Programski upravljana Wikipedija

Strusa, Janko

Undergraduate thesis / Završni rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:103959

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Janko Strusa

Zagreb, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentori:

Prof. dr. sc. Mario Essert, dipl. ing.

Student:

Janko Strusa

Zagreb, 2015.

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Zahvaljujem se voditelju rada prof. dr. sc. Mariju Essertu na razumijevanju, podršci i na stručnim savjetima.

Janko Strusa

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

ZAVRŠNI ZADATAK

Student: **Janko Strusa** Mat. br.: 0248002217

Naslov rada na hrvatskom jeziku: **Programski upravljana Wikipedija**

Naslov rada na engleskom jeziku: *Program controlled Wikipedia*

Opis zadatka:

Wikipedija kao otvorena enciklopedija već dugi niz godina dominira WEBom – najveći pretraživači (*Google, Yahoo*) postavljaju je po prioritetu ispred drugih izvora, a broj dokumenata i korisnika/čitatelja eksponencijalno raste (hrvatska wikipedija ima oko 140.000 članka, čime zauzima 39 mjesto između ukupno 285 nacionalnih wikipedija). Osim svoje enciklopedijsko-informacijske službe Wikipedija ima mogućnost programskog upravljanja, što je važna značajka u odnosu na faktografska postavljanja/editiranja. U ovom radu potrebno je istražiti upotrebu Wikipedije u smislu programskog upravljanja za računalni unos i obradbu podataka u njenom vlastitom radnom okviru (eng. *Wikipedija framework programming*).

1. Proučiti i opisati Wikipedija administriranje;
2. Zamisliti i isprogramirati Wikipediju u Python programu uz pomoć web2py programskog alata;
3. U tako dobivenu Wikipediju ugraditi obrasce za unos i provjeru podataka, te tvorbu podataka programskim putem;
4. Istražiti mogućnosti statističke obradbe podataka i njihove vizualizacije unutar Wiki okoliša.

Zadatak zadan:
25. studenog 2014.

Rok predaje rada:
1. rok: 26. veljače 2015.
2. rok: 17. rujna 2015.

Predviđeni datumi obrane:
1. rok: 2., 3., i 4 . ožujka 2015.
2. rok: 21., 22., i 23. rujna 2015.

Zadatak zadao:
Prof.dr.sc. Mario Essert

Predsjednik Povjerenstva:
Prof. dr. sc. Zoran Kunica

SADRŽAJ

SADRŽAJ I

POPIS SLIKA III

POPIS TABLICA V

SAŽETAK VI

SUMMARY VII

1. POVIJEST 1

1.1. Temelji wiki filozofije 1

1.2. Početak Wikipedije 3

2. USTROJ WIKI SUSTAVA 5

2.1.1. Organizacija 5

2.1.2. Namespaces 5

2.2. Sučelje 5

2.3. Administracija 7

2.3.1. Pretraživanje 7

2.3.2. Uređivanje članaka 7

2.3.3. Kreiranje članaka 9

2.3.4. Praćenje izmjena 10

2.3.5. Multimedijski privici 10

3. WEB2PY 12

3.1. Database Abstraction Layer (DAL) 13

3.1.1. Primjer korištenja baze 13

3.2. Markmin 14

3.3. Web2Py wiki 15

3.4. Administratorsko sučelje 15

4. ANGULARJS 17

4.2. Koncept rada 19

4.2.1. Kontroler 20

4.3. Korištenje unutar Web2Py okruženja 23

5. Wiki aplikacija rađena s Web2Py 24

5.1. Obrasci za unos, provjeru i tvorbu podataka programskim putem 31

5.2. Vizualizacija statističkih podataka 33

5.2.1. Tipovi grafikona za vizualizaciju 34

6. ZAKLJUČAK 37

LITERATURA 38

PRILOZI 39

POPIS SLIKA

Slika 1. Rekonstrukcija HyperText prikaza 2
Slika 2. Prikaz rangiranosti Wikipedije po dnevnoj posjeti od 2002. godine 4
Slika 3. Sučelje Wikipedije 6
Slika 4. VisualEditor traka za uređivanje teksta 8
Slika 5. Wiki uređivač 8
Slika 6. Stranica povijesti članka 10
Slika 7. Prikaz slike unutar članka 11
Slika 8. Mali reproduktor za korištenje unutar teksta 11
Slika 9. Wiki reproduktor za zvuk 11
Slika 10. Korisnički zahtjev 12
Slika 11. Mogućnosti administriranja aplikacije 16
Slika 12. Prikaz konvencionalnog pristupa kreiranja pregleda 18
Slika 13. Način kreiranja pregleda s angularom 19
Slika 14. Primjer AngularJS aplikacije 23
Slika 15. Veza podataka kroz MVC strukturu 23
Slika 16. Ustroj wiki aplikacije 25
Slika 17. Prikaz naslova članka unutar kategorije Hrvatska 26
Slika 18. Glazbeni reproduktor 27
Slika 19. Prikaz članka 28
Slika 20. Obrasci za postavljanje i pregled privitaka 29
Slika 21. Meta stranica za statističke podatke 29
Slika 22. Pregled uvoza xls datoteke 30
Slika 23. Prazan radni list 32
Slika 24. Forma za kreiranje grafa 34
Slika 25. Stupčasti grafikon 34
Slika 26. Površinski grafikon 35
Slika 27. Linijski grafikon 35
Slika 28. Grafikon s mjehurićima 36
Slika 29. Okrugli grafikon 36
Slika 30. Raspršeni grafikon 36

POPIS TABLICA

Tablica 1. Primjer wiki markup sintakse 9
Tablica 2 Markmin sintaksa 27

SAŽETAK

U ovom radu proučit će se mogućnosti wiki sustava te će se razmotriti njihova upotreba prilikom statističke obrade i vizualizacije podataka. Na početku će se prikazati povijesni pregled razvoja wiki načina rada i ideje koje su dovele do sustava koji se danas koristi u jednoj od najvećih online enciklopedija. Koristeći Wikipediju bit će opisan ustroj wiki sustava kroz način njezine organizacije i administracije. Tako će se objasniti postupak uređivanja i kreiranja članaka, te njihova kategorizacija na stranicama Wikipedije. U drugom dijelu bit će prikazan kratak pregled alata koji će se koristiti prilikom razvoja wiki sustava korištenog u ovom radu. Tu će biti objašnjen Web2Py framework temeljen na Python programskom jeziku i AngularJS framework pisan JavaScript programskim jezikom. Završit će se sa pregledom načina rada takve wiki aplikacije unutar koje su integrirani obrasci za unos i vizualizaciju statističkih podataka.

Ključne riječi: Wikipedija, wiki, Web2Py, AngularJS, statistika, statistička obrada, vizualizacija podataka

SUMMARY

This paper studies the possibilities of wiki system and will consider their use in statistical analysis and data visualization. At the beginning of the paper we will present historical overview of the wiki development and the ides that led to system that is today used in of the largest online encyclopedia. On example of Wikipedia there will be described the structure of the wiki way for organization and administration. Here will be explained the process of creating and editing articles and their categorization in the pages of Wikipidea. The second part will present a brief overview of the tools that were used for developing a wiki system that is presented in this paper. There will be explained Web2Py framework based on python programming language and AngularJS framework written in JavaScript programming language. In the end will be presented usage of developed wiki application which will include forms for creating statistical data set and their visualization.

Key words: Wikipedija, wiki, Web2Py, AngularJS, statistics, statistical analysis, data visualization

.

1. POVIJEST

Od svojih je početaka wiki prošao dalek put. Od same ideje i jednostavnog razmišljanja Howarda Cunninghama pa do web stranice Wikipedija, koja je danas postala sinonim za enciklopediju. S preko 4 milijuna članaka Wikipedija je postala vodeći izvor znanja na internetu.

1.1. Temelji wiki filozofije

Nakon završetka školovanja, Howard Cunningham zaposlio se u Tek Labs, gdje je imao zadatak istražiti mogućnosti organiziranja njihovih softverskih projekata. Tako je započeo istraživati načine kako dokumentirati ideje i projekte kako bi zaposlenici unutar različitih odjela mogli dijeliti znanje. Prilikom proučavanja kako to postići, Cunningham je sustav zamislio kao metafore koje vode na pravo mjesto unutar sustava. Nakon dugo vremena Cunningham je napokon otkrio alat koji će mu pomoći realizirati njegovu ideju. Apple Computer je svojim novim softverom zvanim HyperCard omogućio brzo i jednostavno povezivanje sadržaja. Otkriće hyperlinka ili jednostavno linka, ljudima je bilo omogućeno klikom na ekranu doći do sljedećeg teksta ili multimedijskog sadržaja. Iako je HyperCard bilo dobro rješenje bilo je i nedostataka. Povezivanje kartica međusobno bilo je komplicirano. Povezivanje dviju kartica odvijalo se tako da se prvo moralo doći na karticu koju želimo povezati na prijašnju, a bez linka pristup toj kartici je otežan. Stoga je Cunningham doradio program tako da je stvaranje linka postalo jednostavno poput pisanja riječi. Pisanje riječi unutar liste jednostavno je označavalo novi link koji je odmah bio povezan s karticom istoga naziva. U slučaju da kartica ne postoji ona se mogla kreirati jednostavno držanjem pritisnutog linka. Tako je kreiranje novoga sadržaja postalo višestruko jednostavnije i zaposlenici su to prepoznali.

Slika 1. Rekonstrukcija HyperText prikaza

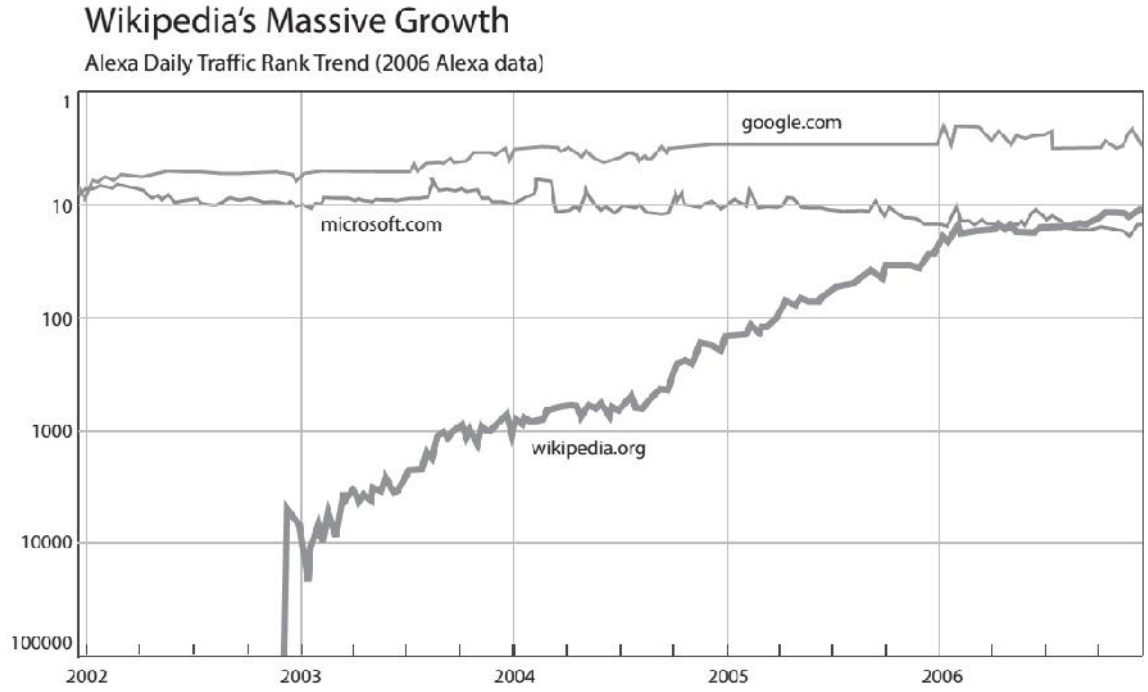
Radom na HyperCard sustavu Cunninghamu je postalo očito da je napravio brz i jednostavan način za organizaciju podataka. Do podataka se moglo doći preko više puteva povezanih kartica. Linkovi su postali moćan alat za organizaciju. Problem je bio u tome što su sva ta znanja, ideje i razmišljanja bila čvrsto vezana za to jedno računalo. Umrežavanje u to vrijeme nije bilo dostupno svima, a fizička sinkronizacija između računala bila bi prezahtjevna i nije omogućavala uređivanje u stvarnom vremenu. Unatoč potencijalu sustava sve je to bio još uvijek samo eksperiment na Cunninghamovu računalu. Pojavom interneta i njegove široke dostupnosti masama, Cunningham je počeo razmišljati o internet verziji HyperCard sustava. Ubrzo je kreirao stranicu gdje su korisnici mogli pretraživati i uređivati sadržaj. Preostalo je izabrati ime za takav sustav, a pronašao ga je prisjećajući se svog medenog mjeseca na

