

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Boštjan Tittl

**PRIMERJAVA RAZVOJA SPLETNE
APLIKACIJE V TEHNOLOGIJAH
PHP IN AJAX**

DIPLOMSKO DELO NA
VISOKOŠOLSLEM STROKOVNEM ŠTUDIJU

MENTOR: dr. Igor Rožanc

Ljubljana, 2009



Št. naloge: 00423/2009

Datum: 15.01.2009

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

Kandidat: **BOŠTJAN TITTL**

Naslov: **PRIMERJAVA RAZVOJA SPLETNE APLIKACIJE V TEHNOLOGIJAH
PHP IN AJAX**

**THE COMPARISON OF PHP AND AJAX WEB APPLICATION
DEVELOPMENT**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Nove spletne tehnologije omogočajo izdelavo spletnih aplikacij, ki so bolj funkcionalne, učinkovitejše in prijaznejše do uporabnika, vendar je zato delo razvijalca pogosto zahtevnejše in zahteva povezavo različnih znanj.

V diplomski nalogi primerjajte razvoj spletne aplikacije v tehnologiji PHP z njeno nadgradnjo, ki je izvedena v tehnologiji AJAX. V ta namen sistematično predstavite razvoj aplikacije za spletne novice s klepetalnico, ki ima več vrst uporabnikov, podatke pa hrani v podatkovni bazi MySQL. Analizo zahtev in načrtovanje izvedite v skladu z zahtevami metodologije RUP, pri sami izvedbi pa izpostavite predvsem prednosti in slabosti obeh pristopov. Nalogo zaključite s sistematično primerjavo obeh rešitev z vidika uporabljenih tehnologij in značilnosti uporabe.

Mentor:

viš. pred. dr. Igor Rožanc

Dekan:

prof. dr. Franc Solina

UNIVERSITY OF LJUBLJANA
FACULTY OF COMPUTER AND INFORMATION SCIENCE

Boštjan Tittl

**THE COMPARISON OF PHP AND
AJAX WEB APPLICATION
DEVELOPMENT**

DEGREE DISSERTATION OF HIGHER PROFESSIONAL STUDIES

MENTOR: Igor Rožanc, PhD

Ljubljana, 2009

POVZETEK

Glavni cilj diplomske naloge je primerjava dveh tehnik razvoja, ki sta danes zelo priljubljeni pri izdelavi interaktivnih spletnih aplikacij. Na primeru izdelave spletne aplikacije za spletne novice primerjamo tehnologijo razvoja v PHP-ju z nadgradnjo le-te s pomočjo AJAX-a.

V drugem poglavju predstavimo osnovne pojme metodologije RUP, ki smo jo izbrali za razvoj spletne aplikacije za spletne novice. Sledi opis uporabljenih tehnologij, tehnik in orodij, s pomočjo katerih smo prišli do želene rešitev danega problema.

Jedro diplomske naloge je razvoj aplikacije za spletne novice. V prvem delu razvoja se ukvarjamo predvsem s tehnologijami kot so: HTML, PHP, CSS, MySQL in JS. Drugi del razvoja je nadgradnja prvega dela s pomočjo AJAX-a, kar omogoča večjo interaktivnost sistema. Jedro vključuje tudi analizo in načrtovanje sistema z vsemi diagrami, ki pridejo pri tem v poštev.

Po razvoju sledi še podrobna predstavitev spletne aplikacije za spletne novice in nekaterih vnosnih mask sistema. Aplikacija omogoča obiskovalcem branje vseh novic, ki se nahajajo v podatkovni bazi. Gost ima možnost registracije in prijave v sistem spletnih novic. Z uspešno prijavo v sistem postane gost uporabnik. Uporabnik ima na voljo razširjeno funkcionalnost sistema, saj lahko sam dodaja novice, spreminja svoje novice ali pa jih celo izbriše. Poleg tega pa ima na voljo tudi klepetalnico, kjer si lahko interaktivno izmenjuje mnenja z ostalimi uporabniki. Uporabnik lahko pridobi tudi status administratorja, kar mu še razširi funkcionalnost. Administrator se razlikuje od uporabnika po tem, da lahko spreminja in briše vse novice, torej ne le svojih, ter da lahko spreminja status vsem uporabnikom.

V zaključku primerjamo uporabljene tehnologije in orodja pri razvoju spletne aplikacije. Pri tem objektivno izpostavimo prednosti in slabosti obeh načinov razvoja. Zapišemo tudi ugotovitve in ideje, do katerih smo prišli pri pisanju diplomske naloge.

Ključne besede

spletne aplikacije, spletne tehnologije, PHP, AJAX, MySQL

ABSTRACT

The main aim of the diploma thesis is the comparison of the two development techniques that are nowadays very-popular in the development of interactive web applications. Based on the example on web news application development, we compare the development technology in PHP with the version of the same technology which was additionally upgraded by means of AJAX.

In the second chapter we introduce the basic terms of the RUP methodology which was chosen for the development of the web news application and after that describe the technologies, techniques and tools used in solving the stated problem.

The main emphasis of the diploma thesis is on the development of the web news application. In the first part of this development we are mostly dealing with the following technologies: HTML, PHP, CSS, MySQL and JS. The second development part is an upgrade of the first one by means of AJAX which enables the system to operate on higher interactivity levels. The main part of the diploma thesis also includes the system analysis and planning – with all diagrams appertaining to this matters.

The subject on development is followed by a detailed presentation of the web news application as well as of some other system input masks. The application enables the visitor to read all the news in the database. He has the option to register and log-in to the web news system. Once this is done, the visitor or guest becomes an user and thus gains access to the expanded functionality which enables him to add, change or even delete his own news. Besides that, the user has a chat room at his disposal where he can interactively communicate with other users. Furthermore, the user can also gain the administrator status and thus expand the functionality even more. Unlike the user, the administrator can change and delete all news and not just the ones created by himself; moreover, he can change the status of all other users.

Finally, we compare the technologies and tools used in web application development. We point out the advantages and disadvantages of both development methods in objective manner. We also state the findings and ideas that sprang up while writing the diploma thesis.

Key words

web applications, web technologies, PHP, AJAX, MySQL

KAZALO

KAZALO	5
1. UVOD	7
2. METODOLOGIJA RUP	8
2.1. ITERATIVNI RAZVOJ.....	9
2.2. JEZIK ZA MODELIRANJE UML.....	11
3. TEHNOLOGIJE IN ORODJA	13
3.1. PREDSTAVITEV TEHNOLOGIJ.....	13
3.1.1. Jezik HTML.....	13
3.1.2. Jezik PHP.....	14
3.1.3. Tehnologija AJAX.....	14
3.1.4. SUPB MySQL.....	16
3.1.5. CSS predloge.....	18
3.2. PREDSTAVITEV ORODIJ.....	19
3.2.1. Orodje PhpMyAdmin.....	19
3.2.2. Orodje Photoshop.....	20
3.2.3. Orodje FileZilla Client.....	21
3.2.4. Apache2triad.....	22
3.2.5. PowerDesigner.....	23
3.2.6. Rational Rose.....	24
4. ANALIZA IN NAČRTOVANJE	25
4.1. ZAJEM ZAHTEV.....	25
4.1.1. Nefunkcionalne zahteve.....	25
4.1.2. Funkcionalne zahteve.....	26
4.2. RAZREDNI DIAGRAM ZA CELOTEN SISTEM.....	29
4.3. DIAGRAMA ZAPOREDJA.....	30
4.4. PODATKOVA BAZA.....	31
4.4.1. Globalni model.....	32
4.4.2. Konceptualni model.....	32
4.4.3. Logični model.....	34
4.4.4. Fizični model.....	35
4.5. STRUKTURA SPLETNE APLIKACIJE.....	36
5. IZVEDBA IN PRIMERJAVA MODULOV V PHP IN AJAX RAZLIČICI	38
5.1. GLASOVANJE NOVIC.....	38
5.2. SPREMINJANJE STATUSA UPORABNIKOM.....	39
5.3. UGOTOVITVE.....	40
6. PREDSTAVITEV IN UPORABA SPLETNIH NOVIC	41
6.1. NAVIGACIJA NA PRVEM NIVOJU.....	41
6.2. NAVIGACIJA NA DRUGEM NIVOJU.....	41

6.3. REGISTRACIJA	42
6.4. PRIJAVA	42
6.5. AŽURIRANJE NOVIC	43
6.5.1. Dodajanje novic	43
6.5.2. Pregled novic	44
6.5.3. Brisanje novic	44
6.5.4. Preurejanje novic	44
6.6. STATUSI UPORABNIKOV	44
6.7. KLEPETALNICA	45
6.8. ISKALNIK NOVIC	46
6.9. GLASOVANJE NOVIC	46
7. ZAKLJUČEK	47
7.1. OCENA UPORABLJENIH ORODIJ	47
7.2. PRIMERJAVA TEHNOLOGIJ	47
7.3. SKLEP	49
8. ZAHVALA.....	50
9. PRILOGA.....	51
9.1. PHP RAZLIČICA GLASOVANJA NOVIC	51
9.2. AJAX RAZLIČICA GLASOVANJA NOVIC	53
9.3. PHP RAZLIČICA SPREMINJANJE STATUSA UPORABNIKOV	55
9.4. AJAX RAZLIČICA SPREMINJANJE STATUSA UPORABNIKOV	58
10. VIRI	61

1. UVOD

Spletne aplikacije so aplikacije, ki so dostopne preko spletnega brskalnika preko omrežja, kot sta internet ali intranet. Sposobnost za posodobitev in vzdrževanje spletnih aplikacij brez distribucij in namestitvev programske opreme na potencialno tisoče računalnikov je ključni razlog za njihovo veliko popularnost. Običajne spletne aplikacije vključujejo spletne pošte, spletne maloprodaje, spletne dražbe, spletne enciklopedije, masivne večigralske spletne igre in še veliko drugih funkcij.

Izdelava dinamičnih spletnih aplikacij je relativno zahtevno opravilo. Jedro kompleksnosti predstavlja programski del takšnih aplikacij in zahtevnejše izvajalno okolje, ki ga ne potrebujemo pri statičnih spletnih aplikacijah. Če lahko rečemo, da so enostavnejše statične spletne aplikacije sposobni narediti računalniški navdušenci, pa potrebuje oseba, ki želi izdelati dinamične spletne aplikacije, kar nekaj znanja in inženirski pristop k njihovi gradnji. Spekter znanj je širok in sega vse od grafičnega oblikovanja do programiranja v različnih programskih jezikih in sistemskega vzdrževanja.

Zato smo si za diplomsko nalogo izbrali primerjavo obstoječih internetnih tehnologij in tehnik, ki pridejo v poštev pri izgradnji spletnih aplikacij. To znanje mi bo prišlo še kako prav pri nadaljnjem strokovnem delu. Za nadzorno predstavitev bomo izdelali spletno aplikacijo za spletne novice, ki jo bomo najprej razvili v PHP tehnologiji. V nadaljevanju bomo aplikacijo nadgradili s tehniko AJAX.

Namen diplomske naloge je:

- predstaviti osnovne pojme in načine izdelave spletnih aplikacij,
- opisati in primerjati tehnologije ter rešitve za razvoj posameznih funkcionalnosti spletne aplikacije za spletne novice,
- objektivno primerjati in oceniti obe izdelani rešitvi.

Cilji diplomske naloge so:

- seznanitev z osnovami metodologije RUP,
- spoznati osnovne postopke razvoja spletnih aplikacij,
- izdelati uporabno spletno aplikacijo za spletne novice,
- optimizacija nekaterih funkcionalnosti sistema.

2. METODOLOGIJA RUP

Metodologija zajema vse, kar redno počnemo, da bi dosegli želen rezultat, torej izdelek ali storitev, ki je cilj našega dela. V primeru razvoja programske opreme to ne pomeni zgolj postopkov, ki so neposredno povezani z razvojem (npr. analiza, načrtovanje itd.), temveč zajema tudi podporne postopke, načine komunikacije med sodelujočimi, pravila odločanja itd. V tem oziru lahko metodologijo opredelimo tudi kot množico dogovorov (konvencij), s katerimi se projektna skupina oz. organizacija strinja [1].

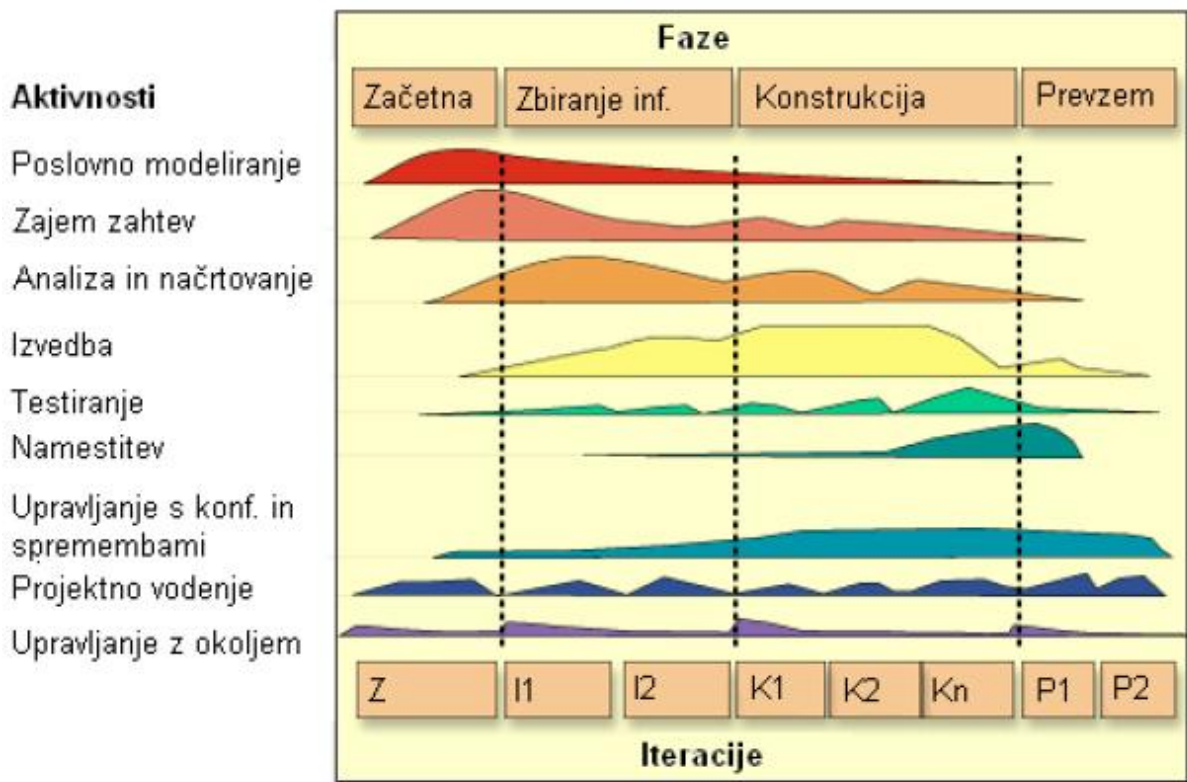
Metodologija RUP (angl. Rational Unified Process) velja za težko metodologijo, saj zajema in podrobno opisuje številne elemente. Kljub njeni teži se je metodologija na področju objektnega razvoja izkazala kot primerna izbira. RUP se izvaja iterativno ter za vizualizacijo uporablja jezik UML [2].

