

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO  
FAKULTETA ZA MATEMATIKO IN FIZIKO

Nejc Župec

# Twitter Bootstrap in razvoj spletnega repozitorija za Cacti

DIPLOMSKO DELO  
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN MATEMATIKA

MENTOR: izr. prof. dr. Janez Demšar

Ljubljana, 2012

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

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*



Št. naloge: 00015/2012

Datum: 12.04.2012

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

Kandidat: **NEJC ŽUPEC**

Naslov: **TWITTER BOOTSTRAP IN RAZVOJ SPLETNEGA REPOZITORIJA ZA CACTI**

**TWITTER BOOTSTRAP AND DEVELOPMENT OF WEB REPOSITORY FOR CACTI**

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

S selitvijo vedno večjega števila aplikacij na splet ter podatkov in procesiranja v oblake postaja zanesljivo delovanje omrežne infrastrukture ključno za normalno delo in življenje. S tem narašča tudi pomen orodij za nadzor pravičnega delovanja omrežnih naprav. Med odprtokodnimi orodji za ta namen je eno najbolj popularnih Cacti. Njegovo širšo uporabo ovira potreba po tem, da mora uporabnik sam definirati predloge za zajem podatkov in izris grafov ali pa jih iskati po spletu, na katerem pa za to ni posebnih repozitorijev.

Zasnujte spletni portal, s katerim si bodo lahko uporabniki Cactija izmenjevali predloge grafov in skript za zajem podatkov. Obenem izkoristite razvoj sistema tudi za analizo uporabnosti razširjene knjižnice Twitter Bootstrap pri hitri postavitvi spletnih portalov.

Mentor:

prof. dr. Janez Demšar



Dekan Fakultete za računalništvo in informatiko:

prof. dr. Nikolaj Zimic



Dekan Fakultete za matematiko in fiziko:

akad. prof. dr. Franc Forstnerič

## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Nejc Župec, z vpisno številko **63090134**, sem avtor diplomskega dela z naslovom:

*Twitter Bootstrap in razvoj spletnega repozitorija za Cacti*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Janeza Demšarja,
- 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 1. septembra 2012

Podpis avtorja:

*Zahvaljujem se sodelavcem Poloni, Primožu in Jerneju iz podjetja NIL Podatkovne komunikacije, d.o.o., ki so mi dali idejo za to diplomsko nalogo in mi z nasveti pomagali pri njeni izvedbi.*

*Zahvaljujem se tudi mentorju, izr. prof. dr. Janezu Demšarju, za posluh pri izbiri teme in ostalo pomoč.*

# Kazalo

**Povzetek**

**Abstract**

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Cilji diplomske naloge . . . . .	5
<b>2</b>	<b>Cacti</b>	<b>6</b>
2.1	Kaj je Cacti? . . . . .	6
2.1.1	Razvoj . . . . .	6
2.2	Kaj sestavlja Cacti in kako deluje? . . . . .	8
2.2.1	Kako deluje Cacti? . . . . .	8
2.2.2	Poller . . . . .	9
2.2.3	RRDtool . . . . .	9
2.3	Grafi . . . . .	11
2.4	Branje podatkov iz omrežnih naprav . . . . .	11
2.5	Predloge . . . . .	12
2.5.1	Primer predloge za graf . . . . .	12
2.5.2	Primer predloge za napravo . . . . .	13
<b>3</b>	<b>Twitter Bootstrap</b>	<b>14</b>
3.1	Razvoj . . . . .	14
3.2	Tehnologija in zgradba . . . . .	15
3.3	Primer enostavnega spletnega obrazca . . . . .	16

## KAZALO

<b>4 Spletni portal Templator</b>	<b>19</b>
4.1 Podroben opis portala . . . . .	19
4.1.1 Moderator . . . . .	20
4.1.2 Administrator . . . . .	23
4.2 Pregled uporabljenih tehnologij . . . . .	23
4.2.1 HTML . . . . .	23
4.2.2 CSS . . . . .	24
4.2.3 JavaScript in knjižnica jQuery . . . . .	25
4.2.4 Programski jezik PHP . . . . .	26
4.2.5 MySQL . . . . .	26
4.3 Organizacija kode . . . . .	28
4.4 Glavne komponente spletne aplikacije . . . . .	29
4.4.1 Prijava in registracija . . . . .	30
4.4.2 Iskalnik in filtriranje . . . . .	30
4.4.3 Priporočilni sistem . . . . .	31
4.4.4 Urejevalnik tinyMCE . . . . .	34
4.4.5 Zagotavljanje standardizacije predlog . . . . .	35
4.4.6 Nadzorna plošča . . . . .	35
<b>5 Sklepne ugotovitve</b>	<b>37</b>
<b>Priloge</b>	<b>43</b>

# Povzetek

V okviru te diplomske naloge je nastal spletni repozitorij Templator, ki omogoča shranjevanje predlog za grafe in naprave, ki jih uporablja Cacti. Cacti je sistem, ki omogoča omrežno monitoriranje. Templator skrbi, da so vse predloge zbrane na enem mestu, standardizirane in sistematično urejene. Vsebuje iskalnik, priporočilni sistem, omogoča prijavo/registracijo za uporabnike ter uvoz in izvoz predlog v obliki XML. V diplomskem delu je tudi razloženo, kako poteka razvoj predlog za grafe in naprave ter kako Cacti sploh deluje. Za razvoj portala je bilo uporabljeno Twitter Bootstrap ogrodje, ki v zadnjem času postaja zelo priljubljeno. Izkazalo se je zelo dobro, saj omogoča na enostaven in hiter način narediti lep uporabniški vmesnik.

## Ključne besede

Twitter Bootstrap, Cacti, spletni portal, XML predloge, grafi, RRDtool

# Abstract

For the purpose of this thesis a web repository Templator was developed. It enables to save graph and device templates which are used by Cacti, a system for network monitoring. With Templator all templates are stored in one place, they are standardized and systematically organized. It consists of a search engine, a recommendation system, a manager for importing/exporting XML templates and enables users to login or register. The thesis also includes a basic overview how templates can be developed and how Cacti works. The web repository was developed with Twitter Bootstrap framework, which is nowadays very popular. It has proved a very powerful tool because it allows a fast and simple building of a website interface.

## Keywords

Twitter Bootstrap, Cacti, web portal, XML templates, graphs, RRDtool

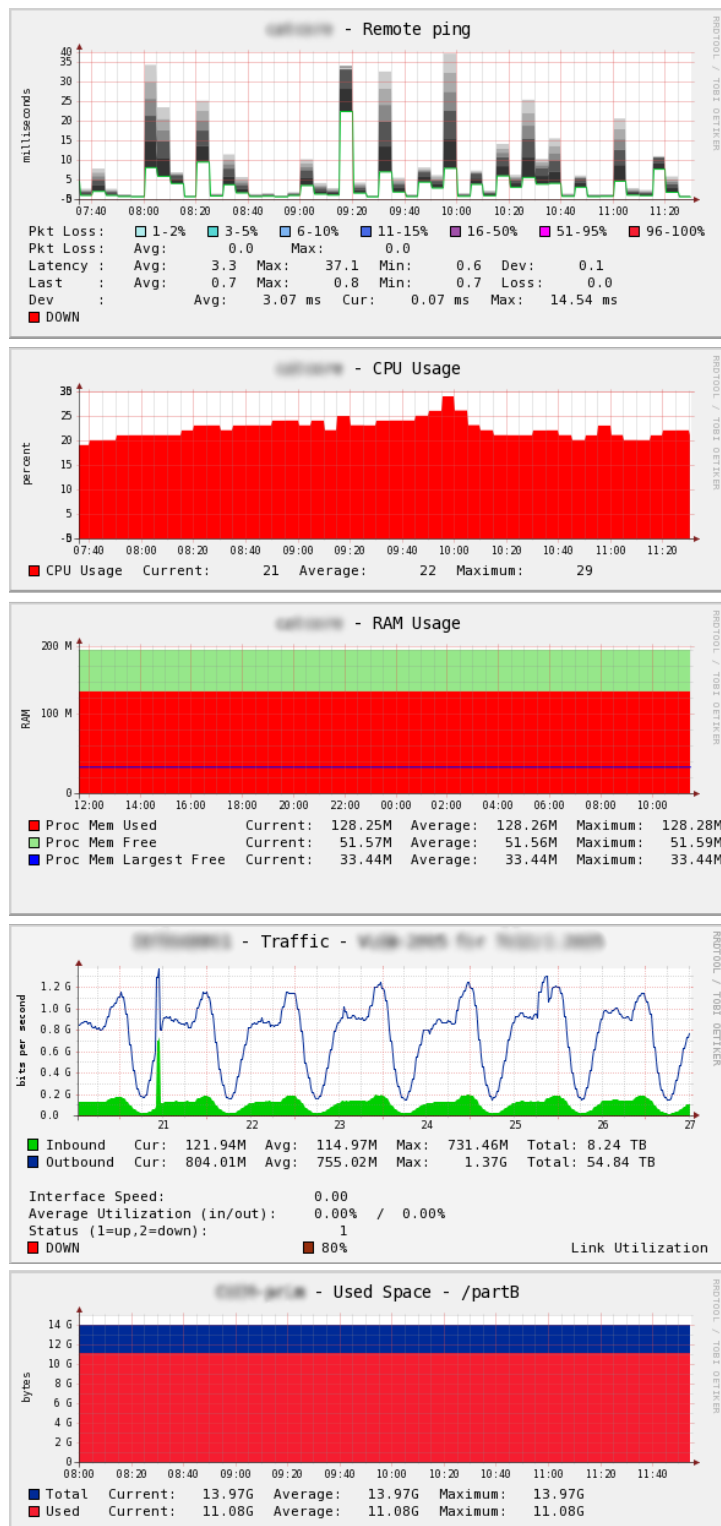
# Poglavje 1

## Uvod

Danes ves svet postaja vse bolj tehnološko napreden. Informacijska tehnologija je postala del vsakdana. Podjetja vse več storitev nudijo preko sodobnih omrežij. Ta omrežja povezujejo končne uporabnike preko številnih omrežnih naprav, kot so usmerjevalniki, stikala, dostopovne točke, požarne pregrade, itd., z zelo zmogljivimi strežniki in podatkovnimi centri. Zato je zelo pomembno, da so te povezave čim bolj zanesljive in stabilne. Vsak izpad ključne omrežne naprave ali strežnika lahko podjetjem povzroča velikanske izgube, ki so lahko zanje tudi pogubne.

Da to takih izpadov ne bi prišlo, oz. da bi bili čim krajši, danes veliko podjetij uporablja omrežno nadzorovanje (angl. network monitoring). Omrežno monitoriranje predstavlja spremljanje različnih parametrov na omrežnih napravah in strežnikih. Predvsem se preverja dosegljivost naprav in pomembne parametre, ki lahko vplivajo na slabšo dosegljivost oz. odzivnost storitev. Ti parametri lahko predstavljajo trenutno stanje procesorja, zasedenost pomnilnika, stanje diskovnega polja in še veliko drugih. Ko naprava postane nedosegljiva oz. vrednost parametra skoči izven dovoljenega območja, sistem opozori systemskega administratorja, da nekaj ni v redu. Tako ima ta možnost v čim krajšem času napako odpraviti oz. izpad preprečiti in s tem zagotoviti zanesljivo storitev.

