno prilogo si je mogoče ogledati s klikom na njeno ime.

Pregled arhiva aktivnosti na izboljšavi omogoča natančno sledenje, kaj in kdaj se je neka stvar na izboljšavi dogodila (sprememba statusa, potrditev izboljšave itd.). Poleg datuma in ure je tu še opis ter oseba, ki je to aktivnost izvedla. Posamezno aktivnost si je mogoče tudi ogledati s klikom na njen opis. Prikažejo se izpolnjeni obrazci, ki so bili povod za to, da se je ta aktivnost tudi dogodila. Kot primer vzemimo situacijo, ko vodja inovativnega jedra potrdi izboljšavo z nekim obrazcem. Takrat se ta dogodek izpiše kot aktivnost z opisom »Izboljšava je bila potrjena s strani vodje«. Ko nekdo klikne na ta opis, pa se mu pokaže izpolnjen obrazec s strani vodje inovativnega jedra.

V nadaljevanju bom v sliki in besedi predstavil celoten proces prijave in obdelave izboljšav od same prijave izboljšave, preko potrjevanj, izračunov do njenega zaključka.

4.1.3.1 Proces prijave in obdelave izboljšav

Kot prvo naj povem to, da na začetku same izdelave te aplikacije tako jaz kot naročnik nismo imeli občutka, da bo proces prijave in obdelave izboljšav postal tako kompleksen, s toliko vejitvami, pogoji in fazami. Prav smešno se je spomniti začetniških besed, ko je bilo rečeno, da bo za izdelavo te aplikacije potrebnih kakšnih dva, tri dni.



Slika 30: UML model procesa prijave in obdelave izboljšav

Na zgornji sliki je prikazan proces, skozi katerega mora vsaka izboljšava. Ko nek uporabnik pristopi do te aplikacije, se nahaja na prvi strani – pregled vseh podanih prijav izboljšav.

Uporabnik se odloči, da bo prijavil izboljšavo, zato klikne na gumb Prijava izboljšave. Odpre se mu obrazec za prijavo izboljšave.

4.1.3.2 Prijava izboljšave

Obrazec za prijavo izboljšave

Page

Prijava izboljšave

5 minut za izboljšavo

Način prijave *
Način prijave določa, ali prijavljate predlog v lastnem imenu ali v imenu drugega zaposlenega. V primeru da prijavljate v imenu drugega zaposlenega vpisite še njegove zahtevane podatke.

Koristni predlog prijavljam:
 V lastnem imenu
 V imenu drugega zaposlenega

Ime in priimek: GAŠPER RUPNIK
 Os./mat. št.: 1234

Naslov izboljšave *
Naslov izboljšave predstavlja vašo izboljšavo, ki jo prijavljate. Naj bo kratek in jedrnat.

Testni naslov izboljšave 6

Opis problema *
Opis problema predstavlja kratko predstavitev problema izboljšave, ki jo prijavljate.

Testni opis problema 6

Opis rešitve *
Opis rešitve predstavlja kratko predstavitev rešitve problema izboljšave, ki jo prijavljate.

Testni opis rešitve problema 6

Podatki prijavitelja *
Podatki prijavitelja predstavljajo potrebne osnovne podatke prijavitelja (ev, % soudeležbe in njegovega nadrejenega vodja).

	Os./mat. št.	Ime in priimek	% soudeležbe	Vodja inovativnega jedra
	1234	GAŠPER RUPNIK	33	BOŠTJAN TUŠAR
X	2345	MIHA SEDEJ	33	BOŠTJAN TUŠAR
X			34	

+ Dodaj novega prijavitelja

Priloge
Priložene datoteke (ena ali več) z datjšo predstavitvijo, dodatnimi pojasnili, slikami itd. *Priloge je potrebno pripeti na obrazec pred oddajo izpolnjenega obrazca.*

Browse... Pripni

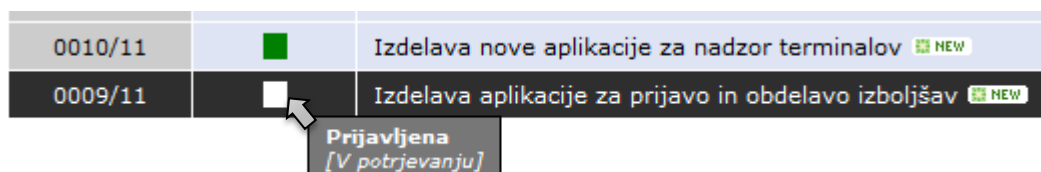
X Izboljsave_Priprava.xlsx
 X ASN001.txt

Oddaj Shrani Prekliči

Slika 31: Obrazec za prijavo izboljšave

Na tem obrazcu določi, ali koristni predlog prijavlja v lastnem imenu ali v imenu drugega zaposlenega. Prijava v imenu drugega zaposlenega omogoča, da lahko izboljšavo prijavijo tudi proizvodni delavci. Ti namreč nimajo dostopa do službenih računalnikov, nimajo dostopa do službene elektronske pošte itd. V primeru prijave v imenu drugega zaposlenega mora tako prijavitelj vpisati njegovo ime in priimek, v primeru prijave v lastnem imenu pa se ime in priimek prepíše iz imenika AD, glede na to, katera oseba je prijavljena v SharePoint. Tu je potrebno vpisati še os./mat. št., ki jo ima vsak zaposleni v korporaciji HIDRIA. V

nadaljevanju mora prijavitelj obvezno vnesti naslov, opis problema in opis rešitve problema izboljšave. Pri podatkih prijavitelja se v prvi vrstici seznama vseh prijaviteljev prepišejo podatki prijavitelja glede na način prijave. Potrebno je izbrati še njegov procent soudeležbe ter vodjo inovativnega jedra, kateremu pripada. Če želimo dodati več prijaviteljev, to naredimo s klikom na napis Dodaj novega prijavitelja. Za vsakega novega prijavitelja moramo ravno tako vpisati os./mat. št. ter ime in priimek, določiti procent soudeležbe in vodjo inovativnega jedra. Vodje inovativnega jedra se nahajajo v posebnem šifrantu, ki ga urejajo administratorji aplikacije. Prijaviteljev na neko izboljšavo je lahko največ 4, posameznega prijavitelja se lahko odstrani s klikom na rdeč križ. Na koncu lahko dodamo še priloge, ki jih je pred oddajo prijave potrebno pripeti na obrazec. Če prijava še ni dokončana oz. je ne želimo še oddati, lahko to storimo s klikom na gumb Shrani. V tem primeru se izboljšava shrani v poseben seznam shranjenih izboljšav, do katerega lahko uporabnik dostopa iz začetne strani aplikacije. Ko pa se odločimo obrazec oddati, kliknemo na gumb Oddaj. Takrat se prijavljena izboljšava prikaže tudi v pregledu vseh podanih prijav izboljšav na glavni strani aplikacije. Njen status je bele barve, kar pomeni, da je trenutno prijavljena in je v potrjevanju. Prav tako je poleg naslova izboljšave oznaka, ki predstavlja to, da gre za novo izboljšavo.



0010/11	■	Izdelava nove aplikacije za nadzor terminalov NEW
0009/11	■	Izdelava aplikacije za prijavo in obdelavo izboljšav NEW

Slika 32: Prikaz prijavljene izboljšave

Ob oddaji izboljšave dobi vodja prvega predlagatelja elektronsko pošto, da je bila oddana nova izboljšava in da jo je potrebo potrditi. Prav tako dobi informacijo o tem, da je bila oddana nova izboljšava, tudi skrbnik izboljšav, ki nadzoruje prijave v inovativnem jedru, katerega vodja je zgoraj omenjena oseba.

4.1.3.3 Potrditev vodje inovativnega jedra

Naslednja naloga v procesu prijave in obdelave izboljšav je torej potrditev vodje inovativnega jedra. To lahko vodja stori tako, da sledi povezavi, ki mu je bila pripisana v elektronski pošti ali pa iz seznama vseh podanih prijav izboljšav izbere to izboljšavo s klikom nanjo. Odpre se mu pregled izboljšav, v katerem lahko s klikom na gumb Potrditev vodje na desni strani izvede potrditev. Ob tem se mu odpre spodnji obrazec.

Obrazec za prijavo izboljšave

Page

Obrazec za strokovno potrditev izboljšave

Potrditev izboljšave s strani vodje

Potrditev vodje *
Tu pride besedilo ...

Izberite eno izmed danih možnosti:

Odobreno

Zavrjneno

Naslov izboljšave *
Naslov izboljšave predstavlja prijavljeno izboljšavo. Naj bo kratek in jedrnat.

Testna izboljšava 6

Nagrada *
Nagrada predstavlja najmanjšo denarno vrednost, ki jo prijavitelj/i dobijo glede na stopnjo prijavljene izboljšave.

Izboljšava spada pod naslednjo stopnjo:

Predlagatelj je predlog sam izvedel
Nagrada:

Predlagatelj je podal rešitev za realizacijo izboljšave (kako je treba narediti)
Nagrada:

Predlagatelj je podal predlog za izboljšavo (kaj je treba narediti)
Nagrada:

Komentar vodje *
Komentar vodje na prijavljeno izboljšavo.

Ostalo *
Dodatne lasnosti prijavljene izboljšave.

Izboljšava je že izvedena

Opraviti izračun prihranka in nagrade

Priloge
Priložene datoteke (ena ali več) z daljšo predstavitvijo, dodatnimi pojasnili, slikami itd.
Priloge je potrebno pripeti na obrazec pred potrditvijo izboljšave.

Slika 33: Obrazec za strokovno potrditev izboljšave

Vodja lahko izboljšavo odobri ali zavrne. Če izboljšavo zavrne, mora izpolniti samo polje Komentar vodje. Ob odobritvi lahko popravi naslov izboljšave, če meni, da predlagatelj ni izbral primernega. Določiti mora nagrado oz. stopnjo izboljšave, s katero določi višino akontacije, ki jo dobijo predlagatelji plačano vnaprej. Tudi v tem primeru je obvezen komentar vodje. Pri ostalih podatkih mora vodja določiti, ali je izboljšava že bila izvedena in ali je potrebno opraviti izračun prihranka in nagrade. V primeru da označi, da je izboljšava že bila izvedena, ta izboljšava niti ne gre skozi status izvajanja. Če pa vodja ne označi, ali je potrebno opraviti izračun prihranka in nagrade, aplikacija tako preskoči koraka izračun prihranka ter izračun nagrade izboljšave. Prav tako lahko vodja izboljšavi priloži datoteke. Samo potrditev lahko preloži na naslednjič, tako da klikne spodaj na gumb Shrani, ali pa svojo odločitev kar odda s klikom na gumb Potrdi.

V primeru, da vodja zavrne izboljšavo, v seznamu vseh izboljšav ta še ni označena kot zavrjnena, saj mora izboljšavo potrditi oz. zavrniti tudi skrbnik izboljšav za to inovativno jedro. Informacijo o izvedeni potrditvi s strani vodje dobijo tako predlagatelji kot skrbnik

izboljšav. Skrbniku izboljšav je pripisano še to, da lahko sedaj izboljšavo tudi strokovno potrdi. To mora storiti do roka 7 dni od prijave izboljšave.

4.1.3.4 Potrditev skrbnika izboljšave

Naslednja naloga v procesu prijave in obdelave izboljšav je torej potrditev skrbnika izboljšave oz. strokovna potrditev. To lahko tako kot prej stori preko pripisane povezave v elektronski pošti ali s klikom na posamezno izboljšavo v seznamu vseh. Ob tem se mu odpre več možnih oblik obrazca, odvisno od tega, kaj je izbral vodja inovativnega jedra.

Slika 34: Obrazec za obravnavo izboljšave v primeru še ne izvedene izboljšave

Slika 35: Obrazec za obravnavo izboljšave v primeru že izvedene izboljšave

Slika 34 prikazuje obliko obrazca, ko je vodja izboljšavo potrdil in označil, da še ni bila izvedena. To, da je vodja izboljšavo potrdil, skrbnika opozori aplikacija že v statusni vrstici na obrazcu. Tukaj lahko torej skrbnik izboljšavo odobri ali zavrne. V obeh primerih mora izpolniti polje Mnenje. V primeru odobritve pa mora določiti še odgovornega za realizacijo, kjer lahko izbere med prvim predlagateljem ali neko drugo osebo. Določiti mora še rok za izvedbo, ki ne sme biti manjši od današnjega dne in ne večji od enega leta.

Slika 35 prikazuje obliko obrazca, ko je vodja izboljšavo potrdil, vendar je označil, da je bila izboljšava že izvedena. V tem primeru je obrazec podoben zgornjemu, razlika je le v tem, da tu ni potrebno določiti niti odgovornega za realizacijo niti roka za izvedbo. Namesto tega je tu potrebno izbrati datum, kdaj je bila izboljšava izvedena. Ta pa ne sme biti večji od današnjega datuma.

Ko je izboljšava potrjena tudi s strani skrbnika izboljšav, se izboljšavi spremeni status iz Prijavljena v Potrjena. Če je bila izboljšava označena kot že izvedena, pa se status spremeni v Izvedena.

0013/11	■	Testna izboljšava 3 NEW
0012/11	■	Testna izboljšava 2 NEW
0011/11	■	Potrjena [V izvajanju] Izboljšava 1 NEW
0010/11	■	Izboljšava nove aplikacije za nadzor terminalov NEW

Slika 36: Prikaz potrjene izboljšave

0011/11	<input type="checkbox"/>	Testna izboljšava 1 NEW
0010/11	<input checked="" type="checkbox"/>	Izdelava nove aplikacije za nadzor terminalov NEW
0009/11	<input type="checkbox"/>	Izvedena aplikacije za prijavo in obdelavo izboljšav NEW [V zaključevanju]

Slika 37: Prikaz izvedene izboljšave

4.1.3.5 Spremljanje izvajanja izboljšave

Če izboljšava še ni bila izvedena, sledi stanje spremljanja izvajanja izboljšave. Izboljšavi se lahko v tem stanju spreminja status ali pa izvajanje dokonča. Za to in za vse naslednje korake je odgovoren samo še skrbnik izboljšave, vodja inovativnega jedra in predlagatelji pa ob zaključkih posameznega koraka dobivajo obvestila preko elektronske pošte.

Obrazec za spremembo statusa

Page

Obrazec za spremembo statusa izboljšave

Sprememba statusa izboljšave

Status izboljšave *
Tu pride besedilo ...