Havajima. Wiki dolazi od havajske riječi za brzo, stoga se Cunninghamu činilo kako dobro utjelovljuje ideju iza njegova sustava.

1.2. Početak Wikipedije

Prethodnik Wikipedije bila je Nupedija. Nupediju su pokrenuli Jimmy Wales i Larry Sanger s nastojanjem da kreiraju online enciklopediju s licencom besplatnog sadržaja, što bi značilo da bi ju korisnici nesmetano mogli koristiti kao izvor informacija i sadržaja. Od autora se očekivalo da budu stručnjaci u svojemu području, a svaki se je članak pregledavao prije objavljivanja. Zbog toga je zainteresiranost bila niska i zato se samo nekoliko članaka uspjelo dovršiti.

Wikipedija je pokrenuta 15. siječnja 2001. godine kao alternativa Nupediji temeljena na wiki sustavu. Započeta je da bi se lakše privukli autori i članci za Nupediju. Uspjeh je bio gotovo trenutačan. Omogućavanjem kreiranja sadržaja svima koji su to željeli sustav je brzo stekao popularnost. Kako su Slashdot i ostale tehničke zajednice pridavale važnost Wikipediji, tako su korisnici iz tehničkih područja brzo prihvaćali ideju iza Wikipedije. Članci su se gomilali i zajednica je rasla. Razvojem informacijskog doba Wikipedija je poslužila kao pravo mjesto za brz dolazak do informacija.



Slika 2. Prikaz rangiranosti Wikipedije po dnevnoj posjeti od 2002. godine

Popularnost Wikipedije naglo raste krajem 2002. godine.[Slika 2]

2. USTROJ WIKI SUSTAVA

2.1. Način rada

Wiki je kolaborativna baza podataka i zamišljena je kao najjednostavnija moguća baza podataka. Osnovni način funkcioniranja svakog wiki sustava je veoma jednostavan. Sastoji se od kreiranja i održavanja sadržaja. Uređivanje sadržaja obavljaju sami korisnici. Tako imamo više korisnika koji rade na jednom članku. Oni su zaduženi za ispravnost, količinu i kvalitetu sadržaja. U današnje je vrijeme takva praksa veoma zastupljena. Korisnici diljem svijeta imaju mogućnost međusobnom suradnjom doprinositi sadržaju, a samim time omogućuje se i brži rast baze podataka.

2.1.1. Organizacija

Kao što smo već spomenuli, na jednom članku može raditi više osoba i na taj način zajedno surađivati u uređivanju istoga sadržaja. To se može činiti kontraproduktivno, ali je u pozadini organizacija drugačija. Iako je dozvoljeno uređivanje svakom korisniku, wiki zajednica ima niz pravila, procedura i savjeta.

2.1.2. Namespaces

Namespaces bile bi meta stranice, što znači da se odnose na drugu wiki stranicu koja u ovom slučaju sadrži originalni članak. Prepoznat ćemo ih po prefiksu s nazivom kategorije namespace stranice iza koje slijedi dvotočka `[[Naziv_kategorije:Članak]]`. Svaki članak može imati više stranica vezanih za njega.

Najvažnija je od svih stranica "Razgovori". Na njoj korisnici diskutiraju o sadržaju i dolaze do zaključka kako najbolje urediti članak. Veoma je važna jer se na njoj korisnici organiziraju. Korištenje te stranice također je važno zbog toga što se sav dodatni sadržaj ne prikazuje unutar članka, već se prikazuje samo profiltrirani sadržaj.

2.2. Sučelje

Wikipedija je rađena u CMS sustavu zvanom MediaWiki, originalno razvijenim za Wikipediju. Koristi PHP jezik za procesiranje i prikaz podataka spremljenih u bazi podataka

poput MySQL. Razvojem MediaWiki funkcionalnost je proširena mnogo više od osnovnih elemenata wiki sustava. Kao primjer mogu se navesti namespaces stranice, povijest uređivanja, personalizacija stranice preko predložaka, višejezičnost itd.

Slika 3. Sučelje Wikipedije

Sučelje Wikipedije veoma je jednostavno. Unutar sučelja nalaze se sljedeći elementi:

- S lijeve strane nalazi se bočna traka koja ima funkciju navigacijskog panela.
- U traci pri vrhu nalaze se tabovi koji su izravno vezani s prikazanom stranicom.
- Lijevo u traci pri vrhu nalaze se tabovi „Članak“ i „Razgovor“. „Članak“ služi za prikaz uređenog članka, a „Razgovor“ za prikaz meta stranice na kojoj se vodi diskusija o uređivanju članka.
- Desno se nalaze tabovi koji su povezani s uređivanjem članka. Tab „Čitaj“ označava da se stranica može samo pregledavati, a „Uredi“ služi za uređivanje sadržaja. U ovom se slučaju uređivanje odvija pomoću modula VisualEditor. Potom se nalazi tab „Uredi kod“ koji služi također za uređivanje sadržaja, ali u ovom se slučaju koristi wiki markup sintaksa. Posljednji tab prikazuje povijest uređivanja članka.
- U krajnjem desnom položaju nalazi se tražilica
- Članak, u ovom slučaju s naslovom „Zagreb“, nalazi se ispod tabova.

2.3. Administracija

Osnovni element wikija je članak. Zbog načina na koji wiki funkcionira članci zapravo nikada nisu završeni i uvijek se postoji mogućnost njihovog uređivanja.

2.3.1. Pretraživanje

Pretraživanje na wiki stranici odvija se kroz integriranu tražilicu. Upisom traženog pojma dobivamo povratnu informaciju o kreiranim člancima. Pretraživanje se odvija po naslovu i po sadržaju članka.

Članci su također kategorizirani. Kategorije služe za jasno odjeljenje područja na koje se članci odnose. Neke od kategorija na hrvatskoj Wikipediji su povijest, zemlja, društvo, tehnologija, znanost, matematika, umjetnost itd.

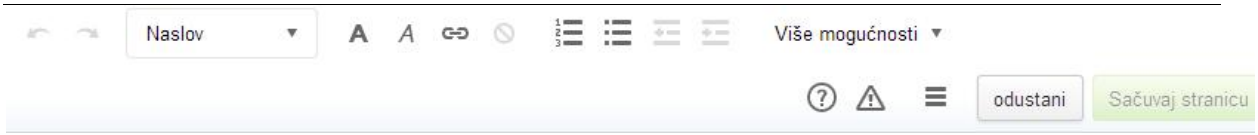
Osim što članci mogu biti kategorizirani, oni također mogu biti i povezani linkovima unutar samih članaka, odnosno, veze između članaka su pomoću ključnih riječi unutar članka. Takav oblik povezivanja naziva se hipertekst. Hipertekst je tekstualna struktura koja se sastoji od međusobno povezanih jedinica informacija, pa zato nema jedinstvenog redoslijeda čitanja, nego je sadržaj dinamički određivan od strane čitatelja.

2.3.2. Uređivanje članaka

Uređivanje se odvija klikom jednog od tabova: „Uredi“ ili „Uredi kod“. Tab „Uredi“ otvara VisualEditor, dok tab „Uredi kod“ otvara klasični wiki editor. U nastavku će biti objašnjen rad i nedostaci pojedinih načina uređivanja.

2.3.2.1. VisualEditor

Uređivanje pomoću VisualEditor je noviji način formatiranja sadržaja. Vrlo je jednostavan i intuitivan za korištenje. Omogućava uređivanje sadržaja bez poznavanja wiki markup sintakse. Pomoću njega sadržaj se uređuje u obliku članka kakav se prikazuje. Svaka je promjena odmah vidljiva. Pisanje i oblikovanje grafički je intuitivno poput pisanja u nekom od tekstualnih programa poput Microsoft Word.

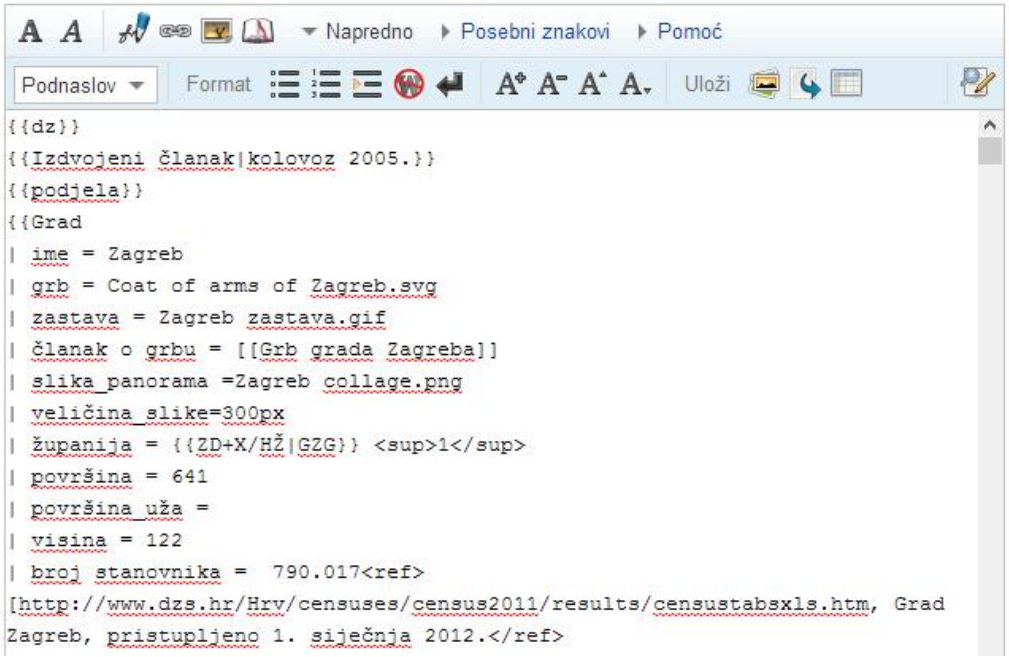


Slika 4. VisualEditor traka za uređivanje teksta

Razvijan je pomoću JavaScript-a, Node.js i PHP-a. Trenutno se nalazi u beta inačici i dostupan je samo kao dodatna opcija korisniku. Postoje neka ograničenja zbog kojih se još ne koristi ravnopravno s wiki markup sintaksom: sporije je od starijeg wiki uređivača; ograničena je podržanost unutar nekih web preglednika; nije moguće uređivanje pojedinih odjeljaka već samo cijele stranice; formatiranje teksta nije uvijek ispravno.

