

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Marko Sedlar

Spletne aplikacije za urejanje beležnic

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Aljaž Zrnec

Ljubljana 2013

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

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00387/2013

Datum: 04.04.2013

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

Kandidat: **MARKO SEDLAR**

Naslov: **SPLETNE APLIKACIJE ZA UREJANJE BELEŽNIC**
WEB APPLICATIONS FOR NOTEBOOK EDITING

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V zadnjih letih se je uporaba mobilnih naprav, kot so mobilni telefoni in tablični računalniki, skokovito povečala. Zaradi tega se je pojavila potreba po shranjevanju podatkov v oblaku in sinhronizaciji podatkov med napravami. Sočasno se je pojavila tudi potreba po sinhronizaciji beležnic. V okviru diplomskega dela predstavite aplikacije za izdelavo in sinhronizacijo beležnic med napravami ter opišite njihove funkcionalnosti. V okviru drugega dela diplomskega dela izdelajte aplikacijo za pisanje beležnic, z vsemi funkcionalnostmi, ki podpirajo njihovo kreiranje, sinhronizacijo in souporabo.

Mentor:

Dekan:

viš. pred. dr. Aljaž Zrnec

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Marko Sedlar, z vpisno številko **63080294**, sem avtor diplomskega dela z naslovom:

Spletne aplikacije za urejanje beležnic

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Aljaž Zrnec
- 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 8. oktober 2013

Podpis avtorja:

Zahvaljujem se vsem, ki so mi pomagali pri izdelavi diplomske naloge. Še posebej se zahvaljujem mentorju viš. pred. dr. Aljažu Zrnecu za nasvete in pomoč.

Zahvaljujem se tudi staršem, ki so me podpirali vsa leta študija.

Seznam uporabljenih kratic in simbolov

HTML - Hyper Text Markup Language; jezik za označevanje nadbесedil

CSS - Cascading Style Sheets; kaskadne stilske podloge

SQL - Structured Query Language; strukturirani povpraševalni jezik (za delo s podatkovnimi bazami)

JSON - JavaScript Object Notation; je preprost format za izmenjavo podatkov)

PHP - PHP Hypertext Preprocessor; programski jezik namenjen izdelavi spletnih strani)

REST - Representational State Transfer; način za prenos podatkov med strežnikom in odjemalcem

XML - Extensible Markup Language; razširljiv označevalni jezik

URL - Uniform Resource Locator; enolični krajevnik vira (spletni naslov)

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Primerjava aplikacij za beležnice	3
2.1	Evernote	4
2.2	Springpad	7
2.3	Google Keep	11
3	Uporabljene spletne tehnologije in orodja	15
3.1	MongoDB	15
3.2	Php	16
3.2.1	RESTful	16
3.3	HTML, JavaScript in CSS	17
3.3.1	jQuery	17
3.3.2	JSON	18
3.4	Eclipse	18
4	Razvoj aplikacije	21
4.1	Funkcionalne zahteve aplikacije	21
4.2	Strežnik	22
4.2.1	Podatkovna baza	22
4.2.2	Programska koda	24

KAZALO

4.3	Odjemalec	26
4.3.1	Sinhronizacija	26
4.3.2	Souporaba	27
4.3.3	Zgodovina sprememb	27
4.3.4	Grafični vmesnik	27
5	Opis Aplikacije	29
5.1	Grafični vmesnik	29
5.2	Shranjevanje	32
5.3	Urejanje besedila	32
5.4	Zgodovina sprememb	33
5.5	Souporaba	34
6	Sklepne ugotovitve	35

Slike

2.1	Slika prikazuje arhitekturo programov za delo z beležnicami. . .	4
2.2	Slika prikazuje spletno aplikacijo Evernote.	5
2.3	Slika prikazuje Windows aplikacijo Evernote.	5
2.4	Izbira tipa beležnice v Springpad.	8
2.5	Zvezek z beležnicami.	9
2.6	Zvezki v Springpad spletni aplikaciji.	10
2.7	Google Keep Android aplikacija.	12
2.8	Google Keep spletna aplikacija.	13
3.1	Rezultat poizvedbe podatkov v MongoDB.	16
3.2	Eclipse z dodatkom Aptana.	19
4.1	Slika prikazuje objektni model podatkovne baze.	23
4.2	Uporabnik shranjen v zbirki, prikazan v JSON obliki.	23
4.3	Beležnica shranjena v zbirki, prikazana v JSON obliki.	24
5.1	Slika prikazuje aplikacijo za zapisovanje beležnic, ki smo jo izdelali.	30
5.2	Aplikacija z grafičnim vmesnikom za mobilne naprave.	31
5.3	Slika prikazuje seznam naslovov v grafičnem vmesniku za mobilne naprave.	31
5.4	Slika prikazuje zgodovino sprememb beležnice.	33
5.5	Slika prikazuje okno, v katerem je seznam uporabnikov, ki si delijo beležnico.	34

Povzetek

V okviru diplomske naloge smo proučili delovanje nekaterih spletnih programov za pisanje beležnic. Bolj podrobno smo si ogledali delovanje spletnih aplikacij Evernote, Springpad in Google Keep. Izdelali pa smo tudi spletno aplikacijo. Aplikacija omogoča ustvarjanje in urejanje beležnic, ogled zgodovine in souporabo z drugimi uporabniki. Med besedilom in naslovi beležnic se samodejno tvorijo povezave in sicer tako, da aplikacija omogoča uporabo kot osebna wiki. Del aplikacije, ki se izvaja na strežniku je napisan z php, podatke pa ima shranjene v podatkovni bazi MongoDB. Drugi del, ki se izvaja v spletnem brskalniku, pa je narejen s programskimi jeziki HTML, CSS in JavaScript. Spletna aplikacija pa je narejena tako, da omogoča dostop do besedil beležnic tudi brez spletne povezave.

Ključne besede: spletna aplikacija, beležnica, php, HTML5, JavaScript, Evernote, Springpad, Google Keep

Abstract

In the thesis we studied the functions of some web programs for writing notebooks. More specifically, we have studied web applications Evernote, Springpad and Google Keep. We have also developed a web-based application. The application allows you to create and edit notes, see the history and share it with other users. Between text and notes addresses are automatically created links, so we can use the application as personal wiki. First part of the application, that is running on the server is written in php, and data is stored in the database MongoDB. Second part is carried out in a Web browser, and is made with programming languages HTML, CSS and JavaScript. The web application is designed to allow access to the notebooks, even without Internet connection.

Key words: web application, notebook, php, HTML5, JavaScript, Evernote, Springpad, Google Keep

Poglavje 1

Uvod

Danes ima vse več ljudi vse več elektronskih naprav, kot so računalniki, pametni telefoni in tablični računalniki. Zato se je pojavila tudi vse večja potreba po sinhronizaciji podatkov med napravami tako, da lahko, če nimamo dostopa do določene naprave, delo nadaljujemo z drugo. Tako je tudi vse večja potreba po sinhronizaciji beležnic med napravami.

V prvem delu diplomske naloge bomo opisali zakaj uporabljati programe za zapisovanje beležnic, ter opisali njihovo delovanje. Pri tem bo poudarek na spletnih aplikacijah Evernote, Springpad in Google Keep.

V drugem delu je poudarek na spletni aplikaciji, ki smo jo razvili, kot primer prototipa aplikacije za sinhronizacijo beležnic. Aplikacija je namenjena pisanju beležnic, narejena pa je kot spletna aplikacija, tako da deluje v spletnih brskalnikih. Poleg pisanja beležnic omogoča še ogled zgodovine sprememb ter deljenje in souporabo beležnic z drugimi uporabniki. Ker so beležnice shranjene na strežniku, imamo lahko vedno dostop do beležnic preko spleta, ker pa so shranjene tudi lokalno, v spletnem brskalniku, lahko dostopamo do njih tudi brez spletne povezave, če imamo prej prenesene beležnice.

V drugem delu smo opisali uporabljene spletne tehnologije, ter izdelavo aplikacije. Na koncu pa se nahaja še nekoliko podrobnejši opis funkcij naše aplikacije.

