

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Alen Bečirhodžić

Platforma za aktivacijo licenc

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana 2012

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

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00263/2012

Datum: 11.04.2012

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

Kandidat: **ALEN BEČIRHODŽIČ**

Naslov: **PLATFORMA ZA AKTIVACIJO LICENC**
PLATFORM FOR LICENCE ACTIVATION

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Zasnujte in izdelajte načrt za sistem za aktivacijo licenc. Sistem naj poleg spletne aplikacije za administracijo platforme vsebuje tudi nabor spletnih storitev, ki omogočajo aktivacijo različnih tipov licenc. Za podatkovno bazo uporabite MS SQL server.

Mentor:


doc. dr. Rok Rupnik



Dekan:


prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Alen Bečirhodžić, z vpisno številko **63070433**, sem avtor diplomskega dela z naslovom:

Platforma za aktivacijo licenc

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika,
- 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 19. november 2012

Podpis avtorja:

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Pregled tehnologij in orodij za razvoj aplikacije	5
2.1	Spletna aplikacija	5
2.2	Programski jezik C#	6
2.3	Microsoft .NET Framework	6
2.4	Microsoft SQL Server	11
2.5	Razvojno okolje	13
2.6	Aplikacija za izdelovanje modela (<i>mockup</i>) uporabniškega vmesnika spletne aplikacije	15
2.7	Sistem za upravljanje z verzijami	17
2.8	Sistem za sledenje napakam (<i>bug tracking system</i>)	18
3	Spletna aplikacija od zajema zahtev do implementacije	21
3.1	Zajem zahtev in analiza podatkov	22
3.2	Načrtovanje in implementacija podatkovne baze	27
3.3	Načrtovanje in implementacija spletne aplikacije	31
3.4	Vstopna stran	37
3.5	Administracija (<i>Administration</i>)	39
3.6	Naročila (<i>Orders</i>)	41

KAZALO

3.7	Prevajanje programske opreme (<i>Messaging</i>)	53
4	Sklepne ugotovitve	61

Povzetek

V diplomskem delu je predstavljen sistem, ki zaposlenim v podjetju in njegovim partnerjem med drugim omogoča naročanje licenc za programsko opremo podjetja CGS plus, aktivacijo naročenih licenc, pregled nad naročili in aktivacijami licenc ter prevajanje programske opreme CGS plus. Cel sistem tvorijo spletna aplikacija in spletni servis za samodejne aktivacije licenc na IIS strežniku ter podatkovna baza na podatkovnem strežniku Microsoft SQL Server. Spletna aplikacija in servis temeljita na programskem jeziku C# in ogrodju Microsoft .NET, izdelani pa sta bili v razvojnem okolju Microsoft Visual Studio.

V diplomskem delu so najprej predstavljene tehnologije in okolje, ki so bili uporabljeni za razvoj celotnega sistema: podatkovne baze in spletne aplikacije s servisom.

Drugi del diplomskega dela se osredotoča na postopek izdelave vse od zajema zahtev do dejanske implementacije.

Ključne besede: spletna aplikacija, spletni strežnik, podatkovni strežnik, licence, podpora partnerjem, Microsoft .NET, C#

Abstract

The diploma work presents system, which provides employees of company and their partners functionalities concerning ordering licenses for CGS plus software, their activation and translation of CGS plus software. Entire system consists of web application with web service for automatic activations on IIS web server and database on Microsoft SQL Server. Web application and web service are based on programming language C# and Microsoft .NET Framework and are written in Microsoft Visual Studio development environment.

The diploma work first presents the technology and environment, which have been used for the development of entire system: the database and web application with web service.

The second part of the diploma work focuses on the process of development of entire system and covers everything from requirements to actual implementation.

Keywords: web application, web server, database server, license, support partners, Microsoft .NET, C#

Poglavje 1

Uvod

Elektronsko poslovanje temelji na izmenjavi podatkov med računalniki in je danes že standard, ki ga uporabljajo podjetja širom sveta. Vsa podjetja stremijo k uspešnemu poslovanju, le-to pa danes ni več možno brez ustreznega informacijskega sistema.

Za uspešno poslovanje med podjetji (B2B ali *business to business*) je potrebno zagotoviti kvaliteten sistem, katerega naloga mora biti v prvi vrsti olajšati poslovanje med podjetjem in njegovimi partnerji ter zagotoviti učinkovito upravljanje z informacijami pridobljenimi s strani uporabnikov sistema.

Zaradi hitrega napredka tehnologij se sčasoma pojavijo zahteve po nadgradnji obstoječih aplikacij. Z novejšimi tehnologijami lahko namreč implementiramo več funkcionalnosti, ki jih s starejšimi nismo mogli ali pa je bilo le-te zelo težko implementirati.

V podjetju, ki se med drugim ukvarja tudi z razvojem programske opreme, že od leta 2003 uporabljajo spletno aplikacijo za poslovanje s partnerskimi podjetji (v nadaljevanju partnerji) v Sloveniji in tujini, kateri tržijo programsko opremo podjetja v svojih matičnih državah. Spletna aplikacija omogoča ročne aktivacije licenc programske opreme podjetja in funkcionalnost za prevajanje programske opreme CGS plus v druge jezike. Spletna aplikacija sicer deluje, vendar pa se je z leti izkazalo, da ji manjka kar nekaj funkcionalnosti.

Prva pomanjkljivost obstoječega sistema poslovanja je način naročanja licenc. Licence se namreč še vedno naročajo prek e-poštnih sporočil ali faksa, ki so velikokrat nepregledna in ne povsem jasna. Zaradi tega se postopek izdajanja računa zavleče, saj je potrebno morebitne nejasnosti pred izdelavo računa partnerju odpraviti, kar velikokrat zahteva tudi kontaktiranje in posredovanje partnerja, ki je naročil licenco.

Druga težava takega sistema je nezmožnost aktivacije licence pri končnem uporabniku programske opreme brez posredovanja partnerja.

Tretja težava obstoječe spletne aplikacije je način prevajanja programske opreme. Partner oziroma njegovi zaposleni s potrebnimi pravicami lahko prevajajo programsko opremo podjetja v svoj jezik kar s pomočjo spletne aplikacije, vendar pa je prevajanje zelo oteženo, saj zahteva preveč interakcije z miško. Poleg tega je trenutni seznam nizov, ki jih lahko partner prevede, razdeljen na več strani, od katerih vsaka stran vsebuje do 20 nizov, vseh pa je lahko več tisoč. Uporabnik tako nima pregleda nad vsemi nizi, kar zelo oteži iskanje točno določenega niza. Nekoč implementirani iskalnik namreč ni deloval najbolje in je bil zaradi njegove nezanesljivosti odstranjen.

Obstoječa aplikacija ima možnost kreiranja programskih modulov in paketov. Programski moduli so programska oprema, ki jo izdeluje podjetje. Ustvari jih lahko samo podjetje. Več modulov je večkrat logično združenih v neko celoto ali pakete, ki se nato tržijo. Pakete imajo možnost ustvariti tako podjetje kot partnerji. Obstoječa spletna aplikacija ponuja samo možnost ustvarjanja novih modulov, nima pa možnosti urejanja njihovega naziva oziroma opisa, ki bi pojasnjevali kaj pravzaprav pomenijo. Slednjo pomanjkljivost lahko zasledimo tudi pri paketih. Tako paketov kot modulov pa enkrat, ko so že narejeni, ni možno izbrisati.

Da bi za podjetje izdelali kvalitetno B2B aplikacijo, smo morali dobro preučiti njene funkcionalne in nefunkcionalne zahteve, kar pa ni vedno lahka naloga, saj se lahko zahteve med samim razvojem spreminjajo. Zato je nujno potrebno, da se razvoja lotimo na pravilen način, ki dopušča poznejše spre-

membe in dopolnitve. Iz tega razloga so danes med razvijalci programske opreme najbolj uveljavljeni agilni pristopi k razvoju programske opreme, ki take spremembe dopuščajo in se jim hitro prilagajajo.

Poglavje 2

Pregled tehnologij in orodij za razvoj aplikacije

2.1 Spletna aplikacija

Spletna aplikacija je aplikacija, do katere se dostopa preko internetnega brskalnika. Za uporabo spletne aplikacije potrebuje uporabnik le na računalniku nameščen spletni brskalnik in delujočo internetno povezavo. Prednosti spletnih aplikacij je veliko, med drugim jih ni potrebno nameščati na računalnike uporabnikov, saj se izvajajo na spletnem strežniku, niso odvisne od platforme, ki jo uporabniki uporabljajo, njihovo nadgrajevanje se izvede le na samem strežniku, ki gostuje spletno aplikacijo, posodobitve pa so vidne vsem uporabnikom hkrati.

Spletne aplikacije sestojijo iz osprednjega dela (*front end*) in zalednega dela (*back end*). Slednji ponavadi vključuje tudi podatkovno bazo (*database*).

Osprednji del spletne aplikacije uporabniki vidijo kot uporabniški vmesnik, ki se prikaže v spletnem brskalniku, ko obiščemo spletno stran aplikacije in predstavlja vez med uporabnikom in zalednim delom.

Zaledni del spletne aplikacije je tisti del aplikacije, kjer se nahaja vsa logika aplikacije in je uporabniku spletne aplikacije skrit. Glavna naloga

zalednega dela spletne aplikacije je zbiranje vnosov uporabnikov in njihova interpretacija. Večina funkcionalnosti spletnih aplikacij se običajno nahaja v zalednem delu.

Podatkovna baza je organizirana struktura podatkov, ki jih uporablja aplikacija. Običajno predstavlja pomemben del spletne aplikacije, njen razvoj pa je potrebno pri bolj kompleksnih aplikacijah zelo skrbno načrtovati.

2.2 Programski jezik C#

Visual C# ali krajše C# (*C sharp*) je objektno orientiran programski jezik, ki ga je razvilo podjetje Microsoft za uporabo z ogrodjem .NET.

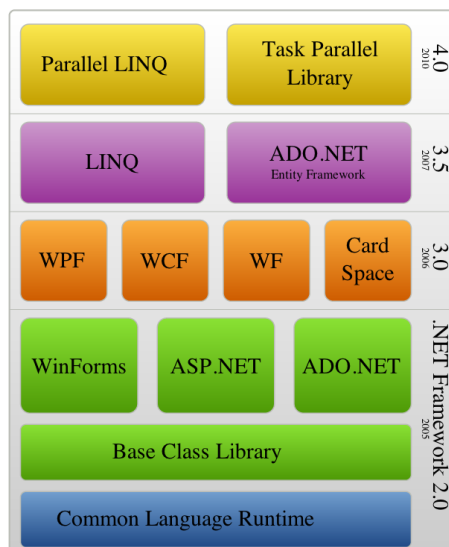
Namen programskega jezika C# je biti enostaven, moderen, vsestranski in objektno orientiran programski jezik. Podobno kot Java vsebuje veliko programsko knjižnico, lovljenje napak, ločevanje vmesnikov od dejanskih implementacij itd. Te lastnosti v kombinaciji z zmogljivimi orodji za razvoj in generiki naredijo C# dober programski jezik za veliko tipov projektov. [9]

2.3 Microsoft .NET Framework

Microsoft .NET Framework je programsko ogrodje razvito s strani podjetja Microsoft za operacijske sisteme Microsoft Windows. Vsebuje veliko osnovno razredno knjižnico (*Base Class Library*), njegova posebnost pa je interoperabilnost programskih jezikov, kar pomeni, da lahko eno aplikacijo napišemo v več programskih jezikih hkrati, kjer vsi uporabljeni programski jeziki med seboj sodelujejo avtohtono (*native*) in uporabljajo enak tip podatkovnih struktur. Programi, napisani za ogrodje .NET Framework, se izvršujejo v programskem okolju imenovanem Common Language Runtime (CLR). CLR je programski navidezni stroj (*application virtual machine*), ki ponuja servise kot so varnost, upravljanje s pomnilnikom in obravnavanje izjem. Osnovna razredna knjižnica in CLR skupaj tvorita ogrodje .NET. [2]

Osnovna razredna knjižnica ogrodja .NET zagotavlja uporabniški vme-

snik, dostopanje do podatkov, povezljivost s podatkovnimi bazami, kriptografijo, razvoj spletnih aplikacij, numerične algoritme in omrežno komunikacijo. Diagram ogrodja prikazuje slika 2.1. [2]



Slika 2.1: Diagram ogrodja Microsoft .NET.

Aktualna različica ogrodja .NET je 4.0. Podjetje Microsoft ponuja integrirano razvojno okolje za razvoj pretežno .NET aplikacij – Microsoft Visual Studio. [2]

2.3.1 ASP.NET

ASP.NET je ogrodje za zasnovo spletnih aplikacij, ki ga razvija in trži podjetje Microsoft. Z njim lahko zgradimo kompleksne dinamične spletne strani, spletne aplikacije in spletne storitve. Ogrodje ASP.NET Framework je bilo skupaj z ogrodjem .NET Framework javnosti prvič predstavljeno januarja 2002 in je naslednik ASP tehnologije. Tako kot ogrodje .NET Framework, temelji tudi ogrodje ASP.NET Framework na programskem okolju CLR, kar omogoča programerjem pisanje ASP.NET programske kode z uporabo .NET programskih jezikov. [1] [5]

Spletne strani ASP.NET, znane kot spletni obrazci (*web forms*), so glavni

gradnik za razvoj spletne aplikacije. Spletni obrazci so vsebovani v datotekah s končnico `.aspx`. Te datoteke tipično vsebujejo tako statičen označevalni jezik (X)HTML, kot označevalni jezik, ki definira strežniške spletne kontrole (*server-side web controls*) in uporabniške kontrole (*user controls*). Dinamično vsebino lahko pišemo v samo datoteko `.aspx` med označbama “<%” in “%>” ali pa uporabimo posebno datoteko *code-behind* s končnico `.cs` za programski jezik C# ali `.vb` za programski jezik Visual Basic. Če uporabimo *code-behind* datoteko, jo moramo navesti na začetku datoteke `.aspx` v parametru Code-Behind (Slika 2.2). Če ustvarjamo spletni obrazec `.aspx` v Visual Studiu, ta namesto nas samodejno ustvari še pripadajočo datoteko `.cs` oziroma `.vb`, tako da nam ni treba tega delati posebej. [1] [5]

```
<%@ Page Title="" Language="C#" MasterPageFile="~/WLM.Master" AutoEventWireup="true"
CodeBehind="Default.aspx.cs" Inherits="WLM.Default" %>
```

Slika 2.2: Primer kako določiti pripadajočo datoteko z izvorno kodo.

Razvoj izgleda spletnih aplikacij je močno olajšan z uporabo predlog. Osnovna stran (*master page*) služi kot predloga, katero uporabljajo spletni obrazci. V predlogi običajno definiramo glavo in nogo spletne strani ter nosilce vsebine (*content placeholders*). Nosilci vsebine so mesta na spletni strani, ki jih lahko ali pa ne uporabijo spletni obrazci za poljubno dinamično vsebino na spletni strani. Tako kot spletni obrazci, lahko tudi osnovna stran vsebuje dinamično vsebino, uporabniške in strežniške kontrole. [1] [5]

ASP.NET AJAX

ASP.NET AJAX je set razširitev za ASP.NET razvit s strani podjetja Microsoft za implementacijo Ajax funkcionalnosti in vsebuje:

- knjižnico Microsoft Now Launched Ajax Library 4.0,
- knjižnico Microsoft Ajax Library 3.5,
- strežniško ogrodje, vključeno v ogrodje .NET,

- paket predlog za Visual Studio za razvoj ASP.NET AJAX aplikacij in
- paket ASP.NET AJAX Preview z novimi funkcionalnostmi, ki bodo vključene v prihodnji različici ogrodja. [6]

Ajax ali asinhroni JavaScript in XML je skupina medsebojno povezanih tehnik za spletni razvoj, ki so uporabljene na strani odjemalca za razvoj asinhronih spletnih aplikacij. Z Ajaxom lahko spletne aplikacije asinhrono izmenjujejo podatke s strežnikom. Kljub temu, da samo ime vsebuje XML, pa lahko odjemalec in strežnik komunicirata tudi v drugi obliki zapisov. Eden bolj pogostih je JSON, ki je bolj učinkovit, saj ne vsebuje toliko označevalnega jezika kot XML, zaradi česar sta prenos podatkov (med strežnikom in odjemalcem) in procesiranje na strežnišku hitrejša. [3]

Ena izmed glavnih prednosti Ajaxa je, da za osvežitev dela spletne strani ne rabimo osvežiti celotne strani, temveč lahko osvežimo le tisti del s spremenjeno vsebino. Ogrodje ASP.NET AJAX omogoča tudi klicanje spletnih storitev z uporabo JavaScripta na strani odjemalca. Za to lahko uporabimo tudi poljubno Javascript knjižnico (npr. jQuery).