2.3.2.2. Wiki uređivač

Za uređivanje članaka, wiki se od početka oslanja na jednostavan pristup uređivanja. Sadržaj se uređuje u običnom polju za unos teksta. Svi web preglednici ga podržavaju, budući da je dio HTML standarda još od ranih početaka. Iako je jednostavan za implementaciju, njegovo je korištenje otežano zbog upotrebe markup jezika. Naravno, markup jezik ne koristi se kako bi krajnjem korisniku otežao postupak, već je njegova svrha zapravo da mu olakša prezentaciju sadržaja i osigura konzistentnost u toj prezentaciji. Mana takvog načina je što korisnik mora naučiti osnove markup jezika.



Slika 5. Wiki uređivač

Novije verzije wiki uređivača oslanjaju se na JavaScript jezik da bi i bez nekog znanja markup jezika korisniku omogućio uređivanja.

Tablica 1. Primjer wiki markup sintakse

Primjer wiki sintakse	Prikaz unutar preglednika
`''italic''`	*italic*
`'''bold'''`	**bold**
`'''''bold i italic'''''`	***bold i italic***
`<strike> precrtani tekst </strike>`	~~precrtani tekst~~
`<nowiki> bez ''wiki'' sintakse </nowiki>`	bez ”wiki” sintakse

2.3.3. Kreiranje članaka

Da bi se članak kreirao, prvo ne smije postojati članak s istim pojmom ili temom koju opisuje. Zbog toga bi svaki korisnik prvo trebao pretražiti ključnu riječ putem tražilice, a ako članak ne postoji, Wikipedija nudi mogućnost kreiranja članaka pod tim naslovom. Osim ciljanog kreiranja članaka, wiki nudi još jednu mogućnost. Unutar već uređenog članka netko je mogao staviti poveznicu na wiki stranicu koja ne postoji. Takve su poveznice kod Wikipedije obojene u crveno i označavaju želju korisnika da se kreira dodatni sadržaj na sljedećoj wiki stranici. Klikom na takav link wiki nas vodi na stranicu za uređivanje teksta, a u ovom je slučaju stranica prazna i potrebno ju je započeti.

Pojam na koji se želi kreirati poveznica stavlja se između dvostrukih uglatih zagrada. U slučaju tvorbe poveznice prema članku „Zagreb“, wiki sintaksa ima sljedeći oblik:

```
[[Zagreb]]
```

U slučaju da već postoji članak s nazivom „Zagreb“, poveznica će biti plave boje i vodit će na taj članak.

2.3.4. Praćenje izmjena

Sve se promjene prate i bilježe. Svaki članak ima svoju povijest uređivanja. Klikom na tab „Vidi stare izmjene“, koji se nalazi iznad svakog članka, vodi nas na stranicu koja prikazuje povijest svih izmjena.

Slika 6. Stranica povijesti članka

Na Slika 6 vide se dvije izmjene od nekoliko desetaka njih. Prati se vrijeme uređivanja, korisnik, promjena veličine članka i komentar koji je korisnik ostavio na uvid ostalima. Nakon uređivanja članka jednog korisnika – u ovom slučaju donjeg na prikazu, drugi je korisnik – onaj iznad njega, vratio članak na jednu od njegovih prethodnih inačica.

2.3.5. Multimedijski privici

Članci se mogu dodatno proširiti ubacivanjem slika, te video i audio zapisa. Prije ubacivanja nekih od njih oni se moraju prvo postaviti na Wikimedia Commons. Wikimedia Commons je online repozitorij koji pohranjuje sve datoteke koje mogu, a i ne moraju biti povezane s Wikipedijom.

Klikom na upload file s lijeve strane otvara se obrazac za postavljanje datoteke. Unutar obrasca treba označiti putanju datoteke na računalu, te ispuniti obrazac. Nisu podržani svi tipovi datoteke. Zato kod postavljanja treba obratiti pažnju na dozvoljene ekstenzije datoteka, inače postavljanje neće uspjeti.

Nakon uspješno postavljene datoteke slijedi postavljanje unutar članka. U daljnjem tekstu slijedi primjer markup sintakse za ubacivanje datoteka.

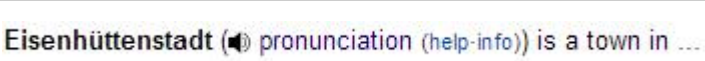
Ubacivanje videa i slike:

```
[[File:imeslike.jpg]]
```

Slika 7. Prikaz slike unutar članka

Ubacivanje zvuka:

```
{{listen |filename=zvuk.ogg | title=Naziv datoteke |description=Opis sadržaja}}
```



Slika 8. Mali reproduktor za korištenje unutar teksta

```
{{Audio|naziv zvuka|link do zvuka}}
```

Slika 9. Wiki reproduktor za zvuk

3. WEB2PY

Projekt Web2Py započet je 2007. godine od strane Massima Di Pierra. Započet je kao edukativni alat s ciljem web razvoja na brži, jednostavniji i sigurniji način. Razvijan je kao web framework otvorenog koda. Dobitnik je nagrade Bossie Award 2001. godine kao najbolji softver otvorenog koda, dok 2012. dobiva nagradu *Technology of the Year* InfoWorlda .

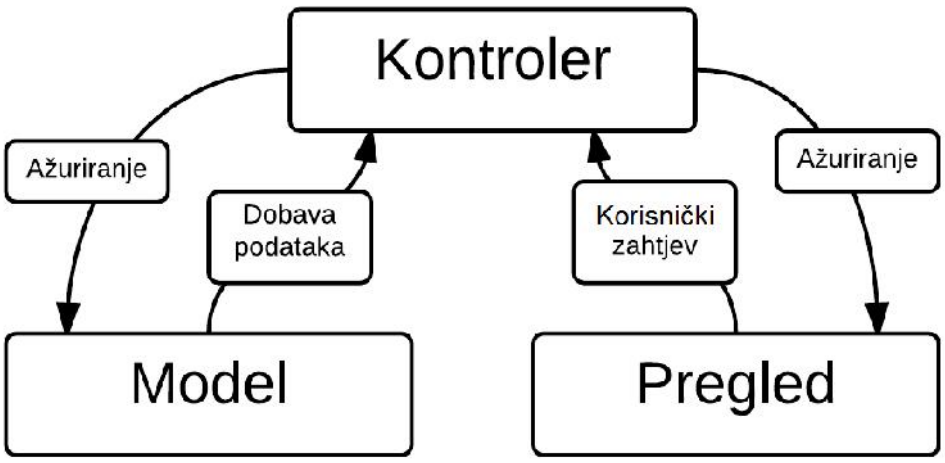
Model View Controler (MVC)

MVC arhitektura temelji se na principu odvajanja komponenti aplikacije ovisno o njihovoj namjeni. Takav dizajn olakšava razvoj kompleksnijih aplikacija i kasnije pridonosi otklanjanju neispravnosti.

MVC sastoji se od:

- **Modela** (*eng. model)* – podaci aplikacije, u ovom slučaju kao posrednik do baze podataka
- **Pregleda** *(eng. view)* **–** prikaz dobavljenih podataka
- **Kontrolera** *(eng. controller***) –** upravljanje podacima i interakcija s korisničkim zahtjevima

Interakcije između te tri komponente određuje rad aplikacije.



Slika 10. Korisnički zahtjev

3.1. Database Abstraction Layer (DAL)

Database Abstraction Layer, skraćeno DAL, niz je procedura koje olakšavaju pristup bazi podataka. Unutar Web2Py nalazi se dodatni sloj koji se brine za generiranje zahtjeva prema bazi podataka. S tim slojem dobivamo mogućnost pristupa bazama podataka od raznih proizvođača poput MySQL, PostgreSQL, SQLite, Oracle i DB2. Svaki od proizvođača ima svoju sintaksu za generiranje zahtjeva prema bazi, pa nam DAL omogućava da se jednom definirani zahtjev za podatkom lako prilagodi različitim bazama. Kao primjer usporedit će se kreiranje tablice unutar MySQL i PostgreSQL baze, te sintaksa koja se koristi u Web2Py.

3.1.1. Primjer korištenja baze

Želi se kreirati nova tablica s nazivom *korisnik*, koja će sadržavati polja *ime* i *prezime*. Uz ta željena polja, za rad baze podataka nužno je da se svako polje može jednoznačno identificirati. Zbog toga se koristi dodatno polje *id*. Svaki unos generira jedinstveni broj unutar polja *id* koji se unosi zajedno s poljima *ime* i *prezime*. U nastavku će se stoga usporediti razlika između definiranja jednog takvog polja.

Kreiranje tablice MySQL 5.0

Za definiciju polja *id* koriste se naredbe:

- INTEGER – definiranje polja kao integer, cijeli broj
- AUTO_INCREMENT – definiranje svakog sljedećeg unosa broja kao broja koji je za jedan veći od prošlog unosa
- PRIMARY KEY – definiranje polja kao jedinstvenog unutar tablice

```
CREATE TABLE korisnik (
  id INTEGER AUTO_INCREMENT PRIMARY KEY,
ime char(50),
prezime char(50),
)
```

Kreiranje tablice PostgreSQL 9.1

Za definiciju polja *id* koriste se naredbe:

- SERIAL – definira polje kao integer i osigurava da je sa svakim unosom taj broj za jedan veći od prošlog unosa
- PRIMARY KEY – definiranje polja kao jedinstvenog unutar tablice

```
CREATE TABLE tablename (
  tablename_id SERIAL PRIMARY KEY,
 ime char(50),
prezime char(50),
)
```

Kreiranje tablice Web2Py

Unutar Web2Py nije potrebno definirati polje jedinstvenog identifikacijskog broja. Web2Py se brine o tome da svaka tablica ima polje *id* sa svojstvom auto-inkrementa.

Za korištenje neke od navedenih baza potrebno je uključiti definicije željene baze podataka. DAL objekt predstavlja vezu s tom bazom.

```
db = DAL('sqlite://storage.sqlite')
```

```
db.define_table('korisnik', Field('ime'), Field('prezime'))
```

U ovom slučaju u pozadini se generira kod za SQLite bazu podataka, ali s uključivanjem definicija drugih baza generirani se kod prilagođava toj bazi. Stoga imamo mogućnosti lakog mijenjanja baza ili korištenja više baza.

Neke od podržanih baza podataka za Web2Py su SQLite, PostgreSQL, MySQL, Oracle itd.

