

**SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I
BRODOGRADNJE**

DIPLOMSKI RAD

**IZRADA WEB APLIKACIJE ZA
POSLOVANJE SERVISA**

Ivan Škember

Split, rujan 2010.



Diplomski studij: **Računarstvo**

Smjer/Usmjerenje:

Oznaka programa: 250

Akadska godina: 2009/10

Ime i prezime: **IVAN ŠKEMBER**

Broj indeksa: 794-2008

ZADATAK DIPLOMSKOG RADA

Naslov: IZRADA WEB APLIKACIJE ZA POSLOVANJE

Zadatak: .

Napraviti analizu procesa servisnog poslovanja. Opisati izabranu tehnologiju i metodu razvoja. Definirati funkcionalne i nefunkcionalne zahtjeve na aplikaciju. Opisati logički dizajn aplikacije. Napraviti web aplikaciju i opisati module razvijene aplikacije.

Prijava rada: 25.03.2010.

Rok za predaju rada: 20.09.2010.

Rad predan:

Predsjednik
Odbora za diplomski rad:

Mentor:

Dr. sc. Sven Gotovac, red. prof.

Dr. sc. Stipo Čelar, doc.

SADRŽAJ

1. SAŽETAK I KLJUČNE RIJEČI.....	3
2. UVOD.....	4
3. OPIS PROBLEMA I POSLOVNOG PROCESA	5
3.1. Opis problema	5
3.1.1. Radni nalozi	5
3.1.2. Radnici	5
3.1.3. Skladište	6
3.1.4. Računi	6
3.1.5. Servisne knjižice	6
3.2. Opis procesa	7
3.2.1. Redovito održavanje	7
3.2.2. Vanredni servis	7
3.2.3. Servis na zahtjev 3. lica	7
4. OPIS KORIŠTENIH METODA I TEHNOLOGIJA.....	9
4.1. ASP.NET 4.0.....	9
4.2. IIS 7.0	10
4.3. AJAX.....	11
4.4. jQuery.....	12
4.5. MS SQL SERVER 2008	13
4.5.1. Entity Framework	14
4.5.2. Full text search	15
4.6. Agilne metode razvoja	15
5. ZAHTJEVI ZA IZRADU APLIKACIJE	17
5.1. Funkcionalni zahtjevi	17
5.2. Nefunkcionalni zahtjevi	17
5.3. Opis aplikacije.....	17
5.4. Modeli entiteta i veza	19
6. OSNOVNI MODULI	23
6.1. Skladište	23

6.1.1.	Popis skladišta.....	23
6.1.2.	Artikli	23
6.1.3.	Dobavljači	25
6.1.4.	Narudžbe	26
6.1.5.	Prijenosi	26
6.2.	Radnici	27
6.2.1.	Popis radnika.....	27
6.2.2.	Popis aktivnosti	28
6.3.	Plovila.....	28
6.3.1.	Popis plovila.....	28
6.3.2.	Popis vlasnika	29
6.4.	Predlošci aktivnosti i zadataka	30
6.4.1.	Predefinirane aktivnosti	31
6.4.2.	Vanjske aktivnosti.....	31
6.4.3.	Predlošci zadataka.....	32
7.	MODULI ZA UPRAVLJANJE POSLOVNIM PROCESOM.....	33
7.1.	Zadaci	33
7.1.1.	Novi redovni zadatak	33
7.1.2.	Novi vanredni zadatak	35
7.1.3.	Novi zadatak za 3. lice	36
7.1.4.	Popis zadataka.....	37
7.2.	Obračuni	37
7.2.1.	Izdavanje ponude	37
7.2.2.	Izdavanje računa	38
7.2.3.	Popis dokumenata	38
8.	ZAKLJUČAK.....	39
9.	PRILOZI.....	40
9.1.	Popis slika	40
9.2.	Popis oznaka i kratica.....	41
9.3.	Ostali prilozi i dokumentacija	41
10.	LITERATURA	42

1. SAŽETAK I KLJUČNE RIJEČI

Sažetak

Kako bi se olakšao proces vođenja poslovanja servisa i omogućio pristup informacijama u svakom trenutku, upravljanje poslovanjem je riješeno kroz web aplikaciju. Aplikacija omogućava rad sa više servisnih lokacija i sa više skladišta. Voditelj marine kreira zadatke koji se izvršavaju na pojedinom brodu. Omogućeno je planiranje zadataka. Svaki zadatak se sastoji od jedne ili više aktivnosti sa pripadajućim artiklima i odabranim radnicima. Pri dodavanju radnika, voditelj ima uvid u raspoloživost pojedinog radnika. Nakon što radnici odrade aktivnosti koje su im dodijeljene, unesu odrađene radne sate i potrošeni materijal. Sada voditelj jednostavno iz odrađenog zadatka može kreirati račun. Na temelju zadatka, stavke računa se automatski dodaju. Voditelj po potrebi izmijeni stavke i izdaje račun. Svi zadaci koje se vrše na plovilu ostaju evidentirani pa je moguće generiranje servisne knjižice plovila. Omogućen je uvid u stanje artikala na skladištu u svakom trenutku, te evidencija svih narudžbi artikala i prijenosa artikala između skladišta.

Ključne riječi

upravljanje skladištem, evidencija raspoloživosti radnika, kreiranje radnih naloga, ASP.NET 4.0, Ajax, jQuery, MS SQL Server 2008

2. UVOD

Upravljanje poslovanjem jednog servisa zahtijeva vođenje ljudskih i materijalnih resursa. Servisi koji nemaju adekvatnu programsku podršku sve rade ručno, tj. voditelj marine piše dnevne radne naloge za svakog radnika. Radnik po radnom nalogu obavlja posao i evidentira potrošeni materijal i radne sate. Te na kraju voditelj na temelju radnog naloga izdaje račun.

Planiranje i efikasno iskorištavanje ljudskih resursa nije lagano bez detaljnog uvida u sve postojeće aktivnosti koje su dodijeljene svakom pojedinom radniku. Kako se ne bi dogodilo da neki artikal nije raspoloživ u trenutku kada radnik izvršava servis potrebno je stalno voditi ažurnu evidenciju svih artikala na skladištu.

Tema ovoga rada je razvoj programskog rješenja koje omogućava lakše upravljanje aktivnostima servisa i rješava uočene probleme u poslovanju. Aplikacija je prilagođena za servis plovila ali uz minimalne preinake se može primijeniti na ostale servise.

U trećem poglavlju su opisani problemi koji se javljaju pri radu servisa kao što su problem vođenja radnih naloga, praćenja zauzeća radnika, izdavanja računa bez adekvatne aplikacije itd. Osim navedenih problema opisani su poslovni procesi u servisu. U četvrtom poglavlju su opisane korištene tehnologije u razvoju programskog rješenja za navede probleme iz trećeg poglavlja, te je opisana korištena metoda razvoja. U petom poglavlju su navedeni fizički i logički zahtjevi za izradu aplikacije. Opisan je logički dizajn aplikacije i moduli od kojih se aplikacija sastoji. U šestom poglavlju detaljno su opisani osnovni moduli aplikacije, tj. moduli u kojima se unose ljudski i materijalni resursi servisa, plovila te vlasnici plovila. U sedmom poglavlju su opisani moduli pomoću kojih se upravlja poslovnim procesom.

3. OPIS PROBLEMA I POSLOVNOG PROCESA

3.1. Opis problema

Pružanje usluge servisiranja za sobom povlači dosta administrativnog posla koji odnosi dosta vremena, a koji se mora dobro odraditi kako bi servis uspješno poslovao.

3.1.1. Radni nalozi

Za svakoga radnika je potrebno kreirati dnevni radni nalog na temelju kojeg će radnik vršiti servis. Kreiranje takvih naloga odnosi dosta vremena jer se radni nalozi kreiraju za svakog radnika posebno a ne po brodu. To dodatno otežava evidenciju svih poslova obavljenih na određenom brodu. Osoba koja kreira radne naloge nema mogućnosti planiranja zadataka.

Prije dodjeljivanja određenog zadatka radniku mora se provjeriti da li su mu na raspolaganju svi artikli za koje se zna da će mu biti potrebni pri servisu. Kreiranje zadatka koji se izvršava nekoliko dana nije moguće, već osoba koja kreira radne naloge mora voditi računa da zadatak nije još odrađen i ponovo da dodijeliti nekom radniku.

Radnici na papiru dobiju dnevni radni nalog i na njega zapisuju koliko su dugo radili neki zadatak i koje su artikle potrošili. Netko mora evidentirati podatke koje je radnik unio. Dodatni problem se javlja u tome da radnici često zaborave naznačiti koje su artikle potrošili, osobito ako se to radi o artiklima koja se naručuju na veliko a troše u litrama ili kilogramima kao što su ulja i sl. Da radnik dobije radni nalog na kojem mu piše da mora obaviti redoviti servis broskog motora i da mu pod artiklima već stoji da mora uzeti 4 litre ulja, tada bi mogućnost pogreške bila puno manja.

3.1.2. Radnici

Potrebno je voditi evidenciju o dostupnosti radnika tj. je li radnik raspoloživ, na godišnjem ili na bolovanju. Taj podatak je potreban osobi koja kreira radne naloge iz razloga da se ne dogodi da dodijeli neki posao radniku koji nije raspoloživ.

Za svakoga radnika se mora voditi evidencija svih poslova koje je odradio iz više razloga:

- Može se vidjeti produktivnost radnika
- Lakše je obračunati plaće
- Zna se tko je bila odgovorna osoba za određeni posao

Bez aplikacije brzo dobivanje informacije tko je odradio određeni posao na nekom brodu nije moguće.

3.1.3. **Skladište**

Potrebno je voditi popis svih artikala koji se nalaze u skladištu. Za svaki artikl se mora znati po kojoj je cijeni nabavljen, koja je prodajna cijena i kolika je trenutna količina u skladištu. Nije moguće brzo dobiti podatak o trenutnom stanju svih artikala. Da bi se dobio taj podatak mora se detaljno evidentirati svako uzimanje sa skladišta, a to znači da se moraju uzeti svi odrađeni dnevni radni nalozi i pregledati koje su artikle radnici uzeli.

Mora se voditi stanje svih narudžbi kako bi se izbjeglo višestruko naručivanje istog artikla. Ako servis raspolaže sa više skladišta tada je posao višestruko složeniji. U tom slučaju treba evidentirati i prebacivanje artikala iz skladišta u skladište. Ako je skladište veliko nije lako pronaći traženi artikl.

Nije moguće planiranje potrošnje, nego se potrošnja artikala evidentira tek nakon što su uzeti. Kako bi se izbjegla situacija da u skladištu ne postoji artikl na dan izvršavanja radnoga naloga netko mora stalno voditi računa o tome. Ne postoji nikakav sistem da obavijesti korisnika da je količina određenog artikla pala ispod dopuštene.

3.1.4. **Računi**

Servis mora izdavati ponude i račune. Računi se izdaju na temelju dnevnih radnih naloga. Ako se želi izdati račun za održavanje u nekom periodu potrebno je proći kroz radne naloge svih radnika u tome periodu kako bi se našli svi servisi na brodu.

Ako neki klijent plaća paušalno održavanje tada treba voditi računa koliko je radnih sati paušalnog održavanja ugovoreno i koliko je radnih sati već potrošeno kako mu se ne bi dvaput izdao račun za isti posao.

Izdavanje računa bez adekvatne aplikacije je komplicirano i korisnik može lako napraviti pogrešku. Nema mogućnost lake pretrage starih računa i ponuda.

3.1.5. **Servisne knjižice**

Ako želimo imati servisnu knjižicu broda, potrebno je evidentirati svaki servis na brodu. To znači da osim evidentiranja servisa pri izdavanju račun, korisnik mora isti servis evidentirati i u servisnoj knjižici.

