

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Goran Verbuč

# **Spletni portal radijske postaje**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana 2014



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

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*





Št. naloge: 00519 / 2013  
Datum: 5.9.2013

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


Kandidat: **GORAN VERBUČ**

Naslov: **SPLETNI PORTAL RADIJSKE POSTAJE  
WEB PORTAL OF A RADIO STATION**


Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Zasnujte spletni portal za radijsko postajo: opredelite za tovrstni portal potrebne funkcionalnosti in tehnološke zahteve. Tehnološka zasnova naj omogoča uporabo portala tudi na različnih vrstah mobilnih naprav. Portal naj vodstvu radijske postaje omogoča širok nabor pregleda statističnih podatkov in izvajanje različnih analiz. Uporabnikom portala pa naj portal v vsakem trenutku dinamično prikazuje najbolj aktualne vsebine. Portal razvijte v NET tehnologiji.

Mentor:   
doc. dr. Rok Rupnik



Dekan:   
prof. dr. Nikolaj Zimic



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Goran Verbuč, z vpisno številko **63060383**, sem avtor diplomskega dela z naslovom:

*Spletni portal radijske postaje*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika;
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela;
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 7. marca 2014

Podpis avtorja:



## ***Zahvala***

*Za nasvete, popravke in ostalo pomoč pri pisanju diplomske naloge se zahvaljujem mentorju doc. dr. Roku Rupniku. Zahvaljujem se tudi Petri, ki je potrpežljivo popravljala slovnične napake, staršem, ki so mi omogočili študij in vsem, ki so kakorkoli pripomogli k nastanku diplomske naloge.*



# Kazalo

## Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Arhitektura sistema</b>	<b>3</b>
2.1	3-nivojska arhitektura . . . . .	3
2.2	Namen aplikacije . . . . .	5
2.3	Podatkovni model . . . . .	6
<b>3</b>	<b>Programska oprema</b>	<b>9</b>
3.1	Ogrodje .NET . . . . .	9
3.2	ASP.NET . . . . .	9
3.3	Jezik HTML . . . . .	10
3.4	CSS . . . . .	10
3.5	Javascript . . . . .	10
<b>4</b>	<b>Dodajanje in urejanje novic</b>	<b>11</b>
<b>5</b>	<b>Serviranje podatkov</b>	<b>13</b>
5.1	Prikaz vsebine na spletni strani (HTML) . . . . .	14
5.2	Prikaz vsebine na Philipsovi Smart televiziji . . . . .	15
5.3	XML in RSS . . . . .	15
5.4	JSON . . . . .	17

<b>6</b>	<b>Predvajanje radijske postaje v živo preko interneta</b>	<b>19</b>
6.1	Zadnje predvajane skladbe . . . . .	20
<b>7</b>	<b>Varnost</b>	<b>23</b>
7.1	Varnostna kopija in restavriranje podatkov . . . . .	23
7.2	Prijava v sistem za upravljanje vsebine . . . . .	24
7.3	Domenska prijava . . . . .	25
7.4	Navadna tekstovna prijava . . . . .	26
<b>8</b>	<b>Statistika obiska</b>	<b>27</b>
<b>9</b>	<b>Raba piškotkov</b>	<b>29</b>
<b>10</b>	<b>Povezava s socialnim omrežjem Facebook</b>	<b>31</b>
<b>11</b>	<b>Pogosti problemi, ki so prisotni pri programiranju spletnega portala</b>	<b>33</b>
11.1	Preprost iskalnik vsebin . . . . .	33
11.2	Prijazni internetni naslovi . . . . .	34
11.3	Nalaganje velikih datotek na strežnik . . . . .	34
11.4	Pravilnost serviranja datotek z nestandardnimi končnicami . .	35
11.5	Šumniki v spletnem naslovu . . . . .	35
11.6	Kako razbremeniti poizvedbe nad bazo . . . . .	36
11.7	Serviranje starih podatkov na spletni strani . . . . .	37
11.8	Prikaz alternativnega sporočila ob obisku neobstoječe strani .	37
11.9	Kaj je potrebno storiti, da nas Google najde? . . . . .	38
11.10	Različno serviranje podatkov v različnih brskalnikih in na različnih napravah . . . . .	39
<b>12</b>	<b>Kaj bi se dalo izboljšati</b>	<b>41</b>
12.1	Prikaz na vseh napravah . . . . .	41
12.2	Celostna grafična podoba (CGP) . . . . .	41
12.3	Vmesnik za zbiranje podatkov . . . . .	42
12.4	Večja pokritost modulov preko sistema CMS . . . . .	43

## *KAZALO*

12.5 Boljši prikaz sorodnih vsebin . . . . .	43
<b>13 Sklepne ugotovitve</b>	<b>45</b>



# Povzetek

Namen diplomskega dela je predstavitev spletnega portala, primerneza radijsko postajo. Spletni portal je mogoče obiskati preko različnih naprav: računalnika, mobilnega telefona, tabličnega računalnika ali pametne televizije. Omogoča ogled multimedijskih vsebin kot so slike, avdio in video izseki. Preko spletnega portala je mogoče poslušati radijsko postajo, podan pa je tudi seznam nazadnje predvajanih skladb in informacija o trenutnem voditelju v etru. Spletni portal nam daje osnovne podatke o radijski postaji, t.j. frekvence, seznam radijskih oddaj, seznam zaposlenih in kontaktne informacije. S pomočjo tega portala poslušalca radijske postaje prepričamo, da lahko do dobrih informacij ali glasbe, ki jih je slišal po radiu, pride tudi preko spleta; zato je potrebno skrbeti, da je portal zanimiv in v ospredje daje najbolj gledane vsebine. V ta namen portal spremlja interno statistiko, s pomočjo katere obiskovalcem v realnem času ponuja najbolj zanimive vsebine, urednikom portala pa pošilja statistiko obiska preko elektronske pošte večkrat dnevno. Pri vseh vsebinah ima uporabnik možnost povratne informacije, kar je ključno za stik s poslušalci. Spletni portal je opremljen z vtičniki socialnega omrežja Facebook. Na ta način radijska postaja ohranja konkurenco na trgu.

**Ključne besede:** spletna aplikacija, radijska postaja, multimedijske vsebine, statistika obiska, ogrodje .NET



# Abstract

The purpose of this thesis is a presentation of a web portal appropriate for a radio station. Listeners can access the portal from various devices: computers, mobile phones, tablets and smart televisions. The system for the management of radio contents is described (hereinafter referred to as CMS- Content Management System). Beside basic parts for web content memorization, there will be some solutions presented, which other systems do not use. The advantage of this web application is an internal statistics, which offers visitors the most interesting contents in real time and sends the reporter and other content writers the statistics multiple times a day to give them a quick insight into the level of interest certain content has reached. The radio station portal was developed in .NET framework with C# programming language. The content is being saved into Microsoft MSSQL base. The web application is hosted on an online server in Windows environment on Framework 4.0. platform.



# Poglavje 1

## Uvod

Kvalitetna radijska postaja mora svoje vsebine iz etra prenesti na splet, zraven pa dobro spremljati kako določeno vsebino sprejmejo poslušalci. Uredniki spletne strani morajo imeti enostaven CMS<sup>2</sup>, ki jim omogoča ažurno objavljanje vsebin na spletu. Sistem mora nato omogočiti, da znajo uporabniki stvari na lahek način povezati s svetovno znanimi iskalniki in socialnimi omrežji. Na tak način lahko radijska postaja postane popularna in konkurenčna. Spletnim uporabnikom je potrebno dopovedati, da ko slišijo nekaj zanimivega po radiu, lahko to stvar kasneje najdejo na spletu.

V diplomski nalogi bodo najprej naštetni osnovni gradniki, ki se uporabljajo za shranjevanje podatkov v bazo, potem pa bo prikazano kako te podatke servirati na različnih mestih v svetovnem spletu. Opisan bo sistem beleženja klikov, ki pomagajo uredniku spletne aplikacije, ko se odloča kaj bo uvrstil višje oz. na bolj vidno mesto. Poleg ročnega sortiranja novic bodo prikazane vsebine, ki se razporedijo dinamično glede na obiskanost. Sistem ima sprogramiran tudi iskalnik in personaliziran prikaz sorodnih vsebin. Našteti bodo primeri, do katerih lahko naletimo pri programiranju spletnega

---

<sup>2</sup>CMS ( angl. Content Management System, slovensko sistem za upravljanje vsebin) je sistem, ki omogoča urejanje in vzdrževanje vsebine spletnih strani brez znanja označevalnega jezika HTML. Urednik spletne strani tako lahko samostojno spreminja besedila, slike in druge elemente spletne strani brez pomoči podjetja ali osebe, ki je stran izdelalo.

portala.

Vsebine, ki jih hrani CMS, se lahko servirajo tudi na drugih napravah. Poleg serviranja podatkov v HTML jeziku, ki je primeren za prikaz na spletni strani in pametni televiziji, bo predstavljen primer, ko mobilna aplikacija pobira podatke v formatu JSON in jih prikazuje na sistemih Android in IOS.