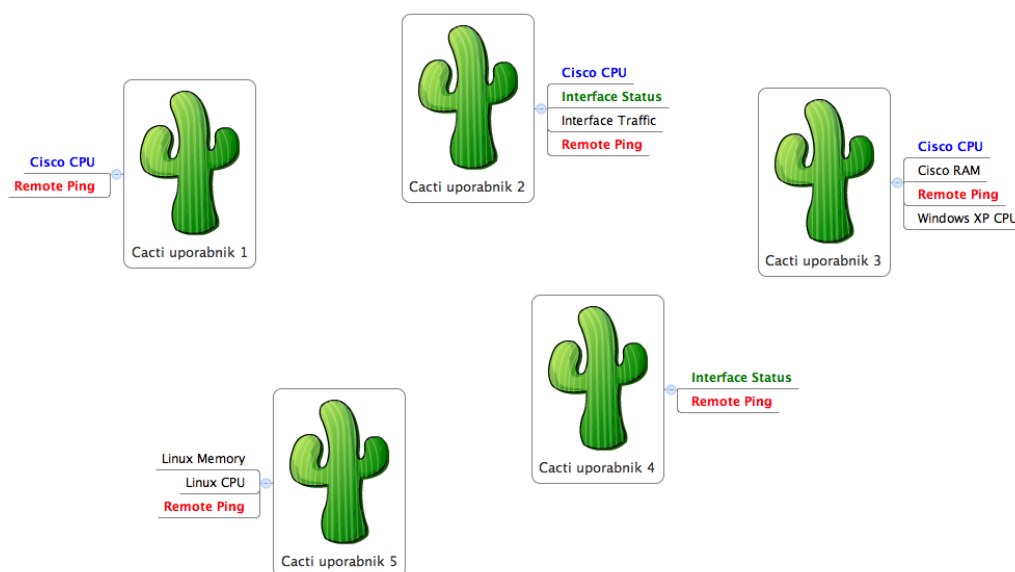
Danes je na trgu veliko sodobnih rešitev, ki omogočajo omrežno nad-



Slika 1.1: Primeri grafov, ki jih uporablja Cacti.

zorovanje. Obstajajo tako plačljive različice, kot tudi odprtokodne. Med slednjimi je zelo popularen Cacti [4].

Cacti je brezplačno odprtokodno orodje, ki temelji na RRDtool-u [5]. RRDtool zna rokovati s podatki v obliki časovnih vrst in jih prikazati v obliki grafov. Primeri grafov so prikazani na sliki 1.1. S pomočjo teh grafov lahko spremljamo številne parametre, ki lahko vplivajo na dosegljivost storitve ali pa nas zanimajo zgolj informativno. Spremljamo lahko skoraj vse, kar je merljivega. Potrebno je le definirati izgled grafa in od kje naj se pobirajo podatki. Ta postopek nam vrne predlogo (v obliki XML), ki je prenosljiva. Cacti namreč omogoča uvoz in izvoz predlog za grafe, prav tako pa tudi predloge za posamezne naprave, ki lahko vsebujejo več različnih predlog za grafe. Na primer, predloga za Ciscov usmerjevalnik vsebuje predloge za grafe, ki prikazujejo trenutno stanje procesorja, stanje pomnilnika, zasedenost pasovne širine na vmesnikih in tako naprej.

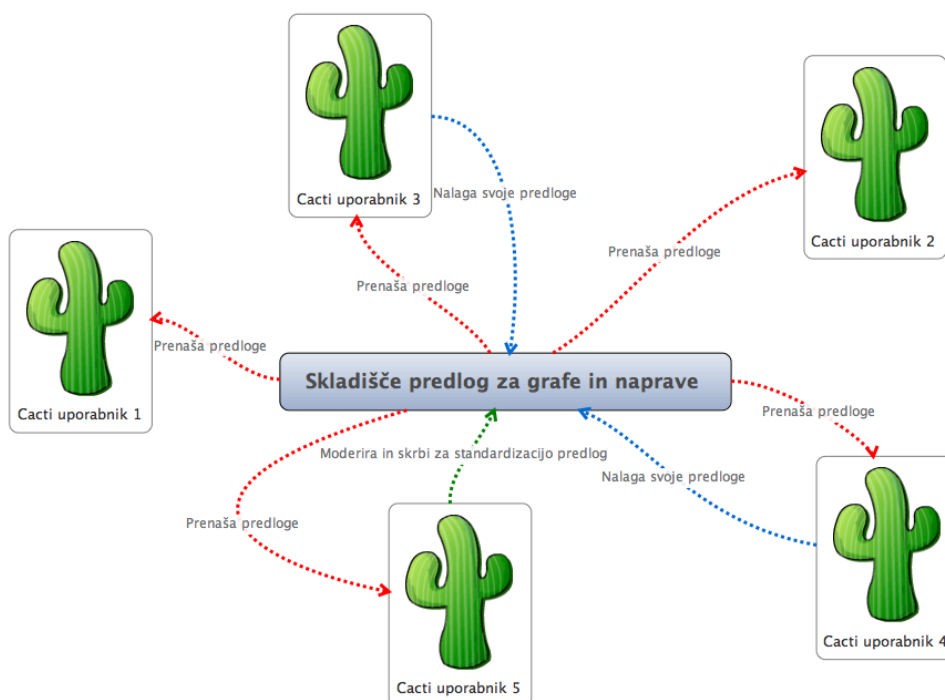


Slika 1.2: Diagram, ki prikazuje različne Cacti uporabnike in njihove grafe.

Z vsako tako predlogo je lahko zelo veliko dela. Potrebno je napisati skripte, ki omogočajo zajem podatkov, dobro razmisliti, kako naj se podatki

smiselno predstavijo in vse skupaj mora tudi stabilno delovati. Trenutno mora vsak uporabnik Cactija sam poskrbeti za razvoj grafov. Na sliki 1.2 lahko vidimo namišljen primer, ki predstavlja 5 različnih uporabnikov Cactija (kaktus predstavlja logotip Cactija). Vsak izmed njih je razvil nekaj svojih grafov. Iz primera se zelo lepo vidi, da je vsak izmed njih razvil graf, ki meri časovne zakasnitve (angl. remote ping) do naprav. Prav tako so trije uporabniki razvili graf, ki predstavlja trenutno stanje procesorja na Cisco napravah (angl. Cisco CPU).

Iz primera lahko vidimo, da vseh pet uporabnikov razvija zelo podobne grafe. Zato bi radi imeli rešitev, ki bi tem uporabnikom omogočala izmenjavo predlog. Ta rešitev mora uporabnikom omogočati, da lahko na enem mestu poiščejo že narejene predloge in prav tako jim mora omogočati, da lahko naložijo nove predloge. To idejo predstavlja diagram na sliki 1.3.



Slika 1.3: Diagram, ki prikazuje skupnost Cacti uporabnikov.

## 1.1 Cilji diplomske naloge

Cilj te diplomske naloge je torej pripraviti sodoben spletni portal, ki bo omogočal izmenjavo predlog za grafe in naprave. Da bo to skladišče zares uporabno, je potrebno izpolniti naslednje cilje:

1. portal naj ima privlačno in sodobno podobo,
2. je preprost za uporabo, enostaven in hitro odziven,
3. vsebuje sodoben iskalnik, ki uporabnikom omogoča enostavno iskanje,
4. ima priporočilni sistem, ki uporabnikom prikaže podobne predloge za grafe oz. naprave,
5. vse predloge naj imajo standardizirano obliko,
6. omogoča urejanje in brisanje predlog.

Prav tako želimo na praktičnem primeru pokazati, kako poteka razvoj spletne strani s pomočjo Twitter Bootstrap CSS ogrodja, ki je trenutno zelo popularen. Predvsem bi radi izpostavili prednosti uporabe tovrstnega ogrodja.

# Poglavje 2

## Cacti

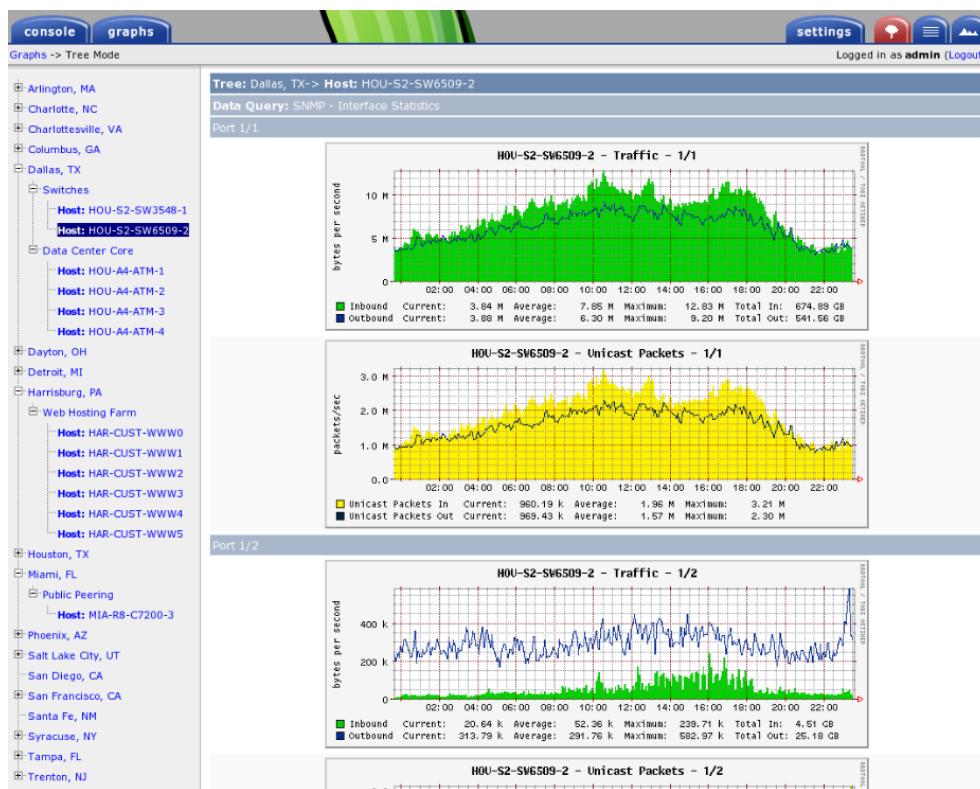
### 2.1 Kaj je Cacti?

Cacti je spletno orodje, ki je namenjeno omrežnemu nadzoru (angl. network monitoring). Je odprtokoden, zaščiten z licenco GNU - General Public License. Osnovan je na orodju RRDtool [5], ki omogoča enostavno shranjevanje podatkov v obliki časovnih vrst. Cacti nadgrajuje RRDtool z enostavnim vmesnikom, s katerim lahko dodajamo in urejamo grafe, sistematično prikazujemo grafe na drevesu ter uvažamo in izvažamo predloge za grafe. Omogoča tudi enostavno upravljanje z uporabniki, katerim lahko dodelimo različne pravice.

#### 2.1.1 Razvoj

Prva verzija je nastala leta 2001. Napisal jo je Ian Berry, ki je takrat delal pri manjšem ponudniku spletnih storitev. Želel je monitorirati vse naprave, ki so jih imeli v svojem omrežju. V tistem času ni bilo na voljo veliko različnih orodij, ki bi to omogočali. Obstajal je RRDtool, ki ga je bilo težko uporabljati in MRTG [23], ki je bil premalo prilagodljiv. V tistem času se je ravno naučil programskega jezika PHP in MySQL-a, zato je na osnovi teh dveh tehnologij pričel z razvojem Cactija. Leta 2004 je pridobil številne nove odprtokodne razvijalce, ki so pokrsbeli, da je Cacti postal stabilen, zanesljiv in skalabilen.

V naslednjih letih se je oblikovala odprtokodna skupnost, ki še danes skrbi za razvoj novih funkcionalnosti, dokumentacijo in spletni forum. Za leto 2013 je napovedan Cacti 1.0, ki naj bi prinesel številne novosti. To so: AJAX, dodatna podpora RRDtool-u, internacionalizacija in še veliko drugih.