Tudi z JavaScriptom lahko pokličemo javne metode v .cs datoteki. Te moramo definirati kot statične spletne metode (*WebMethod*) in skriptne metode (*ScriptMethod*) kot prikazuje slika 2.3.

```
[WebMethod(EnableSession = true)]  
[ScriptMethod(ResponseFormat = ResponseFormat.Json)]  
public static string GetMessageHistory(int moduleId,  
    int languageId, int messageId)
```

Slika 2.3: Primer glave spletne skriptne metode.

Metodo, ki je definirana na sliki 2.3, lahko pokličemo s pomočjo JavaScripta na način, kot je prikazano na sliki 2.4.

```
var moduleId = 10;  
var languageId = 5;  
var messageId = 1001;  
  
PageMethods.GetMessageHistory(moduleId, languageId,  
    messageId, getHistorySuccess, getHistoryError);
```

Slika 2.4: Primer uporabe spletne skriptne metode z JavaScriptom.

2.3.2 Windows Communication Foundation (WCF)

Windows Communication Foundation (WCF) je izvajalno okolje in set aplikacijskih vmesnikov za programiranje (API oziroma *application programming interface*) v ogrodju .NET. Pogosto se uporablja za implementacijo servisno orientiranih arhitektur (SOA) oziroma spletnih servisov. WCF vključuje podporo za najpogosteje uporabljene komunikacijske protokole kot sta SOAP prek HTTP in SOAP prek TCP. Interakcija med WCF končno točko (*endpoint*) in odjemalcem poteka prek SOAP ovojnice. Ta je predstavljena v obliki preprostega XML sporočila zaradi katere je WCF neodvisen od platforme. WCF poleg SOAP sporočil podpira tudi standardni XML, ki ni ovit v SOAP, RSS in JSON. [14]

2.4 Microsoft SQL Server

Microsoft SQL Server je s slabimi dvajsetimi odstotki tržnega deleža eden izmed najbolj uporabljanih sistemov za upravljanje s podatkovnimi bazami (DBMS oziroma *database management system*). [11]

Primarna funkcija podatkovnega strežnika Microsoft SQL Server je shranjevanje podatkov in poizvedovanje na zahtevo s strani drugih aplikacij, tako tistih, ki so nameščene na enakem računalniku, kot tudi tistih, ki do njega dostopajo oddaljeno bodisi prek lokalnega omrežja bodisi prek interneta.

Microsoft SQL Server je na voljo v več različicah:

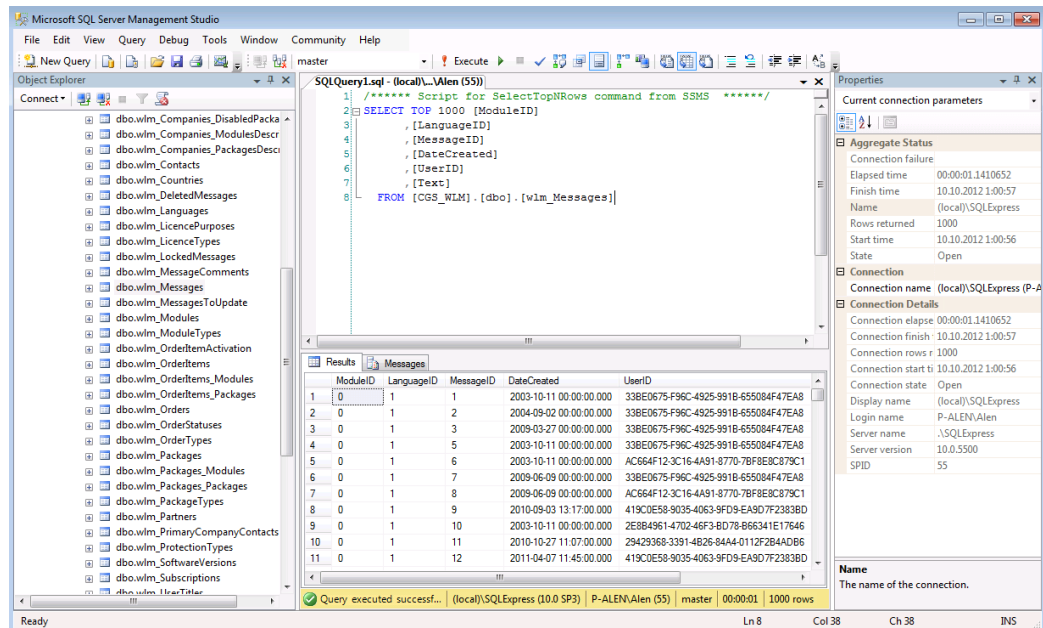
- SQL Server Datacenter,
- SQL Server Enterprise Edition,
- SQL Server Standard Edition,
- SQL Server Web Edition,
- SQL Server Business Intelligence,
- SQL Server Workgroup Edition in
- Microsoft SQL Server Express.

Poleg glavnih naštetih različic je Microsoft SQL Server na voljo še v več specializiranih različicah. [12]

Za našo spletno aplikacijo smo uporabili Express različico, ki je na voljo za brezplačno komercialno rabo.

2.4.1 Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio (slika 2.5) je grafični uporabniški vmesnik za upravljanje s strežniki Microsoft SQL Server in njihovimi podatkovnimi bazami. Na voljo je tudi brezplačni Microsoft SQL Server Management Studio Express za strežnike Microsoft SQL Server Express.



Slika 2.5: Microsoft SQL Server Management Studio.

S pomočjo Microsoft SQL Server Management Studia lahko med drugim ustvarjamo podatkovne baze, jih brišemo, dodajamo tabele, spreminjamo ali brišemo obstoječe in načrtujemo bazo z risanjem diagramov ERD (*entity-relationship diagram*).

2.5 Razvojno okolje

Razvojno okolje (IDE ali *Integrated Development environment*) je aplikacija, ki razvijalcem programske opreme zagotavlja celovito infrastrukturo za razvoj programske opreme. Običajno je sestavljeno iz:

- urejevalnika izvorne kode,
- orodij za avtomatizacijo in
- razhroščevalnika (*debugger*).

Nekatera razvojna okolja vsebujejo prevajalnik ali interpreter (ali oboje).

Namen razvojnega okolja je maksimizacija produktivnosti programerja z uporabo tesno integriranih komponent s podobnim uporabniškim vmesnikom.

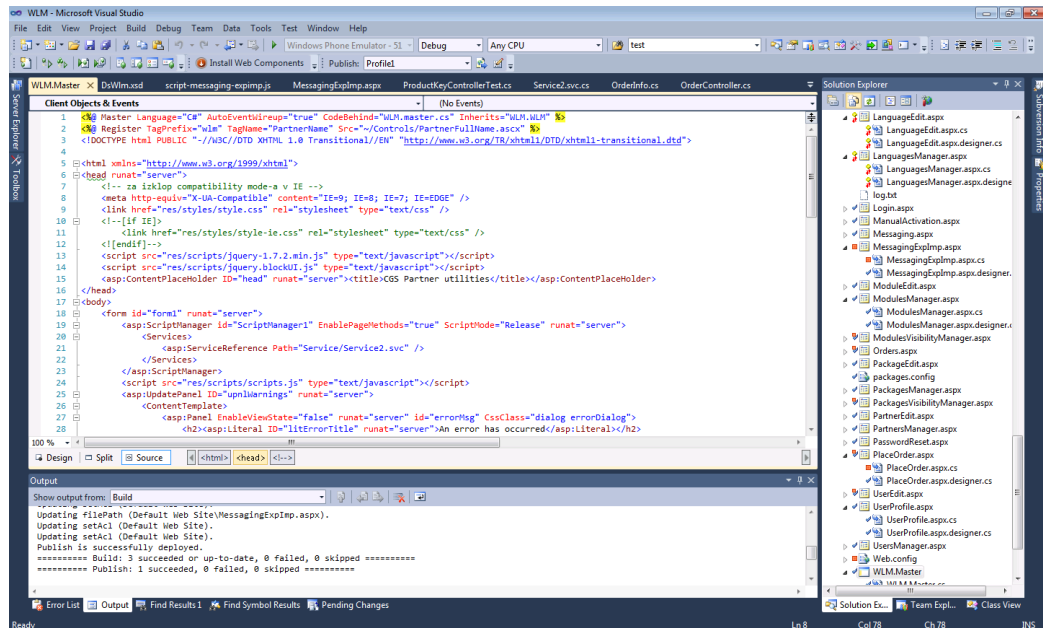
Razvojno okolje tipično predstavlja program v katerem poteka celoten razvoj in ponavadi vključuje funkcionalnosti za avtorstvo, spreminjanje, prevajanje, uvajanje in razhroščevanje programske opreme. Poleg naštetih funkcionalnosti ponuja razvojno okolje še funkcionalnost sprotnega prevajanja kode in nas tako sproti opozarja na sintaktične napake. [10]

2.5.1 Microsoft Visual Studio

Microsoft Visual Studio (slika 2.6) je integrirano razvojno okolje podjetja Microsoft. Uporablja se predvsem za razvoj aplikacij, ki temeljijo na ogrodju .NET. Vključuje urejevalnik izvorne kode, razhroščevalnik, več tipov načrtovalnikov (*designers*) in druga orodja.

Na voljo je v večih različicah:

- Visual Studio Express,
- Visual Studio LightSwitch,
- Visual Studio Professional,



Slika 2.6: Vmesnik razvojnega okolja Microsoft Visual Studio.

- Visual Studio Premium,
- Visual Studio Tools for Office,
- Visual Studio Ultimate in
- Visual Studio Test Professional.

Za razvoj naše spletne aplikacije je zadostovala najbolj aktualna različica Microsoft Visual Studio Professional 2010.

2.6 Aplikacija za izdelovanje modela (*mockup*) uporabniškega vmesnika spletne aplikacije

Razvoja uporabniškega vmesnika spletne aplikacije se ne smemo lotiti zaletavo, še posebej če je spletna aplikacija bolj kompleksna. Če gradimo uporabniški vmesnik “na pamet”, je lahko to za podjetje zelo drago, saj je zelo verjetno, da ta ne bo tako praktičen, kot smo si predstavljali in se bomo morali lotiti izdelave uporabniškega vmesnika od začetka.

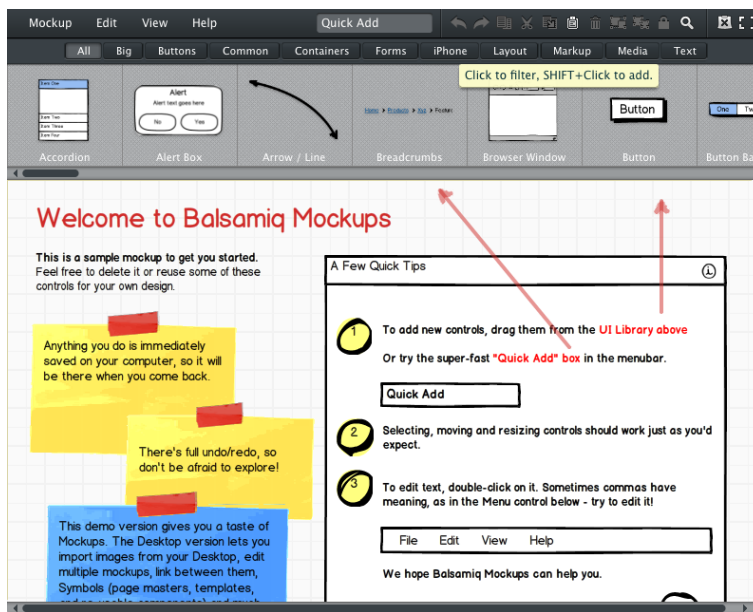
Da bi se izognili tej težavi, je priporočljiva uporaba orodij za izdelovanje modelov uporabniškega vmesnika. Ta vsebujejo najpogosteje uporabljene gradnike v aplikacijah, ki jih dodajamo na delovno površino in tako tvorimo prototip uporabniškega vmesnika za našo aplikacijo. Tako lahko enostavno in hitro vidimo, kako približno bo uporabniški vmesnik izgledal in po potrebi kakšno stvar odstranimo ali dodamo. Hitro lahko naredimo tudi več različnih uporabniških vmesnikov in jih primerjamo med seboj. S takim načrtovanjem uporabniških vmesnikov prihranimo na času in posledično stroških, saj uporabniškega vmesnika ni potrebno dejansko razviti, da bi videli, kako bo na koncu izgledal in ali bo taka postavitev elementov funkcionalna.

2.6.1 Balsamiq Mockups

Balsamiq Mockups je ena izmed aplikacij, ki omogoča hitro izdelovanje modelov uporabniških vmesnikov.

Aplikacija vsebuje enostaven uporabniški vmesnik, sestavljen iz menija, seznama gradnikov in delovne površine (slika 2.7). Gradniki predstavljajo vse od osnovnih elementov kot so povezave, odstavki, vnosna polja, spustni meniji in gumbi, do bolj kompleksnih elementov kot so okno brskalnika, grafi, tabele in dialogi. Med elementi lahko najdemo tudi puščice, oznake za oštevilčenje in virtualne “post-it” listke, katere lahko dodamo na delovno površino in z njimi nek element označimo, dopišemo kakšna naj bo njegova

naloga, kako bo potekala interakcija z njim ipd.



Slika 2.7: Ikone dokumentov kot jih prikazuje vtičnik AnkhSVN.

Narejene modele lahko izvozimo v formatih JPG, PDF, XML in BMML. Aplikacija podpira tudi uvoz že narejenih modelov v formatih XML in BMML.