Poglavje 2

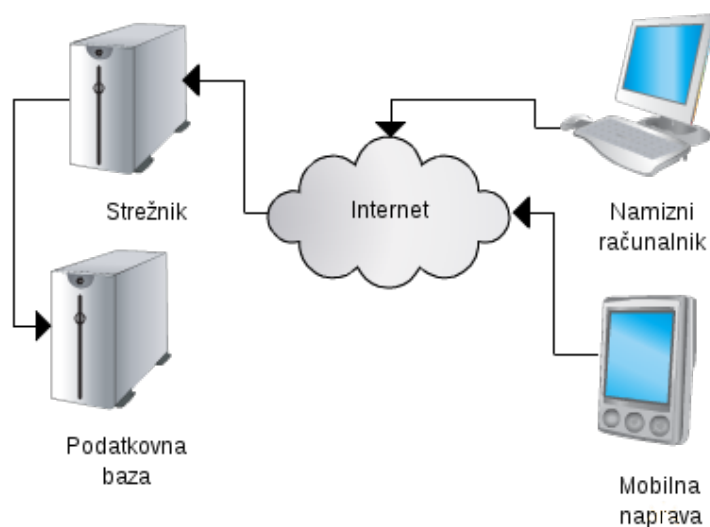
Primerjava aplikacij za beležnice

Na računalnikih imamo pogosto precej stvari, ki si jih moramo zapomniti. Zato jih pogosto zapišemo v kakšno datoteko. Čez čas se nam lahko tega nabere več in takrat več ni tako pregledno. Danes imamo tudi vse več prenosnih naprav, na katerih nam tudi pridejo prav takšne beležnice.

Zato je boljše, da za takšne stvari uporabimo programe, ki so namenjeni urejanju beležnic. Takšni programi nam omogočajo sinhronizacijo beležnic na vse naše naprave, lažje iskanje ter souporaba z drugimi uporabniki.

Arhitektura programov za beležnice je takšna, da aplikacije na napravah, ki imajo sinhronizirane beležnice niso povezane med sabo, ampak so vse naprave povezane le na strežnik. Ko ustvarimo ali spremenimo beležnico, se ta najprej prenese na strežnik. Potem aplikacije na ostalih napravah preverijo na strežniku, če je prišlo do kakšne spremembe in če je, spremenjeno beležnico posodobijo. Shema arhitekture lahko vidimo na sliki 2.1. Podatke o beležnicah pa imajo shranjene na strežnikih in v sami aplikaciji. To omogoča dostop do beležnic tudi brez internetne povezave.

Primerjali smo aplikacije za zapisovanje beležnic, ki omogočajo branje ter urejanje beležnic na več različnih napravah, kot so računalniki in mobilne naprave (pametni telefoni, tablice). Osredotočili smo se predvsem na aplika-



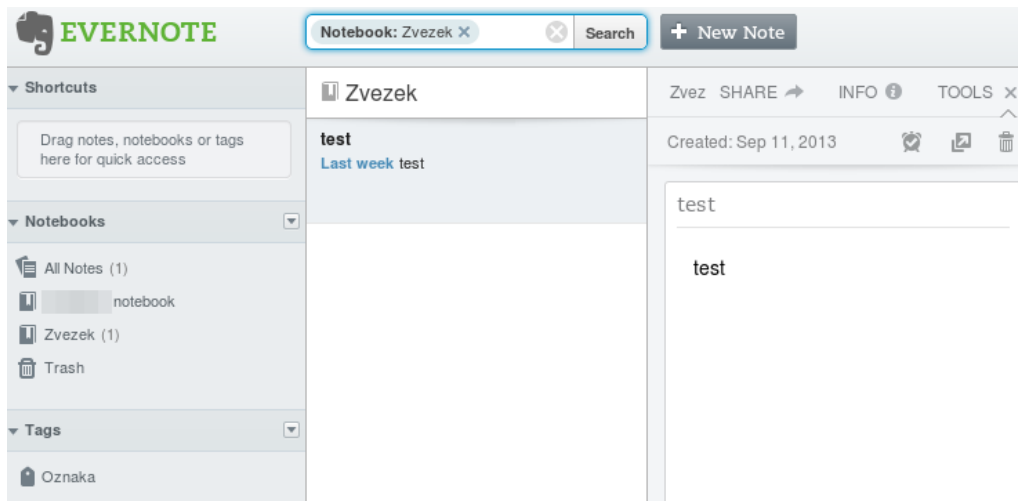
Slika 2.1: Slika prikazuje arhitekturo programov za delo z beležnicami.

cije, ki omogočajo dostop tudi preko spletnega brskalnika. Tako smo si bolj podrobno ogledali delovanje aplikacij Evernote, Springpad in Google Keep, ker delujejo dobro tudi na mobilnih napravah.

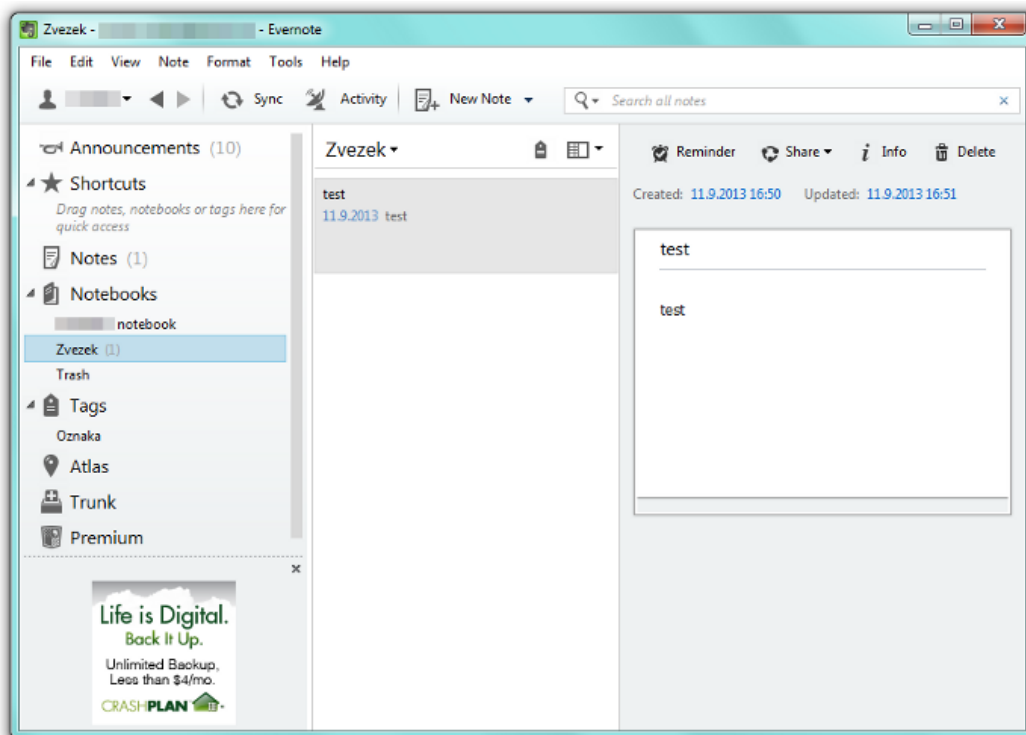
2.1 Evernote

Evernote vsebuje več aplikacij za delo z beležnicami. Testna verzija aplikacije Evernote je izšla junija 2008 [14]. Slika 2.2 prikazuje spletno aplikacijo Evernote.

Na voljo so tudi različice programa, ki jih namestimo na računalnik ali prenosne naprave, med katere spadajo telefoni in tablice. Od računalniških operacijskih sistemov obstajajo različice za Windows in Mac OS X operacijske sisteme. Na sliki 2.3 je prikazan program Evernote nameščen na Windows operacijski sistem. Med prenosnimi napravami pa so podprte naprave z operacijskimi sistemi Android, iOS, Windows phone, Blackberry in WebOS [3].



Slika 2.2: Slika prikazuje spletno aplikacijo Evernote.



Slika 2.3: Slika prikazuje Windows aplikacijo Evernote.

Slika 2.2 prikazuje spletno različico programa Evernote. Če pogledamo sliki 2.2 in 2.3 vidimo, da sta si spletna in Windows različica aplikacije zelo podobni. Tako se ni potrebno ponovno privajati drugemu uporabniškemu vmesniku, ko gremo iz ene naprave na drugo.

Ko uporabljamo Evernote preko aplikacije, ki je nameščena lokalno na računalniku, so tudi podatki shranjeni lokalno. To nam omogoča, da lahko urejamo podatke tudi brez internetne povezave. Ko pa imamo povezavo, takrat pa se podatki prenesejo tudi na strežnike. Prav tako se podatki prenesejo tudi iz strežnikov v nameščeno aplikacijo, če beležnice ustvarimo ali uredimo na kateri drugi napravi. Vse beležnice pa lahko vidimo in urejamo tudi v spletni aplikaciji.

Grafični vmesnik aplikacije je razdeljen na 3 dele (Sliki 2.2 in 2.3). V prvem delu imamo bližnjice, zvezke in oznake. Več beležnic lahko združimo v en zvezek. To lahko naredimo tako, da beležnico v drugem delu grafičnega vmesnika primemo in jo povlečemo v zvezek. Ena beležnica je lahko samo v enem zvezku. Vsaka beležnica pa ima lahko več oznak. V bližnjice pa lahko dodamo posamezne beležnice, zvezke in oznake. To storimo tako, da primemo in potegnemo v bližnjico beležnico, zvezek ali oznako.