RUP je baza znanja, ki vključuje vrsto izkušenj pri razvoju aplikacij, ki so se tekom let pridobile pri delu na obsežnih in uspešnih projektih. Tako opisuje, kako učinkovito uporabiti šest najboljših izkušenj s področja razvoja programske opreme. Te so :

- iterativni razvoj,
- obvladovanje zahtev,
- uporaba komponentne arhitekture,
- vizualno modeliranje,
- preverjanje kakovosti in
- nadzorovanje sprememb.

RUP določa štiri faze, pri čemer lahko vsako fazo razdelimo še na več iteracij:

1. zametek – začetna faza (angl. inception):
 - začetek projekta, zbiranje informacij
 - študija izvedljivosti
2. dodelava – zbiranje informacij (angl. elaboration):
 - analiza in načrtovanje
 - odločitve o arhitekturi
3. izgradnja – konstrukcija (angl. “construction”):
 - kodiranje
4. prehod – prevzem (angl. “transition”):
 - priprava za prenos k uporabniku
 - instalacija pri uporabniku



Slika 1. Shema procesa razvoja po metodologiji RUP.

Kot je razvidno iz slike 1 imamo v posameznih fazah ali iteracijah opravka z večjim številom aktivnosti, ki tečejo vzporedno. V začetni fazi imamo predvsem opravka s poslovnim modeliranjem in zajemom zahtev. V drugi fazi prevladuje aktivnost analize in načrtovanja. Razvidno je tudi, da so aktivnosti testiranje, upravljanje z okoljem in projektno vodenje približno enako prisotni skozi vse štiri razvojne faze.

2.1. Iterativni razvoj

Za mnoge spletne razvijalce je uporaba iteracij ali ponovitev eden izmed ključnih pristopov pri njihovem delu. V praksi to pomeni, da odlašamo izvedbe za določeno časovno obdobje z namenom hitrejšega napredovanja in kasnejšega vračanja k istemu problemu, kjer dosežene rezultate izboljšamo oziroma popravimo. S takšnim načinom dela se lažje prebijemo skozi kritične točke dela.

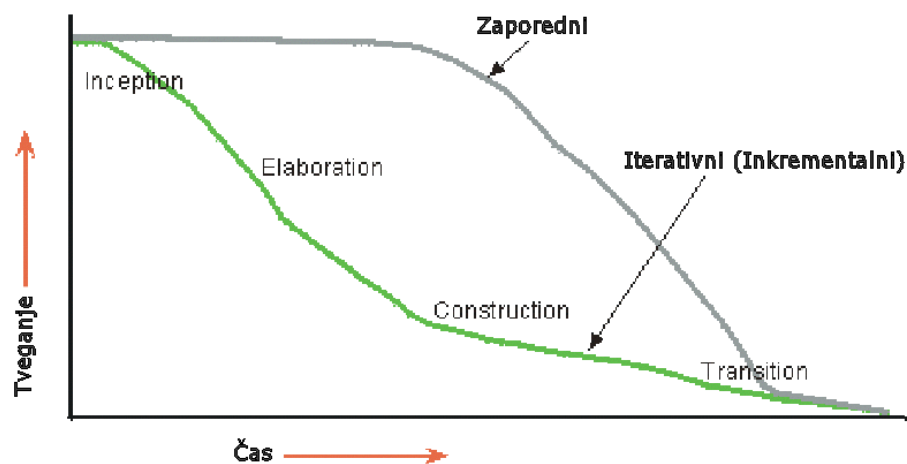
S stališča projektnega vodje je takšen pristop težje obvladljiv, saj je pri planiranju težko oceniti, koliko iteracij bo potrebnih za dosego zastavljenega cilja z želeno kakovostjo izdelka. Tudi vnaprej definirano število iteracij ni prava rešitev problema, saj prav tako ne zagotavlja optimalne rešitve problema.

Resnični cilj iterativnega razvoja je doseči nadzorovano dopolnjevanje dela sistema z namenom odpraviti napake ali vključiti izboljšave, ki temeljijo na povratnih informacijah uporabnikov. V povezavi z iterativnim razvojem srečamo dve trditvi:

- običajno stvari najprej naredimo narobe, preden jih naredimo pravilno, ter
- običajno stvari najprej naredimo slabo, preden jih naredimo dobro.

Pojem iterativen razvoj se pogosto uporablja v primerih, ko se razvijalci želimo izogniti zaporednemu razvoju. Če proces zahteva šest korakov, ki si sledijo drug za drugim, bomo pogosto problem rešili enostavneje in hitreje, če ne bomo zaključili predhodnega koraka pred začetkom naslednjega (kot zahteva zaporedni razvoj), temveč bomo naredili še del drugega, del tretjega in morda nekaj petega, preden bomo popolnoma končali prvega. V skladu s tem po vsaki iteraciji ovrednotimo rezultat in planiramo naslednjo iteracijo, v kateri bomo izboljšali sistem in kakovost izgradnje.

Takšen potek dela je bolj pester in prispeva k večji motiviranosti, hkrati pa zagotovi hitrejše napredovanje, saj se ob nastalem problemu ne zadržujemo predolgo.

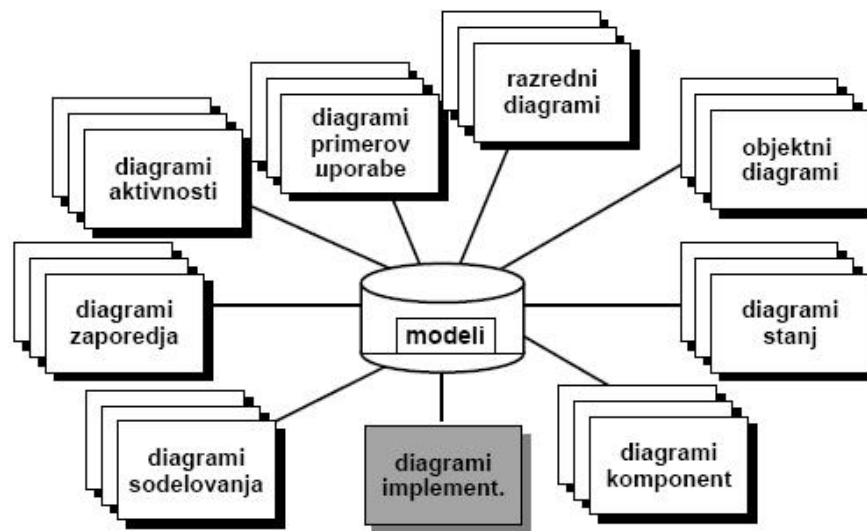


Slika 2. Zgodnje zmanjšanje tveganja pri iterativnem razvoju.

Pozitivna lastnost iterativnega razvoja je tudi zgodnje zmanjševanje tveganja v primerjavi z zaporednim (slika 2), ki se prične že v začetni fazi. V fazi zbiranja informacij se razlika še povečuje, kjer (na meji s konstrukcijo) doseže svoj maksimum. Razlika se popolnoma izniči šele na sredini faze za prevzem.

2.2. Jezik za modeliranje UML

Metodologija RUP uporablja za vizualno modeliranje jezik UML (angl. Unified Modeling Language) . UML je jezik, ki se uporablja za specifikacijo, vizualizacijo, konstrukcijo in dokumentacijo.



Slika 3. UML diagrami.

Diagram je vizualni model, ki predstavlja pogled na določenem nivoju abstrakcije. V nadaljevanju bomo na kratko pojasnili diagrame iz slike 3. Tej so:

- diagrami primerov uporabe predstavljajo interakcijo med akterji (npr. uporabniki, gosti, itd.) in računalniškim sistemom,
- razredni diagrami prikazujejo statično strukturo sistema v obliki razredov, metod, atributov in povezav med razredi,
- objektni diagrami prikazuje statično strukturo sistema in pojasnjuje kompleksne razrede,
- diagrami stanj prikazujejo zaporedje stanj, skozi katere gre objekt v času njegovega obstoja in dogodkov, ki sprožijo prehode med stanji,
- diagrami implementacije prikazujejo fizično zvezo med programskimi in strojnimi sestavnimi deli sistema. Sem sodijo tudi diagrami komponent in arhitekturni diagrami,
- diagrami sodelovanja prikazujejo interakcije med objekti in njihove medsebojne povezave,

- diagrami zaporedja prikazujejo časovno (zaporedno) sodelovanje objektov v interakciji ter
- diagrami aktivnosti prikazujejo dinamično obnašanje sistema z vidika dejavnosti. Primerni so tudi za prikaz sočasnih dejavnosti in operacij.

Poznamo dva tipa pogledov:

- statični pogled za opis strukture (razredni in objektni diagram, komponenti diagram),
- dinamični pogled za opis obnašanja (primeri uporabe, diagram stanja, diagram aktivnosti)

Prednosti UML diagramov so prevsem:

- enoten standard,
- razširljiv (stereotipi),
- prilagodljiv .

Slabosti uporabe UML-ja pa so:

- prepad med analizo in načrtovanjem,
- prebogata notacija (modeliramo lahko 80 odstotkov problemov z okoli 20 odstotkov UML-ja) [3].

3. TEHNOLOGIJE IN ORODJA

Poglavje je namenjeno predstavitvi orodij in tehnologij, ki smo jih izbrali in uporabili pri razvoju spletnih novic. Predstavitev obsega definicijo, uporabo, zgodovino in namen posamezne tehnologije oziroma orodja. Nabor tehnologij in orodij je zelo pester. Enako ali podobno stvar je mogoče izvesti tudi z drugimi tehnologijami oziroma orodji.

3.1. Predstavitev tehnologij

3.1.1. Jezik HTML

HTML (angl. Hyper Text Markup Language) je označevalni jezik za izdelavo spletnih aplikacij. Predstavlja osnovo vsakega spletnega dokumenta. S pomočjo HTML-ja ustvarimo strukturo in semantično ureditev dokumenta. Pišemo ga lahko v vsakem urejevalniku besedil in je dokaj preprost za učenje in uporabo [4].

HTML jezik so osnovali znanstveniki pod vodstvom Tima Berners-Leeja med letoma 1989 in 1990 v Švici, v Evropskem centru za fiziko osnovnih delcev (CERN). Ker so si hoteli najnovejša spoznanja med seboj izmenjavati kar najhitreje, so potrebovali dovolj hiter način za izmenjavo podatkov - besedil, slik in zvoka. Tega pa dotlej internet ni omogočal, saj je bilo do takrat namreč mogoče uporabljati le določene ukaze ali pošiljati preprosta tekstovna sporočila. Za osnovo jezika HTML so uporabili jezik SGML (angl. Standard Generalized Markup Language). Tako so se v začetku devetdesetih let začele pojavljati prve barvne spletne strani in povezave med besedili. S pomočjo obogatene besedila (angl. hypertext) je bilo mogoče med seboj povezati tekstovne ali slikovne strani. HTML jezik nam je tako omogočil izdelavo obogatene besedila in je močno povezan z razvojem svetovnega spleta.

Gradniki HTML jezika so značke (angl. tags). Samostojne značke ne potrebujejo zaključka (npr. nova vrsta), saj vsaka naslednja uporaba iste značke prekliče prejšnjo (primer 1). Pri vseh ostalih pa moramo posebej navesti, do kje seže učinek značke (npr. debeli izpis). V tem primeru uporabimo končno značko. Končna značka se od začetne loči le po dodatnem znaku / pred imenom značke. Vse kar se nahaja v notranjosti značke, se podreja oblikovanju, ki ga značka zahteva.

To besedilo je v prvi vrsti
 To odebeljeno besedilo je
v drugi vrsti

Primer 1. HTML značke.

3.1.2. Jezik PHP

PHP (angl. PHP Hypertext Preprocessor, izvirno Personal Home Page tools) je razširjen odprtokodni programski jezik, ki se uporablja za strežniško uporabo oziroma za razvoj dinamičnih spletnih vsebin, ki črpajo podatke iz podatkovne baze. PHP primarno teče na spletnem strežniku, kjer s pomočjo PHP izvorne kode generira HTML kodo kot izhod [5].

Napisal ga je dansko-kanadski programer Rasmus Lerdorf leta 1994, da bi zamenjal nekaj skript napisanih v Perlu, ki jih je uporabljal za upravljanje svoje spletne strani. Lerdorf je sprva napisal PHP, da bi prikazal svoj življenjepis in hkrati zajemal podatke obiskovalcev strani. Tako je 8. julija 1995 izdal »Personal Home Page Tools«, zatem ko ga je združil s svojim interpreterjem za spletne obrazce (PHP/FI).

Lahko ga primerjamo z Microsoftovim ASP-jem (angl. Active Server Pages) in CGI (angl. Common Gateway Interface) programi. Podoben je običajno strukturiranim programskim jezikom, še najbolj jezikoma C in Perl ter izkušenim programerjem dovoljuje razvoj zapletenih sistemov brez dolgega učenja.

```
$naslov=$_POST['naslov'];
$ategorija=$_POST['ategorija'];
$avtor=$_POST['avtor'];
$besedilo=$_POST['besedilo'];
$datum=date("d.m.Y");
$query="INSERT INTO novica
(id,naslov,besedilo,kategorija,avtor, datum)VALUES
('','$naslov', '$besedilo', '$ategorija','$avtor','$datum')";
$SQL->Query($query);
```

Primer 2. PHP skripta za vnos novice v podatkovno bazo.

Primer 2 prikazuje sprejem spremenljivk, ki so bile poslane preko metode POST in pridobitev datuma na strežniku v formatu dan/mesec/leto. Sledi stavek MySQL, ki poskrbi za vnos podatkov v tabelo novica in njegova dejanska izvršitev.