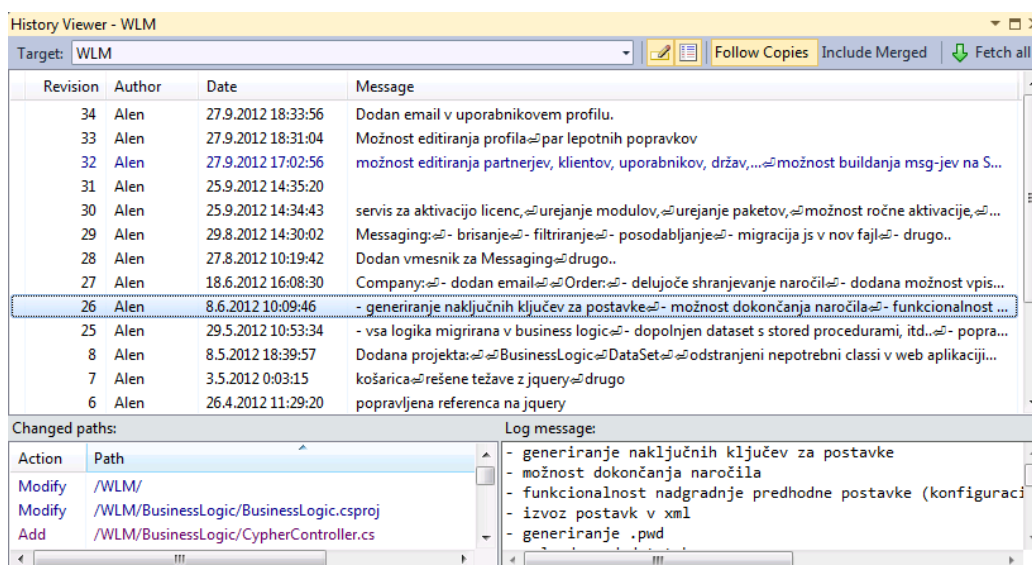
Balsamiq Mockups je na voljo za sisteme Windows, Mac in Linux v brezplačni spletni demonstracijski različici ter brezplačni demonstracijski in plačljivi polno funkcionalni namestljivi različici. [7]

2.7 Sistem za upravljanje z verzijami

Sistem za upravljanje z verzijami SVN oziroma Apache Subversion omogoča razvijalcem ohranjanje različic datotek kot je izvorna koda, spletne strani in dokumentacija. Tako se lahko v katerem koli trenutku vrnemo na katero koli starejšo različico dokumenta. [4]

2.7.1 TortoiseSVN in AnkhSVN

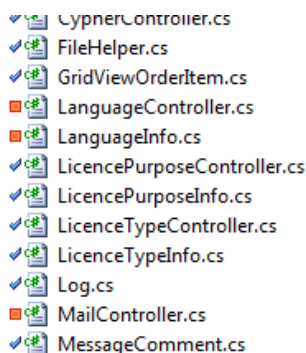
Poznamo več klientov za SVN. Eden boljših brezplačnih je TortoiseSVN za okolje Windows. S pomočjo brezplačnega vtičnika AnkhSVN se odlično integrira v razvojno okolje Microsoft Visual Studio (slika 2.8). [13]



Slika 2.8: AnkhSVN – prikaz zgodovine verzij.

Na sliki 2.9 so prikazane ikone datotek, ki jasno kažejo kateri dokumenti so spremenjeni, kateri niso, pri katerih je prišlo do konfliktov itd.

Tako TortoiseSVN kot AnkhSVN sta na voljo za brezplačno komercialno rabo.



Slika 2.9: Ikone dokumentov kot jih prikazuje vtičnik AnkhSVN.

2.8 Sistem za sledenje napakam (*bug tracking system*)

Sistem za sledenje napakam je aplikacija, katere namen je beleženje napak v programski opremi. Sistemi za sledenje napakam v veliki meri uporabljajo v podjetjih, ki se ukvarjajo z razvojem programske opreme. Uporabniki sistema so ponavadi vodje projektov, razvijalci in testerji. Dosledna uporaba takega sistema velja za odliko dobre ekipe razvijalcev. [8]

2.8.1 Mantis Bug Tracker

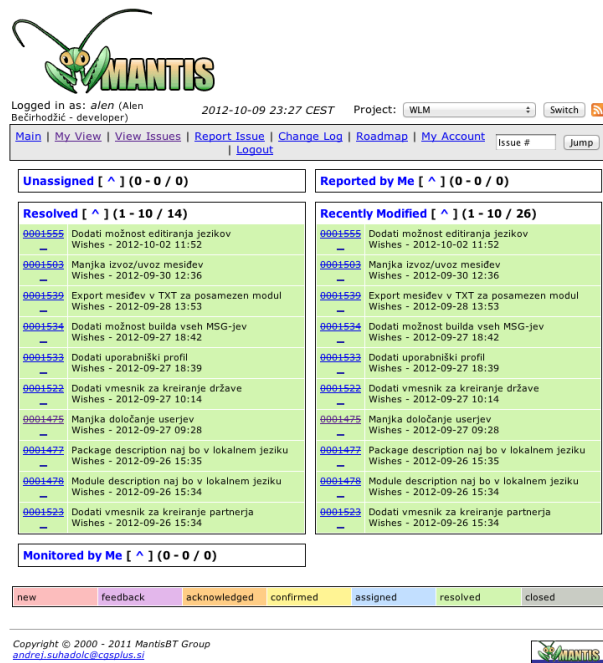
Mantis Bug Tracker (slika 2.10) je brezplačen odprtokoden spletni sistem za sledenje napakam. Pogosto se uporablja tudi kot orodje za upravljanje s projekti.

Sistem Mantis Bug Tracker smo uporabljali med razvojem naše spletne aplikacije kot sistem za sledenje hroščem in sistem za seznam opravil.

Spletno aplikacijo smo uporabljali na sledeč način.

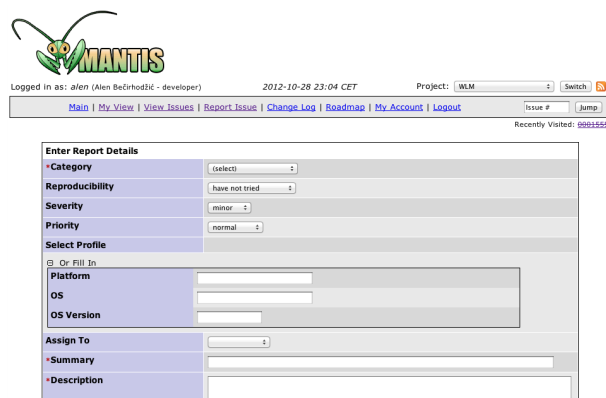
Vodja projekta je v sistem vnesel nov projekt in določil uporabnike, ki so del projekta (razvijalci, testerji itd.). Slednji lahko za ta projekt spremljajo odkrite in odpravljene hrošče ter naloge, ki morajo biti oziroma so že opravljene.

2.8. SISTEM ZA SLEDENJE NAPAKAM (BUG TRACKING SYSTEM)



The screenshot displays the Mantis Bug Tracker interface. At the top, there is a logo for Mantis and a navigation bar with links: Main, My View, View Issues, Report Issue, Change Log, Roadmap, My Account, and Logout. The user is logged in as 'alen' (Alen Bečihodžić - developer) on 2012-10-09 23:27 CEST, working on project 'WLM'. Below the navigation bar, there are four panels showing issue counts: Unassigned (0-0-0), Reported by Me (0-0-0), Resolved (1-10-14), and Recently Modified (1-10-26). Each panel contains a list of issues with their IDs, titles, and timestamps. A status bar at the bottom shows various issue states: new, feedback, acknowledged, confirmed, assigned, resolved, and closed. The footer includes copyright information for MantisBT Group and the Mantis logo.

Slika 2.10: Mantis Bug Tracker.



The screenshot shows the 'Enter Report Details' form in the Mantis Bug Tracker. The form is divided into several sections: Category (dropdown), Reproducibility (dropdown), Severity (dropdown), Priority (dropdown), Select Profile (radio buttons), Platform (text input), OS (text input), OS Version (text input), Assign To (dropdown), Summary (text input), and Description (text input). The form is designed for users to input details about a new bug report.

Slika 2.11: Vmesnik za vnos odkritih hroščev ali nalog, ki jih je potrebno opraviti.

Za vnos hroščev in opravil smo uporabljali enak obrazec (slika 2.11), do katerega smo dostopali s klikom na povezavo “Report issue”. Ker smo tako za hrošče kot za opravila uporabili enak obrazec za vnos, smo ene od drugih ločili tako, da smo jim v času vnosa dodelili eno od v sistem predhodno dodanih kategorij - “hrošči” (*bugs*) in “želje” (*wishes*).

Ko je uporabnik (v našem primeru je bil to ponavadi vodja projekta) v polja vnesel potrebne podatke in potrdil vnos, je bil o tem obveščen uporabnik, kateremu je bil vnos dodeljen oziroma kateri je bil zadolžen za nalogo. Ta je vnos videl na vstopni strani spletne aplikacije pod sekcijo “nerazrešeno” (*unresolved*). Ko je nalogo opravil, je kliknil na vnos in spremenil njegov status. Interni dogovor je bil, da se v primeru odprave napake oziroma opravljene naloge spremeni status vnosa v “odpravljeno” (*resolved*). O vsaki spremembi statusa vnosa je sistem po e-pošti obvestil uporabnika, ki je naredil vnos. Ta je pogledal, če so vse naloge opravljene korektno in ustrezno spremenil status vnosa v “zaključeno” (*closed*), če je bilo vse narejeno korektno oziroma v kakšno od drugih bolj ustreznih možnosti, če naloga ni bila narejena korektno (npr. hrošč ni bil odpravljen). O spremembi statusa je nato sistem po e-pošti obvestil uporabnika, ki je bil zadolžen za nalogo.

Sistem nam je bil v veliko pomoč pri sledenju napredka pri projektu. Poleg tega se je vmes večkrat izkazalo, da je dobro imeti naloge dokumentirane, da med sodelujočimi pri projektu ne prihaja do nesporazumov.

Poglavje 3

Spletna aplikacija od zajema zahtev do implementacije

Pred začetkom izdelave spletne aplikacije se je bilo potrebno podrobno seznaniti s trenutnim načinom poslovanja podjetja s partnerji, dobro preučiti delovanje obstoječe spletne aplikacije in si zastaviti cilje, ki smo jih hoteli doseči. Odločili smo se, da bomo sledili priporočenim smernicam za razvoj programske opreme, ki se deli na naslednje dele:

- zajem zahtev in analiza podatkov,
- načrtovanje aplikacije,
- implementacija aplikacije in
- testiranje in vpeljava.

3.1 Zajem zahtev in analiza podatkov

Na začetku smo se seznanili z obstoječo spletno aplikacijo in njenimi funkcionalnostmi. Definirali smo njene pomanjkljivosti, možnosti za izboljšave in nove funkcionalnosti.

Obstoječa spletna aplikacija je bila narejena leta 2003 in temelji na prvi različici ogrodja Microsoft .NET. Naš namen je bil narediti spletno aplikacijo, ki bo temeljila na najnovejši različici Microsoftovega ogrodja .NET, s katerim lahko poleg novih implementiramo tudi veliko obstoječih funkcionalnosti lažje in predvsem hitreje kot v prejšnjih različicah istega ogrodja.

Namen nove spletne aplikacije je omogočiti podjetju in njegovim partnerjem lažje poslovanje. Med drugim mora vsebovati naročanje novih licenc za programsko opremo, aktivacijo licenc, prevajanje programske opreme podjetja v druge jezike, pregled nad naročenimi licencami in njihovimi aktivacijami. Večino od naštetega v določeni meri podpira že starejša aplikacija, vendar so postopki preveč komplicirani, implementacija pa slaba.

3.1.1 Nefunkcionalne zahteve nove spletne aplikacije

- Dostopnost

Sistem mora biti dostopen od vsepovsod, zato je zopet najbolj smiselno uporabiti spletno tehnologijo.

- Jezik

Uporabniški vmesnik mora biti v angleškem jeziku.

- Možnost nadgradenj

Novo spletno aplikacijo moramo zastaviti modularno, tako da bo dodajanje novih funkcionalnosti možno brez večjih popraviljanj izvirne kode.

- Ogrodje

Spletna aplikacija mora temeljiti na Microsoft .NET ogrodju, napisana pa mora biti v programskem jeziku C#.

- Podatkovna baza

Uporabiti moramo brezplačen podatkovni strežnik Microsoft SQL Server Express, podatkovna baza pa mora vsebovati “entiteta-relacija” diagrame za vse tabele.

- Uporabniški vmesnik

Uporabniški vmesnik moramo zastaviti čim bolj enostavno in pregledno. Tukaj se moramo držati načela več je manj. Tako bomo dosegli boljši izgled in uporabniško izkušnjo v primerjavi s starejšo aplikacijo. Prav tako moramo dati velik poudarek na uporabi tehnologije AJAX in tako omogočili še boljšo uporabniško izkušnjo. Uporabniški vmesnik mora biti čim bolj konsistenten.

3.1.2 Funkcionalne zahteve nove spletne aplikacije

- Naročila in licence

- Omogočati mora naročilo večih licenc.
- Pri naročilu licenc mora sistem generirati naključno serijsko številko.
- Da bo postopek naročila licenc hitrejši, mora aplikacija podpirati generiranje licenc brez posredovanja osebja podjetja.
- Ohraniti moramo podobnost uporabniškega vmesnika z obstoječo spletno aplikacijo.
- Licence morajo biti vezane na podjetje, kateremu pripada uporabnik, ki generira licenco in ne neposredno na uporabnika spletne aplikacije, ki generira licenco.
- Aktivacija licenc mora biti implementirana na naslednja načina:

- * potrebno je narediti spletni servis, ki bo omogočal avtomatično aktivacijo licenc, ko bo uporabnik vpisal serijsko številko v program CGS License Manager.
- * omogočati mora možnost ročne aktivacije.
- Pri aktivaciji licence je potrebno zabeležiti podatke o računalniku, za katerega se licenca aktivira.
- Uporabniki s potrebnimi pravicami morajo imeti možnost pregleda vseh izvedenih naročil licenc in njihovih aktivacij.
- Prevod programske opreme (*messaging sistem*)
 - Aplikacija mora podpirati prevajanje programske opreme podjetja v več jezikov.
 - Uporabniški vmesnik mora nuditi možnost izbire še dveh poljubnih referenčnih jezikov, s katerimi si je možno pomagati ob prevajanju.
 - Dodajanje novih nizov mora biti omogočeno samo administratorjem spletne aplikacije.
 - Obstajati mora možnost zaklepa določenega niza tako, da ga po prvem prevodu partnerji ne morejo več spreminjati.
 - Obstajati mora pregled zgodovine prevodov in možnost povrnitve starejšega prevoda.
 - Administratorji morajo imeti možnost videti prevode enega niza za vse jezike hkrati in možnost popravka več prevodov hkrati za določen niz.
 - Omogočati mora pregled vseh neprevedenih nizov za izbran jezik in vse module hkrati.
 - Implementirati je potrebno iskalnik prevodov, ki mora prevode iskati tako po identifikacijski številki (ID-ju) kot po prevodu, ki ga urejamo in obeh referenčnih prevodih hkrati.

- Izvoz in uvoz prevodov
 - Omogočati mora izvoz prevodov za izbran modul in jezik v datoteko `.txt` zaradi možnosti prevajanja v primeru, da ne moremo dostopati do spletne aplikacije, pri čemer mora datoteka vsebovati vsaj enega ali maksimalno dva referenčna prevoda, ki ju uporabnik izbere. Za datoteke `.txt` mora biti omogočen tudi uvoz, kjer uporabnik izbere jezik, ki naj se posodobi.
 - Omogočati mora izvoz prevodov za izbran modul in jezik v datoteko `.msg`.
 - Omogočati mora generiranje vseh prevodov za vse module naenkrat na določen stežnik v omrežju.
- Uporabniki
 - Vsak uporabnik spletne aplikacije mora pripadati enemu partnerju (je zaposlenec tega partnerja).
 - Vsak uporabnik s potrebnimi pravicami ima možnost urejati stranke svojega podjetja (stranke so večinoma podjetja – možno mora biti urejanje zaposlenih v teh podjetjih).
- Moduli in paketi
 - Aplikacija mora vsebovati vmesnik za pregled modulov in paketov ter njihovo dodajanje in urejanje.
 - Uporabniki s potrebnimi pravicami morajo imeti možnost narediti svoje pakete.
 - Uporabniki s potrebnimi pravicami morajo imeti možnost spremenjati opis lastnih paketov in paketov podjetja. Svojim paketom lahko spremenijo tudi ime. Spremembe opisov paketov podjetja so specifične samo za tistega partnerja, katerega zaposleni je naredil spremembo.