Klijenti koji su ugovorili održavanje broda ne mogu u svakom trenutku vidjeti koji se zadaci trenutno vrše na njihovom plovilu. Osim starih servisa nemaju uvid u troškove koji su obavljani.

3.2. Opis procesa

Servis pruža redovito održavanje brodova koji su pod upravom Charter Management firmi. Osim redovitog i vanrednog održavanja brodova stalnih klijenata, vrši servis brodova za treća lica tj. za osobe koje zatraže određeni servis na svom plovilu a nemaju ugovoreno održavanje.

Voditelj marine kreira radni nalog za svakog radnika. U radnom nalogu stoje aktivnosti koje radnik mora obaviti taj dan i na kojem brodu. Radnik na temelju radnoga naloga izvršava servis brodova i zapisuje koliko je koja aktivnost trajala i koji je materijal uzeo.

3.2.1. Redovito održavanje

Voditelj marine daje radniku radni nalog po kojem mora obaviti održavanje broda. Redovito održavanje se obavlja u određenim vremenskim intervalima ovisno o dogovoru. Sav materijal potreban za servis radnik uzima iz skladišta i pritom zapisuje što je uzeo i koju količinu. Po završetku svakog zadatka na radnom nalogu zapisuje koliko je vremena radio koju aktivnost. Radni nalog predaje upravitelju marine koji na temelju njega izdaje račun vlasniku broda. Ako vlasnik paušalno uplaćuje određeni broj radnih sati održavanja tada mu se radni sati oduzimaju od tog broja a naplaćuju svi koji premaše danu brojku.

3.2.2. Vanredni servis

U vanredni servis spadaju svi zadaci koje je potrebno izvršiti na brodu zbog nastale štete ili drugih nepredviđenih događaja koji ne ulaze u redovito održavanje.

Vlasnik broda ili Charter Management firma koja upravlja brodom zatraži vanredni servis. Vrši se procjena rada na temelju koje voditelj marine kreira radni nalog. Radnik vrši servis i zapisuje koliko je pojedina aktivnost trajala, te koji je materijal potrošen. U slučaju da je radnik izvršio dodatni zadatak koji nije bio u radnom nalogu tada zapisuje koji je zadatak izvršio, koliko je trajao te koji je materijal potrošio.

Voditelj marine izdaje račun vlasniku broda ili osobi koja je prouzročila štetu.

3.2.3. Servis na zahtjev 3. lica

Osoba koja nema ugovoreno održavanje može zatražiti servis ili neku drugu uslugu koju servis pruža. Vrši se procjena na temelju koje se kreira ponuda. Kada stranka prihvati ponudu

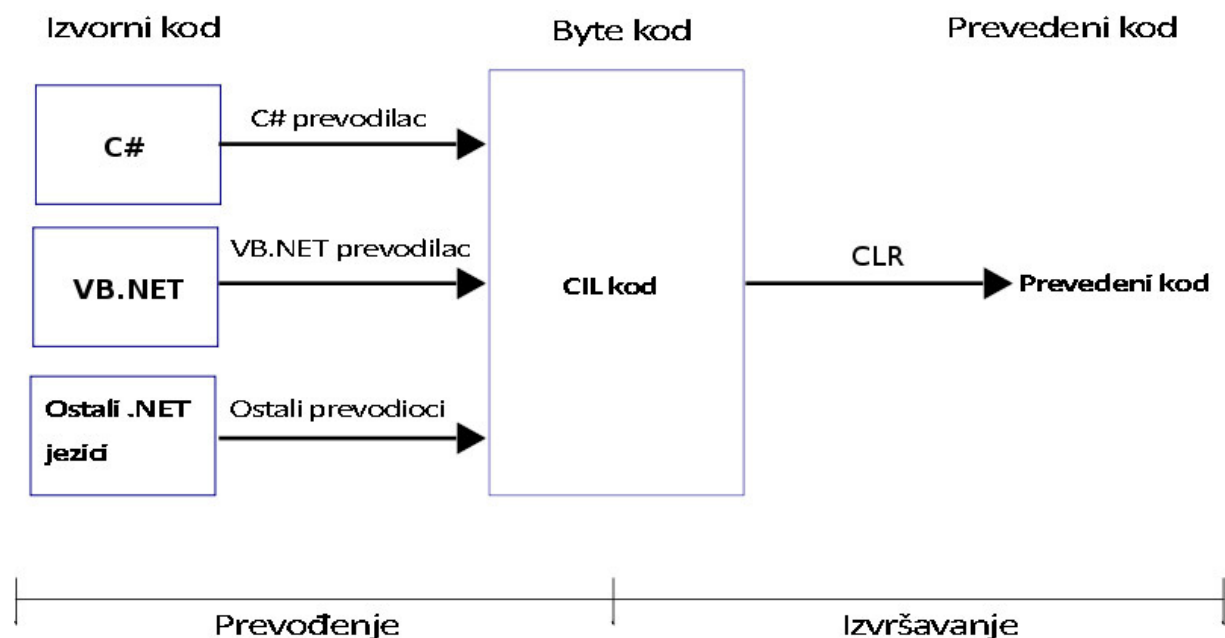
kreće se u kreiranje radnoga naloga prema kojem će radnici raditi. Radnik zapisuje vrijeme izvršavanja pojedinih aktivnosti i potrošeni materijal. Nakon obavljenog servisa slijedi izdavanje računa za danu uslugu.

4. OPIS KORIŠTENIH METODA I TEHNOLOGIJA

Odabrao sam sljedeće tehnologije za razvoj ove aplikacije zbog okruženja u koji će aplikacija biti isporučena, lako dostupnih razvojnih alata, dobre dostupne dokumentacije te predznanja tehnologija.

4.1. ASP.NET 4.0

ASP.NET je web aplikacijsko okruženje koje je razvio Microsoft kako bi omogućio programerima da izrađuju dinamične web stranice, web aplikacije i web servise. ASP.NET je nasljednik Microsoft Active Server Pages (ASP) tehnologije. ASP.NET je izrađen na Common Language Runtime (CLR) tehnologiji, što omogućava programerima da pišu kod koristeći bilo koji od podržanih programerskih jezika.



Slika 4.1 CLR dijagram

CLR je osnovna komponenta .NET-a. To je implementacija Common Language Infrastructure (CLI) standarda, koji definira okruženje za izvršavanje programskog koda. U CLR-u, kod se izražava u obliku byte koda nazvanog Common Intermediate Language (CIL). Programeri koriste CLR na način da pišu kod u odabranom programskom jeziku. Tokom prevođenja .NET prevodilac pretvara takav kod u CIL kod. A tijekom izvršavanja aplikacije CLR prevodilac pretvara CIL kod u kod prilagođen operativnom sustavu na kojem se aplikacija izvršava.

Trenutno su podržani Visual Basic, C# i J#. Za razvoj svoje aplikacije sam koristio C# programski jezik. Ne smiju se koristiti različiti programski jezici unutar iste aplikacije.

.NET stranice se službeno zovu „web forme“. Web forme su stranice koje korisnik „poziva“ direktno ili indirektno kroz svoj preglednik interneta. One su sačinjavaju korisničko sučelje koje daje izgled web aplikaciji. Stranice su pisane kao HTML sa opisom serverskih kontrola uz opis programske logike ili programskog koda koji upravlja kontrolama. Kada ih korisnik zatraži, izvršavaju se na serveru i generiraju HTML koji preglednik interneta može prikazati.

4.2. IIS 7.0

Internet Information Services (IIS) – je web server aplikacija sa setom dodatnih modula koji proširuju funkcionalnost.

Microsoft je prvu verziju izdao za Windowsima NT 3.5 dostupnu kao besplatni dodatak. Od tada do danas IIS je prošao kroz 7 verzija. IIS je drugi najpopularniji web server u svijetu ako promatramo koliki se broj web stranica nalazi na njemu. Najnovija verzija IIS 7 podržavana sljedeće protokole: FTP, FTPS, SMTP, NNTP i HHTTP/HTTPS. Osim podrške za ASP.NET IIS podržava i aplikacije pisane u PHP-u.

IIS 7 je izgrađen na modularnoj arhitekturi. Moduli, se mogu pojedinačno dodavati i uklanjati, tako da samo moduli za potrebne funkcionalnosti mogu biti instalirani.

Kod pune instalacije IIS-a 7 uključeni su neki osnovni moduli. Ti moduli pružaju individualne funkcionalnosti potrebne da bi server mogao obraditi zahtjeve.

- HTTP moduli - Koriste se za obavljanje poslova specifičnih za HTTP u cjevovodu za obradu zahtjeva, kao što su reagiranje na informacije i upite primljene u korisničkom zaglavlju, vraćanje HTTP greški, i preusmjeravanje zahtjeva.
- Sigurnosni moduli - Koristi se za obavljanje poslova vezanih za sigurnost u cjevovodu za obradu zahtjeva, kao što su specifične sheme provjere autentičnosti, obavljanje URL autorizacije i filtriranje zahtjeva.
- Moduli sadržaja - Koriste se za obavljanje poslova vezanih uz sadržaj u cjevovodu za obradu zahtjeva, kao što su obrade zahtjeva za statične slike, vraćanje početne stranice ako klijent ne navodi izvor u zahtjevu, i ispis sadržaja direktorija.
- Kompresijski moduli - Koristi se za obavljanje poslova vezanih za kompresiju u cjevovodu za obradu zahtjeva, kao što je komprimiranje odgovora, primjenjujući Gzip

kompresijsko kodiranje prijenosa na odgovore, te obavlja i pre-kompresiju statičkog sadržaja.

- Moduli skladištenja (Caching) - Koristi se za obavljanje poslova vezanih za spremanje u cjevovodu za obradu zahtjeva, kao što su skladištenje obrađene informacije u memoriji na poslužitelju i korištenje spremljenog sadržaja u kasnijim zahtjevima istog resursa.

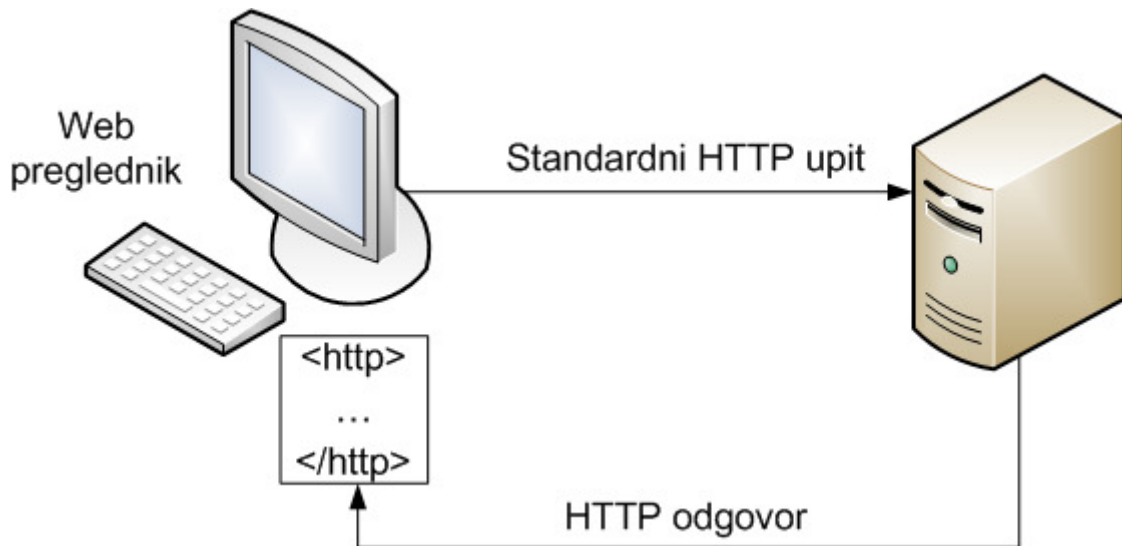
4.3. AJAX

AJAX je relativno nova skraćenica koja se znači Asinkroni JavaScript i XML. To je vrsta zamjenskog pojma koji se koristi za opisivanje vrlo interaktivnih i brzih web aplikacija.