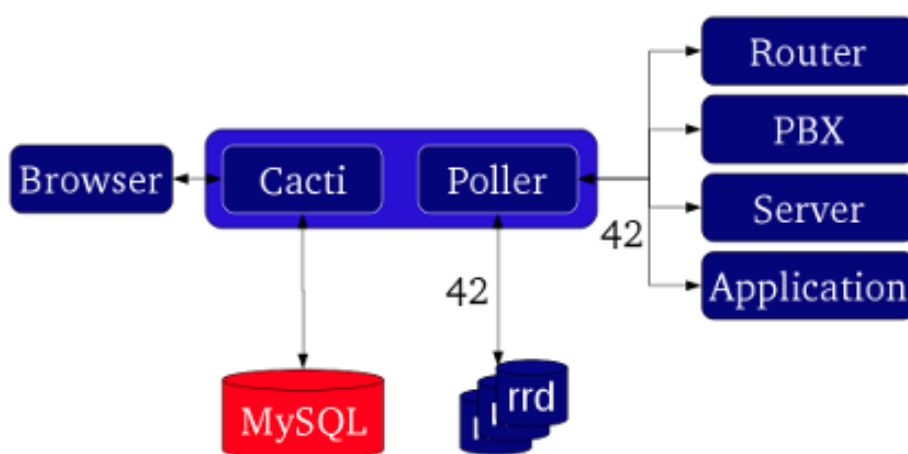


Slika 2.1: Cacti - drevesni pregled grafov.

Osnovni Cacti je možno razširiti s pomočjo vtičnikov (angl. plugins). Cacti namreč vsebuje PIA (Plugin Architecture), ki je neke vrste API (Application programming interface). API vsebuje funkcije, s pomočjo katerih lahko dostopamo do jedra Cactija. Vtičnike lahko razvijamo sami, lahko pa jih prenesemo iz uradne spletne strani [25]. Med njimi je zelo popularen vtičnik `thold` [24], ki omogoča nastavljanje alarmov za posamezne attribute, ki jih spremljamo v Cactiju.

## 2.2 Kaj sestavlja Cacti in kako deluje?

Cacti je sestavljen iz treh glavnih komponent: spletni strežnik (vsebuje spletni portal Cacti in poller), MySQL strežnik in RRDtool. Poller je preko omrežja povezan z napravami, ki jih želimo spremljati. Preko spletnega brskalnika pa upravljamo s celotnim sistemom.



Slika 2.2: Cacti - struktura [26].

### 2.2.1 Kako deluje Cacti?

Cacti ves čas v ozadju izvaja postopek, ki gre skozi tri pomembne faze: pridobivanje podatkov, shranjevanje podatkov in prikaz podatkov v obliki grafov. Za pridobivanje podatkov skrbi poller, za shranjevanje in prikaz pa RRDtool. Rezultat tega postopka so grafi, do katerih lahko dostopamo preko spletnega vmesnika. Ta nam tudi omogoča urejanje grafov in upravljanje z uporabniki, katerim lahko tudi določimo različne pravice.

### 2.2.2 Poller

Pomemben del Cactija predstavlja poller, ki iz naprav pridobiva stanje posameznih parametrov. Ti podatki se potem shranijo v RRD datoteke. Poller je proces, ki ga poganja časovni razporejevalnik cron [17] na določen časovni interval. Praviloma se poganja na 5 min.

**Pridobivanje podatkov:** Cron požene poller, ta pa postopoma začne poganjati skripte, ki pridobivajo željene parametre iz monitoriranih naprav. Te podatke lahko pridobi na več različnih načinov. Najbolj razširjen način je pobiranje podatkov preko SNMP-ja. Ker pa vse naprave tega ne omogočajo, Cacti omogoča pobiranje podatkov tudi s pomočjo skript. Ko poller pobere vse podatke, jih RRDtool zapiše v RRD datoteke.

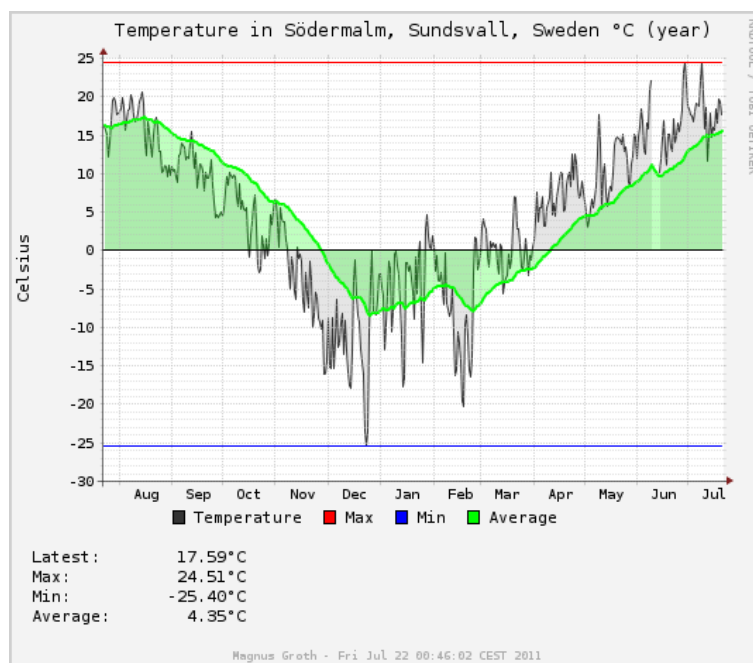
Cacti vsebuje dve implementaciji pollerja. Prva je napisana v PHPju in se imenuje `cmd.php`. Ta verzija pollerja je primerna za nadziranje manjšega števila naprav, ker je PHP relativno počasen. Poller mora svoje delo opraviti v manj kot 5 minutah, če želimo shranjevati podatke na 5 min. Pri večjem številu naprav, npr. 100, se 5 minut hitro preseže. Zato so razvijalci napisali hitrejši poller, v programskem jeziku C. Imenuje se `spine`. Ta omogoča boljšo skalabilnost. Pri manjšem številu naprav pa je `cmd.php` vseeno hitrejši. Uporabnik se lahko sam odloči, katero implementacijo bo uporabljal.

### 2.2.3 RRDtool

RRDtool je odprtokodni, visoko zmogljiv sistem za shranjevanje in prikazovanje podatkov, ki so predstavljeni v obliki časovnih vrst. Časovne vrste so pri monitoriranju omrežnih naprav zelo pogoste (CPU vrednosti, količina prometa na vmesnikih, temperatura). Razvil ga je Tobias Oetiker. Predstavlja nadgradnjo sistema MRTG, ki sicer velja za enostavnejšega od RRDtool-a, vendar pa je premalo prilagodljiv. RRDtool ima dve pomembni nalogi:

**Shranjevanje podatkov:** Podatke shranjuje v RRD (Round Robin Database) datoteke. Podatki se shranjujejo tako, da skozi čas velikost datotek ne narašča, ampak ko količina podatkov naraste do določene meje, se podatki začnejo prepisovati od začetka. S tem zagotovimo, da je velikost podatkov ves čas enaka in s tem obvladljiva.

**Prikaz podatkov:** RRDtool podatke prikazuje v obliki grafov. Pri tem si pomaga s konsolidacijskimi funkcijami, kot so AVERAGE (povprečna vrednost), MINIMUM (najnižja vrednost), MAXIMUM (najvišja vrednost) in LAST (zadnja vrednost). Te vrednosti lahko vidimo v legendi, ki se nahaja pod grafom (primer na sliki 2.3)



Slika 2.3: Graf narejen z RRDtool-om.

Cacti upravlja z veliko količino podatkov, zato je učinkovito shranjevanje ključno. Učinkovito shranjevanje omogoča RRDtool, ki je v primerjavi z relacijsko podatkovno bazo veliko hitrejši.

## 2.3 Grafi

Cacti vse podatke vizualizira v obliki grafov. Za vsak parameter, ki ga spremljamo, je zato potrebno imeti graf, ki ta podatek prikazuje. Preden lahko te podatke ustrezno prikazujemo, je potrebno pripraviti graf. To lahko naredimo na dva načina. Lahko uporabimo XML predlogo, ki jo uvozimo. Če pa predloge nimamo, je potrebno graf najprej razviti. Seveda je prva možnost veliko lažja, problem pa je, da je potrebno te XML predloge imeti. Cacti skupnost na svojih forumih sicer hrani nekaj osnovnih predlog, vendar so ne-standardizirane in slabo urejene. V tem diplomskem delu želimo ta problem rešiti s pomočjo Templatorja, spletnega repozitorija, ki omogoča shranjevanje predlog na enem mestu. Rešitev je predstavljena v poglavju 4.

## 2.4 Branje podatkov iz omrežnih naprav

Danes omrežja sestavljajo številne različne naprave (strežniki, usmerjevalniki, stikala, dostopovne točke, virtualna infrastruktura). Cacti pozna dva glavna načina branja podatkov iz oddaljenih naprav: s pomočjo protokola SNMP in z dodatnimi skriptami.

### SNMP

Protokol SNMP (Simple Network Management Protocol) se uporablja za upravljanje in nadzorovanje omrežnih naprav na daljavo. Ta protokol podpira večina omrežnih naprav (usmerjevalniki, stikala, dostopovne točke, požarne pregrade), ne pa vse. Sporočila se prenašajo preko enostavnega transportnega protokola UDP.

### Skripta

Ker vse naprave ne uporabljajo protokola SNMP, Cacti omogoča pobiranje podatkov s pomočjo skript. Skripte so lahko napisane v enem izmed treh skriptnih jezikov: PHP-ju, Perl-u ali Python-u.

## 2.5 Predloge

Za vsak graf, ki ga želimo prikazovati v Cactiju, je potrebno definirati številne parametre. Te parametre potrebuje RRDtool, da lahko zgenerira vse potrebno za risanje grafa. Zato ima Cacti mehanizem, ki lahko te nastavitve shrani v predloge. Te lahko potem enostavno izvažamo in uvažamo med različnimi Cactiji.

Vse predloge se shranijo v obliki XML. XML je označevalni jezik, ki omogoča opisati strukturirane podatke. Je dovolj enostaven, da ga lahko beremo ljudje in hkrati računalnik. Uporablja se za prenose podatkov.

### 2.5.1 Primer predloge za graf

Spodaj (izvorna koda: 2.1) se nahaja primer XML predloge za graf `Cisco - CPU Usage`. Sestavlja jo več parametrov:

**name:** ime predloge za graf,

**graph:** parametri, ki določajo osnovne lastnosti grafa: velikost, v kakšnem formatu je slika, višina in širina slike, minimalna in maksimalna vrednost, osnovna vrednost, ipd.

**items:** elementi, ki se rišejo na grafu: črte, področja (angl. area), komentarji, legende, vrednosti parametrov,...

**inputs:** podatki o tem, kako graf pridobiva podatke. Lahko gre za navadno SNMP metodo, ali pa drugačne metode, ki pridobivajo podatke s pomočjo skript,

**dependency:** tu se nahajajo predloge, od katerih je odvisna ta predloga.

---

**Izvorna koda 2.1: Primer predloge za graf Cisco - CPU Usage**

---

```
1 <cacti>
2   <hash_0000239a5e6d7781cc1bd6cf24f64dd6ffb423>
3     <name>Cisco – CPU Usage</name>
4     <graph>
5       <!-- parametri, ki določajo ime grafa, velikost, format slike,... -->
6     </graph>
7     <items>
8       <!-- elementi, ki se risejo na grafu (crte, legenda, vrednosti) -->
9     </items>
10    <inputs>
11      <!-- viri podatkov, SNMP oz. dostop do skript -->
12    </inputs>
13  </hash_0000239a5e6d7781cc1bd6cf24f64dd6ffb423>
14 </cacti>
15 <!-- ostale predloge, od katerih je ta predloga odvisna -->
```

---

## 2.5.2 Primer predloge za napravo

Predloge za napravo so le skupek predlog za grafe, ki jih določena naprava vsebuje. Poleg predlog za grafe pa vsebuje še glavo, v kateri je zapisano ime predloge in katere predloge za graf vse vsebuje.