3.1.3. Tehnologija AJAX

Ajax (angl. Asynchronous JavaScript And XML) je skupek spletnih tehnologij uporabnih za izgradnjo interaktivnih in vsebinsko bogatih spletnih aplikacij [6]. Zaradi boljšega razumevanja snovi bomo najprej predstavili nekatere tehnologije, ki tvorijo AJAX.

- a) XML (angl. Extensible Markup Language) je razširljiv označevalni jezik, ki ga na spletu pogosto srečate [4]. XML je preprost jezik podoben HTML-ju, ki nam določa format za opisovanje strukturiranih podatkov ali arhitekturo za prenos oziroma njihovo izmenjavo podatkov med več omrežji. XML spreminja mnogo aspektov

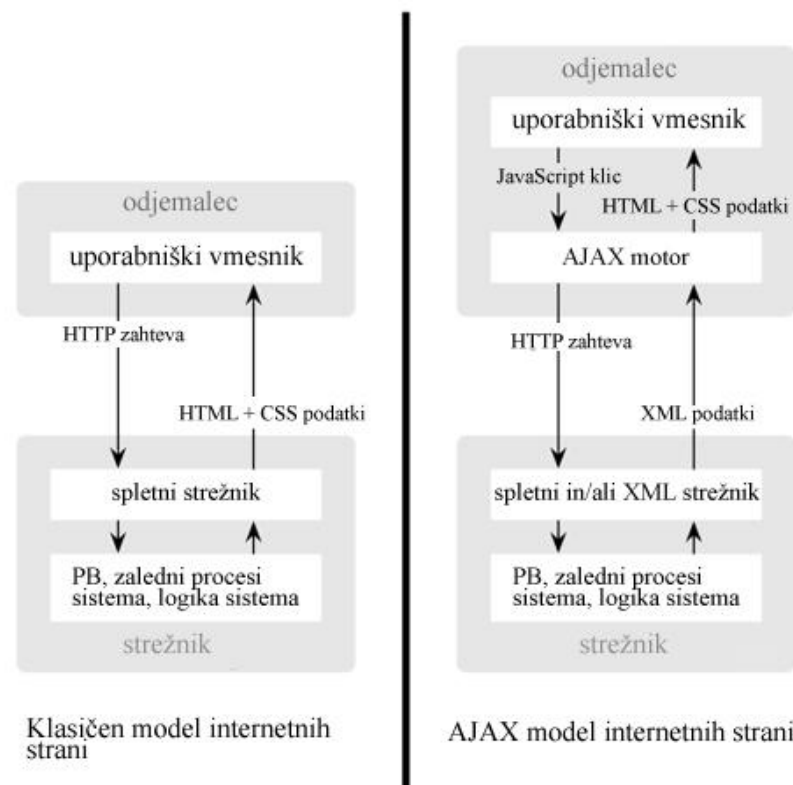
računalništva, še posebej na področju komuniciranja aplikacij in strežnikov. Da pa se ga tudi razširiti, saj si lahko sami izmislimo imena značk. Zelo je uporaben pri komunikaciji, saj ima zelo preprosto in pregledno zgradbo.

XML je razdeljen na 3 dele:

- podatkovni (vanj shranimo podatke v predpisani obliki z zelenimi značkami),
 - deklarativni (skrbi za to, da lahko pri dodajanju novih podatkov vidimo, kaj kakšna značka predstavlja),
 - predstavitevni (z njim oblikujemo izpis podatkov).
- b) JavaScript je objektni skriptni programski jezik, ki ga je razvil Netscape z namenom, da pomaga spletnim programerjem pri ustvarjanju interaktivnih spletnih aplikacij [7]. Program se vgradi ali pa vključi v HTML z namenom, da opravlja naloge, ki nadgrajujejo statične strani (npr. odpiranje novih oken, preverjanje pravilnost vnesenih podatkov, enostavni izračuni,...). Na žalost različni spletni brskalniki pri uporabi JS različno reagirajo in vračajo različen objekt (npr. `XMLHttpRequest`). Za podporo vseh brskalnikov je zato potrebno napisati več različic funkcij. JavaScript lahko pišemo kar v HTML datoteko ali pa vključimo zunanjo JavaScript datoteko.

Spletna aplikacija lahko s pomočjo AJAX-a pridobi podatke s strežnika asinhrono v ozadju, brez da bi bilo potrebno celotno spletno stran osvežiti. Podatke pridobimo s pomočjo objekta `XMLHttpRequest`. Kljub imenu, uporaba JavaScript-a, XML-a ali asinhronega delovanja ni obvezna. Izbira tehnologij tudi ni tako toga, saj lahko namesto JavaScript-a uporabimo tudi VBScript ali kateri podoben skriptni jezik, namesto XML-ja pa denimo JSON (angl. JavaScript Object Notation). Tudi PHP ni obvezen, saj ga lahko mirno nadomestimo s kakšnim podobnim jezikom, ki se izvaja na strežniku, recimo z ASP-jem.

Izraz AJAX se je pojavil leta 2005, vendar tehnike za asinhrono povezave segajo v leto 1996, ko je Internet Explorer predstavil `iFrame` element. Leta 1999 je Microsoft ustvarili `XMLHttpRequest` objekt, kot je ActiveX kontrolnik v Internet Explorerju 5. Razvijalci drugih brskalnikov (Mozilla in Safari), so kmalu sledili z avtohtonimi različicami objekta. S takim obnašanjem so nas prisilili, da moramo za ActiveX kontrolo pisati več kode, če želimo, da je aplikacija kompaktilna z različnimi brskalniki.



Slika 4. Primerjava klasičnega in AJAX modela spletne aplikacije.

Slika 4 nam nadzorno prikazuje razliko med klasičnim (PHP) in AJAX modelom. Bistveno razliko opazimo na strani odjemalca, kjer imamo v AJAX modelu tudi AJAX motor, ki poskrbi za večjo odzivnost in interaktivnost. Klasični razvoj, ki nima AJAX motorja, je zato prisiljen za vsako osvežitev klicati strežnik.

3.1.4. SUPB MySQL

MySQL (angl. My Structured Query Language) je sistem za upravljanje s podatkovnimi bazami, ki nam omogoča shranjevanje raznih podatkov na podatkovnem strežniku v poljubni obliki [8]. Omogoča nam upravljanje s podatki, ki so shranjeni v relacijskih podatkovnih bazah. Osnovni namen MySQL-a je izvedba specifičnih nalog v PB, ter izvajanje poizvedb nad podatki. S spletno aplikacijo se poveže preko strežniških programskih jezikov.

Pri povezavi s podatkovnim strežnikom moramo podati osnovne podatke, kot so:

- uporabniško ime,
- geslo,
- ime podatkovne baze ter
- gostitelja (angl. host).

Značilnosti MySQL-a:

- zasnovan je na množicah (ker izhaja iz relacijskega modela podatkov),
- deklarativnost (povemo kaj in ne kako),
- funkcionalna preprostost, enostavnejši nabor ukazov (v primerjavi z drugimi) in
- predvsem: dovolj zmogljiv za večino spletnih aplikacij in zastoj.

3.1.5. CSS predloge

CSS (angl. Cascading Style Sheets) so predloge, ki določajo videz spletnih aplikacij [9]. Z njimi določamo pisavo, velikosti črk ter vizualno predstavitev spletne aplikacije. HTML naj bi predstavljal semantično strukturo in smiselno hierarhijo dokumenta, CSS pa ima predstavitevno vlogo.

CSS podpirajo novejši spletni brskalniki (Internet Explorer 4.0 in več, Netscape Navigator 4.72, Mozilla, Opera). Spletni brskalniki, ki standarda CSS ne podpirajo, prikažejo običajen HTML dokument, brez oblikovanja zapisanega v CSS datoteki.

CSS lahko v HTML kodo vključimo na dva načina. Prvi in hkrati najbolj priljubljen način je vključevanje zunanje datoteke tipa CSS. Drugi način dopušča pisanje same CSS kode kar v HTML datoteki (primer 3).

```
<html>
  <head>
    <style type='text/css' media='all'>
      #okno1{
        width:200px;
        height:200px;
        background-color:yellow;
      }
      .okno2{
        width:400px;
        height:400px;
        border: 1px dashed blue;
      }
    </style>
  </head>
  <body>
    <div id='okno1'></div>
    <div class='okno2'></div>
  </body>
</html>
```

Primer 3. CSS skripta za oblikovanje HTML značk.

Primer 3 nam prikazuje neposredno vključevanje CSS kode v sam HTML dokument. Nanjo se sklicujemo preko id ali class atributov, katerim ime določimo sami. Razlika med atributoma je razvidna v CSS kodi, kjer za atribut id uporabimo # pred imenom za atribut class pa . .

3.2. Predstavitev orodij

3.2.1. Orodje PhpMyAdmin

PhpMyAdmin je orodje, napisano v PHP-ju namenjeno administraciji MySQL-a preko brskalnika [10].

Z njim lahko:

- ustvarimo in izbrišemo podatkovno bazo,
- ažuriramo tabele,
- ažuriramo polja v tabelo,
- izvršujemo katerikoli SQL ukaz,
- upravljamo s ključi,
- dodeljemo pravice ter
- izvažamo podatke v najrazličnejših formatih.

Poleg vsega je brezplačen in podpira tudi slovenski jezik. Za delovanje PhpMyAdmin potrebujemo predhodno nameščene programe Apache, PHP in MySQL.

Ob uspešni prijavi se odpre privzeta stran (slika 5) za urejanje podatkovnih baz, ki ponuja udobno in učinkovito upravljanje s podatkovnimi bazami. Na levi strani slike vidimo ime baze s šestimi tabelami, ki jih vsebuje. Desna stran slike pa prikazuje podrobnosti o tabelah.

The screenshot shows the PhpMyAdmin interface for a database named 'novice'. The left sidebar lists the tables: chat, chat_users, glas, novica, pasica, and uporabnik. The main area displays a table with the following data:

Tabela	Akcija	Zapisi	Vrsta	Pravilo za razvrščanje znakov	Velikost	Presežek
<input type="checkbox"/> chat		12	MyISAM	latin1_swedish_ci	7,1 KB	2.716 Bytes
<input type="checkbox"/> chat_users		5	MyISAM	latin1_swedish_ci	2,2 KB	-
<input type="checkbox"/> glas		25	MyISAM	latin1_swedish_ci	2,7 KB	-
<input type="checkbox"/> novica		12	MyISAM	latin1_swedish_ci	12,3 KB	-
<input type="checkbox"/> pasica		1	MyISAM	latin1_swedish_ci	2,0 KB	-
<input type="checkbox"/> uporabnik		6	MyISAM	latin1_swedish_ci	2,2 KB	-
6 tabel	Vsota	61	MyISAM	latin1_swedish_ci	28,5 KB	2,7 KB

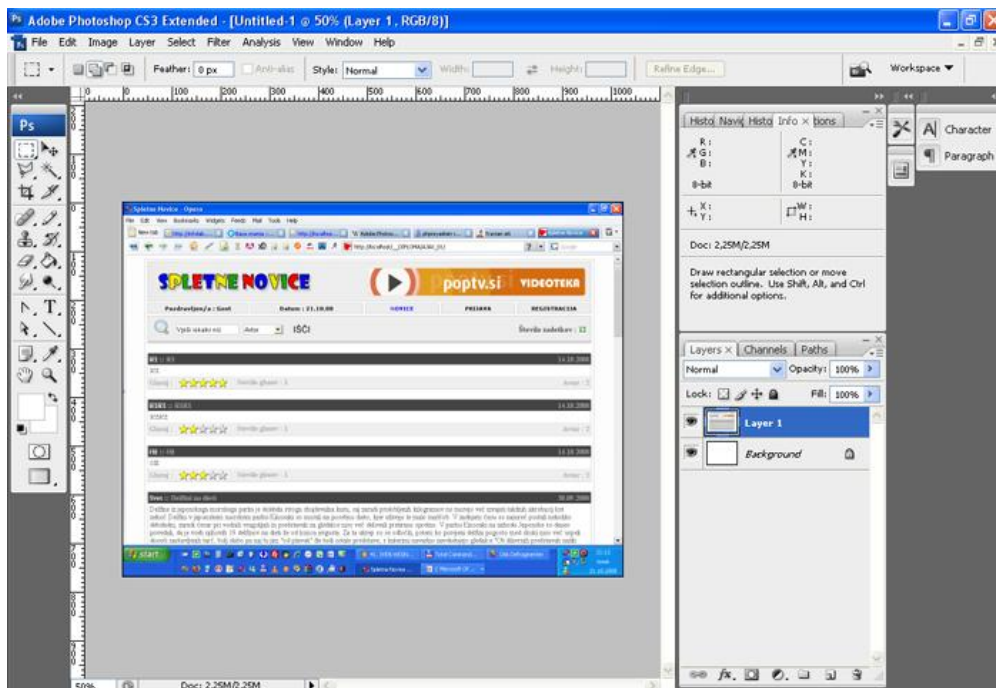
Below the table, there are options to 'Označi vse / Odznači vse / Preveri prekoračene' and a dropdown for 'Z označenim:'. At the bottom, there is a form to 'Ustvari novo tabelo v podatkovni bazi novice' with fields for 'Ime:' and 'Number of fields:', and an 'Izvedi' button.

Slika 5. Prikaz grafičnega uporabniškega vmesnika PhpMyAdmin.

3.2.2. Orodje Photoshop

Adobe Photoshop je računalniški program za oblikovanje slik. Razvil ga je Adobe Systems, ki je trenutno eden izmed vodilnih izdelovalcev grafičnih programov na svetu.

Leta 1987 Thomas Knoll začne pisati prvi grafični program na Mac Plus-u. Nato še v istem letu s pomočjo svojega brata Johna ustvarita program imenovan "Display". Leto kasneje pa program Display preimnujeta v "ImagePro". V letu 1989 pa se za ImagePro začne zanimati Adobe in tako se program ponovno preimenuje. Tokrat v Adobe Photoshop in začne se razvoj. Tako leta 1990 izide Adobe Photoshop 1.0; prva verzija programa. Še istega leta program dopolnijo in izboljšajo in izide verzija 2.0. Dve leti kasneje, torej leta 1993, ustvarijo verzijo 2.5.1. Prva verzija ki je delovala na PowerPC čipu. Leto pozneje Photoshopu dodajo opcijo plasti (angl. layers) v verziji 3.0. Leta 1998, ko izide verzija 5.0, dodajo zgodovino ("History") ter paletu z barvami. Leto pozneje pa izide pravi prvi "Web Ready" Verzija Photoshopa. Pri tem pa naj posebej poudarimo, da je delo s pomočjo plasti (slika 6) občutno hitrejša in učinkovitejša [11].