Ajax je skupina međusobno povezanih web razvojnih tehnika koje se koriste na strani klijenta za stvaranje interaktivnih web aplikacija. Sa Ajaxom, web aplikacije mogu preuzimati podatke s poslužitelja asinkrono u pozadini bez uplitanja sa zaslonom i ponašanjem postojeće stranice. Korištenje Ajax tehnika doveo je do povećanja broja dinamičkih interaktivnih sučelja na web stranicama. Podaci se obično dohvaćaju uz pomoć XMLHttpRequest objekta. Usprkos imenu korištenje XML-a nije potrebno, niti zahtjevi moraju biti asinkroni.

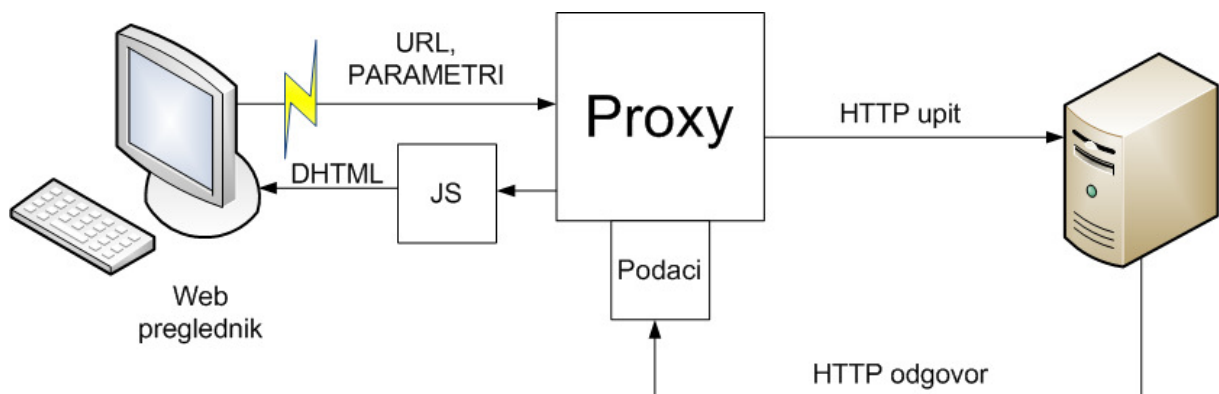
Pojam AJAX je nastao 2005. Nastao je u Java zajednici i opisivao je raspon sličnih tehnologija za implementiranje oblika udaljenog skriptiranja. Danas se svaki oblik udaljenog skriptiranja generalno naziva AJAX.

Klasične web aplikacije rade na način da šalju na Web server forme sa ispunjenim korisničkim podacima i prikazuju korisniku sadržaj koji vraća Web server. Preglednik-server komunikacija se vrši klasičnim HTTP protokolom. HTTP protokol ne održava konekciju, što znači da svaki zahtjev nije vezan sa sljedećim zahtjevom. Komunikacija između preglednika i Web servera se vrši kroz „forme“. Sa korisničke perspektive, transakcija se odvija kroz „stranice“. Svaka korisnička operacija koja šalje novi zahtjev serveru za rezultat ima prikaz potpuno nove stranice.



Slika 4.2 Standardni HTTP zahtjev

Dva osnovna elementa čine AJAX aplikacije živima i uspješnima. Prvo, treba korisnicima prikazati svježije podatke dobivene od servera. Drugo, treba integrirati nove podatke u postojeću stranicu bez potpunog osvježavanja stranice.



Slika 4.3 Ajax upit i odgovor

Ajax radi na način da proxy komponenta šalje standardni HTTP upit sinkrono ili asinkrono i čeka da dobije odgovor. Kad je odgovor stigao i podaci su spremni, proxy poziva odabranu JavaScript funkciju koja osvježava ciljane dijelove stranice.

4.4. jQuery

JavaScript nam omogućava kreiranje dinamičkih stranica kod kojih možemo mijenjati elemente na stranici bez vršenja upita prema serveru. JavaScripta je skriptni programski jezik koji se izvršava na korisničkom računalu unutar web preglednika. Međutim kada imamo mnogo elemenata na stranici kojima pristupamo kod može postati velik i složen.

jQuery je JavaScript biblioteka dizajnirana u svrhu pojednostavljivanja korisničkog JavaScript koda. Prvu verziju je napisao John Resig u siječnju 2006. Danas je koristi preko 30% svih popularnijih web stranica, jQuery je najpopularnija JavaScript biblioteka današnjice.

jQuery je besplatan i otvorenog je koda. Sintaksa je dizajnirana tako da olakša navigaciju elementa, odabir DOM elemenata, kreiranje animacije, obradu evenata i razvoj ajax aplikacija. jQuery također pruža mogućnost da programeri razvijaju razne dodatke koji se izvršavaju uz pomoć jQuery-a. Danas postoji mnoštvo besplatnih dodataka za jQuery koji omogućavaju brzo kreiranje dinamičkih web stranica.

Microsoft i Nokia su najavili planove da uključe jQuery u njihove platforme. Microsoft planira uključiti podršku za jQuery u Visual Studio u svrhu korištenja sa ASP.NET AJAX okruženjem.

4.5. MS SQL SERVER 2008

Microsoft SQL Server je server za upravljanje relacijskim bazama podatka. SQL Server omogućava pohranu, čitanje i manipulaciju podacima. Osnovni jezik za izvršavanje upita je T-SQL. On osim osnovnih SQL upita dozvoljava složenije stvari poput mijenjanja programskog toka (if, else naredbe). Microsoft SQL Server je nastao iz Sybase SQL Servera 1989 godine. To je bila verzija koja se izvršavala na Unix-u. Prva verzija za Windowse je izašla 1993.

Osnovna jedinica pohrane je baza podataka, koja je sastoji od kolekcije tabela sa unesenim stupcima. Osim tablica, baza može sadržavati objekte kao što su „views“, spremljene procedure (stored procedures), indekse i ograničenja (constrains). Podaci u basi se spremaju primarno u dokumente sa završetkom .mdf. Sekundarni tip dokumenta ima završetak .ndf i služi za spremanje metapodataka.

Glavna metoda dohvata podataka iz SQL Server baze podataka je pisanjem upita. Upiti se pišu u T-SQL-u. Upit deklarativno specificira koji podaci moraju biti dohvaćeni. Upit obrađuje procesor upita, koji određuje niz korak koji moraju biti obavljene kako bi se podaci dohvatili. Niz akcija potrebnih da bi se izvršio upit se naziva plan upita.

SQL Server također dopušta definiranje spremljenih procedura (stored procedures). Spremljene procedure su upiti sa definiranim parametrima u T-SQL-u i spremljeni su na serveru. Spremljene procedure mogu primiti vrijednosti koje im klijent šalje kao ulazne

parametre i slati rezultat kako izlazne parametre. One mogu pozivati definirane funkcije, ostale spremljene procedure kako i pozivati samu sebe. Za razliku od ostalih upita, spremljene procedure imaju definirano ime. Budući je procedura spremljena na serveru smanjen je mrežni promet jer klijent ne mora slati serveru kod koji će se izvršiti, nego samo ime procedure i ulazne parametre.

4.5.1. Entity Framework

ADO.NET Entity Framework je razvojno okruženje za objektno-relacijsko mapiranje (ORM) u .NET-u. Entity Framework vrši apstrakciju relacijske (logičke) sheme podataka spremljenih u bazu i predstavlja ih aplikaciji kao konceptnu shemu.

Na primjer, u bazi se unosi o kupcu i njegove informacije mogu biti spremljeni u tablicu Kupci, narudžbe u tablicu Narudžbe a informacije o kontaktima u tablicu Kontakti. Kako bi aplikacija mogla raditi sa ovom bazom mora znati koje se informacije nalaze u kojoj tablici. To znači da relacijska shema baze mora biti kodirana u aplikaciji. Mana ovoga pristupa je da ako se promijeni shema baze, aplikacija nije zaštićena od te promjene.

ADO.NET Entity Framework omogućuje programerima da kreiraju aplikacije koje pristupaju podacima na način da programiraju pristup prema konceptnom aplikacijskom modelu umjesto da programiraju direktno prema shemi relacijske baze podataka. Cilj ovoga način je da se smanji količina koda i olakša održavanje.

Entity model podataka (Entity data model – EDM) specificira konceptni model podataka kroz „Entity-Relationship“ model podataka, koji se primarno bavi sa Entitetima i vezama u kojima oni sudjeluju. Preslikavanje elemenata konceptne sheme u logičku shemu mora biti definirano. EDM shema se izražava u „Schema Definition Language“ (SDL), koji je u principu primjena XML-a. Shema preslikavanja je isto izražena u XML-u.

Koristeći Entity Framework možemo kreirati entitete koji sadrže podatke iz više tablica. Programer vidi taj entitet kako jedan objekt i radi sa njim na način da čita njegova svojstva (Properties) i vrši upite nad njim, a EDM se brine o tome da napravi potrebne operacije nad bazom kako bi mogao pročitati podatke iz više tablica i prezentirati ih kao jedan objekt. npr. Entitet „Kupac“ može imati polje „Ime“ iz tablice „Kupci“ i polje „Email“ iz tablice „KontaktPodaci“.

4.5.2. Full text search

„Full text search“ se koristi da se obave brze jezične pretrage tekstualnih podataka u full-text indeksima djelujući na riječi i fraze na osnovu pravila pojedinog jezika. Full-text upiti mogu sadržavati jednostavne riječi i fraze ili više oblika riječi i fraza.

Full-text pretrage su primjenjive u širokom spektru poslovnih scenarija gdje moramo pretraživati veliki broj informacija da bismo dobili rezultat upita.

Da bismo mogli vršiti full-text upite nad nekom tablicom, moramo prvo kreirati full-text katalog. Zatim za svaku tablicu koju želimo pretraživati moramo kreirati full-text indeks, odabrati stupce koji se pretražuju i odabrati jezik pretrage. Tada moramo pričekati da se generiraju full-text indeksi. Ako imamo mnogo podataka, to može trajati dugo, pa se zbog toga osvježavanje indeksa često vrši u vremenu kad je server najmanje opterećen. Jednom kada su stupci tablice dodani u full-text indeks, aplikacije i korisnici mogu vršiti pretrage.

Umjesto full-text upita možemo koristiti „LIKE“ SQL upite, ali oni samo provjeravaju je li riješ sadrži slova iz upit a ne prepoznaju fraze i slične riječi koje bi mogle biti potencijalni odgovor na upit. Osim toga kada pretražujemo veliki broj nestrukturiranih tekstualnih podataka upiti su mnogo sporiji nego kad koristimo isti full-text upit nad istim podacima. LIKE SQL upit nad milijunima redova tekstualnih podataka može trajati nekoliko minuta, dok full-text upit može trajati nekoliko sekundi ili kraće, ovisno o broju redova vraćenih kako rezultat. Full-text upiti su brži od normalnih SQL upit iz razloga što ne pretražuju redove u tablici nego pretražuju indekse koji su prethodno generirani na temelju podataka u redovima.