Izberite eno izmed danih možnosti:

V izvajanju

V čakanju

Zavrnitev izboljšave
Tu pride besedilo ...

Ali želite zavrni izboljšavo?

Zavrni izboljšavo

Potrdi Prekliči

Slika 38: Obrazec za spremembo statusa izboljšave

Spremembo statusa izboljšave lahko skrbnik ureja z obrazcem na zgornji sliki. Tu ima tri možnosti:

- ali izboljšavo postavi iz izvajanja v čakanje;
- ali izboljšavo postavi iz čakanja v izvajanje;
- ali izboljšavo zavrne iz kakršnega koli razloga in ta razlog navede pod opombami.

Sprememba stanj se beleži tudi v arhivu aktivnosti na izboljšavi.

Arhiv aktivnosti na izboljšavi

Datum	Opis	Oseba
21.11.2011	Izboljšavi je bil spremenjen status v "V izvajanju".	Boštjan Tušar
21.11.2011	Izboljšavi je bil spremenjen status v "V čakanju".	Boštjan Tušar
21.11.2011	Izboljšava je bila potrjena s strani vodje.	Boštjan Tušar
21.11.2011	Izboljšava je bila oddana.	Gašper Rupnik



Slika 39: Beleženje spremembe statusa izboljšave

Ko pa želi skrbnik zaključiti izvajanje izboljšave, to stori s spodnjim obrazcem. Obrazec od skrbnika zahteva zgolj vnos datuma, kdaj je bila izboljšava izvedena. Ta seveda ne sme biti večji od današnjega.

Slika 40: Obrazec za zaključevanje izvajanja izboljšave

Ko je izvajanje izboljšave zaključeno, se njen status spremeni v status Izvedena. Aplikacija o tem obvesti tako predlagatelje kot vodjo inovativnega jedra. Sledita koraka Izračun prihranka in Izračun nagrade, če je vodja označil, da je izračun potreben.

4.1.3.6 Izračun prihranka izboljšave

Pri izračunu prihranka se skrbniku izboljšav prikaže obrazec spodaj. Ob tem lahko izbere dva načina poročila prihranka, in sicer takoj ali po enem letu. Če izbere po enem letu, mu prihranka ni potrebno računati sedaj, ampak ga bo o tem opozorila aplikacija čez leto dni. V primeru, da pa izbere možnost takoj, mora izračunati čisti prihranek izboljšave, tako da vnese skupni prihranek in skupni strošek za obdobje enega leta. Poleg tega lahko priloži tudi kakršnekoli datoteke, ki dodatno pojasnjujejo sam izračun prihranka izboljšave.

Obrazec za obravnavo izboljšave

Page

Izračun prihranka izboljšave

Prihranek izboljšave	
Način poročuna prihranka * Tu pride besedilo ...	Prihranek izboljšave bom poročunal: <input checked="" type="radio"/> Takoj <input type="radio"/> Po enem letu
Čisti prihranek * Tu pride besedilo ...	Skupni prihranek: 3000,00 € / leto
	Skupni strošek: 1000,00 € / leto
	Čisti prihranek: 2000,00 € / leto
Priloge Priložene datoteke (ena ali več) z dodatnimi pojasnili, slikami itd. <i>Priloge je potrebno pripeti na obrazec pred potrditvijo izboljšave.</i>	<input type="text"/> Browse... Pripni
Potrdi Prekliči	

Slika 41: Obrazec za izračun prihranka izboljšave

4.1.3.7 Izračun nagrade izboljšave

Sledi še izračun nagrade, kjer je potrebno skrbniku zgolj določiti korekcijske faktorje, ostale stvari pa mu aplikacija avtomatsko poročuna. Sam poročun lahko pred oddajo še shrani, če ga ne izvede do konca. Spodnja slika prikazuje obrazec, preko katerega lahko izračun nagrade tudi izvede.

Obrazec za obravnavo izboljšave

Page

Izračun nagrade izboljšave

Nagrada za izboljšavo	
Korekcijski faktorji predlagatelja * Tu pride besedilo ...	
AD#20	
Osnova za izračun nagrade - glede na GK:	300.00 €
Nagrada glede na % soudeležbe:	150.00 €
Vpliv na VZPD:	<input checked="" type="radio"/> Da <input type="radio"/> Ne
Vpliv na ekologijo:	<input type="radio"/> Da <input checked="" type="radio"/> Ne
Timsko delo:	<input type="radio"/> Da <input checked="" type="radio"/> Ne
Aplikativnost v ostalih družbah:	<input type="radio"/> Da <input checked="" type="radio"/> Ne
GK več kot 1 leto:	<input type="radio"/> Da <input checked="" type="radio"/> Ne
Samostojnost izvedbe:	<input type="radio"/> Da <input checked="" type="radio"/> Ne
Že izplačano - skontacija:	- 60.00 €
Nagrada predlagatelju:	96.00 €
JANEZ KRANJSKI	
Osnova za izračun nagrade - glede na GK:	300.00 €
Nagrada glede na % soudeležbe:	150.00 €
Vpliv na VZPD:	<input checked="" type="radio"/> Da <input type="radio"/> Ne
Vpliv na ekologijo:	<input type="radio"/> Da <input checked="" type="radio"/> Ne
Timsko delo:	<input type="radio"/> Da <input checked="" type="radio"/> Ne
Aplikativnost v ostalih družbah:	<input checked="" type="radio"/> Da <input type="radio"/> Ne
GK več kot 1 leto:	<input type="radio"/> Da <input checked="" type="radio"/> Ne
Samostojnost izvedbe:	<input type="radio"/> Da <input checked="" type="radio"/> Ne
Že izplačano - skontacija:	- 60.00 €
Nagrada predlagatelju:	102.00 €
Skupaj za izplačilo:	198.00 €

Shrani Potrdi Prekliči

Slika 42: Obrazec za izračun nagrade izboljšave

4.1.3.8 Zaključevanje izboljšave

Izboljšavo je potrebo nato zgolj še zaključiti. To lahko skrbnik izboljšav stori preko obrazca na spodnji sliki, kjer je potreben zgolj končni komentar. Ta je že predpripravljen s strani

aplikacije. Podatki, ki se spreminjajo od izboljšave do izboljšave, pa so označeni v legendi avtomatskih podatkov. Slika spodaj prikazuje ta obrazec.

Slika 43: Obrazec za zaključek izboljšave

4.1.4 Težave in možnosti izboljšav

Skozi proces izdelave aplikacije se mi je pojavilo tudi nekaj težav. Pri samem obrazcu za prijavo izboljšav je bila prva težava, kako enostavno evidentirati tako zaposlene v korporaciji, ki so v imeniku AD, in tiste, ki jih ni – proizvodni delavci. Če bi hotel to rešiti s SharePointovim gradnikom PeopleEditor, bi se obrazec zelo zakompliciral. Tako sem raje naredil svoj gradnik, ki to zmora dosti enostavneje, saj lahko med drugimi prednostmi evidentira tako tiste uporabnike, ki so v imeniku AD, kot tiste, ki jih ni. Ta gradnik bom v nadaljevanju svoje diplomske naloge tudi predstavil.

Naslednji problem pri istem obrazcu je bila zahteva, da je možno na obrazec pripeti več prilog. S SharePointovim gradnikom za priloge se lahko priloži samo eno priložo, zato sem tega modificiral tako, da to omogoča. Pri prilaganju datotek se mi je pojavil še ta problem, kaj naj naredim z že priloženimi datotekami, če uporabnik obrazca za prijavo izboljšave ni niti shranil niti oddal. V ta namen sem samo shranjevanje prilog uredil tako, da se te pred shranjevanjem ali oddajo obrazca shranjujejo v neko začasno mapo, katere vsebina se briše enkrat dnevno.

Seveda je bilo tukaj potrebno rešiti še več manjših problemov v zvezi s tem, kako se kakšno stvar programira v SharePointu, saj je bila ta aplikacija prva, ki sem jo programiral na platformi SharePoint. Med te probleme lahko npr. štejemo problem, kako prikazati statusno

vrstico in obvestilo v njej, kako odpreti neko stran v pojavnem (pop-up)²¹ oknu, kjer so ostale zadeve zatemnjene, kako programsko shranjevati v SharePoint sezname, kako razdeliti nek seznam na strani in omogočiti skoke med njimi itd.

Možnosti izboljšav je kar nekaj. V prihodnosti bi lahko podatkovno bazo za to aplikacijo preselili na nek zunanji Microsoft SQL strežnik in potem SharePoint sezname iz te zunanje podatkovne baze ustvarili avtomatsko s pomočjo storitve BDC za delo z zunanjimi podatki, ki sem jo predstavil v poglavju 3.5 DELO Z ZUNANJIMI PODATKI (BUSINESS DATA). V sami aplikaciji se ta sprememba ne bi nikjer odražala, saj bi bila množica vseh SharePoint seznamov, ki jih aplikacija potrebuje za delovanje, popolnoma enaka. To bi zgolj prineslo manj problemov ob selitvi podatkov aplikacije na nek nov SharePoint strežnik.

²¹ Pojavno (pop-up) okno je podrejeno okno, navadno v oknu spletnega brskalnika, ki se prikaže na zahtevo spletne aplikacije.

4.2 APLIKACIJA ZA NADZOR TERMINALOV

Naročnik, natančneje oseba znotraj poslovne enote HIDRIA IT, ki je zadolžena za nadzor nad terminali, je želela vzpostaviti nov sistem, preko katerega bi bilo mogoče nadzorovati delovanje terminalov v vseh podjetjih znotraj korporacije.

Terminal predstavlja računalnik v zanj prilagojenem ohišju, skupaj z zaslonom in tipkovnico. Na terminal je lahko priklopljen eden ali več strojev preko dodatnega mikrokrmilnika²², ki poskrbi za komunikacijo s strojem. Kot stroj si lahko predstavljamo stroj za štancanje, stroj za razrez pločevine itd.



Slika 44: Izgled terminala ROT001

Terminal je žično ali brezžično povezan v omrežje s strežnikom, na katerem teče programska oprema ProInfo. ProInfo je sistem MES, čigar osnovna naloga je spremljanje, zajem in analiza podatkov v realnem času tako v proizvodnji kot tudi v servisni in projektni dejavnosti.

Terminal od nanj priklopljenega stroja oz. strojev zahteva določene podatke. Ti podatki so lahko dveh tipov, in sicer:

- število udarcev in
- energija.

²² Mikrokrmilnik ali mikrokontroler je čip, ki vsebuje skoraj vse sestavine mikroročunalnika (procesor, notranji pomnilnik, vmesnike ...).

Te podatke nato terminal pošilja strežniku ProInfo, ta pa jih obdela. Obdelani podatki so zatem prikazani na zaslonu terminala.

Takih terminalov je znotraj korporacije že kar lepo število (približno 90) in se neprestano povečuje. Zaradi velikega števila prav teh je potreba po centralnem nadzoru delovanja terminalov in s tem tudi delovanja strojev, ki so povezani z njimi, še toliko večja.

4.2.1 Vzroki za nastanek aplikacije

Tukaj velja najprej omeniti, da je pred nastankom moje aplikacije za nadzor terminalov podobna aplikacija že obstajala. Spisana je bila v sedaj že zastarelem ogrodju ASP, tako kot večina starejših aplikacij, saj so v korporaciji nastale že v prejšnjem desetletju.

Hidria

Hidria Rotomatika :: nadzor terminalov

Vnos Popravek Brisanje Test

Terminal	Opis	Datum	Ura	?	Sporočilo	VNC	ProInfo	?	Stroj	1	Stroj	2	Stroj	3	Stroj	4
AET00000	AET00000 AET Tolmin - Dieseltec hala C	21.11.2011	18:27		TIME	Odprl	-									
AET00001	AET00001 AET Tolmin - Dieseltec hala C	21.11.2011	18:28		TIME	Odprl	-									
AET00002	AET00002 AET Tolmin - Dieseltec hala C	21.11.2011	18:28		TIME	Odprl	-									
AET00003	AET00003 AET Tolmin - Dieseltec hala C	21.11.2011	18:28		TIME	Odprl	-									
AET00004	AET00004 AET Tolmin - Dieseltec hala C	21.11.2011	18:28		TIME	Odprl	-									
AET00005	AET00005 AET Tolmin - Dieseltec hala D	21.11.2011	18:28		TIME	Odprl	-									
AET00008	AET00008 AET Tolmin - Keramika hala A	21.11.2011	18:28		TIME	Odprl	-									
AET00009	AET00009 AET Tolmin - Orodjarna hala B	21.11.2011	18:27		TIME	Odprl	-									
AET00010	AET00010 AET Tolmin - Odprema	21.11.2011	18:27		TIME	Odprl	-									
AET00012	AET00012 AET Tolmin - Gozdna	21.11.2011	18:28		TIME	Odprl	-									
HPT001	TERM1 - Hidria Perles - KR - Hala 2	21.11.2011	18:28		TIME	Odprl	-									
HPT002	TERM2 - Hidria Perles LAMELE - KR - Hala 2	21.11.2011	18:28		TIME	Odprl	2.3.0.0	●								
HPT004	TERM4 - Hidria Perles - KR - Hala 2			☹		Odprl	2.3.0.0	●								
HPT005	TERM5 - Hidria Perles - KR - ???	21.11.2011	18:28		TIME	Odprl	2.3.0.0									
HPT007	TERM7 - Hidria Perles - KR - Hala 1	21.11.2011	18:28		TIME	Odprl	-									
HPT009	TERM9 - Hidria Perles - KR - Hala 1	21.11.2011	18:27		TIME	Odprl	-									
HPT010	TERM10 - Hidria Perles LAMELE - KR - Hala 2	21.11.2011	18:27		TIME	Odprl	2.3.0.0		233110	☹	233120	☹	234150	☹	233720	☹
HPT011	TERM11 - Hidria Perles - KR - Hala 1	21.11.2011	18:27		TIME	Odprl	-									
HPT015	TERM15 - Hidria Perles - KR - Motorji	21.11.2011	18:28		TIME	Odprl	-									
HPT019	TERM19 - Hidria Perles - KR - Merilnica	21.11.2011	18:28		TIME	Odprl	-									
HPT021	TERM21 - Hidria Perles - KR - Merilnica	21.11.2011	18:27		TIME	Odprl	-									
HPT027	TERM27 - Hidria Perles - KR - Merilnica	21.11.2011	18:28		TIME	Odprl	-									
ROT001	ROT001 - Rotomatika - SPI - TC - Orodjarna	21.11.2011	18:27		TIME	Odprl	2.0.5.0	●								
ROT003	ROT003 - Rotomatika - SPI - TC - Orodjarna	21.11.2011	18:28		TIME	Odprl	-									
ROT004	ROT004 - Rotomatika - SPI - Lamele - Lanserji	21.11.2011	18:28		TIME	Odprl	2.0.5.0	●								
ROT005	ROT005 - Rotomatika - SPI - Livarna - Obdelava	21.11.2011	18:28		TIME	Odprl	2.3.0.0		31363	☹	31302	☹				

Copyright by Hidria Rotomatika 2006 - 2008 (Miro Vehar), info@hidria-rotomatika.si

Slika 45: Stara aplikacija za nadzor terminalov

Kot je mogoče videti iz zgornje slike, aplikacija predstavlja seznam vseh terminalov. V seznamu se nahajajo podatki, kot so opis terminala, zadnje javljanje terminala, nameščena verzija ProInfra ter seznam vseh strojev, ki so priključeni nanj. Če je datum in ura zadnjega javljanja terminala večja od nekega dovoljenega odmika od trenutnega časa, se to izriše z žalostnim smeškom. Enako velja za vsak stroj, ki je priključen na terminal, posebej. Rdečo

piko aplikacija izriše pri tistih terminalih, kjer na terminalu ni nameščene programske opreme ProInfo oz. kjer na terminal ni priključenih strojev. Preko te aplikacije je možno tudi dodajati nove terminale oz. popravljati ter brisati obstoječe, pogledati podrobnosti posameznega stroja, priključenega na terminal ter se na terminal povezati preko programske opreme VNC.

