



Št. naloge: 00503/2010

Datum: 15.03.2010

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

Kandidat: **JOSIP VARGA**

Naslov: **SPLETNA APLIKACIJA ZA INTERNO NAROČANJE**
WEB APPLICATION FOR INTERNAL ORDERING

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

V diplomski nalogi sistematično razvijte in predstavite poslovno aplikacijo za podporo procesu naročanja artiklov in storitev na Psihiatrični kliniki Ljubljana.

Izhodišče naloge naj bo celovita predstavitev uporabljenih tehnologij in orodij. Jedro naloge naj zajema celovito načrtovanje aplikacije z uporabo metodologije Rational Unified Process (RUP), ki različnim skupinam uporabnikov omogoča oddajo in upravljanje z internimi zahtevki. V zadnjem delu predstavite izgled razvite aplikacije in njeno uporabo.

Mentor:

viš. pred. dr. Igor Rožanc



Dekan:

prof. dr. Franc Solina

**UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO**

Josip Varga

**SPLETNA APLIKACIJA ZA INTERNO
NAROČANJE**

**DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU**

MENTOR: dr. Igor Rožanc

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si



Št. naloge: 00503/2010

Datum: 15.03.2010

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

Kandidat: **JOSIP VARGA**

Naslov: **SPLETNA APLIKACIJA ZA INTERNO NAROČANJE
WEB APPLICATION FOR INTERNAL ORDERING**

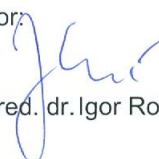
Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

V diplomski nalogi sistematično razvijte in predstavite poslovno aplikacijo za podporo procesu naročanja artiklov in storitev na Psihiatrični kliniki Ljubljana.

Izhodišče naloge naj bo celovita predstavitev uporabljenih tehnologij in orodij. Jedro naloge naj zajema celovito načrtovanje aplikacije z uporabo metodologije Rational Unified Process (RUP), ki različnim skupinam uporabnikov omogoča oddajo in upravljanje z internimi zahtevki. V zadnjem delu predstavite izgled razvite aplikacije in njeno uporabo.

Mentor:


viš. pred. dr. Igor Rožanc



Dekan:


prof. dr. Franc Solina

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a **Josip Varga**,

z vpisno številko 63020306,

sem avtor/-ica diplomskega dela z naslovom:

SPLETNA APLIKACIJA ZA INTERNO NAROČANJE

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)
viš. pred., dr. Igor Rožanc
- 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 14.09.2010

Podpis avtorja/-ice: _____

ZAHVALA

Ob tej priliki, bi se zahvalil staršem, ki sta me potrpežljivo spremljala na nekoliko daljši poti študijski poti.

Zahvaljujem se tudi svojemu mentorju, Igorju Rožancu, za vse napotke in nasvete, ter še posebej za potrpežljivost med zadnjimi dnevi mojega študija.

Zahvala gre tudi vsem sodelavcem, sošolcem in sošolkam, ki so mi stali ob strani in me podpirali na moji poti skozi študentska leta.

Vsem skupaj iskrena hvala!

KAZALO

POVZETEK	7
ABSTRACT.....	8
1 UVOD	9
1.1 Ideja.....	9
1.2 Namen	9
1.3 Cilji.....	9
1.4 Struktura diplomske naloge	9
1.5 Predstavitev Psihiatrične klinike Ljubljana	10
2 POSLOVNE APLIKACIJE.....	11
2.1 Splošen opis in delitev.....	11
2.2 Poslovna aplikacija v diplomski nalogi	12
3 TEHNOLOGIJE IN ORODJA	13
3.1 Označevalni in programski jeziki ter orodja za razvoj aplikacije.....	13
3.1.1 Označevalni jezik HTML	13
3.1.2 Programski jezik PHP	13
3.1.3 Programski jezik JavaScript.....	14
3.1.4 Orodje za spletno programiranje Adobe Dreamweaver	14
3.1.5 Orodje MAMP	14
3.2 Podatkovni strežniki in orodja za administracijo podatkovnih baz	15
3.2.1 Sistem za upravljanje podatkovne baze MySQL.....	15
3.2.2 Orodje PhpMyAdmin.....	15
3.3 Metodologije razvoja in orodja za načrtovanje	16
3.3.1 Metodologija Rational Unified Process (RUP).....	16
3.3.2 Modelirni jezik Unified Modeling Language (UML)	17
3.3.3 Orodje Rational Rose.....	18
3.3.4 Orodje Power Designer	19
4 MODELIRANJE IN IZDELAVA APLIKACIJE	20
4.1 Opis obstoječega stanja	20

	5
4.2 Opis problema	22
4.3 Zajem zahtev	22
4.3.1 Akterji sistema.....	22
4.3.2 Primeri uporabe	23
4.4 Modeli primerov uporabe.....	24
4.5 Procesni model	26
4.5.1 Razredni diagrami	26
4.5.2 Diagrami zaporedja.....	27
4.6 Podatkovni modeli	29
4.6.1 Konceptualni podatkovni model	29
4.6.2 Fizični podatkovni model.....	30
4.7 Opis aplikacije	31
4.7.1 Splošno o aplikaciji.....	31
4.7.2 Uporabniški meni.....	31
4.7.2.1 Podatki o uporabniku.....	32
4.7.2.2 Ukazni meni	32
4.8 Implementacija in analiza narejenega	36
5 SKLEP	37
VIRI.....	38

Seznam uporabljenih kratic

ASP – Active Server Pages – tehnologija za dinamično generiranje spletnih vsebin

B2B – Business 2 Business – komercialne povezave med podjetji

B2C – Business 2 Consumer – komercialne povezave med podjetji in strankami

B2E – Business 2 Employee – komercialne povezave med podjetji in njihovimi zaposlenimi

CGI – Common Gateway Interface – standard za pisanje spletnih programov

CIP – Center za izvenbolnišnično psihiatrijo

CKP – Center za klinično psihiatrijo

CMZ – Center za mentalno zdravje

CSV – Comma Separated Values – format za shranjevanje podatkov

CZOPD – Center za zdravljenje odvisnih od prepovedanih drog

DOC – Document – Microsoftov format za shranjevanje dokumentov

EZOA – Enota za zdravljenje odvisnih od alkohola

HTML – Hyper Text Markup Language – jezik za razvoj enostavnih spletnih strani

JSP – Java Server Pages – tehnologija za hiter razvoj dinamičnih spletnih vsebin

L-/WAMP – Linux/Windows Apache MySQL Php – platforma za razvoj aplikacij na sistemih Linux in Windows

LATEX – orodje za oblikovanje besedil, ponavadi znanstvenih

MAMP – MacOS Apache MySQL Php – platforma za razvoj aplikacij na sistemih MacOS

ODBC – Open Database Connectivity – odprta povezljivost z zbirkami podatkov

PDF – Portable Document Format – Adobov format za shranjevanje dokumentov

Perl – Practical Extraction and Report Language – tolmačeni programski jezik

PHP – PHP Hypertext Preprocessor – jezik za razvoj dinamičnih spletnih strani

PKL – Psihiatrična klinika Ljubljana

RUP – Rational Unified Process – iterativni programski razvojni proces

SQL – Structured Query Language – jezik za pisanje podatkovnih poizvedb

UML – Unified Modelling Language – jezik za modeliranje razvoja programske opreme

VPN – Virtual Private Network – navidezna spletna povezava med različnimi omrežji

XLS – Microsoftov format za shranjevanje preglednic

XML – Extensible Markup Language – označevalni jezik za opis strukturiranih podatkov

POVZETEK

Naloga diplome je sistematično razviti in predstaviti poslovno aplikacijo za podporo procesu naročanja artiklov in storitev v Psihiatrični kliniki Ljubljana. Za doseg cilja smo uporabili tehnologije in orodja kot so: HTML (angl. Hyper Text Markup Language), PHP (trenutno tri črkovni rekurzivni akronim za angl. PHP Hypertext Preprocessor), MySQL, JavaScript, Adobe Dreamweaver, MAMP (angl. Macintosh Apache MySQL PHP), Power Designer, PhpMyAdmin, Rational Rose, RUP (angl. Rational Unified Process) in UML (angl. Unified Modelling Language). HTML, PHP in JavaScript so jeziki za spletno programiranje, Adobe Dreamweaver pa smo uporabili za programiranje celotne aplikacije. Za poganjanje aplikacije smo uporabili orodje MAMP, ki simulira spletni strežnik na lokalnem računalniku in omogoča poganjanje spletnih tehnologij. PhpMyAdmin je priročno orodje, ki služi kot vmesnik za lažjo administracijo podatkovne baze MySQL. S Power Designer-jem smo naredili fizični in konceptualni model podatkovne baze. Celotna aplikacija je bila načrtovana po metodologiji RUP in predstavljena z diagrami modelirnega jezika UML.

Za podporo procesu internega naročanja smo zasnovali spletno aplikacijo, kjer vse faze procesa potekajo v elektronski obliki. Sama izvedba spletne aplikacije obsega delovanje med dvema fazama procesa, ki jasno prikazujeta možno delovanje celotnega sistema. To je bil tudi cilj diplomske naloge. Implementacija spletne aplikacije bi tako prinesla učinkovite (časovne), ekonomske in ekološke izboljšave in posledično finančne prihranke.

Ključne besede:

informacijski sistem, B2E, RUP, UML, spletna aplikacija, poslovna aplikacija, spletni obrazec

ABSTRACT

This thesis task is to systematically develop and represent business application for process of internal ordering in University Psychiatric Hospital Ljubljana. To achieve this goal we used several tools and technologies, such as: HTML (Hyper Text Markup Language), PHP (PHP Hypertext Preprocessor), MySQL, JavaScript, Adobe Dreamweaver, MAMP (Macintosh Apache MySQL PHP), Power Designer, PhpMyAdmin, Rational Rose, RUP (Rational Unified Process) and UML (Unified Modelling Language). HTML, PHP and JavaScript are web programming languages, which we uses in Adobe Dreamweaver for application programming. Web application was running on MAMP, which offers virtual web server and runs PHP, MySQL and other technologies. It also includes PhpMyAdmin, which helps user with easier MySQL database administration. Power Designer was used for creating physical and conceptual data models. For application design we used RUP methodology and UML diagrams for representation.

To support all phases of process of internal ordering, we designed a web application. Implementation of web application includes two critical steps of this process, which demonstrates the use of whole system. The use of new application would bring cost-efficiency and ecological benefits.

Keywords:

Information system, B2E, RUP, UML, web application, business application, web form

1 UVOD

V uvodu bomo opredelili idejo, namen in cilje diplomske naloge. Opisana je tudi struktura diplomske naloge, ki je razdeljena na poglavja.

1.1 Ideja

V času informacijskih tehnologij, v katerem živimo, stremi večina podjetij, ki ponuja svoje storitve, k čim boljši izkoriščenosti in optimizaciji delovnih procesov. V ta namen se veliko vlaga v razvoj aplikacij za elektronsko poslovanje in iščejo cenovno ugodne rešitve informacijskih sistemov, ki pripomorejo k učinkovitejšem nastopanju na čedalje zahtevnejših trgih.