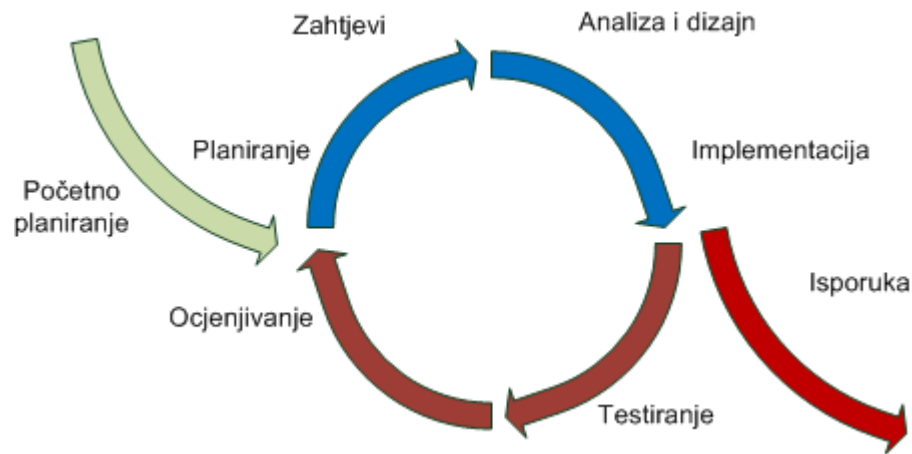
4.6. Agilne metode razvoja

Agilni razvoj softvera je grupa softverskih razvojnih metoda zasnovanih na iterativnom i inkrementalnom razvoju. Zahtjevi i rješenje se razvijaju uz konstantnu suradnju unutar tima. Na ovaj način se potiču učestale inspekcije i adaptacije, timski rad, samoorganizacija i odgovornost.

Mnogo je specifičnih agilnih razvojnih metoda. Mnoge promoviraju razvoj, timski rad, suradnju i prilagođavanje procesa kroz životni ciklus projekta.

Agilne metode razbijaju zadatke u manje inkremente (korake napredovanja) sa minimalnim planiranjem. Iteracije (ponavljanja) su relativno kratke i obično su od jednog do četiri tjedna. Svaka iteracija zahtijeva od tima da prođe kroz cijeli ciklus razvoja softvera, uključujući planiranje, analizu zahtjeva, dizajn, programiranje i testiranje. Nakon toga radna

verzija aplikacije se pokazuje zainteresiranim stranama. Iteracija možda ne dodaje dovoljno funkcionalnosti da bi se softver mogao izbaciti na tržište, ali cilj ovakvog razvoja je da se dobije radna verzija na kraju svake iteracije. Potrebno je nekoliko iteracije da bi se proizvod izbacio na tržište ili da bi se dodale nove funkcionalnosti.



Slika 4.4 Iterativni razvojni proces

Struktura tima u agilnom projektu je obično višefunkcionalna i samo-organizirajuća, tj. ne uzima u obzir korporativne uloge članova tima. Članovi tima snose odgovornost za zadatke koje izvršavaju. Sami odlučuju kako će odgovoriti na zahtjeve svake iteracije.

Pri razvoju ove aplikacije je korištena agilna metoda razvoja. Kao jedan od članova tima sudjelovao sam u svim fazama projekta. Nakon što su zahtjevi bili sastavljeni zajedno sa ostalim članovima tima sam izvršio analizu rješenja.

Kod dizajniranja osnovnog sučelja svi smo davali prijedloge dok nismo dobili zadovoljavajuće rješenje. Jedan član tima je bio zadužen za definiranje funkcionalnosti formi, drugi za grafički dizajn a ja i još jedan član tima smo bili zaduženi za dizajn baze podataka i programiranje sučelja.

Osnovno testiranje je provedeno tokom programiranja svake forme. Pri završetku forme programer testira svoj kod i provjerava zaštitu od pogrešaka pri unosu podataka. Ovo testiranje ne može pronaći sve moguće pogreške ali daje dosta dobre rezultate i ne odnosi previše vremena.

5. ZAHTJEVI ZA IZRADU APLIKACIJE

5.1. Funkcionalni zahtjevi

Aplikacija mora zadovoljiti određene funkcionalne zahtjeve kako bi mogla podržati rad servisa i omogućila upravljanje poslovanjem. Funkcionalni zahtjevi aplikacije su:

- Rad sa više servisnih lokacija
- Rad sa više skladišta
- Unos/izmjena radnika u određenu servisnu lokaciju
- Unos/izmjena vlasnika brodova
- Unos/izmjena informacija o brodovima
- Unos predefiniranih aktivnosti
- Različite kategorije korisnika (Administrator, Voditelj, Radnik, Knjigovođa, Vlasnik plovila)
- Kreiranje radnog naloga
- Ispis dnevnog radnog naloga i zaključivanje pojedinih zadataka
- Izdavanje ponuda i računa na temelju odrađenog radnog naloga
- Prikaz zauzeća radnika po danima
- Prikaz servisne knjižice broda

5.2. Nefunkcionalni zahtjevi

Osim funkcionalnih zahtjeva aplikacija mora zadovoljiti navedene nefunkcionalne zahtjeve:

- Pristup aplikaciji iz preglednika interneta (potrebno korisničko ime i lozinka za pristup aplikaciji)
- Svaka kategorija korisnika ima različita prava pristupa modulima aplikacije
- Mogućnost rada više korisnika u isto vrijeme
- Pretraživanje artikala, radnika i aktivnosti ubrzati korištenjem full-text search tehnologije
- Ispis dokumenata u PDF formatu

5.3. Opis aplikacije

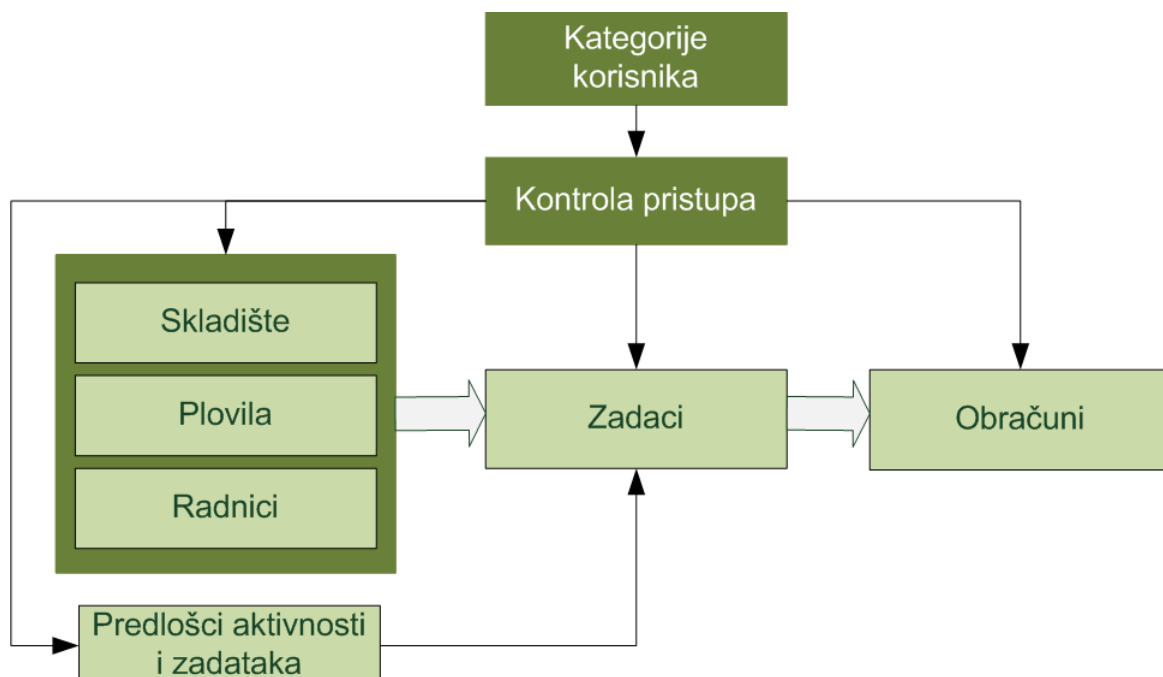
Aplikacija je namijenjena tvrtkama koje se bave održavanjem plovila. Korištenjem sustava moguće je upravljati sa više skladišta i servisnih lokacija. Svako plovilo ima

prispadajući kalendar održavanja, te povijest održavanja. Uzimajući podatke o odrađenom održavanju sustav generira obračune i servisne knjižice. Generirani dokumenti (ponude, radni nalozi, računi) se mogu putem e-maila slati svim zainteresiranim stranama (administratori, voditelji, klijenti, knjigovodstvo).

Aplikacija je sastavljena od modula od kojih svaki upravlja određenim dijelom poslovnog procesa. Moduli aplikacije su:

- Skladište
- Radnici
- Plovila
- Predložci aktivnosti i zadataka
- Zadaci
- Obračuni

Iako se sustav sastoji od modula koji mogu biti predstavljeni zasebnim aplikacijama, integracijom tih modula zaokružen je cjelokupni poslovni proces. Prikazom cjelokupnog procesa, od ulaza materijala u skladište, do izvršenja servisnih aktivnosti omogućeno je praćenje i planiranje troškova, te radnih sati. Dugogodišnjim korištenjem ovakvog sustava u kombinaciji sa statističkim izvještajima koje sustav pruža, omogućena je izgradnja baze znanja, koja će rezultirati donošenjem poslovnih odluka temeljenih na činjenicama.



Slika 5.1 Logički dizajn aplikacije

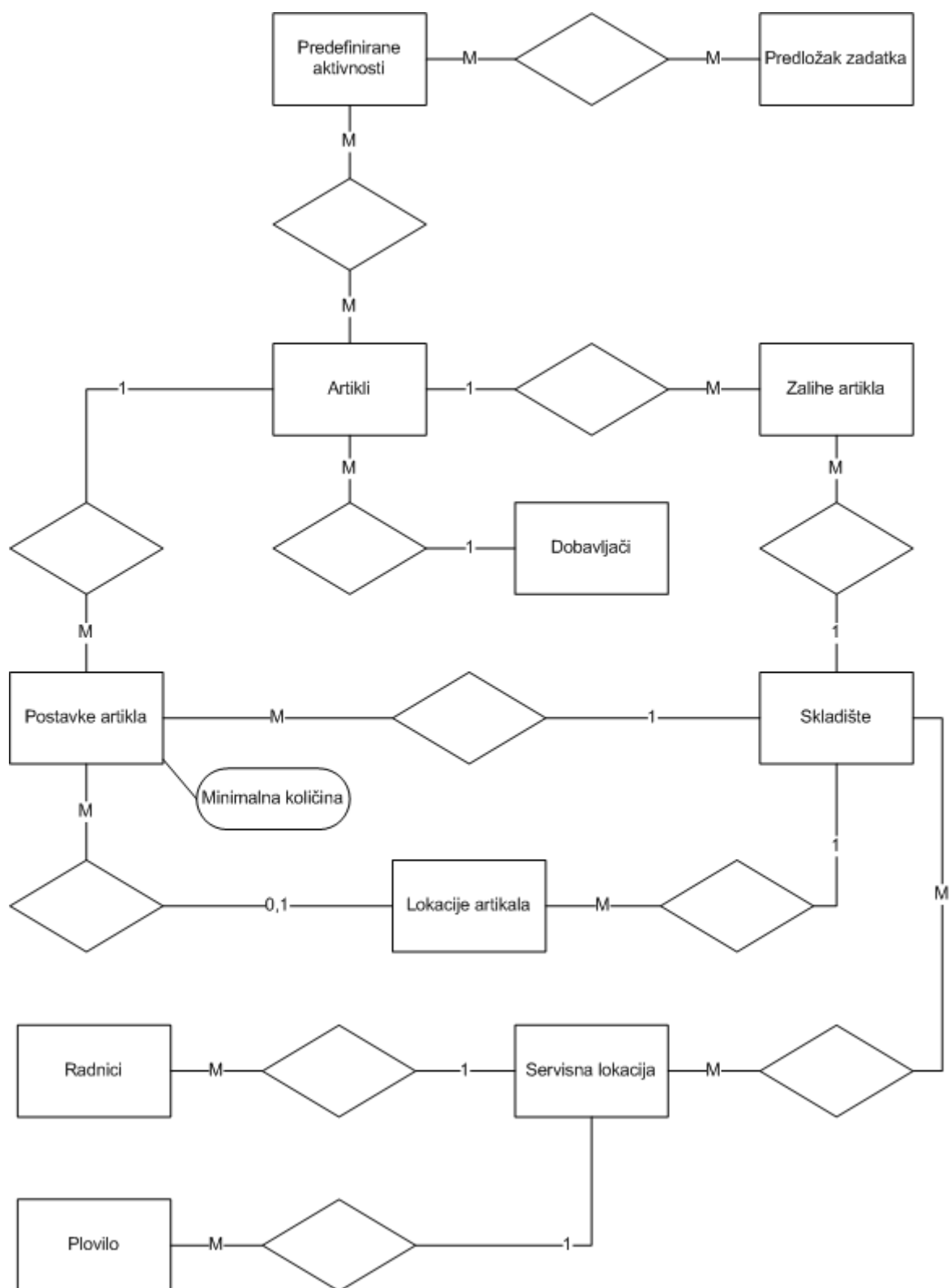
Iz slike 5.1 je jasno vidljiva međuovisnost određenih modula. Prava pristupa dodjeljujemo po kategorijama korisnika a ne po korisniku. Svaki korisnik spada samo pod jednu kategoriju. Kontrola pristupa se brine o pravima pristupa modulima i o svim pravima korištenja istih. Jedan korisnik može imati potpunu kontrolu nad modulom (pregled, unos, izmjena i brisanje podataka), dok drugi korisnik može imati samo pravo pregleda podataka ovisno o tome kojoj kategoriji korisnika su dodijeljeni.