---

**Izvorna koda 2.2: Primer predloge za Cisco Router**

---

```
1 <cacti>
2   <hash_020023cae6a879f86edacb2471055783bec6d0>
3     <name>Cisco Router</name>
4     <graph_templates>hash_0000235d...9966|hash_00002...f79ce</graph_templates>
5     <data_queries>hash_040023d75e406fdeca4fcef45b8be3a9a63cbc</data_queries>
6   </hash_020023cae6a879f86edacb2471055783bec6d0>
7 </cacti>
8 <!-- ostale predloge, od katerih je ta predloga odvisna -->
```

---

## Poglavje 3

# Twitter Bootstrap

Twitter Bootstrap [7] je zbirka sodobnih orodij, s katerimi lahko na enostaven način zastavimo osnovno ogrodje sodobnih spletnih strani. Je odprtokoden in zaščiten z licenco Apache License 2.0. Vsebuje številne lepo in sodobno oblikovane HTML komponente, kot so tipografija, tabele, obrazce, gumbe, ikone, navigacijo, poleg njih tudi JavaScript komponente, kot so modalna okna, zaslonski namigi, spustni meniji in še veliko drugih ter mrežo, ki omogoča enostavno razporeditev elementov.

V zadnjem času je na trgu prisotnih vse več različnih brskalnikov in naprav, zato je enoten izgled spletnih strani zelo težko doseči. Twitter Bootstrap pa razvijalcem omogoča prav to, enoten izgled v skoraj vseh brskalnikih. Postaja vse bolj popularen, saj omogoča na enostaven način zgraditi lep, moderen in funkcionalen vmesnik. Njegovo popularnost dokazuje tudi trenutno prvo mesto na GitHub-ovi lestvici [9] najbolj popularnih projektov.

### 3.1 Razvoj

Z razvojem Bootstrap-a sta pričela Mark Otto in Jacob Thornton, zaposlena pri Twitterju. Ugotovila sta, da prihaja pri razvoju spletnih vmesnikov v njihovem podjetju do nedoslednosti, zato sta pričela z majhno ekipo graditi ogrodje (angl. framework), ki bi to preprečilo. Rodil se je Twitter Bootstrap.

Sprva so ga uporabljali le v podjetju Twitter, potem pa sta projekt avgusta 2011 odprla za vso javnost in ga preselila na GitHub. Danes je repozitorij med svoje priljubljene dodalo že prek 36.000 razvijalcev s celega sveta. Ta številka še vedno vztrajno raste.

## 3.2 Tehnologija in zgradba

Knjižnica Bootstrap temelji na uporabi tehnologije LESS [10]. LESS je predprocesor CSS, ki generira pravila CSS. Omogoča nam, da razširimo statična CSS pravila z dinamičnimi, v katerih lahko uporabljamo tudi spremenljivke, funkcije, aritmetične operacije in gnezdenje.

Delovanje LESS si oglejmo na primeru 3.1.

Izvorna koda 3.1: Primer uporabe tehnologije LESS

---

```
1 /* definicija spremenljivk */
2 @barva_ozadja: #111;
3 @rdeca: #842210;
4 @rob: 1px;
5
6 /* stavki less, ki omogocajo uporabo aritmetike, spremenljivk in funkcij */
7 .obrazec {
8     color: @barva_ozadja;
9     border-left: @rob * 3;
10    border-right: @rob * 4;
11    border-color: desaturate(@rdeca, 10%);
12 }
13
14 /* rezultat */
15 .obrazec {
16     color: #333;
17     border-left: 1px;
18     border-right: 2px;
19     border-color: #7d2717;
20 }
```

---

Kot lahko vidimo, omogoča enostaven vnos spremenljivk. Spremenljivka lahko npr. predstavlja barvo ali pa širino robu. Te spremenljivke lahko potem uporabimo za izračun novih vrednosti, ki so odvisne od teh spremenljivk. To pride pogosto zelo prav, že zaradi narave problema. Pogosto so, na primer, širine slik odvisne od ostalih komponent. Z uporabo spremenljivk lahko tak problem enostavno rešimo. Prav tako je zelo enostavno spreminjati velikost spletne strani, to pa je v zadnjem času še posebej pomembno, ker je na trgu veliko različnih naprav, ki imajo različne velikosti zaslonov.

LESS lahko uporabljamo na dva načina. Prvi način je, da stavke LESS prevaja v navadna pravila CSS neposredno v brskalniku. Pri tem načinu potrebujemo dodatno kodo v JavaScriptu, ki skrbi za prevajanje. Drugi način pa omogoča, da LESS stavke prevedemo že vnaprej in na stran naložimo le pravila CSS, brez dodatnega JavaScripta. Slednjega uporablja tudi Twitter Bootstrap.

To nam omogoča, da si lahko knjižnico Bootstrap prilagodimo po svojih lastnih željah. Definiramo lahko barve gumbov, tabel, navigacijske vrstice, velikost besedila, vrsto pisave in še veliko ostalih parametrov. Ko vse te parametre definiramo, se enostavno požene prevajalnik, ki zgenerira CSS in kodo v JavaScriptu, ki jo lahko vključimo na svojo spletno stran.

LESS tako prinaša veliko prednost pred uporabo samega CSS-a. Na enem mestu lahko popravljamo eno bazično barvo, ki lahko spremeni celoten izgled spletne strani. Pri navadnih pravilih CSS bi to lahko pomenilo, da je vnos potrebno popravljati na več mestih, kar predstavlja nepotrebno izgubo časa.

### 3.3 Primer enostavnega spletnega obrazca

Poglejmo si na primeru, kako lahko uporabimo Twitter Bootstrap. Radi bi naredili enostaven obrazec za prijavo uporabnika, ki ima standardno obliko. Prav tako naj bo obrazec prikazan na vseh sodobnih brskalnikih, kar se da enako.

### Primer enostavnega obrazca

Uporabniško ime

Vnesite Vase uporabniško ime.

Geslo

Vpisite Vase geslo.

Zapomni si me

Slika 3.1: Primer enostavnega obrazca.

Preden se lotimo dela je potrebno uvoziti Bootstrap na našo spletno stran. To naredimo tako, da v glavo dodamo (angl. head) vrstico, ki pove, kje se nahaja datoteka CSS. V našem primeru je to četrta vrstica (koda se nahaja v prilogi 1). Koda v CSS datoteki poskrbi, da so vsi HTML konstrukti lepo oblikovani in podobno vidni v vseh brskalnikih.

Če bi radi uporabili tudi JavaScript komponente, potem je potrebno uvoziti tudi dve JavaScript knjižnici. Seveda je to bootstrap.js datoteka, ki nam omogoča uporabo modalnih oken, spustne menije in tako naprej. Vendar pa za svoje delovanje nujno potrebuje tudi jQuery knjižnico. V našem primeru, se oba vnosa nahajata v peti in šesti vrstici.

Ko uvozimo vse potrebne datoteke, se lahko lotimo pisanja kode HTML. Da nam ni potrebno ugibati, kako se kakšno komponento lepo oblikuje, je najlažje, da poiščemo dokumentacijo Twitter Bootstrap-a [30]. V njej lahko najdemo majhne delčke (angl. snippet) kode, ki jih lahko vključimo v svojo novo spletno stran in preoblikujemo tako, da ustreza našim zahtevam. Bootstrap omogoča zelo enostavno kombiniranje in se ga tako hitro naučimo

uporabljati. To je pravzaprav vse, kar potrebujemo za uporabo te knjižnice. Naš končni rezultat predstavlja slika 3.1.

Spletni obrazci so le eno izmed številnih orodij, ki jih nudi Bootstrap. Zelo enostavna je tudi uporaba mreže, na kateri lahko enostavno razporejamo elemente. V trenutni verziji so omogočene tudi dinamične mreže, ki se prilagajajo velikosti zaslona.

## Poglavje 4

# Spletni portal Templator

Spletni portal izdelan za to diplomsko nalogo se imenuje Templator. Je rešitev za uporabnike Cactija, katerim omogoča hitrejšo postavitve nadzornega sistema, saj jim ni več potrebno razvijati številnih predlog za grafe in naprave, ki predstavljajo glavno komponento Cactija. Templator predstavlja zbirko najrazličnejših predlog na enem mestu, katere lahko uporabniki enostavno prenesejo in uvozijo v Cacti. Je portal, ki vsebuje številne komponente sodobnih spletnih strani. Celoten vmesnik (angl. interface) je izdelan s pomočjo ogrodja Twitter Bootstrap, kar zagotavlja moderen in preprost izgled ter kompatibilnost med različnimi brskalniki in napravami.

### 4.1 Podroben opis portala

Vsi uporabniki lahko dostopajo do repozitorija, ki je razdeljen na dva dela. Prvi je namenjen skladiščenju predlog za grafe, drugi pa predlogam za naprave. V obeh delih repozitorija se nahaja iskalnik, ki ima vgrajeno funkcijo samodokončanja (angl. autocomplete), kar uporabniku omogoča hitrejše in lažje iskanje. Ko uporabnik najde iskano predlogo in klikne nanjo, se mu odpre stran z opisom. Pri vsaki predlogi lahko vidi, kdaj je bila vnešena oz. nazadnje spremenjena, njeno dokumentacijo in podobne predloge, ki jih zagotavlja priporočilni sistem. Če je predloga za graf, potem se prikaže tudi

slika s primerom grafa oz. če gre za predlogo naprave, se prikažejo vse slike s primeri grafov, ki jih ta predloga vsebuje. Če je uporabniku predloga všeč in izpolnjuje njegove želje in zahteve, jo lahko enostavno prenese. Zgenerira se datoteka XML, ki je primerna za uvoz v Cacti.

The screenshot shows the Templator web interface. At the top, there is a navigation bar with 'Templator', 'Home', 'Repository', 'Documentation', and 'About'. A user profile 'admin' is visible in the top right. Below the navigation bar, the 'Repository' section is active, with buttons for '+ Host template' and '+ Graph template'. There are tabs for 'Hosts' and 'Graphs'. A search bar is present with a 'Search' button and a 'Clear' button. Below the search bar, it says 'Filtered templates: 7'. A table lists the templates with columns for 'Template name', 'Tags', and 'Options'. Each row has a 'Download' button (green), an 'Edit' button (orange), and a 'Deactivate' button (red).

	Template name	Tags	Options
1	VIP VM	host vm vip	Download Edit Deactivate
2	Cisco Router	host cisco router	Download Edit Deactivate
3	Cisco PIX - 7.x.x	host cisco pix 7.x.x	Download Edit Deactivate
4	Cisco Catalyst Fixed	host cisco catalyst fixed	Download Edit Deactivate
5	Windows 2000/XP	host windows 2000/xp	Download Edit Deactivate
6	Cisco Switch Modular	host cisco switch modular	Download Edit Deactivate
7	Cisco CUCM	host cisco CUCM	Download Edit Deactivate

At the bottom of the table, there is a pagination control with 'Previous', '1', '2', '3', and 'Next' buttons. Below the table, there is a copyright notice: '© Copyright 2012'.

Slika 4.1: Izgled repozitorija predlog za naprave.

