

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Miha Rojko

**Spletna aplikacija za zajem in hrambo
posnetkov slovenskega govora**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2010



Št. naloge: 01632/2010

Datum: 15.02.2010

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

Kandidat: **MIHA ROJKO**

Naslov: **SPLETNA APLIKACIJA ZA ZAJEM IN HRAMBO POSNETKOV
SLOVENSKEGA GOVORA**

**WEB APPLICATION FOR ACQUISITION AND STORAGE OF
SLOVENIAN SPEECH RECORDINGS**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Na osnovi pregleda obstoječih orodij za razvoj dinamičnih spletnih aplikacij izberite najprimernejše orodje za izdelavo spletne aplikacije, ki bo omogočala zajem in hrambo posnetkov slovenskega govora. Aplikacija naj omogoča hitro in preprosto vnašanje slovenskih govornih besedil velikemu številu uporabnikov ter njihovo shranjevanje v relacijsko podatkovno bazo. Narejena naj bo tako, da bo uporabna pri računalniškem razpoznavanju govora. Razvijte in izdelajte programsko opremo vseh delov aplikacije, naredite njen preizkus in preverite pravilnost njenega delovanja.

Mentor:


prof. dr. Dušan Kodek



Dekan:


prof. dr. Franc Solina

IZJAVA O AVTORSTVU diplomskega dela

Spodaj podpisani **Miha Rojko**,
z vpisno številko **63040146**,

sem avtor diplomskega dela z naslovom:

Spletna aplikacija za zajem posnetkov slovenskega jezika

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom **prof. dr. Dušana Kodeka**
- 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 12.5.2010

Podpis avtorja:

Zahvala

Zahvaljujem se mentorju prof. dr. Dušanu Kodeku za vse nasvete in pomoč pri izdelavi diplomske naloge.

Zahvaljujem se tudi dr. Robertu Rozmanu, ki mi je s svojim poznavanjem področja razpoznavanja govora svetoval pri izdelavi snemalnika govora.

Posebna zahvala moji družini za vso skrb in podporo, ki mi jo je namenjala skozi celotno študijsko obdobje.

Hvala tudi vsem mojim študijskim kolegom, ki so mi velikokrat požrtvalno priskočili na pomoč.

Največja zahvala moji ženi Nuški za veliko mero potrpežljivosti in razumevanja.

Kazalo

Povzetek	1
1 Uvod	5
2 Vzpostavitev razvojnega okolja	9
2.1 Izbira spletne tehnologije	9
2.2 Namestitev spletnega strežnika, podatkovne baze in ostalih paketov	10
2.3 Izbira urejevalnika kode IDE	11
2.4 Pridobivanje potrebnih teoretičnih znanj	12
3 Načrt spletne aplikacije in podatkovne baze	13
3.1 Spletna aplikacija	14
3.1.1 Načrt	14
3.1.2 Datotečna hierarhija	16
3.1.3 Izvedba nekaterih delov spletne aplikacije	18
3.2 Podatkovna baza	22
4 Snemalnik govora	27
4.1 Shema in opis paketov aplikacije snemalnika	27
4.2 Izvedba nekaterih delov snemalnika	30
5 Primeri uporabe	33
5.1 Obiskovalec	33
5.2 Uporabnik	35
5.3 Skrbnik	37
6 Zaključek	39

Seznam slik	41
Seznam tabel	43
Literatura	45

Seznam uporabljenih kratic in simbolov

- CMS** – Sistem za upravljanje vsebin spletnih strani (angl. Content Management System)
- CRUD** – Štiri osnovne funkcije sistema za upravljanje vsebin, preko katerih ustvarja, bere, spreminja in briše vnose v tabelah podatkovne baze (angl. Create, Read, Update and Delete)
- CSS** – Stili, ki določajo izgled spletne strani (angl. Cascading Style Sheets)
- FTP** – Protokol za prenos datotek med računalniki (angl. File Transfer Protocol)
- HTML** – Označevalni jezik za oblikovanje večpredstavnostnih dokumentov, ki omogoča povezavo znotraj dokumenta ali med dokumenti (angl. Hypertext Markup Language)
- IDE** – Urejevalnik kode, ki nudi razvijalcu veliko število orodij, ki olajšajo razvoj aplikacij (angl. Integrated Development Environment)
- JNLP** – Protokol za zagon Java programov direktno iz spletnega brskalnika (angl. Java Network Launching Protocol)
- JRE** – Izvajalno okolje za programe napisne v programskem jeziku Java (angl. Java Runtime Environment)
- ORM** – Tehnika programiranja za pretvorbo podatkov med podatki relacijskih podatkovnih baz in objektno usmerjenimi programskimi jeziki (angl. Object-relational mapping)
- PHP** – Skriptni programski jezik, namenjen za izdelavo dinamičnih spletnih strani (angl. Hypertext Pre-Processor)
- XML** – Razširljivi označevalni jezik (angl. Extensible Markup Language)
- ZIP** – Tip datoteke, ki omogoča stiskanje podatkov

Povzetek

V diplomskem delu je opisana spletna aplikacija za zajem in hrambo posnetkov govora slovenskega jezika. Ta obiskovalcem omogoča, da preko snemalnika govora posnamejo ponujene stavke nekega besedila. Tega v postopku ustvarjanja novega besedila vnese registrirani uporabnik. Spletna aplikacija omogoča tudi dinamično spreminjanje vsebine preko sistema za upravljanje vsebin. Preko njega lahko skrbnik sistema spreminja informacije, ki se prikažejo na posameznih spletnih straneh. Vsi posnetki se zbirajo na strežniku od koder lahko vsak obiskovalec spletne strani le-te prosto prenese k sebi. Posnetki se na strežnik prenesejo s pomočjo snemalnika govora. Ta ponudi obiskovalcu stavke nekega besedila ter mu omogoča večkratno snemanje vsakega od teh. Aplikacija zajame tudi osnovne podatke o govorniku in mu omogoča spreminjanje jakosti naprave za zajem govora. Za implementacijo sistema za upravljanje vsebin je bila izbrana spletna tehnologija PHP, ki v povezavi z relacijsko podatkovno bazo MySQL tvorita ogrodje spletne aplikacije. Snemalnik govora je implementiran s programskim jezikom Java in je kot applet aplikacija vključen v spletno aplikacijo. Podani so tudi opisi nekaterih delov sistema, ki naj služijo kot začetna točka nadaljnjega razvoja spletne aplikacije. Na koncu so opisani še primeri uporabe spletne aplikacije za vrste uporabnikov le-te: obiskovalca, uporabnika in skrbnika.

Ključne besede:

zajem govora, hramba posnetkov, sistem za upravljanje vsebin

Abstract

This diploma thesis describes a web application for capturing and storing speech recordings of the Slovenian language. This application enables any user to record different sentences using the embedded speech recorder application. Sentences are part of the text which has been created by the registered user during the text-creation process. The web application also enables the administrator to manage the contents of the included sites using a content management system. All recordings are stored on a server and are accessible for free downloads. The recordings are uploaded from the speech recorder when the text-recording process is complete. The speech recorder offers the user several sentences which are part of the chosen text. The recorder enables the visitors to record the same statement multiple times. The application also captures basic information about the speaker and enables him to change the microphone level if needed. PHP technology was used to implement the content management system. With MySQL relational database it forms the web application framework. The speech recorder is implemented as a Java Applet application. Descriptions of certain parts of the system are also given – these can serve as a starting point for further development of the application. The thesis concludes with the examples that demonstrate how the application is used by different types of users: visitor, user and administrator.

Key words:

speech capture, storing speech recordings, content management system

1 Uvod

V današnjem času se vedno bolj kažejo potrebe po računalniških sistemih, ki znajo razpoznati govorjene besede in glede na te izvršiti ustrezne operacije. Takšnim sistemom pravimo *razpoznavalni sistemi*. V zadnjih letih se je uporabnost takšnih sistemov zelo povečala in danes jih srečujemo tudi že v vsakodnevnem življenju, npr. kot pomoč ljudem z okvaro sluha, hitro klicanje na prenosnih telefonih, naročanje kino vstopnic, narekovanje besedila računalniku,... Seveda pa je uporaba razpoznavalnih sistemov zelo razširjena tudi v vojski, zdravstvu, letalstvu in na drugih področjih [13].