Moduli Skladište, Plovila i Radnici su grupirani u jednu skupinu jer su oni potrebni za kreiranje zadatka. Zadatak uzima podatke iz njih i mijenja ih po potrebi (npr. rezerviranje određene količine artikala iz skladišta, dodavanje aktivnosti radniku te evidentiranje rada nad određenim plovilom).

5.4. Modeli entiteta i veza

U sljedećim slikama su prikazani entiteti i njihove međusobne veze. Model entiteta i veza omogućava apstraktnu i konceptnu reprezentaciju podataka koje moramo spremati u bazi podataka. Na temelju modela entiteta i veza možemo kreirati relacijski model baze podataka.

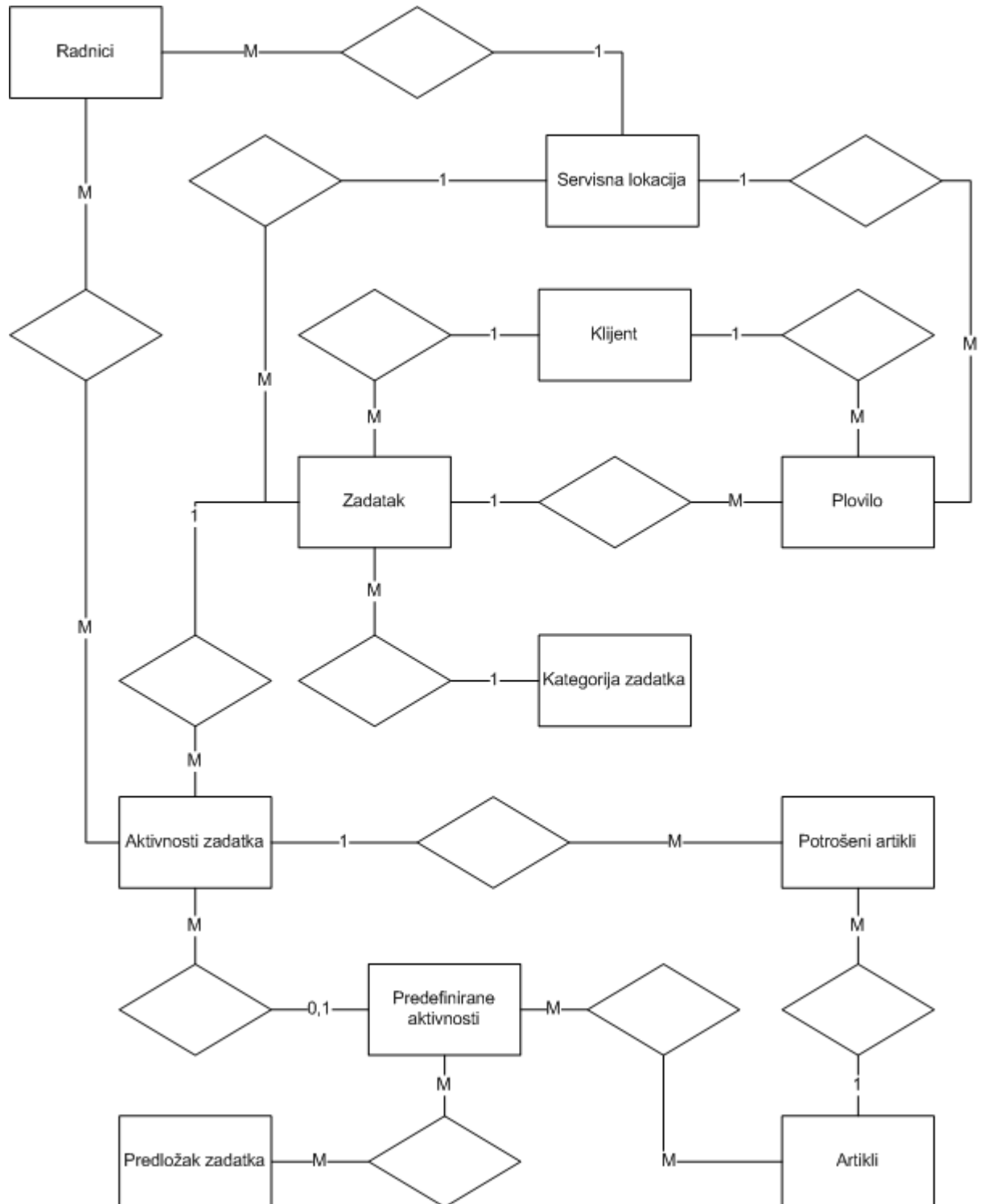
Na slici 5.2 su prikazani entiteti koje koriste moduli Skladište, Plovila, Radnici i Predložci aktivnosti i zadataka. Vidimo da je veza između predložka zadatka i predefiniраниh aktivnosti M-M (many-to-many), to znači da će u relacijskoj bazi podataka ta dva entiteta biti povezana sa među-tablicom koja sadrži primarne ključeve od oba dva entiteta. Isti je slučaj i sa vezom između predefiniраниh aktivnosti i artikala. Entitet „Artikli“ ima definiranog dobavljača te se za svako skladište definiraju zalihe artikala i definirane postavke u kojima se unosi minimalna količina. Za svaki artikl se može definirati i lokacija u skladištu. Entitet skladište je vezan sa servisnom lokacijom sa vezom M-M, na taj način možemo jednoj servisnoj lokaciji dodijeliti više skladišta a jedno skladište može biti dodijeljeno više servisnih lokacija. Radnici i plovila su dodijeljena jednom skladištu.



Slika 5.2 Model entiteti i veze – prvi dio

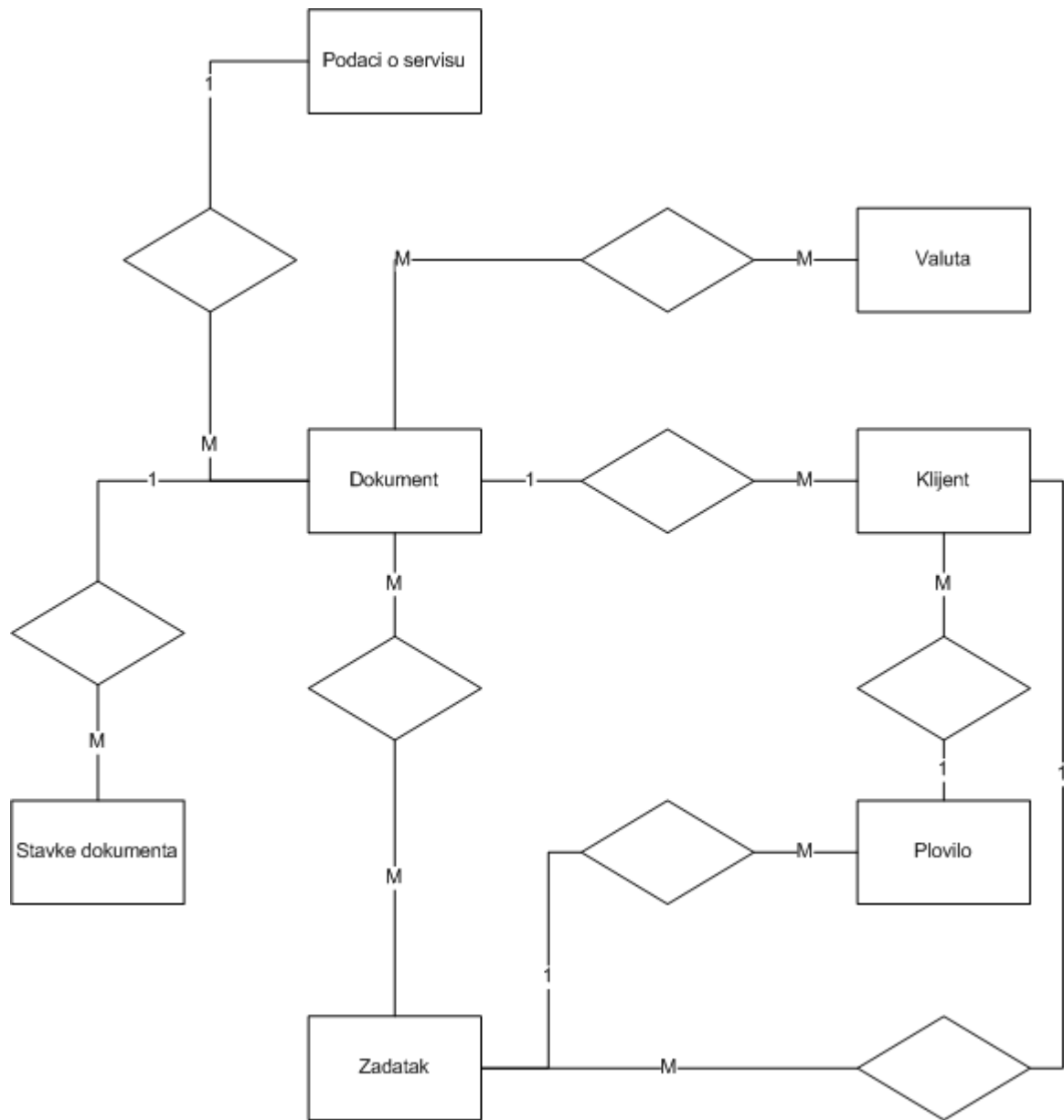
Slika 5.3 prikazuje entitete koje koristi modul Zadaci i vidljive su veze sa entitetima prethodnih modula. Svaki zadatak ima definiranu kategoriju, klijenta i servisnu lokaciju, a plovilo nije obavezno. Vidimo da za svaki zadatak imamo više aktivnosti zadatka, a da svaka

aktivnost zadatka ima više potrošenih artikala. Budući da aktivnosti zadatka mogu nastati iz predefiniраниh aktivnosti, postoji veza 0,1-M između entiteta Predefiniране aktivnosti i Aktivnosti zadatka. Potrošeni artikli se uzimaju od artikala koji su u skladištu pa zbog toga postoji veza između entiteta Potrošeni artikli i Artikli.



Slika 5.3 Model entiteta i veze – drugi dio

Slika 5.4 prikazuje entitete koje koristi modul Obračuni i vidljive su veze sa entitetima modula Zadaci. Dokument ima definiranog klijenta i valutu u kojoj se izdaje pa zbog toga imamo veze između entiteta Dokument i entiteta Klijenti i Valuta. Svaki dokument može imati više stavki pa je zbog toga veza između Dokumenta i Stavki dokumenta 1-M. Budući da je moguće izdati račun za više zadataka, veza između entiteta dokumenti i entiteta Zadaci je M-M.