V drugem delu imamo prikazane naslove beležnic z datumom in nekaj teksta. Tukaj so prikazane različne beležnice. Če kliknemo na katerikoli zvezek, bodo v tem delu prikazane samo beležnice, ki so v tem zvezku. Enako velja za oznake. Če pa vnesemo besedilo v polje za iskanje, bodo tukaj prikazane le beležnice, katerih tekst se ujema z iskanimi besedami.

V tretjem delu je prikazana beležnica. Tukaj jo tudi urejamo. Pri urejanju teksta v beležnici, imamo kar veliko možnosti, kot so spreminjanje barve črk, pisave in velikosti. Označenemu besedilu lahko določimo tudi povezavo na poljuben spletni naslov. K beležnicam lahko vključimo različne datoteke. Dodamo pa jim lahko tudi opomnik.

Da lahko uporabljamo Evernote, si moramo ustvariti račun na njihovi spletni strani. Evernote lahko nato uporabljamo brezplačno, a imamo pri tem nekaj omejitev. Te omejitve so odpravljene ob plačilu. Brezplačna ver-

zija ima omejeno količino prenosa podatkov. V enem mesecu lahko prenesemo 60MB podatkov. V plačljivi verziji pa lahko na mesec prenesemo 1GB podatkov. Plačljiva verzija nam omogoča še iskanje po vsebini PDF datotek, ki so pripete k beležnici [14].

Evernote beležnica nam tako omogoča urejanje beležnic, souporabo ter razvrščanje beležnic v zvezke. Zaradi tega je primerna za uporabo tudi, če imamo večje število beležnic. Slabost je, da je aplikacija plačljiva, če želimo povečati omejitev prenosa podatkov.

Prednosti:

- omogoča oblikovanje besedil (barva pisave, velikost črk),
- razvrščanje beležnic v zvezke,
- souporaba zvezkov z drugimi uporabniki.

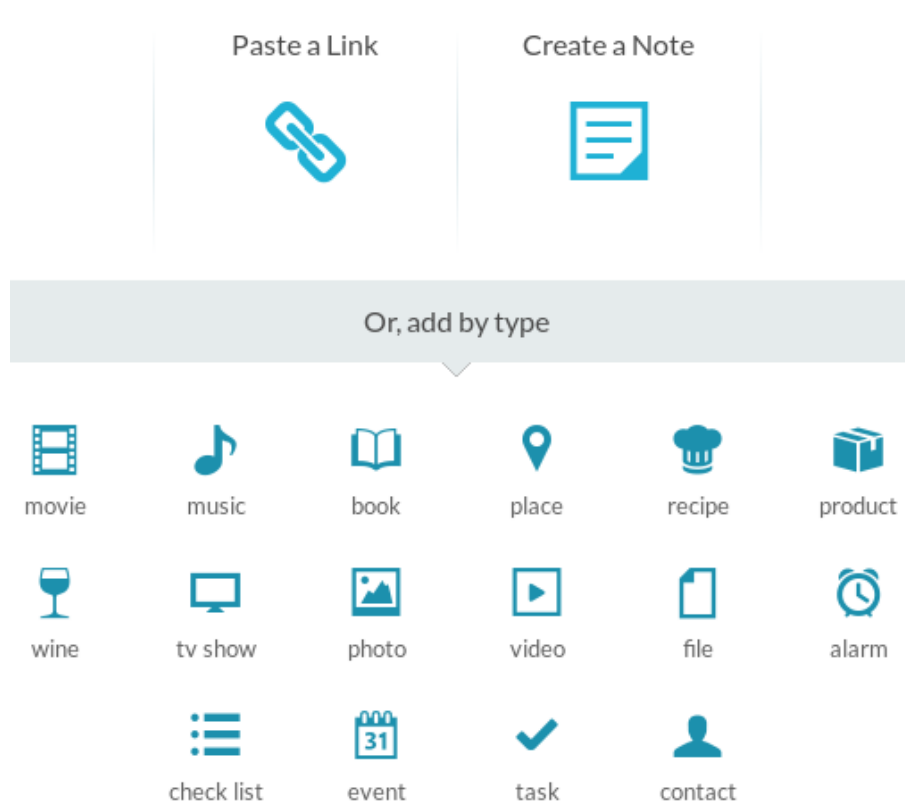
Slabost:

- aplikacija je plačljiva, če želimo povečati omejitev prenosa podatkov.

2.2 Springpad

Springpad beležnica je izšla 12. novembra 2008. Poleg spletne aplikacije za delo z beležnicami, imajo tudi aplikaciji, ki se namestita na mobilne naprave z operacijskima sistemoma Android in iOS [18]. Za uporabo aplikacije se moramo najprej registrirati.

Posamezne beležnice se v Springpadu imenujejo spring. Za ustvarjanje nove beležnice moramo klikniti gumb add v zgornjem desnem kotu. Nato se nam odpre okno, v katerem izberemo tip springa oziroma beležnice. Beležnica je lahko tipa spletni naslov, za navadno besedilo, film, glasba, knjiga in še veliko drugih. Izbiro tipov beležnic lahko vidimo na sliki 2.4. Ko izberemo tip, se nam odpre okno za vnos beležnice, ki pa je prilagojeno tipu beležnice, ki smo ga izbrali. Ko vnesemo potrebne podatke in okno

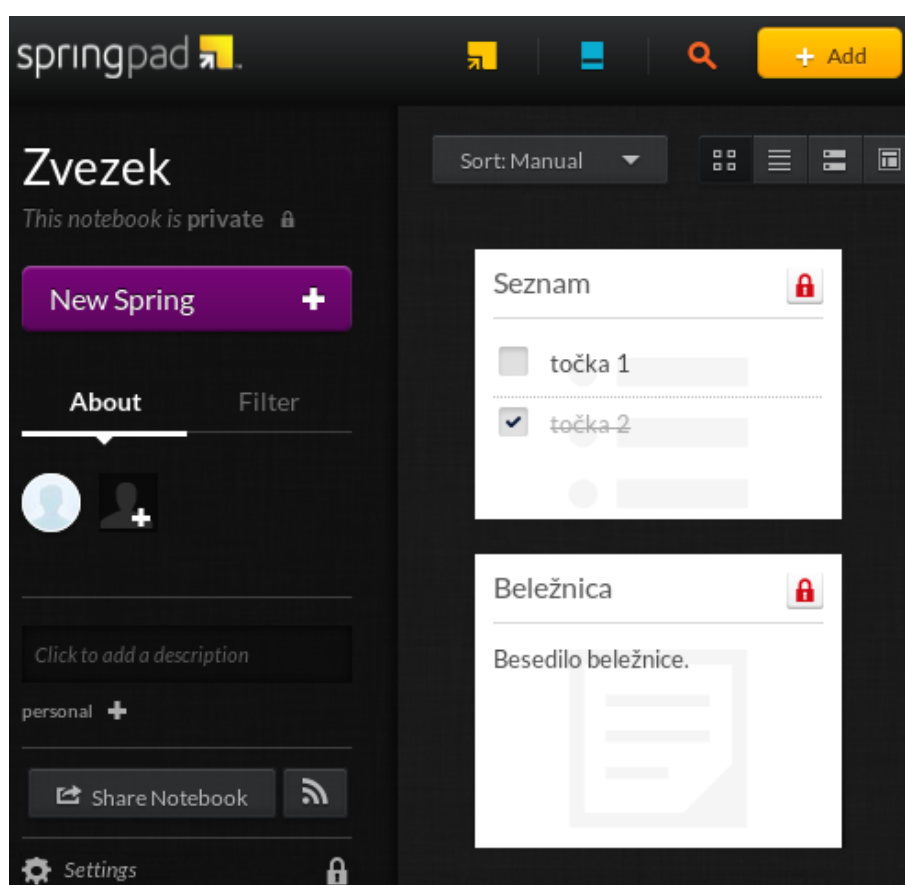


Slika 2.4: Izbira tipa beležnice v Springpad.