3.2. Markmin

Markmin je markup jezik koji dolazi s Web2Py u obliku *helpera*. Njegova sintaksa je dizajnirana za pisanje znanstvenih radova i knjiga. Koristeći markmin *helper* dobivaju se mogućnosti korištenja wiki sintakse i pretvaranje teksta u HTML kod prema markmin pravilima. Također postoji mogućnost pretvaranja teksta u latex ili pdf format.

Posebnost korištenja markmin sintakse leži u tome što se dobiva mogućnost kreiranja nove sintakse, koja će biti prepoznata unutar teksta i shodno njezinoj definiciji urediti tekst.

3.3. Web2Py wiki

Web2Py dolazi s već implementiranim wiki mogućnostima, pa će se zato objasniti upotreba tih mogućnosti.

Za aktivaciju Web2Py wikija, unutar kontrolera *default.py* uredi se `index()` akcija tako da odgovara kodu:

```
def index(): return auth.wiki()
```

S tom jednom linijom koda dobiven je osnovni kostur wiki stranice. Web2Py se pobrinuo za izradu tablica u bazi podataka, forme za kreiranje članaka, njihovo uređivanje, dozvole korisnika itd.

3.4. Administratorsko sučelje

Unutar administratorskog sučelja dobivamo pregled svih instaliranih aplikacija i dobivamo mogućnost njihovog uređivanja.

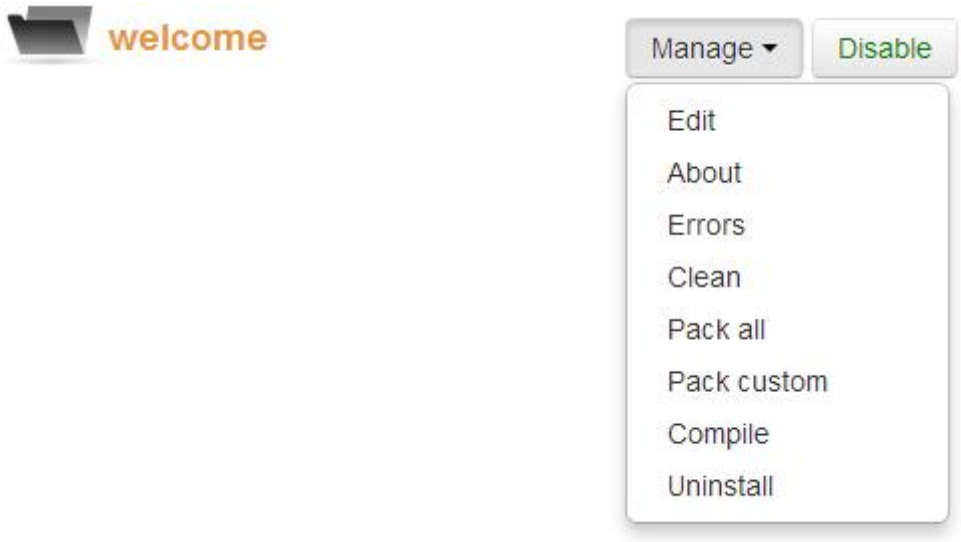
Web2Py dolazi s tri predinstalirane aplikacije:

- **admin** – aplikacija administratorskog sučelja
- **examples** – replika Web2Py internetske stranice i dokumentacija
- **welcome** – aplikacija koja je prikazana pri prvom pokretanju Web2Py, služi kao predložak za lakši početak

Kao administrator postoje sljedeće mogućnosti:

- **Instalacija** aplikacije ispunjavanjem forme *Upload and install packed application* s desne strane.
- **Deinstalacija** aplikacije, s čime ju potpuno brišemo.
- **Onemogućavanje** pregledavanja aplikacije.
- **Čišćenje** privremenih datoteka aplikacije, poput zapisa sesije korisnika, zapis grešaka i brisanje cache memorije.

- **Pakiranje** aplikacije za distribuciju. Dobivamo aplikaciju u *tar* datoteci koja sadrži sve podatke o aplikaciji i sadržaj baze podataka. Služi za instalaciju aplikacije unutar administratorskog sučelja.
- **Kompajliranje** kako bi se distribuirala bez izvornog koda.
- **Uređivanje** svih datoteka aplikacije.



Slika 11. Mogućnosti administriranja aplikacije

4. ANGULARJS

Korištenje web stranica svodi se na slanje zahtjeva prema serveru i dobave odgovora na taj zahtjev. Tako za svaku promjenu prikaza trebamo proći taj proces koji se u cijelosti provodi na strani servera. Prilikom primanja takvog zahtjeva server počinje s obradom zahtjeva na način da vidi koja se informacija očekuje kao odgovor. Nakon toga obradom podataka, dobavljenih od strane korisnika ili baze, dolazi do konačnog rješenja koje se prezentira davaocu zahtjeva kao odgovor. Takav je proces obrade diskretiziran u vremenu i kao takav potražuje mnogo zahtjeva prema serveru. Za svaku promjenu podatka unutar baze ili unutar sučelja korisnika potrebno je kreirati novu HTML stranicu koja unutar sebe sadrži željeni podatak. Zbog toga imamo mnogo redundantnih dobava stranica koje su mogle biti samo uređene na strani korisnika. Korištenje takvog sustava bi za korisnika bilo intuitivno i ne bi iziskivalo vrijeme čekanja učitavanja stranice.

Aplikacija koja će se opisati u ovom radu bit će u mogućnosti kreirati statističke podatke i prikazati ih. Statistički podaci obično dolaze kao velike baze podložne promjenama i redovitim novim unosima. Stoga će se za ovu aplikaciju ugraditi mogućnost dinamičkog rada na tim podacima, a sve u svrhu lakšeg korištenja aplikacije prilikom manipulacije podacima.

Za izradu takve mogućnosti potrebno je koristiti neki od programskih jezika koji će se odvijati na strani klijenta. U ovom slučaju koristit će se JavaScript. JavaScript izvodi se u pregledniku korisnika i potpuno je neovisan od servera, a također je u mogućnosti slati zahtjeve prema serveru koji se u potpunosti odvijaju u pozadini bez znanja korisnika. S takvom mogućnošću ne prekidamo korisnika pri upotrebi aplikacije, a opet imamo komunikaciju prema serveru.

JavaScript kao skriptni jezik omogućava nam mogućnost izvođenja programa na strani klijenta i može se koristiti u njegovom izvornom obliku, no to bi nepotrebno otežalo razvoj i preglednost aplikacije sa stajališta njezine izrade. Koristit će se neki od dostupnih zbirki funkcija ili frameworka koji će izradu aplikacije znatno olakšati. U ovom će se slučaju koristiti JavaScript framework zvan AngularJS.

Angular je strukturni framework za izradu dinamičnih web aplikacija. Pisan je pomoću JavaScript-a i kao takav potpuno se izvodi na klijentskoj strani. Dizajniran je s idejom CRUD

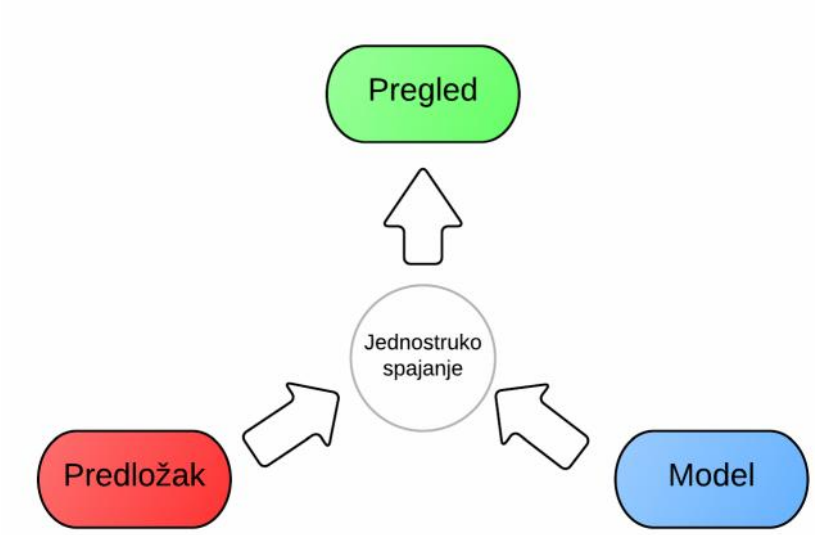
(*eng. Create Read Update and Delete*), što bi predstavljalo jednostavne aplikacije dizajnirane za izradu, čitanje, ažuriranje i brisanje podataka, bez obrade podataka koja bi se odvijala na strani klijenta.

Kako je HTML dizajniran za izradu statičnih stranica, pomoću njega je kreiranje dinamičkih aplikacija nemoguće. Koristeći AngularJS možemo proširiti funkcionalnosti HTML tagova na takav način da ih možemo dinamički obrađivati i dodavati nove mogućnosti koje inače nemamo koristeći HTML.

Neke od značajki AngularJS su tzv. povezivanje podataka, vlastiti sustav za kreiranje i upravljanje komponentama, podrška za upravljanje i validaciju formi, korištenje direktiva itd.

4.1. Povezivanje podataka

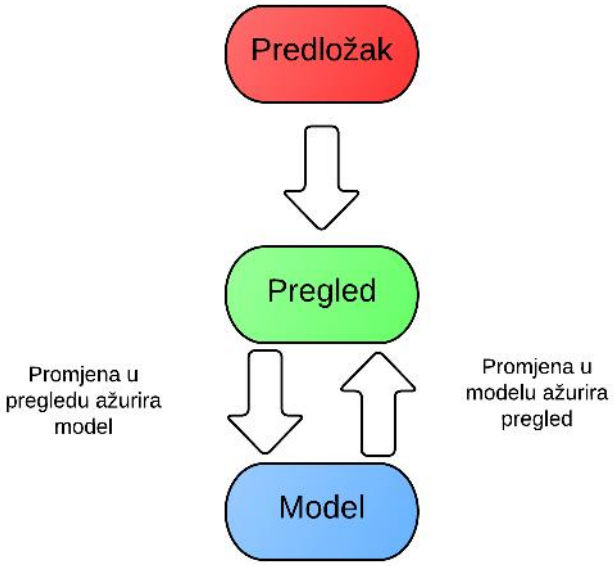
Klasični pristup prikaza podataka sastoji se od spajanja predloška s podacima i prezentiranju unutar preglednika. Nakon toga se promjene unutar modela ne reflektiraju izravno korisniku, niti se unosi korisnika ne reflektiraju kao promjene unutar modela. Za sinkronizaciju je potrebno pisati kod koji je nužan kako bi se moglo upravljati podacima.



Slika 12. Prikaz konvencionalnog pristupa kreiranja pregleda

Unutar aplikacija rađenih s AngularJS prvo se kompajlira HTML kod sa svim dodatnim oznakama i direktivama. Tako kompajlirani HTML prezentira korisniku dinamični pregled

koji se prilikom svake promjene unutar modela ažurira. Također se svaka promjena unutar pregleda reflektira kao promjena unutar modela. Takva instantna projekcija modela olakšava pisanje aplikacija i povećava njihovu preglednost.