# Poglavje 2

## Arhitektura sistema

Portal<sup>1</sup> radijske postaje je bil razvit v ogrodju .NET s programskim jezikom C#. Vsebine se shranjujejo v Microsoftovo bazo MSSQL. Spletna aplikacija gostuje na spletnem strežniku v Windows okolju na platformi Framework 4.0.

### 2.1 3-nivojska arhitektura

Spletni portal radijske postaje ima tri nivoje: odjemalec, spletni strežnik in podatkovna baza.

#### 2.1.1 Odjemalec

Govorimo o uporabniškem vmesniku za končnega uporabnika, ki obiše portal. To lahko stori s pomočjo internetnega brskalnika na uporabnikovem računalniku, z mobilno napravo ali preko pametne televizije. V vseh primerih uporabnik naredi poizvedbo po URL [6] naslovu in počaka, da se mu na dotični napravi pojavijo rezultati (npr. serviranje spletne strani v internetnem brskalniku).

---

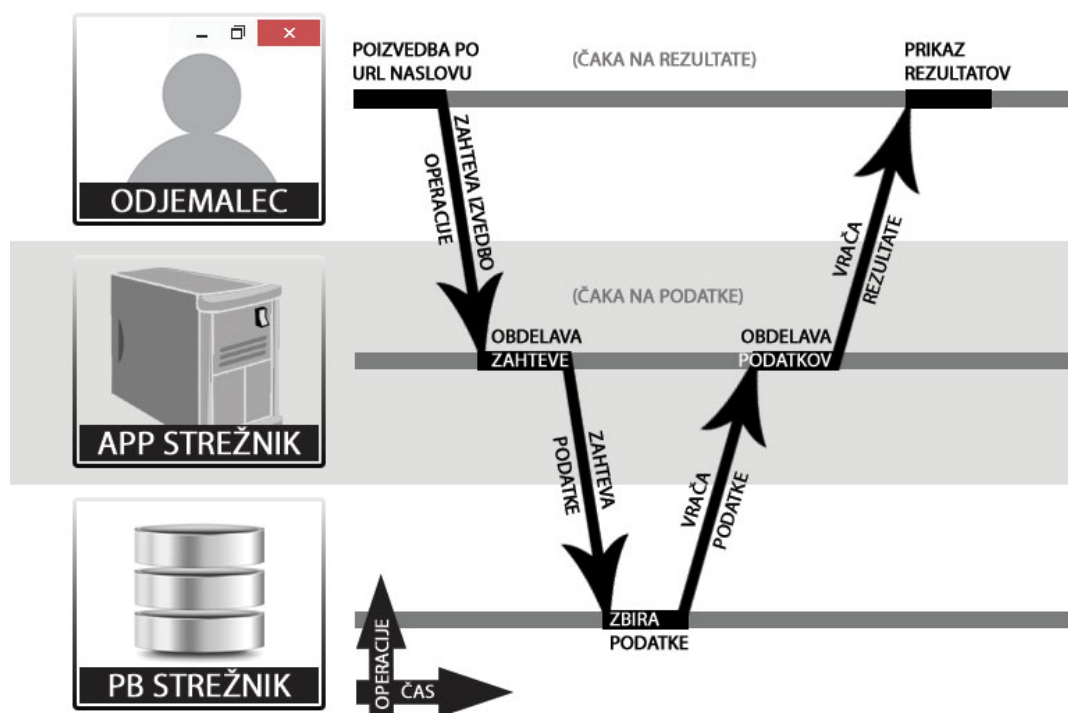
<sup>1</sup>O spletnem portalu govorimo, ko ima spletna aplikacija dodatne vtičnike: personalizirano prikazovanje vsebin, iskalnike, povezave do sorodnih vsebin, obveščanje preko elektronske pošte, možnost prenosa datotek idr. [1]

### 2.1.2 Spletni strežnik

Tu se nahaja glavni del spletne aplikacije, ki je zgrajena na .Net platformi v programskem jeziku C#. Strežnik obdela zahteve iz odjemalčevega nivoja in ga sprocesira. Povpraša za podatke v podatkovno bazo, jih sprocesira in vrne odjemalcu.

### 2.1.3 Podatkovna baza

Hrani vse trajne podatke v zvezi s portalom radijske postaje. S pomočjo jezika SQL [7] in uporabe relacijskih podatkovnih modelov vrača podatke spletnemu strežniku, ki jih servira nazaj odjemalcu. Na sliki 2.1 je podatkovna baza označena s kratico PB.

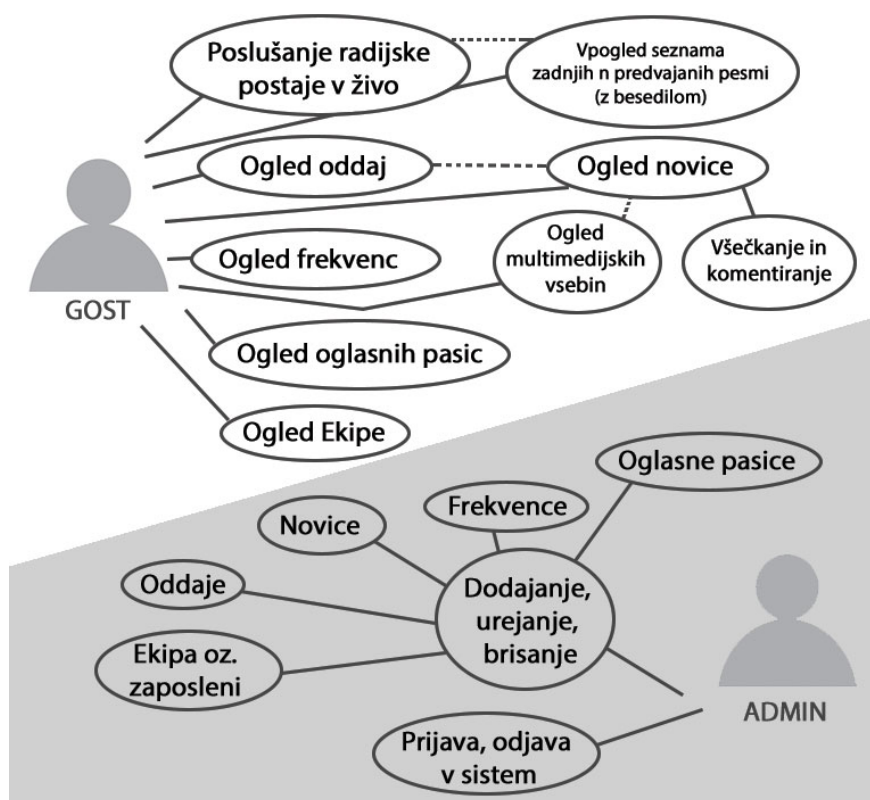


Slika 2.1: Arhitektura sistema.

## 2.2 Namen aplikacije

Namen aplikacije je končnemu uporabniku (odjemalcu) prikazati radijske vsebine na čim bolj enostaven in pregleden način z možnostjo povratne informacije. Potencialni poslušalec mora imeti možnost poslušanja radijske postaje preko spleta, imeti mora vpogled o zadnjih predvajanih skladbah in o tem, kdo je trenutno voditelj v etru. Omogočen je vpogled frekvenc in kontaktnih podatkov, da lahko uporabnik hitro javi, če je prišlo do kake okvare, ki bi ga lahko odvrnila od poslušanja radijske postaje. Vse naštete možnosti se morajo servirati na različnih napravah.

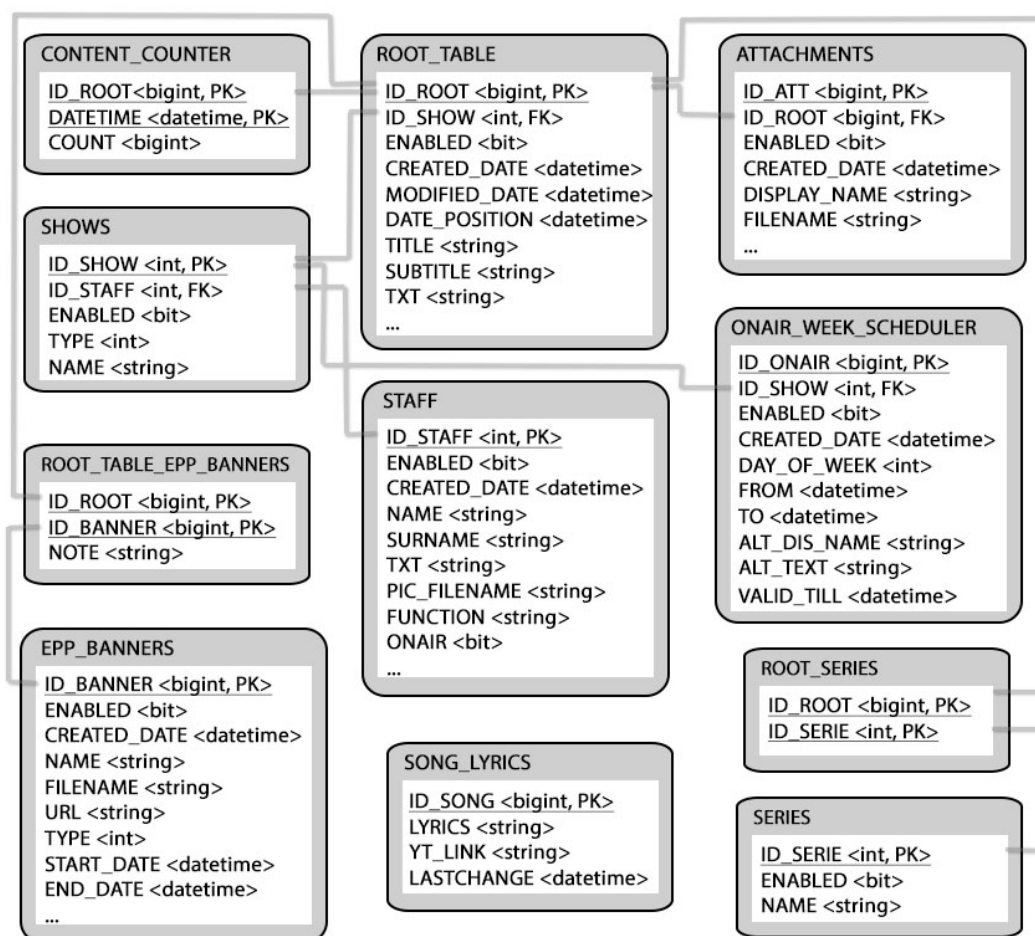
Po drugi strani pa je potrebno zaposlenim na radijski postaji omogočiti preprost sistem za upravljanje z vsebinami na dotičnem spletnem portalu (CMS).