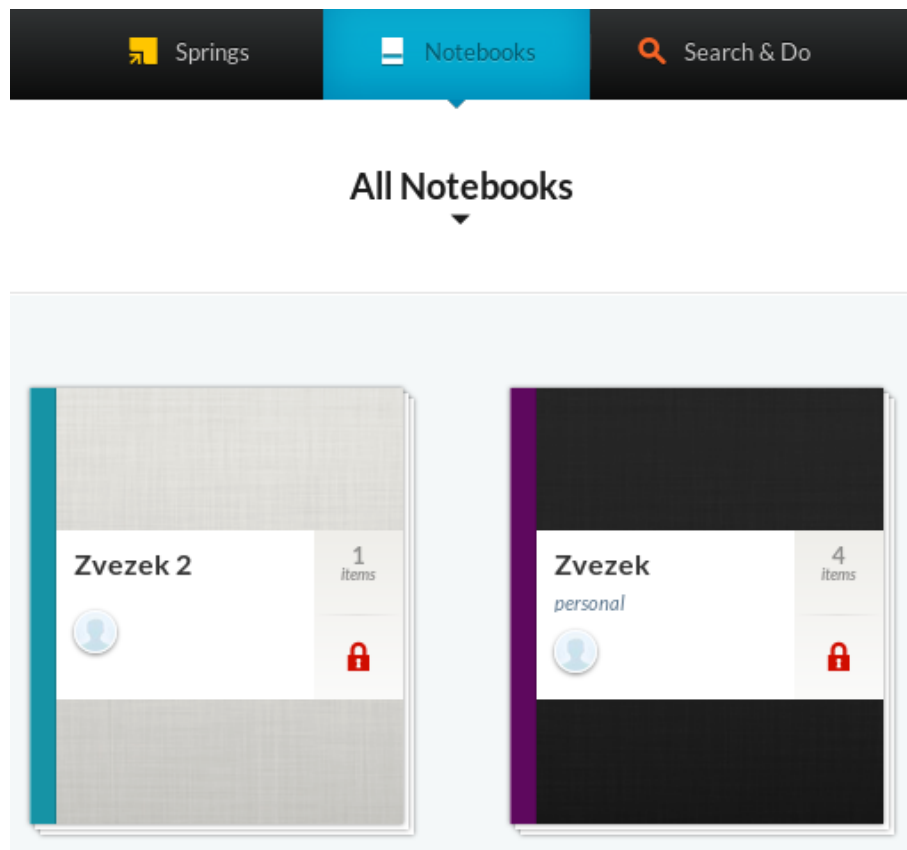
zapremo, se nam ustvari nova beležnica. Na voljo imamo tudi beležnice, v katere se vstavijo slike ali datoteke.

Beležnice lahko razporedimo v zvezke. V oknu za ustvarjanje zvezka, moramo najprej vnesti ime zvezka. Nato imamo na voljo še izbiro barve za ta zvezek. Izberemo lahko barvo platnice ter barvo vezave zvezka. Ko to izberemo se nam odpre zvezek, ter v njem okno za izbiro tipa beležnice (Slika 2.4). Ko naredimo ali odpremo zvezek, se grafični vmesnik obarva v barve, ki smo jih izbrali, ko smo ustvarjali zvezek. Zvezek lahko vidite na sliki 2.6, odprtega pa na sliki 2.5. Zvezke imamo lahko v souporabi z drugimi izbranimi uporabniki. Lahko pa tudi dovolimo, da vidijo vsebino zvezka vsi.

Na vrhu spletne strani imamo tri gumbe. To so Springs, Notebooks in Search & Do. Če kliknemo na Springs, se nam prikažejo beležnice urejene



Slika 2.5: Zvezek z beležnicami.



Slika 2.6: Zvezki v Springpad spletni aplikaciji.

po datumu. Če kliknemo na drugi gumb Notebooks, vidimo seznam vseh zvezkov, kar lahko vidimo na sliki 2.6. Po kliku na tretji gumb za iskanje, pa se nam prikaže polje, kamor lahko napišemo iskane besede. Iskalnik išče po vsebini beležnic ter zvezkov. Tukaj pa lahko iščemo tudi uporabnike.

Springpad nam omogoča izdelavo beležnic različnih tipov. Takšne beležnice so zelo pregledne. Omogoča nam tudi ustvarjanje zvezkov v različnih barvah, ki so prav tako zelo pregledni. Te pa imamo lahko tudi v souporabi z drugimi uporabniki. Slabost je, da ne omogoča združevanja različnih tipov znotraj ene beležnice. Zaradi tega ne moremo ustvariti beležnice z besedilom, ki bi vsebovala povezavo na določeno spletno stran, na katero bi lahko kliknili, ker je za spletne povezave drugačen tip beležnice. Slabost je še ta, da ima

za namizne računalnike le spletno aplikacijo, do katere nimamo dostopa brez spletne povezave.

Prednosti:

- omogoča izdelavo beležnic različnih tipov,
- razvrščanje beležnic v zvezke,
- beležnice in zvezki so zelo pregledni,
- souporaba zvezkov z drugimi uporabniki.

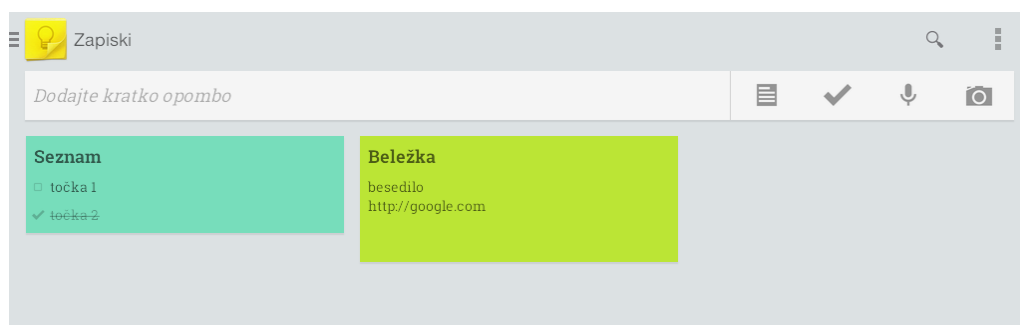
Slabosti:

- ne omogoča združevanja različnih tipov znotraj ene beležnice,
- z namiznimi računalniki ne moremo dostopati do beležnic, brez spletne povezave.

2.3 Google Keep

Keep je novejša aplikacija za zapisovanje, ki so jo naredili pri podjetju Google. Prva verzija je izšla marca 2013 [17]. Pred to aplikacijo so imeli aplikacijo Google Notebook, ki pa od julija 2012 več ni na voljo [4]. Beležnice lahko urejamo v spletni aplikaciji ali v aplikaciji, ki jo namestimo na mobilno napravo z operacijskim sistemom Android. Aplikacijo nameščeno na Android lahko vidimo na sliki 2.7. Če primerjamo grafični vmesnik Android in spletne aplikacije, ki je na sliki 2.8, vidimo, da sta si zelo podobna. Dostop do beležnic brez povezave pa je mogoč le z Android aplikacijo.

V primerjavi z Notebook, ki je imel precej možnosti za urejanje teksta, jih Keep sploh nima. V beležnice lahko vključimo poleg teksta tudi slike. Dodamo jim lahko tudi opomnike. Za boljši pregled jim lahko spremenimo barvo. Primer beležnice lahko vidimo na sliki 2.8, kjer je beležnica v zeleni barvi.



Slika 2.7: Google Keep Android aplikacija.

Poleg beležnic lahko ustvarjamo tudi sezname. Na začetku vsake točke v seznamu je narisana kvadrata, v katerega lahko kliknemo. Ob kliku se v kvadrata nariše kljukica in točka se prečrta. Na sliki 2.8 lahko vidite primer seznama v modri barvi.

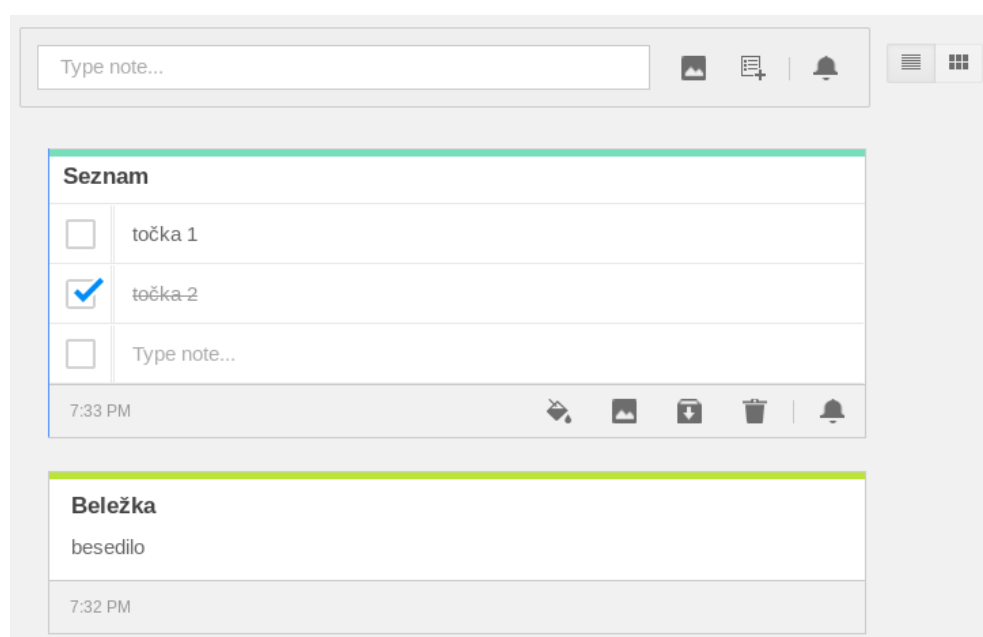
Google Keep je zelo enostavna za uporabo, saj omogoča le ustvarjanje beležnic in seznamov. Zaradi različnih barv so beležnice tudi zelo pregledne. Keep je primeren predvsem za kratke, vsakodnevne zapiske. Slabost pa je, da ne omogoča souporabe beležnice z drugimi uporabniki.