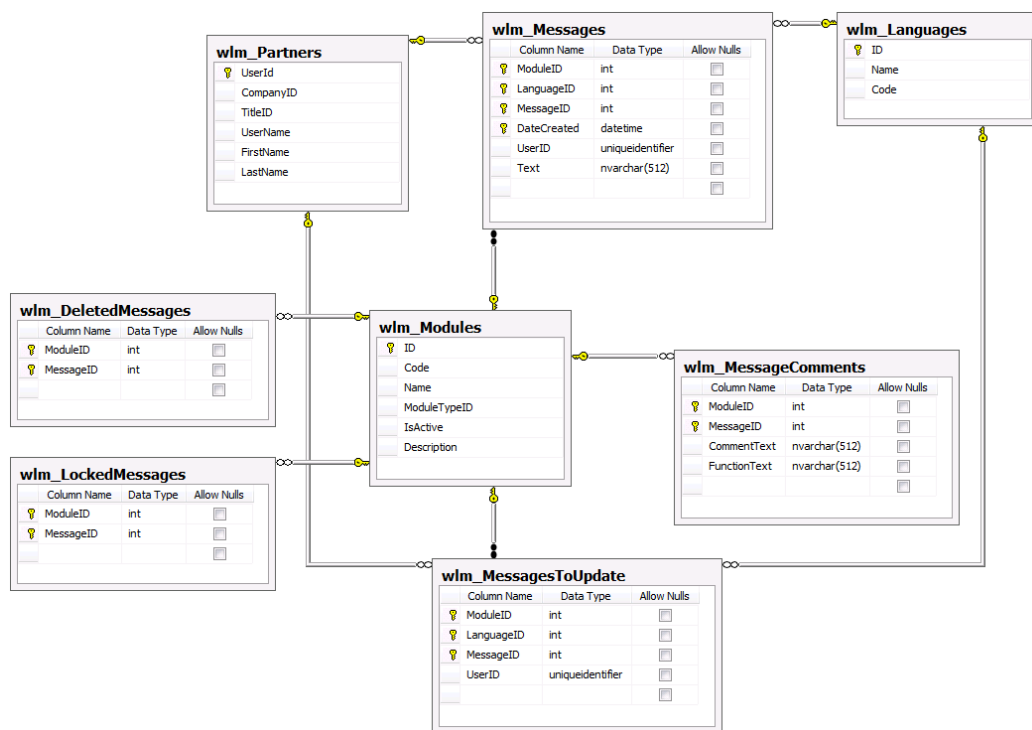
- Uporabniki s potrebnimi pravicami morajo imeti možnost spreminjati opis lastnih modulov in modulov podjetja. Spremembe opisov modulov so specifične samo za tistega partnerja, katerega zaposleni je naredil spremembo.
 - Vsebina paketov se po tem, ko je paket ustvarjen, ne sme spreminjati.
 - Ne smeta obstajati dva paketa z enako vsebino.
 - Uporabniki (razen administratorjev spletne aplikacije) ne smejo imeti možnosti spreminjanja imena modulov in paketov, ki jih je ustvarilo podjetje.
- Prilagodljivost
 - Spletna aplikacija mora vsebovati uporabniške vmesnike za urejanje vsebine (modulov in paketov, uporabnikov, partnerskih podjetij, itd.) brez posredovanja programerja.

3.2 Načrtovanje in implementacija podatkovne baze

Podatkovna baza predstavlja osrednji del celotnega sistema, saj hrani vse podatke, ki so potrebni, da aplikacija deluje. Zaradi tega smo morali biti še posebej pozorni na to, kako jo bomo načrtali, saj lahko vsako poznejše spreminjanje obstoječih entitet povzroči nedelovanje oziroma nepravilno delovanje spletne aplikacije.

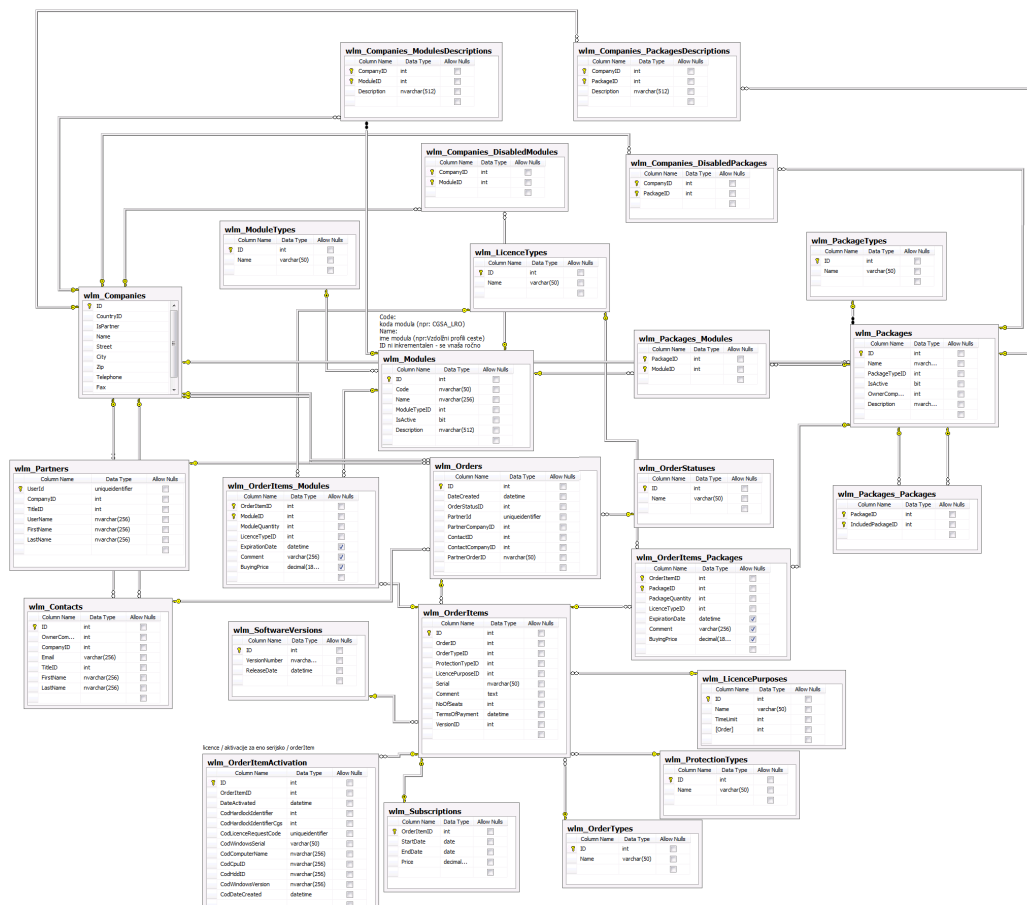
Za načrtovanje podatkovne baze smo uporabili v Microsoft SQL Server Management Studio integrirano orodje za izdelovanje diagramov ER (entiteta-relacija). Tako smo imeli boljši pregled nad relacijami vseh entitet v podatkovni bazi. Ker je bilo entitet veliko, smo te zaradi boljše preglednosti razdelili na več logičnih diagramov:

- messages - vsebuje vse entitete v zvezi s prevajanjem (slika 3.1),
- moduli, paketi, licence – vsebuje vse v zvezi z naročanjem licenc, moduli in paketi (slika 3.2),
- uporabniki – vse v zvezi z uporabniki sistema, podjetji in njihovimi klienti (slika 3.3).

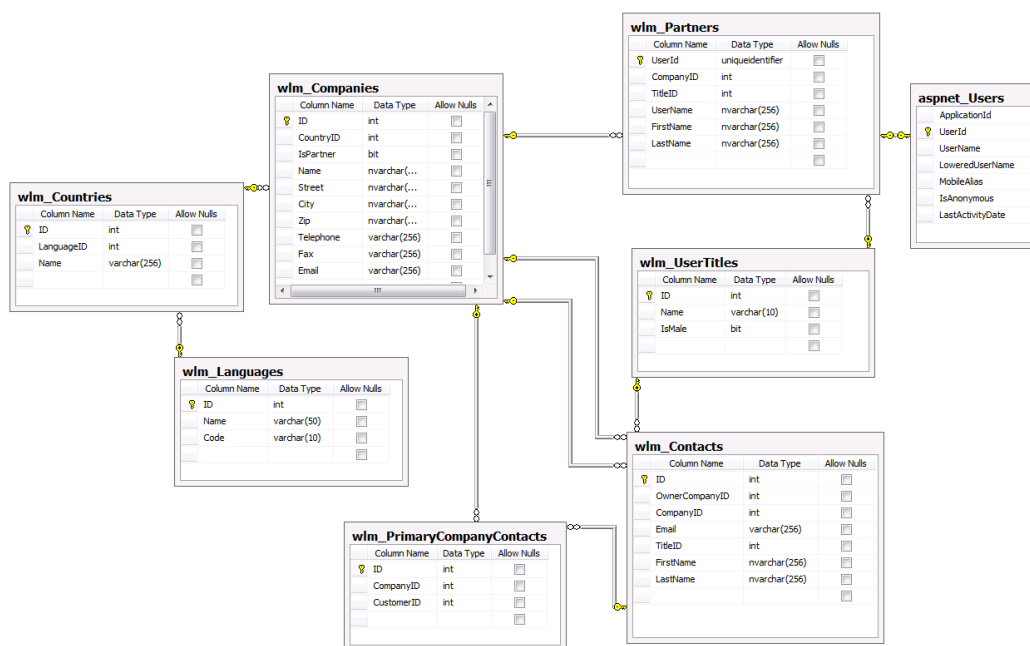


Slika 3.1: Diagram messages.

3.2. NAČRTOVANJE IN IMPLEMENTACIJA PODATKOVNE BAZE 29



Slika 3.2: Diagram moduli, paketi, licence.



Slika 3.3: Diagram uporabniki.

Ker smo se odločili, da uporabimo v ASP.NET že vgrajenega ponudnika za upravljanje z uporabniki spletne aplikacije (*membership provider*), ki uporablja svojo lastno strukturo v podatkovni bazi, smo morali, preden smo začeli z izdelavo diagramov, v Visual Studiu ustvariti nov projekt in zagnati konfiguracijo spletnega mesta (več o tem v poglavju 3). Ko je konfiguracijsko orodje naredilo potrebne tabele in shranjene procedure (*stored procedures*), ki jih ponudnik potrebuje za upravljanje z uporabniki spletne aplikacije, smo se lotili izdelave diagramov.

Potrebno je bilo dobro preučiti, kateri atributi entitet tvorijo primarne in tuje ključe. V primeru, ko ni bil noben od atributov entitete dovolj primeren, da bi služil kot primarni ključ, smo morali uvesti nov atribut, katerega smo praviloma poimenovali kar ID. Imena tujih ključev so bila praviloma sestavljena iz imena entitete, na katerega se tuji ključ navezuje in imena primarnega ključa te entitete. Primer: če smo imeli entiteto z imenom Modules s primarnim ključem ID in entiteto MessageComments s tujim ključem, ki se navezuje na atribut ID v entiteti Modules, smo tuji ključ poimenovali

ModuleID.

Ker je bilo pri zajemu zahtev nemogoče misliti na vse podrobnosti, smo imeli pri načrtovanju podatkovne baze nekaj manjših težav. Zaradi tega smo morali med samo implementacijo spletne aplikacije večkrat sestankovati in določene stvari dodatno doreči, diagrame pa ustrezno popraviti, da so ustrezali dopolnjenim zahtevam.

Ker smo diagrame predhodno shranili, so se samodejno generirale tudi tabele za entitete. Zaradi tega smo imeli pri popravljanju entitet težave pri ponovnem shranjevanju, saj Microsoft Management Studio ne dovoli spreminjanja obstoječih tabel. Da bi se izognili temu problemu, smo morali v nastavitvah to funkcionalnost izklopiti. To smo naredili tako, da smo s klikom na meni "Tools", nato "Options..." in "Designers" odprli nastavitve za urejevalnike. Odkljukali smo možnost "Prevent saving changes that require table re-creation" in nastavitve shranili s klikom na gumb OK.

3.3 Načrtovanje in implementacija spletne aplikacije

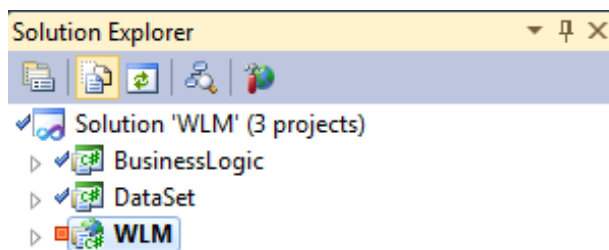
Dobra struktura spletne aplikacije veliko prispeva k napredku implementacije. Izvirne datoteke morajo biti smiselno strukturirane, poimenovane in dobro komentirane.

Dobra praksa je, da zaledni del sistema ločimo od osprednjega dela sistema. Odločili smo se, da aplikacijo razdelimo v tri projekte:

- **BusinessLogic**, ki vsebuje kontrolne razrede s poslovno logiko in razrede za vse entitete v bazi, ki bomo nato uporabili v aplikaciji,
- **DataSet**, ki vsebuje dva DataSeta (predstavljata predpomnjene podatke v podatkovni bazi) – DsAsp za dostop do tabel v bazi, ki jih generira orodje za konfiguracijo spletne aplikacije Web Site Administration Tool in DsWlm za dostop do vseh tabel v bazi, ki smo jih s pomočjo diagramov zgenerirali sami in

- **WLM**, ki predstavlja spletno aplikacijo z vso logiko za uporabniške vmesnike.

V Visual Studiu smo kliknili na meni File, nato New in New Project. Razširili smo drevo Visual C# in izbrali Web. Na desni smo nato izbrali ASP.NET Empty Web Application, ga poimenovali kot WLM in s klikom na OK ustvarili nov projekt za spletno aplikacijo. Temu smo nato po podobnem postopku dodali še dve razredni knjižnici (*class library*), le da smo namesto možnosti Web izbrali Windows in nato na desni Class Library. Eno smo poimenovali BusinessLogic, drugo pa DataSet.

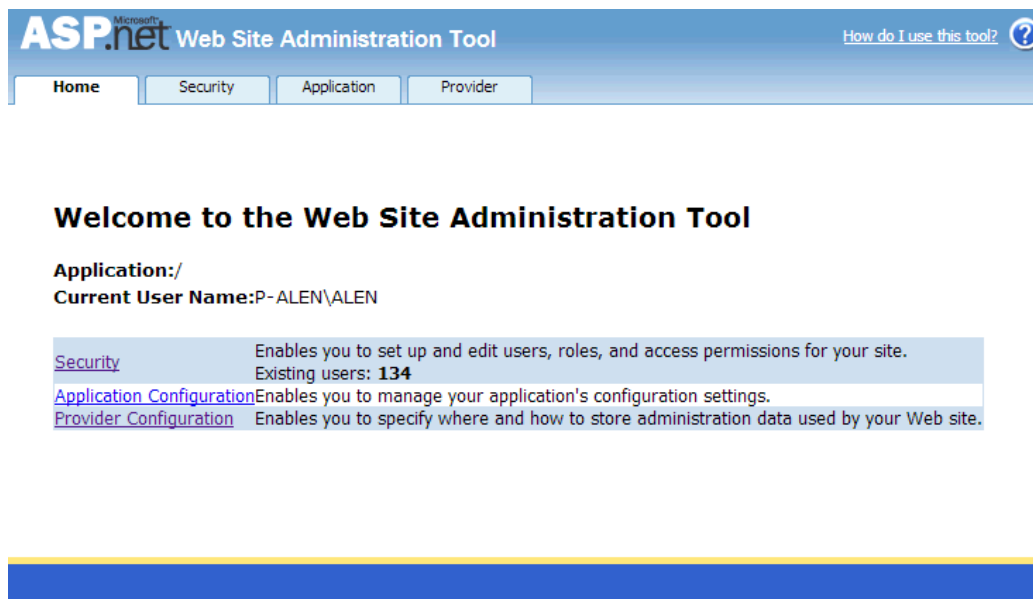


Slika 3.4: Vsebina rešitve WLM.