Slika 13. Način kreiranja pregleda s angularom

4.2. Koncept rada

Rad AngularJS slijedi MVC princip i tako odjeljuje spremanje podatka od obrade i njegovog prikazivanja.

Tako postoji pregled podatka koji se odvija pomoću HTML i angular direktiva.

```
<div ng-app ng-init="qty=1;cost=2">
  <div>
    Količina: <input type="number" ng-model="qty">
  </div>
  <div>
    Cijena: <input type="number" ng-model="cost">
  </div>
```

```
  <div>
    <b>Ukupno:</b> {{qty * cost | currency:'HRK'}}
  </div>
</div>
```

Ovdje je prikazan angular predložak koji će se kompajlirati prilikom pokretanja. Tako transformiran predložak zove se pogled (*eng. view).* Unutar njega će se korisniku prikazivati podaci koji su spremljeni kao varijable unutar modela.

U ovom primjeru koriste se nove opcije označavanja unutar HTML-a koje nazivamo direktive. Direktive dodaju novo značenje atributima ili elementima. Tako imamo atribut `ng-app` koji pokreće direktivu za inicijalizaciju aplikacije. Također imamo i definiranu novu direktivu `ng-model` za element `input` koja dodaje novo ponašanje elementu. Direktiva `ng-model` sprema promjenu varijable ili ažurira vrijednost polja `input` ovisno o promjeni varijable.

Drugi je oblik označavanja korištenje vitičastih zagrada `{{ izraz | filter }}`. Nakon što kompajler susretne takav oblik označavanja zamijenit će ga s novom vrijednošću dobivenu obradom zahtjeva unutar vitičastih zagrada. Na mjestu `izraz` mogu postojati kratki JavaScript izrazi koji nam omogućuju čitanje i pisanje varijabli. Takve varijable nisu globalne varijable. Unutar JavaScripta-a varijable, funkcije i objekti se nalaze unutar radnog područja koje se još naziva *scope*. Angular nam omogućava da *scope* povežemo s izrazom unutar vitičastih zagrada. Radno područje *scope* predstavlja model unutar MVC arhitekture. U primjeru iznad također je sadržano polje `filter`. Filterom se formatira prikaz izraza unutar sučelja web aplikacije. U ovom slučaju koristi se filter *currency* kako bi rješenje prikazao kao valutu.

4.2.1. Kontroler

Za dodavanje mogućnosti obrade podataka koristi se kontroler u kojem se definira logika aplikacije. Instanca kontrolera kreira se pomoću konstruktora koji će se prikazati u nastavku.

Kako bi se kontroler uspješno dodao potrebno je nekom od elemenata unutar pregleda dodati direktivu `ng-controller`. Ova direktiva dodjeljuje instancu kontrolera elementu i svim njegovim podelementima. `InvoiceController as invoice` sprema kontroler kao

varijablu koja se kasnije može pozivati kada se referira na radno područje s varijablama kojemu taj kontroler ima pristup. Tako unutar instance kontrolera postoje reference prema varijablama `qty`, `cost`, `inCurr`, `currencies` i funkcijama `pay()` i `total(c)` koje `se` mogu pozivati prefiksom `invoice`.

```
<div ng-app="invoice1" ng-controller="InvoiceController as invoice">
  <b>Faktura:</b>
  <div>
    Količina: <input type="number" ng-model="invoice.qty" required >
  </div>
  <div>
    Cijena: <input type="number" ng-model="invoice.cost" required >
    <select ng-model="invoice.inCurr">
      <option ng-repeat="c in invoice.currencies">{{c}}</option>
    </select>
  </div>
  <div>
    <b>Ukupno:</b>
    <span ng-repeat="c in invoice.currencies">
      {{invoice.total(c) | currency:c}}
    </span>
    <button class="btn" ng-click="invoice.pay()">Plati</button>
  </div>
</div>
```

Funkcija `invoice.total(c)` prima jedan argument koji se dobiva prolaskom kroz listu definiranih valuta. Također će se to polje automatski ažurirati prilikom promjene bilo koje od varijabla koje se koriste prilikom proračunavanja.

Valute koje će biti vidljive kao opcija unutar aplikacije, spremljene su kao lista i njih ugrađujemo unutar pregleda pomoću direktive `ng-repeat`. Tako će se za svaku vrijednost unutar liste kreirati po jedan `<span></span>` element koji će sadržavati rješenje dobiveno pozivom funkcije `total(c)`, gdje je `c` vrijednost liste dobivena u n-tom koraku na n-tom mjestu liste.

Kontroler kreiramo sljedećim izrazom:

```
angular.module('faktura', []).controller('InvoiceController', function() {});
```

U ovom izrazu kreiran je kontroler `InvoiceController` koji je dodijeljen modulu `faktura`.

Potrebno je unutar kontrolera još definirati sve varijable i funkcije koje će se koristiti ili pri prikazu podataka unutar pregleda ili interno za obradu podataka.

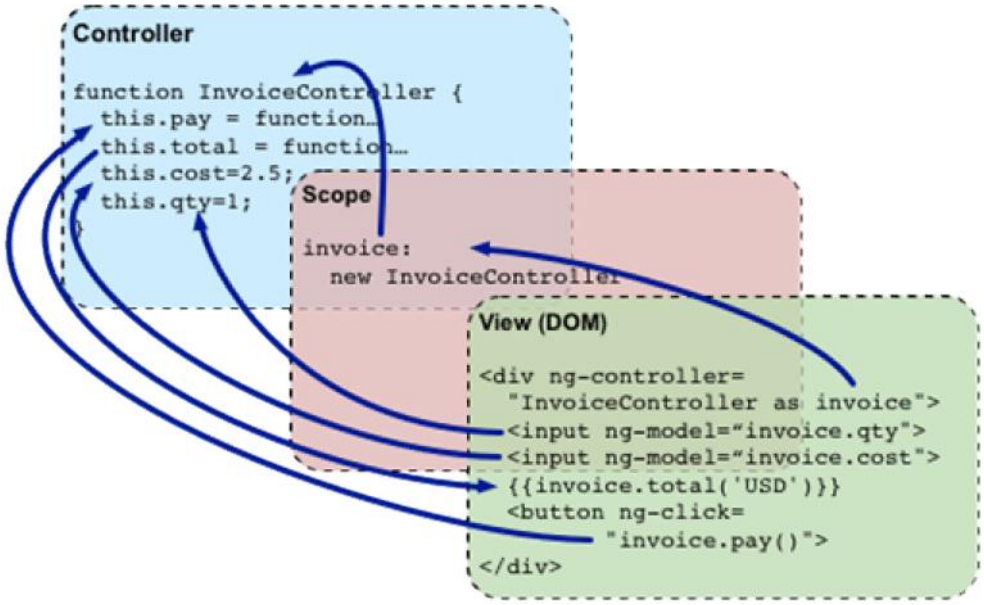
```
angular.module('faktura', [])
  .controller('InvoiceController', function() {
    this.qty = 1;
    this.cost = 2;
    this.inCurr = 'HRK';
    this.currencies = ['HRK', 'EUR', 'USD'];
    this.usdToForeignRates = {
      HRK: 1,
      EUR: 0.13,
      USD: 0.17
    };

    this.total = function total(outCurr) {
      return this.convertCurrency(this.qty * this.cost, this.inCurr, outCurr);
    };
    this.convertCurrency = function convertCurrency(amount, inCurr, outCurr) {
      return amount * this.usdToForeignRates[outCurr] / this.usdToForeignRates[inCurr];
    };
    this.pay = function pay() {
      window.alert("Hvala!");
    };
  });
```

Sada aplikacija poprima sljedeći izgled u kojem se prezentiraju polja za unos količine i cijene, odabir valute, pregleda cijena i gumba za plaćanje. Gumb za plaćanje povezali smo s

funkcijom kontrolera preko direktive `ngClick` koja klikom na gumb, u ovom slučaju, pokreće funkciju `pay()` koja se nalazi unutar kontrolera.

Slika 14. Primjer AngularJS aplikacije



Slika 15. Veza podataka kroz MVC strukturu

4.3. Korištenje unutar Web2Py okruženja

Web2Py i AngularJS koriste istu sintaksu unutar predložaka kako bi odijelio svoj kod od HTML markup jezika. U oba se slučaju koriste dvostruke vitičaste zagrade unutar kojih se definira kod za izvođenje. Zbog toga se konfigurira AngularJS na način da koristi drugačiju sintaksu. U sljedećem se primjeru koristi sintaksa s vitičastom, zatim uglatom i na kraju opet vitičastom zagradom.

```
tables.config(function($interpolateProvider, $locationProvider) {
    $interpolateProvider.startSymbol('{[{');
    $interpolateProvider.endSymbol('}]}');
});
```

5. Wiki aplikacija rađena s Web2Py

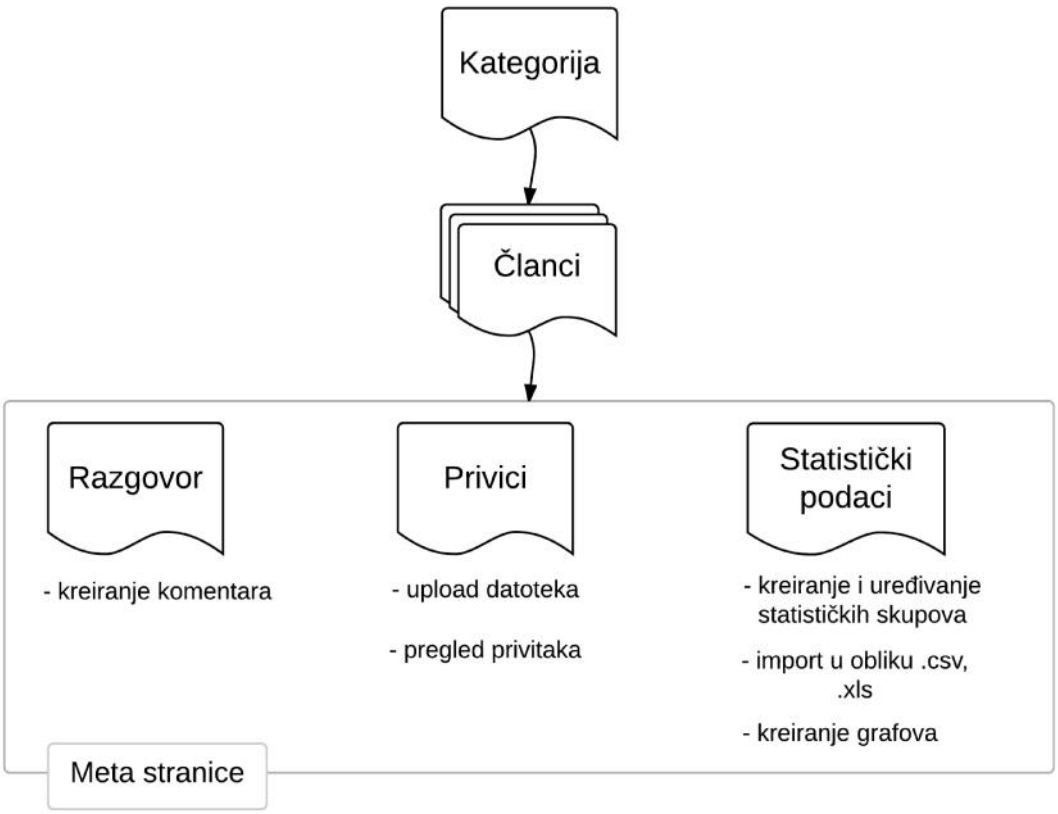
Osnova wiki stranice je članak. Stoga je za izradu wiki aplikacije potrebno napraviti sve funkcije koje su nužne za kreiranje članaka, njihovo moderiranje i kategorizaciju. Uz članke obično postoje i dodatni sadržaji poput slika, dokumenata, glazbe itd. Iz tog proizlazi da su potrebne forme za upload tog sadržaja i stranice za njihov prikaz.