Prednosti:

- je zelo enostaven za uporabo,
- omogoča ustvarjanje beležnic in seznamov,
- beležnicam lahko spreminjamo barvo.

Slabost:

- ne omogoča souporabe beležnic,
- na spletne naslove se ne da kliknit,
- z namiznimi računalniki ne moremo spreminjati beležnic, brez spletne povezave.



Slika 2.8: Google Keep spletna aplikacija.

Poglavje 3

Uporabljene spletne tehnologije in orodja

Za razvoj aplikacije smo uporabljali več tehnologij. Za shranjevanje podatkov smo uporabili podatkovno bazo MongoDB. Za spletni strežnik pa smo uporabili Apache. Prvi del Aplikacije, ki je na strežniku, smo naredili v programskem jeziku php, za drugi, ki se izvaja na strani odjemalca oziroma v spletnem brskalniku, pa smo uporabili jezike HTML, JavaScript in css. Vso programsko kodo smo napisali v programu Eclipse, z dodatkom Aptana.

3.1 MongoDB

MongoDB je zelo zmogljiva, prilagodljiva in razširljiva podatkovna baza za splošno uporabo [1]. Je tudi najbolj razširjena neSQL podatkovna baza, ker je zelo enostavna za namestitev in je zelo hitra [2]. Podatke nima shranjene v tabelah ampak v zbirkah, v njih pa so shranjeni dokumenti. Zbirke nimajo vnaprej določene sheme, tako da lahko vsebujejo dokumente z različnimi shemami. Podatki pa so shranjeni v JSON obliki. Slika 3.1 prikazuje rezultat poizvedbe v MongoDB. Rezultat poizvedbe je vse, kar se nahaja v zbirki test, v tem primeru je to le en dokument. Podatki so v JSON obliki.

```
>
db.test.find();

[
  { "ime" : "Janez", "_id" : { "$oid" : "522df753cc93742c160d56e5" },
    "priimek" : "Novak", "naslov" : { "mesto" : "Ljubljana", "ulica" : "Slovenska
    cesta" } }
]
>
```

Slika 3.1: Rezultat poizvedbe podatkov v MongoDB.

3.2 Php

PHP ali PHP Hypertext Preprocessor je programski jezik, ki generira dinamično HTML kodo, za spletno stran. Prva verzija je izšla leta 1995, naredil jo je Rasmus Lerdorf. Imenovala se je Personal Home Page Tools, kar v slovenščini pomeni orodja za osebno spletno stran. Danes je php najpogostejši jezik za razvoj dinamičnih spletnih strani [2].

Za izvajanje spletnih strani narejenih s php potrebujemo spletni strežnik. Uporabili smo spletni strežnik Apache. Prva verzija je izšla leta 1995, naredil pa jo je Robert McCool [10].

3.2.1 RESTful

RESTful je način za prenos podatkov med strežnikom in odjemalcem. Leta 2000 ga je predstavil Roy Fielding. Za RESTful spletne storitve ni pomemben jezik v katerem so narejene, le uporabljati se mora HTTP metoda za prenos informacij, struktura imenikov mora biti kot URL in podatki se morajo prenašati v XML ali JSON obliki. RESTful pozna tudi različne oblike zahtev (ang. request methods). To so:

- GET - za pridobitev,
- POST - za ustvarjanje,
- PUT - za spreminjanje,

- DELETE - za izbris.

[8]

3.3 HTML, JavaScript in CSS

HTML, CSS in JavaScript so programski jeziki s katerimi se izdelava vsebina, izgled ter logika spletne aplikacije, ki se prikazuje na računalniku uporabnika.

HTML ali Hyper Text Markup Language je označevalni jezik s katerim izdelujemo spletne strani. HTML datoteka vsebuje HTML elemente, ki so v oklepajih (<, >) in so ponavadi v parih (<p>... </p>)[15].

CSS je kratica za Cascading Style Sheets. To je preprost slogovni jezik s katerim definiramo, kako naj se prikažejo HTML elementi na spletni strani. Elementom lahko določimo barvo, velikost, pozicijo in še veliko drugih atributov. Bistvo uporabe CSS datotek je, da ločimo izgled spletne strani od vsebine. Tako se koda z izgledom ne ponavlja na vsaki pod strani. Lažja pa je tudi zamenjava izgleda [12].

JavaScript je skriptni programski jezik, ki se uporablja za dinamično izvajanje spletnih strani. Izvaja se na strani odjemalca spletne strani v spletnem brskalniku. Omogoča nam komunikacijo z uporabnikom, z klikanjem ali vnosem podatkov, asinhrono komunikacijo s strežnikom, tako da lahko podatke prenesemo iz ali na strežnik brez osveževanja spletne strani. Omogoča nam tudi spreminjanje vsebine in izgleda spletne strani [16].

3.3.1 jQuery

jQuery je knjižnica za JavaScript jezik. Z njo lažje iščemo in spreminjamo vsebino HTML dokumenta, olajša nam delo z dogodki, animacijami in Ajaxom, ker ima metode, ki delujejo v več spletnih brskalnikih [6]. Celotna knjižnica se nahaja v le eni datoteki. Vključitev jQuery knjižnice v HTML spletno stran je zelo enostavna. Lahko jo prenesemo iz strežnika in se na njo sklicujemo na lokalnem strežniku. Lahko pa jo vključimo tudi tako, da jo prenesemo s spletnega naslova <http://code.jquery.com/jquery-latest.pack.js>, v tem primeru je

ne potrebujemo na lokalnem strežniku. Oba načina imata nekaj prednosti in slabosti. V prvem načinu imamo več kontrole nad knjižnico, lahko jo tudi spremenimo. Vendar jo moramo, ko pride novejša različica, ročno zamenjati, kar pa ni tako zelo nujno. Če se sklicujemo na zadnjo različico knjižnice na njihovi domači strani, bomo imeli vedno zadnjo različico knjižnice, kar pa lahko pripelje do nevšečnosti, če nam z novejšo knjižnico aplikacija ne deluje pravilno.

3.3.2 JSON

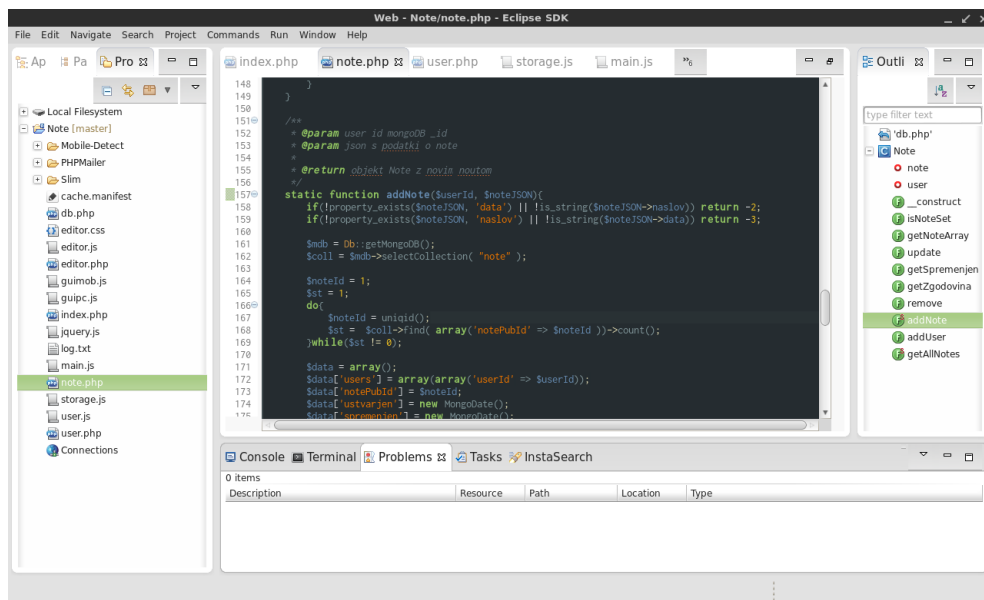
JSON je preprost format za izmenjavo podatkov, ki pa je popolnoma neodvisen od programskega jezika. Kratica pomeni JavaScript objektni zapis (ang. JavaScript Object Notation). Zapis je precej enostaven za branje in pisanje tudi ljudem. Podatki so zapisani v parih. Najprej ime in nato še vrednost, ločuje pa ju dvopičje. Lahko pa tudi vsebuje tabelo, kjer ima tabela ime, posamezne vrednosti pa so našteje [9]. Primer JSON lahko vidite na sliki 3.1.