Razpoznavanja govora je zapleten problem, ki se lahko rešuje na različne načine. Zato je bilo v zadnjih 50. letih razvitih veliko dopolnjujočih se modelov in iz njih izhajajočih algoritmov, ki se še danes uporabljajo pri reševanju problemov na področju razpoznavanja govora. Prva skupina takšnih modelov so stroji s končnim številom stanji, katerih predstavniki so deterministični in nedeterministični končni avtomati. V naslednjo skupino modelov spadajo regularne gramatike ter kontekstno-neodvisne gramatike, ki jih drugače poimenujemo kar sistemi pravil. Tretja skupina so predikativni računi, ki se uporabljajo za modeliranje semantike jezika. Četrta skupina modelov so verjetnostni modeli, kot so skriti modeli Markova (HMM), ki se danes uporabljajo na področjih razpoznavanja govora, razumevanja dialogov, sinteze govora ter drugod [4]. Prav slednji so danes najbolj prisotni pri procesih razpoznavanja govora.

Ko govorec izgovori neko besedo ali stavek, se govorni signal prenese do razpoznavalnika govora, ki na osnovi slovarja in gramatike jezika prepozna posamezne besede. Nato jezikovni analizator iz prepoznanih besed s pomočjo semantičnih pravil določi pomen, na osnovi katerega se ekspertni sistem odloči za ustrezno akcijo [8]. Vsak razpoznavalnik govora je potrebno najprej naučiti prepoznavati besede jezika. Zato se poleg slovarja in gramatike potrebujemo tudi akustični model jezika. Akustični model jezika je na osnovi posnetkov govora in prepisov govorjenih tekstov ustvarjena statistična predstavitev zvokov – fonemov, ki sestavljajo izgovorjene besede [1]. Zato je zelo pomembno, kako in na kakšnih učnih primerih se uči razpoznavalnik govora. Iz vsega omenjenega izhaja ugotovitev, da je za kakršnokoli razpoznavanje govora najprej potrebno imeti primerne in kvalitetne posnetke govora jezika, ki ga želimo razpoznavati.

Danes se na svetovnem spletu nahaja kar nekaj baz s posnetki številnih svetovnih jezikov, med katerimi pa slovenščine žal ne najdemo. Ena izmed takšnih baz je tudi VoxForge [15], ki v diplomskem delu služi kot zgled in vodilo pri razvoju spletne aplikacije za zajem in hrambo posnetkov slovenskega jezika. Nekatero od teh baz posnetkov so prosto dostopne, medtem ko so ostale plačljive ali pa jih velika podjetja držijo le zase. Zato je v tej diplomski nalogi obravnavana spletno aplikacijo, ki nudi zajem, hrambo in prost prenos posnetkov slovenskega jezika.

Prvi sklop diplomske naloge obravnava vzpostavitev delovnega okolja. Omenjene so spletne tehnologije PHP, Java Enterprise Edition in ASP.NET. Izbira vrste tehnologije je zelo pomembna odločitev, ki ji je vredno nameniti pozornost, saj v veliki meri definira naslednji korak. Ta je izbira in namestitev spletnega strežnika in relacijske podatkovne baze. Za realizacijo diplomske naloge je bila izbrana spletna tehnologija PHP. Navedeni so razlogi za in proti ter načini namestitve vseh potrebnih paketov za začetek razvoja spletne aplikacije. V nadaljevanju so omenjeni tudi razpoložljivih urejevalnikov kode IDE, ki lahko odločilno pripomorejo k lažjemu in hitrejšemu napredku. Tudi na tem področju danes obstaja precej različnih rešitev. Primerjamo predvsem urejevalnika Eclipse in NetBeans. Na koncu prvega sklopa so omenjena še znanja, ki so zaželeni za razvijalca spletne aplikacije. Predstavljenih je nekaj zanimivih in priporočljivih virov informacij.

Nadaljevanje diplomskega dela je osredotočeno na načrtovanje spletne aplikacije in podatkovne baze. Pred začetkom načrtovanja je potrebno poznati zahteve, ki jih mora sistem izpolnjevati. Zato so na začetku najprej opisane vse funkcionalnosti, ki jih mora izvrševati sistem. Današnje spletne aplikacije na svojih straneh in podstraneh nimajo več samo statične vsebine, temveč jih je vedno več takšnih, ki omogočajo dinamično spreminjanje vsebin. Takšni sistemi se imenujejo sistemi za upravljanje vsebin (CMS). V nadaljevanju so ponazorjeni sestavni deli takšnega sistema in zaporedja akcij, ki se izvršijo na posameznih straneh in podstraneh. Ko je načrt končan, sledi implementacije posameznih delov. Zaradi morebitne uporabe in razširitve funkcionalnosti sistema je podan tudi opis datotečne hierarhije sistema. Za konec sklopa je opisan še konceptualni model podatkovne baze in v njej vsebovane tabele.

Naslednji sklop diplomskega dela obravnava osrednji sestavni del spletne aplikacije za zajem in hrambo posnetkov slovenskega jezika - snemalnik govora. Na začetku predstavimo tehnologijo Java Applet, ki omogoča uporabo sistemskih virov na strani uporabnika. Omenjamo tudi prednosti

uporabe JNLP oziroma Java Web Start tehnologije. Namen aplikacije je omogočati uporabniku, da na najlažji možni način posname ponujena besedila in jih naloži na strežnik, kjer se posnetki shranijo. Zato v nadaljevanju načrtujemo zgradbo takšne aplikacije, v kateri so predstavljeni moduli (paketi) in njihovo medsebojno povezovanje. Za lažji nadaljnji razvoj je opisan tudi način delovanja določenih delov snemalnika. Za konec sklopa je opisana še izvedba nekaterih delov in postopkov snemalnika govora.

V predzadnjem delu diplomske naloge podamo nekaj primerov uporabe spletne aplikacije. Primeri uporabe so prikazani za vse vrste obiskovalcev spletne strani: za navadne obiskovalce, ki lahko le snemajo in prenašajo ponujene tekste, uporabnike, ki poleg snemanja tudi dodajajo tekste, kot tudi za skrbnike sistema, ki s pomočjo sistema za spreminjanje vsebine spreminjajo vsebino strani in podstrani. Navodila so podana v obliki slik zaslonskih mask in s spremljajočimi tekstovnimi navodili.

V zaključnem sklopu diplomskega dela ovrednotimo vse zastavljene zahteve, ki jih mora izpolnjevati sistem za zajem in hrambo posnetkov slovenskega jezika ter navedemo še nekatere možnosti za izboljšave.

2 Vzpostavitev razvojnega okolja

V poglavju je najprej opredeljena tehnologija za implementacijo spletne aplikacije. Glede na izbiro namestimo ustrezno različico strežnika in podatkovne baze. Pred začetkom razvoja izberemo tudi najprimernejši urejevalnik kode IDE in navedemo nekaj virov informacij, ki ponudijo bralcu dovolj teoretičnih osnov, da lahko prične z načrtovanjem in implementacijo spletne aplikacije.

2.1 Izbira spletne tehnologije

Vsak razvijalec, ki želi razviti dinamične spletne aplikacije, se lahko danes odloča med številnimi spletnimi tehnologijami. Med njimi lahko največkrat zasledimo odprtokodna PHP in Java Enterprise Edition (Java EE), nosilec katere je Sun Microsystems ter plačljivi Microsoftov ASP.NET. Ob brskanju po spletu lahko naletimo na dokaj deljena mnenja o tem, katera tehnologija je najprimernejša. Vsaka od njih ima namreč svoje prednosti in slabosti.

Med kriteriji, po katerih se ocenjuje ustreznost posamezne spletne tehnologije se največkrat pojavijo [16]:

- vnaprejšnje in dinamično prevajanje kode, ki omogoča hitrejše izvajanje spletne aplikacije,
- če je jezik skriptni, se čas izvajanja programa podaljša, saj je potrebno jezik interpretirati,
- možnost objektnega programiranja omogoča ponovno uporabo že napisane kode in zmanjša čas razvoja spletne aplikacije,
- podprti jeziki, v katerih lahko implementiramo spletno aplikacijo,
- odprtokodnost.

V nadaljevanju v tabeli 2.1 prikazujemo medsebojno primerjavo zgoraj omenjenih spletnih tehnologij po pravkar omenjenih kriterijih.

Kriterij	.NET	Java	PHP
Prevajanje kode	Da	Da	Ne – možnost namestitve razširitev, ki to omogočajo
Skriptni jezik	Ne	Ne	Da
Možnost objektnega programiranja	Da	Da	Da
Podprti jeziki za implementacijo	C++, C#, VB.NET, J#,...	Java	PHP
Odprtokodnost	Ne	Da	Da

Tabela 2.1: Prikaz najpomembnejših kriterijev primerjave spletnih tehnologij.