1.2 Namen

Namen diplomske naloge je:

- na splošno predstaviti poslovne aplikacije,
- opisati in primerjati tehnologije in rešitve za razvoj poslovnih aplikacij,
- predstaviti obstoječo rešitev poslovnega procesa,
- definirati zahteve za proces naročanja blaga in storitev in
- sistematično razviti rešitev poslovne aplikacije za proces naročanja artiklov in storitev v javnem zavodu

1.3 Cilji

Cilji diplomske naloge so:

- spoznati celoten postopek razvoja poslovne aplikacije,
- izdelati aplikacijo za izvedbo dela procesa naročanja artiklov in storitev v Psihiatrični kliniki Ljubljana in
- ugotoviti prednosti in slabosti predstavljenih tehnologij in rešitev, uporabljenih pri razvoju poslovne aplikacije.

1.4 Struktura diplomske naloge

V uvodu je predstavljena ideja, namen in cilji diplomske naloge, ter opis strukture po poglavjih.

Drugo poglavje opredeli nekatere pojme, ter splošno opiše in predstavi delitev poslovnih aplikacij glede na funkcionalnost, način in prostor delovanja. Opisana je poslovna aplikacija v diplomski nalogi.

Tretje poglavje podrobno predstavlja metodologije, programske in modelirne jezike, orodja, podatkovne strežnike, tehnologije in programske pakete, ki se lahko uporabljajo pri razvoju in izdelavi poslovnih aplikacij.

V četrtem poglavju je podan opis obstoječega stanja, opis problema in jasno definirane zahteve pri izdelavi aplikacije. Predstavljeni so podatkovni modeli ter diagrami modelirnega jezika UML. Podrobno je opisan postopek izvedbe aplikacije in opis funkcij, ter način uporabe končnega izdelka. Za lažjo predstavbo so dodane slike aplikacije. To poglavje predstavlja jedro diplomske naloge.

Peto poglavje podaja sklepne ugotovitve.

Na koncu so navedeni še uporabljeni viri.

1.5 Predstavitev Psihiatrične klinike Ljubljana

Psihiatrična klinika Ljubljana (v nadaljevanju PKL), v kateri sem zaposlen, je javni zdravstveni zavod, ki opravlja zdravstveno dejavnost na sekundarni in terciarni ravni ter izobraževalno in raziskovalno dejavnost na področju zdravstva.

Ustanovitelj PKL je Republika Slovenija, ustanoviteljske pravice in obveznosti pa izvršuje Vlada Republike Slovenije.

PKL izvaja svoje strokovne funkcije in naloge znotraj zdravstvenega sektorja, sektorja za zdravstveno nego, sektorja za izobraževanje ter sektorja za raziskovalno dejavnost.

Zdravstveni sektor sestavljajo naslednje strokovno organizacijske enote:

- Center za klinično psihiatrijo (CKP),
- Center za mentalno zdravje (CMZ),
- Center za izvenbolnišnično psihiatrijo (CIP) in
- Center za zdravljenje odvisnih od prepovedanih drog (CZOPD).

Strokovne enote zdravstvenega sektorja so lokacijsko razpršene tako, da skupaj z Enoto za zdravljenje odvisnosti od alkohola delujejo na petih različnih lokacijah v Ljubljani. To je tudi eden od vzrokov za vpeljavo spletne aplikacije, ki je predstavljena v diplomski nalogi.

Tiskani interni zahtevki, ki po sedanjem sistemu služijo za interno naročanje blaga in storitev, potujejo med lokacijami preko kurirjev. Dani proces je zelo zamuden, zato bi bili po vpeljavi interne spletne aplikacije, zahtevki dostopni z manjšimi časovnimi zamiki. Tako bi bil sam proces naročanja občutno hitrejši.

2 POSLOVNE APLIKACIJE

2.1 Splošen opis in delitev

Pojem poslovne aplikacije se nanaša na katerokoli aplikacijo, ki je v pomoč pri poslovanju in nastopa v procesih in postopkih povezanih z njim. Poslovne aplikacije lahko razdelimo po funkcionalnosti s poslovnega vidika, lahko jih razvrstimo po načinu in prostoru delovanja. Ta pojem bomo omejil predvsem na elektronsko poslovanje, kjer se delijo poslovne aplikacije na področja B2B (angl. Business 2 Business), B2C (angl. Business 2 Consumer) in B2E (Business 2 Employee). V eno od teh področij spada tudi aplikacija predstavljena v tej diplomski nalogi. [1]

Delitev aplikacij s poslovnega vidika:

- **B2B – aplikacije za medpodjetniško poslovanje**

Te aplikacije se uporabljajo med poslovnimi partnerji. Sprva je bila povezava med temi aplikacijami direktna, kasneje je direktne povezave zamenjal internetni protokol z varnostnimi mehanizmi, npr. navidezna zasebna omrežja (VPN – angl. Virtual Private Network). Kot primer so aplikacije med uvozniki in prodajalci.

- **B2C – aplikacije za poslovanje med podjetjem in kupcem**

Aplikacije te vrste so večinoma namenjene strankam, predvsem končnim kupcem, ki do njih dostopajo preko interneta s pomočjo internetnega brskalnika. Sem spadajo tudi takšne aplikacije, ki jih uporabnik namesti na svoj sistem (uporabniške aplikacije oziroma odjemalci).

- **B2E – aplikacije znotraj podjetja oziroma združbe**

Tovrstne aplikacije se uporabljajo izključno za interne namene (intranet), tako da niso vidne in dostopne zunanjim uporabnikom. Prav tako so lahko realizirane kot namizne aplikacije(odjemalci) ali spletne aplikacije.

Delitev aplikacij po načinu in prostoru delovanja:

- **servisne aplikacije (angl. Service based applications)**

Njihov namen je povezovanje sistemov z drugimi aplikacijami. Primeri servisnih aplikacij so spletni servisi.

- **aplikacije, ki delujejo v ozadju (angl. Back-end applications)**

Po navadi delujejo kot opravila in procesi v ozadju, zato uporabniki nimajo direktnega dostopa do njih.

- **uporabniške aplikacije (angl. Front-end applications)**

Delujejo lahko kot spletne aplikacije z grafičnim vmesnikom. Do njih dostopamo preko spletnega brskalnika, ali kot samostojne namizne aplikacije (odjemalci).

2.2 Poslovna aplikacija v diplomski nalogi

Poslovna aplikacija, obravnavana v diplomski nalogi, bo spadala v skupino B2E aplikacij. Predstavlja interno spletno aplikacijo z grafičnim vmesnikom, ki bo dosegljiva izključno znotraj podjetja. V našem primeru, znotraj javnega zavoda, Psihiatrične klinike Ljubljana.

Nadomestila bo trenutni sistem internega naročanja artiklov in storitev, pri katerem poteka celotni proces izključno v tiskani obliki. Dostopna bo vsem zaposlenim znotraj javnega zavoda. Vsak uporabnik bo imel natančno določeno vlogo in pravice pri uporabi aplikacije. Bistvo aplikacije je naročanje artiklov in storitev. Na voljo bo tudi možnost elektronskega vpogleda v naročene artikle in sledljivost v vseh fazah izvedbe. Od kreiranja zahtevka, do izvedbe dejanskega naročila zelenega artikla oziroma storitve.

3 TEHNOLOGIJE IN ORODJA

3.1 Označevalni in programski jeziki ter orodja za razvoj aplikacije

3.1.1 Označevalni jezik HTML

HTML, oziroma jezik za označevanje nadbesedila, je označevalni jezik za izdelavo spletnih strani. Predstavlja osnovo spletnega dokumenta. S pomočjo HTML-ja ustvarimo strukturo in semantično ureditev dokumenta. [2] Pišemo ga lahko v vsakem preprostem urejevalniku besedil. V nadaljevanju, bo opisano orodje, ki nam pisanje HTML-ja zelo olajša. Razvil ga je Tim Berners-Lee na inštitutu CERN leta 1990. Do danes je doživel že veliko sprememb, a osnovni gradniki jezika so ostali enaki. Obstaja množica spletnih brskalnikov, ki omogočajo prikaz HTML-ja. Najbolj priljubljeni so Internet Explorer, Firefox, Chrome, Safari in Opera.

3.1.2 Programski jezik PHP

PHP je razširjen odprto kodni programski jezik, ki se uporablja za strežniške uporabe oziroma za razvoj dinamičnih spletnih strani. Lahko ga primerjamo z Microsoftovim sistemom ASP (angl. Active Server Pages), VBScript in JScript, Sun Microsystemovim sistemom JSP (angl. Java Server Pages) in Java ter sistemom CGI (angl. Common Gateway Interface) in Perl. Podoben je običajno strukturiranim programskim jezikom, in sicer najbolj jezika C in Perl. Najbolj izkušenim programerjem dovoljuje razvijanje zapletenih uporab brez dolgega učenja. [3]

PHP primarno teče na spletnem strežniku, kjer jemlje PHP izvorno kodo za vhod in generira dinamično spletno stran kot izhod.

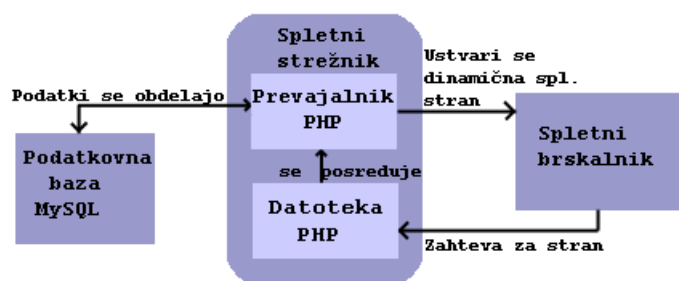
Uporabljamo ga lahko na vseh glavnih operacijskih sistemih: Windows, Linux, Unix, MacOS, RISC OS in drugih.

Uporabo PHP-ja členimo na tri področja:

- izvajanje izvorne kode na strežniku (angl. server-side scripting),
- razvoj namiznih aplikacij (angl. writing desktop applications) in
- izvajanje ukazov v ukazni vrstici (angl. command line scripting).

Pomembna lastnost PHP-ja je podpora številnim podatkovnim bazam. V današnjem času je najpogostejša povezava PHP-ja z MySQL podatkovno bazo (slika 1). PHP podpira povezave z vsemi pomembnimi podatkovnimi bazami: dBase, Empress, FilePro, Hyperwave, DB2, Ingres, FrontBase, MS-SQL, ODBC (angl. Open Database Connectivity), Oracle, PostgreSQL, SQLite, Sybase itd.

PHP ima svoje slabe lastnosti, ki so v glavnem povezane z njegovo sintakso. Zaradi nestrukturiranosti postanejo programi v PHP-ju težko berljivi in obvladljivi, celo za samega avtorja programa.