### 4.1.1 Moderator

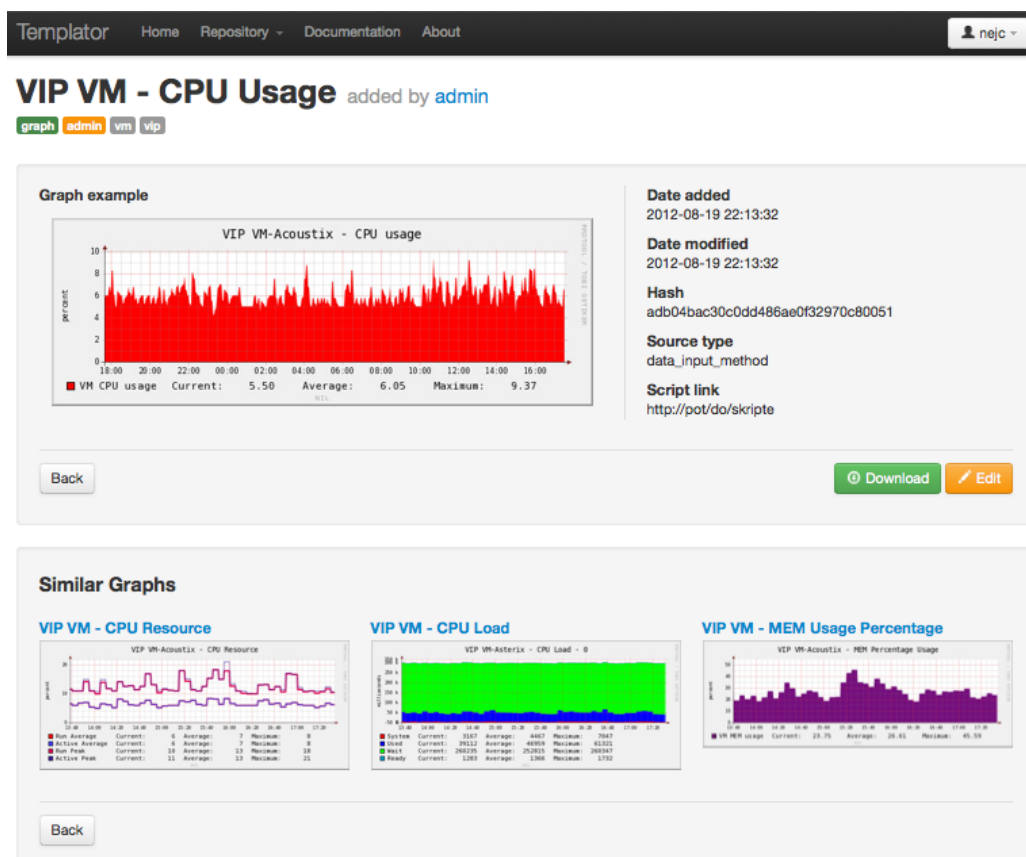
Ko se uporabnik registrira, postane moderator. S tem si pridobi dodatne pravice, ki mu omogočajo dodajanje novih predlog za grafe in naprave. Dodajanje novih predlog poteka prek spletnega obrazca, ki poskrbi, da je vnos čim bolj standardiziran. To pomeni, da uporabnika vodi skozi celoten postopek in mu pri tem svetuje (pomoč ob strani), npr. kako naj predlogo poimenuje. Obstajata dva spletna obrazca, prvi vodi uporabnika skozi postopek za nala-

ganje predlog za grafe, drugi pa za dodajanje predlog za naprave. Poglejmo si, kako uporabnik vnaša predlogo za graf.

1. Sprva mora vnesti čim več različnih oznak (angl. tags). Ta postopek uporabnika "prisili", da dobro razmisli, kaj ta predloga sploh predstavlja. Te oznake potem uporablja tudi iskalnik za filtriranje in priporočilni sistem. Zato je ključno, da je ta del vnosa nove predloge dobro opravljen.
2. Ko uporabnik vnese vse oznake, se na podlagi le-teh zgenerira ime, ki predstavlja smernico, kako naj bo predloga poimenovana. To ime lahko uporabnik tudi prilagodi.
3. Tretji korak predstavlja nalaganje datoteke XML, v kateri je predloga definirana. Hkrati se sproži funkcija, ki iz datoteke prebere zgoščeno vrednost in prebrska po bazi, če predloga morda že obstaja. V primeru, da predloga z enako zgoščeno vrednostjo že obstaja, se uporabniku prikaže modalno okno rdeče barve, ki ga na to opozori. Ta funkcija tudi pregleda, za kakšno vrsto grafa gre, ali pridobiva podatke preko SNMP ali pa preko skripte. Če podatke pobira preko skripte, od uporabnika zahteva tudi vnos skripte, v nasprotnem primeru pa lahko nadaljuje z naslednjim korakom.
4. V četrtem koraku uporabnik naloži sliko, ki prikazuje primer grafa. Ta mora biti dovolj velika, potem pa funkcija v ozadju prilagodi velikost in jo shrani v bazo. V bazo se vneseta dve različni velikosti, glede na potrebe prikazovanja.
5. Peti korak predstavlja pisanje dokumentacije. Vnos poteka s pomočjo urejevalnika TinyMCE [11]. Podana je osnovna struktura vnosa dokumentacije. Uporabnik mora vpisati, kaj graf predstavlja, kako pridobiva podatke, posebnosti, za katere tipe naprave je graf primeren itd.
6. V šestem koraku se požene t.i. pregledovalnik. To je funkcija v JavaScriptu, ki preveri, ali je uporabnik upošteval vsa pravila. Je vnesel

dovolj oznak, ali se je držal navodil za ime predloge, ali je naložil dovolj veliko sliko itd. Ta potem izpiše povzetek, če so vse te točke izpolnjene, v nasprotnem primeru pa uporabnika opozori, da se ni držal vseh navodil.

Na koncu uporabnik predlogo samo še shrani in ta se vnese v bazo. Ta predloga še vedno ni vidna v repozitoriju. O vnosu je obveščen administrator, ki mora vnos preveriti. Če se administratorju zdi vnos primeren, predlogo potrди in ta postane vidna za obiskovalce spletnega repozitorija. Podobno se vnaša tudi predloge za naprave. Primer, kako izgleda vnešen opis grafa, je viden na sliki 4.2.



Slika 4.2: Primer opisa predloge za graf.

### 4.1.2 Administrator

Poleg navadnih uporabnikov in moderatorjev pa obstaja še tretji način dostopa. To je administratorski način. Ta naziv lahko pridobi vsak moderator, katerega določi administrator. Administrator ima na voljo še dve dodatni funkciji, ki jih moderator nima. Ima dostop do potrjevanja novih predlog, katere lahko tudi ureja in komentira. Prav tako pa ima tudi dostop do nadzorne plošče, ki omogoča urejanje uporabnikov, oznak, vseh predlog in ročno poganjanje skripte, ki poskrbi za priporočilni sistem.

## 4.2 Pregled uporabljenih tehnologij

Templator je narejen s klasičnimi spletnimi tehnologijami: HTML, CSS, JavaScript, PHP in MySQL. Poganja ga spletni strežnik Apache [12], ki velja za najbolj razširjenega na svetu. Sledijo kratki opisi posameznih tehnologij.

### 4.2.1 HTML

HTML (HyperText Markup Language) [13] je označevalni jezik za oblikovanje večpredstavnostnih dokumentov. Razvil ga je Tim Berners-Lee leta 1990 v CERN-u. HTML sestavlja velika množica značk (angl. tags). Vsaka značka predstavlja različne elemente HTML. To so spletne povezave (`<a>`), odstavki (`<p>`), skoki v novo vrstico (`<br />`), slike (`<img>`), tabele (`<table>`), naslovi (`<h1>`) in še veliko drugih. Večinoma se pojavljajo v parih, npr., znački `<p>` in `</p>` označujeta začetek in konec odstavka. Brskalnik te značke sprva prebere in jih interpretira, potem pa jih vizualno prikaže.

V veljavo počasi prihaja HTML5 standard, ki prinaša številne novosti, katere bodo programerjem olajšale delo. Med novostmi najdemo tudi nove značke, ki so bolj semantične: `<footer>`, `<nav>`, `<audio>`, itd. Dodali so tudi nove API-je, ki omogočajo enostavno risanje na ploskev (angl. canvas), urejanje dokumentov, povleci in spusti (angl. drag and drop),... Najnovejše verzije spletnih brskalnikov te novosti že lepo podpirajo, zato novosti

standarda HTML5 izkorišča tudi Twitter Bootstrap.

---

Izvorna koda 4.1: Primer kode HTML

---

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Pozdravljen, svet!</title>
5   </head>
6   <body>
7     <h1>Pozdravljen, svet!</h1>
8   </body>
9 </html>
```

---

### 4.2.2 CSS

CSS (Cascading Style Sheet) je preprost slogovni jezik, s katerim lahko določimo izgled dokumenta, ki je napisan z označevalnim (angl. markup) jezikom, npr. HTML. Uradno je nastal leta 1996, ko je število spletnih strani hitro raslo in je izgled postajal vse bolj pomemben. Z njim lahko v obliki pravil definiramo barvo, velikost, odmike, obrobe in položaje elementov. Bistvo CSS-a je v tem, da ločimo strukturo strani od njene predstavitve, kar na dolgi rok olajša vzdrževanje spletne strani.

---

Izvorna koda 4.2: Primer kode CSS

---

```
1 div.login {
2   margin-top: 120px;
3   background-color: #fff;
4   border-radius: 10px;
5   border: 1px solid #ccc;
6 }
```

---

Sintaksa je zelo enostavna. Vsako pravilo je sestavljeno iz selektorja (v primeru 4.2 je selektor `div.login`), v zavutih oklepajih pa definiramo vrednosti posameznih lastnosti.

Največja težava CSS-a je v tem, da ga različni brskalniki razumejo različno. Tako je težko zagotoviti enak izgled v vseh brskalnikih. Razvijalci si tako

pomagajo s t.i. CSS triki, ki vsaj okvirno zagotovijo enak izgled. Podobni triki so uporabljeni tudi v knjižnici Twitter Bootstrap.

### 4.2.3 JavaScript in knjižnica jQuery

JavaScript je objektni programski jezik, ki se primarno uporablja na strani odjemalca, lahko pa tudi na strani strežnika s pomočjo dodatnih knjižnic. Prvič so ga uporabili leta 1995 v spletnem brskalniku Netscape Navigator [22]. Po sintaksi spominja na programski jezik C, po imenu pa na Javo, čeprav nimata veliko skupnih lastnosti. JavaScript med drugim omogoča enostavno interakcijo s komponentami HTML preko drevesa DOM (Document Object Model). Omogoča nam tudi validacijo obrazcev, izdelavo enostavnih spletnih iger, animacije, nalaganje novih vsebin z AJAX-om, brez osveževanja oz. novega nalaganja strani. Prav AJAX nam omogoča, da so spletne strani hitre, odzivne in prijazne do uporabnika. Danes ga uporabljajo vse moderne spletne strani, kot so Facebook, Google, LinkedIn, ipd.

Izvorna koda 4.3: Primer kode JavaScript

---