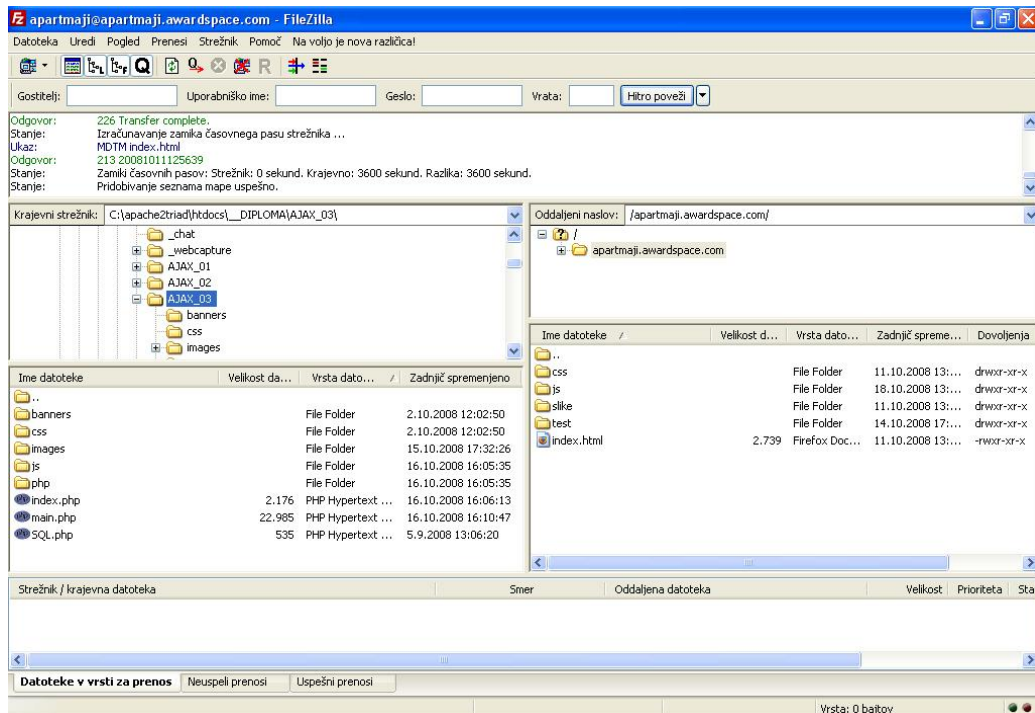


Slika 6. Prikaz programa Photoshop z opcijo plasti (angl. layers).

Orodje Photoshop smo uporabili za izdelavo grafične podobe spletne aplikacije. Na sliki 6 je razvidna orodna vrstica (levo), slika v obdelavi (v sredini) in plasti, ki tvorijo sliko (desno).

3.2.3. Orodje FileZilla Client

FileZilla Client je FTP (angl. File Transfer Protocol) odjemalec, ki nam omogoča enostavno povezavo s FTP strežniki [12]. Program nam omogoča enostavno dodajanje, brisanje in spreminjanje posameznih datotek, ki se nahajajo na oddaljenih strežnikih ter prenos datotek na odjemalca. Poleg tega nudi funkcijo vleci in spusti (angl. drag and drop), ki nam prenos datotek še poenostavi, saj nam ni treba vedeti specifičnih ukazov, ki to omogočajo.



Slika 7. Prikaz programa FileZilla Client.

Slika 7 nam prikazuje bistvo programa FileZilla Client. Na levi polovici slike so prikazane datoteke in mape na našem računalniku, desni del pa prikazuje datoteke in mape, ki se nahajajo na strežniku.

3.2.4. Apache2triad

Apache2triad je paket, ki vključuje različne programe, ki so nam v pomoč pri vzpostavitvi HTTP strežnika in delu s podatkovnimi bazami [13]. Namestitev je relativno enostavna, kar je verjetno razlog za še večjo priljubljenost paketa. Večje težave lahko srečamo pri konfiguraciji strežnika, saj imamo na voljo najrazličnejše opcije.

Izbiramo lahko med strežniki:

- Apache (spletni strežnik)
- MySQL (podatkovni strežnik)
- PostgreSQL (podatkovni strežnik)
- XMail (e-mail strežnik)
- SlimFTP (ftp strežnik)

Orodje podpira prevajalnike za:

- PHP (skriptni jezik)
- Perl (skriptni jezik)
- Python (skriptni jezik)

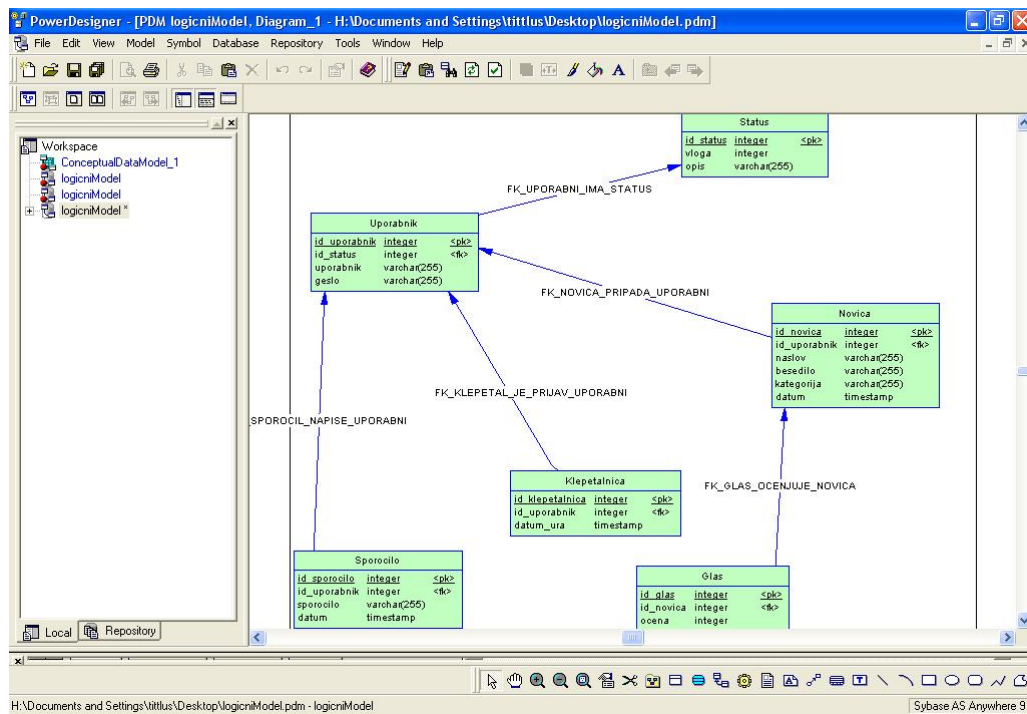
Grafični uporabniški vmesniki pa so:

- Apache2TriadCP (control panel)
- phpMyAdmin (MySQL GUI)
- phpPgAdmin (PostgreSQL GUI)
- phpSQLiteAdmin (SQLite GUI)
- AWStats (webserver monitor)
- UebiMiau (E-mail client)
- PHPXmail (XMail GUI)
- PHPsFTP (SlimFTP GUI)

Od zgoraj naštetih grafičnih uporabniških vmesnikov smo uporabili le phpMyAdmin, ki nam je olajšal upravljanje podatkovne baze MySQL.

3.2.5. PowerDesigner

PowerDesigner je orodje za učinkovito podatkovno modeliranje [14]. Omogoča nam lažjo vizualizacijo, analizo in manipulacijo metapodatkov. Z njim lahko oblikujemo konceptualne modele, ki jih zna avtomatsko pretvoriti v logične. Generira nam lahko tudi skripte za fizično podatkovno bazo, ki jih tvori iz logičnega podatkovnega modela.

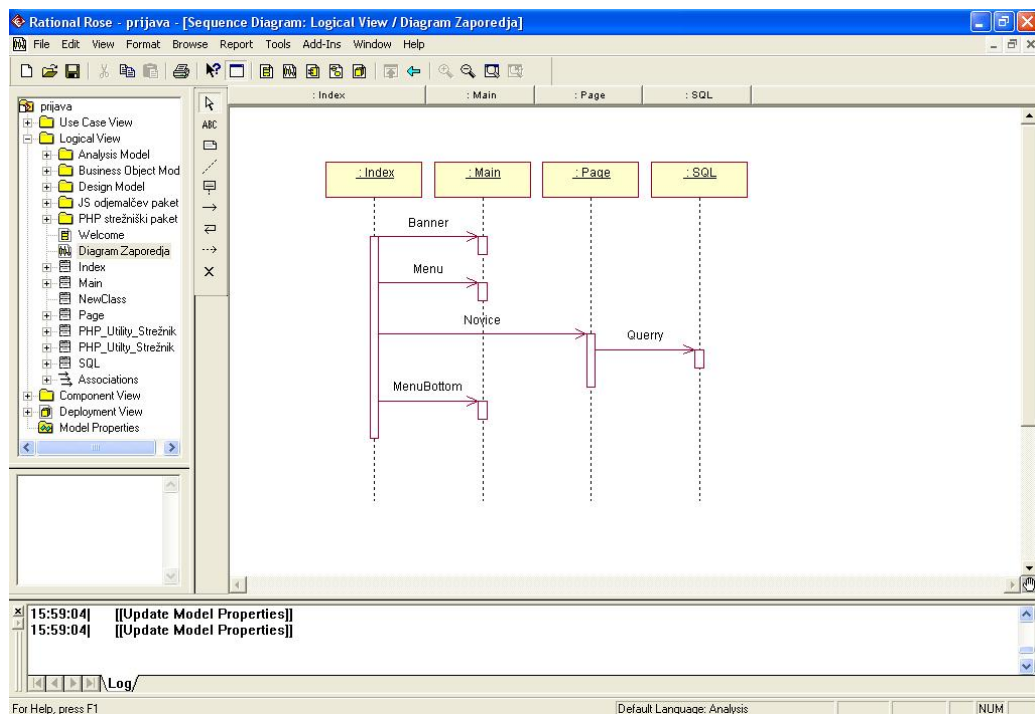


Slika 8. Prikaz programa PowerDesigner.

Slika 8 prikazuje delovno okolje odprtega projekta (levo) in logični podatkovni model za spletno aplikacijo (desno).

3.2.6. Rational Rose

Rational Rose je orodje za vizualno modeliranje, ki podpira diagramske tehnike UML [15]. Diagrami, ki jih zajema, opisujejo strukturo in obnašanje sistema. Implementira množico orodij za razvoj modelov in diagramov za dokumentacijo modelov. Dobrodošla je tudi funkcija spreminjanja modelov v izbrano izvorno kodo. Prav tako podpira metodologijo RUP.



Slika 9. Orodje Rational Rose.

Slika 9 prikazuje uporabo orodja Rational Rose pri izgradnji diagrama zaporedja za našo spletno aplikacijo. Orodje smo uporabili tudi pri izgradnji diagrama primerov uporabe, kjer se je izkazalo za učinkovito in enostavno za uporabo.

4. ANALIZA IN NAČRTOVANJE

V tem poglavju bomo predstavili:

- funkcionalne in nefunkcionalne zahteve sistema,
- diagram primerov uporabe za celotni sistem,
- konceptualni razredni diagram,
- diagrame zaporedja za določene module sistema,
- načrtovanje podatkovne baze in
- strukturo celotne spletne aplikacije.

4.1. Zajem zahtev

4.1.1. Nefunkcionalne zahteve

V nadaljevanju bomo podali nekaj najpomembnejših nefunkcionalnih zahtev sistema (torej tistih zahtev, ki se nanašajo na tehnične ter ostale nevsebinske zahteve sistema).

1. Strojna in programska oprema

Strojna oprema zahteva osebni računalnik, ki ima omogočen dostop do interneta. V okviru programske opreme, potrebujemo HTTP, PHP in MySQL strežnik. Za lažjo namestitev vseh strežnikov priporočamo programski paket Apache2triad. Zagotoviti moramo tudi JS podporo za brskalnik.

2. Performanse

Performanse spletne aplikacije so pogojene s performansi strežnika in odjemalca (vključno s hitrostjo povezave oziroma prenosa podatkov).

3. Dostopnost

Zaradi zagotavljanja čim večje dostopnosti je smiselno uporabiti spletno tehnologijo, ki to v največji meri omogoča. Dostopnost je pogojena z dostopnostjo strežnika in odjemalca.

4. Upravljanje s podatki

Sistem mora zagotoviti učinkovitost tudi v primeru obravnavanja večje količine podatkov, ki so hranjeni v relacijski podatkovni bazi sistema. Le-ta mora zagotoviti učinkovito upravljanje s podatki.

5. Varnost

Sistem mora zagotavljati varnost dostopa do podatkov v podatkovni bazi, ki omogoča določanje pravic na nivoju statusa uporabnikov. Ker gre za aplikacijo za spletne novice, varnost ni najbolj pomembna, saj ne gre za občutljive podatke, ki bi jih zlonamerneži izkoristili oziroma zlorabili.

6. Uporabniški vmesnik

Uporabnost aplikacije zahteva učinkovit uporabniški vmesnik. Ta mora biti ustrezno prilagojen uporabnikom, da lahko v najmanjšem možnem času najdejo željeno novico in/ali jo ažurirajo.

4.1.2. Funkcionalne zahteve

Ker ima diagram primerov uporabe poleg prednosti tudi pomanjkljivost, da so neznani (pred) pogoji, bomo zajem zahtev opisali tudi tekstovno s pomočjo kartic z zgodbami (angl. story card).

Kartice z zgodbami po navadi napiše naročnik sistema. Vsaka kartica vsebuje določeno funkcionalnost sistema, ki jo je treba realizirati. Zaradi lažjega razumevanja kartic z zgodbami bomo najprej predstavili akterje sistema.

Akterji sistema so:

- Gost: neprijavljen obiskovalec spletnih novic
- Uporabnik: prijavljen obiskovalec spletnih novic s statusom 0
- Administrator: prijavljen obiskovalec spletnih novic s statusom 1
- Blokiran uporabnik: neprijavljen obiskovalec spletnih novic s statusom 2

Kartice oziroma funkcionalnost sistema za gosta velja tudi za vse druge akterje, tako kot je funkcionalnost uporabnika del funkcionalnosti administratorja.

1. KARTICA

Gost lahko pregleduje novice, ki so jih dodali uporabniki aplikacije za spletne novice. Poleg naslova in vsebine novice se gostu prikažejo podatki o avtorju, datumu vpisa, oceni ter kategoriji novice.