Slika 1: Prikaz povezave med PHP in MySQL

3.1.3 Programski jezik JavaScript

JavaScript je objektni skriptni programski jezik, ki ga je ustvaril Netscape. Namen je, da pomaga spletnim programerjem pri ustvarjanju interaktivnih spletnih strani. Razvit je bil neodvisno od Java, vendar si z njo deli številne lastnosti in strukture. Sodeluje lahko s HTML kodo in s tem poživi stran z dinamičnim izvajanjem. Podpirajo ga vsi novejši spletni brskalniki.

JavaScript se veliko uporablja za ustvarjanje dinamičnih spletnih strani. Program se vgradi ali vključi v HTML z namenom, da opravlja naloge, ki niso mogoče s samo statično stranjo. Npr. odpiranje novih oken, preverjanje pravilnosti vnesenih podatkov v obrazcih pred vnosom na strežnik, enostavni izračuni, spreminjanje slik, ko zapeljemo miškin kazalec čez njo, itd. Na žalost različni spletni brskalniki izpostavijo različne objekte za uporabo. Za podporo vseh brskalnikov je zato potrebno napisati več različic funkcij. [4]

3.1.4 Orodje za spletno programiranje Adobe Dreamweaver

Adobe Dreamweaver je orodje za spletno programiranje. Razvito je v podjetju Allaire Systems, ki ga je Macromedia pridobila okoli leta 1998, in je zdaj v lasti Adobe Systems, ki je kupil Macromedio leta 2005. [5]

Dreamweaver je na voljo za MacOS in Windows operacijske sisteme. Nedavne različice vključujejo podporo za spletne tehnologije, kot so CSS, JavaScript, ter različne strežniške skriptne jezike in okvirje, vključno z ASP.NET, ColdFusion, JSP in PHP. Zaradi široke podpore spletnim tehnologijam, je Dreamweaver priljubljeno orodje med spletnimi programerji.

3.1.5 Orodje MAMP

MAMP je brezplačno orodje za izvajanje dinamičnih spletnih strani na računalnikih z operacijskim sistemom MacOS. Kot že ime pove, teče na MacOS-u, vsebuje spletni strežnik Apache, podatkovni strežnik MySQL in podpira izvajanje PHP, Perl in Python programskih jezikov za spletno razvijanje. Namestitev orodja je enostavna in hitra. [6]

Z njegovo brezplačno uporabo odpravimo potrebo po samostojnih strežnikih, saj v enem paketu dobimo popolno orodje za izvajanje in razvijanje dinamičnih spletnih strani na lokalnem računalniku.

Podobni paketi so na razpolago tudi za druge operacijske sisteme, WAMP za Windows ter LAMP za Linux.

3.2 Podatkovni strežniki in orodja za administracijo podatkovnih baz

3.2.1 Sistem za upravljanje podatkovne baze MySQL

MySQL je večnitni, večuporabniški, relacijski podatkovni upravljalni sistem. Na voljo je odprtokodna različica in komercialna verzija. Odlikujeta jo hitrost in fleksibilnost ter standardiziran jezik za povpraševanje in izdelavo poizvedb SQL (angl. Structured Query Language) [7].

Obstaja množica API vmesnikov, napisanih v različnih programskih jezikih, ki omogočajo dostop do MySQL podatkovnih baz. C, C++, C#, Delphi, Java, Lisp, Perl, PHP, Python, Ruby, FreeBasic, Tcl in drugi. Vsak od teh programskih jezikov uporablja svoj vmesnik API. Poleg tega obstaja še MyODBC vmesnik, ki omogoča drugim programskim jezikom, ki podpirajo ODBC vmesnike, povezavo z MySQL podatkovno bazo. Primer sta ASP in Coldfusion.

MySQL je najbolj primeren za uporabo pri manjših podatkovnih bazah. Danes se najbolj uporablja za implementacijo spletnih aplikacij v povezavi s prav tako popularnim PHP programskim jezikom. Za njiju se je uveljavil vzdevek »dinamični duet« (angl. dynamic duo). Razvoj aplikacij in servisov v PHP/MySQL arhitekturi je preprost, učinkovit, hiter ter odlično podprt z dokumentacijo na internetu.

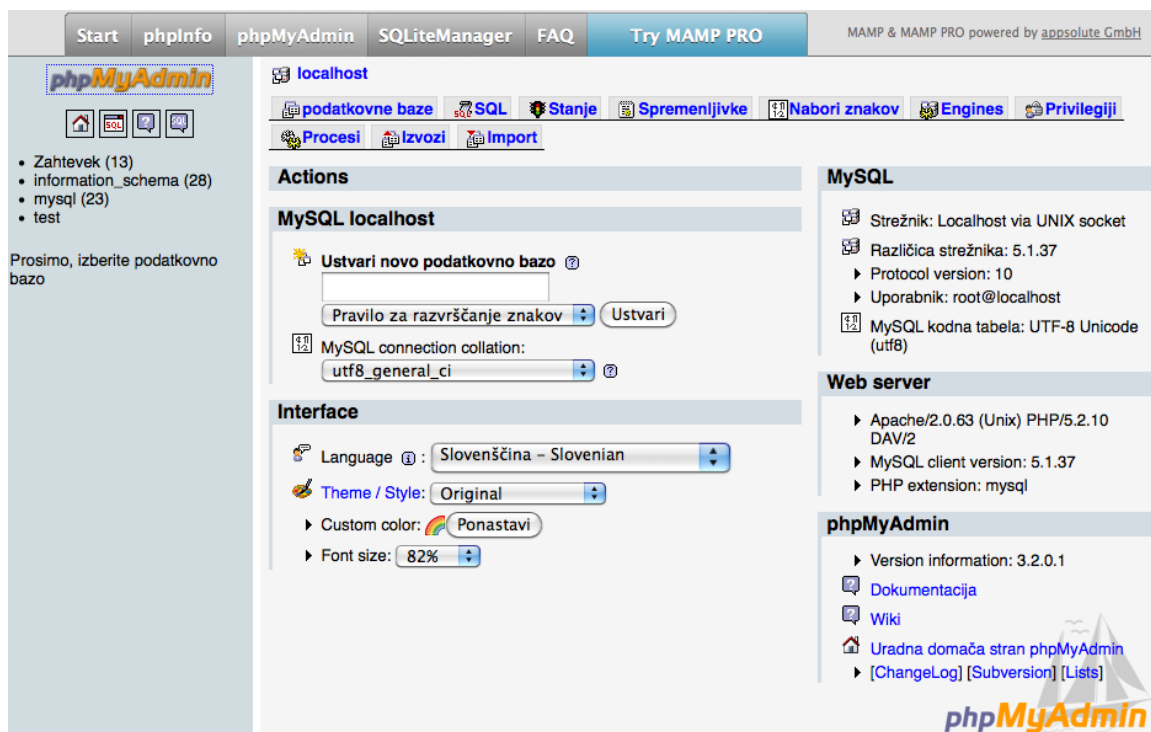
Za administracijo MySQL-a so na voljo tako orodja za izvajanje iz ukazne vrstice (ukaza: `mysql` in `mysqladmin`) kot tudi okenske aplikacije z grafičnim uporabniškim vmesnikom (MySQL Administrator in MySQL Query Browser). Zelo razširjena in popularna alternativa, napisana v PHP-ju, je odprtokodna spletna aplikacija phpMyAdmin, ki jo bomo opisali v naslednjem podpoglavju.

3.2.2 Orodje PhpMyAdmin

PhpMyAdmin (slika 2) je orodje napisano v programskem jeziku PHP. Namenjeno je administraciji MySQL podatkovnega strežnika preko spletnega brskalnika. Odlikuje ga odličen grafični uporabniški vmesnik. Na voljo je v 57 jezikih, med drugim tudi v slovenščini. [8]

Naloge, ki jih opravlja so:

- ustvarjanje, spreminjanje in odstranjevanje podatkovnih baz, tabel, polj in vrstic,
- izvajanje, urejanje in shranjevanje SQL stavkov in skript,
- upravljanje uporabnikov MySQL strežnika ter pravic na njem,
- uvažanje podatkov iz CSV in SQL ter izvažanje v različne formate: CSV, SQL, XML, PDF, DOC, XLS, LATEX itd.,
- globalno iskanje v podatkovni bazi MySQL itd.



Slika 2: Grafični uporabniški vmesnik phpMyAdmin

3.3 Metodologije razvoja in orodja za načrtovanje

3.3.1 Metodologija Rational Unified Process (RUP)

RUP je iterativna razvojna metodologija, razvita pri podjetju Rational Software Corporation, katerega je leta 2003 prevzelo podjetje IBM. [9]

RUP je primer procesa, ki:

- daje smernice za učinkovit razvoj kakovostne programske opreme,
- zmanjšuje tveganje in povečuje predvidljivost,
- zajema in vpeljuje najboljše izkušnje,
- pospešuje skupno vizijo,
- vpeljuje načrt za uvedbo orodij in
- omogoča enostaven in hiter dostop do informacij v elektronski obliki.

RUP opisuje, kako učinkovito uporabiti šest najboljših izkušenj s področja razvoja programske opreme:

- iterativni razvoj,
- obvladovanje zahtev,
- uporaba komponentne arhitekture,
- vizualno modeliranje,

- preverjanje kakovosti in
- nadzorovanje sprememb.

RUP zajema štiri faze življenjskega cikla programske opreme:

- začetna faza – vzpostavitev projekta, opredelitev okvirjev obravnavanega področja, načrtovanje virov, ipd.,
- zbiranje informacij – o obravnavanem področju, specifikacija značilnosti, načrtovanje arhitekture,
- konstrukcija izdelka in
- prevzem – predaja izdelka v uporabo končnemu uporabniku.

Vsaka faza življenjskega cikla razvoja je sestavljena iz postopkov, kjer se tudi izoblikujejo specifični modeli. Ti bodo podrobno predstavljeni v nadaljevanju:

- poslovno modeliranje – primeri uporabe poslovnega okolja in konceptualni model poslovnega okolja,
- zajem zahtev – model primerov uporabe,
- analiza in načrtovanje – model načrta,
- izvedba – model izvedbe in
- testiranje – model testiranja.

Metodologijo RUP smo izbrali zato, ker smo jo spoznali na predavanjih in vajah na Fakulteti za računalništvo in informatiko. Tako je bil čas za razvoj aplikacije po tej metodologiji občutno krajši. Metodologija RUP je svetovno uveljavljena pri modeliranju in razvoju aplikacij, zato se nam njena uporaba zdi smiselna.