V aplikaciji smo uporabili JSON za shranjevanje podatkov v lokalni spomin v spletnih brskalnikih, pri prenosu podatkov med strežnikom in odjemalcem. V tem formatu pa so shranjeni tudi podatki v MongoDB podatkovni bazi.

3.4 Eclipse

Eclipse je programsko razvojno okolje, ki podpira številne jezike. Program je odprtokoden in napisan večinoma v Javi in deluje na več operacijskih sistemih [13].

Da Eclipse podpira spletne programske jezike, smo mu namestili dodatek Aptana. Aptana podpira vse spletne programske jezike, ki smo jih potrebovali, pri izdelavi aplikacije. To so php, JavaScript, HTML in CSS. Podpira pa še nekaj drugih jezikov. [11]. Slika 3.2 prikazuje razvojno okolje Eclipse.



Slika 3.2: Eclipse z dodatkom Aptana.

Poglavje 4

Razvoj aplikacije

Razvoj aplikacije smo razdelili na dva dela. Na del, ki se izvaja na strežniku in del, ki se izvaja pri uporabniku v spletnem brskalniku. Komunikacija pa poteka preko Ajaxa in RESTful spletne storitve.

4.1 Funkcionalne zahteve aplikacije

Aplikacija bo podprla naslednje funkcionalnosti:

- pisanje in urejanje beležnic: lahko se ustvarja in briše beležnice ter ureja obstoječe,
- oblikovanje besedila: del besedila se bo lahko napisal z odebeljeno pisavo,
- samodejno iskanje spletnih povezav: če bo med besedilom beležnice napisan spletni naslov, ga bo označilo tako, da se bo lahko kliknilo nanj,
- delovanje na različnih napravah: aplikacija mora delovati na različnih napravah, kot so namizni računalniki, pametni telefoni in tablični računalniki. Naša aplikacija bo imela, za razliko od ostalih, za to le eno aplikacijo, ki bo delovala na vseh napravah,

- sinhronizacija med napravami: spremembe beležnic se morajo samodejno sinhronizirati na vseh napravah,
- dostop brez povezave: do beležnic moramo imeti dostop tudi, če nimamo spletne povezave,
- souporaba: omogočati mora souporabo beležnic z drugimi uporabniki,
- ogled zgodovine sprememb: lahko si bomo ogledali starejše verzije beležnic,
- osebna wiki: to nam mogoča da uporabljamo beležnice, kot wiki zato, ker se samodejno ustvarijo povezave med besedilom v beležnici in naslovi ostalih beležnic. V tej funkcionalnosti se naša aplikacija tudi razlikuje od ostalih opisanih v Poglavju 2.

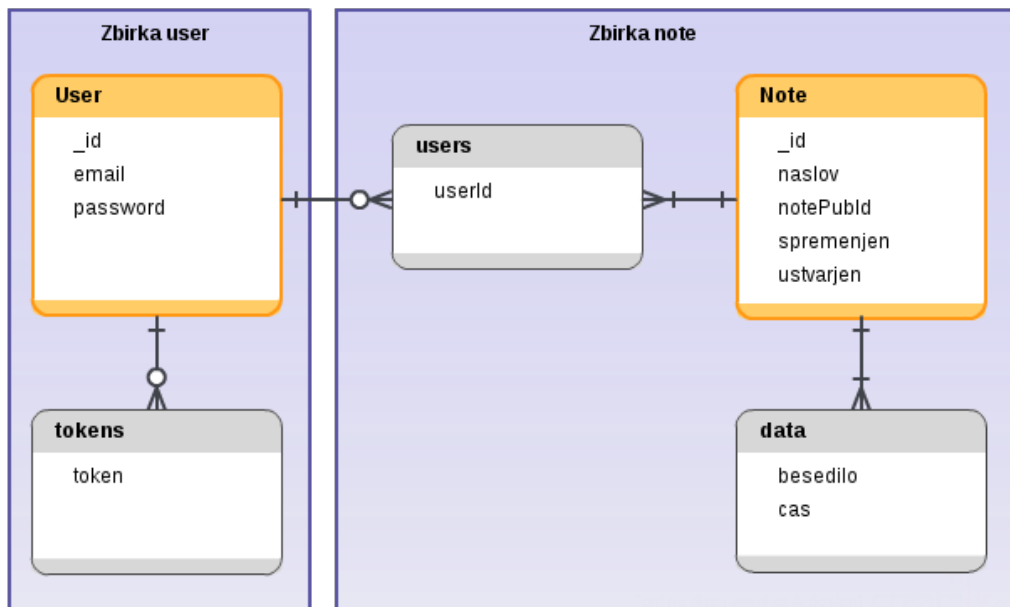
4.2 Strežnik

Del aplikacije, ki se izvaja na strežniku smo naredili s programskim jezikom php. Ta del ima podatkovno bazo ter RESTful spletno storitev za komunikacijo z odjemalcem.

4.2.1 Podatkovna baza

Podatke imamo shranjene v MongoDB podatkovni bazi. Ker sheme dokumentov v podatkovni zbirki ni potrebno določiti vnaprej, potrebuje aplikacija le dve zbirki. V prvi zbirki, ki se imenuje user, so shranjeni vsi uporabniški podatki. V drugi, ki se imenuje note, pa so shranjene vse beležnice. Objektivi model podatkovne baze vidimo na sliki 4.1.

V zbirki z uporabniki imamo shranjen elektronski naslov in geslo uporabnika. Poleg tega imamo v zbirki shranjene še žetone, s katerimi uporabnik dostopa do beležnic. Slika 4.2 prikazuje, shemo podatkov zbirke user. Ker shema ni vnaprej definirana, imajo lahko uporabniki različno število žetonov (ang. tokens). Na sliki 4.2 vidimo, da ima uporabnik dva žetona.



Slika 4.1: Slika prikazuje objektni model podatkovne baze.

```

{
  "_id": "52404ca4d191881c1f000000",
  "email": "user@email.si",
  "password": "2d4dc7a5c7e81c5f338d0bf66bae46a67b8b547f",
  "tokens": [
    {
      "token": "a42fb6010a6b8b376aefd1217341bb4592350371"
    },
    {
      "token": "6c0156d28d55f26c24161cfaeb0c3bc98318e224"
    }
  ]
}

```

Slika 4.2: Uporabnik shranjen v zbirki, prikazan v JSON obliki.

```
{
  "_id": "51b60c7fd191888931000001",
  "data": [
    {
      "besedilo": "besedilo v1",
      "cas": "2013-06-10T17:27:27.124Z"
    },
    {
      "besedilo": "besedilo v2",
      "cas": "2013-06-10T17:28:15.852Z"
    }
  ],
  "naslov": "Naslov beležke",
  "notePubId": "51b60c7f1e414",
  "spremenjen": "2013-06-10T17:28:15.852Z",
  "users": [
    {
      "userId": "52404ca4d191881c1f000000"
    },
    {
      "userId": "51adef36d19188df4a000000"
    }
  ],
  "ustvarjen": "2013-06-10T17:27:27.124Z"
}
```

Slika 4.3: Beležnica shranjena v zbirki, prikazana v JSON obliki.

Druga zbirka vsebuje vse podatke o beležnici. Primer vidimo na sliki 4.3. Objekt data vsebuje podatke o besedilu beležnice in času spremembe. Besedil s časom je lahko več. To nam omogoča, da imamo shranjenih več različic besedila, tako da lahko gledamo zgodovino sprememb. V objektu users imamo shranjene identifikacijske številke uporabnikov, ki imajo dostop do beležnice. V tem objektu je lahko tudi samo en uporabnik, če beležnice ne delimo z nikomer.

4.2.2 Programska koda

Pri izdelavi programske kode smo uporabili objektno usmerjen pristop. Tako imamo dva glavna razreda. User za delo z uporabniki in note za delo z beležnicami. Oba objekta imata tudi dostop do podatkovne baze. V razredu user imamo vse metode, ki so potrebne za upravljanje z uporabniki. Prav

tako imamo v razredu note vse metode za delo z beležnicami.

Vsi spletni naslovi, ki kažejo na spletno aplikacijo, so preusmerjeni na `index.php`. Ko dostopamo do korenkega imenika `/` na strežniku, nam ta vrne HTML kodo spletne aplikacije. Preveri se tudi kakšna je naprava, ki dostopa do strežnika, tako da je grafični vmesnik prilagojen odjemalcu. Za ugotavljanje tipa naprave, ki dostopa do strežnika, smo uporabili knjižnico `Mobile Detect`. To je lahka PHP knjižnica, ki razbere tip naprave iz podatkov v HTTP glavi [7].