Kot je bilo omenjeno že v uvodu, smo se v primeru diplomskega dela odločili za spletno tehnologijo PHP. Kljub v tabeli 2.1 omenjenim pomanjkljivostim ima kar nekaj prednosti, ki so pretehtale [6]:

- prosto-dostopnost,
- preprosta uporaba in hitro učenje,
- izvaja se lahko na večini operacijskih sistemov (Windows, Linux, Mac OS in na večini Unix sistemov),
- močna tehnična podpora in veliko primerov uporabe,
- varnost,
- podpira uporabo velikega števila podatkovnih baz,
- prilagodljivost.

2.2 Namestitev spletnega strežnika, podatkovne baze in ostalih paketov

Z odločitvijo za neko spletno tehnologijo je povezana tudi izbira strežnika. Danes je zelo razširjen odprtokodni strežnik Apache [2], ki nastaja pod okriljem Apache Software Foundation.

Poleg strežnika Apache potrebujemo za razvoj spletne aplikacije še druge pakete, kot so PHP [12], podatkovna baza MySQL [10] ter FTP

strežnik FileZilla [5]. Postopkov za nameščanje in konfiguracijo razvojnega okolja je več. Vsak, ki ima namen vzpostaviti razvojno okolje mora natančno vedeti, kako želi nastaviti vse nastavitve paketov, da bodo delovali pravilno. To je velikokrat dolgotrajen postopek, ki zahteva poglobljeno poznavanje paketov, njihovih zmožnosti in zahtev. Zaradi teh razlogov je bilo razvojno okolje za spletno aplikacijo vzpostavljeno z namestitvenim programom XAMPP [18] za operacijski sistem Windows.

XAMPP vsebuje pakete Apache, MySQL, PHP, phpMyAdmin, Perl, FileZilla FTP Server in Mercury Mail Transport System. Vsi ti paketi se po končani namestitvi nahajajo v eni sami mapi. Lokacijo te si lahko izberemo po svojih lastnih željah. Znotraj te mape so poleg priročnih skript za zagon in zaustavitev XAMPP aplikacije tudi mape nameščenih paketov. Na tem mestu je potrebno omeniti tudi, da XAMPP ob nameščanju paketov le-te nastavi tako, da so takoj po namestitvi razvojnega okolja pripravljeni za delo. Z drugimi besedami razvijalcu ni potrebna dolgotrajna in velikokrat mukotrpa nastavljanje vsakega paketa posebej. Ko se namestitve zaključi, ima razvijalec vse pravice nad vsemi datotekami in mapami. Če želimo pravice spremeniti, moramo to narediti naknadno. Zato postopek namestitve razvojnega okolja s pomočjo XAMPP namestitvene aplikacije ni primeren za produkcijska okolja.

2.3 Izbira urejevalnika kode IDE

Danes lahko na spletu zasledimo veliko število urejevalnikov IDE. IDE (Integrated Development Environment) je programska aplikacija, ki nudi razvijalcu celovito podporo za hitrejše, natančnejše in lažje razvijanje projektov. Nekateri izmed njih ponujajo le pomoč pri pisanju kode s sintaktično analizo, avtomatičnim dopolnjevanjem (code completion), označevanjem rezerviranih besed (syntax highlight), medtem ko drugi omogočajo še mnogo več, kot na primer:

- avtomatično prevajanje in končno izgradnjo projekta,
- postopno izvajanje kode (debugging),
- vizualno urejanje grafičnega vmesnika.

Na tem mestu omenimo nekatere urejevalnike, ki nam lahko pomagajo pri razvoju PHP projektov. To so:

- NetBeans,
- Eclipse PDT,
- Zend Studio,
- Notepad++,
- Adobe Dreamwaver,
- UltraEdit

in drugi. Med vsemi temi smo izbrali urejevalnik NetBeans.

NetBeans [11] je odprtokodni urejevalnik IDE, ki je produkt Sun Microsystems. Osnovan je na programskem jeziku Java in vsebuje podporo za implementacijo različnih tehnologij: Java, JavaScript, PHP, Python, Ruby, Groovy, C, C++,... Poleg tega vsebuje odličen grafični urejevalnik, ki nam zelo poenostavi razvoj grafičnega vmesnika snemalne aplikacije, ki jo omenjamo v 4. poglavju.

2.4 Pridobivanje potrebnih teoretičnih znanj

Za konec poglavja omenimo še, kje lahko razvijalec, ki želi izdelati spletno aplikacijo s tehnologijo PHP, dobi začetne informacije, ki opisujejo in pojasnjujejo osnovne koncepte razvijanja takšnega izdelka.

Eden od takšnih virov informacij je knjiga za začetnike o PHP in MySQL tehnologiji [6], ki v začetnih poglavjih zelo pregledno in razumljivo predstavi omenjeni tehnologij, ki skupaj tvorijo ogrodje spletne aplikacije.

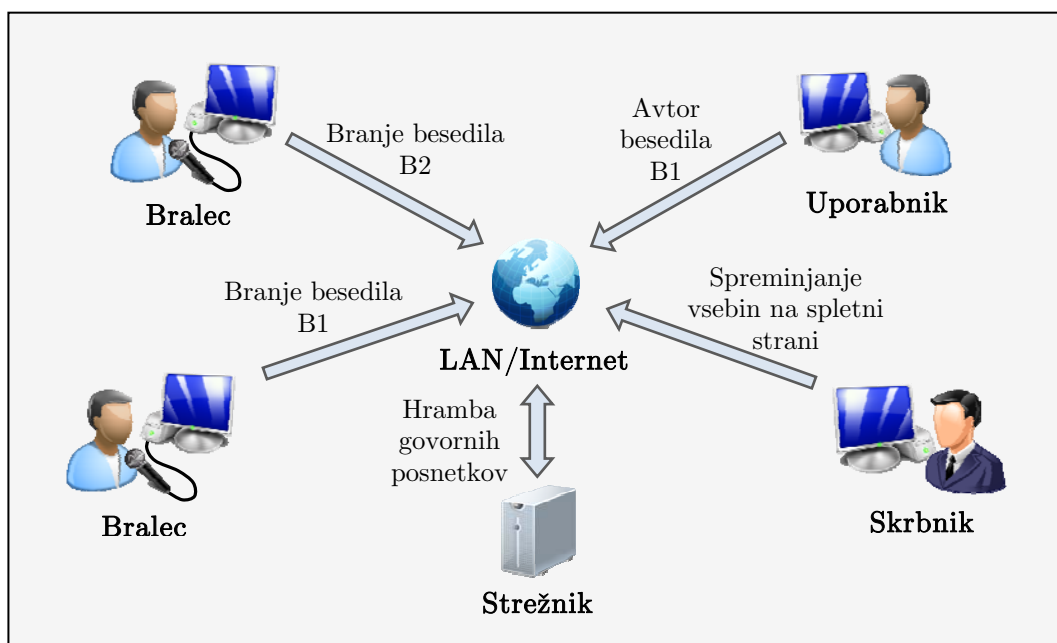
Za nadaljnje informacije in nasvete je odličen vir domača stran PHP [12], kjer lahko razvijalec najde dokumentacijo vseh razredov in funkcij. Poleg večine od njih so podani še praktični primeri, ki zelo olajšajo postopek razvoja spletne aplikacije.

Zelo priporočljiv začetni (velikokrat pa tudi stalni) vir informacij je spletna stran W3School Online [17], kjer lahko razvijalec izve, katere tehnologije se trenutno uporabljajo na določenem področju in vsako od teh na osnovi velikega števila primerov podrobneje spozna.

Na koncu omenimo še spletno stran Lynda.com [9], ki pa z razliko od zgornjih virov ponuja plačljiva video predavanja. Ta so zastavljena tako, da poslušalec ob poslušanju in gledanju navodil izdela projekt, ki mu lahko služi kot začetna točka razvoja svoje spletne aplikacije.

3 Načrt spletne aplikacije in podatkovne baze

Pred začetkom načrtovanja sistema za upravljanje vsebine (angl. CMS) opredelimo še shemo sistema. Sistem sestavljajo računalniki in strežnik, ki so priključeni v lokalno omrežje (angl. LAN) oziroma internet. Računalniki morajo omogočati priklop mikrofona, preko katerega se zajema govor. Shema sistema za zajem govora je predstavljena na sliki 3.1.



Slika 3.1: Shema sistema za zajem in hrambo posnetkov slovenskega jezika. Uporabnik predstavlja obiskovalca, ki ustvari besedilo B1 z namenom pridobiti posnetke. Bralec je obiskovalec, ki ta besedila snema. Posnetki se nato shranijo na spletnem strežniku. Skrbnik preko sistema za spreminjanje vsebine upravlja vsebino, ki jo vidijo obiskovalci.