3.3.2 Modelirni jezik Unified Modeling Language (UML)

UML je jezik za specifikacijo, vizualizacijo, konstrukcijo in dokumentacijo izdelkov objektno-orientiranega programske-intenzivnega sistema. Predstavlja zbirko tehnik, ki so se izkazale za uspešne pri modeliranju velikih in kompleksnih sistemov. UML uporablja predvsem grafične zapise, ki prikažejo načrtovanje programske opreme projektov. Uporaba UML pripomore tudi k lažji komunikaciji med razvijalci in projektnimi skupinami, pri raziskavi možne različice načrtov in overitev arhitekturne zasnove programske opreme oziroma sistema v razvoju. [10]

Cilji razvoja z UML-jem so:

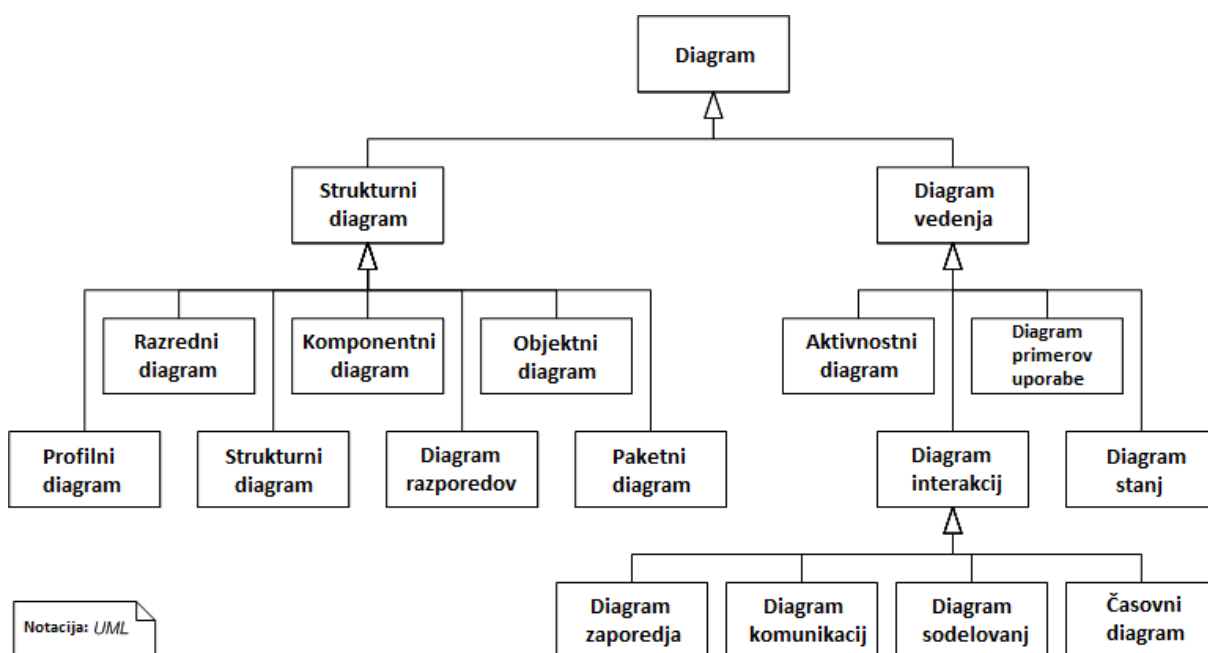
- zagotoviti uporabnikom enoten jezik za vizualno modeliranje, razvoj modelov in izmenjevanje le-teh med seboj,
- zagotoviti razširljivost in specializacijo mehanizmov za razširitev ključnih pojmov,
- ponuditi formalno podlago za razumevanje modelirnega jezika,

- podpirati visoko-nivojske razvojne koncepte kot so sodelovanje, ogrodja, vzorci in komponente in
- vključevanje najboljših praks.

UML loči tri skupine modelov in z njimi povezane diagramske tehnike:

- funkcijski model – prikaže funkcionalnost sistema z uporabnikovega vidika in vsebuje diagram primerov uporabe,
- objektni model – prikaže strukturo in podstrukture sistema in vsebuje razredne diagrame in
- dinamični model – prikaže notranje delovanje sistema in vsebuje diagrame zaporedja, sodelovanja in stanja.

UML pozna 14 različnih tipov diagramov, ki so razdeljeni v dve kategoriji. Sedem diagramov predstavlja strukturno delitev, drugih sedem predstavlja obnašanje sistema med katere spadajo štiri diagrami interakcij. Kako so kategorizirani, je razvidno na naslednjem razrednem diagramu (slika 3).



Slika 3: Hierarhično kategorizirani UML diagrami

3.3.3 Orodje Rational Rose

Rational Rose je objektno orientirano orodje za modeliranje, ki temelji na UML. Drži in podpira metodologijo RUP. Diagrami, ki jih zajema UML, opisujejo strukturo in obnašanje objektno usmerjenega sistema. Rational Rose ponuja mnogo orodij za razvoj modelov in diagramov za dokumentacijo modelov in izvoz oziroma oblikovanje v izvorno kodo. Dve priljubljeni funkciji Rational Rose-a sta sposobnost iterativnega (evolucijskega) razvoja ter povratnega inženirstva. Iterativni razvoj omogoča to, da se aplikacijo razvija po fazah in pomeni zaključek ene faze, začetek naslednje faze. Nato začne razvijalec razumevati, kako komponente med seboj sodelujejo in to omogoča spremembe v načrtovanju. Rose nato s

funkcijo povratnega inženirstva posodobi ostale modele ter zagotavlja skladnost programske kode. [11]

Diagrami za modeliranje, ki jih Rose podpira so: [12]

- diagrami primerov uporabe,
- diagrami interakcij,
 - diagrami zaporedja,
 - diagrami sodelovanja,
- razredni diagrami,
- diagrami stanj,
- komponentni diagrami in
- postavitveni diagrami.

Uporabljeni diagrami bodo predstavljeni v naslednjem poglavju diplomske naloge.

3.3.4 Orodje Power Designer

Power Designer je orodje za grafično modeliranje podatkovnih baz. Uporablja različne standardizirane tehnike modeliranja (UML) za zagotavljanje poslovnih analiz in formalnega načrtovanja podatkovnih baz. V diplomski nalogi je uporabljen za izdelavo diagramov »entiteta-razmerje«. [13]

Za diagrame »entiteta-razmerje«, se v Power Designer-ju uporabljajo naslednji simboli (slika 4):



Slika 4: Simboli diagrama »entiteta-razmerje« v Power Designer-ju

Modelirne tehnike, ki jih Power Designer podpira:

- podatkovno modeliranje – modeliranje podatkovnih baz (konceptualni, logični in fizični podatkovni model),
- aplikativno modeliranje – vse UML diagramske tehnike z objektno/relacijsko usmerjenim modeliranjem in implementacijo aplikacij in
- modeliranje poslovnih procesov – podpora spoznavanja in izdelave diagramov za izvedbo procesov.

Pristojne osebe za izpolnjevanje internih zahtevkov in ugotavljanje potreb po posameznem artiklu ali storitvi so:

Predstojnik organizacijske enote (OE) CKP za vsa stroškovna mesta OE:

- za osnovna sredstva za potrebe OE,
- za servisiranje in nadomestne dele za nemedicinsko opremo OE,
- za tekoče in investicijsko vzdrževanje opreme in prostorov OE,
- za vse storitve za potrebe OE, razen tistih za katere je s predmetnim navodilo izrecno določeno drugače,
- za delovne obleke zaposlenih v OE razen za zaposlene v zdravstveni negi in
- za drobni inventar in potrošni material za potrebe OE, razen tistega za katerega je s predmetnim navodilom izrecno določeno drugače.

Prav tako je za OE CMZ pristojen predstojnik OE CMZ za vsa stroškovna mesta razen EZOA, vodja EZOA za stroškovna mesta EZOA in alkohološke ambulante, predstojnik OE CZOPD za stroškovna mesta OE CZOPD ter prav tako predstojnik OE CIP, vsi za enako blago oziroma opremo in storitve, navedene predhodno pri predstojniku OE CKP. Pristojne osebe, vsaka za določeno področje so še: pomočnik strokovnega direktorja PKL za zdravstveno nego, pomočnik strokovnega direktorja PKL za izobraževalno dejavnost, vodja službe za preprečevanje samomora, vodja lekarne, pomočnik generalnega direktorja PKL za upravne zadeve, pomočnik generalnega direktorja PKL za poslovne zadeve, strokovni direktor PKL ter generalni direktor PKL.

V obrazcu interni zahtevek mora pristojna oseba opredeliti najmanj:

- stroškovno mesto (z nazivom),
- oddelek ali službo (kadar ne gre za samostojno stroškovno mesto),
- naziv blaga ali storitve z mersko enoto in količino,
- datum,
- čitljivo ime in priimek pooblaščenca osebe in
- podpis pooblaščenca osebe.

Vsaka od pristojnih oseb s pisnim pooblastilom prenese svoje pristojnosti po tem navodilu na največ dve osebi z najmanj VI. stopnjo izobrazbe in sicer tako, da je prvo-imenovani pooblaščenec stalno pooblaščen, drugo-imenovani pooblaščenec je pooblaščen zgolj v primeru odsotnosti prvo-pooblaščenega.

Vsak podpisan interni zahtevek se posreduje vodji oddelka na nabave.

Vodja oddelka za nabave po prejemu ustrezno podpisanega internega zahtevka preveri ali je možno naročilo izpolniti na podlagi že sklenjenega pravnega posla oziroma z delom zaposlenih v PKL in interni zahtevek posreduje v izvedbo pristojni osebi.

Če naročila ni mogoče izpolniti na podlagi že sklenjenega pravnega posla ali z delom zaposlenih v PKL, se izvede ustrezen postopek oddaje javnega naročila, skladno z določbami posebnega pravilnika.

Proces naročanja blaga oziroma storitev je opisan v naslednjem podpoglavju.

4.2 Opis problema

Zaposleni na PKL izpolni interni zahtevek in ga odda v pregled in podpis pristojni osebi. Le-ta ga pregleda in v primeru, da se strinja z vsebino zahtevka, ga podpiše. Nato zahtevek nadaljuje pot v oddelek za nabavo, kjer ga enolično označijo in preverijo podpisnika. V primeru, da je na zahtevku naročen potrošni material, le-ta potuje v skladišče. Tam se material izda, če je le-ta na zalogi, drugače oddelek za nabavo izbere ustrezen način naročila materiala. Ko ne gre za potrošni material, potuje zahtevek v oddelek za plan in analize. Tam pristojna oseba preveri in označi, ali je predmet naročila v planu in ali so zagotovljena denarna sredstva za izvedbo naročila. Nato potuje zahtevek nazaj v oddelek za nabavo. V primeru, da je v planu in so sredstva zagotovljena, preveri in izbere način izvedbe naročila. Možni načini izvedbe naročila so:

- javno naročilo,
- javno naročilo male vrednosti (pogodba),
- javno naročilo male vrednosti (pogodba stalna),
- javno naročilo male vrednosti (naročilnica),
- javno naročilo male vrednosti (naročilnica-enostavna) in
- javno naročilo male vrednosti (gotovina).