Največje težave pri tej stari aplikaciji in s tem povezan vzrok za nastanek nove predstavlja osveževanje seznama terminalov ter prikaz vseh vrst napak na posameznem terminalu in strojih, ki so priključeni nanj.

Osveževanje seznama pri stari aplikaciji namreč poteka tako, da se po nekem določenem intervalu stran v brskalniku avtomatsko v celoti ponovno naloži. To pa postane še kako moteče pri pregledovanju seznama, saj se ob tem prikaz strani premakne na začetek/na vrh, pa tudi sam proces osvežitve za to potrebuje opazen časovni interval, medtem pa je stran prazna.

Sam prikaz vseh vrst napak na posameznem terminalu in posameznem stroju, ki je povezan nanj, je pri stari aplikaciji slab, ker je pri velikem številu terminalu zelo težko hitro ugotoviti, pri kateremu je prišlo do težav. Napaka se prikaže v obliki rdeče pike ali žalostnega smeška, kar lahko človek ob velikem številu terminalov in strojev kaj kmalu spregleda. Ker se število terminalov neprestano povečuje, bi bila ta težave vsakič le večja.

Tukaj je potem še kar nekaj manjših vzrokov za nastanek nove aplikacije. Ti so sledeči:

- nazornejša predstavitev tega, na kateri lokaciji znotraj podjetja in v katerem podjetju znotraj korporacije se nahaja posamezen termina;
- nazornejša predstavitev tega, ali je terminal v omrežje HIDRIE povezan brezžično ali žično;
- možnost filtriranja seznama terminalov po vseh parametrih, ki so predstavljeni s stolpci seznama.

Nenazadnje je potrebno tu omeniti še to, da lahko med vzroke za nastanek nove aplikacije štejemo tudi vizijo poslovne enote Hidria IT, da se vse spletne aplikacije, ki so bile v korporaciji HIDRIA kdajkoli narejene in so še v uporabi, preseli na platformo SharePoint.

Sama aplikacija za nadzor terminalov je v pisanju diplomske naloge v fazi uvajanja. Seveda se je potrebno ob tem zavedati, da bodo tudi po končanju uvedbe aplikacije s strani naročnikov še vedno prihajale zahteve za nove popravke in dodelave aplikacije, kar je v fazi življenja posamezne aplikacije popolnoma normalno, sicer lahko ta čez čas postane popolnoma neuporabna.

4.2.2 Izgradnja aplikacije

Projekt izdelave aplikacije za nadzor terminalov je že na prvi pogled manj obsežen kot projekt izdelave aplikacije za prijavo in obdelavo izboljšav, ki sem jo predstavil v prejšnjih poglavjih.

V nadaljevanju bom predstavil proces nastajanja same aplikacije. Tekom samega procesa sem izvedel več majhnih sestankov s posamezniki ter po koncu faze nastajanja aplikacije še en večji sestanek z vsemi skupaj.

Prvi sestanki so bili namenjeni temu, da se spoznam s problematiko, ki jo pokriva aplikacija, določimo tehnične in vsebinske zahteve ter proces izvajanja aplikacije. Nato je sledila izdelava aplikacije in ko je bila ta dokončana, je sledil zadnji, večji sestanek. Pregledali smo delovanje aplikacije ter definirali še nekatere popravke.

V nadaljevanju bom predstavil tehnične in vsebinske zahteve, vloge uporabnikov, organizacijo podatkov ter sam proces nadzora terminalov s to aplikacijo.

4.2.2.1 Tehnične in vsebinske zahteve

Vsekakor je bila želja vseh nas, da pripravimo enotno vstopno točko za ogled in nadzor delovanja vseh terminalov znotraj korporacije HIDRIA in z njimi povezanih strojev. Ta enotna vstopna točka naj bi bila ob tem čim bolj uporabniku prijazna, iz nje naj bi se dalo čim hitreje pridobiti čim več informacij na čim bolj enostaven način.

Tako je nastal seznam vseh zahtev:

- Aplikacija naj pridobiva podatke o terminalih iz strežnika ProInfo preko zunanje podatkovne baze Microsoft SQL.
- Aplikacija naj prikazuje seznam vseh terminalov, kjer naj bodo v stolpcih prikazane naslednje lastnosti posameznega terminala:
 - o ime terminala,
 - o tip omrežja terminala,
 - o podjetje terminala,
 - o lokacija terminala,
 - o opis terminala,
 - o datum in ura zadnjega javljanja terminala,
 - o nameščena verzija programske opreme ProInfo na terminalu,
 - o stroj 1, ki je priključen na terminal,
 - o stroj 2, ki je priključen na terminal,
 - o stroj 3, ki je priključen na terminal in
 - o stroj 4, ki je priključen na terminal.
- Podatki v seznamu naj se posodablja tako, da to ne bo motilo uporabnika pri pregledovanju.
- Napake na posameznem terminalu ali napake na posameznem stroju, ki je priključen na terminal, naj bodo vidne na prvi pogled s spremembo barve celotne vrstice.
- Za posamezen terminal naj bo že v seznamu videno, ali je v omrežje priključen brezžično ali žično.
- Podatke o posameznem terminalu naj bo možno popravljati.
- Podatke o posameznem stroju naj bo možno popravljati.
- Aplikacija naj omogoča dodajanje/brisanje strojev posameznemu terminalu.
- Za vsak terminal naj bo možno vpisovati seznam dogodkov. Posamezen dogodek naj vsebuje opis, opombo ter informacijo o tem, kdo ga je izvedel oz. kdo ga je vnesel.
- Aplikacija naj omogoča urejanje šifrantov vseh lokacij.
- Šifrant vseh podjetij naj bo fiksna - urejalo se ga bo ročno v podatkovni bazi.

- Seznam terminalov naj bo mogoče filtrirati po vseh parametrih/stolpcih.

4.2.2.2 Vloge uporabnikov

Tudi v tej aplikaciji imamo uporabniške vloge. Razlika od aplikacije za prijavo in obdelavo izboljšav je v tem, da imamo tu samo dve uporabniški vlogi.

VLOGA UPORABNIKOV	OPIS VLOGE
Vsi zaposleni	Ta uporabniška vloga ima najmanj pravic. V njo avtomatično spada vsak zaposleni znotraj korporacije HIDRIA. Tak uporabnik ima zgolj pravico ogleda seznama terminalov ter strojev, priključenih nanj. Seznam se mu prikazuje vključno z vsemi oznakami in podatki, kot so vrsta omrežne povezava, morebitne napake na terminalu oz. strojih itd. Prav tako lahko uporabnik te uporabniške vloge vidi dogodke, ki so bili zabeleženi na določenemu terminalu.
Administratorji	Ta uporabniška vloga ima največ pravic. Poleg pravic, ki jih imajo vsi zaposleni, ima uporabnik te uporabniške vloge še pravice ažuriranja seznama vseh strojev, ki so priklopljeni na določen terminal. Prav tako ima uporabnik pravice dodajanja novih dogodkov, ki so se dogodili na določenem terminalu, ažuriranja šifrantov lokacij, spreminjanja podatkov terminalov in spreminjanja podatkov strojev.

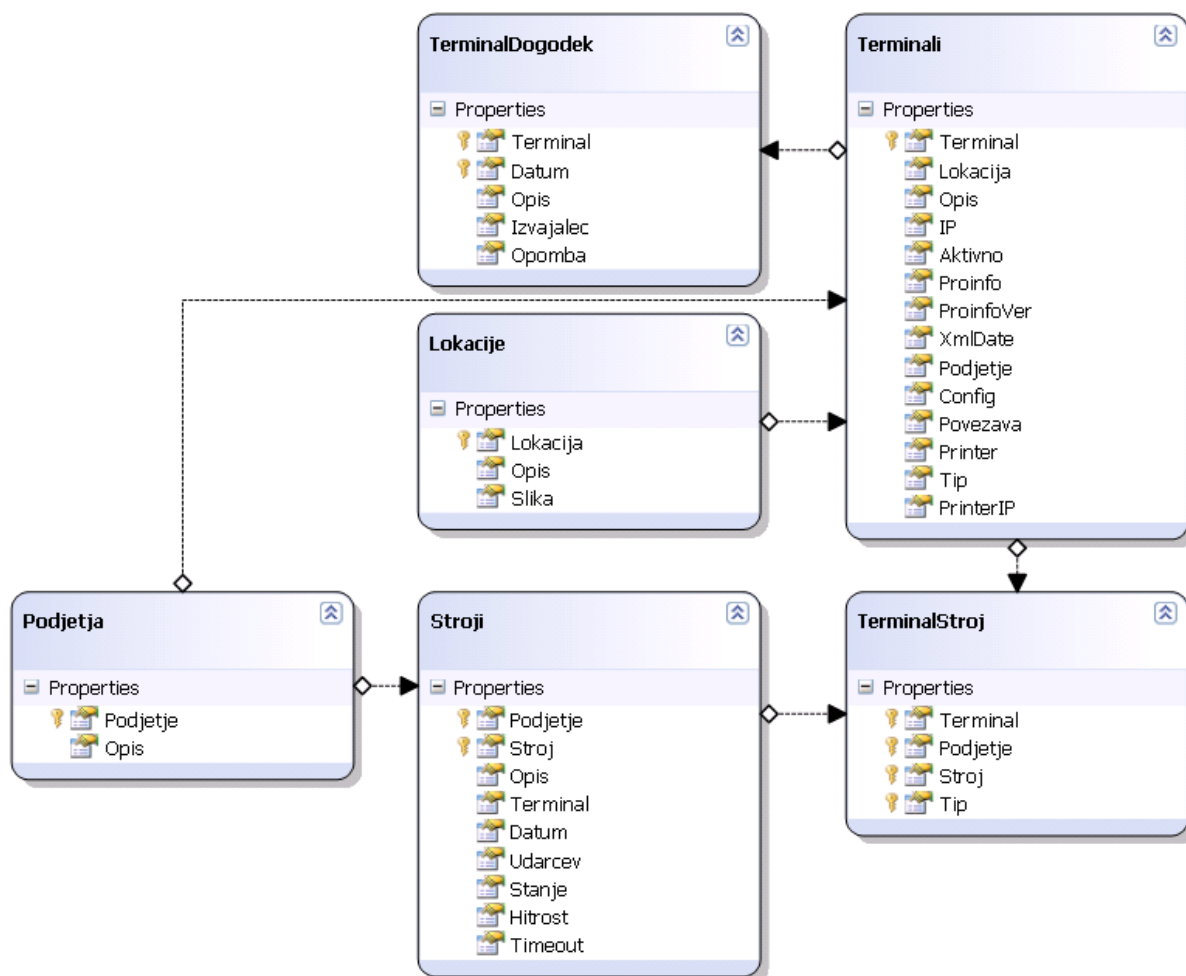
Tabela 26: Vloge uporabnikov pri aplikaciji za nadzor terminalov

4.2.2.3 Organizacija podatkov

Podatki, ki jih aplikacija prikazuje, niso shranjeni v nekem SharePoint seznamu, ampak aplikacija pobira podatke iz zunanje podatkovne baze Microsoft SQL.

Za dostop do tabel zunanje podatkovne baze pri tej aplikaciji nisem uporabil že predstavljene storitve BDC za delo z zunanjimi podatki, saj se mi pri tej aplikaciji to preprosto ni zdelo smiselno zaradi majhnega števila tabel, ki jih podatkovna baza za to aplikacijo vsebuje, po drugi strani pa niti ni potrebe, da bi tabele zunanje podatkovne baze s storitvijo BDC spreminjal v SharePoint sezname, saj jih pri tej aplikaciji ne potrebujemo. Za branje zunanje podatkovne baze sem zato raje uporabil komponento LINQ, ki pretvori jezikovne konstrukte SQL v jezikovne konstrukte LINQ in obratno.

Tabele v zunanji podatkovni bazi je pripravil sodelavec, ki je zadolžen za nadzor nad programsko opremo ProInfo ter povezavo te podatkovne baze s strežnikom ProInfo. Sam sem nato med tabelami ustvaril še relacije ter obnašanja ob brisanju in posodabljanju, saj to omogoča še lažje delo s podatki preko komponente LINQ.



Slika 46: Organizacija podatkov aplikacije za nadzor terminalov

V glavni tabeli *Terminali* so shranjeni podatki o vseh terminalih.