Kot že omenjeno, so osnovne zahteve, ki jih mora spletna aplikacija izpolnjevati, snemanje, hramba in možnost prenosa govornih posnetkov. K tem zahtevam dodajamo še:

- spletna aplikacija naj omogoča skrbniku, da lahko ta ustvari, bere, spreminja in briše zavihke strani in njihove vsebine (CRUD – Create, Read, Update and Delete).
- spletna aplikacija mora omogočati uporabniku, da lahko ta hitro in enostavno vnaša tista besedila, za katera želi, da se prikazujejo v snemalniku govora,
- v primeru, da se je uporabnik v postopku ustvarjanja besedila zmotil ali pa želi, da se kakšen stavek besedila izpusti ali na novo doda, mora spletna aplikacija omogočati tudi naknadno urejanje besedil,
- v primeru velikega števila novih besedil naj spletna aplikacija ponudi bralcu tiste, ki so bile ustvarjene zadnje.

3.1 Spletna aplikacija

3.1.1 Načrt

Slika 3.2 predstavlja načrt spletne aplikacije za zajem in hrambo posnetkov. Na njem vsak kvadrat ponazarja stanje sistema, v katerem se obiskovalec oziroma skrbnik nahaja. Vsako stanje lahko razumemo kot eno ali več spletnih strani, ki morajo izpolnjevati vse v alinejah našete zahteve.

Javne strani so vse, do katerih lahko dostopa vsakdo, brez predhodne prijave v sistem. Komponente teh strani so:

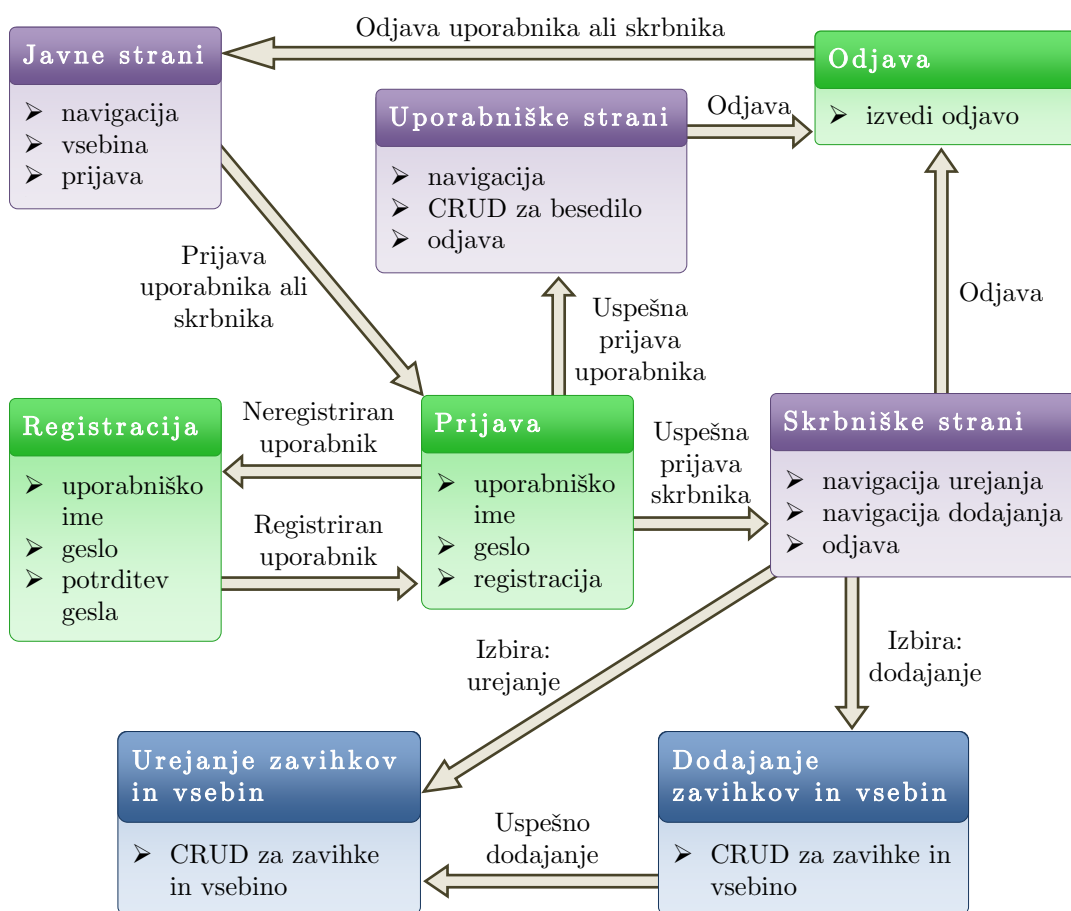
- *navigacijski meni*, ki preusmeri obiskovalca na druge javne strani,
- *vsebina strani*, ki lahko vsebuje več naslovov,
- *prijava v sistem*.

Uporabniške strani omogočajo uporabniku, da dodaja, spreminja ali briše svoja besedila. Pred tem se mora uporabnik uspešno prijaviti v sistem. Komponente, ki sestavljajo uporabniške strani so:

- *navigacijski meni*, ki omogoča uporabniku izbor zelenega besedila,
- *CRUD za besedilo*,
- *odjava uporabnika iz sistema*.

Skrbniške strani predstavljajo jedro sistema za upravljanje vsebine. Do njih lahko s predhodno prijavo dostopa le skrbnik. Komponente teh strani so naslednje:

- *navigacija urejanja*, ki skrbniku omogoča izbiro posameznega zavihka oziroma vsebine znotraj le-tega,
- *navigacija dodajanja*, s katero skrbnik pove sistemu ali želi dodati nov zavihek ali novo vsebino,
- *odjava* skrbnika iz sistema.



Slika 3.2: Načrt spletne aplikacije za zajem in hrambo posnetkov. Kvadrati predstavljajo stanje sistema, v katerem se nahaja obiskovalec, povezave pa označujejo prehode iz enega stanja sistema v drugega ob uspešni izvedbi akcij komponent.

Strani za urejanje oziroma dodajanje zavihkov in vsebin omogočajo skrbniku, da preko CRUD-a za zavihke in vsebino spreminja pripadajoče informacije. Seveda mora biti skrbnik pred tem uspešno prijavljen v sistemu.








Uporabnik ali skrbnik se v sistem prijavljata preko strani za prijavo. V postopku prijave vneseta svoje uporabniško ime in geslo, sistem pa ju na osnovi tega preusmeri na uporabniške ali pa na skrbniške strani.

Stran za odjavo iz sistema preko komponente za *odjavo* uporabnika ali skrbnika uniči pripadajočo sejo in izvede preusmeritev na interne strani.

















Stran za registracijo omogoča kreiranje novega uporabnika, ki dobi pravice za kreiranje novih besedil. Sam postopek registracije je preprost. Bodoči uporabnik vnese želeno uporabniško ime in geslo, nato pa geslo potrdi s ponovnim vnosom le tega.

3.1.2 Datotečna hierarhija

Datotečna hierarhija spletne aplikacije predstavlja razporeditev map - paketov in datotek spletne aplikacije. V primeru nadaljnjega razvoja sistema sliki 3.3 in 3.4 predstavljata začetno točko za spoznavanje z načinom implementacije .








	applet	Mapa aplikacije snemalnika govora.
	cron	V njej se nahaja skripta za generiranje datotek ZIP, ki si jih obiskovalci lahko prenesejo k sebi.
	domains	Mapa hrani vse posnetke govorov.
	downloads	Mapa, v kateri se nahajajo vse datoteke ZIP, ki si jih lahko obiskovalci prenesejo k sebi.
	flash	Mapa z animacijo Flash.
	images	V njej so shranjeni vsi slikovni gradniki spletnih strani in podstrani.
	includes	Mapa, ki vsebuje vse večkrat uporabljene skripte. Natančnejši opis sledi v nadaljevanju.