2. KARTICA

Pri iskanju specifičnih novic je gostu v pomoč iskalnik, kateremu je dodan filter, ki novice išče in izpiše po naslednjih atributih: avtor, naslov, besedilo in datum.

3. KARTICA

Gost ima možnost registracije, preko katere pridobi veljaven uporabniški račun. Pri tem mora podati uporabniško ime in geslo.

4. KARTICA

Po uspešni registraciji, se lahko gost prijavi v sistem.

5. KARTICA

Gost ima možnost enkratnega glasovanja določene novice. Da to omogočimo, si moramo zapomniti gostov IP (angl. Internet Protocol address) naslov.

6. KARTICA

Uporabnik ima možnost varne odjave iz sistema. Namen odjave je preprečevanje zlorabe uporabniškega računa.

7. KARTICA

Uporabnik lahko dodaja novice, pri tem mora podati naslov, kategorijo, in besedilo novice. Avtor in datum vpisa novice se vpišeta avtomatsko.

8. KARTICA

Če se uporabnik, pri vnosu novice slučajno zmoti, ima možnost popravka novice.

9. KARTICA

Uporabniku dopuščamo tudi možnost izbrisa lastnih novic.

10. KARTICA

Uporabnik ima na voljo arhiv, kjer se mu prikažejo vse novice, ki jih je napisal in uspešno dodal.

11. KARTICA

Če želimo komunicirati z ostalimi uporabniki, imamo možnost interaktivne komunikacije v klepetalnici (samo v AJAX izvedbi).

12. KARTICA

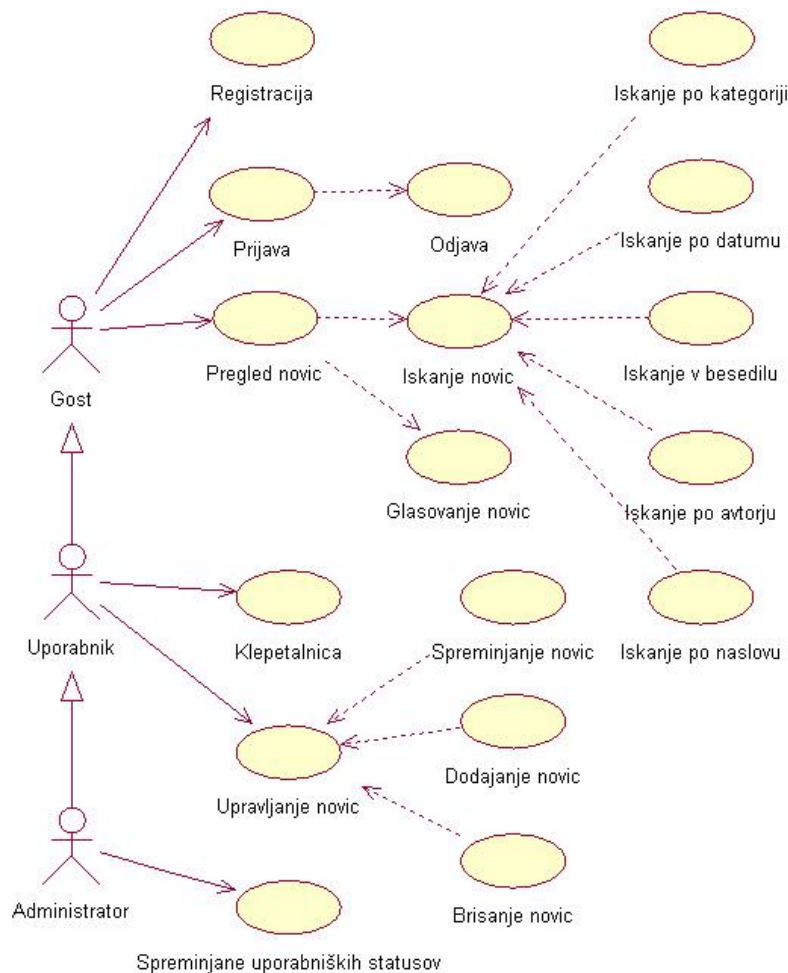
Administrator lahko spreminja in izbriše novice celotnega sistema.

13. KARTICA

Administrator ima možnost spreminjanja statusa vsem uporabnikom. Status lahko obsega naslednje vrednosti:

- 0 - uporabnik,
- 1 - administrator ali
- 2 - blokiran uporabnik.

Zaradi lažje predstavitve funkcionalnosti sistema, bomo v nadaljevanju podali tudi diagram primerov uporabe (angl. use case diagram) (slika 10).



Slika 10. Diagram primerov uporabe za spletne novice.

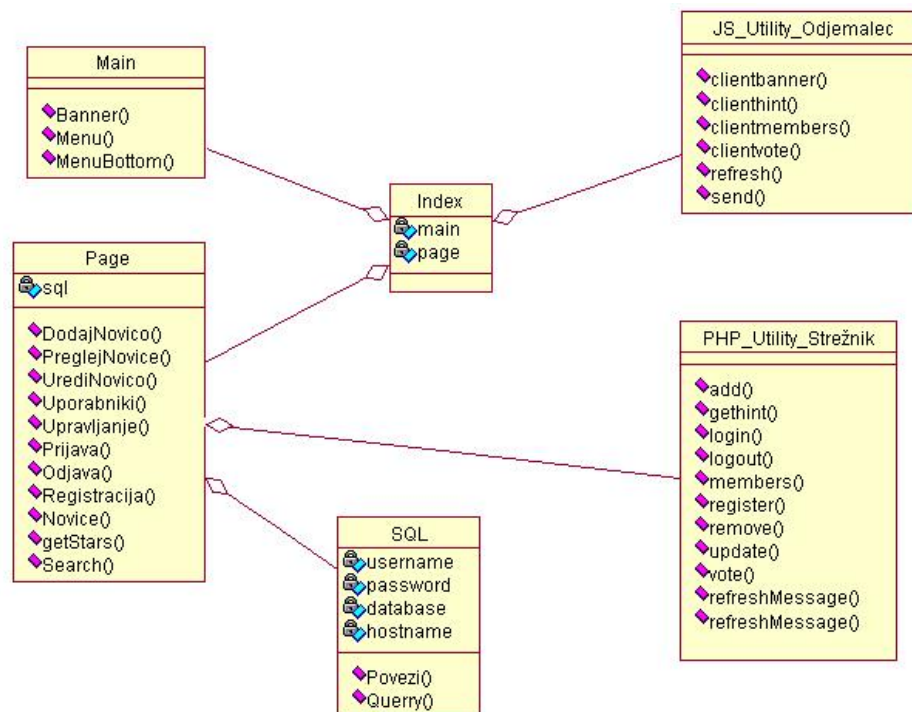
Slika 11 prikazuje prioriteto realizacij posameznih kartic z zgodbami, glede na posamezne izdaje aplikacije za spletne novice. V prvi izdaji implemeniramo ogrodje strani in vse, kar potrebujemo za prikaz novic, ki se nahajajo v podatkovni bazi. V drugi izdaji dodamo še možnost registracije, prijave in odjave. Tretja izdaja dopolnjuje prikaz novic z njihovim glasovanjem in poskrbi za implementacijo iskalnika novic. V četrti izdaji dodamo še vse ostale funkcionalnosti aplikacije, razen klepetalnice, ki jo implementiramo v zadnji izdaji.

IZDAJA	KARTICA
1.	1
2.	3,4,6
3.	2,5
4.	7,8,9,10,12,13
5.	11

Slika 11. Kartice, ki pridejo v poštev pri posamezni izdaji sistema.

4.2. Razredni diagram za celoten sistem

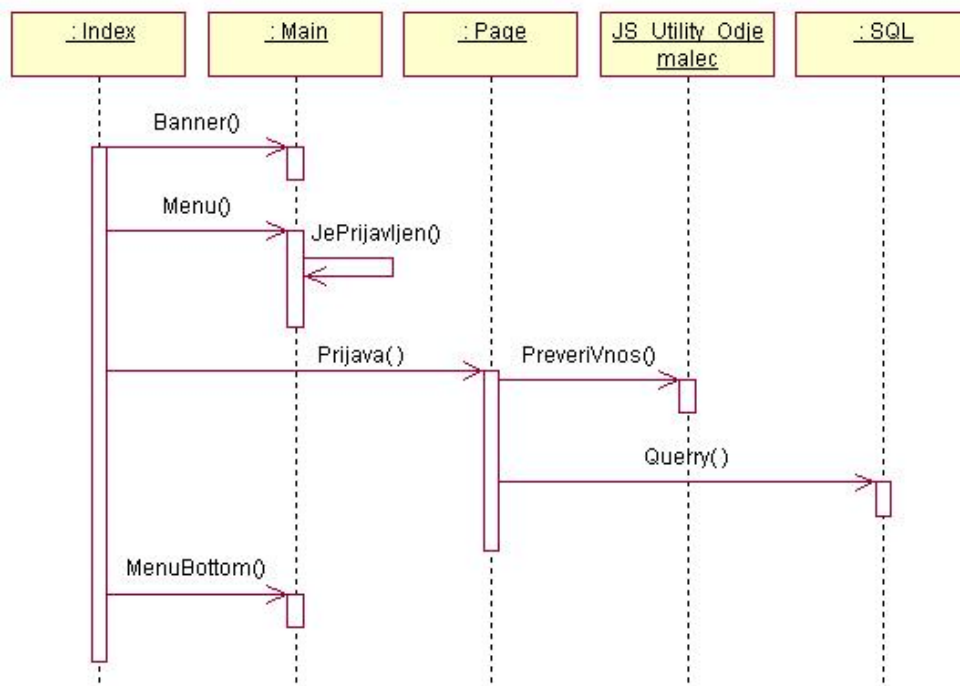
Razredni diagram predstavlja statično strukturo sistema s pomočjo razredov, njihovih metod in atributov.



Slika 12. Razredni diagram celotnega sistema.

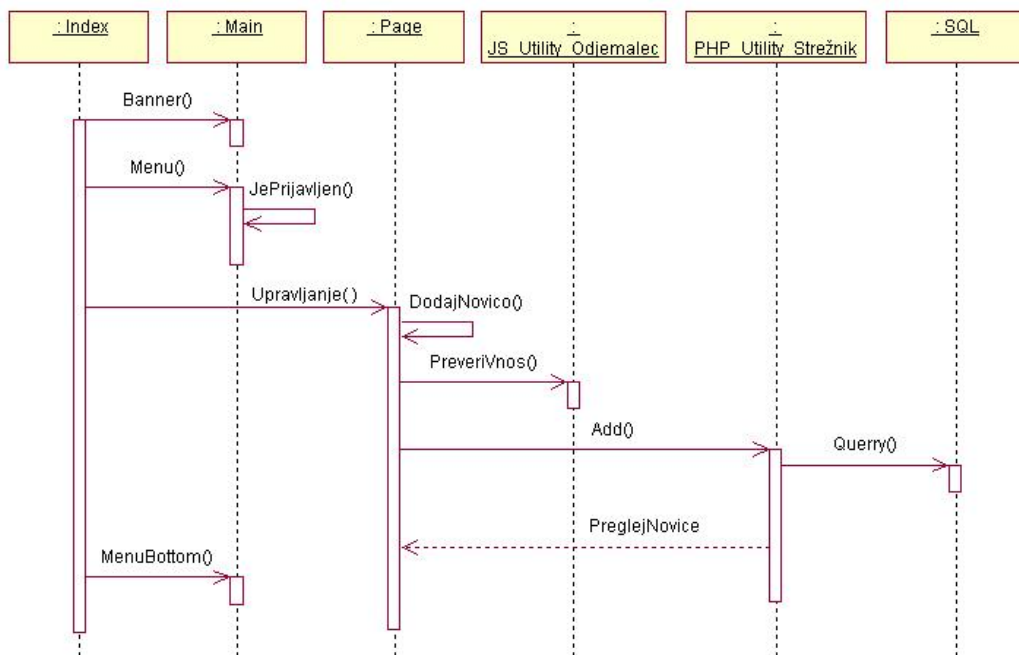
Atributi, metode in razredi so prikazani na sliki 12. `Index` je glavni razred, ki predstavlja ogrodje naše aplikacije. `Main` razred služi prikazu pasice, navigacije na prvem nivoju in avtorskih pravic. Razred `Page` poskrbi za prikaz vseh podstrani aplikacije in implementacijo nekaterih metod, kot so: iskalnik, odjava in povprečna ocena novice. Za povezavo s podatkovno bazo uporabimo razred `SQL`.

4.3. Diagrama zaporedja



Slika 13. Diagram zaporedja za prijavo v sistem.

Slika 13 prikazuje časoven potek dogodkov, ki so potrebni za prikaz aplikacije in prijavo v sistem. Na samem začetku razred `Index` pokliče dve metodi razreda `Main`. Prva metoda poskrbi za prikaz pasice, medtem ko druga za prikaz navigacije na prvem nivoju. Sledi klic funkcije razreda `Page`, ki prikaže vnosni obrazec za prijavo in preveri uspešnost prijave s pomočjo razreda `SQL`. Na samem koncu pokličemo še metodo `MenuBottom` razreda `Main`, ki poskrbi za prikaz avtorskih pravic.



Slika 14. Diagram zaporedja za vnos sveže novice.

Slika 14 prikazuje časoven potek klicev metod, ki so potrebne za vnos sveže novice v sistem. Metode opisane v prejšnjem razdelku ne bomo še enkrat opisovali. Metoda Upravljanje v bistvu poskrbi za ažuriranje novic. Vsebuje tudi podfunkcijo DodajNovico, ki je nujna za dodajanje novic. Preden se novica dejansko doda preko metode Add razreda PHP_Utility_Strežnik, razred JS_Utility_Odjemalec poskrbi za celovitost vnesenih podatkov. Po uspešnem vnosu novice v podatkovno bazo, nas sistem preusmeri na podstran pregleda novic.

4.4. Podatkovna baza

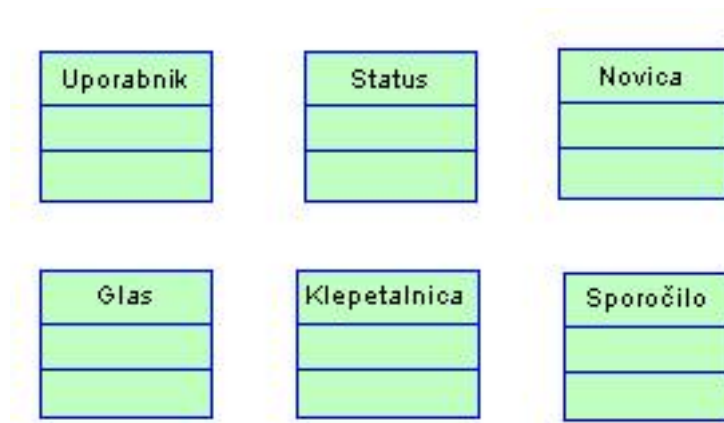
Podatkovna baza je pomemben element, brez katerega bi si težko oz. nemogoče predstavljali dinamične spletne aplikacije.