PARAMETER	OPIS
Terminal	Ime terminala.
Lokacija	ID lokacije, na kateri se nahaja terminal.
Opis	Opis terminala.
IP	IP naslov terminala.
Aktivno	Zastavica, ali je terminal aktiven ali ne.
ProInfo	Zastavica, ali je na terminalu programska oprema ProInfo nameščena ali ne.
ProInfoVer	Verzija nameščene programske opreme ProInfo.
XmlDate	Datum zadnjega javljanja terminala strežniku.
Podjetje	ID podjetja, v katerem se nahaja terminal.
Config	Vsebina konfiguracijske datoteke XML.
Povezava	Tip povezave terminala z omrežjem (brežžična, žična).
Printer	Ime tiskalnika, ki je priključen na terminal [če obstaja].

Tabela 27: Parametri tabele Terminali

Parametra *ProInfoVer* in *XmlDate* v tabeli avtomatsko posodablja strežnik ProInfo, ostale parametre pa ureja ta aplikacija.

V tabeli *Stroji* so shranjeni podatki o vseh strojih.

PARAMETER	OPIS
Podjetje	ID podjetja, v katerem se stroj nahaja.
Stroj	Šifra stroja.
Opis	Opis stroja.
Terminal	Ime terminala, na katerega je stroj priključen.
Datum	Datum zadnjega javljanja stroja strežniku.
Udarcev	Število udarcev.
Stanje	Stanje stroja.
Hitrost	Hitrost stroja.

Tabela 28: Parametri tabele *Stroji*

Parametre *Datum*, *Udarcev*, *Stanje* in *Hitrost* v tej tabeli avtomatsko posodablja strežnik ProInfo, ostale parametre pa ureja ta aplikacija.

V tabeli *Podjetja* se nahaja šifrant vseh podjetij znotraj korporacije. Parametri posameznega podjetja so samo številka ID podjetja (parameter *Podjetje*) ter ime podjetja (parameter *Opis*). Vsebina te tabele se bo urejala kar ročno v podatkovni bazi, zato uporabniškega vmesnika zanjo ni potrebno izdelovati. Povezana je s tabelo *Terminali* in *Stroji* in s tem določa, kje se posamezni terminal oz. stroj nahaja.

V tabeli *Lokacije* se nahaja šifrant vseh lokacij znotraj korporacije. Parametri posamezne lokacije so številka ID lokacije (parameter *Lokacija*), ime lokacije (parameter *Opis*) ter slika lokacije (parameter *Slika*) s ptičje perspektive in z označbami, kje se posamezni terminal znotraj lokacije nahaja. Povezana je s tabelo *Terminali* in s tem določa, na kateri lokacije se posamezni terminal nahaja.

Tabeli *Stroji* in *Terminali* povezuje vmesna tabela *TerminalStroj*. Parametri povezovalne tabele so poleg primarnih ključev obeh tabel, ki ju povezuje, še parameter *Tip*, ki določa, kaj terminal na stroju spremlja – število udarcev ali energijo.

Tu imamo še tabelo *TerminalDogodek*, ki je namenjena beleženju dogodkov na posameznem terminalu. Poleg imena terminala (*Terminal*), kateremu pripada dogodek, vsebuje še naslednje parametre:

PARAMETER	OPIS
Datum	Datum vnosa dogodka.
Opis	Opis dogodka.
Izvajalec	Kdo je izvajalec dogodka oz. kdo je dogodek zabeležil.
Opomba	Dodatne opombe na dogodek [če so potrebne].

Tabela 29: Parametri tabele *TerminalDogodek*

4.2.3 Delovanje aplikacije

Glavni del aplikacije predstavlja stran s pregledom vseh terminalov. Tu so vsi terminali prikazani naenkrat (ni deljenja seznama terminalov na strani).

Nadzorna plošča



Pregled vseh terminalov

Terminal	R	O	T	Podjetje	Lokacija	Opis	Datum in ura	ProInfo	Stroj 1	Stroj 2	Stroj 3	Stroj 4
ROT001	S		/	Hidria Movent	Koper	TC - Orodjarna	25.10.2011 13:19	2.0.5.0	234567	234567	123456	123456
ROT003	S		/	Hidria Movent	Koper	TC - Orodjarna	/	/	7684321	/	/	/
ROT004	S		/	Hidria Movent	Koper	Lanserji	25.10.2011 13:26	2.0.5.0	/	/	/	/
ROT005	N		/	Hidria Movent	Koper	Obdelava	25.10.2011 13:26	2.0.5.0	/	/	/	/
ROT006	S		/	Hidria Movent	Koper	Lamele	25.10.2011 13:26	2.0.5.0	/	/	/	/
ROT007	N		/	Hidria Movent	Koper	Lamele - Rotor	25.10.2011 13:26	2.0.5.0	/	/	/	/
ROT008	N		/	Hidria Movent	Koper	Lamele	/	/	/	/	/	/
ROT009	N		/	Hidria Movent	Koper	Obdelava	/	/	/	/	/	/
ROT011	S		/	Hidria Movent	Koper	Motorji	25.10.2011 13:26	2.0.5.0	/	/	/	/
ROT012	S		/	Hidria Movent	Koper	Motorji	25.10.2011 13:26	2.0.5.0	/	/	/	/
ROT013	S		/	Hidria Movent	Koper	Livarna	25.10.2011 13:26	2.0.5.0	/	/	/	/

Posodobljeno: 21.11.2011 ob 18:41:32

Slika 47: Glavna stran s pregledom vseh terminalov

Prvi stolpec prikazuje z barvo status delovanja določenega terminala.

BARVA STATUSA	OPIS
Zelena	Terminal in stroji delujejo pravilno.
Oranžna	Terminal je potrebno povezati s strežnikom ProInfo (parametra XmlDate ter ProInfo se ne posodabljata).
Rdeča	Terminal oz. stroji ne delujejo pravilno – pojavila se je ena izmed naslednjih možnih napak: <ul style="list-style-type: none"> - terminal se ne odziva na pakete ICMP²³, s katerimi preverjamo, ali je ta dosegljiv (pinganje); - strežnik ProInfo ni dobil novejšega odziva terminala; - strežnik ProInfo ni dobil novejšega odziva enega izmed strojev, priključenih na terminal.

Tabela 30: Barve statusov delovanja določenega terminala

V primeru da terminal oz. stroji ne delujejo normalno, se z rdečo barvo pobarva celotna vrstica zaradi lažjega in hitrejšega pregleda napak.

Aplikacija tako za vsak terminal najprej preveri, ali je v parametru *XmlDate* in *ProInfo* sploh kaj zapisano. Tako ugotovi, ali je terminal povezan s strežnikom ProInfo.

Če sta podatka zapisana, aplikacija poskuša pingati terminal po IP naslovu, kjer za dovoljeni časovni interval (timeout) odgovora vzamem 150 ms. Za pinganje sem se odločil zato, ker človek v primeru, ko so z nekim oddaljenim računalnikom problemi, največkrat najprej stori

²³ Internet Control Message Protocol (ICMP) se uporablja za pošiljanje nadzornih sporočil in sporočil stanja internetnega omrežja [21].

to, da ta oddaljen računalnik poizkusi pingati. Tako aplikacija kar namesto človeka izvede pošiljanje paketa ICMP strežniku in če ne dobi odgovora, to prikaže z rdečo barvo.

Če je oddaljen računalnik oz. terminal vrnil odgovor na paket ICMP, aplikacija preveri, če se datum in ura, vpisana v parametru *XmlDate*, razlikuje od trenutnega datuma in ure za manj kot dve minuti. V primeru da to ne drži, pobarva vrstico z rdečo barvo. Enako velja tudi za stroje, priklopljene na terminalu, le da v tem primeru namesto parametra *XmlDate* primerja parameter *Datum*. Aplikacija prikaže opis napak v pojavnem oknu, ki se prikaže, ko uporabnik pridrži kazalec na prvem stolpcu, ki prikazuje status terminala.

S kakšno vrsto povezave je terminal povezan v omrežje, prikazuje ikona v stolpcu O (okrajšava za omrežje), če pa na njej pridržimo kazalec, se nam v pojavnem oknu prikaže še števila IP terminala.

↑	R	O	T	Podjetje	Loka
	S		/	Hidria Movent	Koper
	S		/	Hidria Movent	Koper
	S		/	Hidria Movent	Koper
	N			Hidria Movent	Koper
	S		/	Hidria Movent	Koper
	N				
	N		/	Hidria Movent	Koper

IP naslov: 10.101.34.12
 Tip povezave: Brezžična povezava

Slika 48: Prikaz dodatnih informacij o vrsti povezave

Stroji posameznega terminala so prikazani v dveh barvah. Barvi predstavljata, kaj na stroju spremljamo – energijo ali število udarcev. Tako je to enostavno mogoče ugotoviti že iz pregleda vseh terminalov. Če pridržimo kazalec na posameznem stroju, se nam pokažejo vsi njegovi parametri.

	Stroj 1	Stroj 2	Stroj 3	Stroj 4
	234567	234567	123456	123456
	76543			/
	/			/
	/			/
	/			/

Stroj: 234567 - Testni stroj 3
 Podjetje: 310 - Hidria Movent
 Spremljanje: Energija
 Datum in ura: /
 Št. udarcev: /
 Stanje: /
 Hitrost: /

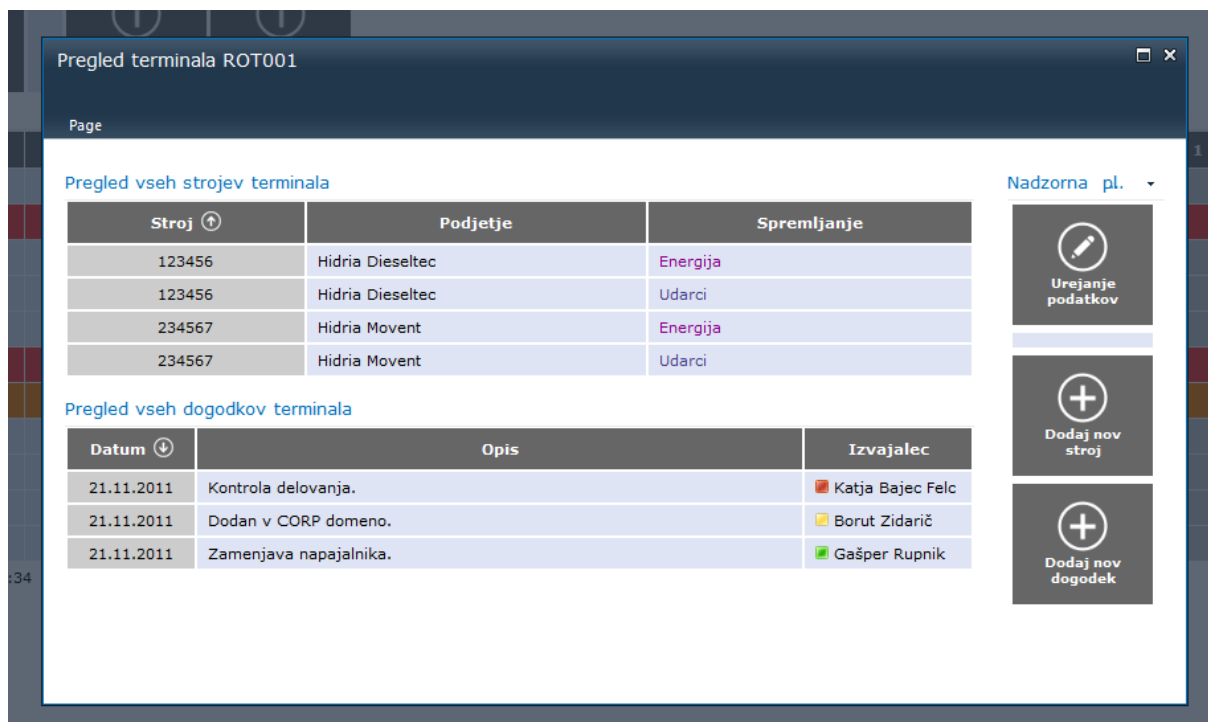
Slika 49: Prikaz strojev terminala

Ena izmed glavnih zahtev je bila osveževanje pregleda vseh terminalov, ne da bi to motilo uporabnika, ki bi si pregled terminalov ogledoval. To je bilo v stari aplikaciji rešeno s

ponovnim nalaganjem celotne strani, tukaj pa v ta namen uporabljam AJAX²⁴ komponento UpdatePanel, ki omogoča nemoteno posodabljanje. Posodobitev sproža časovnik, ki se sproži vsakih 20 sekund, prav tako pa je možno posodobitev sprožiti ročno že prej s klikom na gumb za posodobitev seznama nad pregledom vseh terminalov.

4.2.3.1 Podrobnosti posameznega terminala

S klikom na posamezni terminal se pokaže stran s podrobnostmi posameznega terminala. Podrobnosti posameznega terminala zavzemajo pregled vseh strojev terminala ter pregled vseh dogodkov terminala.



Slika 50: Prikaz podrobnosti posameznega terminala

Pregled vseh strojev terminala obsega seznam vseh strojev ter informacijo o tem, kaj na posameznem stroju spremljamo (udarce ali energijo). Tako je lahko v tem pregledu nek stroj dvakrat, prvič spremljamo udarce, drugič energijo.

Pregled vseh dogodkov terminala predstavlja seznam vseh dogodkov, ki so se kdajkoli zgodili na terminalu. Dogodki so ob velikem številu razdeljeni po straneh. Prikazani so z datumom, opisom ter izvajalcem. Ura se prikaže, ko uporabnik pridrži kazalec na datumu dogodka, opombe pa, ko uporabnik pridrži kazalec na opisu dogodka. Izvajalec je prikazan s statusom in dodatnimi funkcionalnostmi za komuniciranje preko programske opreme Microsoft Lync. Za izbiro izvajalca sem uporabil gradnik za izbiro oseb, ki sem ga izdelal sam in ga bom predstavil v nadaljevanju kot enega izmed projektov.

²⁴ Asinhroni JavaScript (AJAX) je skupina medsebojno povezanih spletnih razvojnih tehnik, uporabljenih za ustvarjanje interaktivnih spletnih aplikacij. Z njim si lahko spletne aplikacije izmenjujejo podatke s strežnikom asinhrono v ozadju, brez potrebe po ponovnem nalaganju strani [22].

Tu lahko uporabnik, ki spada pod uporabniško vlogo administratorja, ureja podatke o posameznem terminalu, dodaja terminalu nove stroje in dogodke.