Kot je razvidno iz strukture na sliki 3.4, predstavljata projekta BusinessLogic in DataSet zaledni del, projekt WLM pa osrednji del spletne aplikacije. Vsi trije projekti so v Visual Studiu združeni eno rešitev (*solution*), ki smo jo shranili pod imenom WLM.

3.3. NAČRTOVANJE IN IMPLEMENTACIJA SPLETNE APLIKACIJE³

Preden smo lahko pričeli z delom, smo morali s konfiguracijskim orodjem Web Site Administration Tool (slika 3.5) v bazi ustvariti potrebne in shranjene procedure za delo z uporabniškimi računi. Orodje smo pognali s klikom na skrajno desno ikono v Solution Explorerju, prikazanem na sliki 3.4.



Slika 3.5: Vstopna stran orodja za konfiguracijo Web Site Administration Tool.

Orodje med drugim ponuja tudi kreiranje uporabniških vlog (*user roles*) in uporabniških računov. S pomočjo uporabniških vlog lahko kategoriziramo uporabnike in nato celotni vlogi dodelimo pravice, namesto da bi to delali za vsakega uporabnika posebej. En uporabnik ima lahko več uporabniških vlog.

Glede na zajete zahteve smo ugotovili, da potrebujemo štiri uporabniške vloge, na podlagi katerih bomo uporabnikom določili njihove pravice. S klikom na zavihek Security in nato na povezavo Create or Manage roles smo ustvarili naslednje vloge:

- Administrator z neomejenimi pravicami,
- Dealer s pravico generiranja novih licenc in ročno aktivacijo licenc,

- Developer s pravico prevajanja, izvoza in uvoza nizov in ročno aktivacijo licenc ter
- Translator s pravico prevajanja obstoječih nizov.

Za začetek smo ustvarili nekaj testnih računov, s katerimi smo lahko preizkušali spletno aplikacijo in funkcionalnosti. To smo storili s klikom na zavihek Security in nato na povezavo Create user. Vsakemu uporabniku smo dodelili eno od prej omenjenih vlog (te uporabniške račune smo pozneje med razvojem izbrisali in jih nadomestili z obstoječimi iz stare aplikacije tako, da smo s pomočjo stavkov SQL pridobili izpis v taki obliki, da smo ga lahko potem kopirali in prilepili v tabelo aspnet_Membership, ki se nahaja v novi bazi).

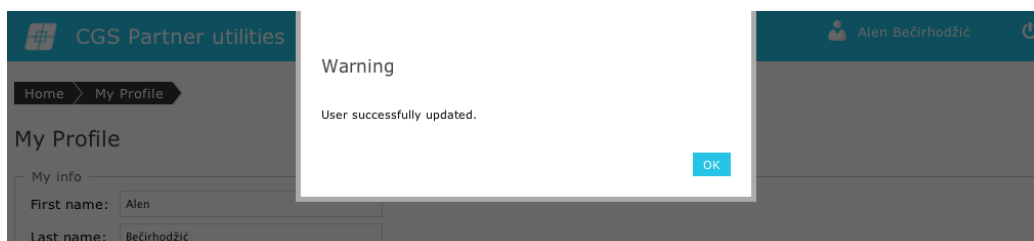
V korenskem direktoriju spletne aplikacije smo ustvarili Master Page z imenom WLM.Master, iz katere smo pozneje izpeljali vse ostale strani spletne aplikacije. WLM.Master deluje kot okvir naše spletne aplikacije, v katerem določimo dele strani, kjer mora biti dinamična vsebina strani in sicer tako, da na tistem mestu dodamo element ContentPlaceholder. V našem primeru smo vstavili en ContentPlaceholder v glavo (*head*) in enega v telo (*body*) spletne strani.

Ko smo dodajali nove strani, smo to storili z desnim klikom na WLM projekt, Add in nato New Item, kjer smo izbrali Web Form using Master Page, jo ustrezno poimenovali in v naslednjem koraku izbrali našo osnovno stran WLM.Master. Visual Studio nam je samodejno zgeneriral vse potrebne datoteke, tako da smo lahko takoj pričeli s programiranjem funkcionalnosti.

V WLM.Master smo v glavo vključili še datoteke `.css`, v katere smo pisali stile in s tem oblikovali vsebino. Prav tako smo v glavo vključili še povezavo do jQuery knjižnice. V glavo spletne strani, ki je vidna na vseh straneh smo na levi vključili logotip podjetja in naziv spletne aplikacije, ki predstavljata povezavo na vstopno stran, na desni pa uporabniško kontrolo, ki izpisuje ime trenutno prijavljenega uporabnika, ki predstavlja povezavo do uporabniškega profila, in gumb za odjavo iz sistema.

3.3. NAČRTOVANJE IN IMPLEMENTACIJA SPLETNE APLIKACIJE⁵

Morali pa smo narediti še uporabnikom prijazen način izpisovanja opozoril in napak. Zato smo naredili še eno datoteko `script.js` s kodo JavaScript in v osnovo stran vključili še element `UpdatePanel` za izpisovanje napak in opozoril in ga privzeto skrili. Vsebino elementov za napake oziroma opozorila smo nato v datoteki `script.js` preverjali vsakič, ko se je sprožila zahteva. Če je pri zahtevi prišlo do napake ali opozorila, smo ustreznemu elementu v `UpdatePanelu` prek javne metode v `WLM.Master` strani določili tekst napake ali opozorila, kar je skripta zaznala in napako ali opozorilo s pomočjo jQuery knjižnice ustrezno prikazala (slika 3.6).



Slika 3.6: Primer opozorila.

3.3.1 Prijava v sistem

Poskrbeti smo morali, da do aplikacije ne smejo dostopati anonimni uporabniki. V datoteko `web.config` smo dodali vsebino na sliki 3.7.

```
<authorization>
  <deny users="?" />
</authorization>
```

Slika 3.7: Primer preprečitve dostopa anonimnim uporabnikom v datoteki `web.config`.

Nato smo izdelali prijavno stran (slika 3.8) in stran za ponastavitev gesla (slika 3.9) po standardnem postopku za izdelavo strani z osnovno stranjo. Za ti dve strani smo morali omogočiti dostop vsem uporabnikom podobno kot prikazuje slika 3.7, le da smo `deny` (prepreči) zamenjali z `allow` (dovoli). Vprašaj pri atributu `users` simbolizira neprijavljene uporabnike.

CGS Partner utilities

Partner login

User Name:

Password:

[Log In](#)

[Lost password?](#)

CGS plus Software 2012 ©

Slika 3.8: Stran za prijavo.

CGS Partner utilities

Forgot Your Password?

Enter your User Name to receive your password.

User Name:

[Submit](#)

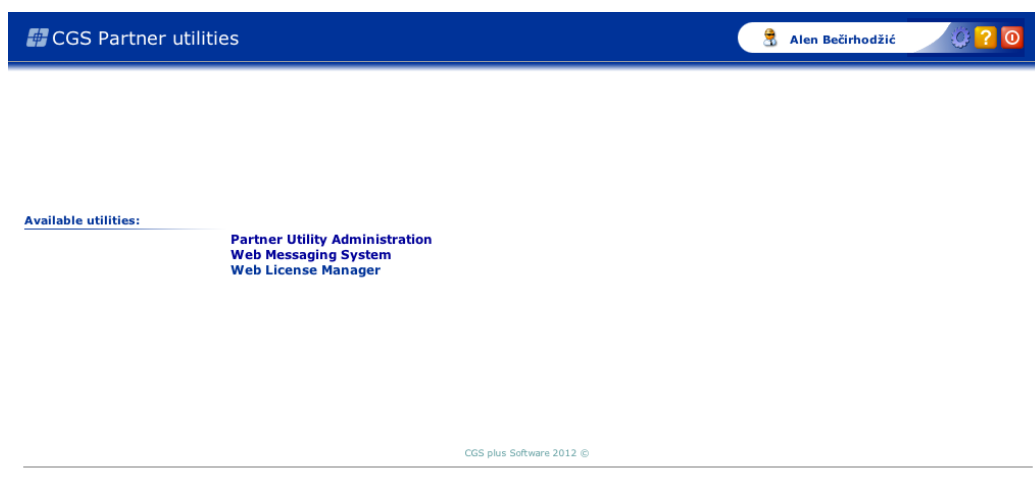
CGS plus Software 2012 ©

Slika 3.9: Stran za ponastavitev gesla.

3.4 Vstopna stran

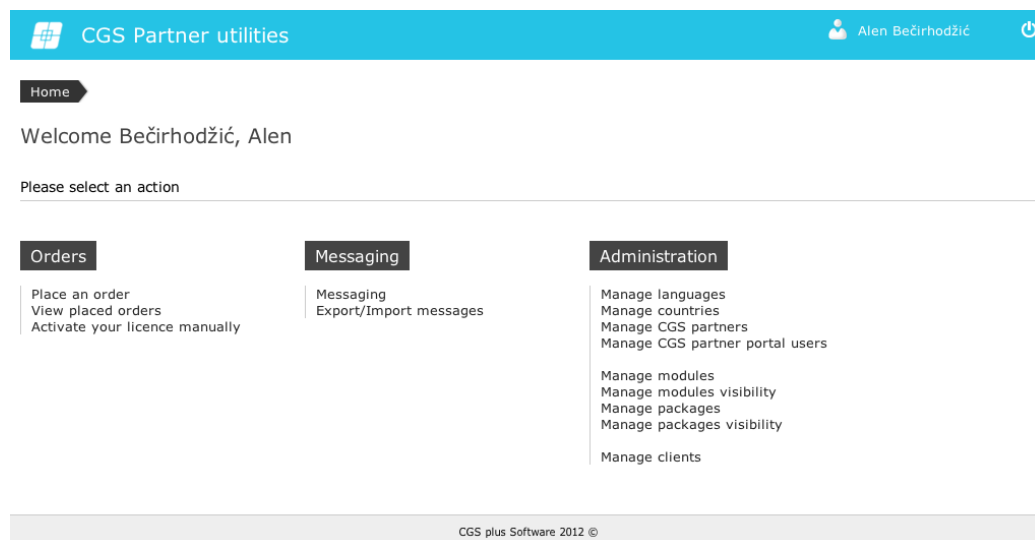
Vstopna stran je prva stran, ki jo uporabnik vidi, ko se uspešno prijavi v spletno aplikacijo in vsebuje povezave do njenih glavnih funkcionalnosti.

Izgled obstoječe spletne aplikacije (slika 3.10) je bil že zastarel in ga je bilo potrebno prenoviti.

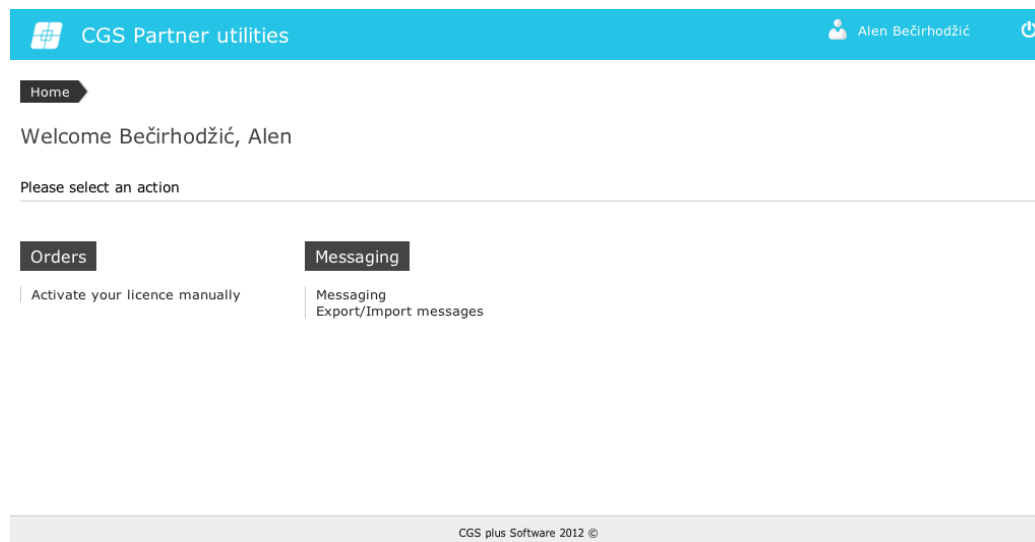


Slika 3.10: Obstoječa vstopna stran.

Funkcionalnosti v novi spletni aplikaciji smo logično razdelili v tri glavne sklope (Orders, Messaging in Administration), od katerih vsak vsebuje pripadajoče module. Prikaz povezav do modulov se prilagaja glede na pravice, ki jih ima prijavljeni uporabnik. Primer: če je prijavljen administrator, so prikazane vse možnosti (slika 3.11), če je prijavljen razvijalec, pa se mu pokažejo samo za razvijalce specifične funkcionalnosti (slika 3.12).



Slika 3.11: Vstopna stran kot jo vidi administrator.



Slika 3.12: Vstopna stran kot jo vidi razvijalec.

3.5 Administracija (*Administration*)

V razdelek za administracijo smo združili vse funkcionalnosti, ki so kakorkoli povezane z nastavitvami in urejanjem modulov, paketov, uporabnikov itd. Čeprav smo se tega dela spletne aplikacije aplikacije lotili najpozneje, ga bom opisal najprej, saj bo pripomogel k boljšemu razumevanju delovanja spletne aplikacije.

Pod administracijo smo vključili naslednje module:

- **Manage languages**

S tem modulom lahko urejamo jezike. Na te se močno navezujejo države, od njih pa je odvisna tudi funkcionalnost nekaterih drugih modulov. Dober primer tega je modul za prevajanje, ki dovoli prevajati nize samo v tistem jeziku, ki je določen državi, kateri pripada partner oziroma njegov zaposleni, ki uporablja spletno aplikacijo. Modul lahko uporabljajo samo administratorji spletne aplikacije.

- **Manage countries**

Modul omogoča urejanje držav. Te določimo partnerjem. Modul lahko uporabljajo samo administratorji spletne aplikacije.

- **Manage CGS partners**

S tem modulom lahko urejamo partnerje podjetja. Modul lahko uporabljajo samo administratorji spletne aplikacije.

- **Manage CGS partner portal users**

Z modulom za upravljanje z uporabniki spletne aplikacije lahko na lahek način dodajamo, brišemo in spreminjamo uporabnike spletne aplikacije. Tukaj jim lahko določimo tudi vloge, na podlagi katerih dobijo pravico do uporabe določenih ali pa vseh funkcionalnosti sistema. Modul lahko uporabljajo samo administratorji spletne aplikacije.

- **Manage modules**

Modul omogoča urejanje programskih modulov podjetja. Te lahko dodajajo in odstranjujejo samo administratorji spletne aplikacije. Urejanje opisov modulov je omogočeno tudi nekaterim zaposlenim pri partnerjih. Ti lahko opise prilagodijo po svoje (na primer jih prevedejo v svoj jezik). Prilagojeni opisi modulov so vidni samo tistim uporabnikom, ki pripadajo enakemu partnerju kot zaposleni, ki je opis uredil.

- **Manage modules visibility**

Z modulom lahko določimo, kateri moduli so vidni katerim partnerjem. Modul lahko uporabljajo samo administratorji spletne aplikacije.