Slika 2.2: Primeri uporabe spletnega portala radijske postaje.

## 2.3 Podatkovni model

V fazi analize je dobro kreirati podatkovni model. ER diagram (angl. Entity-Relationship Diagram [38]) na sliki 2.3 prikazuje osnovnih deset entitet diplomske naloge.



Slika 2.3: ER diagram.

Glede na podatkovni model je bilo kreiranih deset tabel. Glavna tabela za pomnenje radijskih vsebin oz. novic je `ROOT_TABLE`. Vsaka vsebina ima svoj univerzalen ključ (PK - primarni ključ), ki se avtomatsko povečuje. V tej tabeli so shranjeni podatki, kot so naslov, podnaslov in dodatna vsebina posamezne novice. Zabeleženo je kdaj se mora novica prikazati, v kateri od oddaj ipd. Dodajanje in urejanje novic je opisano v poglavju 4.

Tabela `SHOW` pove v katero od radijskih oddaj spada določena vsebina iz tabele `ROOT_TABLE`.

Vsaka novica lahko vsebuje tudi multimedijske vsebine, kot so video, avdio in slikovne datoteke, shranjene so v tabeli `ATTACHMENTS`. Vsaka priloga se v tabeli prikazuje kot tekstovno polje z imenom datoteke, ki obstaja na strežniku. Prav tako se vsebine iz `ROOT_TABLE` lahko povezujejo v serije (`SERIES`), to pa pomeni, da novice še dodatno grupiramo z različnimi filtri. V tabeli `STAFF` so shranjeni vsi zaposleni, ki delajo na radijski postaji. Vsak zaposleni ima svoj unikatni ključ in se lahko navezuje v katero od oddaj. V tabeli so shranjeni osnovni podatki o zaposlenih, kot so ime, priimek, funkcija in prikazna slika. Obstaja tudi polje (`ONAIR`), ki pove, če ta oseba dela tudi kot voditelj v etru ali ne. Vsi voditelji so nato povezani v tabeli `ONAIR_WEEK_SCHEDULER`, ki pove kateri dan v tednu in ob kateri uri dela kateri od radijskih voditeljev.

V tabeli `EPP_BANNERS` so shranjene vse pasice, ki jih končni uporabnik vidi kot reklame. To je bistveno za preživetje komercialnega radijskega portala. `SONG_LYRICS` pa služi za shranjevanje besedil posameznih pesmi, ki jih predvaja radijska postaja v etru.



# Poglavje 3

## Programska oprema

### 3.1 Ogrodje .NET

Vsaka spletna aplikacija potrebuje svoje ogrodje (angl. Framework), v katerem se izvaja. Eno najbolj razširjenih je Microsoftovo .NET ogrodje [2]. Deluje na operacijskih sistemih Windows. Razvijalci spletnih aplikacij lahko programirajo v večih programskih jezikih, najbolj razširjen je C# [9], v katerem je napisan tudi izdelek te diplomske naloge. Za začetek dela v okolju ASP.NET je potrebno poznati osnovne programiranja. To se je moč naučiti s pomočjo diplomske naloge ASP.NET [4] ali preko internetnega vodiča za prve korake v tem okolju [3]. V grobem zadeva deluje tako, da preko vnosnega polja (angl. text box) vsebino shranimo v bazo in jo na spletni strani predstavimo s pomočjo oznak (angl. labels).

### 3.2 ASP.NET

Gre za skupek orodij podjetja Microsoft, ki so primerni za spletnega programerja. Z ASP.NET [10] je možno narediti dinamične spletne aplikacije s pomočjo vizualnega vmesnika (angl. dynamic interface) in kode, ki jo programer doda projektu. CMS radijske postaje je bil narejen z Microsoftovim programom Visual Studio .NET 2012.

### 3.3 Jezik HTML

V okolju ASP.NET je moč programirati spletne aplikacije. Pri tem moramo poznati jezik HTML (angl. Hyper Text Markup Language) [11]. Gre za oznake z golim besedilom, ki ga znajo prikazati spletni brskalniki kot spletne strani. Servirati ga znajo tudi mobilne naprave in nekatere pametne televizije, potrebno ga je le malenkost modificirati za vsako napravo posebej.

### 3.4 CSS

Spletni strani lahko na enostaven način spreminjamo izgled s slogi. Rečemo lahko na primer, da bomo imeli več objektov na spletni strani z enakim robom in enakim ozadjem - najlažje to naredimo s pojmom CSS (Cascading Style Sheets) [12].

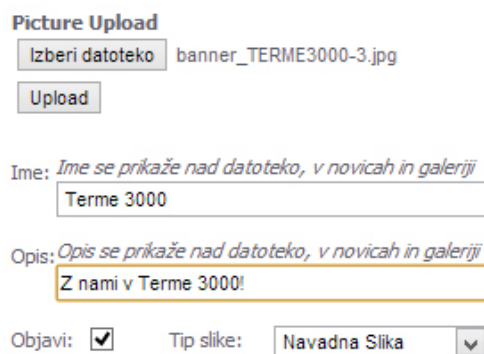
### 3.5 Javascript

Pri spletnih straneh zaradi boljšega delovanja ne moremo mimo jezika Javascript [14], ki se izvaja znotraj HTML strani, torej na strani končnega obiskovalca spletne strani. Olajša nam delo (lahko preračunavamo številke, menjamo sloge, beremo dokumente XML [15], ipd.), saj ni potrebno vsakič spraševati strežnika po vsebini, ampak se spreminja aktivno. Danes je zelo razširjena knjižnica JQuery [18], t.j. skupek skript, napisanih z jezikom Javascript.

## Poglavje 4

# Dodajanje in urejanje novic

Dodajanje in urejanje novic deluje tako, da se v bazo zabeležijo reči iz osnovnih gradnikov, največkrat iz tekstovnega vnosnega polja (angl. text box). Vsaka novica ima svojo šifro (ID), ki jo s pomočjo sistema CMS lahko urejamo. V prvi fazi gre samo za tekst, ki je v novici. Po uspešnem vnosu v bazo lahko dotični novici dodajamo priloge (Slika 4.1), npr. slike, .pdf datoteke, audio in video posnetke. Tudi omenjene multimedijske vsebine se v bazo shranijo kot goli tekst z imenom datoteke. V poglavju 5 je predstavljeno kako lahko te podatke serviramo končnim uporabnikom spleta in mobilnih aplikacij.



**Picture Upload**

Izberi datoteko banner\_TERME3000-3.jpg

Upload

Ime: *Ime se prikaže nad datoteko, v novicah in galeriji*  
Terme 3000

Opis: *Opis se prikaže nad datoteko, v novicah in galeriji*  
Z nami v Terme 3000!

Objavi:  Tip slike: Navadna Slika

Slika 4.1: Dodajanje multimedijskih vsebin (slika, avdio, video).

Pri dodajanju in urejanju novice je bilo razvitih več dodatnih filtrov, ki služijo za bolj personaliziran prikaz. Urednik CMS-ja si na primer lahko nastavi kdaj novica postane vidna. Doda ji lahko različne oznake, ki kasneje služijo pri ponujanju sorodnih vsebin. Več takih filtrov prikazuje spodnja slika 4.2.

ID Novice: 19393

Datum vnosa:

### UREJANJE

Naslov:

Podnaslov:

Glas:  *(Kadar prispevek vsebuje fotogalerijo)*

Ključ besede

\*Oznake se dodajo v meta tage (google search)

Oddaja:

Oznaka:

\* Sorodne teme naj imajo enako oznako.  
(Uredi oznake)

Serijska:

Objavi:

Prva Stran:

Banner:

Galerija:

Števec obiska:

AdvanceCMS:

Mobile:

YT link:

---

**B I U ABC** |

Zoki Kralj se vrača, tokrat z glasbeno uspešnico Gangnam Zoki!