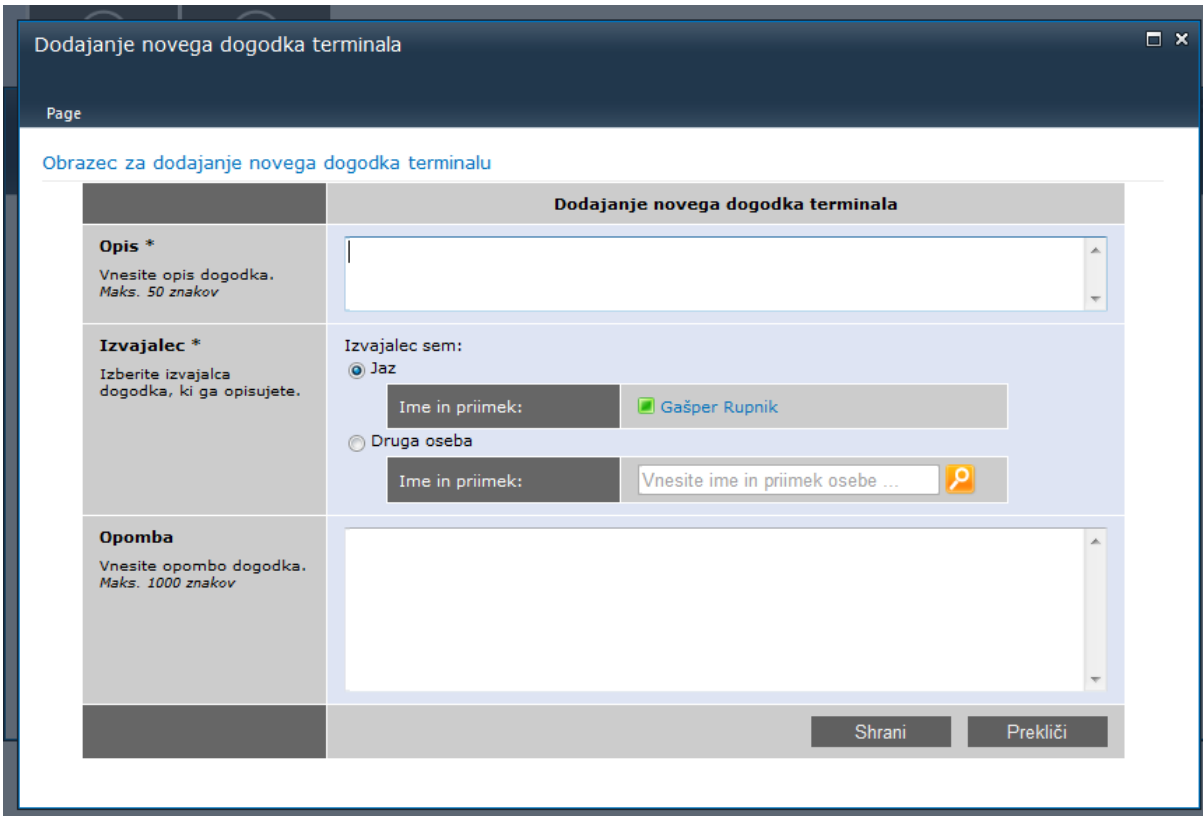
Funkcionalnost urejanja podatkov o posameznem terminalu omogoča, da preko obrazca spremenimo nastavitve, kot je ime terminala, naslov IP, tip omrežne povezave, podjetje, lokacija, opis, dodatne parametre (ali je terminal aktiven, ali je ProInfo nameščen), konfiguracijsko datoteko XML in tiskalnik. Podatke terminala je možno popraviti ali pa enostavno izbrisati terminal iz podatkovne baze.

Dodajanje novega terminala	
Terminal * <small>Vnesite identifikacijsko ime terminala. Maks. 10 znakov</small>	ROT001
Omrežje * <small>Vnesite IP naslov računalnika in izberite tip omrežne povezave terminala.</small>	IP naslov: 10.101.34.36 Tip omrežne povezave: <input checked="" type="radio"/> Žična povezava <input type="radio"/> Brezžična povezava
Podjetje * <small>Izberite podjetje, v katerem se terminal nahaja.</small>	310 (Hidna Movent)
Lokacija * <small>Izberite lokacijo, v kateri se terminal nahaja.</small>	KP (Koper)
Opis * <small>Vnesite dodaten opis terminala. Maks. 50 znakov</small>	TC - Orodjarna
Dodatni parametri * <small>Označite dodatne parametre terminala.</small>	<input checked="" type="checkbox"/> Terminal aktiven <input type="checkbox"/> ProInfo nameščen
XML config * <small>Vnesite vsebino XML config datoteke terminala.</small>	<test> Testno besedilo </test>
Tiskalnik <small>Vnesite ime priključenega tiskalnika. Maks. 10 znakov</small>	Ime tiskalnika: <input type="text"/> IP naslov: <input type="text"/>
<input type="button" value="Izbrši"/> <input type="button" value="Popravi"/> <input type="button" value="Prekliči"/>	

Slika 51: Urejanje podatkov o posameznem terminalu

Pri dodajanju novega stroja terminalu je potrebno zgolj izbrati stroj, ki obstaja v šifrantu strojev, ter tip uporabe – ali bomo spremljali število udarcev ali energijo. Stroje terminala lahko nato urejamo oz. brišemo tako, da v pregledu vseh strojev terminala kliknemo na posamezni stroj. Odpre se nam obrazec za spreminjanje/brisanje stroja terminala.

Ko pa želimo terminalu dodati nov dogodek, se nam pokaže obrazec, kot ga je možno videti na spodnji sliki. Od nas zahteva opis, izbiro izvajalca ter opsijsko tudi opombo. Kot izvajalca privzeto ponudi uporabnika, ki ima ta obrazec trenutno odprt, kar ugotovi glede na to, kdo je trenutno prijavljen v SharePointu. Možno pa je izbrati tudi drugo osebo, za kar sem uporabil gradnik za izbiro oseb, ki sem ga naredil sam in ga bom predstavil v enem izmed naslednjih poglavij.



The screenshot shows a web form titled "Dodajanje novega dogodka terminalu". The form is divided into three main sections on the left, each with a corresponding input area on the right:

- Opis ***: "Vnesite opis dogodka. Maks. 50 znakov". The input area is empty.
- Izvajalec ***: "Izberite izvajalca dogodka, ki ga opisujete." It features two radio buttons: "Jaz" (selected) and "Druga oseba". Under "Jaz", there is a field "Ime in priimek:" with the value "Gašper Rupnik". Under "Druga oseba", there is a field "Ime in priimek:" with a placeholder "Vnesite ime in priimek osebe ..." and a search icon.
- Opomba**: "Vnesite opombo dogodka. Maks. 1000 znakov". The input area is empty.

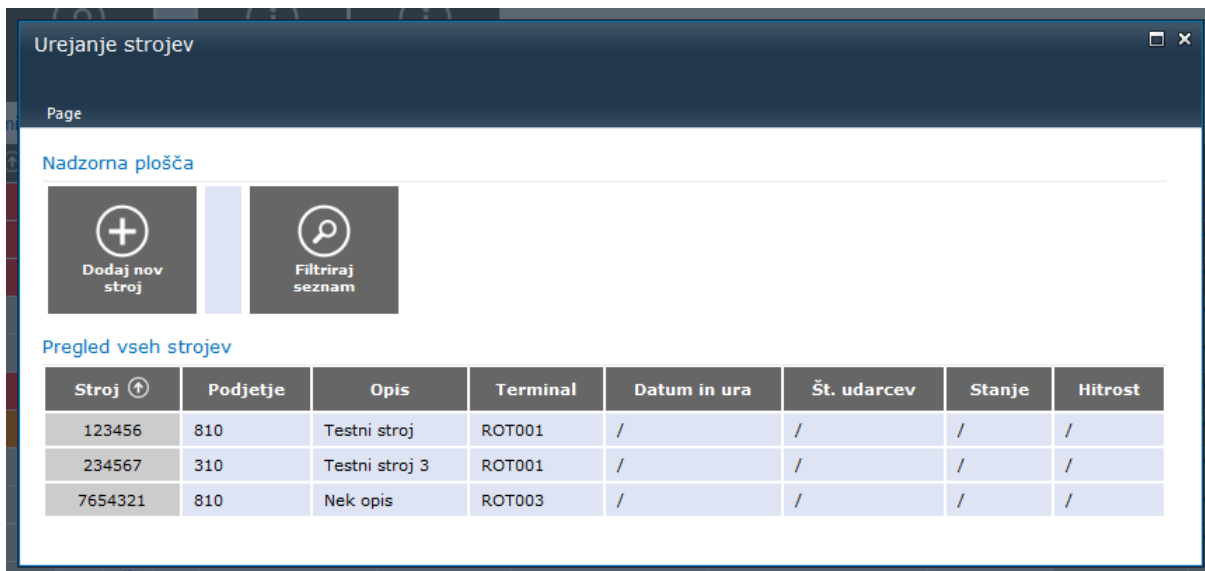
At the bottom right of the form are two buttons: "Shrani" and "Prekliči".

Slika 52: Dodajanje novega dogodka terminalu

Na glavni strani lahko uporabnik z uporabniško vlogo Administrator poleg posodabljanja ter filtriranja seznama ureja še šifranta vseh strojev in lokacij.

4.2.3.2 Šifranti aplikacije

Šifrant strojev deluje na podoben način kot pregled vseh terminalov, saj ravno tako z barvami prikazuje status posameznega stroja glede na parametre datum in ura, št. udarcev, stanje in hitrost. Ostale tri podatke (stroj, podjetje in opis) pa lahko uporabnik ureja preko te aplikacije s klikom na posamezni stroj.



Slika 53: Prikaz šifranta strojev

Pri šifrantu lokacij lahko tako urejamo šifro posamezne lokacije in opis ter dodajamo sliko, ki predstavlja ptičjo perspektivo lokacije skupaj s označenimi terminali.



Slika 54: Prikaz šifranta lokacij

4.2.4 Težave in možnosti izboljšav

Skozi proces izdelave aplikacije se mi je pojavilo par težav, ki pa sem jih vse uspel tudi rešiti.

Prva težava je definitivno bila ta, kako pripraviti avtomatski sistem osveževanja podatkov v pregledu vseh terminalov, da bi bil čim bolj nemoteč za uporabnike. Kot že rečeno, sem to enostavno rešil z AJAX komponento UpdatePanel ter sprožilcem na časovniku.

Druga težava je bil ping terminalov. Ob nalaganju seznama terminalov pri pregledu vseh terminalov mora aplikacija pingati vsakega izmed terminalov. V primeru, da od nobenega ne dobi odziva, bi tako nalaganje strani trajalo kar precej časa. Ta problem sem rešil tako, da aplikacija počaka na odgovor terminala ob pošiljanju paketa ICMP maksimalno 150 ms, kar

se je izkazalo za pravo vrednost, saj je minimalna vrednost, ki je še sprejemljiva za omrežje korporacije HIDRIA.

Drugih večjih težav ni bilo. Glede samih možnosti izboljšav pa menim, da bi se dalo pripraviti še nekaj dodatnih funkcionalnosti. Ena izmed možnosti za izboljšave bi lahko bilo avtomatsko ugotavljanje tipa omrežne povezava (brezžična ali žična) glede na IP naslov terminala.

Poleg tega bi lahko aplikacijo v naslednjih fazah opremili s sistemom za avtomatsko opozarjanje o napakah na posameznem terminalu preko elektronske pošte, sporočila SMS itd. Še več, lahko bi pripravili aplikacijo za pametne telefone, ki bi omogočala vse to, kar omogoča ta aplikacija za SharePoint, poleg tega pa bi lahko aplikacija kar preko potisnih obvestil obveščala o napakah na posameznem terminalu. Tako bi bil odzivni čas odgovornih za nadzor terminalov še hitrejši.

4.3 DODATEK ZA GRAFIČNI PRIKAZ OPRAVIL

Kot naslednji SharePointov projekt bom predstavil tistega, ki je od mene zahteval dosti manj časa kot oba predhodnika. V tem primeru ne gre za SharePoint aplikacijo ampak za dodatek za grafični prikaz opravil, ki se nahajajo v nekem SharePointovem seznamu za opravila.

Naročnik, natančneje direktor poslovne enote HIDRIA IT, mi je dal nalogo, naj preverim, ali je ustrezna rešitev na voljo že v osnovni verziji SharePointa ali pa obstaja kak dodatek, s katerim bi lahko grafično prikazovali opravila, ki so navedena v SharePointovemu seznamu za opravila.

V SharePointu za to obstaja prav poseben dodatek, ki opravila izrisuje v Ganttovem diagramu²⁵, vendar ta ne ustreza našim potrebam in zahtevam, saj nudi preveč informacij in zato trpi tudi enostavnost prikaza opravil.

4.3.1 Vzroki za nastanek dodatka

Na sliki spodaj imamo primer vpisanih nalog oz. opravil v SharePointovem seznamu za opravila. Človek si ob taki obliki prikaza opravil težko predstavlja, kakšna je odvisnost med njimi, kakšen je njihov status itd. Dosti boljšo predstavbo bi dobili že z enostavnim grafom vozlišč in povezav med njimi.

Type	Title	Assigned To	Status	Priority	Due Date	% Complete	Predecessors
	Vzpostavitev sistema informacijske varnosti v korporaciji	Boštjan Tušar	In Progress	(2) Normal	1/21/2012	10 %	
	Varovanje intelektualne lastnine	Miloš Šturm	In Progress	(2) Normal	3/7/2012	15 %	Vzpostavitev sistema informacijske varnosti v korporaciji
	Tehnično varovanje	Miha Menard	In Progress	(2) Normal	3/14/2012	5 %	Vzpostavitev sistema informacijske varnosti v korporaciji
	Kadri in procesi	Tanja Kenda	In Progress	(2) Normal	9/6/2011	50 %	Vzpostavitev sistema informacijske varnosti v korporaciji

Slika 55: Primer vpisanih opravil v SharePointovem seznamu za opravila

Naročnik si je želel, da bi bila opravila prikazana z enostavnimi objekti (kvadrat, krog, elipsa itd.), ki so med seboj povezani, povezava pa predstavlja časovno odvisnost enega opravila od drugega – natančneje, če sta opravila A in B povezani in je opravilo A bližje korenu grafa opravil, se mora opravilo A izvesti pred opravilom B oz. je opravilo B časovno odvisno od opravila A. Barva objekta predstavlja status, v katerem se opravilo nahaja. V posameznem objektu, ki seveda predstavlja opravilo, so zapisani ime opravila ter njegove podrobnosti, kot so rok za izvedbo ter ime in priimek odgovornega.