Slika 3.3: Datotečna hierarhija - 1.del.

 javascripts	Mapa z vsemi JavaScript funkcijami.
 manager	Mapa, v kateri se nahajajo skrbniške strani, preko katerih skrbnik spreminja vsebino spletnih strani.
 stylesheets	Mapa z CSS datotekami.
 upload	Začasna mapa, v katero se prenesejo obiskovalčevi posnetki iz snemalnika govora.
 applet.php Tip: Datoteka PHP	Javna snemalna stran, kjer lahko obiskovalec posname svoje branje besedil.
 config.php Tip: Datoteka PHP	Nastavitvena skripta, ki se kliče na začetku vsake strani.
 download.php Tip: Datoteka PHP	Javna stran, od koder lahko obiskovalec prenese .zip datoteko s posnetki.
 editText.php Tip: Datoteka PHP	Uporabniška stran, kjer lahko uporabnik ureja že ustvarjena besedila ali pa jih izbriše.
 finish_upload.php Tip: Datoteka PHP	Skripta, ki poskrbi, da se poslani posnetki prenesejo v mapo <i>domains</i> , podatki o posnetkih pa v bazo podatkov.
 index.php Tip: Datoteka PHP	Naslovna spletna stran sistema.
 login.php Tip: Datoteka PHP	Javna stran za vpis obiskovalca v sistem.
 logout.php Tip: Datoteka PHP	Javna stran za odjavo uporabnika iz sistema.
 newText.php Tip: Datoteka PHP	Uporabniška stran, na kateri lahko uporabnik ustvari novo besedilo.
 read.php Tip: Datoteka PHP	Javna stran, na kateri lahko obiskovalec izbere tisto besedilo, ki ga želi posneti.
 register.php Tip: Datoteka PHP	Javna stran, kjer se lahko obiskovalec registrira kot novi uporabnik.
 text.php Tip: Datoteka PHP	Domača stran uporabnika, kjer lahko vidi vsa svoja besedila.

Slika 3.4: Datotečna hierarhija - 2.del.

Sliki 3.3 in 3.4 ponazarjata hierarhijo in vlogo map spletne aplikacije. Omenimo posebej še mapo *includes*, ki je lokacija vseh večkrat uporabljenih skript. Te se kličejo iz vseh strani in podstrani sistema, zato slika 3.5 prikazuje njihovo razporeditev in na kratko opiše naloge, ki jih izpolnjujejo v sistemu.

 database	Paket omogoča pridobivanje podatkov iz podatkovne baze na osnovi ORM (angl. object-relational mapping).
 footer.php Tip: Datoteka PHP	V tej skripti je shranjen zaključni del spletne strani – informacije o pravicah in avtorju.
 functions.php Tip: Datoteka PHP	Skripta vsebuje vse večkrat uporabljene funkcije, ki so skupne vsem stranem in podstranem.
 header.php Tip: Datoteka PHP	V tej skripti je shranjen začetni del vsake spletne strani – meta podatki, css, JavaScript, flash animacija.
 navigation.php Tip: Datoteka PHP	Skripta vsebuje funkcije, ki generirajo navigacijske komponente - menije.
 session.php Tip: Datoteka PHP	Skripta poskrbi, da se ob nalaganju strani nastavi seja, v katero se shranjujejo pomembnejše informacije uporabnika.
 site_init.php Tip: Datoteka PHP	Skripta poskrbi, da se vključijo v stran vse skripte, ki so skupne vsem stranem in podstranem. Vzpostavi tudi povezavo s podatkovno bazo.

Slika 3.5: Prikaz notranje zgradbe paketa *includes* in opis nalog posameznih skript.

3.1.3 Izvedba nekaterih delov spletne aplikacije

Za zaključek poglavja 3. *Spletna aplikacija* opišimo še nekaj pomembnejših delov sistema in pojasnimo način implementacije le teh.

Dostop do podatkov v podatkovni bazi je implementiran s tehniko ORM. ORM (Object-relational mapping) je realiziran kot skupek PHP objektov, kjer je vsak objekt abstrakcija tabele v podatkovni bazi. Objekt vsebuje njen opis, vključno z njenim imenom v bazi, vsemi imeni atributov, njihovimi tipi in seznamom primarnih ključev. Vsak objekt deduje od *table.php*, kjer so implementirane metode za kreiranje, branje, spreminjanje in brisanje posameznih vnosov tabele v podatkovni bazi. Vse poizvedbe se nahajajo v *Querys.php*. Ta poskrbi, da se ob njenem

kreiranju (v skripti *site_init.php*) vzpostavi povezava s podatkovno bazo in inicializirajo vsi objekti ORM.

Vse spletne strani se dinamično generirajo s tehnologijo PHP glede na zahtevo, ki jo pošlje obiskovalec. Osnovni postopek prikazuje slika 3.6. Na

```
<?php

// vključevanje vseh skript, ki jih potrebuje spletna stran
require_once("config.php");
require_once(INCLUDES_DIR."/session.php");
require_once(INCLUDES_DIR."/site_init.php");
require_once(INCLUDES_DIR."/header.php");

// klicanje funkcije iz skripte header.php, ki zgenerira glavo
// spletne strani
echo generate_header("Posnetki slovenskega jezika",
    "stylesheets/public.css", "", "flash/header.swf");

// meni
echo public_navigation($sel_subject); ?>

<!-- glavni del strani -->
<div id="borders-mid">
    <div id="borders-up">
        <div id="borders-down">

            <!-- vsebina strani -->

        </div>
    </div>
</div>

<!-- zaključek strani -->
<?php require_once(INCLUDES_DIR."/footer.php"); ?>
```

Slika 3.6: Prikaz osnovnega postopka generiranja spletne strani.

začetku generiranja spletne strani se najprej zažene skripta *config.php*. Ta definira nekatere spremenljivke, med drugim tudi *INCLUDES_DIR*, ki predstavlja absolutno pot do mape *includes*. Iz te mape se v nadaljevanju zaženejo skripte *session.php*, *site_init.php* in *header.php*. Njihove vloge v

sistemu so razložene na sliki 3.5. Zatem se pokliče metoda *generate_header()*, ki sprejme kot argumente:

- naslov spletne strani,
- seznam poti do datotek CSS (Cascading Style Sheets),
- seznam poti do datotek JavaScript,
- pot do animacijske datoteke Flash;

in vrne HTML niz. Nato se pokliče metoda za generiranje menija *generate_menu*. Metoda kot argument sprejme spremenljivko, v kateri je shranjena vrednost zavihka, ki ga je uporabnik izbral. Na tak način metoda pravilno označi izbrani zavihek v meniju, ki se bo prikazal obiskovalcu. Sledi glavni del dokumenta HTML, to je vsebina. Ta se razlikuje od strani do strani. Za konec se pokliče še skripta *footer.php*, ki zaključi datoteko HTML z informacijami o avtorju, letu izdelave in avtorskih pravicah.

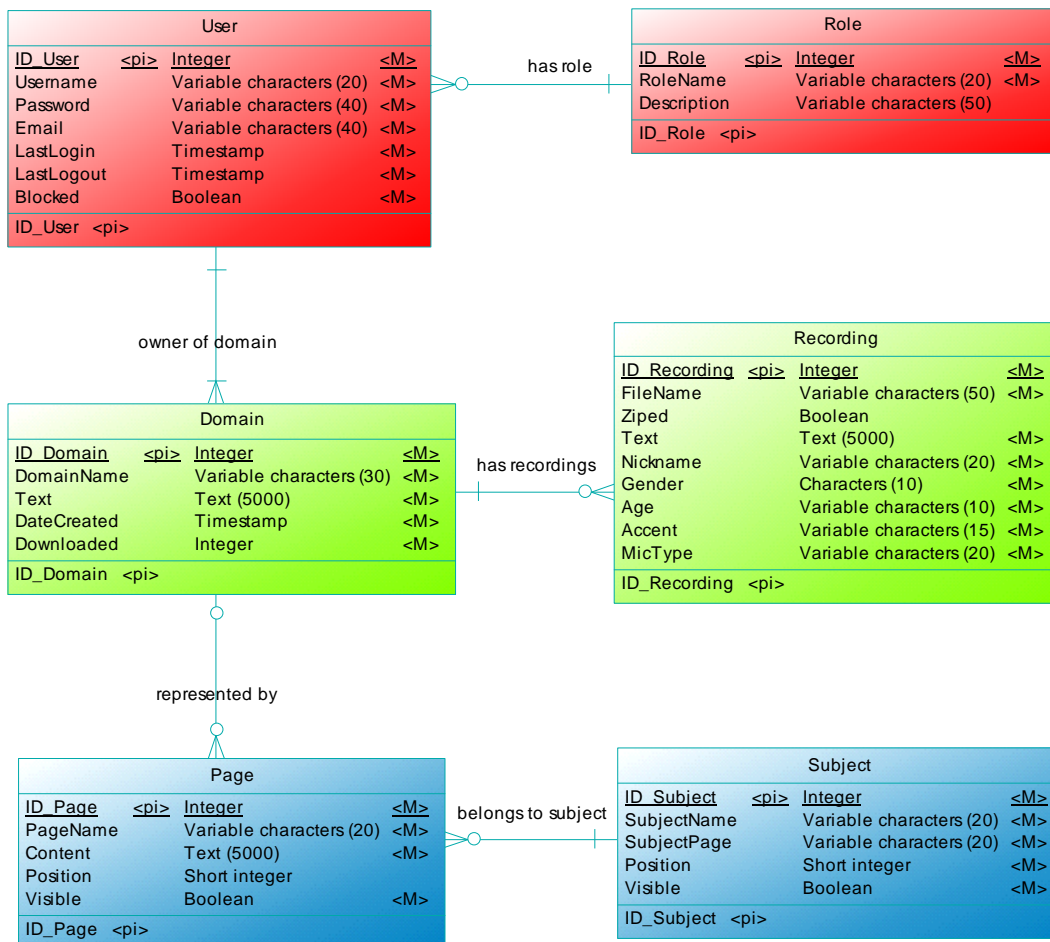
Postopek shranjevanja poslanih posnetkov in njihovih opisnih informacij (v obliki datoteke XML) na strežnik oziroma podatkovno bazo se prične s prenosom preko protokola FTP v mapo *upload*. Za vsako narekovanje (sestavlja ga več posnetkov), ki ga obiskovalec odpošlje iz snemalnika govora, se v omenjeni mapi ustvari začasna mapa z unikatnim imenom. Ko snemalnik govora odpošlje vse posnetke in so ti uspešno shranjeni v mapo *upload/»unikatno_ime«*, se pokliče skripta *finish_upload.php*, ki za vsak posnetek v podatkovno bazo vnese podatke o govorniku, ime posnetka in prepis narekovanega besedila.