Izdelava podatkovne baze poteka v štirih korakih:

- globalni model,
- konceptualni model,
- logični model,
- fizični model.

4.4.1. Globalni model

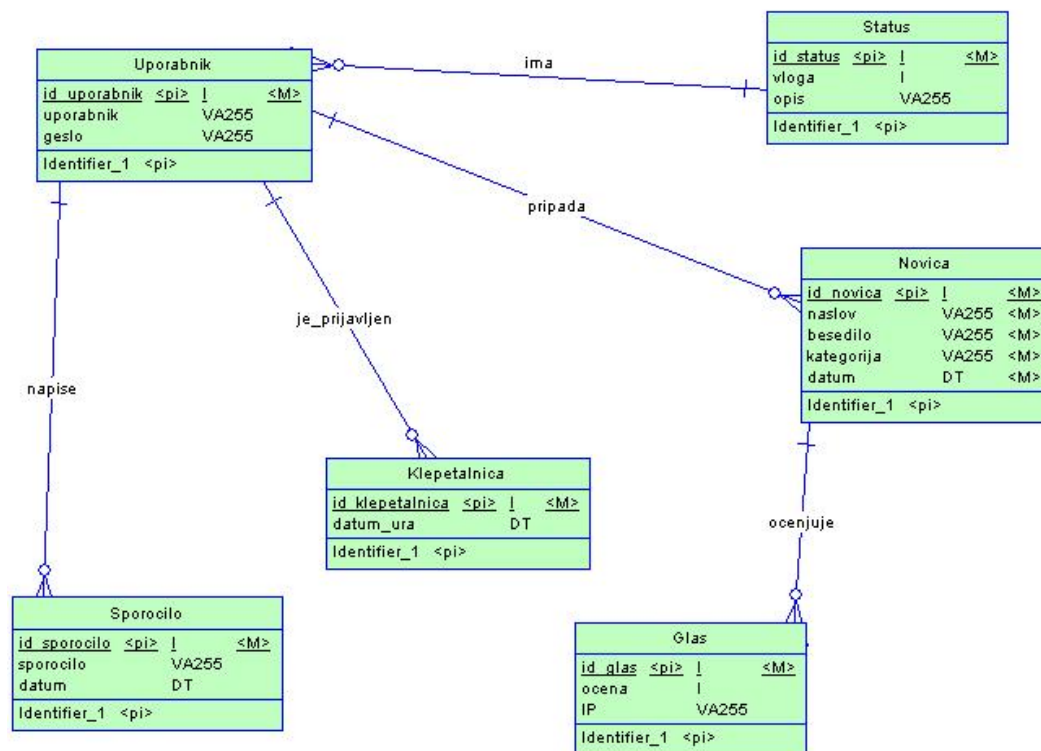
Globalni podatkovni model opravimo z analizo realnega procesa. Obsega grafični prikaz podatkovne strukture s pomočjo entitetnega diagrama (slika 15) in opis entitet. Zaradi večje preglednosti, bomo entite opisali v naslednjem podpoglavju.



Slika 15. Entitetni diagram.

4.4.2. Konceptualni model

Konceptualni podatkovni model opravimo z določitvijo entitet, ki nastopajo v samem procesu. Entitetam iz globalnega podatkovnega modela dodamo attribute, enolične identifikatorje in povezave do drugih entitet, če jih primer vsebuje.



Slika 16. Konceptualni podatkovni model.

V nadaljevanju bomo podrobneje opisali posamezne entitetne tipe konceptualnega podatkovnega modela s slike 16.

1. Uporabnik

Enolični identifikator entitetnega tipa Uporabnik je `id_uporabnik`, ki praktično pomeni zaporedno številko registriranega uporabnika v sistemu. Poleg tega vsebuje tudi podatke, ki so nujno potrebni za prijavo v sistem, torej uporabnika in geslo.

2. Status

Namen entitetnega tipa Status je razlikovanje med pravicami posameznih uporabnikov. Če bi bila aplikacija namenjena samo eni vrsti uporabnikov, tega tipa ne bi potrebovali. Poleg enoličnega identifikatorja `id_status` vsebuje še dva atributa: `vloga`, ki ponazarja pravice posameznika in `opis`, ki je namenjen njeni razlagi.

3. Novica

Entiten tip Novica je eden izmed najpomembnejših. Vsebuje vse podatke, ki so potrebni za vnos in prikaz novic. Atribut `id_novica` enolično določa vsako novico. Da bi vedeli kdaj je novica dodana, nam služi atribut `datum`. Ostali atributi so: `naslov`, `besedilo` in `kategorija`.

4. Glas

Ker ima vsak obiskovalec aplikacije za spletne novice pravico enkratnega glasovanja novice, potrebujemo entiteto `Glas`, kjer hranimo podatke o tem, kdo je glasoval in kakšno oceno je dal določeni novici.

5. Klepetalnica

Entitetni tip `Klepetalnica` uporablja le sistem, ki je realiziran v AJAX-u. Potrebujemo ga, da prikažemo, kdo se trenutno nahaja na podstrani klepetalnica.

6. Sporočilo

Ker je klepetalnica brez sporočil neuporabna potrebujemo tudi entitetni tip `Sporočilo`. Poleg sporočila vsebuje tudi časovno označbo ter uporabnika, ki je sporočilo napisal.

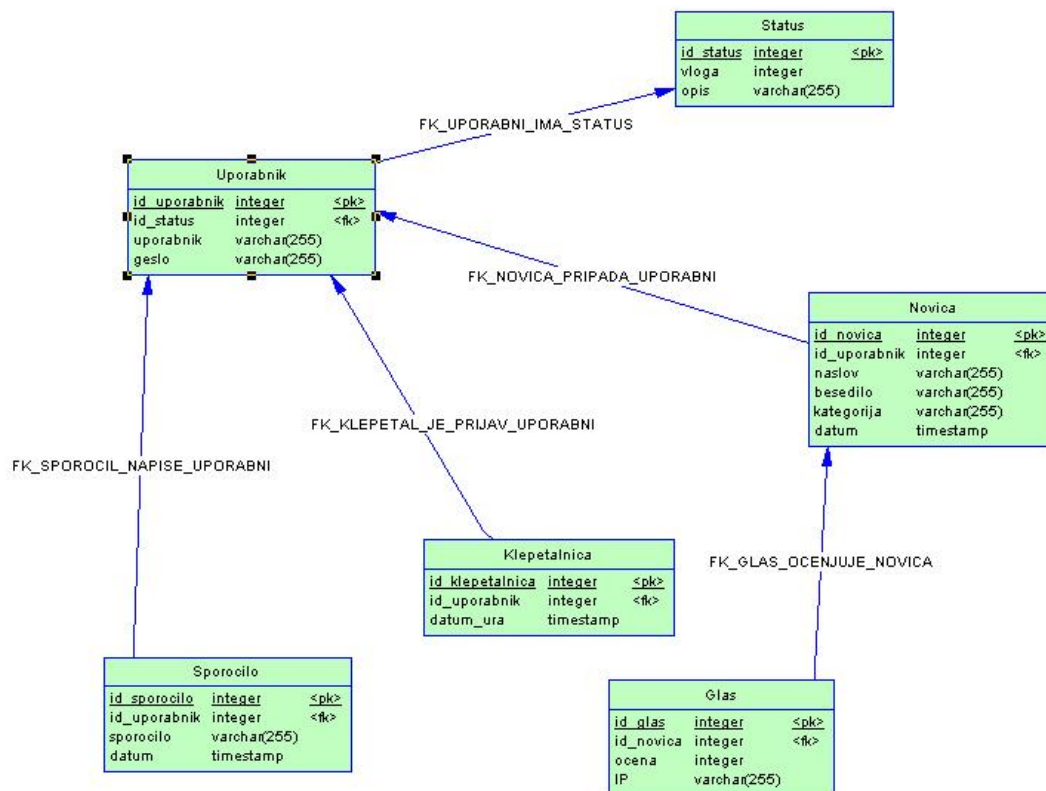
4.4.3. Logični model

Pretvorba iz konceptualnega v logični podatkovni model je pravzaprav pretvorba E-R (entitetno relacijskega) diagrama v relacijski podatkovni model (logični). Logični podatkovni model določa sistem za upravljanje s podatkovno bazo.

Poznamo več vrst logičnih podatkovnih modelov:

- hierarhični,
- mrežni ali
- relacijski.

Kot smo nakazali že v poglavju tehnologije in orodja, bomo v našem primeru izbrali relacijski podatkovni model, katerega podpira tudi MySQL.



Slika 17. Logični podatkovni model.

Slika 17 prikazuje logični podatkovni model, ki ga je avtomatsko generiralo orodje PowerDesigner. Pri tem je upoštevalo naslednja pravila:

- pretvorba enoličnih identifikatorjev v ključe,
- pretvorba entitet v tabele,
- pretvorna atributov v stolpce tabel,
- pretvorba povezav s kardinalnostjo:
 - 1:Y v tuj ključ in
 - X:Y v vmesno tabelo.

4.4.4. Fizični model

Fizični podatkovni model prikazuje organizacijo podatkov v dejanski obliki, v kateri bodo shranjeni v bazi podatkov. V našem primeru ga ni potrebno ustvariti ročno, saj orodje PowerDesigner vsebuje funkcijo, s pomočjo katere lahko iz logičnega podatkovnega modela ustvarimo fizični podatkovni model.

Primer skripte fizičnega podatkovnega modela, za tabelo Glas:

```

/*=====*/
/* Table: Glas */
/*=====*/
create table Glas
(
  id_glas          int          not null,
  id_novica        int          not null,
  ocena            int,
  IP               varchar(255),
  primary key (id_glas)
)
type = InnoDB;

```

Skripto za fizični podatkovni model vnesemo v MySQL bazo preko funkcije uvozi (angl. import). Skripta nam nato generira prazno tabelo Glas z naslednjimi stolpci:

- id_glas tipa int, ki predstavlja enoličen ključ,
- id_novica tipa int, ki predstavlja tuj ključ,
- ocena tipa int in
- IP tipa varchar.

4.5. Struktura spletne aplikacije

Struktura spletne aplikacije predstavlja hierarhijo izvedenih podstrani celotnega sistema, ki jih uporabnik lahko vidi preko HTML in CSS kode. Iz spodnje strukture je razvidna tudi globina spletne aplikacije.

Struktura spletne aplikacije:

- Pregled novic (vključno z iskanjem in glasovanjem novic)
- Registracija
- Prijava
- Ostalo
 - Dodaj novico
 - Preglej lastne novice
 - Spremeni lastne novice
 - Izbriši lastne novice
 - Sprememba statusa uporabnikov
 - Klepetalnica

Pravilna izbira strukture spletne aplikacije je zelo pomembna, saj bistveno vpljiva na učinkovitost načina navigacije in posredno učinkovitost celotnega sistema.

5. IZVEDBA IN PRIMERJAVA MODULOV V PHP IN AJAX RAZLIČICI

Modula glasovanja novic in spreminjanje statusa uporabnikov sta izvedena v PHP in AJAX različici. V našem primeru gre predvsem za izvedbo PHP različice in njeno nadgradnjo s pomočjo tehnologije AJAX. Koda modulov je zaradi večje preglednosti priložena v prilogi, kjer prikazujemo le razlike med PHP in AJAX različico.

5.1. Glasovanje novic

Glasovanje novic je modul, ki nam omogoča spremljanje priljubljenosti novic. Poskrbi za grafični prikaz ocen ter števila glasov novic in dopušča možnost njihovega glasovanja. Podatke, ki jih modul potrebuje, hranimo v MySQL tabeli `Glas`.

1. Razlaga kode PHP različice glasovanja novic iz priloge 9.1.:

- `GetStars`
Argument PHP funkcije predstavlja ključ tabele novice, za katero prikažemo povprečno oceno (z zvezdicami) in število glasov. Novico lahko ocenimo s klikom na eno izmed zvezdic, ki nas s pomočjo HTML povezave preusmeri na `./php/vote.php`.
- `./php/vote.php`
Če uporabnik glasuje prvič določeno novico, shranimo njegov glas v podatkovno bazo. Sledi preusmeritev na podstran Novice, kjer se nam ob glasovani novici prikaže vrnjeno sporočilo.

2. Razlaga kode AJAX različice glasovanja novic iz priloge 9.2.:

- `GetStars`
Funkcija ostaja večinoma nespremenjena. Razlika je le v preusmeritvi (ko kliknemo na eno izmed zvezdic), ki se v PHP izvedbi izvrši s pomočjo HTML povezave, pri AJAX izvedbi pa kličemo JS funkcijo `Vote`.
- `Vote`
JS funkcija skrbi za klic in prenos argumentov v `./php/vote.php` skripto.

- `./php/vote.php`
Če uporabnik glasuje prvič določeno novico, shranimo njegov glas za to novico v podatkovno bazo. Sledi pridobitev svežih podatkov o številu glasov in povprečni oceni za določeno novico ter osvežitev grafične podobe (zvezdice), ki jo vrnemo kot izhod funkcije preko PHP ukaza `echo`.
- `stateChanged`
JS funkcija se sproži, ko strežnik (`./php/vote.php` skripta) vrne odgovor. Poskrbi tudi za osvežitev tekstovnih in grafičnih podatkov.

5.2. Spreminjanje statusa uporabnikom

Modul smo razvili zaradi potrebe po različnih vrstah uporabnikov spletne aplikacije. Administratorjem omogoča spreminjanje statusa vsem uporabnikom sistema. Podrobnosti o različnih statusih uporabnikov so bile že opisane v analizi in načrtovanju in sicer v podpoglavju funkcionalne zahteve. Vse potrebne podatke, ki jih potrebuje modul hranimo v MySQL tabeli Glas in tabeli Uporabnik, ki vsebuje ključ tabele Glas (tuj ključ).