V primeru, da zahtevek ni v planu, potuje direktno k generalnemu direktorju PKL-ja. Poleg je potrebno priložiti še obrazložitev ter morebitne priloge (strokovne zahteve, prospekti, sezname). Če direktor ne da soglasja, se postopek naročila zaključi, pristojni osebi, ki je oddala zahtevek se sporoči odločitev, oziroma obrazložitev o zaključku postopka. V primeru odobritve gre interni zahtevek nazaj v oddelek za nabavo, ki nadaljuje že prej opisan postopek o načinu izvedbe naročila.

4.3 Zajem zahtev

Zdaj smo ugotovili obstoječe stanje pri naročanju blaga in storitev v PKL in se seznanili s postopkom pri izvedbi naročila. Po zajemu zahtev za aplikacijo (spletna aplikacija za interno naročanje) se bomo lotili razvoja aplikacije po metodologiji RUP.

Problemska domena obsega delni razvoj spletne interne aplikacije za uporabo znotraj našega javnega zavoda PKL.

4.3.1 Akterji sistema

V aplikaciji so definirane 3 vrste uporabnikov, ki se med seboj razlikujejo po obsegu pravic za uporabo funkcij, ki jih lahko uporabljajo v aplikaciji.

Akterji sistema so:

1. administrator – uporabnik, ki ima po uspešni prijavi v aplikacijo dostop do vseh funkcij in pregled nad celotnim sistemom (celotna funkcionalnost),
2. pooblaščen oseb – uporabnik, ki lahko po uspešni prijavi v aplikacijo pooblasti zahtevek iz pristojne OE, in ga odda. Zahtevek lahko kreira in odda (brez odobritev drugih oseb),
3. uporabnik - uporabnik, ki lahko po uspešni prijavi v aplikacijo kreira zahtevek, ki potem čaka na pregled pooblaščen oseb. Na voljo ima pregled vseh svojih oddanih zahtevkov.
4. tiskalnik – naprava namenjena tiskanju internih zahtevkov, ki pa glede na uporabnost, dostopnost in varnost aplikacije ter ekološke smernice ni zaželeno.

4.3.2 Primeri uporabe

1. Prijava v aplikacijo

Na začetku uporabnika pričaka stran z poljema za vnos uporabniškega imena in gesla. Oboje je administrator predhodno dodelil. V primeru pravilnega vnosa obeh postavk se uporabniku prikaže stran z uporabniškim menijem.

2. Odjava iz aplikacije

Po končanem delu ima uporabnik možnost odjave iz aplikacije. S tem onemogoči uporabo aplikacije drugi osebi v njegovem imenu.

3. Urejanje lastnih podatkov

Uporabniku aplikacije omogoča spreminjanje lastnih podatkov.

4. Pregled uporabnikov

Omogoča pregled uporabnikov aplikacije osebam s statusom pooblaščen oseb ali administratorja.

5. Pregled podrobnosti o uporabnikih

Razširja uporabo pregleda uporabnikov in omogoča pooblaščenim osebam in administratorju pregled podrobnosti o uporabnikih.

6. Dodajanje uporabnika

Omogoča administratorju dodajanje uporabnikov v aplikacijo.

7. Brisanje uporabnika

Omogoča administratorju brisanje uporabnikov iz aplikacije.

8. Urejanje podatkov o uporabnikih

Administrator lahko tudi ureja podatke vseh uporabnikov.

9. Kreiranje internega zahtevka

Vsi uporabniki imajo možnost kreiranja internih zahtevkov.

10. Oddaja internega zahtevka

Interni zahtevek lahko oddajo le pooblašcene osebe in administratorji.

11. Pregled internega zahtevka

Uporabnik na spletni strani vidi le lastne kreirane interne zahtevke, pooblašcana oseba vidi lastne interne zahtevke in vse zahtevke vezane na oddelek za katerega je pristojna, administrator ima vpogled v vse interne zahtevke.

12. Pregled podrobnosti internega zahtevka

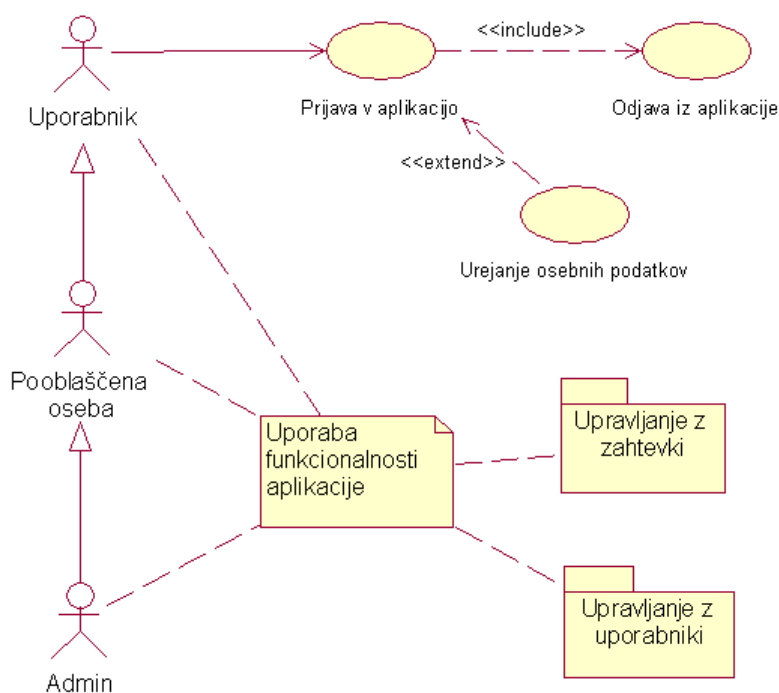
To je primer uporabe, ki razširja uporabo pregleda internega zahtevka. Pravice so definirane na enak način kot pri pregledu internega zahtevka.

13. Tiskanje internega zahtevka

Razširja pregled podrobnosti internega zahtevka tako, da omogoča njegovo tiskanje.

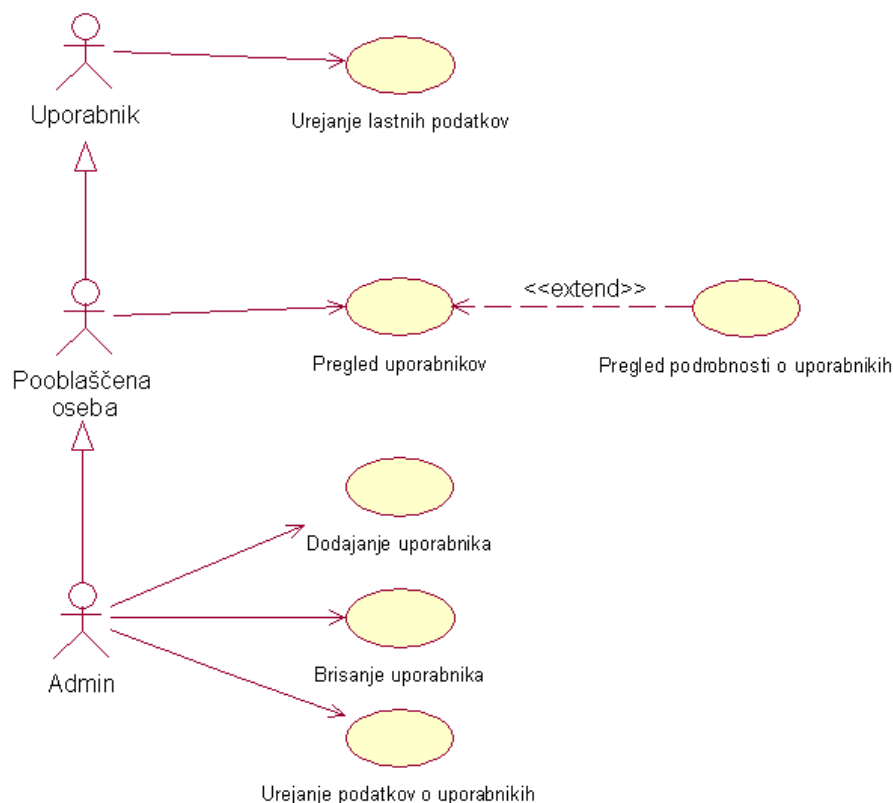
4.4 Modeli primerov uporabe

Modeli primerov uporabe prikazujejo možne primere uporabe aplikacije. Prvi model prikazuje primere uporabe za celotno aplikacijo (slika 5). Ta vsebuje prijavo v aplikacijo, urejanje osebnih podatkov, uporabo paketa za upravljanje z zahtevki in paketa za upravljanje z uporabniki. Odjava iz aplikacije je možna le ob predhodni prijavi. Modeli primerov uporabe so realizirani v programskem paketu Rational Rose.

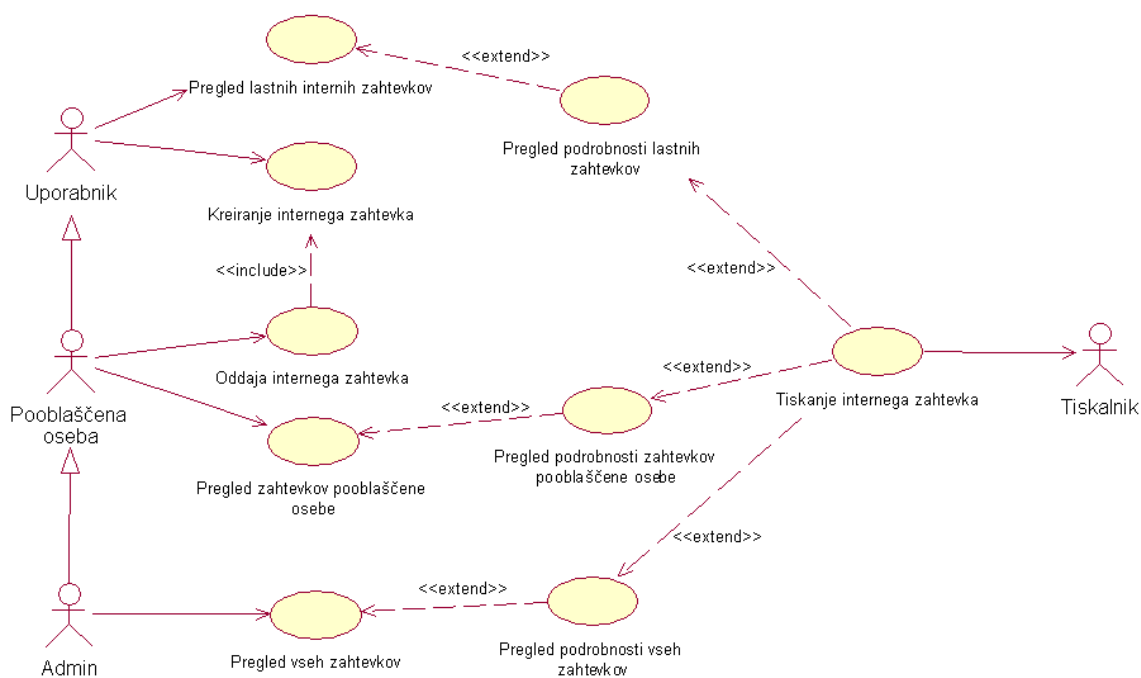


Slika 5: Model primerov uporabe za celotno aplikacijo

V nadaljevanju sta prikazana modela primerov uporabe za paket upravljanja z zahtevki in paket upravljanja z uporabniki (slika 6, 7).