Slika 4.2: Urejanje novice

# Poglavje 5

## Serviranje podatkov

Podatke, shranjene v bazi MS SQL <sup>2</sup>, lahko serviramo na več načinov. V tem poglavju bo predstavljeno, kako izgleda vsebina na spletni strani oz. pametni televiziji in kako v formatih RSS, XML in JSON, ki so primerni tudi za mobilne aplikacije.

---

<sup>2</sup>MS SQL (angl. Microsoft Structured Query Language) je podatkovna baza za spletne aplikacije, ki gostujejo na platformah operacijskega sistema Windows z ogrodjem .NET.

## 5.1 Prikaz vsebine na spletni strani (HTML)



Slika 5.1: Prikaz vsebine na spletni strani

Pri serviranju podatkov v HTML jeziku moramo veliko pozornosti dati različnim internetnim brskalnikom, ki utegnejo različno servirati podatke. Zato je dobro, da pred objavo reči dobro preverimo povsod. Slika 5.1 kaže primer kako izgleda novica v Googlovem brskalniku.

## 5.2 Prikaz vsebine na Philipsovi Smart televiziji

Za potrebe prikaza na Philipsovih Smart televizijah je bila narejena tudi posebna aplikacija, ki uporablja isto bazo. Tako aplikacijo je potrebno prilagoditi po navodilih ponudnika televizij, a v grobem gre še vedno za navaden HTML prikaz. Tudi ta aplikacija, ki jo prikazuje slika 5.2, je narejena v ASP.NET okolju.



Slika 5.2: Prikaz na Philipsovi Smart TV.

## 5.3 XML in RSS

Protokol RSS [16] je družina XML [15] datotek za spletno zlaganje (angl. web syndication), ki ga uporabljajo spletne strani z novicami. Na te novice se lahko naročamo s klikom na ikono (Slika 5.3).



Slika 5.3: Ikona RSS.

Koda XML izgleda tako:

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
  <channel>
    <title>Radio 1 - Mama Manka</title>
    <link>http://www.radio1.si/</link>
    <description>www.radio1.si RSS feed</description>
    <language>SI</language>
    <pubDate>Sun, 09 Feb 2014 11:14:07 GMT</pubDate>
    <ttl>5</ttl>
    <item>
      <title>POČITNICE ZA DUHOVNIKA</title>
      <link>http://www.radio1.si/strani/Fun.aspx?ID=25502</link>
      <description>
        
        Nista si ravno odpočila.
      </description>
      <pubDate>Fri, 07 Feb 2014 13:20:50 GMT</pubDate>
      <category>Mama Manka</category>
    </item>
    <item> ... </item>
  </channel>
</rss>
```

## POČITNICE ZA DUHOVNIKA

7. februar 2014, 14:20:50 →



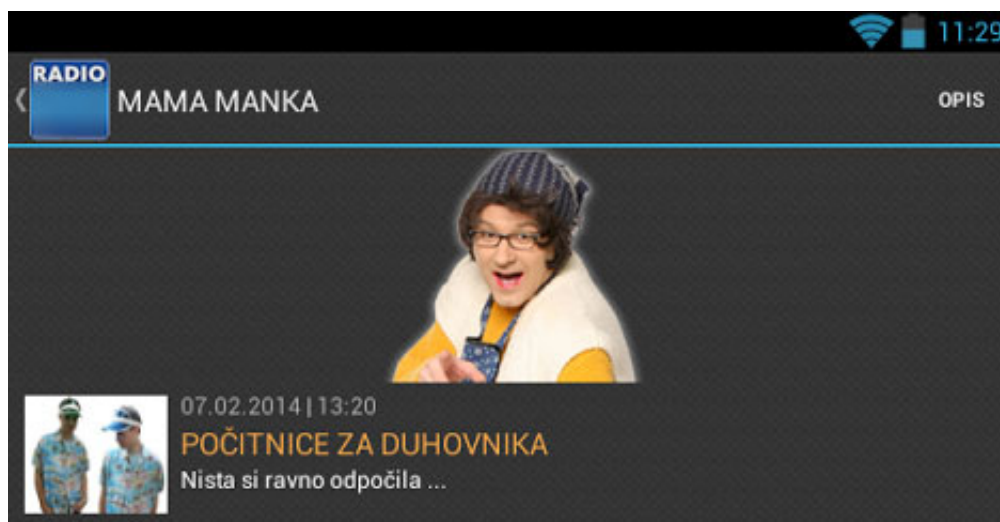
Nista si ravno odpočila

Slika 5.4: Prikaz XML kode kot RSS dnevnik v brskalniku Internet Explorer.

## 5.4 JSON

Podobno lahko enako stvar zapišemo tudi v formatu JSON [17], ki je primeren za prikaz na mobilnih aplikacijah.

```
{
  "CategoryID": "3",
  "CategoryName": "MAMA MANKA",
  "CategoryDescription": "Mama Manka ima rešitev za vsako težavo",
  "CategoryImage": "http://www.radio.si/img/oddaje-mamamanka.png",
  "Type": "Fun",
  "TypeSlo": "Zabava",
  "NewsList": [
    {
      "id": "23447",
      "create_date": "1378900804",
      "title": "POČITNICE ZA DUHOVNIKA",
      "category": "Fun",
      "subtitle": "Nista si ravno odpočila.",
      "Picture": "http://www.radio.si/img/naslovslike.jpg",
    }
  ]
}
```



Slika 5.5: Prikaz JSON kode v mobilni aplikaciji.

Pravilnost kode JSON je možno preizkusiti s pomočjo spletne strani JSON Validator [37].

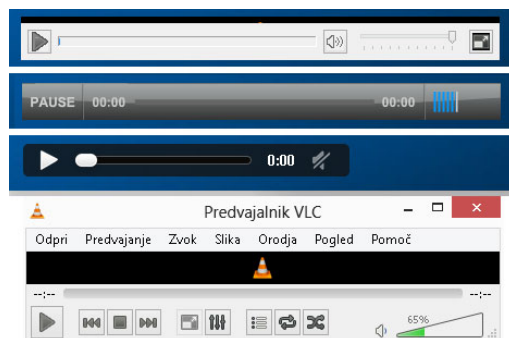


## Poglavje 6

# Predvajanje radijske postaje v živo preko interneta

Predvajanje radijske postaje v živo je rešeno na več načinov, ki jih končni obiskovalec lahko nastavi sam, izbor pa se s pomočjo piškotkov shrani za naslednji obisk. Obstajajo štiri možnosti:

- predvajanje z vgrajenim predvajalnikom,
- predvajanje z zunanjim predvajalnikom,
- predvajanje s tehnologijo Flash/SWF in
- predvajanje s tehnologijo HTML5.



Slika 6.1: Različni predvajalniki: vgrajeni, Flash, HTML5 in zunanji predvajalnik

Več izbire končnemu uporabniku omogoča, da se prej prilagodi in mu ni potrebno nameščati dodatnih programov. Za predvajanje z vgrajenim ali zunanjim predvajalnikom mora biti na sistemu uporabnika nameščen program za predvajanje glasbe (Windows Media Player, VLC, ipd.) in v internetnem brskalniku je potrebno to možnost potrditi. Slabost tega načina je morebitna dodatna namestitvev predvajalnika za nemoteno delovanje. Včasih gre samo za en klik več, a to lahko poslušalca zmoti. Podobno velja za tehnologijo Flash/SWF, kjer je potrebno imeti nameščen Flash predvajalnik [30]. Ostane še predvajanje z novejšo tehnologijo HTML5, vendar pa le-to znajo poganjati samo novejši internetni brskalniki.

Z Google Analitiko je bilo ugotovljeno, da več kot 60% ljudi še vedno uporablja starejše internetne brskalnike, ki pa imajo nameščen Flash predvajalnik, zato je dobro imeti podprte vse možnosti.

V diplomskem delu je predvajanje rešeno tako, da na osnovni strani kliknemo na predvajanje v živo, pri čemer se odpre predvajalnik v novem oknu in tako ga ima poslušalec odprtega neodvisno od ostalih zavihkov brskalnika.

## 6.1 Zadnje predvajane skladbe

V samem predvajalniku je zraven dopisanih še zadnjih deset skladb, ki so bile predvajane. Program za delo v studiu ob vsakem pričetku skladbe generira novo XML datoteko in jo odloži na spletni strežnik. Paziti je potrebno, da taka datoteka nima nastavljenega predpomnenja in se osvežuje v realnem času. Več o tem si lahko preberete v poglavju 11.

Koda XML za zadnje predvajane skladbe je taka:

```
<lastsonggenerator>
  <title>Nazadnje predvajano</title>
  <language>SI</language>
  <pubDate>Sun, 09 Feb 2014 10:44:32 GMT</pubDate>
  <nextsong>
    <filename>0.MPG</filename>
    <artist>JOE COCKER</artist>
    <title>UP WHERE WE BELONG</title>
    <playtimestart>11:51</playtimestart>
  </nextsong>
  <currentsong> ... </currentsong>
</lastsongs>
<n>10</n>
<song> ... </song>
...
</lastsongs>
</lastsonggenerator>
```



Slika 6.2: Celoten predvajalnik za predvajanje radijske postaje v živo preko spleta.



# Poglavje 7

## Varnost

Veliko poudarka je potrebno nameniti varnostnim ukrepom za primer, ko nam del strojne ali programske opreme odpove. Podatke je dobro imeti in servirati iz različnih mest. Pri varnosti je veliko govora o varnostnih kopijah, večina ljudi pa ne izvaja restavriranja podatkov iz takih kopij dovolj pogosto in tako lahko pride do dolgotrajnega problema, ki lahko veliko stane.

### 7.0.1 Replikacija in zrcaljenje podatkov

Za hitro ukrepanje ob izpadu enega od strežnikov ima portal urejeno zrcaljenje baze (angl. mirroring) [34] in datoteke replicirane na treh ločenih mestih s pomočjo DFS [33] replikacije.

## 7.1 Varnostna kopija in restavriranje podatkov

Z orodjem MS SQL Management je nastavljeno, da se vsak dan (ponoči, ko je najmanj obiska) naredi varnostna kopija podatkov. Po uspešnem kreiranju datotek .bak je te dobro kopirati na različna mesta. Portal radijske postaje vse nove datoteke s pomočjo Robocopy [35] skripte za robustno kopiranje arhivira podatke na lokacijsko ločena računalnika, prvi je v Mariboru, drugi

v Novem mestu. Za namen restavriranja podatkov je postavljen slabši virtualni računalnik, kjer se vsak mesec restavrira podatke iz varnostne kopije in lokalno zažene spletni portal. Če bi skrbeli samo za varnostne kopije, ne pa za restavriranje podatkov, bi se lahko zgodilo, da bi se kreirala varnostna kopija, ki pa ne bi bila uporabna in v takih primerih je potrebno ogromno časa, da se tak set arhiva reši, navadno tudi veliko stane.

## 7.2 Prijava v sistem za upravljanje vsebine

Vsaka prijava v CMS mora biti zaščitena, da lahko vsebine dodaja in spreminja samo vnaprej določena množica ljudi. V nasprotnem primeru bi kaj hitro lahko bila spletna stran zlorabljena z neželeno vsebino. Predstavljeni bodo trije primeri zaščite sistema za upravljanje vsebin, ki ga uporablja spletna aplikacija.

### 7.2.1 Microsoftov Office Web App (OWA)

Za prijavo lahko uporabljamo ločeno aplikacijo, ki deluje enako kot spletni vmesnik za pregled e-pošte. Taka aplikacija najprej pogleda, če določeno uporabniško ime in geslo pripadata skupini ljudi s pravicami za urejanje spletne strani, nato pa vrne dolg ključ, ki ga CMS zna prepoznati in tako dopusti njegovo uporabo. Tak sistem je dober, ker je že narejen tako, da ne dopusti velikega števila prijav v kratkem času in je zato varen, programer pa z njim nima dodatnega dela.

Microsoft®  
**Outlook® Web App**

Security ( [show explanation](#) )

This is a public or shared computer  
 This is a private computer

Use Outlook Web App Light

I want to change my password after logging on

Domain\user name:

Password:

[Log On](#)

Connected to Microsoft Exchange  
Secured by Microsoft Forefront Threat Management Gateway  
© 2009 Microsoft Corporation. All rights reserved.

Slika 7.1: Prijava s sistemom OWA.

## 7.3 Domenska prijava

Podjetje, ki ima svoje zaposlene v Microsoftovi domeni [36], se lahko tudi pri prijavi v CMS posluži domenske prijave. V takem primeru postavimo na stran dva vnosna polja z besedilom, prvi služi za uporabniško ime, drugi za geslo, sistem pa potem preveri, če je prijava uspešna in če ima ta uporabnik dostop za uporabo sistema CMS. To storimo s parimi vrsticami kode (Slika 7.2).

```
DirectoryEntry entry = new DirectoryEntry("LDAP://" + "STUDIO", txtUser.Text.Trim(), txtPass.Text);
DirectorySearcher mySearcher = new DirectorySearcher(entry);
mySearcher.Filter = "&(objectClass=user)(|(cn=" + txtUser.Text.Trim() + ")(sAMAccountName=" + txtUser.Text.Trim() + ")))";
SearchResult result = mySearcher.FindOne();
```

Slika 7.2: Primer iskanja uporabnika v domeni.

## 7.4 Navadna tekstovna prijava

Za CMS je dobro, da omogoča tudi prijavo v sistem za primer, ko zgornje ločene aplikacije spodletijo (vzdrževanje, tehnična napaka, ...). V tem primeru si moramo izbrati zelo zapleteno uporabniško ime in geslo. Taka vrsta prijave naj se uporablja zgolj za rezervo, da se ob nepričakovanih dogodkih še vedno lahko dostopa do CMS-ja.

Omeniti je potrebno tudi dodatne možnosti, ko je CMS dostopen samo iz službenih računalnikov. V tem primeru ima podjetje interne DNS [25] zapise, ki jih zunanji svet ne pozna. A v tem primeru uporabnik ne more urejati spletne strani od doma.

## Poglavje 8

# Statistika obiska

Konkurenčen medij mora imeti tudi dober vpogled v statistiko obiska vsebin spletne aplikacije. Le na tak način se lahko v medijski hiši prepričajo katere vsebine so zanimive in katere ne. Na spletu obstaja Googlevo orodje Analitika [24], ki ga uporablja večina spletnih medijev. Omogoča vpogled v statistiko obiska s številnimi filtri. Vendar pa ima tudi nekaj pomanjkljivosti, zato ima diplomsko delo tudi svoj vpogled v statistiko. Beleži klike nad določeno vsebino na urni postavki in pri tem ne uporablja piškotkov (angl. cookies). Na ta način je moč analizirati obisk strani v realnem času. Trenutno je sprogramirano tako, da se statistika dnevno najbolj gledanih vsebin pošlje administratorjem strani sedemkrat dnevno. V vnaprej določenih terminih dobijo poročilo o ogledih vsebin na spletni strani ter na mobilnih napravah z operacijskim sistemom iOS in Android. Tudi končnim obiskovalcem se na lahek način ponudi najbolj gledane in zanimive vsebine (Slika 8.1).

Na dnevni bazi se izračunajo tudi obiski vseh dni v letu in primerjava s predhodnimi leti. Administratorji takega spletnega portala takoj dobijo občutek kdaj je največ uporabnikov na spletu in temu primerno objavljajo zanimive vsebine v tistem terminu. Previdnost je potrebna tudi na ostalih socialnih omrežjih (npr. na Facebooku), kjer še posebej velja, da je treba objavo skrbno pripraviti in jo dati v vpogled ob primernem času.

Omeniti velja še, da je podatkov v bazi za statistiko ogromno, zato je po-

trebno čim manj obremenjevati strežnike. Spletni portal radijske postaje zato vsako uro podatke premakne v ločeno bazo, kjer se potem izvajajo zahtevnejše analize. Na sami spletni strani pa lahko obremenjenost rešimo s shranjevanjem v predpomnilnik (angl. cache) in tako se statistika ne preračunava za vsakega obiskovalca posebej.



Slika 8.1: Primer prikaza najbolj gledanih vsebin na spletu in na mobilni napravi v vnaprej določenem obdobju.

## Poglavje 9

# Raba piškotkov

Piškotek [19] je majhna datoteka, ki se shrani na računalniku ob obisku spletne strani. Piškotki vsebujejo različne informacije, ki jih spletna stran prebere, ko jo ponovno obiščete. Uporablja jih večina sodobnih spletnih strani, tudi spletni portal radijske postaje. Sami po sebi niso škodljivi, vendar jih je potrebno omeniti, saj je na področju EU s 15. 6. 2013 začel veljati spremenjen Zakon o elektronskih komunikacijah. Več o relevantni slovenski zakonodaji glede politike zasebnosti in uporabe piškotkov si lahko preberete v dokumentu, ki je naveden v viru: Smernice Informacijskega pooblaščenca o rabi piškotkov [20].

Spletna stran za boljšo funkcionalnost uporablja spletne piškotke,  
ki ne hranijo osebnih podatkov - [Več o tem.](#)

V redu

Slika 9.1: Opozorilo o uporabi piškotkov.

Pri spletnem portalu radijske postaje je bilo potrebno prikazati veliko sporočilo o rabi piškotkov (Slika 9.1), saj uporablja piškotke tujih subjektov za sledenje števila obiskov (Googlova in Mossova analitika).



## Poglavje 10

# Povezava s socialnim omrežjem Facebook

Slovenski spletni mediji po mojem občutku premalo vlagajo v deljenje povezav na socialnem omrežju Facebook, ki je v zadnjih letih postal svetovno zelo konkurenčen, zato bom v tem poglavju namignil kako urediti "meta oznake" na strani, da bo ogled vsebin znotraj Facebooka podoben tistim, ki jih uporablja npr. Youtube. Za spletne programerje obstaja stran Facebook Developers [22], kjer je to natančno opisano. CMS radijske postaje ima vnaprej določeno kako se vsebina prikaže na Facebooku, če vsebuje fotografijo, avdio ali video posnetek.



Slika 10.1: Primer deljenja povezave na socialnem omrežju Facebook.

### 3 POGLAVJE 10. POVEZAVA S SOCIALNIM OMREŽJEM FACEBOOK

Na spletnem portalu radijske so prisotni tudi drugi vtičniki socialnih omrežij, s katerimi dosežemo, da se interakcije obiskovalcev vidijo tako na spletni strani kot na dotičnem portalu. Tudi na tak način se poveča obisk spletnega portala. Slika 10.2 prikazuje možnost "všečkanja" in komentiranja.



Slika 10.2: Interakcije tujih socialnih omrežij znotraj portala.

# Poglavje 11

## Pogosti problemi, ki so prisotni pri programiranju spletnega portala

V tem poglavju bodo predstavljene rešitve pogostih problemov, ki so nastali med ustvarjanjem spletnega portala radijske postaje. Pojavile so se težave pri nalaganju večjih datotek na strežnik, pravilnost serviranja datotek z nestandardnimi končnicami, kako razbrati šumnike iz naslova spletnega mesta, pojavilo se je vprašanje kako razbremeniti poizvedbe nad bazo, kako končnemu obiskovalcu zagotoviti lahko uporabo spletne strani iz različnih naprav, kako prepričati svetovno znane iskalnike o dotični spletni strani, kaj prikazati ob obisku potekle ali neobstoječe strani, ipd.

### 11.1 Preprost iskalnik vsebin

Vsak popularen portal mora imeti iskalnik vsebin. Na spletu obstaja ogromno različnih predlog za to, a obstaja veliko slabosti (požrešnost, preveč parametrov). Priporočljivo je, da je čimbolj preprost. V tej diplomski nalogi je iskalnik narejen tako, da v vsaki novici pogleda naslov, podnaslov in vsebino novice. Iskanje se izvede v dveh fazah. V prvi fazi znake iz vnosnega

polja, kot so presledek, vejica ali podpičje, zamenjamo z znakom '%'. S tem dosežemo, da ne iščemo točnega zaporedja besed, ampak preverimo samo njihovo prisotnost. Če nam taka poizvedba ne vrne rezultatov, nekatere besede v iskalnem nizu zanemarimo. Spravimo jih v polje nizov in iščemo vsako besedo posebej. Če tudi tu ni rezultatov pomeni, da iskani niz ne obstaja v bazi podatkov. Primer podobne SQL poizvedbe je:

```
SELECT * FROM Root_Table WHERE text LIKE '%niz1%niz%niz3%'.
```

## 11.2 Prijazni internetni naslovi

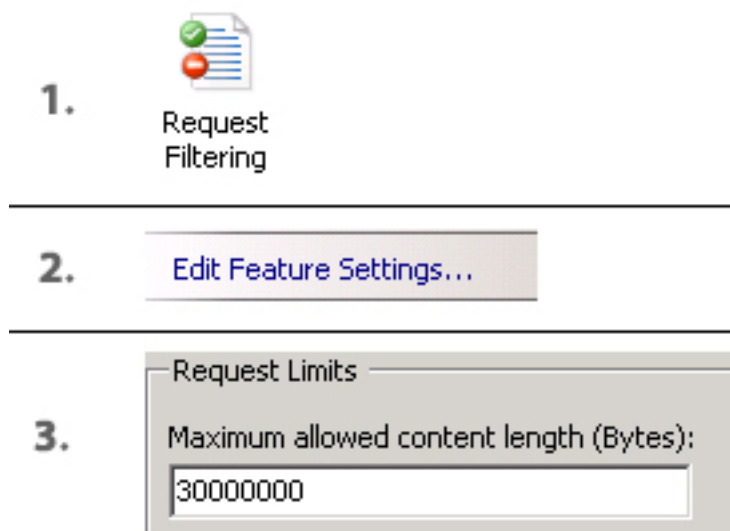
V okolju ASP.NET obstaja možnost rabe t.i. prijaznih internetnih naslovov (angl. Friendly URL names). Privzete datoteke imajo končnico .aspx, torej je končni naslov internetne strani lahko npr. <http://www.radio.si/kontakt.aspx>, le-to pa je smotrno spremeniti v <http://www.radio.si/Kontakt>. V navedenem viru [29] je lepo prikazano, kako se to najlažje naredi. V diplomskem delu je veliko naslovov povezanih še z golimi datotekami, a po korakih se dodajajo prijazni naslovi.

## 11.3 Nalaganje velikih datotek na strežnik

Spletna aplikacija mora biti prilagodljiva in ponuditi prikaz večjih datotek (npr. daljše video vsebine), a tu pride do omejitve pri splošnih nastavitvah. Najprej je potrebno zagotoviti, da imamo dovolj pravic za nalaganje večjih podatkov v datoteki Web.config, to storimo z dodatno vrstico:

```
<httpRuntime maxRequestLength="102407" requestValidationMode="2.0" />
```

Treba pa se je zavedati, da to v številnih primerih ni dovolj. Podobno je potrebno zagotoviti tudi na strani strežnika. V IIS-ju se pomaknemo v napredne nastavitve in spremenimo privzeto vrednost, kot prikazuje spodnja slika.



Slika 11.1: Možnost nalaganja večjih datotek na strežnik v IIS.

## 11.4 Pravilnost serviranja datotek z nestandardnimi končnicami

Po privzetem strežnik, kjer gostuje naša spletna stran, ne bo serviral vseh datotek. Za primer, ko jih nujno potrebujete, je to prav tako potrebno dodatno nastaviti v IIS. Potrebno je dodati t.i. tip MIME [31] (angl. Multipurpose Internet Mail Extensions) in vpisati potrebno končnico (npr. .doc ali .odt), v nasprotnem primeru bo strežnik javil napako, čeprav datoteka obstaja.

## 11.5 Šumniki v spletnem naslovu

V nekaterih primerih podatke prenašamo kar v URL naslovu preko različnih nizov (angl. Query String). V CMS-ju so se pojavile težave pri iskalniku, ki iskalni niz dobi iz internetnega naslova in vsebuje šumnike, npr. ?search-string=čokolada. V takem primeru je potrebno v požarnem zidu (na mestu, kjer gostuje spletna aplikacija) nastaviti, da dovoli take nize. Če to ne po-

maga, je potrebno take znake (angl. special characters) ustrezno enkodirati in ob prikazu dekodirati.

## 11.6 Kako razbremeniti poizvedbe nad bazo

V okolju ASP.NET obstaja lepa rešitev, kjer z dvema parametroma (angl. EnableCaching in CacheDuration) nad gradnikom pošljemo strežniku kako dolgo naj pomni podatke. Na ta način se izognemu nepotrebnim poizvedbam nad bazo in preračunavanju podatkov, ki se ne spreminjajo pogosto.