Članci će se grupirati unutar kategorija koje će biti kreirane od strane administratora. Formatiranje članaka odvijat će se s markmin sintaksom uz neke dodatne elemente.

Kategorije će biti predefinirane. Unutar kategorija korisnik će imati ovlasti za kreiranje novih članaka, uređivanje postojećih i dodavanje sadržaja na meta stranice.

Meta stranice koje će biti povezane s člankom će biti:

- **Privici** – dodavanje dokumenata i ostalih datoteka koje su vezane uz članak
- **Razgovor** – dio predviđen za raspravu vezanu uz članak
- **Statistički podaci** – dodavanje i uređivanje statističkih podataka koji su vezani uz tematiku članka



Slika 16. Ustroj wiki aplikacije

Za kontrolu sadržaja bit će ugrađene dozvole pojedine akcije koje će se odvijati unutar aplikacije. Zato će se korisnici dijeliti na tri grupe:

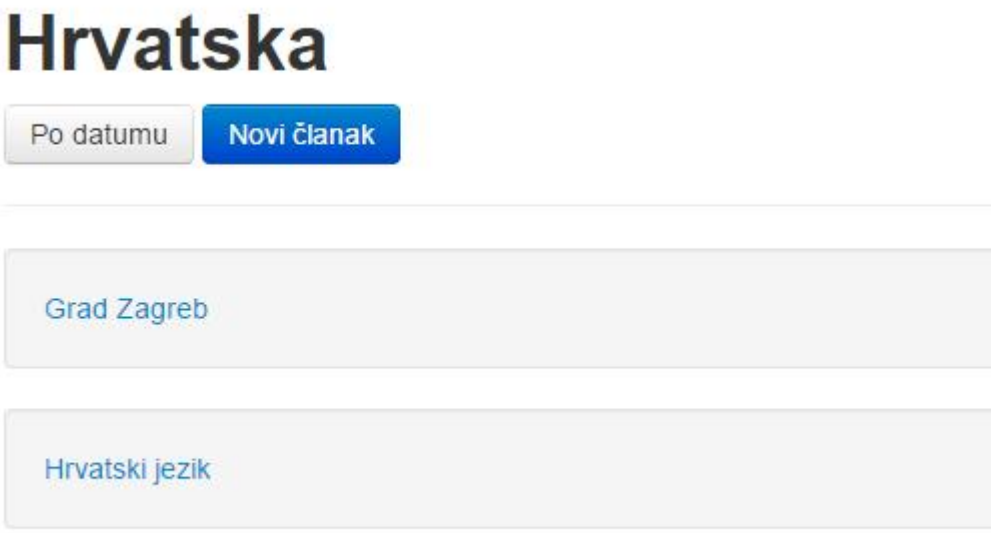
- **administrator** – ima najviše ovlasti, može uređivati i brisati sav sadržaj, uključujući i pojedine korisnike
- **urednik** – ovlasti urednika samo su nad sadržajem kojeg su ostali korisnici kreirali, može uređivati članke i njihove privitke
- **korisnik** – korisnik ima dozvolu kreirati sadržaj i uređivati sadržaj koji je kreirao

Sučelje wiki aplikacije pri vrhu sadrži traku s brzim poveznicama. Tako se unutar trake s lijeve strane nalazi poveznica na početnu stranicu i padajući meni s kategorijama, dok se s desne strane nalazi padajući meni za autorizaciju korisnika. Meni za autorizaciju korisnika sadrži poveznice na stranice za logiranje već postojećeg korisnika, registraciju novog korisnika i opciju traženja zaboravljene nove lozinke za već postojećeg korisnika.

Korisnik koji spada u grupu administratora, unutar trake s lijeve strane sadrži poveznicu za administratorsko sučelje. Unutar tog administratorskog sučelja nalaze se opcije za uređivanje

kategorija, uređivanje korisnika i uređivanje članaka. Na taj način administrator ima potpunu kontrolu nad sadržajem.

Kreirane se kategorije prikazuju kao poveznice na početnoj stranici wiki aplikacije. Klikom na neku od kategorija dobiva se prikaz svih članaka koji se nalaze u toj kategoriji. Članke je moguće sortirati po naslovu ili kronološki. U slučaju da je korisnik logiran, on ima mogućnost kreiranja novog članka pritiskom na tipku „Novi članak“, nakon čega se prikazuje forma za upis naslova i uređivanje članka.



Slika 17. Prikaz naslova članka unutar kategorije Hrvatska

Otvaranjem članka dobiva se prikaz članka s poveznicama za njegove meta stranice i za mogućnost uređivanja. Uređivanje već postojećeg članka moguće je za korisnike koji su kreirali članak ili su u jednoj od grupa administratora ili urednika.

Svaki od članaka ima mogućnost korištenja markmin sintakse za uređivanje izgleda članka, kreiranje poveznica, integriranje slika i slično.

Tablica 2 Markmin sintaksa

Markmin sintaksa	Prikaz u pregledniku
`# naslov`	**naslov**
`## odjeljak`	**odjeljak**
`### pododjeljak`	**pododjeljak**
`**bold**`	**bold**
`''italic''`	*italic*
`~~strikeout~~`	~~strikeout~~
``` ``boja s **bold**``:red ```	boja s **bold**
``` ``boja pozadine``:color[blue:#ffff00] ```	boja pozadine

Za kreiranje poveznica unutar članka koristi se sintaksa s dvije uglate zagrade `[[naziv poveznica]]`. Poveznica može biti vanjska ili između članaka. U slučaju da se koristi poveznica između članaka, dovoljno je koristiti skraćenu poveznicu u obliku `@////id` gdje je *id* jedinstveni identifikator za svaki članak. Također je moguće koristiti i naziv članka u obliku `@////naziv`. U slučaju da se koristi naziv članka potrebno je naziv formatirati na način da se velika slova zamijene malima, razmaci se zamijene povlakama, izbace se svi znakovi poput zagrada i slično te se dijakritička slova zamijene njihovim oblikom bez dijakritičkog znaka. Kao primjer može se navesti naziv članka *Karlovačka županija* koji bi se mogao povezati pomoću `[[Karlovačka županija @////karlovacka-zupanija]]`.

Unutar članka također je moguće postaviti reference unutar članka na način da se kreira *anchor* `[[naziv]]` i poveznica `[[veza na drugi dio članka #naziv]]`.

Datoteke koje sadrže ekstenzije mp3, mp4 ili neku od ekstenzija za slike bit će prepoznate i integrirat će se ovisno o tipu. Stoga je dovoljno na željeno mjesto staviti poveznicu do mp3 datoteke i na tom će se mjestu kreirati glazbeni reproduktor.[Slika 18]

Slika 18. Glazbeni reproduktor

Za integriranje slika moguća je sintaksa koja će odrediti alternativni naziv slike, širinu i poravnanje slike. Takva sintaksa ima oblik `[[alternativni naziv poveznica`

`poravnanje širina]]`, gdje poravnanje može biti *right* i *left* dok se širina definira u pikselima.

Dokumenti koji imaju neku od ekstenzija koje su podržane od strane *Google Doc Viewer* bit će integrirani unutar članka na način da je moguć njihov pregled. Dovoljno je na mjesto gdje se želi postaviti preglednik staviti izravnu poveznicu do dokumenta, bez markmin sintakse.

Formule je moguće integrirati s LaTeX sintaksom *$$formula$$*.

Kreiranje QR koda moguće je pomoću sintakse `qr:poveznica`, nakon čega će se poveznica prikazati kao QR kod.

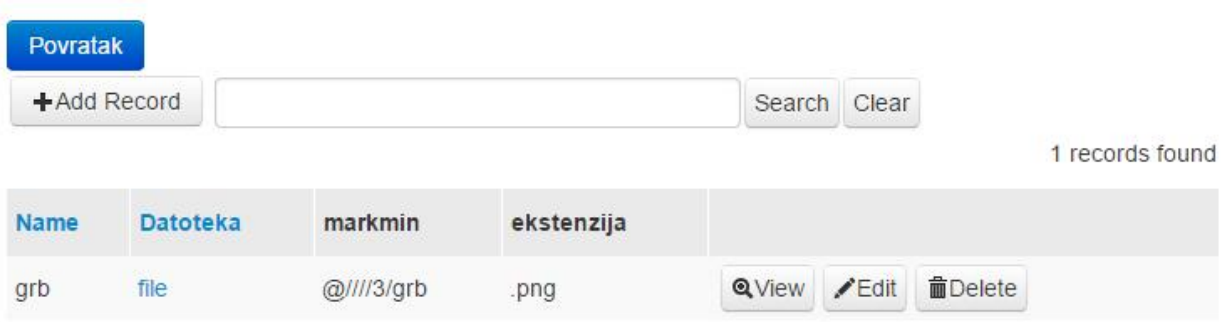
Meta stranice koje su vezane za svaki od članaka su „Razgovor“, „Privici“ i „Statistički podaci“.

Slika 19. Prikaz članka

Stranica razgovora predviđena je za komentiranje i raspravu vezanih uz članak i njegovo uređenje. Registrirani korisnici imaju mogućnost postavljanja kratkih poruka koje će biti vidljive na toj stranici.

Na stranici privitaka omogućeno je postavljanje datoteka poput slika, zvuka, tekstualnih datoteka itd. Na toj se stranici također prikazuju datoteke koje su prethodno postavljene. Lista tih datoteka prikazana je na način da se prikazuje naziv unosa, izravna poveznica za skidanje, markmin sintaksa potrebna za kreiranje linka unutar članka, ekstenzija datoteke te opcije za

njezino pregledavanje, uređivanje i brisanje. Ekstenzija datoteke prikazuje se zbog nadopune markmin sintakse. U slučaju da se markmin sintaksa za kreiranje linka produlji s ekstenzijom onda će Web2Py ugraditi datoteke ovisno o njezinoj ekstenziji. Na taj se način lako mogu unutar članka ugraditi slike ili samo poveznice za skidanje nekih od privitaka.