Slika 5.4 Model entiteta i veze – treći dio

6. OSNOVNI MODULI

Osnovni moduli služe za unos novih, pregled i izmjenu resursa servisa. Oni su temelj sustava na koji se spajaju moduli za upravljanje poslovanjem.

6.1. Skladište

Kako bi se riješio problem vođenja evidencije količina svih artikala razvijen je modul skladište. Ovaj modul služi kako bi se omogućio trenutni uvid u stanje artikala u skladištu/skladištima, kako bi se omogućilo rezerviranje artikala koji se planiraju potrošiti te kako bi se korisnika pravovremeno obavijestilo da je količina određenih artikala pala ispod dopuštene vrijednosti. Sada kada korisnik u svakom trenutku ima ažurne podatke o stanju artikala na skladištu lako mu je donijeti odluku o novim narudžbama ili eventualnim prijenosima artikala iz drugih skladišta koja su dostupna servisu.

6.1.1. Popis skladišta

Aplikacija omogućava rad sa više skladišta i servisnih lokacija. Ova forma služi da se odredi koja skladišta pojedini servis može koristiti.

Korisnik vidi popis svih skladišta koja su dodijeljena lokaciji servisa (slika 6.1). U slučaju da ima administratorska prava može dodavati i uklanjati skladišta iz trenutne servisne lokacije. Za svako skladište može mijenjati naziv i lokaciju.



	Naziv skladišta	Lokacija
	Kaštela	Kaštela
	Solin	Solin

Slika 6.1 Popis skladišta za određenu servisnu lokaciju

6.1.2. Artikli

Nije potrebno isti artikl pojedinačno unositi u svako skladište već ga je potrebno unijeti samo jednom u bilo koje skladište. Tada je potrebno samo unijeti valjane količine artikla u druga skladišta ako artikl i tamo postoji. Svaki artikl ima podatak o minimalnoj količini a ona je različita za sva skladišta. Ako taj podatak nije unesen smatra se da je minimalna količina jednaka nuli. Podatak o minimalnoj količini služi ako želimo obavijestiti korisnika da je trenutna količina određenog artikla pala ispod dopuštene količine kako bi korisnik na vrijeme

izvršio narudžbu. Kako bi se radnicima olakšalo pronalaženje artikla za svaki artikl se može unijeti podatak na kojoj se lokaciji unutar skladišta nalazi.

Novi artikal									
	Haziv artikla	Dobavljač	Lokacija u skladištu	Količina - minimalno	Količina - trenutno	Količina - naručeno	Količina - prijenos	Mjera	Cijena izlazna
	Brašno	Hipodrom d.o.o.	/	0	34,00	0	0	kg	10,00
	Svrđlo 5mm		/	0,00	10,00	5,00	3,00	kom	12,00
	Tekući šampon	Nabava d.o.o.	A1	10,00	10,00	0	0	kom	30,00
	Ulje za podmazivanje		/	0	0	0	0	l	150,00

1 2


Slika 6.2 Popis artikala u skladištu

Klikom na ikonicu za prikaz detalja otvara se forma sa slike 6.3. Tu možemo promijeniti informacije o artiklu.

Naziv artikla	Dobavljač	Lokacija
<input type="text" value="Svrđlo 5mm"/>	<input type="text" value="Nabava d.o.o."/> ▼	<input type="text" value="/"/> ▼
Količina - minimalno	Količina - trenutno	Mjera
<input type="text" value="0,00"/>	<input type="text" value="10,00"/>	<input type="text" value="kom"/>
Cijena - izlazna		
<input type="text" value="12,00"/>		
Spremi		

Slika 6.3 Izmjena podataka o artiklu

Klikom na ikonicu „Naruči“ nam se otvara forma sa slike 6.4.

Naziv artikla	Dobavljač: Nabava d.o.o. Lokacija u skladištu: / Količina - minimalna: 0,00 Količina - trenutna: 10,00 Količina - naručeno: 5,00 Količina - prijenos: 3,00 Mjera: kom Cijena - izlazna: 12,00
<input type="text" value="Svrđlo 5mm"/>	
Datum narudžbe	
<input type="text" value="26.09.2010"/>	
Količina - naručeno	
<input type="text"/>	
Porez	
<input type="text" value="23,00"/>	
Cijena - ulazna	<input type="text"/>
	

Slika 6.4 Nova narudžba

Ova forma na omogućava kreiranje nove narudžbe za odabrani artikl. Sa desne strane imamo informaciju o trenutnom stanju artikla u skladištu.

6.1.3. Dobavljači

Kako bi mogli voditi evidenciju o narudžbama potrebno je svakom artiklu dodijeliti dobavljača. Na taj način korisnik kasnije može raditi izvještaje o narudžbama za svakog pojedinog dobavljača te uvijek ima lako dostupne kontakt podatke.

	Ime	Adresa	Grad	Telefon	E-mail
	Hipodrom d.o.o.	Pravi put 12	Split		
	Links d.o.o.	Teslina 20			
	Nabava d.o.o.	Put grada 10	Split	021/468-342	info@crna-kronika.hr

Slika 6.5 Popis dobavljača

Osim navedenih podataka klikom na link „Detalji“ otvara se forma sa mogućnošću izmjene sljedećih podataka o kupcu:

- Naziv
- Adresa
- Grad
- Telefon
- E-mail

- Fax
- Mobitel
- Web

6.1.4. Narudžbe

Vodi se evidencija svih aktivnih i provedenih narudžbi i korisniku se puža mogućnost kreiranja novih narudžbi. Kada korisnik kreira novu narudžbu ne kreira se narudžbenica već forma služi samo zbog evidencije narudžbe i količine koja je naručena.

	Datum narudžbe	Naziv artikla	Dobavljač	Količina - naručeno	Cijena
	27.8.2010	Svrđlo 5mm	Nabava d.o.o.	5,00	10,00
	30.7.2010	Brašno	Hipodrom d.o.o.	10,00	6,00
	5.7.2010	Brašno	Hipodrom d.o.o.	22,00	7,00
	3.7.2010	Brašno	Hipodrom d.o.o.	8,00	5,26

Slika 6.6 Popis narudžbi

Slika 6.6 prikazuje izgled forme sa popisom narudžbi. Korisnik može mijenjati sve podatke o aktivnoj narudžbi. U slučaju otkazivanja narudžbe može izbrisati istu.

Kada stignu naručeni artikli korisnik odabire link „Primitak i unosi koliko je komada stvarno zaprimljeno. Ovaj podatak se unosi jer se često događa da se naruči određena količina artikala ali dobavljač pošalje veći ili manji broj komada. Jednom kada je narudžba izvršena korisnik je više ne može izbrisati. Ona služi kako povijesni podatak na temelju kojega se kasnije mogu kreirati izvješća.

6.1.5. Prijenosi

Ako korisnik ima više skladišta tada ima mogućnost prijenosa artikala iz skladišta u skladište.

Svaki prijenos je potrebno evidentirati. Kada se kreira novi prijenos količina artikla u prijenosu ostaje ista sve do trenutka kada se ne obavi primitak. Tada sustav automatski korištenjem FIFO (uzima najstarije artikle) metode uzima broj primljenih artikala sa izvorišnog skladišta i unosi je u odredišno skladište. Pri tome pazi da se svi podaci prenesenih artikala sačuvaju informacije o nabavnoj cijeni kako bi se kasnije mogli iskoristiti kod kreiranja izvještaja.

Na formi „Prijenosi“ (Slika 6.7) se nalaze svi aktivni i provedeni prijenosi.

	Datum prijenesa	Naziv artikla	Izvorišno skladište	Određišno skladište	Količina - prijenos
	27.8.2010	Svrđlo 5mm	Split	Solin	3,00
	5.7.2010	Tekući šampon	Kaštela	Solin	7,00
	4.7.2010	Tekući šampon	Kaštela	Solin	12,00
	28.6.2010	Tekući šampon	Kaštela	Solin	1,00

Slika 6.7 Popis prijenesa

Sve aktivne narudžbe se mogu brisati i može se kreirati primitak u kojem kao kod narudžbe upisujemo broj primljenih artikala.

6.2. Radnici

Svi radnici kojima raspolaže servis se unose u ovom modulu. Modul prati zauzeće svakog radnika kako bi omogućio što bolju iskoristivost radnih sati, olakšao planiranje i izbjegao eventualno preopterećivanje radnika.

Svaki radnik vidi aktivnosti koje su njemu dodijeljene. Kad odradi pojedinu aktivnost unese koliko je radnih sati potrošio, i količine svih artikala koje potrošio.

6.2.1. Popis radnika

Novi radnik		
	Ime i prezime	Servisna lokacija
	Joško Katić	Servis Split
	Jure Jurić	Servis Split
	Marko Matic	Servis Kaštela
	Stipe Stipić	Servis Kaštela

Slika 6.8 Popis radnika

Slika 6.8 sadrži popis svih radnika koji rade u servisu, te na kojoj servisnoj lokaciji odrađuju posao.

Klikom na detalji otvara se nova forma u kojoj vidimo sve informacije o radniku:

- Ime
- E-mail
- Lozinka
- Kontakt telefon
- Servisna lokacija

Osim navedenih informacija vidimo „timeline“ kontrolu na kojoj se prikazano zauzeće radnika po danima. Klikom na određeni dan otvara se forma sa popis aktivnosti koje su radniku dodijeljene taj dan.

6.2.2. Popis aktivnosti

Sadrži popis aktivnosti koje su dodijeljene radniku. Na temelju ovoga popisa radnik ispisuje dnevni radni nalog tj. listu aktivnosti koje su mu dodijeljene na odabrani datum.

Popis aktivnosti sadrži sljedeće informacije:

- Datum početka
- Datum završetka
- Brod
- Ime aktivnosti
- Status (nedovršena, dovršena)




Radnik završava aktivnost na način da odabere nedovršenu aktivnost. Nakon toga mu se otvara forma sa prikazanim svim informacijama o aktivnosti. Radnik odabere datum početka i datum završetka u slučaju da su drugačiji od planiranih datuma i unese broj odrađenih radnih sati. Unese količine potrošenih artikala a ako je potrošio neki artikl koji nije na popisu može ga dodati. Nakon što su svi podaci unijeti aktivnost dobiva status dovršene aktivnosti.

6.3. Plovila

Ovaj modul služi za unos informacija o brodovima vlasnicima brodova. Ispravno unesene informacije o vlasniku i o svim njegovim brodovima su preduvjet za točno vođenje evidencije o održavanju svakog pojedinog broda. Svi servisi na pojedinom brodu su detaljno evidentirani. Moguće je uplaćivanje paušalnog održavanja za svaki brod a sustav automatski evidentira potrošene sate. Vlasnik se može uvijek spojiti na aplikaciju i pogledati servisnu knjižicu svog broda.

6.3.1. Popis plovila

Sadrži popis plovila koja su pod upravom servisa, tj. nad kojima se obavlja redovito ili vanredno održavanje. Slika 6.9 prikazuje izgled forme.

	Tip plovila	Haziv plovila	Vlasnik plovila	Lokacija
	Jedrilica	Beneteau First 265	Admiral d.o.o.	Servis Kaštela
	Motorna jahta	Elan 340	Admiral d.o.o.	Servis Split
	Jedrilica	Elan 526	Admiral d.o.o.	Servis Split

Slika 6.9 Popis plovila

Na popisu su prikazane osnovne informacije o brodu. Pritiskom na link „Detalji“ osim osnovnih informacija korisnik ima pravo izmjene sljedećih informacija:

- Cijena paušalnog održavanja
- Broj sati paušalnog održavanja
- Širina
- Gaz
- Duljina
- Motor
- Spremnik za gorivo
- Spremnik za vodu
- Broj kabina
- Broj ležaja
- Broj tuševa/WC-a
- Dodatne informacije

Osim prikaza i izmjene informacija o brodu korisnik na „Timeline kontroli“ vidi sve zadatke koji su se odradili i koji će se odraditi na brodu. Klikom na bilo koji datum mu se prikažu detaljne informacije o svim zadacima.