```
<asp:ObjectDataSource ID="MeniOddajeDataSource" runat="server"
    SelectMethod="Top9GetDataBy"
    CacheDuration="3600"
    EnableCaching="True">
```

### 11.6.1 Tehnologija LINQ

Pomembno je poznati tudi tehnologijo LINQ [21], ki nam omogoča, da naredimo čim manj poizvedb direktno nad bazo, potem pa podatke obdelujemo znotraj programa. Časovno se namreč veliko bolj splača delati s pomnilnikom [5] (angl. RAM), kot pa za vsako stvar gledati v bazo. V današnjih časih so računalniki zelo zmogljivi in si to lahko privoščimo, vseeno pa je potrebno biti dovolj pozoren na prepožrešne funkcije.

### 11.6.2 Požrešne metode

Pri programiranju spletnih aplikacij je potrebno še bolj paziti, da funkcije znotraj strani niso preveč požrešne. Spletna stran se namreč počasneje odpira kot navadna aplikacija (npr. Windows aplikacija [26]), ki je lokalno naložena na računalniku. Tip Slovar (angl. Dictionary) [27] velja za enega najhitrejših in najbolj uporabnih pri številnih okoljih, tudi v ASP.NET. Gre za implementacijo Hash tabele, kjer si npr. lahko shranimo imena naslovov za določeno novico, shranjeno z unikatno številko. Ni potrebno večkrat

povprašati baze po naslovu, ampak si take stvari že vnaprej shranimo v pomnilnik.

Moje izkušnje pravijo, da lahko spletno stran veliko hitreje serviramo končnemu obiskovalcu, če naredimo samo eno poizvedbo nad bazo za določeno poglavje, imena novic shranimo v slovarski tip in tako vse skupaj serviramo iz predpomnilnika. Poleg tega morajo imeti poizvedbe smiselne filtre, da iz baze zares dobimo samo tiste podatke, ki jih potrebujemo.

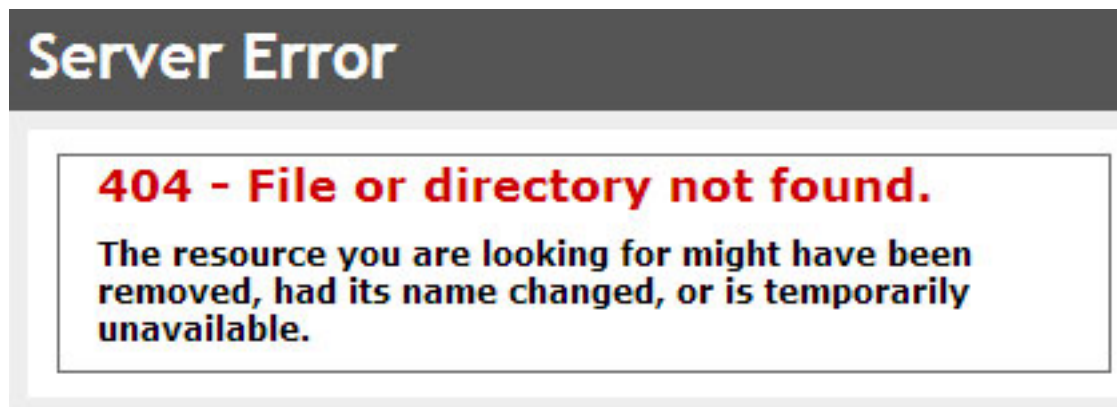
## 11.7 Serviranje starih podatkov na spletni strani

Lahko pa se zgodi ravno obratno, torej, da so datoteke na strežniku osvežene, internetni brskalnik pa še vedno vrača stare podatke. V tem primeru moramo v IIS spremeniti glavo HTTP odgovorov [32] (angl. HTTP Response Headers) za določeno datoteko. Lahko nastavimo, da ni pomnjenja ali pa nastavimo čas pomnjenja. Navadno naletimo na problem pri serviranju XML datotek, ki imajo navadno privzeto pomnjenje nastavljeno na petnajst minut. Spletni portal radijske postaje pogleda tako datoteko, da ugotovi katere pesmi so se nazadnje predvajale v etru in tak podatek mora biti ažuren, zato je v tem primeru bilo potrebno spreminjati nastavitve take datoteke.

## 11.8 Prikaz alternativnega sporočila ob obisku neobstoječe strani

Na spletu velikokrat naletimo na problem, da nekatere povezave (linki) ne delujejo več in posledično dobimo grdo sporočilo, ki ga prikaže brskalnik ali pa strežnik. V okolju .NET se temu elegantno izognemo tako, da ob vsaki napačni poizvedbi vrnemo alternativno stran. To deluje, kadar imamo v datoteki web.config prisoten podoben stavek:

```
<customErrors mode="On" defaultRedirect="Error.html">.
```



Slika 11.2: Privzeti primer serviranja napake v ASP.NET.

**Zahtevana stran trenutno  
ni na voljo.**  
Preusmerjeni boste na začetno stran.

Slika 11.3: Modificirani primer serviranja napake v ASP.NET.

## 11.9 Kaj je potrebno storiti, da nas Google najde?

Velik problem spletnih strani je, da jih svetovno znani brskalniki ne najdejo, ko je to najbolj potrebno, zato je potrebno skrbno obdelati vsako podrobnost, ki jo najdemo v Googlovem orodju za spletne skrbnike [23]. Predvsem je potrebno biti pozoren na pravilne META [13] oznake, indeksiranje spletne strani z ustreznim zemljevidom (datoteka sitemap.xml) in na obogatene izrezke znotraj standardnih HTML elementov. Robotom, ki skrbijo kako naj bo vaša stran predstavljena njihovim uporabnikom, lahko s kratkimi meta zapisi dopoveste, da naj vaše vsebine indeksirajo vsak dan:

```
<meta name="robots" content="index,follow" />  
<meta name="revisit-after" content="1 days" />.
```

Potrebno pa je omeniti tudi primer, ko imamo podstrani, ki jih nočemo med rezultati v iskalnikih. Gre za primere, ko postavimo spletno stran, ki je primerna samo za določeno množico ljudi. Za ta namen dodamo dotični podstrani meta oznako:

```
<meta name="robots" content="noindex" />
```

## 11.10 Različno serviranje podatkov v različnih brskalnikih in na različnih napravah







Dandanes ni dovolj, da spletno stran testiramo v enem okolju, preizkusiti jo je potrebno na čim večih napravah z različnimi internetnimi brskalniki. Najbolj je treba biti pozoren, da: - se je treba na nekaterih mestih izogniti "flash" animacijam s končnico .swf, ker jih nekatere (predvsem mobilne) naprave ne podpirajo; - lahko ima obiskovalec spletne strani različno širino ekrana, potrebno je narediti raziskavo [24] in se prilagoditi večini oz. vsem, če je le možno; - je navadno lažje pretočiti veliko število podatkov na osebem računalniku kot pa na mobilniku, zato se večina odloči, da za mobilnike ponudi okrnjeno spletno stran ali mobilno aplikacijo.

V C# je moč z eno vrstico izvedeti, če je obiskovalec prišel na vašo spletno stran z mobilnikom ali preko osebnega računalnika. Vprašamo po