Za pripravo datotek ZIP, ki jih obiskovalci lahko prenesejo k sebi, skrbi skripta *cron.php*, ki se nahaja v mapi *chron*. Njena naloga je, da v podatkovni bazi preveri, kateri posnetki posameznega besedila še niso bili dodani v datoteko ZIP. Nato vanjo doda vse še nedodane posnetke ter v datoteko XML, katere primer je na sliki 3.7, doda ustrezne informacije o še nedodanih posnetkih. Po končanem izvajanju se v mapi *domains* nahajajo osvežene datoteke ZIP. Vsako od njih sestavljajo od 0 navzgor oštevilčeni posnetki in datoteka XML, ki vsebuje informacije za vse omenjene posnetke.

```
<?xml version="1.0" encoding="UTF-8"?>
<slovenian_speech_recordings>
  <recording>
    <nickname>Miha</nickname>
    <gender>Moški</gender>
    <age>19 - 54</age>
    <accent>Ljubljansko</accent>
    <mic_type>Vgrajeni</mic_type>
    <text>Danes je primeren dan za vožnjo s kolesom.</text>
    <filename>0.wav</filename>
  </recording>
  <recording>
    <nickname>Miha</nickname>
    <gender>Moški</gender>
    <age>19 - 54</age>
    <accent>Ljubljansko</accent>
    <mic_type>Vgrajeni</mic_type>
    <text>Udeležba na čistilni akciji je rekordna.</text>
    <filename>1.wav</filename>
  </recording>
  <recording>
    <nickname>Mateja</nickname>
    <gender>Ženski</gender>
    <age>19 - 54</age>
    <accent>Štajersko</accent>
    <mic_type>Namizni</mic_type>
    <text>Največje jezero v Sloveniji je cerkniško jezero.</text>
    <filename>2.wav</filename>
  </recording>
</slovenian_speech_recordings>
```

Slika 3.7: Primer datoteke XML, v kateri so za vsak posnetek zapisani podatki o govorniku, besedilo, ki ga je bralec posnel, in ime datoteke s posnetkom.

3.2 Podatkovna baza



Slika 3.8: Konceptualni model podatkovne baze sistema, ki omogoča zajem, hrambo in prenos govornih posnetkov. Rdeča barva označuje tabeli, ki opisujeta uporabnika in njegovo vlogo v sistemu, zelena označuje tabeli, ki hranita podatke o besedilu in njenih pripadajočih posnetkih, modra pa označuje tabeli v katerih se hranijo podatki o vsebini spletnih strani.

V nadaljevanju je podan natančnejši opis zgornjih tabel. Za vsako od njih je naveden tekstovni opis tako tabele kot tudi vseh njenih atributov.

a) User

Tabela predstavlja v sistem registriranega uporabnika, ki lahko ustvari nova besedila.

Ime atributa	Podatkovni tip	Obvezen	Opis
ID_User	Integer	✓	Šifra uporabnika.
Username	Varchar (20)	✓	Uporabniško ime.
Password	Varchar (20)	✓	Uporabniško geslo.
Email	Varchar (20)	✓	Elektronski naslov uporabnika.
LastLogin	Timestamp	✓	Datume in čas zadnje prijave.
LastLogout	Timestamp	✓	Datum in čas zadnje odjave.
Blocked	Boolean	✓	Stanje uporabnika.

Tabela 3.1: Tabela uporabnika.

b) Role

Tabela predstavlja šifrant vseh vlog uporabnikov.

Ime atributa	Podatkovni tip	Obvezen	Opis
ID_Role	Integer	✓	Šifra vloge.
Rolename	Varchar (20)	✓	Ime vloge.
Description	Varchar (50)	✗	Opis vloge v sistemu.

Tabela 3.2: Tabela vloge.

c) Domain

Tabela predstavlja besedilo, ki ga ustvari uporabnik z namenom, da lahko obiskovalci preko snemalnika govora posnamejo to besedilo.

Ime atributa	Podatkovni tip	Obvezen	Opis
ID_Domain	Integer	✓	Šifra besedila.
Domainname	Varchar (20)	✓	Naslov besedila.
Text	Text (5000)	✓	Stavki besedila.
Datacreated	Timestamp	✓	Datum nastanka besedila.
Downloaded	Integer	✓	Število prenosov te domene.

Tabela 3.3: Tabela besedila.

d) Recording

Tabela predstavlja posamezen posnetek – to je stavek, ki ga bralec posname s snemalnikom govora in ga prenese na strežnik.

Ime atributa	Podatkovni tip	Obvezen	Opis
ID_Recording	Integer	✓	Šifra posnetka.
Filename	Varchar (50)	✓	Ime datoteke, s posnetkom.
Ziped	Boolean	✗	Ali je posnetek že v datoteki ZIP (za prenos)?
Text	Text (5000)	✓	Prepis stavka izgovorjenega v posnetku.
Nickname	Varchar (20)	✓	Naziv bralca.
Gender	Varchar (20)	✓	Spol bralca.
Age	Varchar (10)	✓	Starost bralca.
Accent	Varchar (15)	✓	Naglas bralca.
MicType	Varchar (20)	✓	Tip naprave za zajem govora.

Tabela 3.4: Tabela posnetka.

e) Page

Tabela predstavlja naslove znotraj neke spletne strani ali podstrani.

Ime atributa	Podatkovni tip	Obvezen	Opis
ID_Page	Integer	✓	Šifra naslova na strani.
Pagename	Varchar (20)	✓	Ime naslov.
Content	Text (5000)	✓	Vsebina.
Position	Short integer	✗	Pozicija naslova v strani.
Visible	Boolean	✓	Vidnost naslova.

Tabela 3.5: Tabela naslova znotraj strani ali podstrani.

f) Subject

Tabela predstavlja posamezno spletno stran aplikacije.

Ime atributa	Podatkovni tip	Obvezen	Opis
ID_Subject	Integer	✓	Šifra spletne strani.
Subjectname	Varchar (20)	✓	Opis spletne strani, ki se pojavi v zavihkih.
SubjectPage	Varchar (20)	✓	Ime datoteke PHP, ki generira to spletno stran.
Position	Short integer	✓	Pozicija spletne strani – lokacija zavihka.
Visible	Boolean	✓	Vidnost spletne strani..

Tabela 3.6: Tabela posamezne spletne strani.

4 Snemalnik govora

Snemalnik govora je implementiran kot applet. Applet je posebna vrsta aplikacije v programskem jeziku Java, ki se lahko izvaja v brskalnikih, če imajo ti nameščeno izvajalno okolje JRE (Java Runtime Environment). Ker je applet vgrajen v spletno stran, je njegovo izvajanje omejeno z kontekstom brskalnika [14], kar pomeni, da ima omejene pravice pri dostopu do sistemskih virov.

Rešitev te omejitve je uporaba tehnologij Web Start in Java Network Launching Protocol (JNLP). Pri uporabi prve se applet ne izvaja več s pravicami, ki veljajo za spletne strani, saj se ne izvaja več znotraj brskalnika. Java Web Start omogoča tudi avtomatični prenos in namestitvev ustreznega izvajalnega okolja JRE [7].