²⁵ Ganttov diagram je namenjen predstavitvi opravil projekta, ki jih lahko uporabljate za spremljanje datumov in poteka dela na opravilih posameznih ekip [23].



Slika 56: Želje naročnika za izgled prikaza opravil

Iz takšnega grafa je lažje ugotoviti, v kakšni fazi so posamezna opravila, kje se je zalomilo – opravilo ni bilo dokončano do roka, kdo je za to odgovoren, kako so opravila odvisna med sabo (kakšen je vrstni red izvajanja opravil) itd.

Sam sem nato preveril na svetovnem spletu za že obstoječe rešitve tega problema, kjer sem se osredotočil zgolj na brezplačne, saj se mi je zdela rešitev tega problema dokaj enostavna in bi zato raje naredil svojo, kot pa naročniku predlagal nakup druge. Ker rešitve tega problema nisem dobil na svetovnem spletu, sem se zato lotil z izdelavo svoje.

4.3.2 Izgradnja dodatka

Projekt izdelave dodatka za grafični prikaz opravil seveda ni tako obsežen, kot vsi do sedaj predstavljeni projekti, vendar ga je zaradi zanimive rešitve vredno prikazati v tem mojem diplomskem delu.

Tekom samega procesa ni bilo potrebe po izvedbi kakršnegakoli sestanka, saj je bilo potrebno naročnika kontaktirati zgolj ob nujnih stvareh, na koncu izdelave pa je seveda sledil kratek sestanek, zgolj kot pregled delovanja ter načrti za nadaljnjo izboljševanje, saj je naročnik delovanje samega dodatka lahko preveril že prej.

V nadaljevanju bom predstavil tehnične in vsebinske zahteve, vloge uporabnikov, organizacijo podatkov ter delovanje dodatka.

4.3.2.1 Tehnične in vsebinske zahteve

Želja naročnika je bila poenostavitev prikaza nalog posameznega zaposlenega znotraj poslovne enote HIDRIA IT. Ker je sam prikaz seznama nalog lahko ob velikem številu nalog precej nepregleden in ker Ganttov diagram za naše potrebe ponuja preveč informacij, je bilo potrebno za dober prikaz seznama nalog natančno določiti zahteve.

Tako je nastal seznam vseh zahtev, ki se glasijo:

- V grafu posamezno nalogo prikažemo kot objekt poljubne oblike.
- Soodvisnost dveh nalog prikažemo s povezavo med njima.
- Graf nima ciklov.
- Vsaka naloga ima maksimalno enega predhodnika.
- Začetna naloga je samo ena.
- Posamezna naloga v grafu je predstavljena z naslednjimi podrobnostmi:
 - naslov naloge z večjo pisavo,
 - status naloge:

STATUS	OPIS
0	Se ne izvaja.
2	V začetku izvajanja.
3	Sredina izvajanja.
4	Izvajanje skoraj končano.
5	Izvajanje končano.

Tabela 31: Seznam statusov naloge in pripadajoči opisi

- datum predvidenega končanja naloge,
- zadolženi za nalogo in
- barva objekta, ki predstavlja nalogo, je pobarvana v skladu s statusom:

STATUS	BARVA
0	Bela
2	Rdeča
3	Oranžna
4	Zelena
5	Modra

Tabela 32: Seznam statusov naloge in pripadajoče barve

4.3.2.2 Vloge uporabnikov

V tej aplikaciji za razliko od prejšnjih nimamo uporabniških vlog, saj gre tu zgolj za dodatek SharePointovemu seznamu za opravila in je namenjen grafičnemu prikazu opravil. Grafičen prikaz lahko torej vidi vsak zaposleni, ki lahko vidi tudi SharePointov seznam za opravila.

4.3.2.3 Organizacija podatkov

O kakšni posebni organizaciji podatkov pri tej aplikaciji ni govora, saj za delovanje uporablja zgolj en SharePointov seznam za opravila.

SharePointov seznam za opravila vsebuje naslednje stolpce oz. parametre posamezne naloge:

- Tip naloge (Type),
- **Naziv naloge (Title),**
- **Zadolženi (Assigned To),**
- **Status (Status),**
- Prioriteta (Priority),

- **Datum končanja (Due Date),**
- **% narejenega (% Complete) in**
- **Predhodniki (Predecessors).**

Od vseh teh so za nas potrebni samo tisti, ki so v zgornjem seznamu odebeljeni, poleg tega pa sem parameter »Predhodniki« spremenil tako, da je namesto več predhodnikov možen samo en. Tako je dodatek za grafičen prikaz opravil prilagojen samo za delo s temi parametri.

<input type="checkbox"/> Naziv	<input type="checkbox"/> Zadolžen	Status	Datum končanja	Predhodnik
Vzpostavitev sistema informacijske varnosti v korporaciji	Boštjan Tušar	(2) V začetku	1/21/2012	
Varovanje intelektualne lastnine	Miloš Šturm	(2) V začetku	3/7/2012	Vzpostavitev sistema informacijske varnosti v korporaciji
Tehnično varovanje	Miha Menard	(2) V začetku	3/14/2012	Vzpostavitev sistema informacijske varnosti v korporaciji
Kadri in procesi	Tanja Kenda	(3) Sredina	9/6/2011	Vzpostavitev sistema informacijske varnosti v korporaciji
Informacijske tehnologije	Boštjan Tušar	(3) Sredina		Vzpostavitev sistema informacijske varnosti v korporaciji
Nadzor dostopov	Boštjan Tušar	(2) V začetku		Informacijske tehnologije
Uvedba sistema za spremljanje varnostnih incidentov	Boštjan Tušar	(4) Skoraj končano		Ostale naloge
Neprekinjeno poslovanje	Ervin Pagon	(3) Sredina		Informacijske tehnologije
Vzdrževanje UPS-ov	Janko Lapajne	(4) Skoraj končano		Neprekinjeno poslovanje
Ostale naloge	Boštjan Tušar	(3) Sredina		Vzpostavitev sistema informacijske varnosti v korporaciji
Videonadzor objektov	Miha Menard	(2) V začetku		Tehnično varovanje
Ureditev nalepk	Tatjana Bolčina Kacin	(5) Končano		Videonadzor objektov
Izobraževanje uporabnikov	Tatjana Bolčina Kacin	(0) Se ne izvaja		Ostale naloge
Poenotenje Jantar	Tanja Kenda	(3) Sredina		Kadri in procesi
Ureditev server sob (Klima, UPS, dostop, alarm, javljanje)	Miha Menard	(4) Skoraj končano		Informacijske tehnologije

[+ Add new item](#)

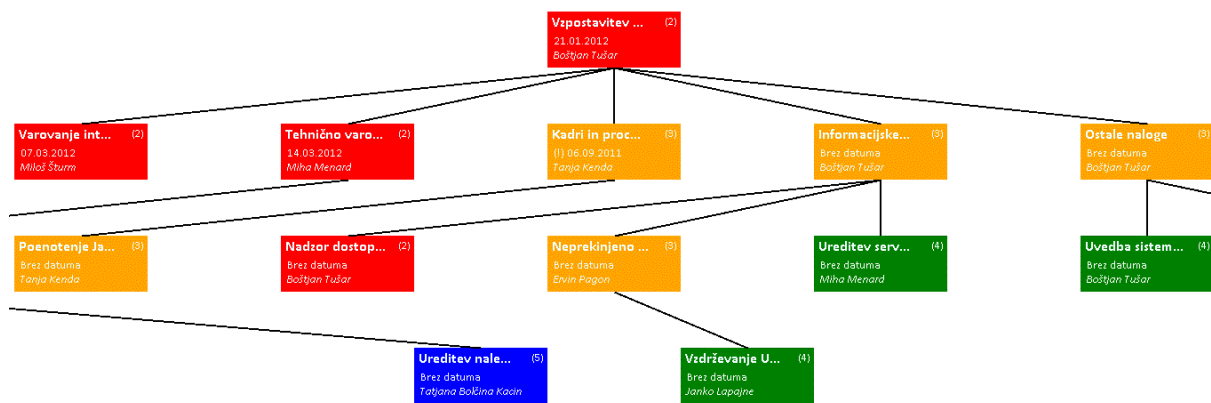
Slika 57: Prirejen SharePointov seznam za prikaz opravil

4.3.2.4 Delovanje dodatka

Ko so bile zbrane vse zahteve in določena organizacija podatkov, je prišla na vrsto sama izgradnja dodatka. Ker je za nas dovolj, da imamo pri nekem seznamu opravil samo eno začetno opravilo (torej opravilo, ki je brez predhodnikov), da ima neko nezačetno opravilo lahko samo enega predhodnika, ciklov ni, dobimo za izris poenostavljen graf, ki mu natančneje pravimo drevo²⁶.

Pojavljala so se mi vprašanja v zvezi s tem, kako bi na SharePoint strani sploh lahko risal - ali ima mogoče SharePoint že vgrajeno kakršnokoli komponento za lažje izrisovanje. Izrisovanje bi po vrhu vsega moralo biti še interaktivno – s posodabljanjem seznama opravil bi se moral izris drevesa avtomatsko posodabljati (drevo bi se večalo ali manjšalo, odvisno od tega, ali bi opravila dodajali ali brisali). Po dobrem premisleku sem spisal del strani (Web Part) za SharePoint, ki tako drevo tudi izriše.

²⁶ Drevo je nelinearna podatkovna struktura. Osnovni elementi v drevesu so vozlišča. Vsako vozlišče ima lahko več naslednikov, vendar kvečjemu enega predhodnika. Koren je vrhnje vozlišče, edino vozlišče v drevesu brez predhodnika [24].



Slika 58: Izgled izrisa drevesa

Dodatek za grafičen prikaz opravil ob osveževanju strani v SharePointovemu seznamu za opravila najprej poišče začetno vozlišče – torej vozlišče, ki nima nobenega predhodnika. To vozlišče označi za obiskano.

Nato poišče vsa vozlišča, ki imajo za predhodnika to začetno vozlišče, ki je seveda že bilo obiskano, in jih označi za obiskana. Za vsako od teh novih vozlišč nato poišče naslednja vozlišča, ki imajo za predhodnika prav njih itd. Uporabljam torej t.i. iskanje v širino (Breadth-first search)²⁷ kot postopek preiskovanja seznama opravil. S tem postopkom zgradim sredinsko poravnano drevo, prav sredinska poravnava pa mu omogoči še jasnejšo in enostavnejšo predstavitev na sliki prikaza opravil.

Zadnji problem, ki se mi je nato še pojavil, je bil, kako tako drevo tudi izrisati. To sem rešil s pomočjo knjižnice za izrisovanje *System.Drawing* programskega jezika C#. S pomočjo nje na delu strani (Web Partu) ustvarim sliko GIF, na kateri izrišem graf. Višina slike je enaka številu zaporednih stopenj opravil, širina slike pa maksimalnemu številu opravil na neki stopnji. Barva vsakega vozlišča v grafu pomeni status opravila, ki ga vozlišče predstavlja.

Če malo podrobneje pogledamo zgornjo sliko, si lahko celotno sliko GIF predstavljamo kot mrežo, sestavljeno iz celic širine X in višine Y. Ker je drevo sredinsko poravnano, je začetno vozlišče drevesa (opravilo) vedno na sredini. V naslednjih stopnjah drevesa, kjer je vozlišče znotraj neke stopnje več kot samo eno, so sosednja vozlišča med sabo ločena ravno za eno celico mreže. Tak način izrisovanja mi je bil najlažji za realizacijo in sem zato lahko zanj porabil manj časa, kar je bil tudi cilj tega projekta.

Ker ta dodatek ob nalaganju strani vsakič na novo izriše drevo, ni bilo problemov z zahtevo, da mora biti izrisovanje avtomatsko ob posodabljanju SharePointovega seznama za opravila, saj se pri dodajanju, popravljanju in brisanju nekega elementa iz seznama, stran vsakič ponovno naloži in zato tudi graf ponovno izriše.

²⁷ Iskanje v širino je eno najpreprostejših iskanj po grafu. Reče se mu tudi "Breadth first search" (BFS). Iskanje v širino običajno implementiramo z vrsto. (najdena vozlišča dodajamo na koncu v vrsto, pregledujemo pa prva v vrsti). Časovna zahtevnost je $O(V+E)$. Začnemo v nekem vozlišču S in najprej poiščemo vse njegove sosede. Potem od vsakega najdenega soseda poiščemo njegove sosede, vozlišča, ki jih nismo našli v prvem koraku. Nadaljujemo, dokler ne najdemo ciljnega vozlišča ali pa ne preiščemo celotnega grafa [7].

4.3.3 Težave in možnosti izboljšav

Velika, ampak tudi edina težava je bila prav gotovo to, kako sploh drevo izrisati na SharePointovo stran. Imel sem idejo, da bi to naredil enostavno s html tabelo, kjer bi bile celice tabele vozlišča v grafu. Problem bi v tem primeru bil, kako na nek pameten način izrisati povezave med njimi. Zaradi tega sem na koncu izbral izrisovanje na sliko, saj si s tem načinom pri izrisovanju zelo svoboden.

Možnosti izboljšav pri tem dodatku je kar nekaj. Te sva z naročnikom že pregledala in določila, s čim se bo uporabnost dodatka čez čas še izboljšala. Pa si jih pogledjmo:

- V SharePointov seznam za opravila dodamo stolpec za komentarje o opravilih. Komentarji se na grafičnem prikazu vidijo v balončku, ko se z miško postaviš na določeno opravilo na sliki.
- Ob daljših naslovih opravil namesto treh pik (»...«) raje uporabimo dinamično raztezanje okvirčka, ki predstavlja določeno opravilo (tako opravila ne bodo več fiksne velikosti).
- Na grafu naj bodo razmiki med opravili manjši, da lahko pride grafičen prikaz čim manjši.
- Ko nastane veliko opravil, je težko vedeti, kam novo opravilo dodati, zato bi pred ime opravila dodali še številčenje in sicer na naslednji način:
 - o Prvi nivo: 1
 - o Drugi nivo: 2.1, 2.2, 2.3 ...
 - o Tretji nivo: 3.1, 3.2 ...
 - o Itd.
- Ob kliku na kvadrateg v grafu, ki predstavlja opravilo, naj se odprejo podrobnosti posameznega opravila. Tako ne bo za to potrebno iskati opravila v SharePointovemu seznamu za opravila.