6.3.2. Popis vlasnika

Svako plovilo mora imati unijete podatke o vlasniku. Ispravno unijeti podaci su preduvjet da omogućimo vlasnicima plaćanje paušalnog održavanja. Osim toga kada imamo ispravne podatke možemo vlasniku slati račune za održavanje i uvijek imamo dostupne kontakt podatke.

	Vlasnik plovila	Grad	Telefon	e-mail
	Admiral d.o.o.	Split	021/886-987	
	Kapato d.o.o.	Split	021/768-648	
	Malison d.o.o.	Split	021/625-465	
	Tremor d.o.o.	Split	021/684-623	

Slika 6.10 Popis vlasnika

Pritiskom na link „Detalji“ prikazuju forma sa slike 6.11 na kojoj možemo izmijeniti podatke o vlasniku plovila.

Tip klijenta

Naziv OIB Država

Grad Poštanski broj Adresa

Telefon Fax Mobitel

Email Web

Ime banke Broj računa Email knjigovođe

[Spremi](#)

Slika 6.11 Izmjena podataka o vlasniku

6.4. Predlošci aktivnosti i zadataka

Svaki zadatak se sastoji od jedne ili više aktivnosti a svaka aktivnost može imati pripadajuće artikle, te se za svaku aktivnost mora odrediti koji je radnik ili izvršavaju. Ako uzmemo za primjer servis koji trenutno ima 10 zaprimljenih brodova, proces kreiranja zadataka može oduzeti dosta vremena. Svrha ove aplikacije je da olakša organizaciju poslovanja ali da pri tome voditelju marine ne stvori dodatan posao već da mu ubrza i olakša vođenje poslovnog procesa.

Kako bi se ubrzao proces kreiranja novih zadataka sve aktivnosti sa pripadajućim artiklima se mogu unijeti u aplikaciju. Kada korisnik kreira novi zadatak ne mora za svaku aktivnost unositi cijenu i pripadajuće artikle već pomoću „full text“ pretrage pronađe odgovarajuću predefiniranu aktivnost i samo odabere vrijeme početka i završetka aktivnosti.

6.4.1. Predefinirane aktivnosti

Predefinirane aktivnosti su aktivnosti koje izvršavaju radnici servisa i koje koriste artikle sa skladišta.

Svaka aktivnost osim naziva i cijene može sadržavati pripadajuće artikle koji se koriste uz nju.

Npr. aktivnost „Zamjena ulja“ ima pripadajući artikl „Ulje“.

Popis aktivnosti sadrži sljedeće informacije:

- Naziv aktivnosti
- Cijena
- Paušalno plaćanje (DA/NE)

Klikom na link „Detalji“ otvara se forma u kojoj korisnik može mijenjati navedene informacije a osim toga vidi i može mijenjati popis svih artikala koje aktivnost koristi.

6.4.2. Vanjske aktivnosti

Vanjske aktivnosti su aktivnosti koje izvršava neka osoba ili tvrtka na zahtjev servisa. Tu aktivnost ne izvršavaju radnici servisa ali može koristiti artikle sa skladišta. Ako se neka vanjska aktivnost često ponavlja možemo je spremiti u aplikaciju kako je ne bi morali svaki put nanovo unositi u procesu kreiranja zadatka.

Vanjska aktivnost sadrži sljedeće informacije:

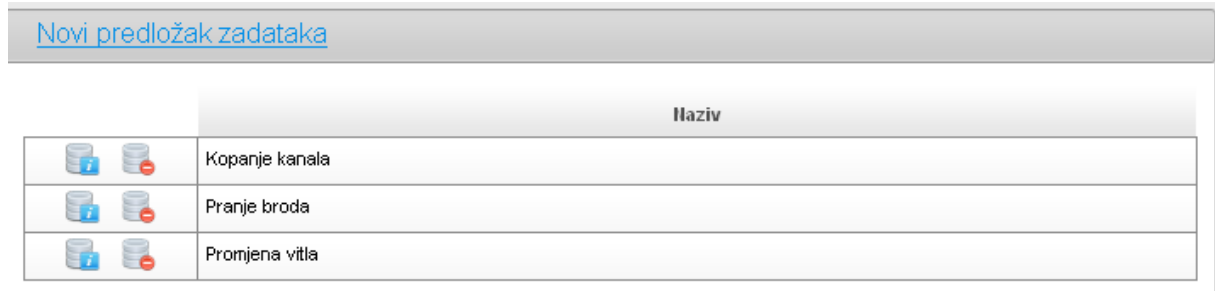
- Naziv aktivnosti
- Cijena



Vanjska aktivnost se ne može plaćati paušalno.

Primjer vanjske aktivnosti je npr. „Izron sidra“ koju obavlja ronilac koji nije radnik u servisu i koji servisu naplaćuje tu uslugu 500 kn.

6.4.3. Predložci zadataka

Ako se neki zadatak redovito ponavlja u određenim vremenskim razmacima tada ga možemo spremi kao predložak. U biti predložak zadatka nije ništa drugo nego kolekcija predefiniраниh aktivnosti i vanjskih aktivnosti grupiranih pod nekim imenom.



Novi predložak zadataka	
	Naziv
	Kopanje kanala
	Pranje broda
	Promjena vitla

Slika 6.12 Popis predložaka zadataka

Korisnik kod kreiranja novoga predložka definira naziv predložka i odabere željene aktivnosti. Sljedeći put kada korisnik kreira novi zadatak jednostavno učita željeni predložak i u zadatak mu se dodaju sve aktivnosti iz predložka.

Svaki predložak zadatka sadrži sljedeće stavke:

- Naziv predložka
- Aktivnosti (predefiniране i/ili vanjske)

Svaka od navedenih stavki se korištenjem predložka unutar kreiranog zadatka može mijenjati. Svaki predložak se može koristiti pri kreiranju bilo koje kategorije zadatka.

7. MODULI ZA UPRAVLJANJE POSLOVNIM PROCESOM

Koristeći resurse osnovnih modula omogućavaju upravljanje ljudskim i materijalnim resursima, komunikaciju između radnika i voditelja, te naplatu odradenog posla.

7.1. Zadaci

Ovaj modul služi za rad na postojećim i novim zadacima. Svaki zadatak sastoji se od jedne ili više predefiniраниh ili vanjskih aktivnosti sa pripadajućim artiklima. Također mu se pruža mogućnost korištenja nekog predloška zadatka kreiranog u prethodnom modulu, pri čemu može mijenjati željene stavke. Svakoј aktivnosti se mora pridodati radnik koji će je izvršiti i mora se odrediti vrijeme početka, završetka i trajanja aktivnosti. Kad se kreira novi zadatak korisnik prolazi kroz nekoliko koraka. Budući da kreiranje novog zadatka ima mnogo stavki na ovaj način se želi korisniku olakšati unošenje i pregled informacija.

Postoje 3 kategorije zadatka:

- Redovni zadatak
- Vanredni zadatak
- Zadatak za 3. lice

U sljedećim koracima će detaljno biti objašnjeno kreiranje svih kategorija zadatka.

7.1.1. Novi redovni zadatak

- Ova kategorija predstavlja zadatke, odnosno skup aktivnosti koji korisnik namjerava koristiti nad istim plovilom više puta u određenim vremenskim intervalima
- Prvi korak pri kreiranju redovnog zadatka je odabir podkategorije zadatka (ljetno, zimsko...)
- Bira plovilo nad kojim se će se zadatak izvršavati
- U drugom koraku unosi sve aktivnosti koje će se odraditi. Unos aktivnosti je moguć na tri načina:
 - Učitavanje željenog predloška zadatka sa pripadajućim aktivnostima. Nakon što je učitao predložak ima mogućnost mijenjanja aktivnosti
 - Unos aktivnosti odabirom neke od predefiniраниh aktivnosti
 - Ako se radi o aktivnosti koju ne izvršava radnik servisa (npr. ronjenje) tada se odabire unos „Vanjske aktivnosti“. Mora se odrediti naziv aktivnosti, i cijena radnoga sata

[Predložci zadataka](#)
Predefinirane aktivnosti
[Vanjske aktivnosti](#)

Naziv aktivnosti

Trajanje (h) Cijena radnog sata

Datum početka Datum završetka

[Dodaj](#)

Ukloni	Naziv aktivnosti	Početak	Završetak	Trajanje (h)	Prioritet
✕	Pranje	01.09.2010	01.09.2010	1	Srednji ▼
✕	Promjena vitla	01.09.2010	01.09.2010	3	Srednji ▼

[Natrag](#) [Dalje](#)

Slika 7.1 Odabir aktivnosti zadatka

- U ovom koraku sustav omogućava pohranu zadatka kao novi predložak zadatka
- Ako su korištene predefinirane aktivnosti onda su već odabrani artikli koji će se koristiti a na korisniku je da po potrebi izmijeni popis artikala i korigira količine artikala
- U slučaju ako kreiramo zadatak koji će se izvršavati za npr. mjesec dana tada ne želimo da se odmah rezerviraju artikli sa skladišta. Tada korisnik unutar sustava postavlja datum upozorenja.
- Na datum upozorenja korisnik prima obavijest o nadolazećem zadatku. Korisnik ponovo otvara zadatak i ima mogućnost izmjene svih postavki zadatka
- Nakon što su odabrane aktivnosti za svaku se mora odrediti planirani početak, planirani završetak i predviđeno trajanje aktivnosti u satima. Trajanje aktivnosti u satima služi radi planiranja radnih sati radnika i naplate.
- Za svaku aktivnost se određuje prioritet kako bi radnik znao koju aktivnost treba prije odraditi. Postoje 3 izbora prioriteta aktivnosti:
 - Niski
 - Srednji
 - Visoki

- Sada kada su određene sve aktivnosti koje će se odraditi ide se na 3 korak u kojem se za svaku aktivnost moraju odabrati radnici. Sustav nam prvo nudi radnike koji u periodu trajanja aktivnosti imaju najviše slobodnih radnih sati.

Aktivnost: **Pranje** Tip: **Normalna** Broj radnika: **1** Broj artikala: **1**

Početak	Završetak	Trajanje	Prioritet	Status
01.09.2010	01.09.2010	1	Srednji ▼	Aktivna ▼

[Sakrij](#)

Radnici **Artikli**

Ime artikla [Dodaj](#)

	Naziv artikla	Mjera	Količina	Cijena	Popust	V.Artikl
✕	Tekući šampon	kom	1	30,00	0	<input type="checkbox"/>

Slika 7.2 Odabir radnika i artikala

- Ako imamo više radnika koji rade istu aktivnost tada moramo definirati koji je radnik voditelj aktivnosti tj. tko nosi odgovornost.
- Nakon što smo unijeli sve radnike, zadatak se aktivira i svi se odabrani artikli rezerviraju sa skladišta.

7.1.2. Novi vanredni zadatak

- Pod kategorijom vanrednih zadataka spadaju zadaci koje je potrebno izvršiti u slučaju štete i sličnih nepredviđenih događaja
- Prvi korak pri kreiranju vanrednog zadatka je odabir podkategorije zadatka (šteta...)
- Bira plovilo nad kojim se će se zadatak izvršavati
- U drugom koraku unosi sve aktivnosti koje će se odraditi. Unos aktivnosti je moguć na tri načina:
 - pomoću predložka zadatka
 - pomoću predefiniranih aktivnosti
 - odabirom vanjske aktivnosti
- Zadatak se može spremi kao novi predložak zadatka
- Mijenjaju se artikli i količine
- Ako se zadatak ne izvršava odmah unosi se datum upozorenja.
- Na datum upozorenja korisnik prima obavijest o nadolazećem zadatku. Korisnik ponovo otvara zadatak i ima mogućnost izmjene svih postavki zadatka

- Za svaku aktivnost se unosi datum početka, završetka i ukupno trajanje aktivnosti u radnim satima.
- Bira se prioritet izvršavanja
- Sada kada su određene sve aktivnosti koje će se odraditi ide se na 3 korak u kojem se za svaku aktivnost moraju odabrati radnici. Sustav nam prvo nudi radnike koji u periodu trajanja aktivnosti imaju najviše slobodnih radnih sati.
- Ako imamo više radnika koji rade istu aktivnost tada moramo definirati koji je radnik voditelj aktivnosti tj. tko nosi odgovornost.
- Nakon što smo unijeli sve radnike, zadatak se aktivira i svi se odabrani artikli rezerviraju sa skladišta.