```
1 $('#search-graphs').typeahead ({
2   source: function (typeahead, query) {
3     $.ajax({
4       type: "POST",
5       url: "/action/ajax/get_all_graphs.php",
6       dataType: "json",
7       data: "query=" + query,
8       success: function (data) {
9         typeahead.process(data);
10      }
11    });
12  }
13 });
```

---

Templator uporablja tudi knjižnico jQuery [20], ki programerjem poenostavi pisanje skript. Danes velja za najbolj popularno knjižnico JavaScript, saj je nameščena na več kot 5500 spletnih strani, izmed 10.000 najbolj prilju-

bljenih [19]. Zaradi svoje enostavnosti jo uporablja tudi Twitter Bootstrap in posledično Templator.

#### 4.2.4 Programski jezik PHP

PHP (Hypertext Preprocessor) je programski jezik, ki omogoča razvoj dinamičnih spletnih vsebin na strani strežnika. Nastal je leta 1994 in ga danes uporablja več kot 75 % spletnih strani. Podoben je jezikom C in Perl, popularen pa je predvsem zaradi svoje enostavne sintakse. Omogoča kombiniranje kode HTML s kodo PHP, ki omogoča dinamično spreminjanje vsebine. Na osnovi PHP-ja je nastalo več ogrodij (angl. framework), Zend Framework, Symfony ter najpopularnejši CMS-ji (Content Management System), kot so Wordpress, Joomla in Drupal.

Izvorna koda 4.4: Primer kode PHP

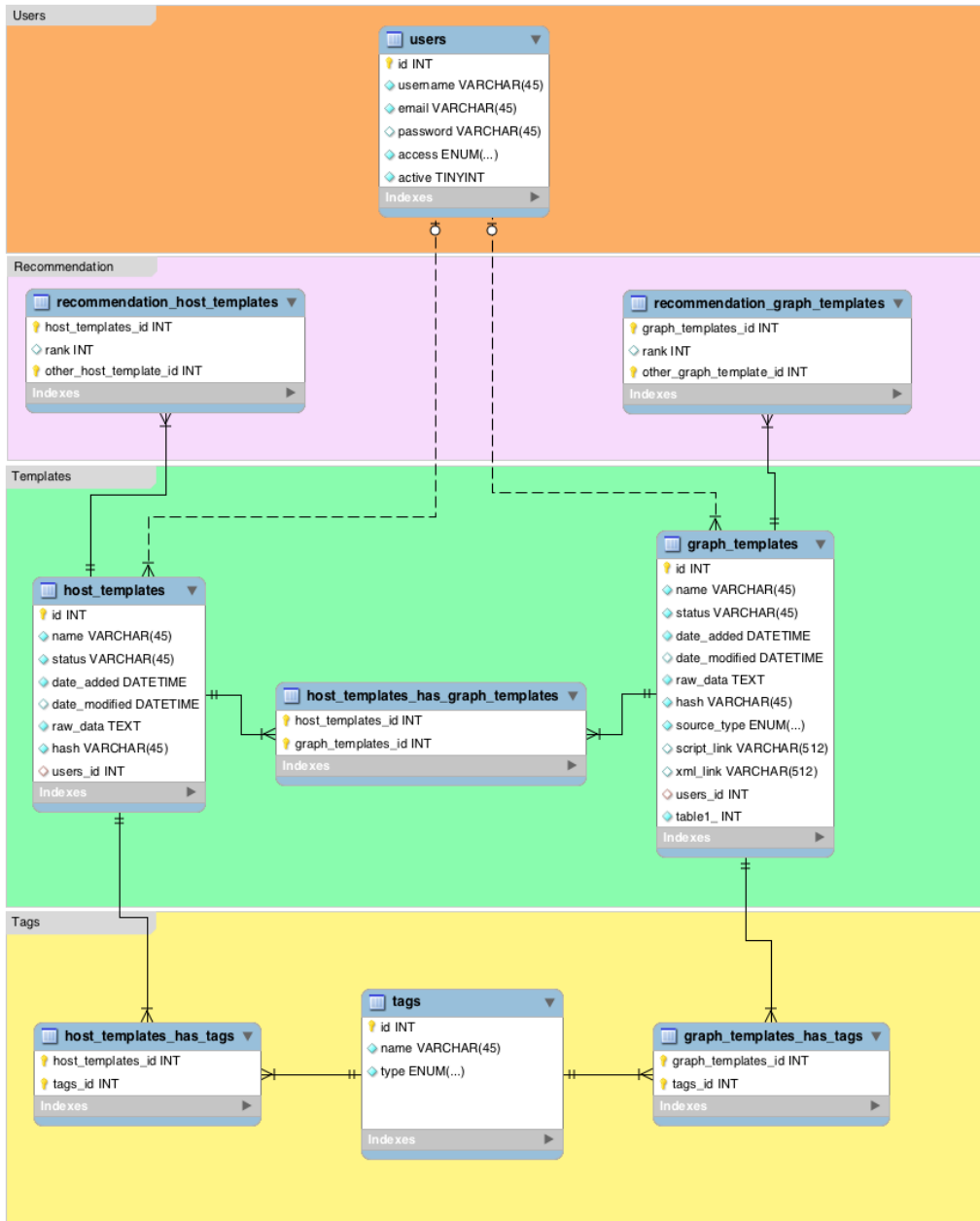
---

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Pozdravljen svet!</title>
5   </head>
6   <body>
7     <?php
8       $name="Nejc";
9       print "<h1>Pozdravljen, $name!</h1>";
10    ?>
11   </body>
12 </html>
```

---

#### 4.2.5 MySQL

MySQL je odprtokodna podatkovna baza. Spada med relacijske baze in jo lahko upravljamo s pomočjo jezika SQL. Prva verzija je izšla leta 1995 in danes velja za najbolj popularno bazo na svetu, predvsem zaradi svojih zmogljivosti, zanesljivosti in enostavne uporabe.

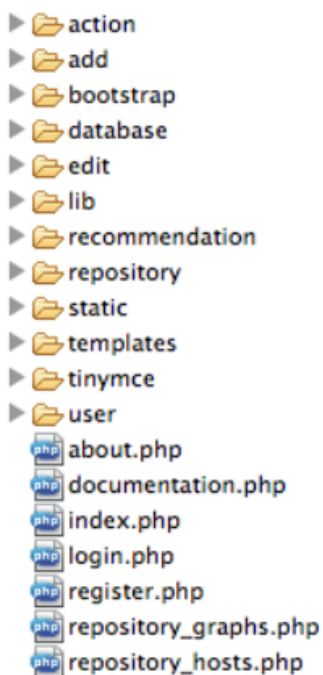


Slika 4.3: Model podatkovne baze.

V osnovi MySQL nima grafičnega vmesnika, ki bi omogočal enostavno administracijo podatkovnih zbirk. Vsebuje le dostop preko ukazne vrstice. Obstaja pa uradno orodje z grafičnim vmesnikom, to je MySQL Workbench, ki ga razvija Oracle in je brezplačno. Omogoča načrtovanje, razvoj in administracijo podatkovne baze MySQL. S pomočjo tega orodja je bil razvit tudi podatkovni model za spletni portal Templator, ki je prikazan na sliki 4.3.

### 4.3 Organizacija kode

Templator je enostavna aplikacija in je napisana v 'čistem' PHPju, torej ni osnovan na nobenem ogrodju (angl. framework). Koda je razdeljena v tri večje sklope, po zgledu modela MVC (Model-View-Controller), model-pogled-kontroler.



Slika 4.4: Organizacija kode.

Na sliki 4.4 je predstavljena organizacija kode po datotekah in mapah.

**Model** predstavlja mapa `lib`, v kateri se nahajajo skripte in funkcije, ki dostopajo do podatkovne baze MySQL. Za dostop se uporablja razred `MySQLi` (i pomeni *improved*), ki med drugim omogoča uporabo vnaprej pripravljenih stavkov (anlg. *prepared statements*). To pohitri delovanje, saj se koda prevede le enkrat, prav tako pa preprečuje SQL-vrinjanje (anlg. *SQL-injection*).

**Pogled** predstavlja večina ostalih datotek. V njih se nahaja predvsem HTML koda, ki tvori osnovno ogrodje spletne strani. Tu se uporablja Twitter Bootstrap knjižnica.

**Kontroler** se nahaja v mapi `action`. Tu se nahajajo skripte, ki jih kličejo zahtevki AJAX in spletni obrazci. Vsaka funkcija najprej preveri, če ima uporabnik dovolj pravic za izvajanje, nato pa se izvedejo in vrnejo rezultat.

**Ostale** datoteke:

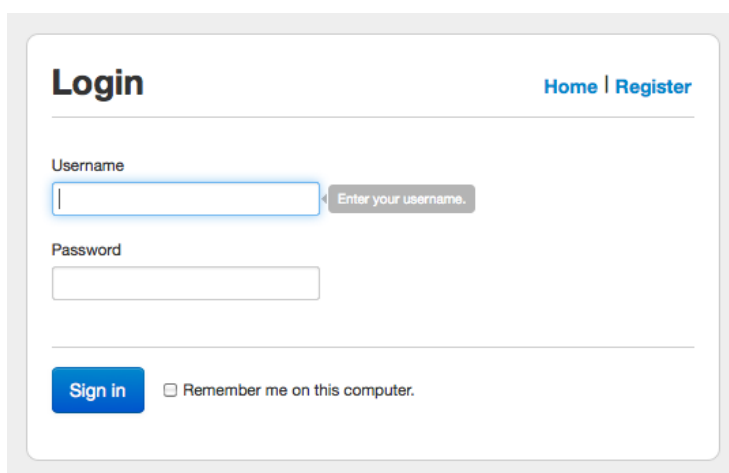
1. `bootstrap` - Mapa, ki vsebuje vse potrebno za delovanje Twitter Bootstrap knjižnice. To so HTML in JavaScript datoteke.
2. `recommendation` - Tu se nahajajo skripte za priporočilni sistem.
3. `static` - Vsebuje slike, CSS in JavaScript kodo.
4. `tinymce` - Vsebuje JavaScript kodo, ki omogoča uporabo TinyMCE urejevalnika.

## 4.4 Glavne komponente spletne aplikacije

V tem poglavju so opisane vse glavne komponente, ki sestavljajo spletni portal `Templator`. Vsaka komponenta je predstavljena s scenarijem uporabe, opisom delovanja, sliko za lažjo predstavo ter krajšim opisom, kako je narejena. Pomembna komponenta je tudi Twitter Bootstrap, ki pa je že opisana v svojem poglavju 3.

### 4.4.1 Prijava in registracija

Spletni portal Templator omogoča prijavo in registracijo. Oba obrazca izkoriščata zmogljivosti knjižnice Twitter Bootstrap. Na tem mestu predvsem velja omeniti zaslonske namige (angl. tooltip), ki jih je zelo enostavno uporabiti. Potrebno je le nekaj klicev JavaScript, ki so zelo lepo opisani v dokumentaciji Bootstrapa.



The image shows a login form with the following elements:

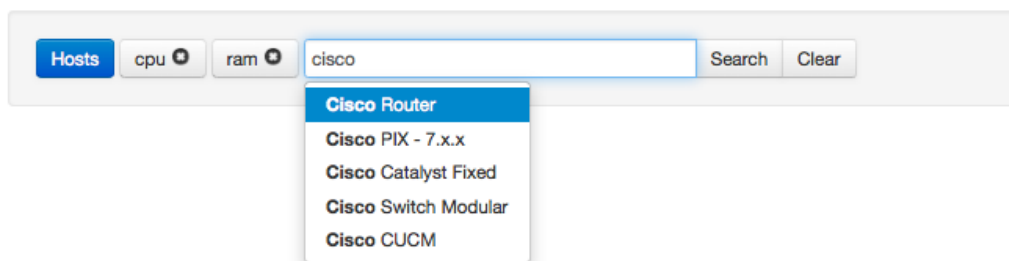
- Title:** Login
- Navigation:** Home | Register
- Username:** A text input field with a tooltip that says "Enter your username." when the field is focused.
- Password:** A text input field.
- Buttons:** A blue "Sign in" button and a checkbox labeled "Remember me on this computer."

Slika 4.5: Prijavni obrazec.