Slika 6: Model primerov uporabe za upravljanje z uporabniki



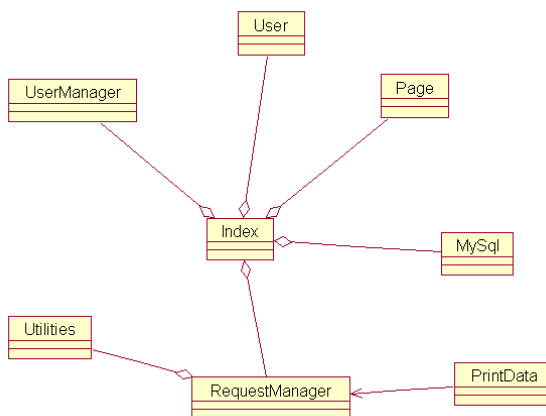
Slika 7: Model primerov uporabe za upravljanje z zahtevki

4.5 Procesni model

Izdelava procesnega modela je naslednja faza razvoja po metodologiji RUP. Izdelali smo razredne diagrame ter diagrame zaporedja. V nadaljevanju podpoglavja so nekateri od teh diagramov prikazani in opisani, vse smo realizirali v programskem orodju Rational Rose.

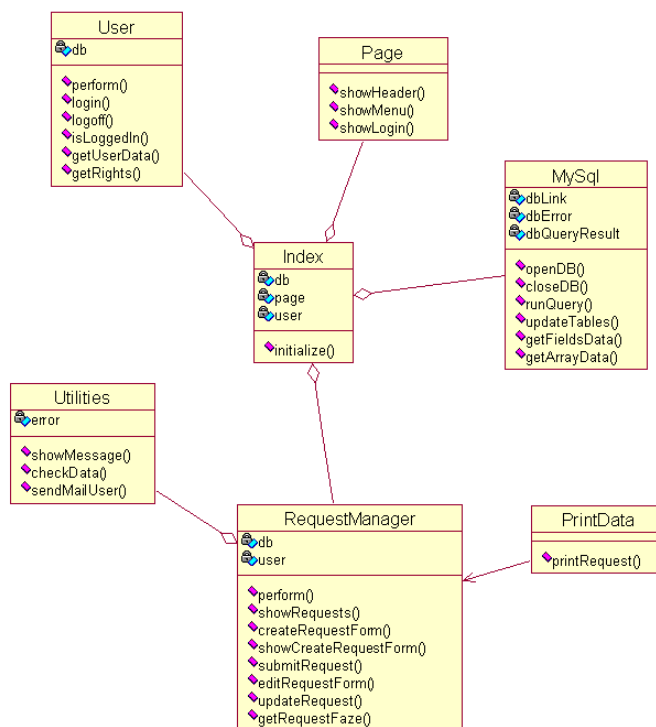
4.5.1 Razredni diagrame

Naslednji diagram prikazuje vse uporabljene razrede v naši aplikaciji ter način njihove uporabe (slika 8).



Slika 8: Razredni diagram s prikazom razredov za celotno aplikacijo

V naslednjem diagramu vidimo razrede aplikacije, za primer uporabe oddaje internega zahtevka. Vidni so tudi njihovi atributi in operacije (slika 9).



Slika 9: Razredni diagram s prikazom razredov, atributov in funkcij za oddajo internega zahtevka

4.5.2 Diagrami zaporedja

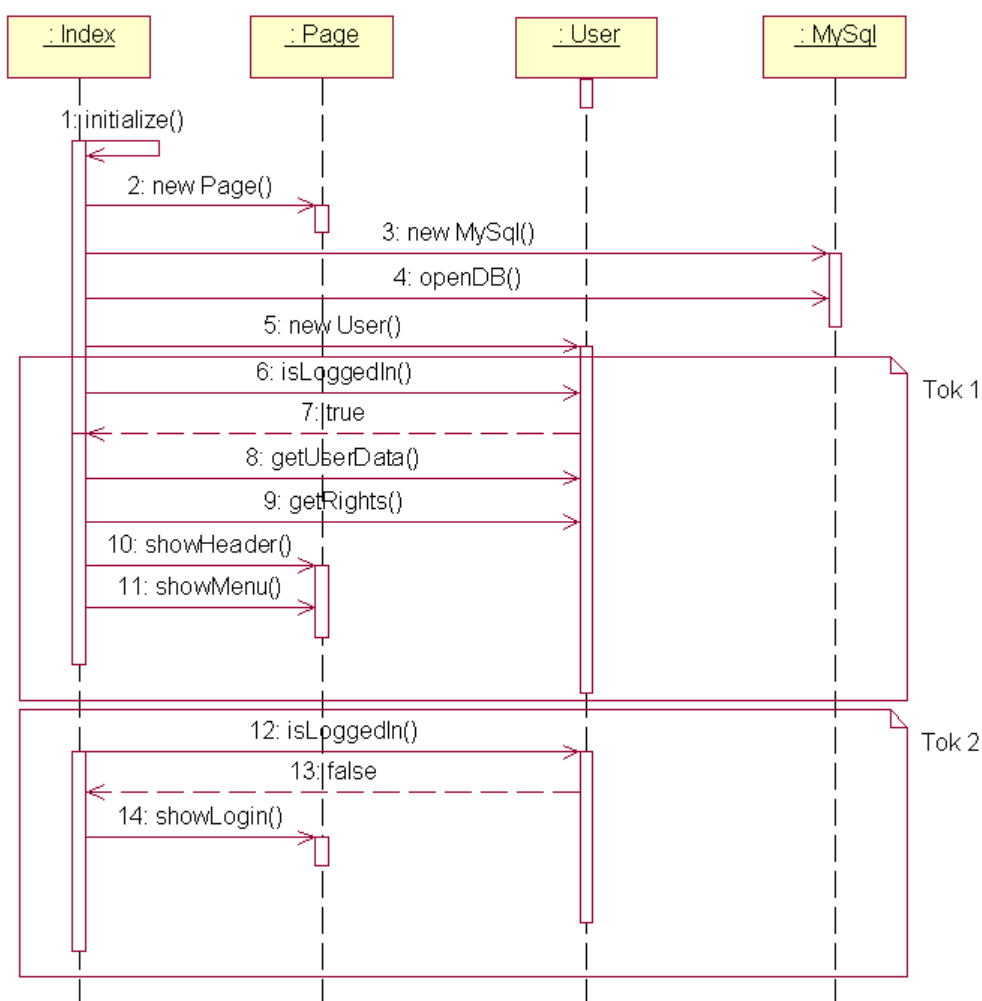
Prvi diagram zaporedja prikazuje zaporedje prikaza prve strani (slika 10). Objekt `Index` se inicializira s svojo metodo `initialize()`. Nato z metodo `Page()` ustvari nov objekt `Page` in z metodo `MySQL()` nov objekt za delo z podatkovno bazo `MySQL`. Z metodo `openDB()` kreira povezavo z bazo. Od tu naprej sta možna dva alternativna toka.

Prvi tok (alternativa 1)

Metoda `isLoggedIn()` ugotovi, da je uporabnik prijavljen v aplikacijo. Z metodama `getUserData()` in `getRights()` pridobi podatke o prijavljenem uporabniku. Vzporedno pridobi tudi nivo njegovih pravic. Na objektu `Page` se prikaže glava in uporabniku prilagojen meni. Za to poskrbita metodi `showHeader()` ter `showMenu()`.

Drugi tok (alternativa 2)

Metoda `isLoggedIn()` ugotovi, da uporabnik ni prijavljen v aplikacijo. Objekt `Index` nato prikaže obrazec za prijavo s pomočjo funkcije `showLogin()`, objekta `Page`.



Slika 10: Diagram zaporedja za prikaz prve strani

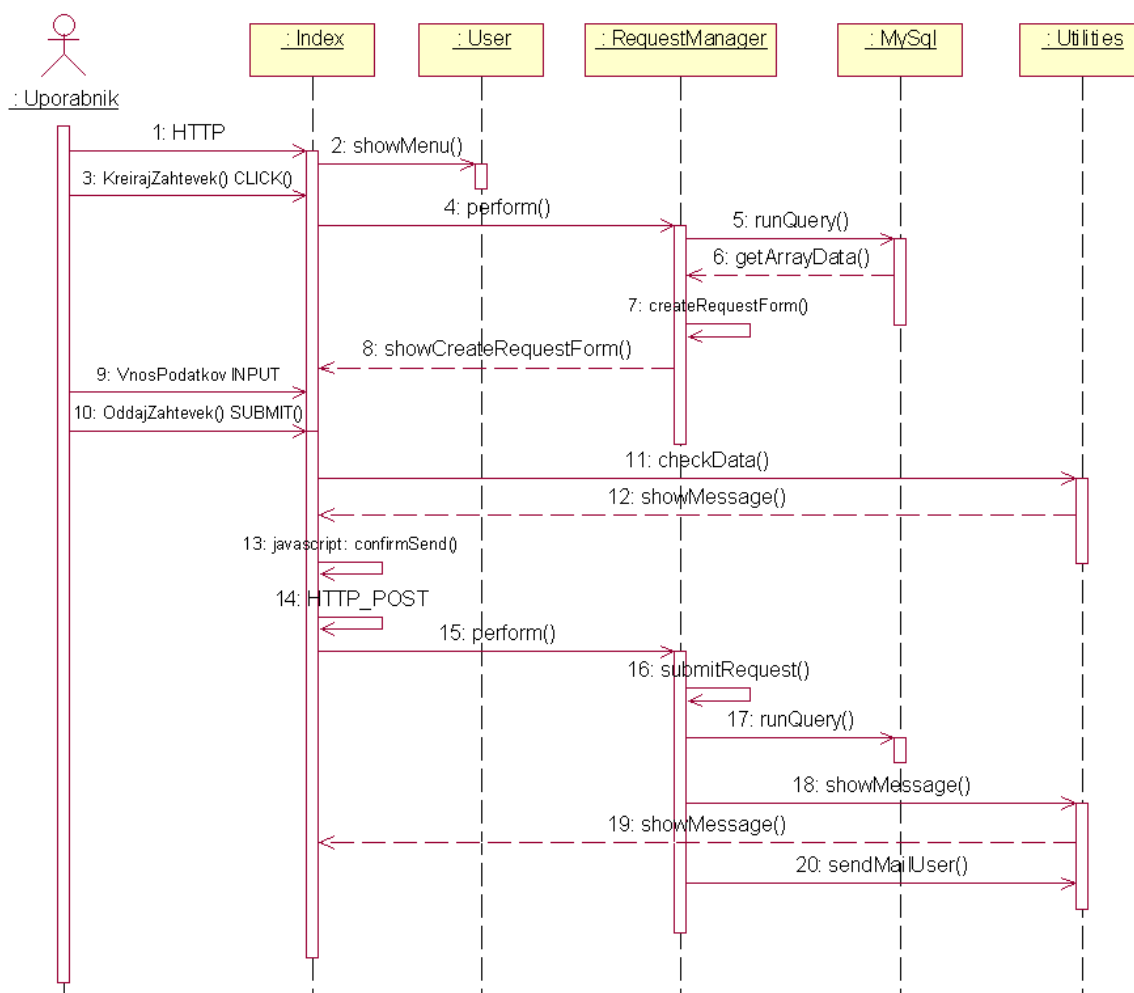
Drugi diagram zaporedja pa prikazuje zaporedje kreiranja internega zahtevka (slika 11). Uporabnik preko spletnega brskalnika, ki uporablja `http` protokol, dostopa do intranetne

strani in objekta `Index`. `Index` nato pokliče metodo `showMenu()` objekta `User`, ki na strani prikaže uporabniški meni.