4.4 GRADNIK ZA IZBIRO OSEB

Kot zadnji SharePointov projekt bom predstavil gradnik za izbiro oseb. Z izbiro oseb je mišljeno iskanje neke osebe v imeniku AD na eni izmed SharePointovih strani. Pri programiranju na platformi SharePoint imamo v ta namen na voljo t.i. gradnik PeopleEditor, ki ga je moč videti na sliki spodaj. Več o vzrokih za izdelavo svojega gradnika pa v nadaljevanju.



Slika 59: SharePointov gradnik PeopleEditor

Naročnika za ta dodatek praktično ni bilo, saj sem do ideje za izdelavo prišel kar sam, ker menim, da je gradnik PeopleEditor lahko v nekaterih primerih znotraj korporacije HIDRIA zelo težaven, skorajda neuporaben.

4.4.1 Vzroki za nastanek gradnika

Znotraj korporacije smo imeli do sedaj za vsako podjetje svojo domeno. Ravno v času pisanja diplomske naloge pa izvajamo prehod vseh zaposlenih iz vseh podjetij na novo skupno domeno. Za selitev uporabnikov na novo domeno uporabljamo orodje Active Directory Migration Tool (ADMT).

Po uspešnem zaključku selitve nekega uporabnika na novo domeno ima uporabnik dva uporabniška računa – enega še vedno na stari domeni in enega na novi, saj se starega ne briše. Ker bomo brisanje starih domen in z njimi povezanih starih uporabniških računov izvajali šele po uspešni selitvi vseh uporabnikov iz vseh domen, v tem trenutku SharePointov gradnik PeopleEditor najde za take uporabnike dva uporabniška računa. To lahko predstavlja malce zmede pri manj večjih uporabnikih računalnikov, ko iščejo neko osebo, saj ne moremo od njih zahtevati, da vedo, kaj predstavlja uporabniški račun »ROTOMATIKA\gasrup« in kaj »CORP\gasrup«. »ROTOMATIKA\gasrup« je namreč namišljen račun nekega uporabnika iz stare domene, »CORP\gasrup« pa račun istega uporabnika iz nove domene. Problem postane še večji, ko v vnosno polje gradnika PeopleEditor vnesemo zgolj uporabniško ime »gasrup«, gradnik pa nam sicer najde uporabnika, vendar njegov stari/neaktivni račun – »ROTOMATIKA\gasrup«, kar avtomatično pomeni napačen zadetek.

Upal bi si trditi, da je malo takih oseb v korporaciji, ki poznajo uporabniška imena ostalih zaposlenih, da bi lahko z gradnikom PeopleEditor neko osebo poiskali samo po uporabniškem imenu. Večina si jih najverjetneje osebo lažje zapomni po imenu ali priimku oz. kar po obeh hkrati. Problem je v tem, da gradnik PeopleEditor najde neko osebo po uporabniškem imenu sprejemljivo hitro (v velikih primerih vrne napačen uporabniški račun zaradi migracij), kar pa ne velja za iskanje osebe po imenu in/ali priimku. V primeru, ko imamo v imeniku AD okoli 1500 uporabnikov, traja tako iskanje približno pol minute, kar pa za marsikaterega uporabnika ni sprejemljivo. Hitrejše iskanje osebe po imenu in/ali priimku je sicer možno z dodatnim

iskalnikom, ki ga lahko na gradniku PeopleEditor priključimo s klikom na drugo ikono poleg vnosnega polja (ikona v obliki odprte knjige), vendar nam v tem primeru še vedno ostanejo podvajanja uporabnikov zaradi migracij.

Problem z gradnikom PeopleEditor lahko predstavlja tudi situacija, ko želimo z njim evidentirati tudi uporabnike, ki jih v imeniku AD ni. V korporaciji HIDRIA namreč proizvodnih delavcev nimamo v imeniku AD – torej ne morejo dostopati do službenih računalnikov, nimajo službenega elektronskega naslova itd. Ta problem bi sicer lahko rešili tako, da bi v takih primerih poleg gradnika PeopleEditor postavili še eno vnosno polje za vpis imena in priimka uporabnika, ki ga v imeniku AD ni. Vendar bi bilo dosti enostavneje, če bi to znal kar gradnik za izbiro oseb.

Vse to so vzroki, ki so me pripeljali do tega, da sem izdelal svoj gradnik, ki reši vse zgoraj našteje probleme.

4.4.2 Izgradnja gradnika

Podobno kot za prejšnji projekt velja tudi za tega. Projekt ni obsežen, ima pa zanimivo rešitev zgoraj predstavljenih problemov, zato ga je prav tako vredno prikazati v tem mojem diplomskem delu.

Ker tu naročnika ni bilo, saj sem zamisel za izdelavo dobil sam, tudi sestanki niso bili potrebni. Gradnik sem po zaključku izdelave uporabil tako v aplikaciji za prijavo in obdelavo izboljšav kot v aplikaciji za nadzor terminalov, ki sem ju že predstavil v predhodnih poglavjih.

V nadaljevanju bom predstavil tehnične in vsebinske zahteve, vloge uporabnikov, organizacijo podatkov ter samo delovanje gradnika.

4.4.2.1 Tehnične in vsebinske zahteve

Ker sem tokrat v vlogi naročnika kar sam, sem tudi tehnične in vsebinske zahteve sestavil sam. Zahteve so nastale iz prej navedenih problemov, ki se pojavljajo ob uporabi gradnika PeopleEditor.

Te zahteve so naslednje:

- gradnik naj omogoča iskanje tako oseb, ki so v imeniku AD, kot oseb, ki jih ni;
- gradnik naj zna v primeru podvajanja oseb zaradi migracij sam izbrati pravo;
- iskanje osebe naj bo možno po imenu in/ali priimku;
- iskanje po imenu in/ali priimku naj bo izvedeno v sprejemljivem času;
- v primeru več oseb z enakim imenom in priimkom naj gradnik ponudi vse zadetke s podrobnostmi za lažjo izbiro prave osebe;
- gradnik naj najdenim osebam prikazuje status z dodatnimi funkcionalnostmi za komuniciranje preko programa Microsoft Lync.

4.4.2.2 Vloge uporabnikov

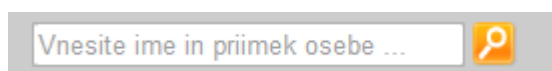
Tudi v tej aplikaciji nimamo uporabniških vlog, saj gre v tem primeru samo za gradnik, ki ga lahko vstavimo na neko SharePoint stran, tej pa lahko določamo svoje uporabniške vloge in s tem povezane pravice za uporabo tega gradnika.

4.4.2.3 Organizacija podatkov

Pri tem gradniku ni v ozadju nobenega SharePointovega seznama ali posebne zunanje podatkovne baze, ki bi jo gradnik potreboval za delovanje. Gradnik za delovanje uporablja le imenik AD.

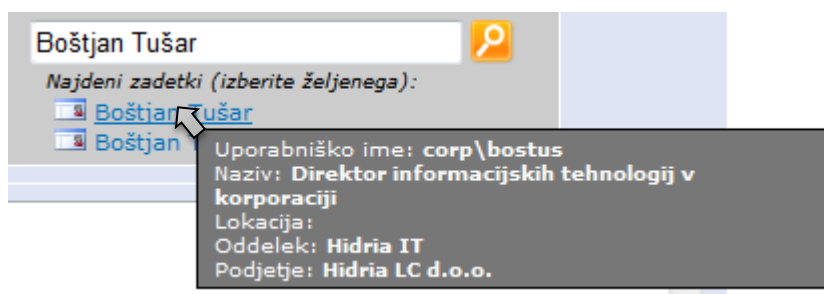
4.4.2.4 Delovanje gradnika

Ko sem določil vse zahteve, je prišla na vrsto sama izgradnja gradnika. Zamislil sem si gradnik z enim vnosnim poljem in enim gumbom, da bo le ta čim bolj enostaven. V vnosno polje bo uporabnik vnesel iskalni niz, z gumbom pa pričel proces iskanja.



Slika 60: Izgled mojega gradnika za izbiro oseb

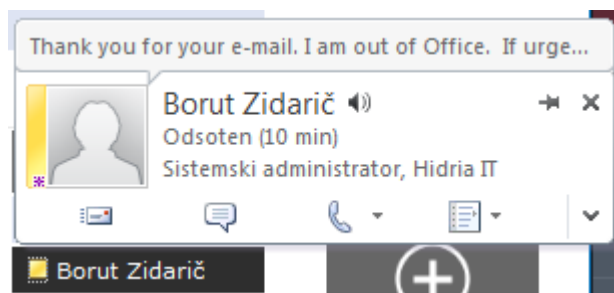
V primeru da bo iskalnik našel več zadetkov, bo gradnik za izbiro prave osebe pod vnosnim poljem prikazal vse zadetke z dodatnimi podatki za lažjo odločitev. Podatki naj bodo uporabni, kot so v imeniku programske opreme Microsoft Outlook²⁸. Uporabnik bo pravo osebo izbral s klikom nanjo.



Slika 61: Izbira prave osebe v primeru več zadetkov

Ob najdbi prave osebe, ki je tudi v imeniku AD (torej ni proizvodni delavec), naj gradnik prikaže status osebe z dodatnimi funkcionalnostmi za komunikacijo preko programske opreme Microsoft Lync.

²⁸ Microsoft Outlook je računalniški program, ki predstavlja odjemalec elektronske pošte. Trži ga podjetje Microsoft, ki ga je tudi razvilo.



Slika 62: Prikaz statusa osebe z dodatnimi funkcionalnostmi za komunikacijo preko programske opreme Microsoft Lync

V vnosno polje lahko uporabnik vnese ime in/ali priimek neke osebe. Ob vnosu ta gradnik najprej odstrani odvečne presledke na začetku in koncu vnosa (funkcija Trim) ter odstrani morebitne dvojne prostore znotraj njega. S tem naredimo gradnik imun na napake uporabnika, ko vnese, kje je kakšen presledek preveč.

Gradnik nato izdelava vse možne permutacije²⁹ besed niza, ki predstavlja ime in/ali priimek iskane osebe. S tem se znebimo problemov glede pravega vrstnega reda imen in priimkov iskane osebe.

Orodje v naslednjem koraku preveri v imeniku AD na vseh možnih domenah vse možne permutacije besed (torej vse možne vrstne rede imen in priimkov iskane osebe). Vse najdene uporabniške račune shranjuje v podatkovno strukturo slovar³⁰, kjer za ključ vnosa v slovar vzame parameter SID³¹ AD uporabnika, ki predstavlja identifikacijsko številko tega uporabnika v imeniku AD in je seveda unikatna. To bo kasneje omogočalo lažje in predvsem hitrejše odstranjevanje najdenih uporabniških računov, ki jih je iskana oseba uporabljala v stari domeni pred migracijo na novo domeno.

Za selitev uporabnikov na novo domeno namreč uporabljamo že omenjeno orodje ADMT. Ta ob selitvi uporabnika na novo domeno v njegov nov uporabniški račun shrani prejšnji SID v atributu SIDHistory, kar nam omogoča, da enostavno s pomočjo prej izdelanega slovarja ugotovimo, kateri uporabniški račun je iskana oseba uporabljal prej in ga torej gradnik ne prikaže.

Glede na število najdenih zadetkov imamo naslednje tri možne scenarije:

- če je najden natančno en zadetek iz imenika AD, je torej gradnik dobil pravo osebo;
- če je najdenih zadetkov več, ponudi uporabniku možnost izbire pravega zadetka z dodatnimi podatki o uporabniku kot so naziv, lokacija, oddelek, podjetje in vzdevek;
- če gradnik ne najde nobenega zadetka, shrani zgolj vpisan ime in priimek kot osebo, ki je ni v imeniku AD – torej je proizvodni delavec.

²⁹ Permutacija je v matematiki z medsebojnimi zamenjavami preurejeno zaporedje znanega končnega števila elementov [25].

³⁰ Slovar je podatkovna struktura, ki ji pravimo tudi preslikava. Uporabljamo jo, kadar imamo podatke, ki so urejeni po geslih ali ključih [26].

³¹ SID predstavlja identifikacijsko številko nekega objekta Active Directory (imenika AD). Objekt imenika AD je lahko uporabnik, računalnik, skupina itd.

4.4.3 Težave in možnosti izboljšav

Večjih težav pri izdelavi tega gradnika ni bilo. Potrebno je bilo le najti nek način, kako ugotoviti, katerih uporabniških računov se ne uporablja več zaradi selitve na novo domeno. Ko sem ugotovil, da se pri selitvi uporabnikov z orodjem ADMT stare SID shranjuje v poseben parameter, je bila rešitev problema že na vidiku.

Pri tem gradniku vidim možnosti izboljšave. Ena izmed možnosti bi lahko bila že ta, da bi gradnik nadgradil v tej smeri, da bi znal osebe iskati tudi po uporabniških imenih. Tega v tej verziji gradnika nisem vključil, saj se ljudje med seboj bolj poznajo po imenih in priimkih kot pa po uporabniških imenih, ki jih uporabljajo znotraj korporacije.

5 ZAKLJUČEK

S platformo SharePoint sem se prvič srečal pri študentskem delu v poslovni enoti HIDRIA IT korporacije HIDRIA. Kot je bilo že omenjeno v uvodu, je korporacija HIDRIA pred kratkim stari intranet zamenjala z novim t.i. Hidrianetom, ki je nastal na platformi SharePoint. Popolne zamenjave starega intraneta z novim pravzaprav še ni bilo, saj se trenutno še vedno uporablja večina spletnih aplikacij s starega intraneta, kjer je iz novega ustvarjena samo povezava do njih. Dejansko se novi intranet na platformi SharePoint trenutno uporablja zgolj kot skladišče raznoraznih datotek, kar je pravzaprav primarna funkcija SharePointa in je to razvidno tudi že iz njegovega imena. Manjka pravzaprav prenos vseh aplikacij s starega intraneta, ki so bile takrat napisane še v navadnem ogrodju ASP, na novi Hidrianet. To dejansko pomeni, da je potrebno stare aplikacij ponovno programirati na način, prilagojen platformi SharePoint.