- **Manage packages**

Modul ponuja funkcionalnosti urejanja paketov. Pakete, ki jih ustvari uporabniki, zaposleni pri podjetju, vidijo vsi partnerji. Ti lahko paketom prilagodijo opise, ki so, podobno kot je to pri modulih, vidni samo tistim uporabnikom, ki pripadajo enakemu partnerju kot uporabnik, ki je uredil opis modula.

- **Manage packages visibility**

S tem modulom lahko določimo, kateri paketi so vidni katerim partnerjem. Vidnost paketov se lahko določa samo za pakete, ki jih je naredilo podjetje. Modul lahko uporabljajo samo administratorji spletne aplikacije.

- **Manage clients**

Modul je namenjen upravljanju s strankami partnerjev in ga lahko uporabljajo samo določeni uporabniki. Seznam strank predstavlja seznam podjetij, katerim partnerji tržijo programsko opremo podjetja. Partnerji lahko vnašajo tudi zaposlene za vsako podjetje posebej.

Module za urejanje smo zastavili zelo podobno, posledično so uporabniški vmesniki zelo podobni. Zaradi njihove konsistentnosti ne zmedejo uporabnika, ki se prvič srečuje s spletno aplikacijo.

3.6 Naročila (*Orders*)

Razdelek “Orders” združuje sledeče možnosti:

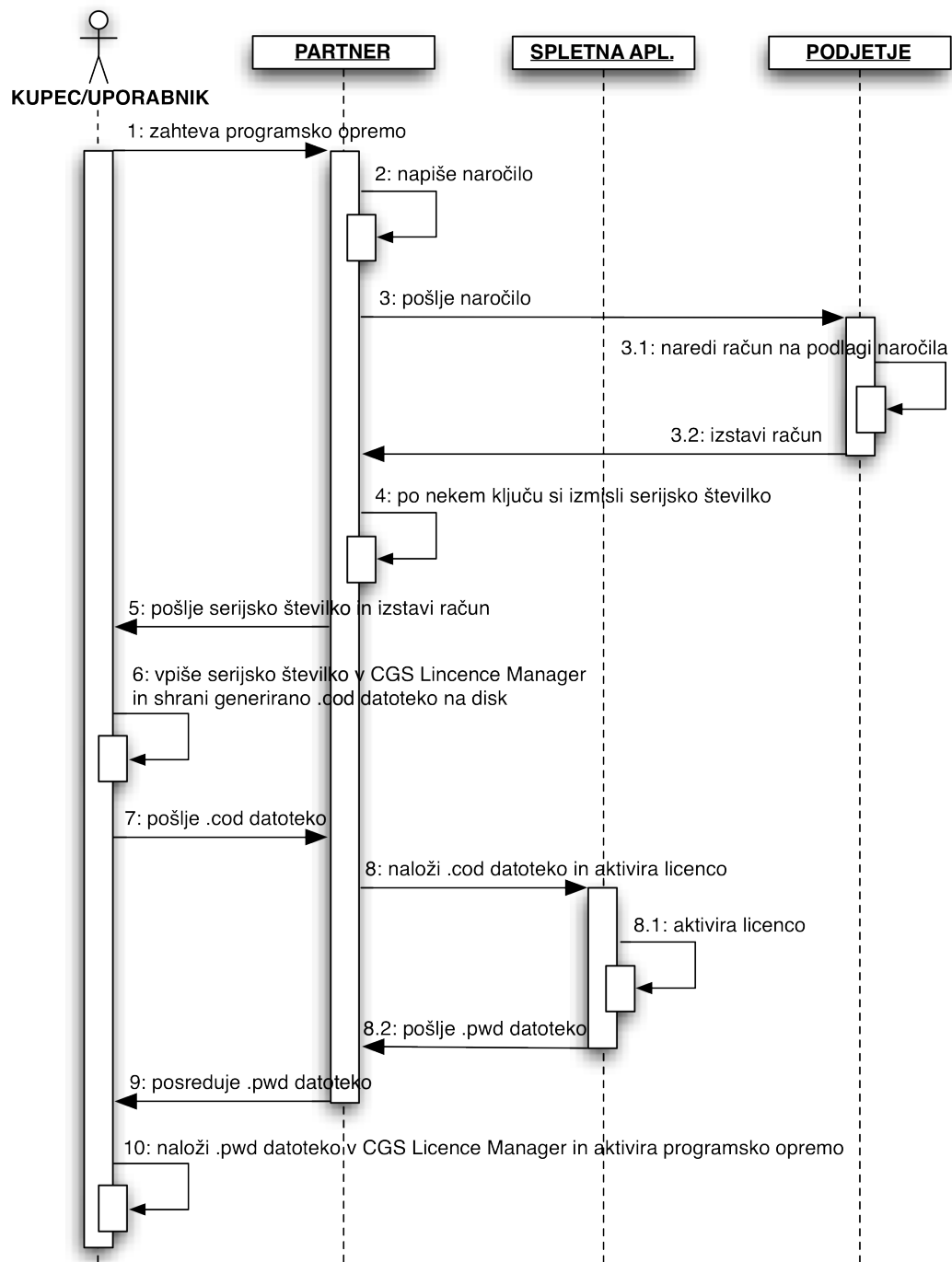
- **Place an order** - naročilo in generiranje licenc,
- **View placed orders** - pregled opravljenih naročil in
- **Activate your licence manually** – aktivacija licenc prek spletne aplikacije.

3.6.1 Postopek naročanja licenc in aktivacija

Za lažje razumevanje postopka prodaje in licenciranja programske opreme podjetja je potrebno razložiti nekaj pojmov.

Podjetje prodaja programsko opremo tako doma kot v tujini. Prodaja v tujini poteka prek zastopnikov (v nadaljevanju partnerjev). Partnerji podjetja oziroma njegovi zaposleni so tudi uporabniki obstoječe spletne aplikacije in bodoči uporabniki nove spletne aplikacije. Njihova naloga je, da tržijo programsko opremo podjetja kupcem, ki so ponavadi podjetja, končni uporabniki programske opreme pa zaposleni v teh podjetjih.

Obstoječi postopek naročanja licenc in aktivacije poteka na način, kot ga prikazuje diagram na sliki 3.13.



Slika 3.13: Obstoječi postopek naročanja in aktivacije licenc.

Slika 3.14 prikazuje obstoječi vmesnik za konfiguracijo in aktivacijo licenc.

License ID: 2012702541
Serial number: 0000-0000-0000-0010
Version: 2013
Hardlock ID: 20244900

Computer name: P-URKE7
CPU Type: BFEBFBFF0001067A
HDD Serial: 20244900

Windows Version: Microsoft Windows NT 6.1.7601 Service Pack 1
Windows Serial: [REDACTED]

Protection Type: Software - Workstation
 Hardlock - Workstation
 Hardlock - Network
 Software - Network

Purpose: Commercial
 Courses and Promotion
 Dealer
 Development
 Educational (Schools,...)
 Lease
 Test (Customers)
 Test (Internal)

Dealer: CGS plus
Company: CGS plus d.o.o.
User: Uroš Barlič

Module or Package Name	License Type	Value	Edit	Delete
CGSA_PK	Permanent		Edit	Delete
CGSA_PI	Permanent		Edit	Delete
CGSA_AK	Permanent		Edit	Delete
CGSA_ARO	Permanent		Edit	Delete
CGSA_LK	Permanent		Edit	Delete
CGSA_LRO	Permanent		Edit	Delete
CGSA_CK	Permanent		Edit	Delete
CGSA_CRO	Permanent		Edit	Delete
CGSA_TR	Permanent		Edit	Delete
CGSA_TC	Permanent		Edit	Delete
CGSA_R	Permanent		Edit	Delete

Remove all modules and packages [Remove]
Add new module: [1] [CGSAPPS (Geo Engineering osnovni modul)] [Add]
Add new package: [1] [PLATEIA + OKSTRA] [Add]
 Add Modules from Package separately
Set license type and value for all modules and packages
 [None] [Set]
Create new package [Create]

Comment: [Text Area] [Generate Password]

CGS plus Software 2012 ©

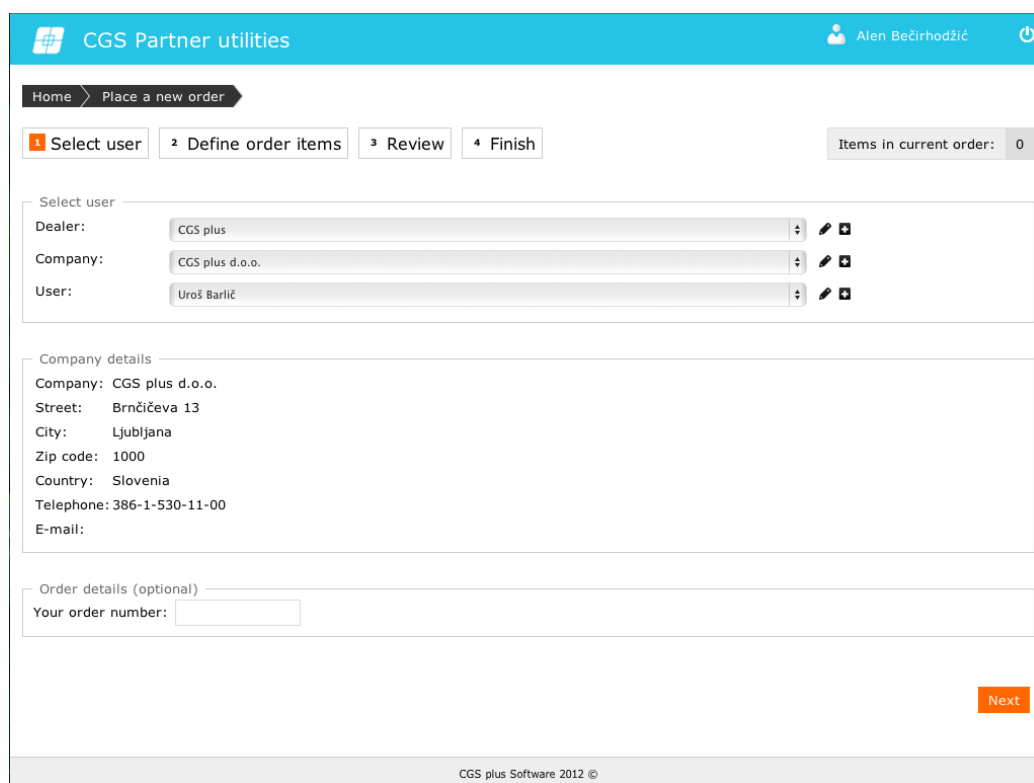
Slika 3.14: Obstoječi vmesnik za konfiguracijo in aktivacijo licenc.

Kot lahko razberemo iz diagrama, gre pri naročanju in aktivaciji licenc za zelo kompleksno proceduro, saj je potrebnih preveč korakov, da pride kupec do licence za naročeno programsko opremo. Naš cilj je bil zmanjšati število korakov tako, da nekatere stvari avtomatiziramo in s tem pohitrimo proces prodaje licenc.

Poleg obstoječih funkcionalnosti kreiranja licenc smo se odločili, da v nov sistem implementiramo tudi funkcionalnost naročanja ene ali več licenc hkrati, samodejno izdelavo in pošiljanje potrditev o naročilih in obvestil o novih naročilih ter možnost aktivacije licenc pri končnem uporabniku brez posredovanja partnerja.

Sistem naročil licenc smo v modulu **Place an order** implementirali kot spletni čarovnik (*wizzard*) s štirimi koraki:

- Select user (slika 3.15) – v tem koraku izberemo podjetje, ki je kupec licence in uporabnika, ki bo licenco uporabljal. Če postopek izvaja uporabnik z administratorskimi pravicami, lahko izbere tudi partnerja oziroma dealerja, ki bo licenco prodal. Partner ima zdaj možnost vpisati tudi svojo številko naročila, če naročilo vodi pod svojo številko.



Slika 3.15: Korak “Select user” v čarovniku.

- Define order items (slika 3.16) – v tem koraku uporabnik nastavi lastnosti in vsebino licence, ki jo hoče generirati. Med drugim določi tudi cene posameznih modulov in paketov, po katerih jih bo prodal kupcu (partnerji imajo pri prilagajanju cen nekaj svobode). Partner lahko ta korak zaključi in se pomakne na naslednjega ali pa se odloči, da bo k naročilu dodal še kakšno licenco preden nadaljuje.

CGS Partner utilities
Alen Bečirhodžić

Home > Place a new order

1 Select user 2 Define order items 3 Review 4 Finish
Items in current order: 0

Select licence details

Version: Purpose:

New/Upgrade: Protection type:

Serial number:

No. of seats:

Add modules and packages

Item	Licence type	Expiration	Buying price	Comment
PLATEIA	Permanent		10000.00 €	
Plateia - road design software				
CGS_EXTENSIONS	By date	02.11.2013	3500.00 €	
FERROVIA	Permanent		7000.00 €	
MSG_SP	<input type="text" value="By date"/>	<input type="text" value="02.11.2013"/>	<input type="text" value="400.00"/>	
SA_LONG	Permanent		350.00 €	
SA_TOPOC	Permanent		300.00 €	

Set licence type and expiration date for modules and packages:

Apply to all

Add new packages:

Add

Add modules from packages separately

[+] Add new module:

Clear list
Create package from list

Comment

Click here to add a comment.

Buying price:

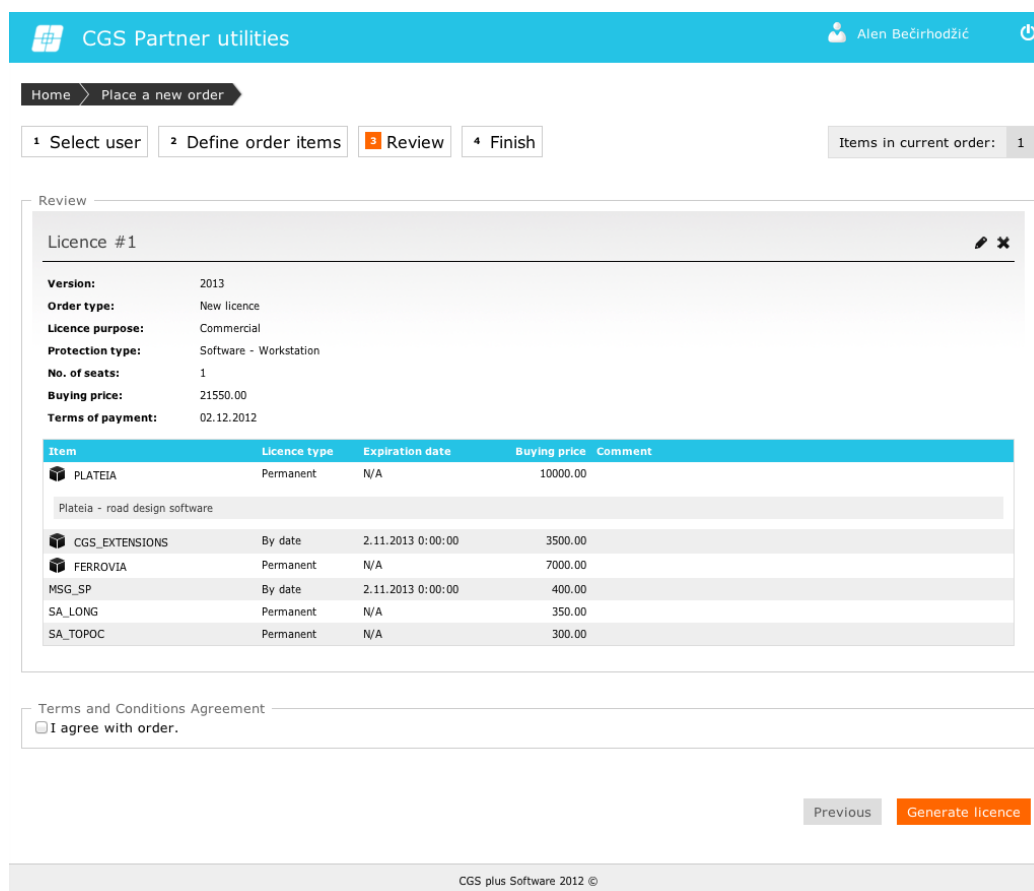
Terms of payment:

Save and add another item

Previous Save and continue to review

Slika 3.16: Korak “Define order items” v čarovniku.