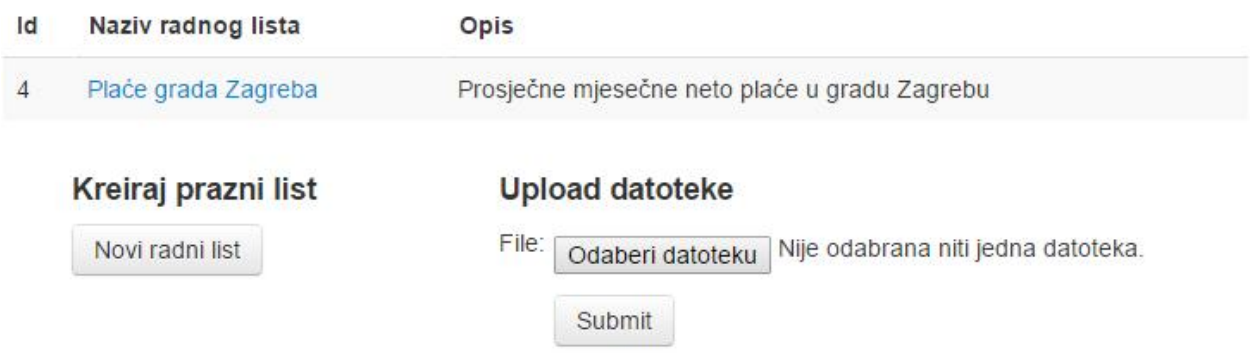


Slika 20. Obrasci za postavljanje i pregled privitaka

Postavljanje novog privitka pokreće se pritiskom tipke „Add Record“, nakon čega se dobiva forma za odabir datoteke i odabira njezinog imena. Budući da se koristi markmin sintaksa za povezivanje privitaka pomoću njihovog naziva, važno je datoteku imenovati na način da ne sadrži razmak ili dijakritičke znakove.

Tipka „Povratak“ služi na vraćanje na pregled članka.

Stranica sa statističkim podacima prikazuje listu radnih listova čiji su statistički podaci vezani uz članak. Svaki od radnih listova ima svoj naziv i opis podataka koje sadrži. Naziv radnog lista ujedno je i poveznica do njegovog prikaza. Kreiranje novog praznog radnog lista moguće je klikom na tipku „Novi radni list“, dok se za uvoz i kreiranje novog radnog lista koristi forma „Upload datoteke“.



Slika 21. Meta stranica za statističke podatke

Uvoz radnog lista moguć je za datoteke sa csv i xls ekstenzijom. Prilikom odabira datoteke forma će prepoznati tip datoteke i shodno tome pokrenuti odgovarajuće akcije. U slučaju da se koristi nepodržana datoteka javit će se poruka koja informira korisnika o nepodržanosti datoteke. Ako je datoteka podržana sljedeći je korak kratki pregled datoteke unutar preglednika. Prilikom pregleda radnog lista koji se uvozi korisnik dobiva uvid u sadržaj datoteke, odabir naslova i opisa, a u slučaju da se uvozi xls datoteka s više radnih listova, daje se mogućnost odabira koji će se radni listovi spremiti u bazu podataka.

Slika 22. Pregled uvoza xls datoteke

Nakon odabira naslova i opisa potrebno je pokrenuti spremanje pritiskom na tipku „Spremi“. Spremanje se pokreće u pozadini i to na način da se prvo kreira radni list koji se potom

popunjava s podacima iz datoteke. U slučaju da se uvoze velike količine podataka, spremanje može potrajati neko vrijeme.

5.1. Obrasci za unos, provjeru i tvorbu podataka programskim putem

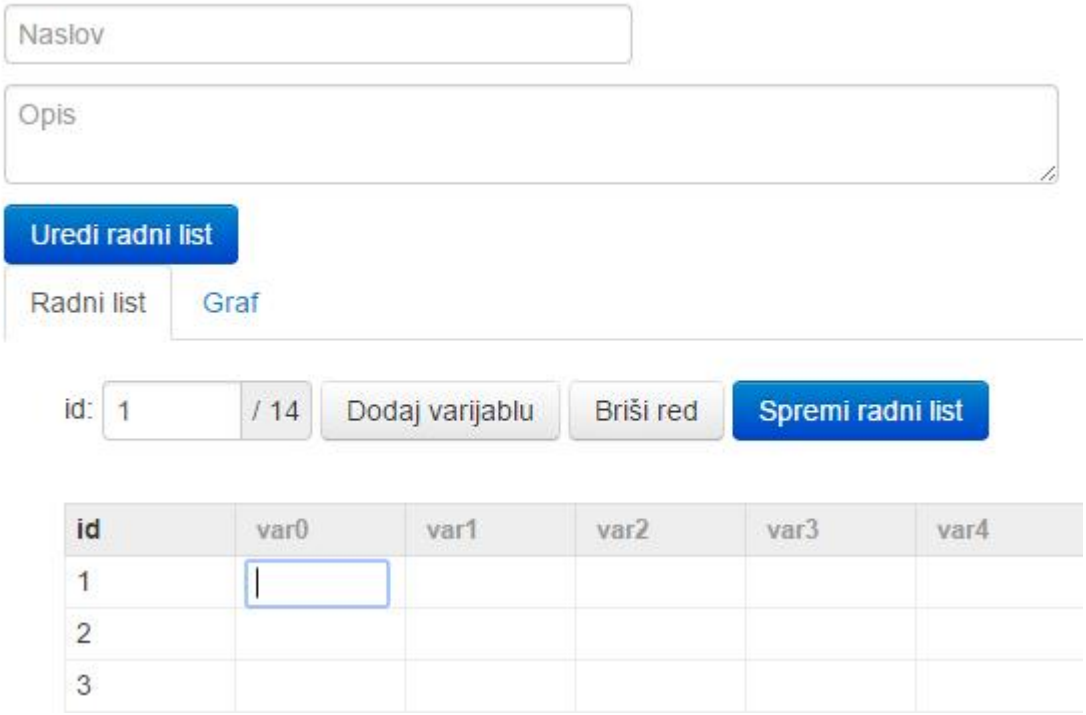
Kako bi se radni listovi mogli popunjavati podacima i uređivati, potrebno je kreirati obrasce za unos i akcije za manipulaciju podataka.

Podaci za statističku obradu obično se skupljaju u obliku proračunskih tablica sličnih onima unutar Microsoft Excel okruženja. Najzastupljeniji je format takav da redovi predstavljaju jedan unos subjekta, dok su stupci varijable vezane za taj subjekt. Takav format za unos podataka upotrijebit će se i u ovom radu kako bi se unutar wiki sučelja mogli kreirati novi setovi podataka.

Za prikaz, uređivanje i kreiranje podataka koristit će se AngularJS koji će kreirati tablicu unutar web preglednika. Tablica će služiti za definiranje varijabli i unos novih setova podataka. Korištenjem programskog jezika, koji će se odvijati na lokalnoj razini, korisniku će se olakšati preglednost i omogućit će mu se jednostavna kontrola nad podacima.

Kako bi se podaci spremili unutar baze koristit će se Web2Py. Veza između AngularJS i Web2Py bit će asinkronim komuniciranjem između preglednika i servera. Slanjem podataka ili naredbi u JSON formatu, Web2Py će moći uz minimalnu komunikaciju biti u stanju održavati stanje baze podataka onakvim kakav korisnik ima unutar preglednika.

Ćelije radnog lista sadržavat će pojedini podatak koji će u radnom listu biti definiran svojom adresom.



Slika 23. Prazan radni list

Kako bi se uspješno spremio radni list potrebno je definirati naslov radnog lista i kratak opis podataka koje sadrži ili će sadržavati. Ova dva polja smatraju se dovoljnim za kreiranje novog radnog lista koji će se moći popunjavati i uređivati.

Prvi je stupac proračunske tablice *id* i služi kao identifikator za svaki od unosa. Njegovo uređivanje nije moguće. Svaki od ostalih stupaca predstavlja jednu varijablu za svaki uneseni subjekt. Naziv varijable nalazi se unutar zaglavlja proračunske tablice i potrebno ga je eksplicitno definirati. Ukoliko se varijabla ne definira spremanje radnog lista i dalje će biti moguće, no za svaku daljnju obradu podataka u smislu kreiranja grafa i slično bit će potrebno imati definirane sve varijable koje bi se mogle koristiti. Kako je vidljivo sa Slika 23., unutar praznog radnog lista postoje identifikatori varijabli u smislu „var“ plus redni broj stupca. Ti identifikatori nisu definirane varijable, već se koriste za lakše snalaženje unutar proračunske tablice dok se ne definiraju varijable.

Smjernice za definiranje naziva varijable koje je potrebno slijediti jesu da se ne smiju koristiti razmaci ili dijakritički znakovi, te je poželjan opisni *camelCase* format.

Tablica je dizajnirana na način da prikazuje samo dio sveukupnih redova. Ograničenje broja prikazanih redova implementirano je kako bi se poboljšao odziv radnog lista. U slučaju da radni list sadrži mnogo unosa koji se istodobno prikazuju unutar preglednika, odziv radnog lista bi se drastično smanjio i narušio korisničko iskustvo. Stoga je za kretanje kroz radni list predviđen fiksan dio redova koji se ažuriraju ovisno o korisnikovu kretanju kroz tablicu. Stupci se prikazuju u cijelosti jer je pretpostavka kako se broj varijabli neće povećavati iznad optimalnog broja.

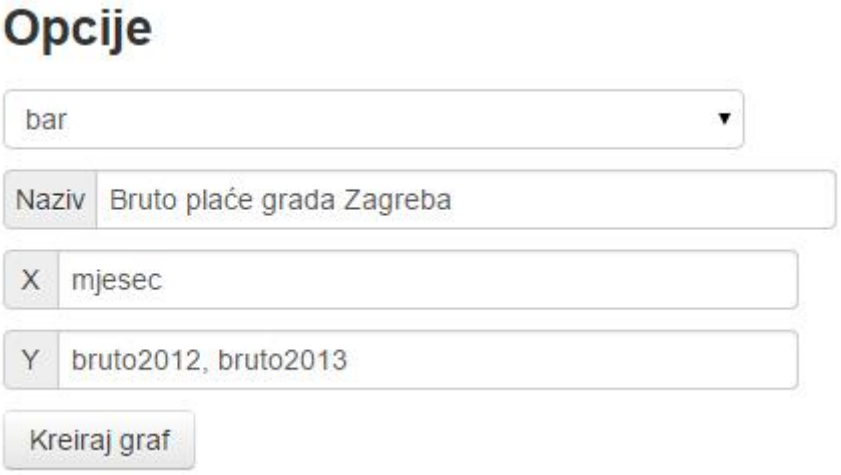
Kretanje kroz radni list moguće je na dva načina. Jedan je pomoću menija iznad proračunske tablice u kojem se nalazi indikator trenutnog prvog prikazanog reda i informacija o posljednjem redu koji sadrži unos. Nakon upisivanja željenog reda za prikaz vidljiva tablica će se ažurirati na način da prikaz započne od tog željenog reda. Drugi način za prolaz kroz radni list je pomicanjem kotačića miša u smjeru u kojem se želi pomaknuti prikaz.

U meniju iznad proračunske tablice također se nalaze akcije koje služe za dodavanje novog stupca koji ima smisao nove varijable, brisanje reda s kojim se briše pojedini unos i spremanje radnog lista u bazu podataka. Brisanje reda odvija se na način da se prvo odabere ćelija unutar reda koji se želi obrisati nakon čega se pritiskom na tipku „Briši red“ briše red i ažurira tablica. Dodavanje nove varijable uvijek se odvija na način da se doda novi stupac na zadnje mjesto radnog lista. Pokretanjem spremanja radnog lista na server se u JSON formatu šalju podaci iz tablica i informacije o radnom listu te se spremaju u bazu. Kao potvrda o uspješnom spremanju radnog lista tipka „Spremi radni list“ poprimit će zelenu boju, a u slučaju da je došlo do greške ona poprima crvenu boju.