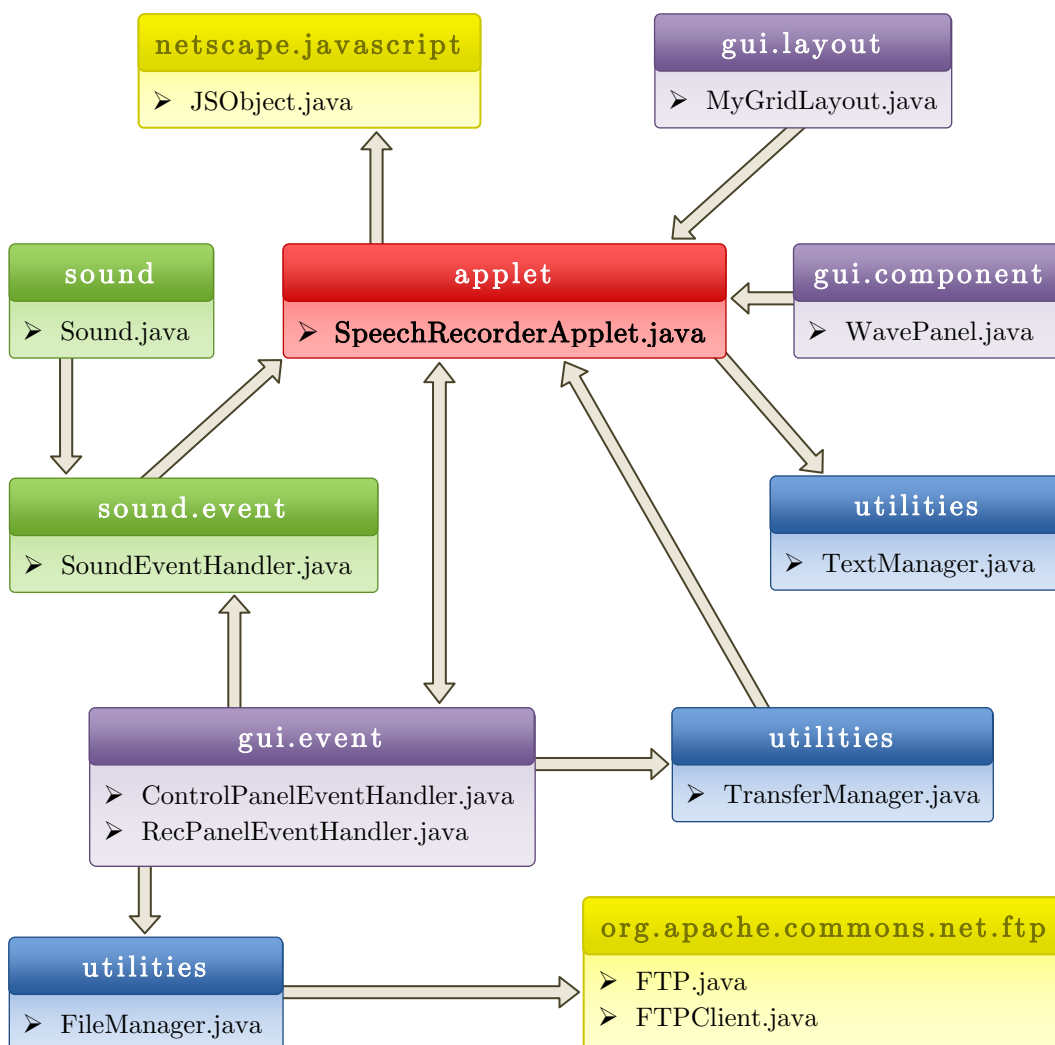
JNLP je definiran z datoteko XML, ki vsebuje množico pravil, kako naj se zažene Java Web Start aplikacija. V datoteki XML so zapisani tudi podatki o lokaciji paketa JAR (Java archive), v katerem je shranjena aplikacija, ime glavnega razreda le-te in dodatni parametri. Izvajanje Java Web Start aplikacije se prične tako, da se datoteka JNLP posreduje izvajalnemu okolju JRE, ki iz naslova, določenega v datoteki JNLP, prenese applet na uporabnikov računalnik in ga zažene [7].

Navedimo še zahteve, ki jih mora izpolnjevati snemalnik govora. Na prvem mestu mora omogočati zajem govornega signala preko ustreznih sistemskih virov. V primeru, da govorni signal nima ustrezne jakosti, mora aplikacija omogočiti uporabniku, da jo primerno uravna. Poleg posnetkov je potrebno imeti tudi podatke o govorcih, ki so snemali, zato naj snemalnik omogoča enostavno zajemanje tudi teh podatkov. Za konec omenimo tudi zahtevo po dinamičnem prilagajanju velikosti okna snemalnika glede na velikost tekstovnih polj. Ko bralec zaključi z branjem ponujenih besedil, lahko izbere naslednje primere, ki pa se lahko razlikujejo po dolžini stavkov. Zato mora znati snemalnik prilagoditi svojo velikost.

4.1 Shema in opis paketov aplikacije snemalnika

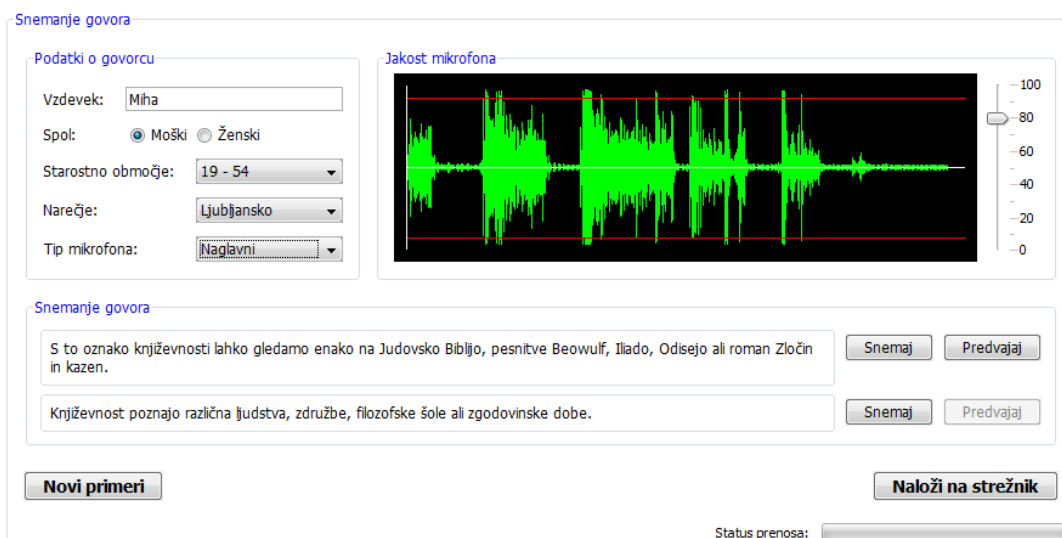
Slika 4.1 prikazuje zgradbo aplikacije snemalnika. Ta je razdeljena na več logičnih enot – programskih paketov, kjer vsak opravlja določeno nalogo. V

nadaljevanju so te naloge predstavljene z namenom lažjega razumevanja delovanja snemalnika.



Slika 4.1: Prikaz programskih paketov, ki sestavljajo aplikacijo snemalnika govora. Barvni kvadrati predstavljajo posamezne pakete, povezave med njimi pa njihovo komuniciranje oziroma spreminjanje atributov drugega.

Paket **applet** je na sliki 4.1 označen z rdečo barvo. Vsebuje glavni razred aplikacije snemalnika *SpeechRecorderApplet.java*. Ta, s pomočjo še



Slika 4.2: Grafični vmesnik snemalnika govora.

nekaterih razredov opisanih v nadaljevanju, poskrbi za inicializacijo vseh grafičnih komponent in drugih atributov.

Paket **gui.components** vsebuje razred *WavePanel.java*, ki v vmesniku aplikacije predstavlja površino, kjer se izrisuje jakost vhodnega signala, ko

uporabnik preneha s snemanjem določenega primera. Nato lahko uporabnik glede na izris presodi, ali naj zmanjša ali poveča jakost naprave za zajem govora.

Naloga paketa **gui.layout** je pravilno razvrščanje tekstovnih polj in gumbov za snemanje in predvajanje posnetkov v rubriki *Snemanje govora* znotraj aplikacije. Za to skrbi razred *MyGridLayout.java*.

Paketa *gui.components* in *gui.layout* preko svojih razredov skrbita za pravilen izgled in razporeditev nekaterih grafičnih elementov snemalnika govora. V paketu *gui* pa obstaja še en paket, ki preko razredov *ControlPanelEventHandler.java* in *RecPanelEventHandler.java* skrbi za interakcijo z uporabnikom in njegovo informiranje. Ta paket se imenuje **gui.event**. Razred *RecPanelEventHandler.java* spremlja stanje grafičnih elementov v rubriki *Snemanje govora* in glede na dogodke sproži primerno akcijo. Vsi paketi, ki so del paketa *gui* so na sliki 4.1 označeni z vijolično barvo.