1. Razlaga kode PHP različice spreminjanja statusa uporabnikov iz priloge 9.3.:

- `Uporabniki`
Funkcija skrbi za pridobitev in prikaz podatkov in prikaz grafičnega uporabniškega vmesnika za spreminjanje statusa uporabnikov. Funkcija prikaže uporabniško ime in status uporabnika. Če želimo uporabniku spremeniti status, kliknemo na okrogel gumb, ki nas s pomočjo HTML povezave preusmeri na spodnjo skripto.
- `./php/members.php`
Skripto preveri pravice uporabnika, ki želi spremeniti status določenemu uporabniku. Če so pravice zadostne, to tudi naredi s spremembo vpisa v podatkovno bazo. Sledi preusmeritev na prejšnjo podstran.

2. Razlaga kode AJAX različice spreminjanja statusa uporabnikov iz priloge 9.4.:

- `Uporabniki`
Funkcija ostaja bistveno nespremenjena. Razlika je spet v preusmeritvi, ki se v tem primeru izvrši neposredno preko spodnje JS funkcije.
- `setStatus`
Funkcija služi prenosu argumentov in klicu spodnje PHP skripte.

- `./php/members.php`
Funkcija se razlikuje od AJAX izvedbe, po tem da vrne osvežen grafični uporabniški vmesnik.
- `stateChanged1`
JS funkcija, ki se sproži, ko zgornja skripta vrne odgovor. Namenjena je osveževanju določenih podatkov, v tem primeru podatkov o uporabnikih.

5.3. Ugotovitve

V nadaljevanju sledi predstavitev glavnih prednosti in slabosti obeh izvedenih različic, ki veljajo za oba zgoraj predstavljena modula.

Prednosti PHP različice:

- enostavnejša in hitrejša realizacija,
- manjši obseg kode,
- enostavna povezljivost s podatkovno bazo MySQL.

Slabosti PHP različice:

- ponovna naložitev celotne strani,
- počasnejše delovanje,
- izguba fokusa (sistem nas vrže na vrh strani).

Prednosti AJAX različice:

- osvežitev le potrebnih podatkov,
- hitrejše delovanje,
- fokus ostaja nespremenjen.

Slabosti AJAX različice:

- težavna predvsem za začetnike,
- časovno bolj potratna realizacija (večji obseg kode).

6. PREDSTAVITEV IN UPORABA SPLETNIH NOVIC

Uporabniški vmesnik spletnih novic predstavlja večino funkcionalnosti sistema. V grobem ga delimo na tri področja:

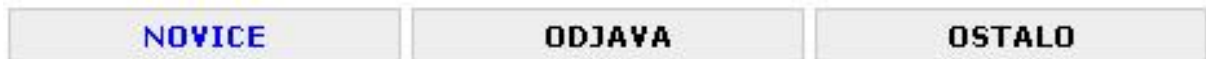
- na navigacijski meni,
- vnosne obrazce,
- sporočila o napakah.

6.1. Navigacija na prvem nivoju

Z navigacijo na prvem nivoju lahko dostopamo do različnih vsebin podstrani spletnih novic. Naj omenimo, da je vsebina navigacije, ki je namenjena gostu različna od tiste, ki je namenjena uporabniku ali administratorju (slika 18,19).



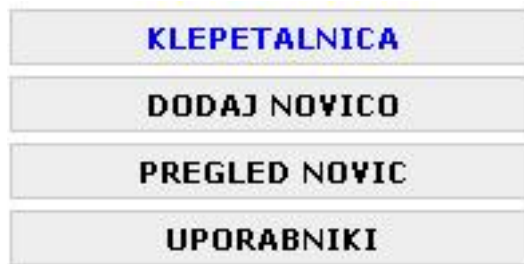
Slika 18. Navigacija gosta na prvem nivoju.



Slika 19. Navigacija uporabnika na prvem nivoju.

6.2. Navigacija na drugem nivoju

Vsebina navigacije na drugem nivoju (slika 20) se prikaže le uspešno prijavljenim obiskovalcem. Razdelimo jo lahko na tisto, ki se prikaže uporabniku in na tisto, ki se prikaže administratorju. Ločitev je nujna, saj je funkcija spreminjanje statusa uporabnikov na voljo le administratorju.



Slika 20. Navigacija administratorja na drugem nivoju.

6.3. Registracija

Registracija je namenjena generiranju novega uporabniškega računa. Kot je razvidno iz slike 21 mora uporabnik vnesti novo geslo dvakrat. Tak pristop je smiseln, saj bi se lahko uporabnik pri vnosu novega gesla zatipkal, kar bi mu onemogočilo nadaljnjo prijavo.

 A registration form with a light gray background and a thin border. At the top, it has a header section with the text ': : REGISTRACIJA'. Below the header are three input fields: the first is labeled 'Uporabnik:', the second is labeled 'Geslo:', and the third is also labeled 'Geslo:'. At the bottom of the form is a button with the text '>> VSTOP <<'.

Slika 21. Vnosni obrazec za registracijo v sistem.

Pri registraciji lahko pride do naslednjih napak in s tem povezanih sporočil:

- Sporočilo o napaki neustrezne dolžine uporabniškega imena ali gesla,
- Sporočilo o napaki zasedenega uporabniškega imena,
- Sporočilo o napaki neujemanja gesel.

6.4. Prijava

S prijavo v sistem gost pridobi možnost upravljanja lastnih novic in interaktivnega dopisovanja v klepetalnici. To stori s pomočjo prijavnega vnosnega obrazca iz slike 22.

Slika 22. Vnosni obrazec za prijavo v sistem.

Prijava je uspešna, če se nahaja v podatkovni bazi natanko tako uporabniško ime in geslo, kot ga uporabnik vtipka v prijavnih vnosni obrazec.

6.5. Ažuriranje novic

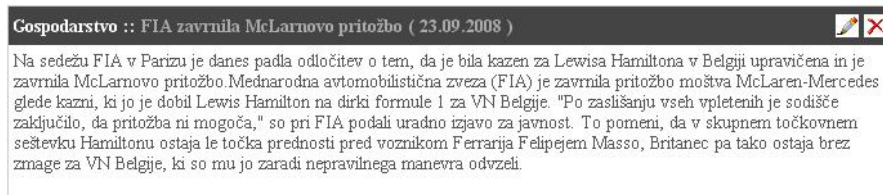
6.5.1. Dodajanje novic

Možnost dodajanja novic je dana vsem prijavljenim obiskovalcem sistema, tako navadnim uporabnikom, kot administratorjem. Novico dodamo s pomočjo vnosnega obrazca za dodajanje novic, ki je prikazan na sliki 23. Uspešno dodana novica, se prikaže na glavni strani Novice, kjer jo lahko vsi preberejo.

Slika 23. Vnosni obrazec za dodajanje novic v sistem.

6.5.2. Pregled novic

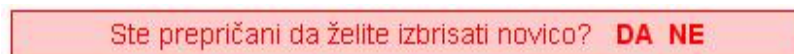
Pregled novic ločimo na dva dela: na tistega za administratorje in tistega za uporabnike. Vendar tukaj ne gre za razliko o razširjeni funkcionalnosti, temveč razlika tiči v tem, da administrator lahko upravlja z vsemi novicami, uporabnik pa le z lastnimi. Kot nam prikazuje slika 24 vsaki novici pripada kategorija, naslov, datum, besedilo ter dva gumba. Levi (pisalo) služi preusmeritvi na preurejanje novice, desni (x) pa izbrisu novice.



Slika 24. Pregled novice.

6.5.3. Brisanje novic

Kot je bilo opisano v prejšnjem razdelku dosežemo brisanje novice s klikom na rdeč gumb v obliki črke x (slika 24). Preden se novica dejansko izbriše, nas sistem vpraša za potrditev (slika 25). Po potrditvi se novica izbriše in njen priklic ni več mogoč.



Slika 25. Potrditev brisanja novice.

6.5.4. Preurejanje novic

Sam proces preurejanja novic je dokaj podoben tistemu za njihovo dodajanje. V bistvu gre za isti vnosni obrazec, razlika je le v tem, da se pri preurejanju novic, avtomatsko spomnijo predhodno vneseni podatki, ki se nahajajo v podatkovni bazi.

6.6. Statusi uporabnikov

Vsak uporabnik ima določen status (slika 26), katerega namen je ločevanje med različnimi akteriji sistema, glede na njihove pravice. Status uporabnika lahko zajema eno izmed naslednjih vrednosti:

- navaden uporabnik (zelen krogec),
- administrator (moder krogec),

- blokiran uporabnik (rdeč krogec).

1. uporabnik : 2	status : Blokiran uporabnik			
2. uporabnik : 24ur	status : Navaden uporabnik			
3. uporabnik : Tittl	status : Administrator			
4. uporabnik : TittluS	status : Navaden uporabnik			

Slika 26. Spreminjanje in prikaz statusa uporabnikov.

6.7. Klepetalnica

Klepetalnica je izvedena samo v AJAX različici spletnih novic. Njena implementacija je v PHP verziji izpuščena, saj bi bila nesmiselna zaradi samega načina osveževanja aplikacije.



Slika 27. Klepetalnica namenjena uporabnikom spletnih novic.

Slika 27 prikazuje klepetalnico, ki jo na grobo lahko razdelimo na tri področja:

- na prikaz uporabnikov klepetalnice (desno),
- na prikaz kratkih sporočil uporabnikov, ki vsebujejo tudi ime avtorja in uro (levo),
- na del z vnosnim poljem in gumbom, ki služi dodajanju novih kratkih sporočil (spodaj).

6.8. Iskalnik novic

Iskalnik novic je namenjen hitremu iskanju novic. Vsebuje vnosno polje, kamor vtipkamo iskalni niz, izbirno polje, ki določa po katerem atributu se bo izvedlo iskanje, gumb za pričetek iskanja ter prikaz števila zadetkov. AJAX različica iskalnika, nam ob vnosu iskalnega niza prikaže tudi morebitne predloge (slika 28).



Slika 28. Iskalnik novic.

Če iskalnik ne najde nobene novice, ki ustreza iskalnim pogojem, nas o tem primerno obvesti.

6.9. Glasovanje novic

Zaradi lažjega spremljanja priljubljenosti novic (in morda nekoč nagrajevanja najuspešnejših) je vsem obiskovalcem spletnih novic na voljo njihovo glasovanje. Glas je oddan s klikom na eno izmed petih zvezdic (slika 29).



Slika 29. Glasovanje novic.

7. ZAKLJUČEK

V zadnjem delu diplomske naloge bomo podali ugotovitve. Primerjali bomo predvsem obe tehnologiji in ocenili orodja, ki smo jih uporabili pri izdelavi, ter zapisali njihove prednosti in slabosti.

7.1. Ocena uporabljenih orodij

Orodja, ki so olajšala razvoj spletne aplikacije za spletne novice, se niso razlikovala med PHP in AJAX izvedbo.

V obeh primerih smo uporabili:

- za grafiko: Photoshop,
- za podatkovno bazo in upravljanje s podatki: mySQL,
- za programiranje: Notepad++,
- za modeliranje : PowerDesigner in Rational Rose.

Večina uporabljenih orodij se je izkazala za pravilno in učinkovito izbiro. Nadomestili bi lahko le orodje Photoshop z eno izmed zastojnih različic (npr. Gimp) in orodje Notepad++ z enim izmed programov za izdelavo spletnih aplikacij (npr. Dreamweaver ali FrontPage).

7.2. Primerjava tehnologij

Ugotovili smo, da AJAX omogoča nov način razvoja spletnih aplikacij, ki povečuje njihovo interaktivnost, hitrost delovanja in uporabnost. Zato ni nič nenavadnega, da se omenjene tehnologije in tehnike pri razvijanju najsodobnejših rešitev poslužuje vedno več programerjev spletnih aplikacij.

Večina popularnih spletnih aplikacij iz generacije Web 2.0 uporablja AJAX (npr google maps ali facebook). V prihodnosti lahko pričakujemo še večjo uporabo AJAX-a s strani razvijalcev sodobnih dinamičnih spletnih aplikacij.

PHP prednosti:

- deluje lahko na večini operacijskih sistemov,
- lahka integracija z podatkovno bazo (npr. MySQL),
- lahka uporaba (prijazen do začetnikov),
- spletne aplikacije lahko razvijemo zelo hitro,
- aplikacije so zelo stabilne in neodvisne od brskalnika,
- veliko literature in pomoči iz spletnih skupnosti,
- brezplačna uporaba.

PHP slabosti:

- slaba podpora za OOP (angl. Object Oriented Programming),
- skromen opis napak, če se le-te pojavijo,
- neuporaben na odjemalčevi strani.

AJAX prednosti :

- aplikacija je bolj prijazna do uporabnika,
- ponuja nove funkcionalnosti (npr. klepetalnica).
- aplikacija se hitreje odziva,
- med strežnikom in odjemalcem se pošilja občutno manjše število podatkov, saj aplikacije ni potrebno ponovno naložiti v celoti ter
- ni potrebe po predhodni namestitvi vtičev (angl. plug-in), za pravilno delovanje.

AJAX slabosti oziroma pomanjkljivosti :

- izdelava takih aplikacij zahteva znanje:
 - enega izmed strežniških jezikov (v našem primeru PHP),
 - enega izmed odjemalčevih jezikov (v našem primeru JS),
 - enega izmed označevalnih jezikov (v našem primeru HTML) in
 - poznavanje osnove tehnologije DOM (angl. Document Object Model).
- potrebna je ugotovitev tipa brskalnika, saj se klici XMLHttpRequest objekta, potrebne za asinhrono klice, razlikujejo od brskalnika do brskalnika.
- če vsebuje JS koda napake, nam jo brskalnik ne javi, kot se zgodi pri PHP skripti.
- podatki se osvežujejo samo znotraj aplikacije, zato je njena navigacija z brskalnikom (naprej, nazaj,..) otežena,
- brskalniki, ki ne uporabljajo skriptnega jezika JS ali podobnega, imajo težave s prikazom aplikacije ter pravilnim delovanjem,
- spremenljivke shranjene s pomočjo JS, se pri osvežitvi s pomočjo gumba brskalnika, ponovno nastavijo na privzete vrednosti.

7.3. Sklep

Diplomska naloga prikazuje smernice, orodja, tehnike in tehnologije, ki jih programer potrebuje pri razvoju interaktivnih spletnih aplikacij.

Pri razvoju spletne aplikacije za spletne novice, nismo naleteli na bistvene težave.

Spletna aplikacija za spletne novice je v splošnem namenjena informiranju uporabnikov s pomočjo kratkih novic določenih kategorij. Ažurnost, globina (št. novic iste kategorije) in širina (št. različnih novičarskih kategorij) novic je odvisna od uporabnikov samih.