- Review (slika 3.17) – ko uporabnik zaključi z dodajanjem licenc v naročilo, ima v tem koraku pregled nad vsemi licencami, ki so v naročilu in nad njihovimi podrobnostmi. S klikom na Generate licence(s) spletna aplikacija zgenerira licence in za vsako licenco po eno naključno serijsko številko.



Slika 3.17: Korak “Review” v čarovniku.

- Finish (slika 3.18) – v tem koraku se uporabniku izpišejo naročene licence in njihove serijske številke.

Naročilo licenc bo po novem potekalo na način, kot ga prikazuje diagram na sliki 3.19.

CGS Partner utilities Alen Bečirhodžić

Home > Place a new order

1 Select user 2 Define order items 3 Review 4 **Finish** Items in current order: 0

Congratulations!

You have successfully finished your order!

- 1. Serial: ed1d296643d12b5f

Included packages:

- 1x PLATEIA
- 1x FERROVIA
- 1x CGS_EXTENSIONS

Included modules:

- 1x MSG_SP
- 1x SA_LONG
- 1x SA_TOPOC

You can activate your software online or by manually generating .pwd file.

Online activation (preferred)

Online activation is the most simple activation process. All you have to do is to enter your serial number(s) in CGS Licence Manager. The rest will be taken care by CGS Licence Manager itself.

[Show me how to do that](#)

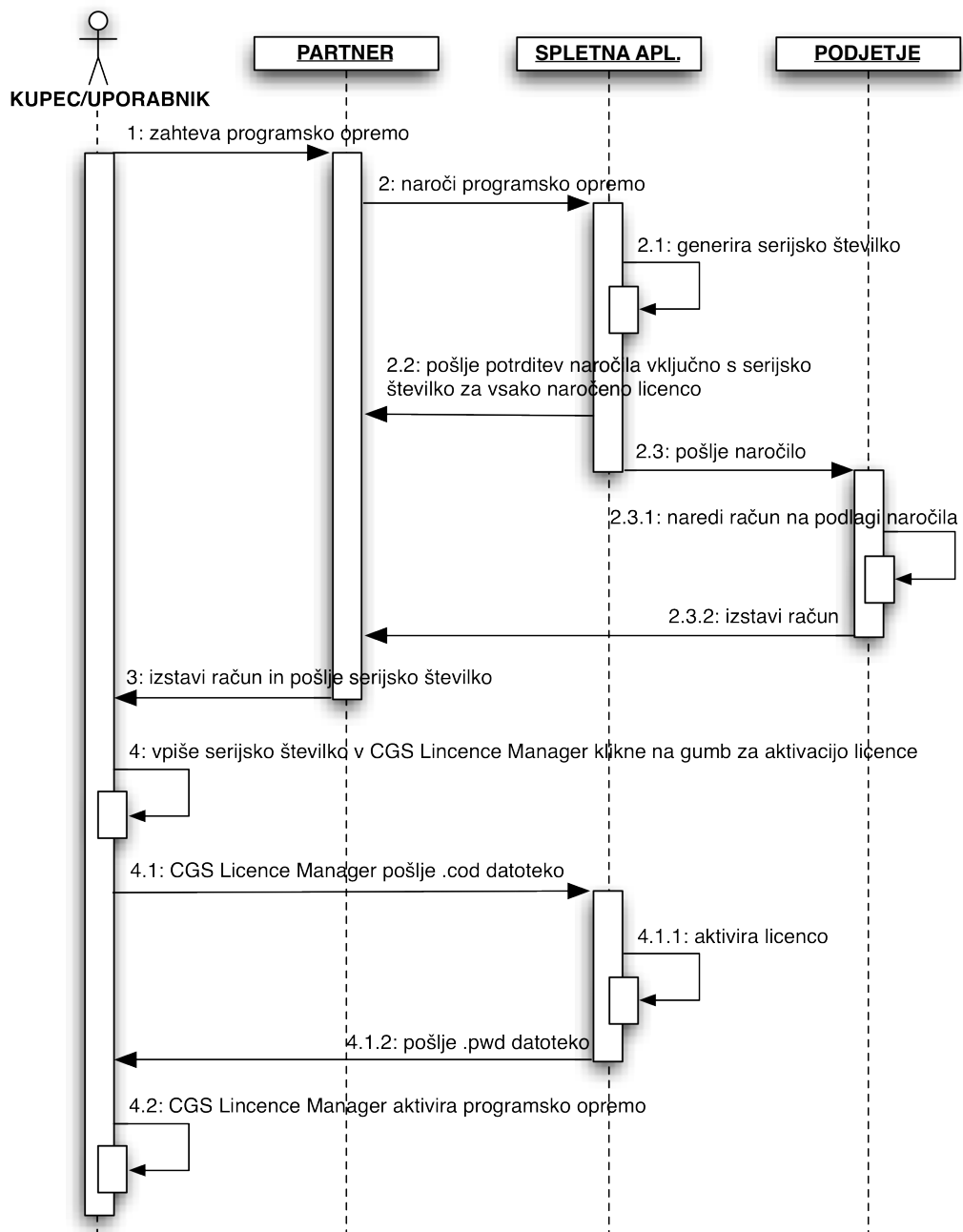
Activation via .pwd file

You can activate your software via .pwd file. First, you have to generate .cod file via CGS Licence Manager and upload it .pwd generator. After that the .pwd file(s) will be generated for you. After you download generated .pwd(s), you must point CGS Licence Manager to that file(s).

[Finish and go to manual activation](#)

[Finish](#)

Slika 3.18: Korak “Finish” v čarovniku.



Slika 3.19: Nov postopek naročanja in aktivacije licenc.

Kljub temu, da je nov način aktivacije licenc veliko hitrejši in boljši, smo se odločili, da bomo funkcionalnost ročne aktivacije licenc s posredovanjem partnerja obdržali, saj obstaja možnost, da uporabnik ne bo imel dostopa do interneta na računalniku, na katerem hoče aktivirati programsko opremo. V tem primeru lahko uporabnik še vedno z dobljeno serijsko številko generira datoteko `.cod`, jo prenese na računalnik, ki ima povezavo na internet in jo pošlje partnerju. Slednji datoteko `.cod` naloži na spletno aplikacijo, ta pa mu, če je aktivacija uspešna, pošlje po e-pošti datoteko `.pwd`, ki jo potem partner posreduje kupcu. Ta naloži datoteko `.pwd` v program CGS Licence Manager in tako aktivira programsko opremo.


Pri aktivaciji licence smo implementirali dodatno zaščito. Licenca se v primeru ročne aktivacije lahko aktivira samo na tistem računalniku, na katerem je bila prvič zgenerirana datoteka `.cod`. V primeru samodejne aktivacije se lahko licenca aktivira samo na tistem računalniku, na katerem se je zgodila prva aktivacija licence. Aktivacija na drugem računalniku po tem ni več mogoča.

Kot je razvidno iz diagrama na sliki 3.19, smo postopek nakupa licenc z novim sistemom občutno skrajšali. Nov sistem prodaje in aktivacije licenc v primerjavi s starim prinaša številne prednosti. Glavne so našteje v tabeli 3.1.

Nov sistem	Star sistem
Možnost naročanja večih licenc hkrati. Sistem o naročilu samodejno obvesti podjetje.	Nima možnosti naročanja licenc. Partner mora podjetju posebej poslati naročilo po faksu ali e-pošti.
Podjetje dobi naročilo v pregledni in konsistentni obliki.	Partner mora čakati na podjetje za izstavljen račun.
Serijske številke so sestavljene iz črk in števil in so naključno generirane, zaradi česar jih praktično ni mogoče uganiti.	Serijske številke določa partner po nekem svojem ključu. Ta je predvidljiv in zato obstaja nevarnost, da ga nekdo ugane. Serijske številke sestavljajo samo števila, vsaka naslednja pa je ponavadi inkrement prejšnje. Zaradi tega so serijske številke veliko bolj predvidljive in obstaja velika možnost zlorabe.
Možnost samodejne aktivacije. Program CGS Licence Manager s pomočjo spletnega servisa samodejno aktivira licenco, ki pripada vpisani serijski številki. Licenco se lahko prav tako aktivira ročno s posredovanjem partnerja.	Možna je samo ročna aktivacija s posredovanjem partnerja.
Naročilo licence in aktivacija pri končnem uporabniku se lahko izvede v manj kot pol ure.	Od naročila licence do aktivacije pri končnem uporabniku lahko mine od nekaj ur do nekaj dni (odvisno od odzivnosti partnerja in kupca).

Tabela 3.1: Prednosti novega sistema pred starim

Slika 3.20 prikazuje primer html dokumenta, ki ga spletna aplikacija pošlje partnerju po e-pošti, ko zaključi z naročilom.



CGS plus d.o.o.
 Brnčičeva 13
 SI-1000 LJUBLJANA
 Slovenia

Phone: +386 1 530 11 00
 Fax: +386 1 530 11 32
 e-mail: info@cgplus.si

Dear partner,

Here is a summary of your recent order.

Order confirmation

Reseller
 Alen Bečirhodžić
 CGS plus
 Brnčičeva 13
 1000 Ljubljana
 Slovenia

Buyer
 Uroš Barlič
 CGS plus d.o.o.
 Brnčičeva 13
 1000 Ljubljana
 Slovenia

Serial number ed1d-2966-43d1-2b5f
Version 2013
Order type New licence
Licence purpose Commercial
Protection type Software - Workstation
Buying price 21550,00
Terms of payment 02.12.2012

Comment:

Ordered items

Item name	Licence type	Exp. date	Buying price	Comment
PLATEIA	Permanent	01.01.0001	10000,00	
Plateia - road design software				
FERROVIA	Permanent	01.01.0001	7000,00	
CGS_EXTENSIONS	By date	02.11.2013	3500,00	
MSG_SP	By date	02.11.2013	400,00	
SA_LONG	Permanent	01.01.0001	350,00	
SA_TOPOC	Permanent	01.01.0001	300,00	

Slika 3.20: Primer potrdila o naročilu.

Modul **View placed orders** predstavlja uporabniški vmesnik, s pomočjo katerega lahko za izbranega partnerja pogledamo naročene licence in njihove morebitne aktivacije. Modul je še v fazi implementacije.

Modul **Activate your licence manually** (slika 3.21) ponuja uporabnikom spletne aplikacije ročno aktivacijo licenc. Vmesnik je preprost in enostaven za uporabo. Da bi licenco aktivirali, moramo na spletno aplikacijo preko obrazca naložiti datoteko `.cod`. Če je računalnik, na katerem je bila datoteka generirana, upravičen do aktivacije licence, jo spletna aplikacija aktivira in prijavljenemu uporabniku preko e-pošte pošlje datoteko `.pwd` za aktivacijo z

nadaljnimi navodili. Datoteko ponavadi partnerji posredujejo strankam.

CGS Partner utilities Alen Bečirhodžić

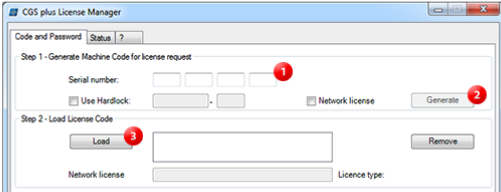
Home > Manual license activation

Manual activation

Activate your licence manually

Here you can activate your licence by manually generating .pwd file and loading it into the CGS License Manager. You can do that by following the steps below.

1. In **CGS Licence Manager** enter the serial number of the licence the system has generated for you when your order was placed.
2. Press **Generate** and save **.cod** on your hard drive.
 - a. Upload **.cod** file to the form below and press **Activate licence** button. If everything went OK, you should get email with generated **.pwd** file.
 No file chosen
 - b. Save **.pwd** file you received by email on your hard drive.
3. Load **.pwd** file from your hard drive to **CGS Licence Manager**.



Slika 3.21: Modul za ročno aktivacijo licence.

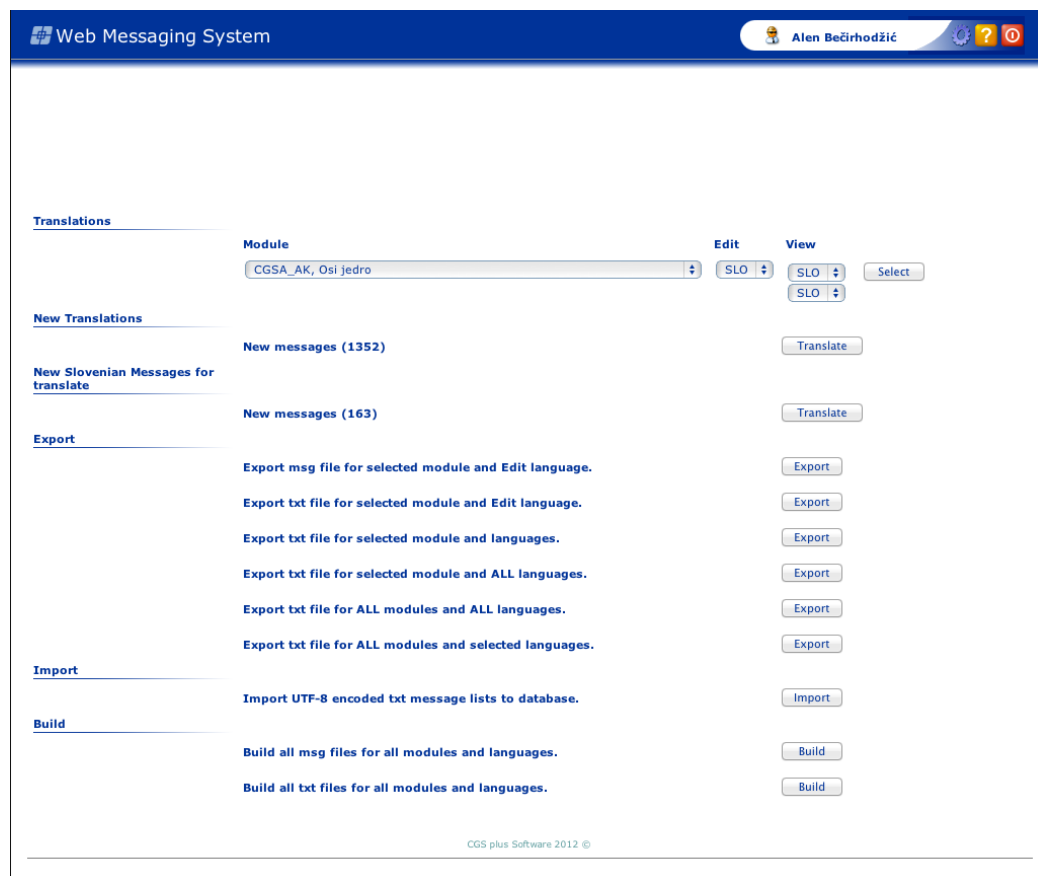
3.7 Prevajanje programske opreme (*Messaging*)

Modul **Messaging** je orodje za prevajanje nizov v programskih modulih podjetja v druge jezike. Pri prevajanju imamo možnost izbire dveh referenčnih jezikov, po katerih se lahko med prevajanjem zgledujemo. Prevedeni nizi so shranjeni v podatkovni bazi spletne aplikacije. Spletna aplikacija ima možnost izvoza nizov v primernem formatu za nadaljno uporabo v modulih ali za prevajanje v načinu brez povezave.