Uporabnik klikne na povezavo `Kreiraj Zahtevek`, to ustvari objekt `RequestManager` in pokliče njegovo metodo `perform()`, za inicializacijo. Objekt `RequestManager` potem sproži metodo `runQuery()`, ki ob kreiranju objekta `MySQL` naredi poizvedbo, ter z metodo `getArrayData()` vrne ustrezne podatke iz podatkovne baze. `RequestManager` s pomočjo teh podatkov in svoje metode `createRequestForm()`, ustvari obrazec za vnos podatkov internega zahtevka. Metoda `showCreateRequestForm()`, ga prikaže na uporabnikovi strani.

Ko uporabnik izpolni obrazec in klikne povezavo `Oddaj zahtevek`, se sproži metoda `checkData()`, objekta `Utilities`, ki preveri vnesene podatke. Če je vse pravilno in uporabnik preko funkcije `confirmSend()` potrdi oddajo zahtevka, se podatki oddajo s pomočjo metod `submitRequest()` in `runQuery()`.

Objekt `RequestManager` sproži še metodo `showMessage()`, iz objekta `Utilities`, ki na strani prikaže uporabniku katere podatke je oddal. Nato sproži še metodo `sendMailUser()`, ki pošlje uporabniku in naslednjemu uporabniku v procesu elektronsko sporočilo o oddaji internega zahtevka.



Slika 11: Diagram zaporedja za kreiranje internega zahtevka

2. Letni Plan, Konto, Faza Izvedbe, Priloge, Nabava, Nabava JN

Entitetni tipi Letni_Plan, Konto, Faza_Izvedbe, Nabava in Nabava_JN vsebujejo enolični identifikator (ID_LP, ST_Konta, ID_Faze, ID_Nabave in ID_Nabave_JN), ter atribut Opis. Vsi predstavljajo šifrante. Entitetni tip Priloge vsebuje poleg identifikatorja (ID_Prilog), še attribute Strokovne_Zahteve, Sezname in Prospekti, ter prav tako služi kot šifrant.

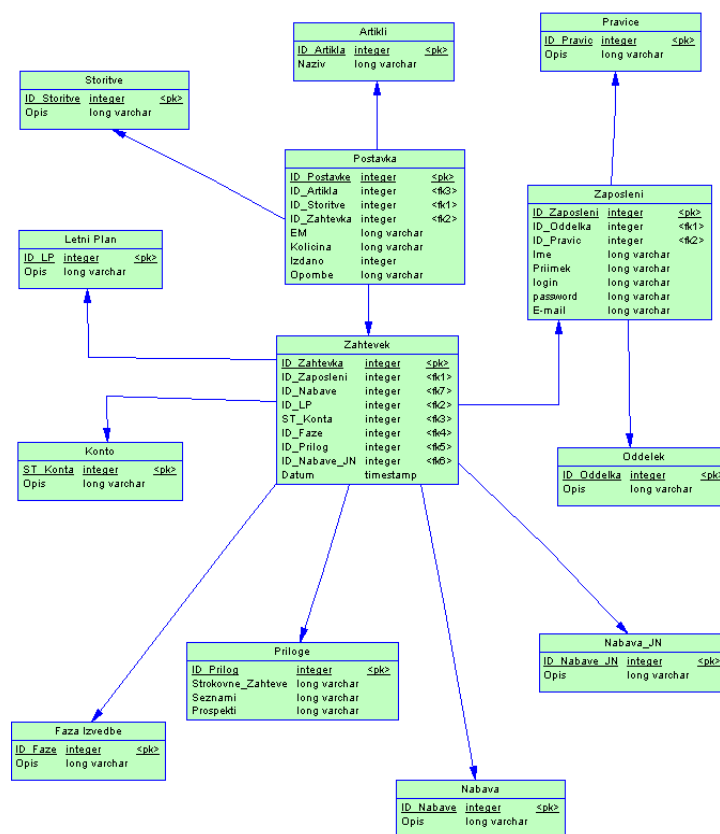
3. Postavka, Storitve, Artikli

Entitetni tip Postavka, poleg enoličnega identifikatorja ID_Postavke, vsebuje še tuji ključ ID_Zahtevka, ker lahko vsebuje en zahtevk mnogo postavk. Preostali atributi so EM (enota mere), Kolicina, Izdano ter Opombe. Na entitetni tip Postavka sta vezana še dva entitetna tipa. To sta Storitve in Artikli, ki služita kot šifranta. Oba vsebujeta enolične identifikatorje (ID_Storitve in ID_Artikla), ter atributa Opis oziroma Naziv.

4. Zaposleni, Oddelek, Pravice

Aplikacijo uporabljajo zaposleni, ki so realizirani s pomočjo entitetnega tipa Zaposleni. Le-ta vsebuje enolični identifikator ID_Zaposleni ter attribute Ime, Priimek, login, password, E-mail. Preko tujih ključev (ID_Oddelka in ID_Pravic), iz entitetnih tipov Oddelek in Pravice, ugotovimo v kateri oddelek spada zaposleni, ter katero vrsto pravic ima.

4.6.2 Fizični podatkovni model



Slika 13: Fizični podatkovni model

Na fizičnem podatkovnem modelu so poleg primarnih ključev, za boljši pregled, prikazani tudi tuji ključ. To ni vidno na konceptualnem podatkovnem modelu. Poudarjeni so tudi tipi atributov.

4.7 Opis aplikacije

V naslednjem poglavju bomo predstavili razvito aplikacijo. Spoznali bomo njen namen, funkcionalnosti, uporabniški vmesnik, ter možnosti za nadaljnji razvoj in uporabo v realnem okolju.

4.7.1 Splošno o aplikaciji

Aplikacija je namenjena uporabi zaposlenih v PKL, namesto obstoječega procesa internega naročanja blaga in storitev. Ta trenutno poteka izključno v tiskani obliki. Razvita je tako, da teče na spletnem strežniku in je dostopna samo znotraj PKL. Njen primarni namen je kreiranje in oddajanje internih zahtevkov v elektronski obliki. Poleg tega, omogoča uporabnikom z ustreznimi pravicami tudi pregled, urejanje in brisanje uporabnikov, ter pregled oddanih zahtevkov z fazami v kateri se trenutno nahajajo. Tistim uporabnikom, ki sodelujejo v procesu internega naročanja (vodja oddelka za nabavo, vodja oddelka za plan in analize, direktorja itd.), omogoča urejanje in dopolnjevanje specifičnih delov internega zahtevka, za katere imajo pooblastila.

4.7.2 Uporabniški meni

Uporabniku se na prvi strani prikaže obrazec za prijavo v aplikacijo (slika 14). Za uporabnike in njihove podatke smo uporabili obstoječo bazo uporabnikov, ki imajo v PKL dodeljeno elektronsko pošto. Dodajanje novih uporabnikov si bomo ogledali kasneje.



PSIHIATRIČNA KLINIKA LJUBLJANA
UNIVERSITY PSYCHIATRIC HOSPITAL

PRIJAVA UPORABNIKA

Uporabniško ime	<input type="text"/>
Geslo	<input type="password"/>
<input type="button" value="Prijava se"/>	

Slika 14: Obrazec za prijavo

Po uspešni prijavi v aplikacijo, se uporabniku prikaže uporabniški meni (slika 15). Uporabniški meni vsebuje glavo z osnovnimi podatki o uporabniku ter ukazni meni.



PSIHIATRIČNA KLINIKA LJUBLJANA
UNIVERSITY PSYCHIATRIC HOSPITAL

Pozdravljen/-a, Josip Varga, Admin

[Moj Profil](#) | [Odjava](#)

Interni zahtevki

- [Ustvari interni zahtevek](#)

Oddali ste naslednje interne zahtevke:

- [Interni zahtevek st. 3](#)
- [Interni zahtevek st. 6](#)

Uporabniki

- [Dodaj uporabnika](#)
- [Seznam uporabnikov](#)

Slika 15: Uporabniški meni z uporabnikovimi podatki in ukaznim menijem

4.7.2.1 Podatki o uporabniku

Osnovni podatki o uporabniku:

- glava prikaže uporabniku njegovo ime in priimek ter nivo dodeljenih pravic,
- preko povezave *Moj Profil* ima vsak uporabnik vpogled v svoje uporabniške podatke, ki jih lahko tudi ureja (slika 16) in
- povezava *Odjavi* nudi uporabniku možnost odjave iz aplikacije.

Slika 16: Urejanje uporabniških podatkov

4.7.2.2 Ukazni meni

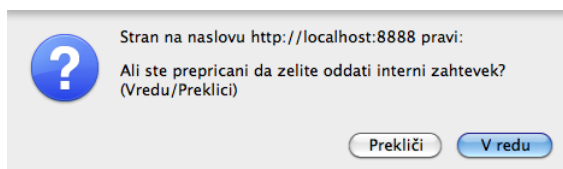
1. Interni zahtevki

Uporabnik s klikom na povezavo kreira novi interni zahtevek (slika 17). Na zahtevku izbere oddelek naročanja ter izpolni vsebino postavke. Z gumbom na desni, ima možnost dodajanja poljubnega števila postavk na zahtevku.

Zap. št.	Naziv blaga/storitve	Enota mere	Količina	Izdano	Opombe
1	CD mediji	kos	25		Kakšno dobro znamko prosim
2	USB kljuc 8GB	kos	5		Hitrost prenosa vsaj 25MB/s
3	USB kljuc 16GB	kos	3		Hitrost prenosa vsaj 10MB/s

Slika 17: Kreiranje novega internega zahtevka

Po kliku na gumb *Oddaj*, ga kontrolno sporočilo narejeno v programskem jeziku JavaScript [14], opozori na oddajo zahtevka (slika 18). Če je prepričan v oddajo, izbere *V redu*. S tem se zahtevek s postavkami shrani v podatkovno bazo [15]. Aplikacija mu ustrezno dodeli atribut *ID_Faze*, ki služi za prepoznavanje faze, v kateri se zahtevek nahaja.