V ozadju se izvajajo tudi regularni izrazi, ki preverjajo, kaj je uporabnik vnesel. Predvsem od vsakega uporabnika zahtevajo kompleksno geslo. Če vnos ne ugotovi vsem zahtevam, uporabnika na to opozorijo dodatna navodila in rdeče obarvana vnosna polja.

### 4.4.2 Iskalnik in filtriranje

Iskalnik igra zelo pomembno vlogo pri takem spletnem repozitoriju. Ko enkrat število predlog naraste na nekaj 100, lahko dober iskalnik prinese veliko dodano vrednost in uporabniku omogoča najti tisto, kar išče. Naš iskalnik ima dve pomembni funkciji, ki uporabniku pomagata pri iskanju. To sta funkcija samodokončanja (angl. autocomplete) in filtriranje po oznakah.



Slika 4.6: Iskalnik s funkcijo samodokončaj in možnostjo filtriranja.

Samodokončanje je narejeno s pomočjo AJAXa (Asynchronous JavaScript and XML). Deluje tako, da za vsako črko, ki jo uporabnik vnese, brskalnik pošlje znak strežniku, strežnik pa odgovori z vsemi možnimi zadetki, ki vsebujejo ta podniz znakov. Če uporabnik izbere eno izmed ponujenih možnosti iz spustnega menija, ga strežnik preusmeri neposredno na izbrano predlogo.

Na sliki 4.6 je prikazano, kako deluje filtriranje. Ko uporabnik vnese `cpu` in `ram`, iskalnik išče le še med predlogami, ki imajo med oznakami `cpu` in `ram`. Tako lahko uporabnik hitro poišče vse predloge, ki vsebujejo določene oznake in si s tem pomaga pri iskanju.

### 4.4.3 Priporočilni sistem

Priporočilni sistemi so danes postali zelo popularni. Z njimi se srečujemo dnevno na največjih svetovnih portalih, kot so Facebook, Amazon, YouTube, Last.FM, itd. Ti portali ves čas zbirajo ogromne količine podatkov, ki jih potem z računalniško obdelavo spremenijo v ustrezna priporočila. Ta priporočila jim nudijo dodano vrednost na več različnih načinov. Uporabniku lahko ponudijo lažje iskanje, priporočijo podobne izdelke za nakup, podaljšajo čas obiska spletne strani, saj ljudem ponudijo vsebino, ki jih zanima, ipd.

Spletni portal Templator zaenkrat še nima tako velikih količin podatkov. Ima pa vsaka predloga vnešenih kar nekaj različnih oznak, s katerimi jo podrobno opisujejo. Te oznake nam zato omogočajo izdelavo zelo enostavnega

priporočilnega sistema tako, da za vsako predlogo poiščemo tiste predloge, ki imajo najbolj podobne oznake. To lahko izračunamo s pomočjo Jaccardovega podobnostnega koeficienta 4.1.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4.1)$$

Jaccardov podobnosti koeficient nam pove, kako zelo podobni sta si dve množici med seboj. Vsaka predloga je v tem primeru predstavljena z množico oznak. Koeficient vrača vrednosti na intervalu  $[0, 1]$ , pri čemer 0 pomeni, da sta množici povsem različni (nimata skupnih elementov), 1 pa pomeni, da sta množici enaki (vse elemente imata skupne).

Poglejmo si primer, kako lahko Jaccardov koeficient uporabimo, če želimo med seboj primerjati več predlog. Predloga **Cisco Router - CPU usage** ima naslednje oznake: Cisco, Router, CPU, 800, 1800, average, 5min, predloga **Cisco SCE - CPU usage** ima oznake: Cisco, SCE, CPU, average, 5min in predloga **Linux - RAM** ima oznake: Linux, CPU. Vsako predlogo lahko opišemo kot množico oznak:

$$A = \{Cisco, Router, CPU, 800, 1800, average, 5min\} \quad (4.2)$$

$$B = \{Cisco, SCE, CPU, average, 5min\} \quad (4.3)$$

$$C = \{Linux, CPU\} \quad (4.4)$$

Sedaj lahko med temi množicami izračunamo Jaccardove koeficiente.

$$J(A, B) = \frac{|\{Cisco, CPU, average, 5min\}|}{|\{Cisco, Router, CPU, 800, 1800, average, 5min, SCE\}|} = \frac{4}{8} = \frac{1}{2} \quad (4.5)$$

$$J(A, C) = \frac{|\{CPU\}|}{|\{Cisco, Router, CPU, 800, 1800, average, 5min, Linux\}|} = \frac{1}{8} \quad (4.6)$$

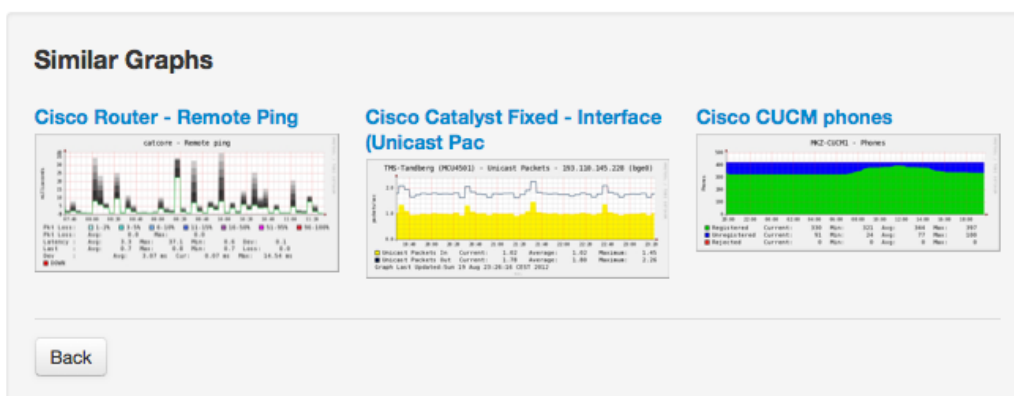
$$J(B, C) = \frac{|\{CPU\}|}{|\{Cisco, SCE, CPU, average, 5min, Linux\}|} = \frac{1}{6} \quad (4.7)$$

$$(4.8)$$

Iz primera lahko vidimo, da sta si na podlagi Jaccardovega koeficienta, predlogi A in B najbolj podobni, saj je njun koeficient najvišji, kar lahko sklepamo tudi po lastni presoji. Sedaj znamo ugotoviti, kdaj sta si dve predlogi podobni.

Da dobimo celoten priporočilni sistem, moramo za vsako predlogo poiskati najbolj podobne, glede na Jaccardov koeficient. To storimo tako, da za vsako predlogo izračunamo Jaccardov koeficient do vseh ostalih, te koeficiente sortiramo in tiste z najvišjo vrednostjo zapišemo v bazo.

Ta algoritem ima časovno kompleksnost  $O(n^2)$ , kjer  $n$  predstavlja število predlog. To pomeni, da je lahko algoritem počasen, če je predlog zelo veliko. Zaradi velike časovne zahtevnosti se skripto, ki izračuna priporočilni sistem, poganja ponoči (požene jo cron [17]) in ne ob vsakem dodajanju nove predloge. Administrator pa jo lahko požene tudi ročno, na nadzorni plošči v zavihku recommendation.



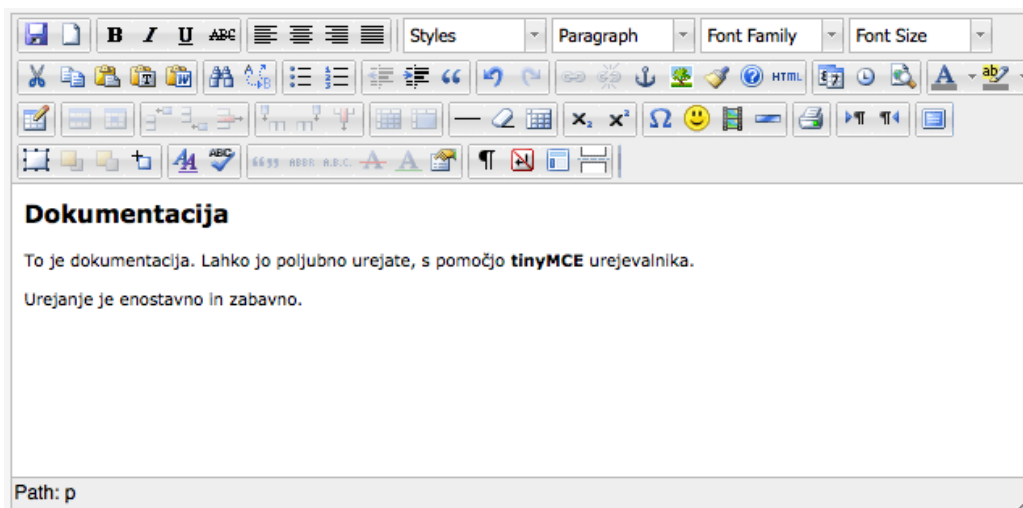
Slika 4.7: Priporočilni predlogi za graf Cisco Router - RAM Usage.

Spletni portal Templator ima dva taka enostavna priporočilna sistem, za obe vrsti predlog posebej. Primer priporočila za graf je na sliki 4.7. Dodana vrednost, ki jo prinašata, je predvsem v tem, da uporabnikom omogoča lažje iskanje med različnimi predlogami. Tu je še veliko prostora za izboljšave. Predvsem bi bilo dobro imeti veliko večje število vnešenih predlog. Dodatno bi lahko tudi obtežili določene oznake, ki so bolj pomembne ipd.

#### 4.4.4 Urejevalnik tinyMCE

Za vsako predlogo je potrebno vnesti dokumentacijo. Suhoparno besedilo brez oblikovanja hitro postane nepregledno. Rešitev, ki omogoča enostavno urejanje je spletni urejevalnik, ki omogoča urejanje kode HTML. V našem primeru si želimo, da je enostaven, omogoča osnovno urejanje besedila, ga je enostavno uporabljati in da je odprtokoden. Tvrstnih urejevalnikov je kar veliko, med njimi je tudi tinyMCE, ki ga uporablja spletni portal Templator.

TinyMCE [11] je enostaven WYSIWYG (what you see is what you get) urejevalnik, kar bi v prevodu pomenilo, tak kot je dokument med oblikovanjem, tak bo tudi, ko bo objavljen. Napisan je v JavaScript-u in je združljiv z vsemi modernimi spletnimi brskalniki.



Slika 4.8: Urejevalnik tinyMCE.

Deluje tako, da komponento `textarea` spremeni v urejevalnik, za kar poskrbi koda, ki se požene, ko uporabnik naloži stran. TinyMCE verzije 3.x omogoča zelo enostavno prilagajanje orodne vrstice in izgleda. Primer se nahaja na sliki 4.8.

### 4.4.5 Zagotavljanje standardizacije predlog

Eden izmed ciljev te diplomske naloge zahteva tudi to, da so predloge vnešene čim bolj podobno, na nekem standardnem nivoju, ki je vnaprej določen. To zagotavlja t.i. pregledovalnik pravil (angl. guideline checker), ki preveri, če je moderator res upošteval vsa pravila. Nahaja se na koncu vnosnega obrazca, ki se samodejno požene, ko uporabnik želi shraniti predlogo v bazo. Lahko pa ga požene tudi uporabnik sam s pomočjo gumba **Check**.

Pregledovalnik pravil ni nič drugega kot funkcije JavaScripta, ki gredo skozi vnosna polja in preverijo vsa vnaprej predpisana pravila. Ko konča pregled, zapiše rezultate v pregledni obliki. Uporabnik lahko vseeno prekrši napotke pregledovalnika, vendar pa se bo predloga vnesla z opozorilom, ki ga bo videl tudi administrator, ki bo predlogo potrjeval.



Slika 4.9: Guideline checker - preverjanje standardizacije.