5.2. Vizualizacija statističkih podataka

Kako bi se kreirani podaci mogli staviti u neki kontekst, podaci se mogu vizualizirati unutar grafova. Za kreiranje takvih grafova koristit će se *Google Charts API.*

Svaki radni list uz karticu za prikaz ćelija također sadrži i karticu gdje se nalazi forma za odabir opcija grafa i njegovo kreiranje. Tako je moguće definirati tip grafa, naziv i varijable koje se nalaze na apscisi i ordinati. Varijable koje se koriste pri kreiranju grafa moraju biti definirane unutar radnog lista, te će se njihove vrijednosti koristiti kao pojedine točke na grafikonu.



Slika 24. Forma za kreiranje grafa

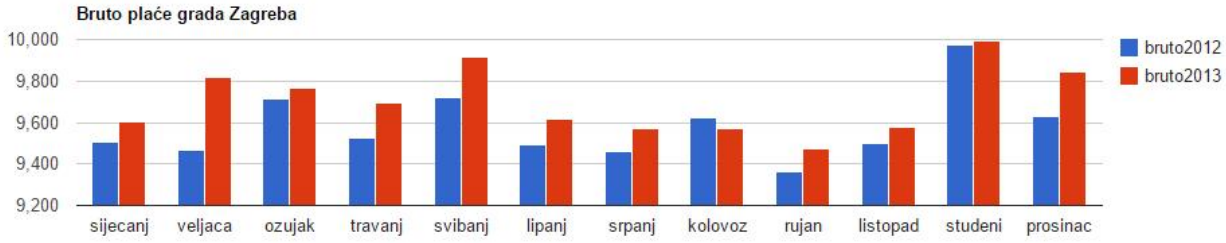
Forma prikazana na Slika 24. tako će kreirati graf tipa *bar*, što označava stupčasti grafikon. Naziv će biti „Bruto plaće grada Zagreba“. Na apscisi se nalazi varijabla *mjesec*, dok se na ordinati nalaze dvije varijable *bruto2012* i *bruto2013* odijeljene zarezom. Apscisa prima samo jednu varijablu, dok se na ordinati ovisno o tipu grafikona može koristiti više varijabli odvojenih zarezom.

Nakon pravilno ispunjene forme pritiskom na tipku „Kreiraj graf“ ispod forme će se prikazati novi grafikon koji odgovara odabranim opcijama.

5.2.1. Tipovi grafikona za vizualizaciju

5.2.1.1. Bar

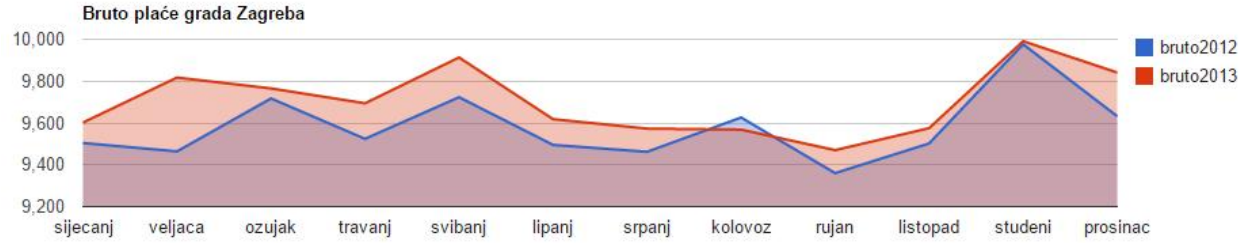
Stupčasti grafikon može na apscisi primiti varijablu tipa string ili tipa broj. Ordinata prima više varijabli koje sadrže samo brojeve.



Slika 25. Stupčasti grafikon

5.2.1.2. Area

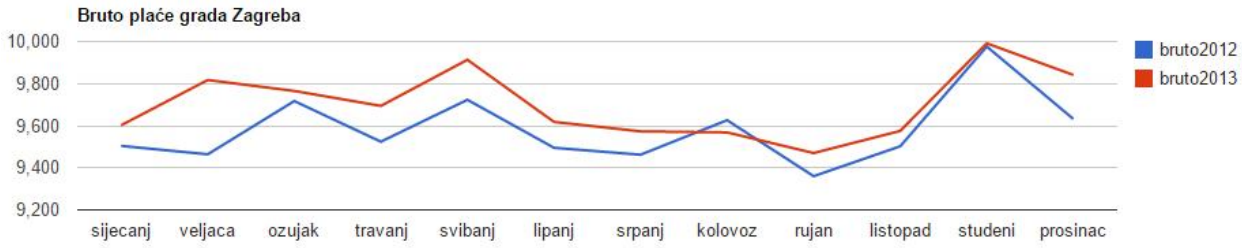
Apscisa površinskog grafa može biti string ili broj, dok ordinata mora sadržavati samo brojeve.



Slika 26. Površinski grafikon

5.2.1.3. Line

Linijski grafikon, poput površinskog, na apscisi može sadržavati brojeve ili stringove, dok ordinata mora sadržavati samo brojeve.

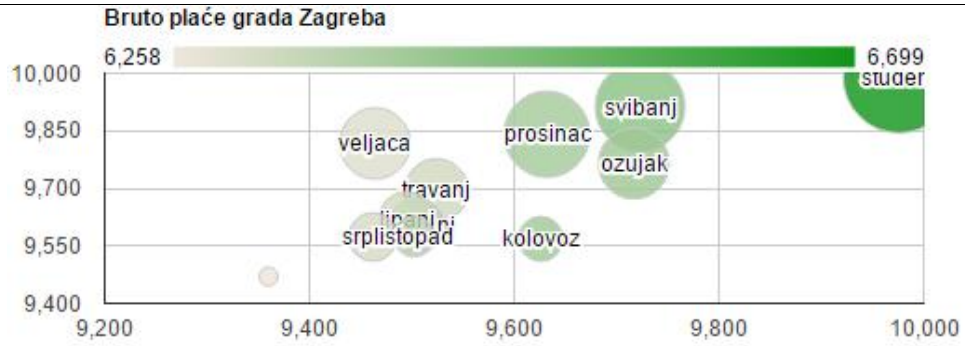


Slika 27. Linijski grafikon

5.2.1.4. Bubble

Grafikoni s mjehurićima mogu prikazati podatke u dvije do četiri dimenzije. Prve dvije dimenzije su koordinate, treća dimenzija se prikazuje kao boja, dok je četvrta dimenzija prikazana kao veličina mjehurića.

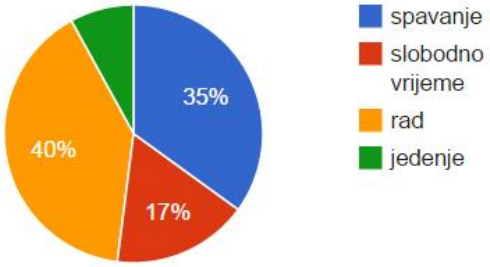
U ovom slučaju prilikom kreiranja grafikona na apscisi se zapravo bira varijabla koja će označavati naziv pojedinog mjehurića i ona mora biti string. Ordinata se popunjava na način da se prvo stavlja varijabla koja će obilježavati koordinatu na apscisi, a zatim druga varijabla određuje koordinatu na ordinati. Treća varijabla određuje nijansu boje, a četvrta veličinu mjehurića. Varijable koje se popunjavaju na ordinatu moraju biti tipa broj.



Slika 28. Grafikon s mjehurićima

5.2.1.5. Pie

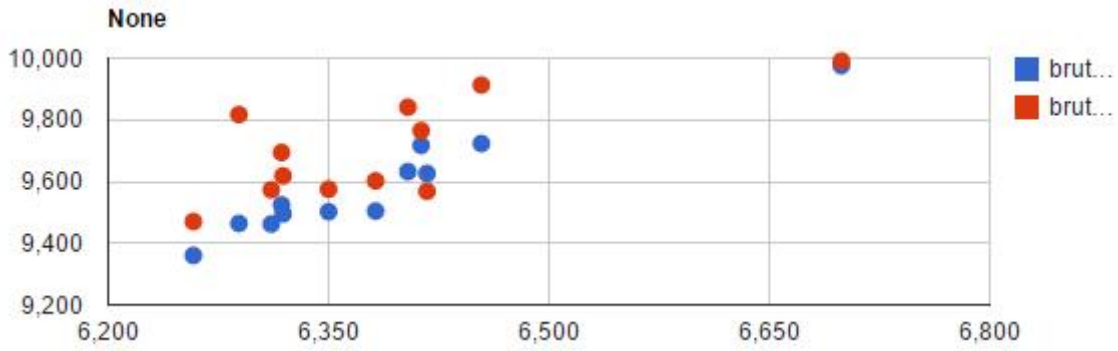
Okrugli grafikon prima dvije varijable od kojih se na apscisi nalazi varijabla sa stringovima, dok ordinata sadrži varijablu s brojevima.



Slika 29. Okrugli grafikon

5.2.1.6. Scatter

Raspršeni grafikon na apscisi i ordinati prima samo varijable s brojevima. Na apscisi je moguće koristiti više varijabli.



Slika 30. Raspršeni grafikon

6. ZAKLJUČAK

Razmatranjem proširivanja wiki filozofije na mogućnosti statističke obrade dobivena je nova dimenzija u kreiranju članaka. Umjesto da se uređivanje članaka sastoji samo od dokumentiranja činjenica, sa statističkom obradom se uređivanje članaka promiče u sferu znanstvenog istraživanja.

U ovom je radu osmišljen jedan takav sustav koji bi osim klasičnih mogućnosti wiki sustava sadržavao forme za kreiranje i uvoz statističkih setova podataka. Također su integrirane mogućnosti uređivanja i vizualizacije tih statističkih podataka.

Kao daljnja mogućnost razvoja moglo bi se razmotriti integriranje vizualiziranih podataka unutar članka. Također bi se valjalo razmotriti kreiranje mogućnosti statističke obrade i formi koje bi korisnik mogao koristiti za statističku obradu.

LITERATURA

[1] Andrew Lih: The Wikipedia Revolution, Hyperion, 2009.

[2] Phoebe Ayers, Charles Matthews, Ben Yates: How Wikipedia Works, No Starch Press, 2008.

[3] Michael Miller: Wikipedia In 10 Minutes, Pearson Education Inc., 2010.

[4] Dan O'Sullivan: Wikipedia A New Community of Practice?, Ashgate Publishing, 2009.

[5] John Broughton: Wikipedia: The Missing Manual, O'Reilly Media, 2008.

[6] Model-view-controller, http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller, 2.2.2015.

[7] Massimo Di Pierro: Web2py Complete Reference Manua,l 5th Edition, 2013.

[8] Massimo Di Pierro et al.: Web2py Application Development Cookbook, Packt Publishing, 2012.

[9] Brad Green, Shyam Seshadri, AngularJS, O'Reilly Media, 2013.

[10] Dan Wahlin: AngularJS in 60 Minutes, Wahlin Consulting, 2013./2014.

[11] Guide to AngularJS, https://docs.angularjs.org/guide, 13.10.2014.

PRILOZI

I. CD-R disc