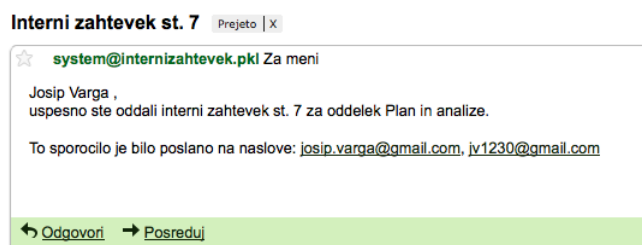


Slika 18: Potrditev oddaje internega zahtevka

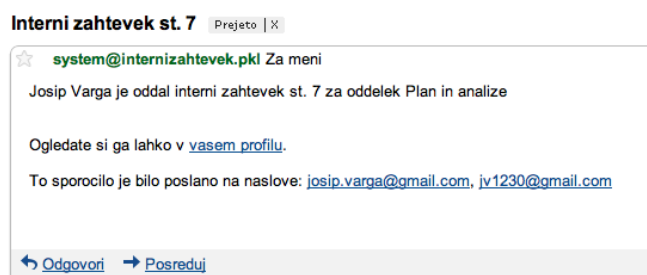
Uporabniku se pokaže še povzetek oddanega naročila (slika 19). Aplikacija pošlje elektronsko sporočilo [16] na uporabnikov elektronski naslov (slika 20) in na naslov uporabnika, ki je naslednji v procesu naročanja (slika 21).



Slika 19: Povzetek oddanega internega zahtevka



Slika 20: Poslano sporočilo na naslov uporabnika, ki je kreiral interni zahtevek



Slika 21: Poslano sporočilo na naslov uporabnika, ki je naslednji v procesu internega naročanja blaga in storitev

V primeru izbire možnosti **Prekliči**, ga vrne nazaj na zahtevek. Uporabnik, ki je naslednji v procesu lahko preko svojega profila, ali v elektronskem sporočilu, ki ga je prejel, odpre kreiran zahtevek. Nato ga pregleda, izpolni svoj del procesa in odda naprej (slika 22).

INTERNI ZAHTEVEK št. 6

Oddelek: Informatika

Zap. št.	Naziv blaga/storitve	Enota mere	Količina	Izdano	Opombe
1	DVD mediji	kos	5		Kakšno dobro znamko prosim
2	CD mediji	kos	25		0

Izpolni oddelek za plan in analize

Je v planu
 Ni v planu

Denarna sredstva so zagotovljena
 Denarna sredstva niso zagotovljena

Zahtevek je oddal/-a Josip Varga, 2010-08-16 17:23:50

Slika 22: Vodja oddelka za plan in analizo pregleda zahtevek, označi status plana in denarnih sredstev ter odda naprej

Razviti del procesa naročanja, obsega v aplikaciji kreiranje internega zahtevka in njegovo oddajo oziroma potrditev oddaje s strani pooblaščenega osebe. Interni zahtevek se nato pojavi v uporabniškem profilu vodje oddelka za plan in analizo. Le-ta ga pregleda, izbere ali je predmet naročila v planu, ter podatek o zagotovljenih denarnih sredstvih. Po izbiri gumba Oddaj in potrditvi oddaje, se zahtevek dopolni z izbranimi parametroma. Od tu naprej bi se proces ponavljal na enak način, vendar ga zaradi časovne omejitve nismo razvili.

Prav tako dobi uporabnik tudi povezave na vse interne zahtevke, ki jih je oddal oziroma kreiral. S klikom na zahtevek, se mu odprejo podrobnosti izbranega zahtevka (slika 23). V primeru, da je uporabnik pooblaščen oseba za določen del procesa naročanja blaga in storitev, se mu na tem mestu pokažejo še vsi oddani interni zahtevki, ki čakajo na njegovo obdelavo (slika 24).

INTERNI ZAHTEVEK št. 6

Oddelek: Informatika

Zap. št.	Naziv blaga/storitve	Enota mere	Količina	Izdano	Opombe
1	DVD mediji	kos	5		Kakšno dobro znamko prosim
2	CD mediji	kos	25		0

Zahtevek je oddal/-a Josip Varga, 2010-08-16 17:23:50

Slika 23: Podrobnosti o oddanem/kreiranem internem zahtevku

Pozdravljen/-a, Milena Miklavc, Pooblaščen/a oseba

[Moj Profil](#) | [Odiava](#)

Interni zahtevki

- [Ustvari interni zahtevke](#)

Oddali ste naslednje interne zahtevke:

- [Interni zahtevke st. 5](#)

Naslednji interni zahtevki čakajo obdelavo:

- [Interni zahtevke st. 1](#)
- [Interni zahtevke st. 2](#)
- [Interni zahtevke st. 4](#)
- [Interni zahtevke st. 5](#)
- [Interni zahtevke st. 6](#)

Uporabniki

- [Dodaj uporabnika](#)
- [Seznam uporabnikov](#)

Slika 24: Interni zahtevki, ki čakajo na obdelavo prijavljene pooblaščen/a osebe

2. Uporabniki

Če je uporabnik pooblaščen/a oseba ali administrator, ima v uporabniškem meniju povezavo za dodajanje novih uporabnikov (slika 25). Prav tako, ima na voljo seznam vseh uporabnikov, s katerega ima možnost pregleda podrobnosti posameznega uporabnika, omogočeno mu je tudi brisanje uporabnikov (slika 26).

DODAJANJE UPORABNIKA

Oddelek:

Ime:

Priimek:

Login:

Password:

Potrdi Password:

E-mail:

Pravice:

Slika 25: Dodajanje novih uporabnikov

SEZNAM UPORABNIKOV

ID	Ime	Priimek	Login	Password	E-mail	Pravice	Uredi	Izbriši
1	Josip	Varga	vargaj	••••••••	josip.varga@gmail.com	Administrator	Uredi	Izbriši
2	Siniša	Stikovič	stikovics	••••••	stikovics@test.si	Pooblaščen/a oseba	Uredi	Izbriši
3	Miloš	Zupančič	zupancicm	••••••••••	miloszu@email.si	Uporabnik	Uredi	Izbriši
4	Urška	Zupančič	zupancicu	••••	urska.zupancic@slotest.si	Pooblaščen/a oseba	Uredi	Izbriši
5	Milena	Miklavc	miklavcm	••••••	milenageneracija@hotmail.com	Pooblaščen/a oseba	Uredi	Izbriši

Slika 26: Seznam uporabnikov s pregledom podrobnosti in brisanjem uporabnika

4.8 Implementacija in analiza narejenega

Razvito aplikacijo smo namestili na testni strežnik. Njeno delovanje ter način uporabe smo pokazali določenim pooblaščenim osebam. Glede na to, da je uporaba aplikacije zelo enostavna, so uporabniki z veseljem privolili v testiranje.

Po nekaj tednih uporabe, smo prejeli pohvale na račun enostavnosti in funkcionalnosti razvite aplikacije. Uporabniki so nam predali tudi nekaj zanimivih idej in pripomb, ki jih bomo upoštevali pri nadaljnjem razvoju aplikacije.

5 SKLEP

Cilj diplomske naloge je bilo sistematično razviti in predstaviti aplikacijo za proces internega naročanja blaga in storitev. Uspelo nam je razviti aplikacijo, v kateri uporabnik ustvari interni zahtevek, ga izpolni in odda v obdelavo. V večini primerov uporabniki niso pooblašene osebe, zato se pojavi interni zahtevek najprej v uporabniškem profilu pooblašene osebe, za konkreten oddelek. Pooblašena oseba ga pregleda in če se z vsebino strinja, odda naprej v obdelavo. Ob oddaji, se vsem uporabnikom udeležnim v procesu do tega trenutka, pošlje elektronsko sporočilo, ki jih opozori na novi interni zahtevek. Pooblašena oseba ima v nadaljnjem koraku možnost obdelave svojega dela zahtevka.

Aplikacija je bila razvita od faze kreiranja internega zahtevka, do faze, ko interni zahtevek prevzame vodja oddelka za plan in analize. Nadaljnji razvoj bi obsegal še pot v oddelek za nabavo, kjer bi se izvedlo dejansko naročilo blaga oziroma storitev. Dodaten proces naročanja vključuje primer, ko blago oziroma storitev ni v planu, ali ni zagotovljenih denarnih sredstev. V tem primeru potuje zahtevek k direktorju, ki ga pregleda in se odloči o »usodi« naročila. V procesu se pojavljajo tudi nepredvidljive situacije, ki zahtevajo še natančnejše testiranje in preučitev vseh možnosti.

V nadaljevanju bo potrebno vse manjkajoče dele aplikacije razviti in preizkusiti v realnem okolju. Pričakujemo, da bo aplikacija v prihodnosti nadomestila trenutno časovno, ekonomično in ekološko neučinkovit sistem.

Pri izvedbi diplomske naloge smo imeli nekaj manjših težav s kompatibilnostjo programske opreme in časovno stisko. Ostalo je potekalo po načrtu.

VIRI

- [1] *E-Commerce and E-Business/Concepts and Definitions*. Dostopno na: http://en.wikibooks.org/wiki/E-Commerce_and_E-Business/Concepts_and_Definitions
- [2] *HTML*. Dostopno na: <http://en.wikipedia.org/wiki/HTML>
- [3] *PHP*. Dostopno na: <http://en.wikipedia.org/wiki/PHP>
- [4] *JavaScript*. Dostopno na: <http://en.wikipedia.org/wiki/JavaScript>
- [5] *Adobe Dreamweaver*. Dostopno na: http://en.wikipedia.org/wiki/Adobe_Dreamweaver
- [6] *MAMP*. Dostopno na: <http://www.mamp.info/en/mamp/index.html>
- [7] *MySQL*. Dostopno na: <http://en.wikipedia.org/wiki/MySQL>
- [8] *phpMyAdmin*. Dostopno na: <http://en.wikipedia.org/wiki/PhpMyAdmin>
- [9] *IBM Rational Unified Process*. Dostopno na: http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process
- [10] *What is UML?* Dostopno na: http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/what_is_uml.htm
- [11] *Rational Rose*. Dostopno na: http://searchcio-midmarket.techtarget.com/sDefinition/0,,sid183_gci516025,00.html
- [12] *Using Rational Rose to Create Object-Oriented Diagrams*. Dostopno na: <http://www.cs.virginia.edu/~horton/cs494/info/rose-howto-v2.pdf>
- [13] *Power Designer*. Dostopno na: <http://www.sybase.com/products/modelingdevelopment/powerdesigner>
- [14] *Javascript Tutorial*. Dostopno na: <http://www.tizag.com/javascriptT/index.php>
- [15] J. Gerner, M. K. Glass, G. Mailer, E. Naramore, Y. L. Scouarnec, J. Stolz, *Beginning PHP, Apache, MySQL Web Development*, Indianapolis: Wiley Publishing, Inc., 2005, pogl. 6