Lahko pa tudi sami določimo kateri grafični vmesnik bi želeli imeti. Če pristopamo do imenika `/d` bomo dobili grafični vmesnik za računalnike tudi na mobilni napravi. Ob dostopu na `/m` pa dobimo dostop do mobilne verzije.

Do vseh funkcij, ki jih ima strežnik se dostopa preko RESTful spletne storitve. Metode za delo z uporabniki so v imeniku `/api/user/` in nato se doda zadnje ime, ki pove kaj želimo narediti. Vsi načini zahteve (ang. request methods) za delo z uporabnikom so tipa POST. Za dodajanje uporabnika je pot `/api/user/register`, kot podatke pa mora vsebovati naslov elektronske pošte uporabnika in geslo. V tem primeru se ustvari nov uporabnik, če še nobeden uporabnik nima enakega elektronskega naslova.

Na takšen način dostopamo še do naslova za prijavo in odjavo. Pri prijavi je potrebno poslati naslov elektronske pošte in geslo. Če sta oba pravilna, se nazaj pošlje kot odgovor žeton. Žeton se nato tudi shrani v podatkovno bazo. Vsi žetoni morajo biti med seboj različni. Zato to še prej preverimo v bazi. Vsak uporabnik ima lahko tudi več žetonov. S tem žetonom nato spletna aplikacija dostopa do strežnika. Pri odjavi pošljemo žeton na naslov `/api/user/logout`.

Za dostop in delo z beležnicami se uporablja naslov `/api/note/`. Novo beležnico dodamo tako, da pošljemo žeton uporabnika, naslov beležnice in besedilo beležnice na naslov `/api/note/` z načinom zahteve PUT. Na enak način tudi spremenimo vsebino beležnice, le da takrat na konec naslova dodamo še šifro beležnice, ki jo želimo spremeniti. Na podoben način se tudi izbriše in dobi vsebina beležnice. Pri obeh se pošlje le žeton uporabnika na

naslov `/api/note/` s šifro. Pri brisanju je način zahteve `DELETE`, pri prenosu pa `POST`.

Za delo z beležnicami imamo še naslov `/api/note/`, z načinom zahteve `POST`, ki vrne seznam beležnic s časi zadnje spremembe. S pomočjo tega seznama se na odjemalcu, najdejo beležnice, ki jih je potrebno posodobiti. Na naslovu `/api/note/` z dodano šifro beležnice in časom pa dobimo vsebino beležnice, ki je bila ob takratnem času. Beležnico dodamo v skupno uporabo tako, da pošljemo žeton uporabnika in elektronski naslov uporabnika s katerim želimo deliti beležnico, na naslov `/api/note/share/` in dopisano šifro beležnice.

Pri vsakem klicu strežnik odgovori s podatki v `JSON` obliki. Tudi če je način zahteve takšen, da ni potrebno, da pošlje strežnik odgovor, ga vseeno pošlje. V odgovoru je zapisano, če se je željena zahteva uspešno izvedla. Če se ne, je v odgovoru tudi razlog, zakaj je prišlo do napake.

4.3 Odjemalec

Del aplikacije, ki se izvaja na spletnem brskalniku, smo naredili s `HTML5`, ki za razliko od prejšnjih verzij, omogoča lokalno shranjevanje podatkov v spletnem brskalniku. To nam je omogočilo, da smo naredili spletno aplikacijo, ki je dosegljiva tudi takrat, ko nimamo internetne povezave [5].

Spletna aplikacija je narejena tako, da je ni nikoli potrebno osvežiti. Celotna aplikacija se vedno izvaja na enaki spletni strani. Vsa komunikacija med spletnem brskalnikom in strežnikom pa poteka z `Ajaxom`.

4.3.1 Sinhronizacija

Beležnice so shranjene na strežniku in tudi lokalno v spletnem brskalniku. Vsakih 30 sekund (ta čas lahko spremenimo v nastavitvah.) se aplikacija poveže z strežnikom in preveri spremembe. Strežnik pošlje lokalni aplikaciji seznam vseh beležnic s časom zadnje spremembe za vsako beležnico posebej. Ta čas ustvari le strežnik in je čas, ko se je sprememba beležnice prenesla

na strežnik in ne, ko je bila beležnica napisana. Te čase ima shranjene tudi lokalna aplikacija in jih primerja s prejšnjimi. Če se ne ujemata, pomeni da je prišlo do spremembe beležnice. Takrat se s strežnika prenese novejša verzija.

4.3.2 Souporaba

Aplikacija omogoča souporabo beležnice z več uporabniki. Pri souporabi lahko pride do situacije, ko dva uporabnika sočasno pišeta v beležnico. Uporabnik, ki prvi shrani beležnico, jo normalno shrani. Pri drugemu pa pride do konflikta, saj ni spreminjal zadnje vsebine, ampak pred zadnjo. Zato se beležnica shrani kot nova. Ime pa dobi enako kot prejšnja s to razliko, da se na konec dopiše: "(nov)". Nova beležnica je prav tako v skupni uporabi z enakimi uporabniki kot prejšnja.

4.3.3 Zgodovina sprememb

Pri vsaki beležnici lahko vidimo zgodovino sprememb. Ker imajo spletni brskalniki omejeno količino podatkov, ki jih lahko shranimo v lokalno shrambo [19], zgodovina beležnic ni shranjena lokalno. Tako lahko gledamo zgodovino beležnic le, če imamo povezavo s strežnikom.

4.3.4 Grafični vmesnik

Spletna aplikacija je lahko prikazana na dva načina. Prvi način je bolj primeren za računalnike, drugi pa za mobilne naprave. Večina kode je kljub temu enaka za oba načina. Razlikujeta se le v dveh datotekah. Če je aplikacija prikazana na računalniku, je v HTML kodo vključena JavaScript datoteka `guipc.js`. Mobilne naprave pa imajo vključeno datoteko `guimob.js`. V teh datotekah je vsa koda, ki poskrbi, da je grafični vmesnik prilagojen napravi.

Poglavje 5

Opis Aplikacije

Za izvajanje spletne aplikacije, potrebujemo spletni brskalnik, ki podpira HTML5. Zaradi tega aplikacija deluje na vseh večjih sistemih, ki imajo novejši spletni brskalnik. Aplikacijo lahko vidite na sliki 5.1.

5.1 Grafični vmesnik

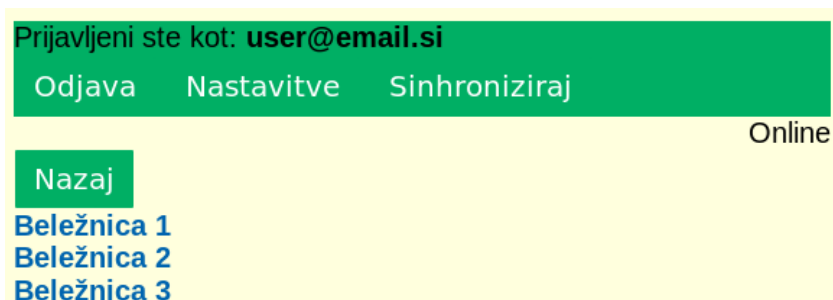
Aplikacija ima dva grafična vmesnika. Prvi grafični vmesnik je namenjen uporabi na namiznih računalnikih (slika 5.1), drugi pa za mobilne naprave, ki imajo ekrane občutljive na dotik (slika 5.2). Pravilen grafični vmesnik se nam dodeli samodejno glede na to, kakšno napravo uporabljamo. Grafična vmesnika se razlikujeta v tem, da ima mobilna verzija večje gumbе, ter da so elementi razporejeni tako, da je aplikacija bolj pregledna na ožjih napravah. Zaradi tega nima mobilna verzija prikazanega seznama beležnic ob strani, ampak ima gumb Seznam beležnic, ki prikaže vse naslove beležnic. Prikazane naslove lahko vidite na sliki 5.3. Prav tako je drugačno še okno, ki se odpre, ko se želimo registrirati, spremeniti nastavitve, ogledati zgodovino in dodati beležnico v souporabo. Pri mobilni verziji je to okno celozaslonsko, pri navadni pa je le na sredini ekrana.



Slika 5.1: Slika prikazuje aplikacijo za zapisovanje beležnic, ki smo jo izdelali.



Slika 5.2: Aplikacija z grafičnim vmesnikom za mobilne naprave.



Slika 5.3: Slika prikazuje seznam naslovov v grafičnem vmesniku za mobilne naprave.