### 4.4.6 Nadzorna plošča

Do nadzorne plošče lahko dostopa le administrator. Tu lahko pregleduje, ureja in briše vse pomembne podatke, ki jih vsebuje Templator.

Administrator ima več nalog. Najpomembnejša predstavlja pregledovanje novih predlog, ki jih posredujejo moderatorji. V zavihku pending lahko vidi vse predloge, ki še niso bile potrjene. Tu lahko predloge ureja, zbríše

ali pa napiše sporočilo moderatorju, kaj mora še popraviti, da bo predloga sprejemljiva za naš repozitorij.

Na naslednjem jezičku se nahaja administracija uporabnikov. Administrator lahko za vsakega uporabnika nastavlja nivo dostopa (moderator, administrator) oz. ga omogoči/onemogoči. Lahko pa ga tudi izbríše.

Tretji jeziček omogoča urejanje oznak. Vsako oznako je možno preimenovati in ji spremeniti tip. Obstaja namreč več različnih tipov oznak, kar omogoča večjo preglednost pri vnosu. Vsako oznako je možno tudi izbrisati, posledično se zbríšejo tudi vsi vnosi oznak pri vseh vnešenih predlogah.

Četrty jeziček je namenjen poganjanju skript za oba priporočilna sistema. Kot smo že omenili v poglavju 4.4.3, je izračun vseh priporočil časovno zahteven. Zato je skripte potrebno poganjati ročno.

Username	Email	Access	Active	Options
admin	admin@neki.si	admin	active	<a href="#">Edit</a> <a href="#">Delete</a>
jernej	jerry@gmail.com	admin	active	<a href="#">Edit</a> <a href="#">Delete</a>
miha	miha@example.com	moderator	inactive	<a href="#">Edit</a> <a href="#">Delete</a>
nejc	nejc@gmail.com	moderator	active	<a href="#">Edit</a> <a href="#">Delete</a>
janez	janez@hocevar.si	moderator	active	<a href="#">Edit</a> <a href="#">Delete</a>

© Copyright 2012

Slika 4.10: Nadzorna plošča.

## Poglavje 5

### Sklepne ugotovitve

Rezultat tega diplomskega dela predstavlja spletni portal Templator. Preverili smo tudi prednosti uporabe ogrodja Twitter Bootstrap, ki v zadnjem času postaja zelo popularno spletno orodje.

Templator predstavlja spletni repozitorij za predloge grafov in naprav, ki jih uporablja Cacti. Te predloge omogočajo uporabnikom Cactija hitro vzpostavitev sistema za monitoriranje. Templator zagotavlja standardiziran uvoz in enostaven izvoz predlog. Predloge so sistematično urejene, kar omogoča enostaven pregled, katere predloge že obstajajo in katere še ne. Vsebuje klasične komponente sodobnih spletnih aplikacij, kot so iskalnik, priporočilni sistem, upravljanje s podatki preko nadzorne plošče, prijavo/registracijo uporabnikov itd. Da smo lahko zagotovili sistematično ureditev, je bilo sprva potrebno preučiti, kako so predloge za Cacti sestavljene. Zatem je bilo potrebno pripraviti model podatkovne baze, ki te podatke shranjuje. Sledil je razvoj vmesnika in nato logike, ki vse komponente povezuje med seboj.

Trenutno skupnost Cacti zbira predloge, vendar jih je malo in so slabo in nesistematično urejene. Posledično jih je tudi težko najti, saj se nahajajo na njihovem uradnem spletnem forumu. Tako je vsak uporabnik Cactija prisiljen v iskanje predlog na spletu. Pri tem pa ni nobene garancije, da iskana predloga sploh obstaja. Ko uporabnik ugotovi, da predloge še ni, je

potreben razvoj, ki lahko zahteva veliko časa, kar številne uporabnike odvrne od uporabe Cactija.

Odlično se je izkazalo ogrodje Twitter Bootstrap, ki resnično omogoča hiter razvoj in izgradnjo lepega grafičnega vmesnika, ki je v teh časih ključen za uspeh na svetovnem trgu. V pomoč je zelo lepo in pregledno napisana dokumentacija. Ker za projektom stoji veliko število razvijalcev, se projekt lepo širi in vsak dan je na voljo več modulov, ki omogočajo še lažji razvoj spletnim razvijalcem.

Ta projekt svoje poti s tem diplomskim delom še ni zaključil. Prizadevamo si, da bi Templator postal uraden repozitorij in tako del odprtokodne skupnosti. S tem bi projekt dobil tudi nove razvijalce, ki bi lahko obstoječe funkcionalnosti še razširili. Upamo, da bo nekoč vseboval veliko zbirko predlog in s tem prispeval k boljši skupnosti.

# Slike

1.1	Primeri grafov, ki jih uporablja Cacti. . . . .	2
1.2	Diagram, ki prikazuje različne Cacti uporabnike in njihove grafe. . . . .	3
1.3	Diagram, ki prikazuje skupnost Cacti uporabnikov. . . . .	4
2.1	Cacti - drevesni pregled grafov. . . . .	7
2.2	Cacti - struktura [26]. . . . .	8
2.3	Graf narejen z RRDtool-om. . . . .	10
3.1	Primer enostavnega obrazca. . . . .	17
4.1	Izgled repozitorija predlog za naprave. . . . .	20
4.2	Primer opisa predloge za graf. . . . .	22
4.3	Model podatkovne baze. . . . .	27
4.4	Organizacija kode. . . . .	28
4.5	Prijavni obrazec. . . . .	30
4.6	Iskalnik s funkcijo samodokončaj in možnostjo filtriranja. . . . .	31
4.7	Priporočilni predlogi za graf Cisco Router - RAM Usage. . . . .	33
4.8	Urejevalnik tinyMCE. . . . .	34
4.9	Guideline checker - preverjanje standardizacije. . . . .	35
4.10	Nadzorna plošča. . . . .	36

# Literatura

- [1] R. Nixon. *Learning PHP, MySQL, and JavaScript: A Step-By-Step Guide to Creating Dynamic Websites (Animal Guide)*. O'Reilly, 2009.
- [2] S. Tahaghoghi, H. Williams. *Learning MySQL*. O'Reilly, 2007.
- [3] (2012) S. Koren. *iSlovar - terminološki slovar informatike*. Dostopno na: <http://www.islovar.org/>
- [4] (2012) Cacti: the complete RRDtool-based graphing solution. Dostopno na: <http://www.cacti.net/>
- [5] (2012) T. Oetiker. *RRDtool*. Dostopno na: <http://oss.oetiker.ch/rrdtool/>
- [6] (2012) Slika Cactijevega logotipa. Dostopno na: <http://en.wikipedia.org/wiki/File:Cacti3.png>
- [7] (2012) Twitter Bootstrap knjižnica. Dostopno na: <http://twitter.github.com/bootstrap/>
- [8] (2012) Twitter Bootstrap. Dostopno na: [http://en.wikipedia.org/wiki/Twitter\\_Bootstrap](http://en.wikipedia.org/wiki/Twitter_Bootstrap)
- [9] (2012) Github - Popular Starred Repositories. Dostopno na: <https://github.com/popular/starred>

- 
- [10] (2012) LESS (stylesheet language). Dostopno na:  
[http://en.wikipedia.org/wiki/LESS\\_\(stylesheet\\_language\)](http://en.wikipedia.org/wiki/LESS_(stylesheet_language))
- [11] (2012) TinyMCE urejevalnik. Dostopno na:  
<http://www.tinymce.com/>
- [12] (2012) Apache HTTP Server. Dostopno na:  
[http://en.wikipedia.org/wiki/Apache\\_HTTP\\_Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server)
- [13] (2012) HTML. Dostopno na:  
<http://en.wikipedia.org/wiki/Html>
- [14] (2012) MySQL. Dostopno na:  
<http://en.wikipedia.org/wiki/MySQL>
- [15] (2012) MySQL Workbench. Dostopno na:  
<http://www.mysql.com/products/workbench/>
- [16] (2012) Jaccard Similarity coefficient. Dostopno na:  
[http://en.wikipedia.org/wiki/Jaccard\\_index](http://en.wikipedia.org/wiki/Jaccard_index)
- [17] (2012) Cron. Dostopno na:  
<http://en.wikipedia.org/wiki/Cron>
- [18] (2012) AJAX. Dostopno na:  
[http://en.wikipedia.org/wiki/Ajax\\_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- [19] (2012) jQuery. Dostopno na:  
<http://en.wikipedia.org/wiki/JQuery>
- [20] (2012) jQuery knjižnica. Dostopno na:  
<http://jquery.com/>
- [21] (2012) HTML5. Dostopno na:  
<http://en.wikipedia.org/wiki/Html5>
- [22] (2012) Netscape Navigator. Dostopno na:  
[http://en.wikipedia.org/wiki/Netscape\\_Navigator](http://en.wikipedia.org/wiki/Netscape_Navigator)

- 
- [23] (2012) Multi Router Traffic Grapher. Dostopno na:  
<http://en.wikipedia.org/wiki/MRTG>
- [24] (2012) Vtičnik Thold za Cacti. Dostopno na:  
<http://docs.cacti.net/plugin:thold>
- [25] (2012) Seznam vtičnikov za Cacti. Dostopno na:  
<http://docs.cacti.net/plugins>
- [26] (2012) Slika, ki prikazuje shemo Cactija. Dostopno na:  
[http://docs.cacti.net/manual:087:2\\_basics.0\\_principles\\_of\\_operation](http://docs.cacti.net/manual:087:2_basics.0_principles_of_operation)
- [27] (2012) How do SNMP MIBs and OIDs work? Dostopno na:  
<http://www.paessler.com/knowledgebase/en/topic/653-how-do-snmp-mibs-and-oids-work>
- [28] (2012) Simple Network Management Protocol. Dostopno na:  
[http://en.wikipedia.org/wiki/Simple\\_Network\\_Management\\_Protocol](http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol)
- [29] (2012) Slika, ki prikazuje SNMP OID Tree. Dostopno na:  
[http://snmpnation.com/wp-content/uploads/SNMP\\_OID\\_MIB\\_Tree.png](http://snmpnation.com/wp-content/uploads/SNMP_OID_MIB_Tree.png)
- [30] (2012) Dokumentacija knjižnice Twitter Bootstrap:  
<http://twitter.github.com/bootstrap/components.html>
- [31] (2012) XML - Extensible Markup Language. Dostopno na:  
<http://en.wikipedia.org/wiki/XML>

# Priloge

---

## A Enostaven spletni obrazec

Izvorna koda 1: Primer zelo enostavnega obrazca.

---

```
1 <html lang="en">
2   <head>
3     <title>Primer zelo enostavne forme</title>
4     <link rel="stylesheet" href="/bootstrap/css/bootstrap.css" />
5     <script type="text/javascript" src="/static/js/jquery.js"></script>
6     <script type="text/javascript" src="/bootstrap/js/bootstrap.js"></script>
7   </head>
8
9   <body>
10    <form class="span5">
11      <legend>Primer zelo enostavne forme</legend>
12
13      <label>Uporabniksko ime</label>
14      <input name="uporabniksko_ime" type="text">
15      <span class="help-block">Vnesite Vase uporabniksko ime.</span>
16
17      <label>Geslo</label>
18      <input name="geslo" type="text">
19      <span class="help-block">Vpisite Vase geslo.</span>
20
21      <label class="checkbox">
22        <input type="checkbox"> Zapomni si me
23      </label>
24
25      <div class="form-actions">
26        <div class="pull-right">
27          <button type="button" class="btn">Preklici</button>
28          <button type="submit" class="btn btn-primary">Prijava</button>
29        </div>
30      </div>
31    </form>
32  </body>
33 </html>
```

---