Iz moje velike radovednosti do spoznavanja novih tehnologij sem zato izven delovnega časa preučeval to platformo, saj se mi je že na prvi pogled zdela zelo zanimiva, prepričan sem bil da veliko obeta. S pomočjo literature, ki sem jo dobil preko svetovnega spleta, sem pridobil vse to znanje in nastali so izdelki, ki sem jih predstavil v tem svojem diplomskem delu. Izkazalo se je, da je platforma SharePoint upravičila vsa moja začetna pričakovanja in imam veliko željo za raziskovanje tudi vnaprej, saj skoraj z vsakim dnem spoznam nekaj novega in to mi da dodaten zagon, ker vidim, kako obsežna je ta platforma.

Vesel sem, da sem imel možnost izdelati dva večja projekta – aplikacijo za prijavo in obdelavo izboljšav ter aplikacijo za nadzor terminalov in s tem pridobil še veliko praktičnih znanj in prijemov pri programiranju platforme SharePoint. Vse to znanje in izkušnje mi bodo lahko še kako prav prišle pri naslednjih projektih na platformi SharePoint. V načrtu je med drugim preseliti na nov Hidrianet tudi najbolj uporabljeno spletno aplikacijo znotraj korporacije HIDRIA – centralno pomoč uporabnikom (CPU). Pri njej vidim kar nekaj podobnosti z aplikacijo za prijavo in obdelavo izboljšav. Prav tako vidim tu idealno priložnost razširiti funkcionalnost te aplikacije tudi na aplikacijo za pametne telefone, saj SharePoint to zelo dobro omogoča.

Svoje znanje, ki sem ga pri preučevanju platforme SharePoint pridobil, pa širim tudi izven meja korporacije HIDRIA. Tako se z njim udeležujem tudi v Osnovni šoli Črni Vrh, kjer programiram na platformi SharePoint dva ločena izdelka. Prvi je spletna ređovalnica, drugi pa spletne učilnice. Ravno ti dve aplikaciji sem pred petimi leti izdelal, takrat še v ogrodju PHP, ter ju predstavil kot maturitetni izdelek. Nekaj manjših izdelkov, kot so aplikacija za izdelavo urnikov in jedilnikov, so v tem času tudi že dokončani.

Urejanje razrednega urnika

Stran

Urniki_Urejanje_VWP

Ure	Od	Do	Ponedeljek	Torek	Sreda	Četrtek
1	7:30	8:15	{prazno}	{prazno}	{prazno}	{prazno}
2	8:20	9:05	LVZ / GVZ	{prazno}	{prazno}	{prazno}
3	9:25	10:10	{prazno}	GEO	{prazno}	{prazno}
4	10:15	11:00	{prazno}	{prazno}	{prazno}	{prazno}
5	11:05	11:50	{prazno}	{prazno}	{prazno}	{prazno}
6	11:55	12:40	{prazno}	{prazno}	{prazno}	{prazno}
7	12:45	13:30	{prazno}	{prazno}	{prazno}	{prazno}
8	13:35	14:20	{prazno}	{prazno}	{prazno}	{prazno}

Predmet 1 *
Izberite enega izmed podanih predmetov kot 1. predmet.

Biologija (BIO)

Neobvezni podatki

Predmet 2
Izberite enega izmed podanih predmetov kot 2. predmet.

{prazno}

Predmet 3
Izberite enega izmed podanih predmetov kot 3. predmet.

{prazno}

Izbrisi Shrani Prekliči

Slika 63: Aplikacija za izdelavo urnikov

Mislím, da je v platformi SharePoint še kar dolga in lepa prihodnost, saj se izdelek iz dneva v dan bolj širi tako med nove uporabnike kot tudi po funkcionalnosti.

KAZALO SLIK

Slika 1: Zgradba platforme SharePoint.....	6
Slika 2: Organizacija strani na platformi SharePoint	7
Slika 3: Centralna administracija v SharePointu.....	8
Slika 4: Osnovno programiranje preko brskalnika	10
Slika 5: Programsko orodje SharePoint Designer 2010.....	11
Slika 6: Visual Studio 2010.....	12
Slika 7: Expression Blend	13
Slika 8: Visual Studio predloge za platformo SharePoint	15
Slika 9: Prikaz soodvisnosti med nekaterimi objekti objektnega modela SharePointa	17
Slika 10: Povezava platforme SharePoint z zunanjo podatkovno bazo v navezi s komponento LINQ..	32
Slika 11: BCS projekt v programskem orodju Visual Studio 2010	33
Slika 12: Prikaz sekvenčnega delovnega toka	34
Slika 13: Prikaz delovnega toka s stanji.....	34
Slika 14: Prikaz nekaterih vgrajenih storitev za delo s SharePoint seznamami.....	36
Slika 15: Izbira varnostnih tehnologij v Centralni administraciji SharePointa.....	38
Slika 16: Windows Phone 7 Office Hub	39
Slika 17: Programiranje aplikacij za OS Windows Phone 7 s programskim orodjem Visual Studio 2010	40
Slika 18: Sprednja stran kartončka "5 minut za izboljšavo"	43
Slika 19: Zadnja stran kartončka "5 minut za izboljšavo"	44
Slika 20: Organizacija podatkov za aplikacijo za prijavo in obdelavo izboljšav	53
Slika 21: Vstopna stran aplikacije za prijavo in obdelavo izboljšav	54
Slika 22: Prikaz statusa posamezne izboljšave	54
Slika 23: Prikaz daljših opisov v pregledu vseh prijav izboljšav	55
Slika 24: Prikaz datuma in ure	55
Slika 25: Prehajanje med stranmi.....	56
Slika 26: Prikaz možnih akcij, legende statusov in urejanja po vseh parametrih seznama.....	56
Slika 27: Prikaz podrobnosti posamezne izboljšave	57
Slika 28: Prva stran obrazca "5 minut za izboljšavo"	58
Slika 29: Druga stran obrazca "5 minut za izboljšavo"	58
Slika 30: UML model procesa prijave in obdelave izboljšav.....	60
Slika 31: Obrazec za prijavo izboljšave	61
Slika 32: Prikaz prijavljene izboljšave	62
Slika 33: Obrazec za strokovno potrditev izboljšave	63
Slika 34: Obrazec za obravnavo izboljšave v primeru še ne izvedene izboljšave	64
Slika 35: Obrazec za obravnavo izboljšave v primeru že izvedene izboljšave.....	65
Slika 36: Prikaz potrjene izboljšave	65
Slika 37: Prikaz izvedene izboljšave.....	66
Slika 38: Obrazec za spremembo statusa izboljšave	66
Slika 39: Beleženje spremembe statusa izboljšave	67
Slika 40: Obrazec za zaključevanje izvajanja izboljšave.....	67

Slika 41: Obrazec za izračun prihranka izboljšave	68
Slika 42: Obrazec za izračun nagrade izboljšave.....	69
Slika 43: Obrazec za zaključek izboljšave.....	70
Slika 44: Izgled terminala ROT001	72
Slika 45: Stara aplikacija za nadzor terminalov	73
Slika 46: Organizacija podatkov aplikacije za nadzor terminalov.....	77
Slika 47: Glavna stran s pregledom vseh terminalov.....	79
Slika 48: Prikaz dodatnih informacij o vrsti povezave	80
Slika 49: Prikaz strojev terminala.....	80
Slika 50: Prikaz podrobnosti posameznega terminala.....	81
Slika 51: Urejanje podatkov o posameznem terminalu	82
Slika 52: Dodajanje novega dogodka terminalu	83
Slika 53: Prikaz šifrantov strojev.....	84
Slika 54: Prikaz šifrantov lokacij.....	84
Slika 55: Primer vpisanih opravil v SharePointovem seznamu za opravila	86
Slika 56: Želje naročnika za izgled prikaza opravil	87
Slika 57: Prirejen SharePointov seznam za prikaz opravil	89
Slika 58: Izgled izrisa drevesa	90
Slika 59: SharePointov gradnik PeopleEditor	92
Slika 60: Izgled mojega gradnika za izbiro oseb.....	94
Slika 61: Izbiro prave osebe v primeru več zadetkov.....	94
Slika 62: Prikaz statusa osebe z dodatnimi funkcionalnostmi za komunikacijo preko programske opreme Microsoft Lync	95
Slika 63: Aplikacija za izdelavo urnikov.....	98

KAZALO TABEL

Tabela 1: Najbolj uporabljeni atributi oz. metode objekta tipa SPContext	18
Tabela 2: Najbolj uporabljeni atributi oz. metode objekta tipa SPFarm.....	18
Tabela 3: Najbolj uporabljeni atributi oz. metode objekta tipa SPWebApplication	19
Tabela 4: Najbolj uporabljeni atributi oz. metode objekta tipa SPSite	20
Tabela 5: Najbolj uporabljeni atributi oz. metode objekta tipa SPWeb	21
Tabela 6: Najbolj uporabljeni atributi oz. metode objekta tipa SPList.....	22
Tabela 7: Najbolj uporabljeni atributi oz. metode objekta tipa SPListItem	23
Tabela 8: Najbolj uporabljeni atributi oz. metode objekta tipa SPFile	24
Tabela 9: Najbolj uporabljeni atributi oz. metode objekta tipa SPField	25
Tabela 10: Najbolj uporabljeni atributi oz. metode objekta tipa SPQuery	26
Tabela 11: Najbolj uporabljeni atributi oz. metode objekta tipa SPUser.....	27
Tabela 12: Najbolj uporabljeni atributi oz. metode objekta tipa SPException	28
Tabela 13: Najbolj uporabljeni atributi oz. metode objekta tipa SPUtility	28
Tabela 14: Najbolj uporabljeni atributi, ki veljajo za vse odzivnike na dogodke	29
Tabela 15: Dogodki na SharePoint seznamih	29
Tabela 16: Dogodki na elemente SharePoint seznama.....	30
Tabela 17: Dogodki na elektronsko pošto.....	31
Tabela 18: Dogodki na delovnih tokovih.....	31
Tabela 19: Metode za preslikavo parametrov iz zunanje tabele v preslikano tabelo.....	32
Tabela 20: Najbolj uporabljene aktivnosti delovnih tokov v SharePointu	35
Tabela 21: Stopnje izboljšav	47
Tabela 22: Statusi izboljšav	51
Tabela 23: Vloge uporabnikov pri aplikaciji za prijavo in obdelavo izboljšav	52
Tabela 24: Pozicije priloženih datotek	54
Tabela 25: Število dni do zaključka posameznega statusa.....	55
Tabela 26: Vloge uporabnikov pri aplikaciji za nadzor terminalov	76
Tabela 27: Parametri tabele Terminali.....	77
Tabela 28: Parametri tabele Stroji	78
Tabela 29: Parametri tabele TerminalDogodek	78
Tabela 30: Barve statusov delovanja določenega terminala	79
Tabela 31: Seznam statusov naloge in pripadajoči opisi.....	88
Tabela 32: Seznam statusov naloge in pripadajoče barve	88

LITERATURA

- [1] Steve Fox, *Beginning SharePoint 2010 Development*, Indianapolis, Canada, 2010.
- [2] (2010) Microsoft SharePoint 2010 Developer Walkthroughs. Dostopno na: <http://www.scribd.com/doc/36635167/Share-Point-2010-Developer-Walk-Through-Guide>
- [3] Todd Klindt, Shane Young, Steve Caravajal, *Professional SharePoint 2010 Administration*, Indianapolis, Canada, 2010, pogl. 9.
- [4] Charlie Holland, *The Complete Reference, Microsoft SharePoint 2010 Web Applications*, McGraw-Hill Companies, 2011, pogl. 14.
- [5] Inovativni center Hidrie, *Inovacijska dejavnost v Hidrii, Pravidnik (v.0)*, Hidria, Spodnja Idrija, 2011.
- [6] (2011) SharePoint 2010 and Windows Phone 7 Training Kit. Dostopno na: <http://www.microsoft.com/download/en/details.aspx?id=26813>
- [7] (2006) Iskanje v širino. Dostopno na: http://wiki.fmf.uni-lj.si/wiki/Iskanje_v_širino
- [8] Sistem ProInfo. Dostopno na: <http://www.omikron.si/proinfo.php>
- [9] Pisarniški program Microsoft Office. Dostopno na: <http://office.microsoft.com/sl-si/>
- [10] Tehnologija Language Integrated Query (LINQ). Dostopno na: http://en.wikipedia.org/wiki/Language_Integrated_Query
- [11] Programska arhitektura Model-View-Controller (MVC). Dostopno na: <http://en.wikipedia.org/wiki/Model-view-controller>
- [12] Programsko ogrodje .NET (.NET framework). Dostopno na: http://en.wikipedia.org/wiki/.NET_Framework
- [13] Content Management System (CMS sistemi). Dostopno na: http://en.wikipedia.org/wiki/Content_management_system
- [14] Microsoft Silverlight. Dostopno na: http://en.wikipedia.org/wiki/Microsoft_Silverlight
- [15] Globally Unique Identifier (GUID). Dostopno na: http://en.wikipedia.org/wiki/Globally_unique_identifier
- [16] Uniform Resource Locator (URL). Dostopno na: http://en.wikipedia.org/wiki/Uniform_Resource_Locator
- [17] Extensible Markup Language (XML). Dostopno na: <http://en.wikipedia.org/wiki/XML>
- [18] Enota za merjenje količine podatkov Byte. Dostopno na: <http://en.wikipedia.org/wiki/Byte>
- [19] Customer Relationship Management (CRM sistemi). Dostopno na: http://en.wikipedia.org/wiki/Customer_relationship_management
- [20] Active Directory (AD) oz. aktivni imenik. Dostopno na: http://en.wikipedia.org/wiki/Active_Directory
- [21] Internet Control Message Protocol (ICMP). Dostopno na: http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
- [22] Asynchronous JavaScript and XML (AJAX). Dostopno na: [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- [23] Gantt chart. Dostopno na: http://en.wikipedia.org/wiki/Gantt_chart
- [24] Tree data structure. Dostopno na: [http://en.wikipedia.org/wiki/Tree_\(data_structure\)](http://en.wikipedia.org/wiki/Tree_(data_structure))

- [25] Permutation. Dostopno na: <http://en.wikipedia.org/wiki/Permutation>
- [26] Associative array. Dostopno na: http://en.wikipedia.org/wiki/Associative_array
- [27] Quest Software. Dostopno na: <http://www.quest.com/sharepoint/>