7.1.3. Novi zadatak za 3. lice

- Ova kategorija opisuje zadatke koje je zatražila (fizička ili pravna) osoba koja nije evidentirana u sustavu kao vlasnik plovila
- Zadatak se ne izvršava na plovilu koje je pod upravljanjem korisnika, stoga sustav preskače stavku za odabir plovila
- Kao i prije spomenute kategorije zadataka, i ovakav zadatak se kreira kao skup jedne ili više aktivnosti koje koriste resurse servisa (artikli iz skladišta i pripadajući radni sati)
- idući korak je odabir jednog od 3 načina kreiranja zadatka:
 - pomoću predloška zadatka
 - pomoću predefiniраниh aktivnosti
 - odabirom vanjske aktivnosti
- Mijenjaju se artikli i količine
- Ako se zadatak ne izvršava odmah unosi se datum upozorenja.
- Na datum upozorenja korisnik prima obavijest o nadolazećem zadatku. Korisnik ponovo otvara zadatak i ima mogućnost izmjene svih postavki zadatka
- Za svaku aktivnost se unosi datum početka, završetka i ukupno trajanje aktivnosti u radnim satima.
- Bira se prioritet izvršavanja
- Sada kada su određene sve aktivnosti koje će se odraditi ide se na 3 korak u kojem se za svaku aktivnost moraju odabrati radnici. Sustav nam prvo nudi radnike koji u periodu trajanja aktivnosti imaju najviše slobodnih radnih sati.

- Ako imamo više radnika koji rade istu aktivnost tada moramo definirati koji je radnik voditelj aktivnosti tj. tko nosi odgovornost.
- Nakon što smo unijeli sve radnike, zadatak se aktivira i svi se odabrani artikli rezerviraju sa skladišta.

7.1.4. Popis zadataka

Slika 7.3 sadrži popis svih završenih i aktivnih zadataka.

	Datum kreiranja	Datum upozorenja	Ime broda	Ime klijenta	Tip zadatka	Status
	14.9.2010	17.9.2010	Elan 526	Admiral d.o.o.	Redovni zadatak	Planiran
	14.9.2010	1.10.2010	Beneteau First 265	Admiral d.o.o.	Redovni zadatak	Planiran
	14.9.2010		Elan 340	Admiral d.o.o.	Redovni zadatak	Aktivan
	14.9.2010		Beneteau First 265	Admiral d.o.o.	Redovni zadatak	Aktivan
	17.9.2010		Beneteau First 265	Admiral d.o.o.	Redovni zadatak	Aktivan
	17.9.2010	3.9.2010	Beneteau First 265	Admiral d.o.o.	Vanredni zadatak	Planiran
	17.9.2010	9.9.2010	Beneteau First 265	Admiral d.o.o.	Vanredni zadatak	Aktivan
	17.9.2010		Elan 526	Admiral d.o.o.	Redovni zadatak	Završen

Slika 7.3 Popis zadataka

Svrha ove forme je da evidentira sve aktivne i završene zadatke radi jednostavnog uvida u trenutno stanje servisa. Korisnik može odabrati bilo koji aktivni zadatak i mijenjati njegove stavke. Administrator može jednostavno pronaći sve završene zadatke i odabrati izdavanje računa za njih. Izdavanje računa je detaljno opisano u modulu obračuni.

7.2. Obračuni

Prethodni moduli su omogućili praćenje stanja artikala na skladištu i praćenje radnih sati radnika, a ovaj modul uzima te informacije i na temelju njih kreira račune i ponude.

7.2.1. Izdavanje ponude


Nekada je klijentu potrebno dati uvid u troškove servisa prije nego se servis obavi, pa se tada izdaje ponuda. Kod kreiranja ponude administrator odabire klijenta iz baze ako je već unesen a ako nije unosi ga u sustav. U slučaju da se posao vrši na brodu unosi ime broda. Odabire se datum izdavanja i dospijeca te valuta u kojoj se izdaje ponuda. Potom administrator pojedinačno unosi svaku stavku ponude. Kako bi se olakšao unos sustav korisniku nudi izbor predefiniраниh stavki, a na korisniku je da za svaku stavku unese cijenu i količinu. Nakon što su svi podaci uneseni ponuda se sprema i ispisiuje ili se šalje e-mailom.

7.2.2. Izdavanje računa

Kako bi se izdavanje računa ubrzalo i kako bi se izbjegle ljudske pogreške, direktno iz završenih zadataka je moguće kreirati račun. Na administratoru je da pregleda završeni zadatak i prepravi ga ako su radnici nešto pogriješili a potom bira kreiranje računa. Na temelju stavki zadatka (radni sati, artikli) se popune stavke računa. Na korisniku ostaje samo da korigira stavke po potrebi, unese datum izdavanja i dospijeća te odabere valutu. Ako se radi o stranoj valuti, tada korisnik može ručno unijeti vrijednost tečaja ako ponuđena vrijednost nije ažurna. Nakon što je račun spremljen, generira se broj računa i račun se može ispisati ili poslati e-mailom.

7.2.3. Popis dokumenata

Sadrži popis svih ponuda i računa. Svrha ove forme je jednostavan pregled i pretraga svih dokumenata. Klikom na link „Detalji“ korisniku se otvara dokument i nudi mu se mogućnost storniranja dokumenta, ispisa dokumenta i slanja dokumenta e-mailom.

	Datum isporuke	Datum valute	Storno	Kupac	Radni nalozi	Tip	Broj računa	Cijena
	2.10.2010	2.10.2010		Kapato d.o.o.	25	Račun	002/2010	1.200,00 kn
	2.10.2010	2.10.2010		Malison d.o.o.	27	Račun	003/2010	365,00 kn
	1.10.2010	1.10.2010		Korniza d.o.o.	20, 21	Račun	001/2010	1.000,00 kn
	28.9.2010	28.9.2010		Tremor d.o.o.		Ponuda	001/2010	5.600,00 kn

Slika 7.4 Popis dokumenata

8. ZAKLJUČAK

Korištenje aplikacije za upravljanje poslovanjem servisa olakšava kontrolu cijelog poslovnog procesa i omogućava u svakom trenutku uvid u trenutno stanje, kao i uvid u povijesne podatke. Radni nalozi i računi su povezani tako da je kreiranje računa ubrzano i olakšano. Korištenje aplikacije povećava produktivnost, olakšava komunikaciju između radnika i voditelja, te smanjuje utjecaj ljudske pogreške. Budući da se radi o web aplikaciji, rad je moguć sa bilo kojeg računala s vezom na internet. Moguć je i pristup putem mobitela koji ima internet preglednik s podrškom za javascript.

Razvijena aplikacija je usmjerena najviše na planiranje ljudskih resursa i kontrolu potrošnje, dok napredne mogućnosti upravljanja skladištem i napredne knjigovodstvene mogućnosti predstavljaju plan daljnjeg razvoja.

9. PRILOZI

9.1. Popis slika

Slika 4.1 CLR dijagram	9
Slika 4.2 Standardni HTTP zahtjev	12
Slika 4.3 Ajax upit i odgovor.....	12
Slika 4.4 Iterativni razvojni proces	16
Slika 5.1 Logički dizajn aplikacije	18
Slika 5.2 Model entiteta i veze – prvi dio.....	20
Slika 5.3 Model entiteta i veze – drugi dio.....	21
Slika 5.4 Model entiteta i veze – treći dio	22
Slika 6.1 Popis skladišta za određenu servisnu lokaciju.....	23
Slika 6.2 Popis artikala u skladištu	24
Slika 6.3 Izmjena podataka o artiklu	24
Slika 6.4 Nova narudžba.....	25
Slika 6.5 Popis dobavljača.....	25
Slika 6.6 Popis narudžbi	26
Slika 6.7 Popis prijenosa.....	27
Slika 6.8 Popis radnika	27
Slika 6.9 Popis plovila	29
Slika 6.10 Popis vlasnika.....	30
Slika 6.11 Izmjena podataka o vlasniku	30
Slika 6.12 Popis predložaka zadataka.....	32
Slika 7.1 Odabir aktivnosti zadatka	34
Slika 7.2 Odabir radnika i artikala.....	35
Slika 7.3 Popis zadataka	37
Slika 7.4 Popis dokumenata.....	38

9.2. Popis oznaka i kratica

ASP.NET	Active Server Pages
CLR	Common Language Runtime
CIL	Common Intermediate Language
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
SMTP	Simple Mail Transfer Protocol
NNTP	Network News Transfer Protocol
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IIS	Internet Information Services
AJAX	Asinkroni JavaScript i XML
XML	Extensible Markup Language
DOM	Document Object Model
SQL	Structured Query Language
ORM	objektno-relacijsko mapiranje
EDM	Entity data model

9.3. Ostali prilozi i dokumentacija

MS Project - DIP_2009-10_Skember-I_794-2008_Servisno-poslovanje.mpp

Evidencija rada - DIP_2009-10_Skember-I_794-2008_Servisno-poslovanje.xlsx

10. LITERATURA

- [1] Wikipedia: „*ASP.NET*“, s Interneta, <http://en.wikipedia.org/wiki/ASP.NET>, 24.08.2010.
- [2] Wikipedia: „*Common Language Runtime*“, s Interneta, http://en.wikipedia.org/wiki/Common_Language_Runtime, 21.08.2010.
- [3] Microsoft: „*What is ASP.NET Web Forms?*“, s Interneta, <http://www.asp.net/web-forms/what-is-web-forms>, 24.08.2010.
- [4] Wikipedia: „*Internet Information Services*“, s Interneta, http://en.wikipedia.org/wiki/Internet_Information_Services, 24.08.2010.
- [5] Microsoft: „*IIS overview*“, s Interneta, <http://www.iis.net/overview>, 24.08.2010.
- [6] Wikipedia: „*Ajax (programming)*“, s Interneta, http://en.wikipedia.org/wiki/Ajax_%28programming%29, 23.08.2010.
- [7] Esposito D. (2007.): „*Introducing Microsoft ASP.NET AJAX*“ Microsoft Press. ISBN 9780735624139
- [8] Wikipedia: „*jQuery*“, s Interneta, <http://en.wikipedia.org/wiki/JQuery>, 18.08.2010.
- [9] Wikipedia: „*ADO.NET Entity Framework*“, s Interneta, http://en.wikipedia.org/wiki/ADO.NET_Entity_Framework, 21.07.2010.
- [10] Microsoft: „*ADO.NET Entity Framework*“, s Interneta, <http://msdn.microsoft.com/en-us/library/bb399572.aspx>, 26.08.2010.
- [11] Microsoft: „*Full-Text Search (SQL Server)*“, s Interneta, <http://msdn.microsoft.com/en-us/library/ms142571.aspx>, 26.08.2010.
- [12] Wikipedia: „*Agile software development*“, s Interneta, http://en.wikipedia.org/wiki/Agile_software_development, 29.08.2010.
- [13] Waters K.: „*Agile Project Management Questions Answered*“, s Interneta, <http://agilesoftwaredevelopment.com/>, 16.11.2009.
- [14] Wikipedia: „*Scrum (development)*“, s Interneta, http://en.wikipedia.org/wiki/Scrum_%28development%29, 31.08.2010.
- [15] Collabnet: „*Scrum Methodology*“, s Interneta, <http://scrummethodology.com/>, 31.08.2010.