5.2 Shranjevanje

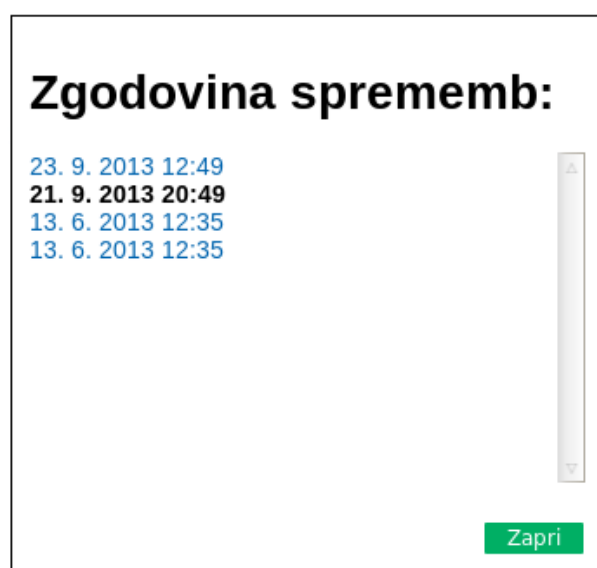
Da lahko uporabljamo aplikacijo in pišemo beležnice ni potrebno, da se prej registriramo in prijavimo. Beležnice se bodo shranjevale v lokalno shrambo tako, da se ne brišejo, tudi če zapremo spletni brskalnik. Vendar v tem primeru ne bomo imeli dostopa do njih z drugih naprav. Če se potem, ko že napišemo nekaj beležnic, prijavimo, se bodo te beležnice prenesle tudi na strežnik. Če imamo že kaj beležnic na strežniku, se prenesejo tudi v lokalno shrambo, tako da bomo do njih lahko dostopali brez povezave. Ko se odjavimo, se beležnice iz lokalne shrambe pobrišejo.

V aplikaciji je nastavljen čas osveževanja na 30 sekund. To je čas, ki določi, kako pogosto naj aplikacija preveri na strežniku, če je prišlo do kakšne spremembe. V primeru, da je do spremembe prišlo, se na odjemalca prenese ustrezne beležnice. Ta čas lahko tudi spremenimo v nastavitvah. Če nastavimo čas na 0 se aplikacija ne bo sinhronizirala s strežnikom samodejno. Lahko pa kadarkoli pritisnemo na gumb Sinhroniziraj, ki sproži sinhronizacijo. Kadar shranimo beležnico, se takoj za tem izvede tudi sinhronizacija, tako da je beležnica takoj tudi na strežniku.

5.3 Urejanje besedila

Besedilo beležnice je lahko v načinu ogleda (slika 5.1) ali urejanja (slika 5.2). V način urejanja pridemo tako, da kliknemo Nov ali Uredi za urejanje besedila. Takrat se nam prikaže gumb B s katerim lahko odebelimo tekst. Ko kliknemo gumb Shrani pridemo nazaj v način ogleda. Če vnesemo v besedilo URL, ga bo program zaznal in v načinu ogleda bomo lahko kliknili nanj in nam bo odprlo povezavo. Aplikacija ima za URL vse besede, ki se začnejo z `http://` ali `https://`.

Ko smo v načinu ogleda lahko aplikacija deluje tudi kot Wikipedija, tako da poišče v besedilu, če se mogoče kje nahaja enaka beseda ali zaporedje besed, kot je naslov kakšne beležnice. Če najde takšne besede, jih izpiše tako, da lahko na njih kliknemo in nam odpre besedilo te beležnice. Primer



Slika 5.4: Slika prikazuje zgodovino sprememb beležnice.

lahko vidite na sliki 5.1, kjer je Beležnica 2 v besedilu obarvana z modro barvo, ki pa za razliko od URL povezave ni podčrtana. Ko kliknemo na povezavo nam odpre beležnico 2.

5.4 Zgodovina sprememb

Na strežniku je shranjenih več verzij beležnic. To nam omogoči, da lahko pogledamo katero starejšo verzijo beležnice. To storimo tako, da kliknemo na gumb zgodovina. Nato se nam odpre okno, ki ga vidimo na sliki 5.4 in v njem so naštetih časi in datumi beležnic, katere imamo shranjene v zgodovini. Ko kliknemo na datum, se nam odpre beležnica iz tistega časa. Datum in čas beležnice, ki jo imamo trenutno odprto, je napisan odebeljeno in se ga ne da izbrati.



Slika 5.5: Slika prikazuje okno, v katerem je seznam uporabnikov, ki si delijo beležnico.

5.5 Souporaba

Posamezne beležnice imamo lahko tudi v souporabi z drugimi uporabniki. V skupno uporabo damo beležnico tako, da kliknemo gumb Souporaba. Ko se nam odpre novo okno, ki ga vidimo na sliki 5.5 lahko vanj vnesemo elektronski naslov uporabnika s katerim želimo deliti beležnico. V tem oknu lahko vidimo seznam vseh elektronskih naslovov uporabnikov s katerimi si beležnico delimo.

Souporabo zapustimo tako, da kliknemo Zapusti souporabo. S tem bomo zapustili souporabo, beležnica pa bo ostala enaka. Na ostale uporabnike v souporabi to ne vpliva. Beležnico v souporabi pa lahko tudi zberemo. Beležnica se v tem primeru ne bo zbrisala, ampak bo le ta uporabnik izstopil iz skupne uporabe. Šele ko zadnji uporabnik v skupni uporabi zberse beležnico, se ta zberse tudi na strežniku.

Poglavje 6

Sklepne ugotovitve

V prvem delu diplomske smo preučili in opisali nekatere spletne aplikacije za zapisovanje beležnic. Na podlagi teh informacij pa smo se odločili, katere funkcije naj ima naša aplikacija.

V drugem delu pa smo se posvetili izdelavi lastne spletne aplikacije za ustvarjanje beležnic. Opisali smo spletne tehnologije, s pomočjo katerih smo ustvarili aplikacijo, opisali smo izdelavo aplikacije ter na koncu še predstavili aplikacijo.

Menimo, da nam je uspelo doseči zastavljeni cilj, ki je bil izdelava aplikacije za pisanje beležnic, saj ima aplikacija vse načrtovane funkcije. To so, da deluje na več različnih napravah, omogoča dostop brez povezave, lahko gledamo zgodovino sprememb, ter da imamo lahko beležnice v souporabi z drugimi uporabniki. Aplikacijo pa se lahko uporablja tudi kot osebna wiki, saj samodejno ustvarja povezave med besedilom in naslovi beležnic.

Aplikacijo bi se dalo še izboljšati z novimi funkcijami, kot so združevanje beležnic v zvezke ter z možnostjo dodajanja slik in datotek k beležnicam. Potrebno bi bilo tudi izboljšati grafični vmesnik, tako da bi postal bolj atraktiven in nekoliko bolj intuitiven.

Literatura

- [1] Kristina Chodorow,
“MongoDB: The Definitive Guide”,
- [2] Peter MacIntyre, Brian Danchilla, Mladen Gogala,
“Pro PHP Programming”,

Spletni viri

- [3] Evernote (2013) Dostopno na:
<http://evernote.com/evernote/#>
- [4] Google Notebook (2013) Dostopno na:
<http://www.google.com/googlenotebook/faq.html>
- [5] HTML5 Web Storage (2013) Dostopno na:
http://www.w3schools.com/html/html5_webstorage.asp
- [6] jQuery (2013) Dostopno na:
<http://jquery.com/>
- [7] Mobile Detect (2013) Dostopno na:
<http://mobiledetect.net/>
- [8] RESTful Web services (2013) Dostopno na:
<http://www.ibm.com/developerworks/webservices/library/ws-restful/>

- [9] Uvod v JSON (2013) Dostopno na:
<http://json.org/json-sl.html>
- [10] Wikipedia: Apache HTTP Server (2013) Dostopno na:
http://en.wikipedia.org/wiki/Apache_HTTP_Server
- [11] Wikipedia: Aptana (2013) Dostopno na:
<http://en.wikipedia.org/wiki/Aptana>
- [12] Wikipedia: CSS (2013) Dostopno na:
<http://sl.wikipedia.org/wiki/CSS>
- [13] Wikipedia: Eclipse (software) (2013) Dostopno na:
[http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software))
- [14] Wikipedia: Evernote (2013) Dostopno na:
<http://en.wikipedia.org/wiki/Evernote>
- [15] Wikipedia: HTML (2013) Dostopno na:
<http://en.wikipedia.org/wiki/HTML>
- [16] Wikipedia: JavaScript (2013) Dostopno na:
<http://en.wikipedia.org/wiki/JavaScript>
- [17] Wikipedia: Keep (2013) Dostopno na:
http://en.wikipedia.org/wiki/Google_Keep
- [18] Wikipedia: Springpad (2013) Dostopno na:
<http://en.wikipedia.org/wiki/Springpad>
- [19] Wikipedia: Web storage (2013) Dostopno na:
http://en.wikipedia.org/wiki/Web_storage