```
Request.Browser.IsMobileDevice
```

in dobimo odgovor da (true) ali ne (false). Na ta način se je potem lažje odločiti kakšen naj bo prikaz. Navada je, da se obiskovalca to vpraša prvič in ga po potrebi samodejno preusmeri na mobilno stran ali pa mu ponudi prenos aplikacije za mobilnik, če obstaja. CMS radijske postaje ima že narejeni mobilni aplikaciji za prikaz na napravah, ki uporabljajo IOS ali Android, torej trenutno najbolj znana operacijska sistema za mobilnike. Z zgornjim stavkom se z lahkoto odrečemo tudi Flash animacijam, ko predpostavimo, da jih mobilnik ni sposoben prikazati.







Za raziskave se uporablja Googlovo orodje Analitika [24], kjer se da razbrati s kakšnimi napravami in s katerimi brskalniki prihajajo obiskovalci.

Browser	Visits	% Visits
1. Chrome	173,258	 36.47%
2. Internet Explorer	119,745	 25.20%
3. Firefox	110,495	 23.26%
4. Android Browser	38,571	 8.12%
5. Safari (in-app)	14,099	 2.97%
6. Safari	13,335	 2.81%

Operating System	Visits	% Visits
1. Android	72,253	 73.31%
2. iOS	23,999	 24.35%
3. Windows Phone	1,395	 1.42%

Operating System	Visits	% Visits
1. Windows	366,480	 77.13%
2. Android	72,516	 15.26%
3. iOS	23,985	 5.05%
4. Macintosh	6,364	 1.34%
5. Linux	3,303	 0.70%
6. Windows Phone	1,462	 0.31%

Slika 11.4: Primer statistike s pomočjo Googlovega orodja Analitika.

# Poglavje 12

## Kaj bi se dalo izboljšati

Pri programiranju praktično nikoli ne moremo trditi, da je izdelek končan, ampak se je treba ozreti v prihodnost in razmisliti kaj bi se še dalo dodati oz. izboljšati. V prvi fazi se o tem začne pogovarjati, ko pri kakšnem drugem portalu vidimo prednosti, ki jih naš izdelek nima. Tudi pri tem diplomskem delu ostajajo odprta vprašanja, ki se bodo sproti reševala.

### 12.1 Prikaz na vseh napravah

Portal ima možnost serviranja podatkov na trenutno najbolj prodajanih napravah na trgu. Vseeno pa bo v prihodnosti potrebno razmisliti tudi o ostalih napravah, za katere sistem še ni razvit (npr. za Windows Phone in BlackBerry). Zaenkrat je priporočljivo uporabljati CMS v internetnem brskalniku, zato je delo z njim na napravah z manjšim zaslonom oteženo. Bilo bi smiselno razmisliti tudi o okrnjeni verziji sistema za upravljanje z vsebinami.

### 12.2 Celostna grafična podoba (CGP)

Velik pomen pri učinkoviti spletni strani ima tudi grafika. Če danes pogledamo pet let staro stran, hitro opazimo, da ni privlačna. Spreminjajo se

knjižnice (jQuery), ki iz leta v leto omogočajo boljšo učinkovitost pri prehodih med grafičnimi gradniki. Po drugi strani pa se je potrebno zavedati, da vse več uporabnikov hoče do ažurnih podatkov preko malega zaslona (npr. na telefonu). Tako je najbolje, da se razvijeta dve grafični podobi: ena za prikaz na velikih zaslonih in druga na malih. V ta namen je bila tudi razvita mobilna aplikacija, v prihodnosti bi bilo pametno storiti podobno še na sami spletni strani (t.i. mobilna spletna stran).

### 12.3 Vmesnik za zbiranje podatkov

Diplomsko delo še nima uporabniškega vmesnika za nagradne igre, kjer naročnik navadno zahteva, da se zbirajo podatki o uporabniku (ime, priimek, kraj, spol, ...). To se zaenkrat rešuje tako, da se ročno izdelava obrazec za prijavo, ki se ga vgnezdi v potrebno novico. Dobro bi bilo, da bi imel urednik že v CMS-ju možnost, da sam izdelava tak obrazec.

Ime in priimek:	<input type="text"/>
Naslov:	<input type="text"/>
Telefon:	<input type="text"/>
E-pošta:	<input type="text"/>
<input checked="" type="checkbox"/> Strinjam se s pravili nagradne igre	
<input type="button" value="SODELUJ!"/>	

Slika 12.1: Prikaz vmesnika za zbiranje podatkov.

## 12.4 Večja pokritost modulov preko sistema CMS

Večji del spletnega portala lahko uredniki urejajo sami s pomočjo CMS-ja, a to ne velja za cel spletni portal. Včasih je potreba po novem modulu - primer je lahko pasica, ki se mora prikazovati na nestandardnih mestih - CMS pa objave na takem mestu ne dovoli. V takem primeru moram kot programer spletnega portala sam dodajati reči. Trudim se, da vsak modul posebej programiram tako, da je čimprej na voljo uredniku spletnega portala.

## 12.5 Boljši prikaz sorodnih vsebin

V CMS-ju je možno pri vsaki novici dopisati ključne besede, s pomočjo katerih se lahko prikažejo sorodne vsebine. Zaenkrat sistem pregleda skupne besede in jih vrne po naključnem zaporedju. Tak pristop je za celoten sistem požrešen, zato ga bo potrebno izboljšati. Potrebno bo bolje indeksirati vse ključne besede in vsako sorodno vsebino dodatno otežiti glede na obisk.

### Sorodne vsebine:

#### Lord Gaga



Mišič dobil prvo plačo in prvi videospot

#### Pravljica za Gašparja Gašparja



Pa lahko noč!

Slika 12.2: Prikaz sorodnih vsebin.



## Poglavje 13

### Sklepne ugotovitve

Cilj diplomskega dela je bil prikaz portala, ki ga programiram zadnja štiri leta. Uporablja ga več slovenskih in tujih radijskih postaj. Največ poudarka sem dal na pogoste probleme, ki nastanejo pri programiranju obsežnega portala. Večina dela govori o stvareh, ki jih konkurenčni portali nimajo. Z narejenim sem zadovoljen, saj sem razvil kar nekaj modulov v programskem jeziku C#, ki jih klasični odprtokodni sistemi nimajo. V prvem letu je bil portal narejen izključno za serviranje podatkov preko spletne strani. Potem so stvari postale še zanimivejše in zelo ponosen sem bil, ko sem izrisal in sprogramiral aplikacijo za Philipsovo Smart televizijo. Sledilo je sodelovanje s podjetjem, ki je izdelalo aplikacijo za mobilne operacijske sisteme IOS in Android, sam pa sem pripravil komunikacijo med aplikacijo in obstoječo bazo. Sledili so še številni programi, ki so v realnem času pokazali katere vsebine so najbolj zanimive in na ta način so narasli ogledi vsebin. Statistiko obiskov beležim in prikazujem na številne načine in izkazali so se za pravilno razmišljanje. S portalom radijske postaje se trudim, da bi tudi v prihodnje ostal konkurenčen na trgu. Zavedam se, da se metode pri taki stvari vedno nadgrajujejo, zato se učim kako najboljše posodobiti obstoječi portal. Želim, da je poslušalcu radijske postaje oz. obiskovalcu spleta čimbolj enostaven za uporabo. Vse cilje in želje sem tudi navedel v ločenem poglavju 12. Upam, da mi nekatere stvari uspe izboljšati že kmalu.



# Literatura

- [1] Definicija spletnega portala. Dostopno na:  
<http://www.businessdictionary.com/definition/portal.html>,  
september 2013
  
- [2] .Net Framework. Dostopno na:  
<http://msdn.microsoft.com/en-us/library/zw4w595w.aspx>,  
september 2013
  
- [3] Vodič za postavitev prve spletne strani v okolju Asp.Net. Dostopno na:  
[http://www.w3schools.com/aspnet/webpages\\_intro.asp](http://www.w3schools.com/aspnet/webpages_intro.asp),  
september 2013
  
- [4] Diplomaska naloga Asp.Net. Dostopno na:  
[http://rc.fmf.uni-lj.si/matija/OpravljeneDiplome/PetraUdir\\_DiplomskaNaloga.pdf](http://rc.fmf.uni-lj.si/matija/OpravljeneDiplome/PetraUdir_DiplomskaNaloga.pdf),  
september 2013
  
- [5] RAM. Dostopno na:  
<http://www.webopedia.com/TERM/R/RAM.html>, september 2013
  
- [6] URL. Dostopno na:  
<http://searchnetworking.techtarget.com/definition/URL>, september  
2013
  
- [7] SQL. Dostopno na:  
<http://www.w3schools.com/sql/>, september 2013

- [8] MS SQL. Dostopno na:  
<http://www.host-shopper.com/what-is-ms-sql.html>, september 2013
- [9] Programski jezik C#. Dostopno na:  
[http://www.webopedia.com/TERM/C/C\\_sharp.html](http://www.webopedia.com/TERM/C/C_sharp.html), september 2013
- [10] Asp.Net. Dostopno na:  
<http://www.techterms.com/definition/aspnet>, september 2013
- [11] HTML. Dostopno na:  
[http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp), september 2013
- [12] CSS. Dostopno na:  
[http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp), september 2013
- [13] META oznake. Dostopno na:  
[http://www.webopedia.com/TERM/M/meta\\_tag.html](http://www.webopedia.com/TERM/M/meta_tag.html), september 2013
- [14] Javascript. Dostopno na:  
[http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp), september 2013
- [15] XML. Dostopno na:  
[http://www.w3schools.com/xml/xml\\_what\\_is.asp](http://www.w3schools.com/xml/xml_what_is.asp), september 2013
- [16] RSS Dostopno na:  
<http://www.whatisrss.com/>, september 2013
- [17] JSON. Dostopno na:  
<http://www.w3schools.com/json/>, september 2013
- [18] JQuery. Dostopno na:  
<http://www.w3schools.com/jquery/>, september 2013
- [19] Raba piškotkov. Dostopno na:  
<http://piskotki.net/kaj-so-piskotki-in-kako-delujejo/>, september 2013

- 
- [20] Smernice Informacijskega pooblaščenca o rabi piškotkov. Dostopno na: [https://www.ip-rs.si/fileadmin/user\\_upload/Pdf/smernice/Smernice\\_o\\_uporabi\\_piskotkov.pdf](https://www.ip-rs.si/fileadmin/user_upload/Pdf/smernice/Smernice_o_uporabi_piskotkov.pdf), september 2013
- [21] Tehnologija LINQ. Dostopno na: <http://msdn.microsoft.com/en-us/library/bb397926.aspx>, januar 2014
- [22] Facebook Developers. Dostopno na: <https://developers.facebook.com/docs/plugins/share/>, september 2013
- [23] Googlovo orodje za spletne skrbnike. Dostopno na: <http://www.google.com/webmasters/>, november 2013
- [24] Googlovo orodje Analize. Dostopno na: <http://www.google.com/analytics/>, november 2013
- [25] DNS. Dostopno na: <http://www.howtogeek.com/122845/htg-explains-what-is-dns/>, november 2013
- [26] Primerjava odzivnosti spletne in lokalne aplikacije. Dostopno na: <http://www.c-sharpcorner.com/blogs/416/windows-forms-application-versus-web-application.aspx>, januar 2014
- [27] Tip slovar (angl. Dictionary). Dostopno na: <http://www.dotnetperls.com/collections>, januar 2014
- [28] DIS slovarček. Dostopno na: <http://dis-slovarcek.ijs.si>, februar 2014
- [29] Friendly URLs. Dostopno na: [http://www.codeguru.com/csharp/.net/net\\_asp/using-friendly-urls-in-asp.net-web-forms.htm](http://www.codeguru.com/csharp/.net/net_asp/using-friendly-urls-in-asp.net-web-forms.htm), februar 2014
- [30] Flash predvajalnik. Dostopno na: <http://get.adobe.com/flashplayer/>, februar 2014

- [31] Tip MIME. Dostopno na:  
<http://www.aibn.com/help/Learn/mimetypes.html>, februar 2014
- [32] HTTP Response Headers. Dostopno na:  
[http://technet.microsoft.com/en-us/library/cc771148\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc771148(v=ws.10).aspx),  
februar 2014
- [33] DFS replikacija. Dostopno na:  
<http://technet.microsoft.com/en-us/library/cc771058.aspx>,  
februar 2014
- [34] Zrcaljenje baze. Dostopno na:  
<http://technet.microsoft.com/en-us/library/ms189852.aspx>,  
februar 2014
- [35] Robustno kopiranje s skripto robocopy. Dostopno na:  
<http://technet.microsoft.com/en-us/library/cc733145.aspx>,  
februar 2014
- [36] Microsoftova domena. Dostopno na:  
<http://windows.microsoft.com/en-us/windows7/what-is-the-difference-between-a-domain-a-workgroup-and-a-homegroup>, februar 2014
- [37] JSON Validator. Dostopno na:  
<http://jsonlint.com/>, februar 2014
- [38] ER Diagram. Dostopno na:  
<http://searchsqlserver.techtarget.com/definition/entity-relationship-model>, februar 2014