Vsak modul ima svojo enolično identifikacijsko številko, ki smo jo poimenovali `ModuleID`, prav tako ima enolično identifikacijsko številko vsak niz v modulu, katero smo poimenovali `MessageID`. Nizi so kot identifikacijske številke statično vpisani v izvorni kodi modulov. Identifikacijske številke nizov se nato za vsak modul in jezik v času prevajanja izvorne kode modulov v izvršljive datoteke pretvorijo v nize.

Sistem za upravljanje s prevodi v obstoječi aplikaciji (slika 3.22) in modul za prevajanje (slika 3.23) sta preveč nepregledna in zapletena za uporabo. Preden vstopimo v prevajalnik, moramo izbrati jezik, katerega bomo prevajali in še dva referenčna jezika, s katerimi si lahko med prevajanjem pomagamo.

Za vnos prevod enega niza je potrebnih več klikov, da bi se pomaknili do naslednjega, pa moramo vsakič znova še enkrat klikniti. Hitro prevajanje z obstoječim uporabniškim vmesnikom je tako zaradi preveč klikanja nemogoče. V primeru, da pri prevajanju naredimo napako, nimamo nobene zgodovine prevodov in prevoda zato ne moremo povrniti v predhodno stanje. Ustvarjanje novih nizov je prav tako zamudno, saj moramo zapustiti uporabniški vmesnik, ustvariti nov niz in se nato spet vrniti nazaj na prejšnji uporabniški vmesnik, če hočemo nadaljevati s prevajanjem. Ker se večkrat dodaja več nizov hkrati in se njihove identifikacijske številke povečujejo z inkrementom 1, moramo pri dodajanju novih nizov vsakič znova na roko povečati število. Iskalnika ni, zato je iskanje obstoječih prevodov zamudno.



Slika 3.22: Vstopna stran sistema za upravljanje s prevodi.

Web Messaging System Alen Bežirhodžić

Module: CGSA_AK, Osi jedro Modules View mode

Edit mode - Message Selector:

Module-Msg	F	ENG	DEU	SLO
20-1		2013	2013	2013
20-2		AXES Kernel	ACHSEN Basis	Osi jedro
20-100		MODULE	MODUL:	MODUL:
20-101		AXES-CORE	ACHSEN Basis	OSI-jedro
20-102		AXES-ROADS	ACHSEN Straßen	OSI-cesta
20-103		AXES-RAILROAD	ACHSEN Eisenbahn	OSI-železnice
20-104		AXES-WATER	ACHSEN Flussregulierung	OSI-vodarstvo
20-105		No login	Kein Login	Ni prijave
20-369		Highway	Autobahn	
20-370		1st Road Category	Strassenkategorie 1	
20-371		2nd Road Category	Strassenkategorie 2	
20-372		3rd Road Category	Strassenkategorie 3	
20-373		4th Road Category	Strassenkategorie 4	
20-374		5th Road Category	Strassenkategorie 5	
20-375		Fiat	Fiat	

Pages: << First < Prev Next >> Last >>> 1 2 3 4 5 6 7 8 9 10 ...

20-1 Add new Copy Delete

Language: Message text:

SLO	2013	Update
ENG	2013	Update
DEU	2013	Update
HRV	2013	Update
POL	2013	Update
SCG	2013	Update
CZ	2013	Update
RUS	2013	Update
HUN	2013	Update
RO	2013	Update
MK	2013	Update
BG	2013	Update
LIT	2013	Update
X		Update
AUT		Update
SWE		Update
NOR		Update
TUR	2013	Update
US		Update

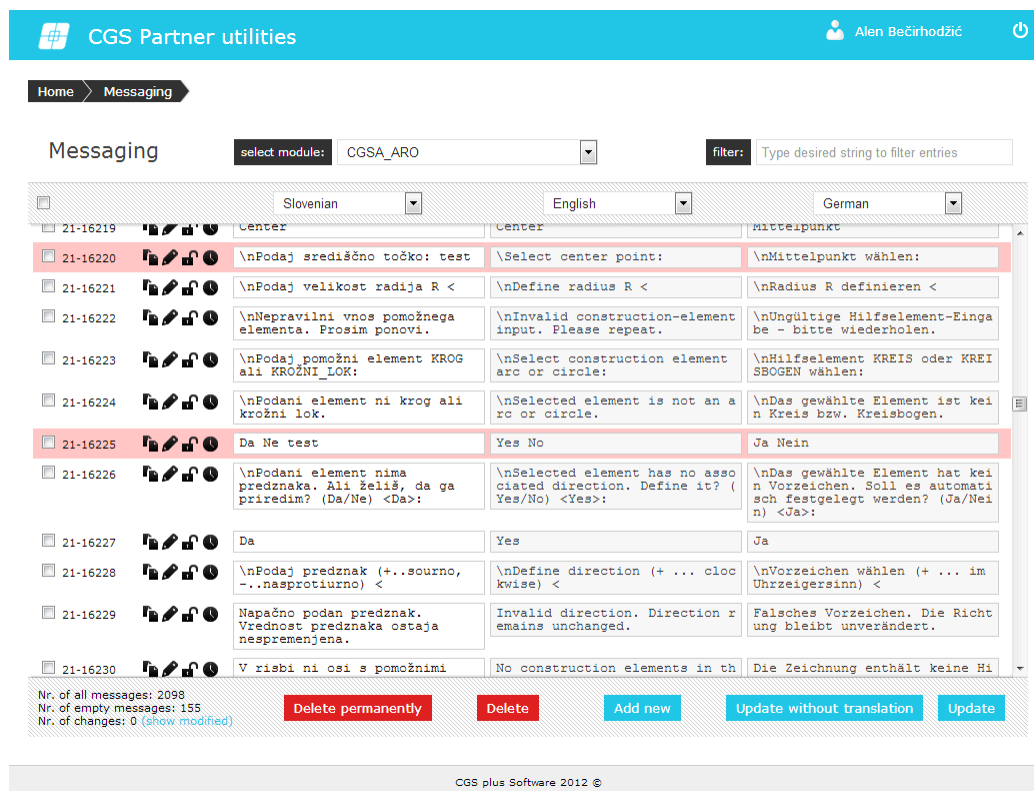
Function: Module Version

Comment:

Update All

Slika 3.23: Modul za prevajanje.

Vse naštete probleme smo z novim modulom za prevajanje (slika 3.24) odpravili. Do modula lahko dostopamo neposredno iz vstopne strani s klikom na povezavo Messaging v razdelku Messaging. Naredili smo uporabniški vmesnik, ki omogoča hitro preklapljanje med prevodi modulov, menjavanje jezikov, pregled samo neprevedenih nizov vseh modulov, itd.

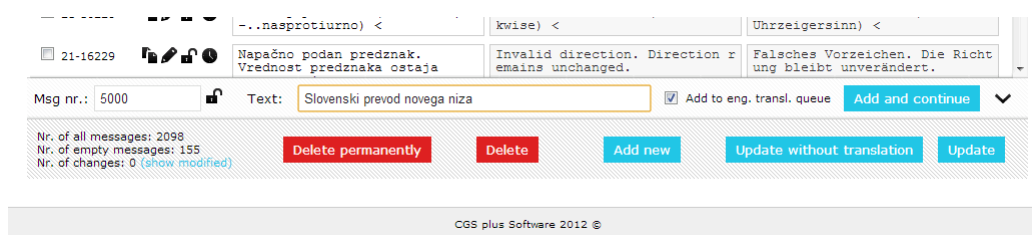


Slika 3.24: Modul za prevajanje v novi spletni aplikaciji kot ga vidi administrator.

Ko kakšen prevod popravimo, se cela vrstica prevoda obarva z drugo barvo, da lahko spremenjene nize ločimo od nespremenjenih. Vrstice s spremenjenimi prevodi so obarvane toliko časa, dokler sprememb ne uveljavimo.

Dodajanje novih nizov smo poenostavili (slika 3.25). S klikom na gumb Add new se prikaže vrstica za dodajanje, kjer vpišemo identifikacijsko številko niza. Takoj lahko tudi odklenemo ali zaklenemo niz in s tem preprečimo ali omogočimo spreminjanje prevodov enkrat, ko so že vnešeni. V tekstovno

polje nato vpišemo prevod niza in izberemo ali želimo, da se niz pojavi kot nepreveden v angleškem jeziku. Na koncu zaključimo s klikom na gumb Add and continue. Na seznam se doda nov niz, ID novega niza za dodajanje se poveča za ena, tekstovno polje se izprazni in fokus se postavi nanj za vnos novega niza. Od te točke naprej lahko dodajamo nove nize tako, da vpišemo tekst prevoda v tekstovno polje, pritisnemo na tipko enter in vse skupaj ponovimo za vsak niz, ki ga hočemo dodati.



Slika 3.25: S klikom na gumb Add new se pojavi vrstica za dodajanje novega niza.

Vse zahteve se izvajajo asinhrono s pomočjo Ajaxa, zato med njimi ne prihaja vedno znova do osveževanja strani. Osveži se le spremenjena vsebina. To smo dosegli s klici spletnih skriptnih metod.

Skupaj z modulom za prevajanje programske opreme smo implementirali tudi modul **Export/Import messages** za uvoz in izvoz prevodov.

Modul podpira naslednje funkcionalnosti:

- Izvoz v MSG format

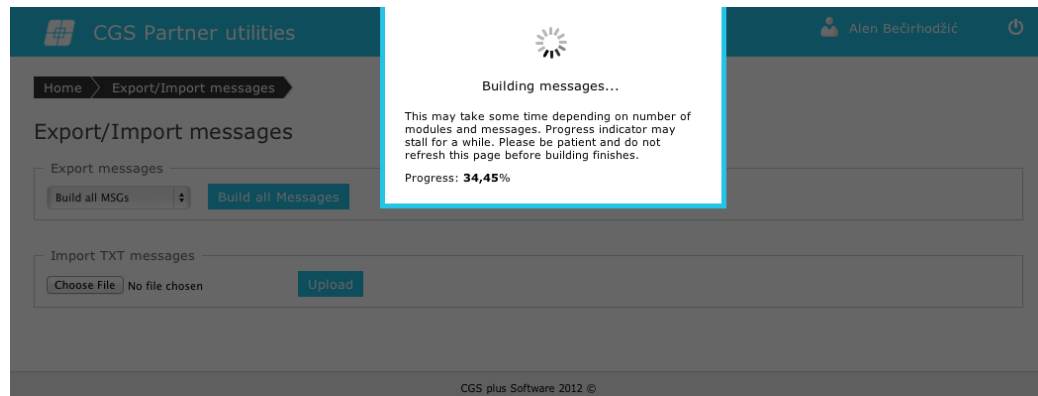
Datoteka s končnico `.msg` vsebuje prevod enega modula v enem jeziku. Te datoteke so namenjene razvijalcem programskih modulov podjetja.

- Izvoz v TXT format

Izvoz v datoteko `.txt` je koristen za tiste, ki za prevajanje ne uporabljajo spletne aplikacije. Izvoženo datoteko lahko uvozimo tudi v Microsoft Excel ali drug podoben program.

- Izdelava MSG datotek za vse module in vse jezike

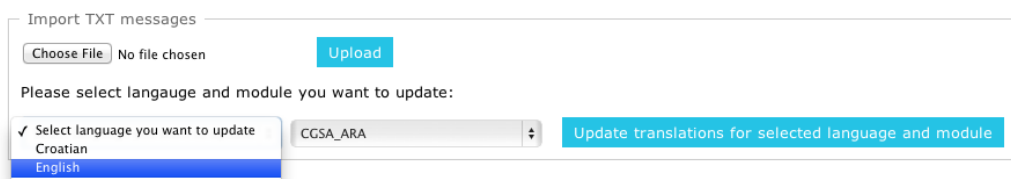
Ta funkcija omogoča, da z enim klikom ustvarimo datoteke `.msg` za vse kombinacije modulov in jezikov hkrati. Ker proces ustvarjanja vseh datotek `.msg` traja nekaj časa, smo implementirali še funkcionalnost za spremljanje napredka, kot je prikazano na sliki 3.26. Proces poteka asinhrono, kar nam omogoča, da lahko spletno aplikacijo naprej neovirano uporabljamo. Datoteke se ustvarijo na datotečnem strežniku, uporablja pa jih aplikacija za avtomatizirano prevajanje programskih modulov podjetja.



Slika 3.26: Prikaz napredka med izdelavo vseh datotek `.msg`.

- Uvoz iz TXT datotek

Ko prevode v izvoženi datoteki `.txt` ustrezno dopolnimo, jo lahko s pomočjo tega modula uvozimo (slika 3.27). Če datoteka ne vsebuje nobenih sprememb, nas spletna aplikacija na to opozori in ne dovoli uvoza. Sistem samodejno prepozna kateri jeziki so zapisani v datoteki in samo te ponudi za uvoz. Prav tako iz seznama modulov samodejno izbere tistega, kateremu so prevodi namenjeni.



Import TXT messages

No file chosen

Please select language and module you want to update:

✓ Select language you want to update
Croatian
English

CGSA_ARA

Slika 3.27: Primer uvoza datoteke `.txt`.

Poglavje 4

Sklepne ugotovitve

Diplomsko delo zajema opis tehnologij in aplikacij, ki smo jih potrebovali, da smo izdelali spletno aplikacijo za potrebe poslovanja podjetja s partnerji. Spletna aplikacija skupaj s spletnim servisom, ki smo ga izdelali, omogoča veliko lažje in hitrejše naročanje in aktivacijo programske opreme. Prav tako bo spletna aplikacija razbremenila zaposlene v podjetju, ki morajo dnevno razbirati naročila partnerjev in jih prepisovati v bolj pregledno obliko. Obvestila o naročilih bodo po novem točno definirana in vedno v enaki obliki.

Spletna aplikacija ponuja možnost lahke nadgradnje z novimi funkcionalnostmi. V bodoče lahko v spletni aplikaciji implementiramo še obvestila za partnerje, možnost prenosa programske opreme podjetja kar iz spletne aplikacije in druge uporabne funkcionalnosti.

Literatura

- [1] M. MacDonald, A. Freeman, M. Szpuszta, "Pro ASP.NET in C# 2010, Fourth Edition", New York: Apress, 2010, str. 1-275.
- [2] (2012) .NET Framework. Dostopno na:
http://en.wikipedia.org/wiki/.NET_Framework.
- [3] (2012) Ajax (programming). Dostopno na:
[http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)).
- [4] (2012) Apache Subversion. Dostopno na:
http://en.wikipedia.org/wiki/Apache_Subversion.
- [5] (2012) ASP.NET. Dostopno na:
<http://en.wikipedia.org/wiki/ASP.NET>.
- [6] (2012) ASP.NET AJAX. Dostopno na:
http://en.wikipedia.org/wiki/ASP.NET_AJAX.
- [7] (2012) Balsamiq. Dostopno na:
<http://www.balsamiq.com>.
- [8] (2012) Bug tracking system. Dostopno na:
http://en.wikipedia.org/wiki/Bug_tracker
- [9] (2012) C# (programming language). Dostopno na:
[http://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)).

- [10] (2012) Integrated development environment. Dostopno na:
http://en.wikipedia.org/wiki/Integrated_development_environment.
- [11] (2012) Market share of statistics shows Oracle is still brand leader in database software. Dosegljivo na
<http://www.technoreporter.com/editors-choice/oracle-is-still-brand-leader>.
- [12] (2012) Microsoft SQL Server. Dosegljivo na
http://en.wikipedia.org/wiki/Microsoft_SQL_Server.
- [13] (2012) TortoiseSVN. Dostopno na:
<http://en.wikipedia.org/wiki/TortoiseSVN>.
- [14] (2012) Windows Communication Foundation.
Dosegljivo na http://en.wikipedia.org/wiki/Windows_Communication_Foundation.