Kot smo pričakovali je spletna aplikacija, ki je bila razvita s pomočjo tehnike AJAX naprednejša od aplikacije (samo) v PHP tehnologiji. Kljub prednostim AJAX-a, moramo paziti, da ne pretiravamo z njegovo uporabo. Naj za primer navedemo funkcijo osveževanja sporočil klepetalnice, ki je lahko ob velikem številu uporabnikov zelo obremenjujoča za strežnik.

V nadaljevanju podamo določene ideje za izboljšavo spletne aplikacije, do katerih smo prišli med samim razvojem in izvedbo. Te so:

- potrditev registracije preko elektronske pošte,
- pošiljanje novih novic uporabnikom preko elektronske pošte,
- pošiljanje kratkih sporočil drugim uporabnikom preko sistema spletne novice,
- RSS modul za spremljanje tekočih novic,
- uporaba napisanih knjižic za JS (npr. Mootools), kar bi občutno zmanjšalo dolžino JS kode.

8. ZAHVALA

Iskreno se zahvaljujem svojemu mentorju dr. Igorju Rožancu za njegovo strokovno pomoč ter koristne nasvete. Posebej naj pohvalim odlično usmerjanje pri opravljanju mojega diplomskega dela.

Zahvaljujem se tudi sošolcu Mateju Goveku, ki si je vzela čas zame in mi pomagal z nasveti in izkušnjami.

Posebno bi se zahvalil tudi celotni družini, še posebej mami in očetu, ki sta me v času študija psihično in finančno podpirala in vseskozi vspodbujala.

9. PRILOGA

9.1. PHP različica glasovanja novic

```
// Lokacija : ./page.php
function GetStars($nid)
{
    $SQL = new SQL;
    $query="SELECT * FROM glas WHERE ID_novica='$nid'";
    $result=$SQL->Query($query);
    $num = mysql_num_rows($result);

    $povprecje=0;

    for($i=0;$i<$num;$i++)
    {
        $ocena = mysql_result($result,$i,"ocena");
        $povprecje += $ocena;
    }
    $star = 0;
    if($num>0)    $star = round( $povprecje/$num );

    $query=$_GET['query'];
    $atribut=$_GET['atribut'];

    for($i=1;$i<6;$i++)
    {
        echo "<a href =
                './php/vote.php?vote=$i&nid=$nid&query=$query&atribut
                =$atribut'
            title='Ocena : $i'>";
        if($i<=$star) echo "<div id='starY'></div>";
        else          echo "<div id='starN'></div>";
        echo "</a>";
    }
    echo "Stevilo glasov : ".$num;

    if($_GET['nid']==$nid) echo "<font color='red'> To novico ste
        že glasovali!</font>";
}
```

```

    if($_GET['ok']==$nid) echo "<font color='blue'> Hvala za vaš
        glas!</font>";
}

// Lokacija : ./php/vote.php
<?php
    include_once("../SQL.php");

    $nid = $_GET['nid'];
    $vote = $_GET['vote'];
    $ip = $_SERVER['REMOTE_ADDR'];
    $query1=$_GET['query'];
    $atribut1=$_GET['atribut'];

    $SQL = new SQL;
    $query="SELECT * FROM glas WHERE ID_novica='$nid' AND
        IP='$ip'";
    $result=$SQL->Query($query);
    $num = mysql_num_rows($result);

    if($num==0)
    {
        $SQL = new SQL;
        $query="INSERT INTO glas(ID,ID_novica,ocena,IP) VALUES('',
            '$nid', '$vote', '$ip)";
        $result=$SQL->Query($query);

        header('Location:
            ../index.php?main=novice&query='.$query1.'&atribut='.$atribut.'&ok='.$nid.'');
    }
    else
    {
        header('Location:
            ../index.php?main=novice&query='.$query1.'&atribut='.$atribut.'&nid='.$nid.'');
    }
?>

```

9.2. AJAX različica glasovanja novic

```
// Lokacija : ./js/clientvote.js
var xmlhttp
var nid

function Vote(i,id)
{
  nid=id
  xmlhttp=GetXmlHttpRequest()

  if (xmlhttp==null)
  {
    alert ("Browser does not support HTTP Request")
    return
  }

  var url =
  "./php/vote.php?nid="+nid+"&vote="+i+"&sid="+Math.random()
  xmlhttp.onreadystatechange=stateChanged
  xmlhttp.open("GET",url,true)
  xmlhttp.send(null)
}

function stateChanged()
{
  if (xmlhttp.readyState==4 || xmlhttp.readyState=="complete")
  {
    if(xmlhttp.responseText!="")
    {
      document.getElementById(nid).innerHTML =
      xmlhttp.responseText
    }
    else
    {
      document.getElementById(nid+"D").innerHTML = "<font
      color='red'> To novico ste že glasovali!</font>"
    }
  }
}

// Lokacija : ./php/vote.php
```

```

<?php
    include_once("../SQL.php");

    $_nid = $_GET['nid'];
    $_vote = $_GET['vote'];
    $_ip = $_SERVER['REMOTE_ADDR'];

    $SQL = new SQL;
    $query="SELECT * FROM glas WHERE ID_novica='$_nid' AND
    IP='$_ip'";
    $result=$SQL->Query($query);
    $num = mysql_num_rows($result);

    $echoResult="";

    if($num==0)
    {
        $SQL = new SQL;
        $query="INSERT INTO glas(ID,ID_novica,ocena,IP) VALUES ('',
        '$_nid', '$_vote',
        '$_ip')";
        $result=$SQL->Query($query);

        $SQL = new SQL;
        $query="SELECT * FROM glas WHERE ID_novica='$_nid'";
        $result=$SQL->Query($query);
        $num = mysql_num_rows($result);
        $povprecje=0;

        for($i=0;$i<$num;$i++)
        {
            $ocena = mysql_result($result,$i,"ocena");
            $povprecje += $ocena;
        }
        $star = 0;
        if($num>0) $star = round( $povprecje/$num );

        for($i=1;$i<6;$i++)
        {
            if($i<=$star)
            {
                $echoResult=$echoResult."<div id='starY'
                onClick='Vote('".$i."', '$_nid.'');></div>";
            }
        }
    }
}

```

```

    }
    else
    {
        $echoResult=$echoResult."<div id='starN'
        onClick='Vote(\".$i.\"','\".$_nid.\"');'></div>";
    }
}

$echoResult = $echoResult. "<div style='float:left;'>
Stevilo
glasov : ".$num."</div>";
$echoResult = $echoResult. "<div style='float:left;'
id='\".$_nid.'D'.'"
name='\".$_nid.'D'.'">".
"<font color='blue'>Hvala za vas glas!</font>". "</div>";

echo $echoResult;
}
?>

```

9.3. PHP različica spreminjanje statusa uporabnikov

```

// Lokacija : ./page.php
function Uporabniki($act,$id)
{
    $status=$_SESSION['status'];

    if($status==1)
    {
        $SQL = new SQL;
        $query="SELECT * FROM uporabnik ORDER BY user ASC";
        $result=$SQL->Query($query);
        $num=mysql_num_rows($result);

        for($i=0;$i<$num;$i++)
        {
            ?>
            <div id='nHeader1'>
                <table width='100%' border='0px'>
                    <tr>

```

```

        <td align='left' id='nNaslov'>
<?
    $user = mysql_result($result , $i, 'user');
    $status = mysql_result($result , $i, 'status');
    $j=$i+1;

    $stat='';
    if($status==0)
        $stat="Uporabnik";
    if($status==1)
        $stat="Administrator";
    if($status==2)
        $stat="Blokiran uporabnik";

    echo "<span id='nDatum'>". $j. ". uporabnik :
        </span>". $user;
    echo "<span id='nDatum'> | status : </span>". $stat;
?>
</td>
<td align='right' id='nDatum'>
    <table>
    <tr>
    <td>
    <?
        if($status==0)
        {
        ?>
        <div id='gumbZ'></div>
            <a title="Administrator" href =
                "./php/members.php?status=1&user=<? echo
                $user; ?>">
            <div id='gumbC'></div>
        </a>
            <a title="Blokiran Uporabnik" href =
                "./php/members.php?status=2&user=<?
                echo $user; ?>">
            <div id='gumbC'></div>
        </a>
    <?
        }
        else if($status==1)
        {
        ?>

```

```

<a title="Uporabnik" href =
    "./php/members.php?status=0&user=<? echo $user;
    ?>">
    <div id='gumbC'></div>
</a>

        <div id='gumbM'></div>
<a title="Blokiran Uporabnik" href =
    "./php/members.php?status=2&user=<? $user;
    ?>">
    <div id='gumbC'></div>
</a>
<?
}
else
{
?>
    <a title="Uporabnik" href =
        "./php/members.php?status=0&user=<? echo
        $user;
        ?>">
            <div id='gumbC'></div>
        </a>
    <a title="Administrator" href =
        "./php/members.php?status=1&user=<? echo
        $user; ?>">
        <div id='gumbC'></div>
    </a>
    <div id='gumbR'></div>
    <?
    }
    ?>
    </td>
</tr>
</table>
</td>
</tr>
</table>
</div>
<div id='spacer'></div>
<?
}
}

```

```

}

// Lokacija : ./php/members.php
<?
    session_start();
    $SessionStatus=$_SESSION['status'];
    session_write_close();

    $user=$_GET['user'];
    $status=$_GET['status'];

    include_once("../SQL.php");
    $SQL = new SQL;

    if($SessionStatus==1)
    {
        $query="UPDATE uporabnik SET status='$status' WHERE
            user='$user'";
        $result=$SQL->Query($query);
    }
    header('Location:
        ../index.php?main=upravljanje&pid=uporabniki');
?>

```

9.4. AJAX različica spreminjanje statusa uporabnikov

```

// Lokacija : ./js/members.js
var xmlHttp
var userID
var status

function setStatus(userStatus,user)
{
    xmlHttp=GetXmlHttpRequest()
    userID=user
    status=userStatus

    if (xmlHttp==null)
    {
        alert ("Browser does not support HTTP Request")
    }
}

```

```

        return
    }

    var url =
        "./php/members.php?userID="+userID+"&status="+status+"&sid
        "+Math.random()
    xmlHttp.onreadystatechange=stateChanged1
    xmlHttp.open("GET",url,true)
    xmlHttp.send(null)
}

function stateChanged1()
{
    if (xmlHttp.readyState==4 ||
        xmlHttp.readyState=="complete")
    {
        if(xmlHttp.responseText!="")
        {
            document.getElementById(userID).innerHTML =
                xmlHttp.responseText

            if(status==0)
                document.getElementById(userID+'D').innerHTML =
                    "Uporabnik"
            if(status==1)
                document.getElementById(userID+'D').innerHTML =
                    "Administrator"
            if(status==2)
                document.getElementById(userID+'D').innerHTML =
                    "Blokiran Uporabnik"
        }
    }
}

// Lokacija : ./php/members.php
<?
    session_start();
    $SessionStatus=$_SESSION['status'];
    session_write_close();

    $userID=$_GET['userID'];
    $status=$_GET['status'];

```

```

include_once("../SQL.php");
$SQL = new SQL;

$refreshedState="";

if($SessionStatus==1)
{
    $query="UPDATE uporabnik SET status='$status' WHERE
    ID='$userID'";
    $result=$SQL->Query($query);

    if($status==0)
    {
        $refreshedState="<div id='gumbZ'
        onClick='setStatus(0, ".$userID.");'></div>".
        "<div
        id='gumbC'
        onClick
        ='setStatus(1, ".$userID.");'></div>".
        "<div
        id='gumbC'
        onClick
        ='setStatus(2, ".$userID.);'></div>";
    }
    if($status==1)
    {
        $refreshedState="<div id='gumbC'
        onClick='setStatus(0, ".$userID.");'></div>".
        "<div id='gumbM'onClick
        ='setStatus(1, ".$userID.);'></div>".
        "<div id='gumbC' onClick
        ='setStatus(2, ".$userID.);'></div>";
    }
    if($status==2)
    {
        $refreshedState="<div
        id='gumbC'
        onClick='setStatus(0, ".$userID.);'></div>".
        "<div id='gumbC' onClick
        ='setStatus(1, ".$userID.);'></div>".
        "<div
        id='gumbR'
        onClick
        ='setStatus(2, ".$userID.);'></div>";
    }
}
echo $refreshedState;
?>

```

10. VIRI

- [1] Kaj je metodologija in zakaj jo potrebujemo? Dostopno na:
<http://www2.gov.si/mju/emris.nsf/0/4F5A54F79681D61AC1256E9E003E08C8?OpenDocument>
- [2] Agilne metodologije razvoja informacijskih. Dostopno na:
<http://bajecm.fri.uni-lj.si/CRP2001/Clanki/Agilne%20Metodologije%20Razvoja%20IS.pdf>
- [3] UML, The Unified Modeling Language. Dostopno na:
<http://marvin.fri.uni-lj.si/KPS/UML%20diagrami.ppt>
- [4] Matjaž Jogan, Bojan Kverh, Aleš Leonardis, Igor Lesjak, Peter Peer, Franc Solina. Uporabniška programska oprema. Založba FE in FRI, 2001.
- [5] PHP. Dostopno na:
<http://sl.wikipedia.org/wiki/Php>
- [6] Cristian Darie, Bogdan Brinzarea, Filip Chereches, Mihai Bucica. AJAX And PHP - Building Responsive Web Applications. Založba PACKT, 2006.
- [7] JS. Dostopno na:
<http://sl.wikipedia.org/wiki/JS>
- [8] MySQL. Dostopno na:
<http://sl.wikipedia.org/wiki/MySQL>
- [9] CSS. Dostopno na:
<http://sl.wikipedia.org/wiki/CSS>
- [10] phpMyAdmin. Dostopno na:
http://www.phpmyadmin.net/home_page/index.php
- [11] Photoshop. Dostopno na:
<http://www.adobe.com/products/photoshop/compare/>
- [12] Filezilla Client. Dostopno na:
<http://filezilla-project.org/>
- [13] Apache2Triad. Dostopno na:
<http://apache2triad.net/>

- [14] PowerDesigner. Dostopno na:
<http://www.sybase.com/products/modelingdevelopment/powerdesigner>
- [15] Rational Rose. Dostopno na:
<http://www-01.ibm.com/software/awdtools/developer/rose/>

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Boštjan Tittl, z vpisno številko 63020303, sem avtor diplomskega dela z naslovom:

PRIMERJAVA RAZVOJA APLIKACIJE V TEHNOLOGIJAH PHP IN AJAX

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom dr. Igorja Rožanca,
- so elektronska oblika diplomskega dela, naslov, povzetek ter ključne besede identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 28.1.2009.

Podpis avtorja: _____