Paket **sound** oziroma razred *Sound.java* predstavlja del aplikacije snemalnika, ki skrbi za zajem in predvajanje zvoka. Njegova naloga je tudi upravljanje jakosti naprave za zajem zvoka. Ko uporabnik v rubriki

Snemanje govora pritisne na gumb *Predvajaj* se preko razreda *Sound.java* začne predvajanje izbranega posnetka. Ko se ta konča, se sproži dogodek *LineEvent.Type.STOP*, ki za paket **sound.event** oziroma razred *SoundEventHandler.java* pomeni, da mora izvesti ustrezne akcije.

Naslednji paket predstavlja razrede aplikacije, ki skrbijo za vse ostale naloge. Paket se imenuje **utilities**, njegove naloge pa obsegajo:

- izbira besedil, ki so ponujena bralcu,
- generiranje naslova za hrambo in predvajanje posnetkov,
- pošiljanje posnetkov.

Za prvo od navedenih nalog skrbi razred *TextManager.java*, ki prejme besedilo iz podatkovne baze. Te nato shrani v svojo notranjo podatkovno strukturo in ob vsaki zahtevi uporabnika po novih primerih ustvari poljubni množico stavkov, ki jih posreduje glavnemu razredu. *FileManager.java* je zadolžen za generiranje pravih naslovov za shranjevanje in predvajanje posnetkov glede na indeks primera, ki ga izbere uporabnik. Za izpolnjevanje zadnje zahteve skrbi razred *TransferManager.java*, ki po končanem narekovanju prenese vse posnetke iz lokalnega začasnega direktorija v direktorij *uploads* na strežnik, kot omenjeno v poglavju 3.1.2.

Slika 4.1 prikazuje še dva paketa, ki sta označena z rumeno barvo. To sta paketa, ki nista bila implementirana v sklopu diplomske naloge. Prvi med njima je **org.apache.commons.net.ftp**. Paket implementira večino osnovnih internetnih protokolov: FTP/FTPS, NNTP, SMTP, POP3, Telnet,... [3]. Drugi paket pa je **netscape.javascript**, ki je del izvajalnega okolja JRE. Najdemo ga znotraj datoteke *plugin.jar*. Paket skrbi za klicanje funkcij JavaScript, ki so definirane v izvorni kodi spletne strani. Omogoča pa tudi, da preko JavaScript kode kličemo javno dostopne funkcije applet aplikacije.

4.2 Izvedba nekaterih delov snemalnika

Omenimo še nekaj pomembnejših delov snemalnika oziroma načinov njihove implementacije.

Snemalnik mora za svoje izvajanje vedeti, koliko primerov stavkov nekega besedila želimo ponuditi bralcu naenkrat in katero besedilo mu ponujamo. Ti dve vrednosti lahko podamo aplikaciji ob njenem zagonu kot parametra. Vrednosti teh v funkciji *init()* znotraj snemalnika preberemo s

klicem metode *getParameter(String paramName)*. Dobljeni vrednosti nato shranimo v lokalne spremenljivke glavnega razreda. Podatek o ponujenem številu stavkov bralcu služi v funkciji *createGUI()* kot zgornja omejitev za ustvarjanje grafičnih elementov za narekovanje v rubriki *Snemanje govora*.

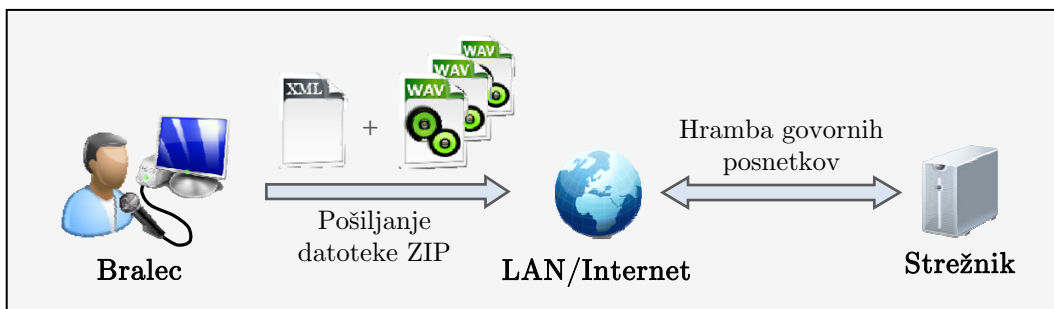
Snemalnik potrebuje za svoje pravilno delovanje tudi besedilo, ki ga mora prikazati uporabniku. Postopek pridobivanja besedila se prične s povezavo na podatkovno bazo znotraj skripte *applet.php*. Nato se v izvorno kodo snemalne strani, ki jo generira *applet.php*, doda JavaScript funkcijo *sendTextToApplet()*, ki jo prikazuje *slika*. Ta izvede klic funkcije *getTextFromJavaScript(String text)* glavnega razreda snemalne aplikacije, kjer argument *text* predstavlja želeno besedilo, pridobljeno iz podatkovne baze.

```
function sendTextToApplet() {  
    document.getElementById('captureApplet').  
        getTextFromJavaScript('$new_text');  
}
```

Slika 4.3: Primer JavaScript funkcije, ki izvede klic funkcije znotraj applet aplikacije, ki jo znotraj kode HTML označuje id *captureApplet*. *\$new_text* predstavlja PHP spremenljivko s besedilom, dobljenim iz podatkovne baze.

Ker so stavki znotraj posameznih besedil različnih dolžin, je potrebno spreminjati višino rubrike Snemanje govora, s tem pa tudi višino cele aplikacije znotraj spletne strani. To je realizirano s funkcijo *resize()*, znotraj glavnega razreda snemalnika, ki posreduje svojo trenutno velikost brskalniku preko klica funkcije JavaScript *resizeApplet(width, height)*.

Za konec omenimo še način implementacije prenosa posnetkov iz bralčevega računalnika na strežnik. Ko bralec izbere opcijo Naloži na strežnik, se najprej ustvari datoteka XML, ki vsebuje podatke o posnetkih in govorcju. Ta se ustvari kar v začasnem direktoriju, kamor so bili shranjeni tudi posnetki. Zatam razred *TransferManager.java* poskrbi za prenos vseh datotek iz začasnega direktorija na strežnik, natančneje v direktorij *upload unikatno_ime* kot omenjeno v poglavju 3.1.3. Postopek je prikazan tudi na sliki 4.4.



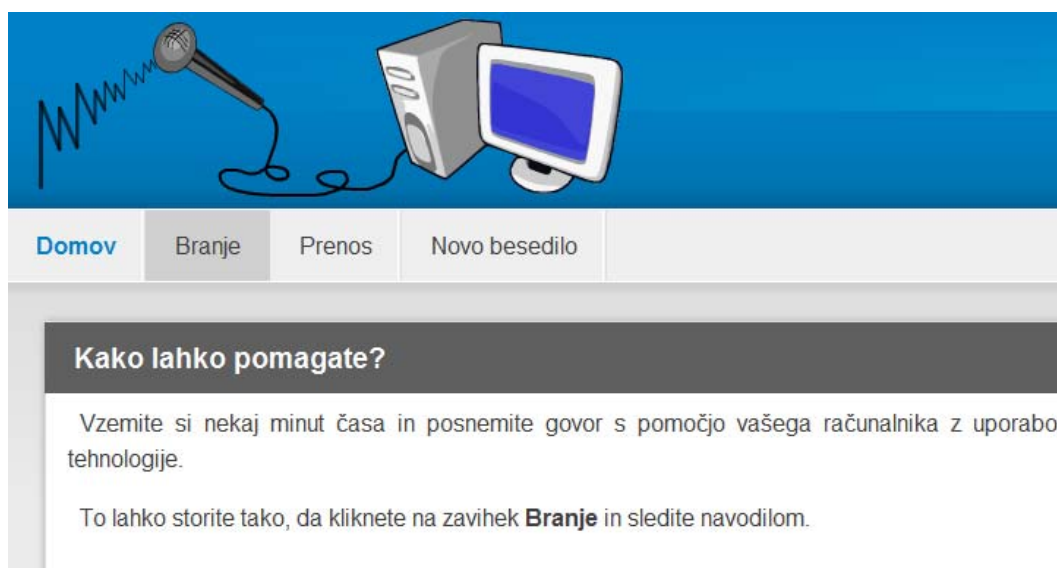
Slika 4.4: Shema pošiljanja posnetkov z bralečevega računalnika na strežnik.

5 Primeri uporabe

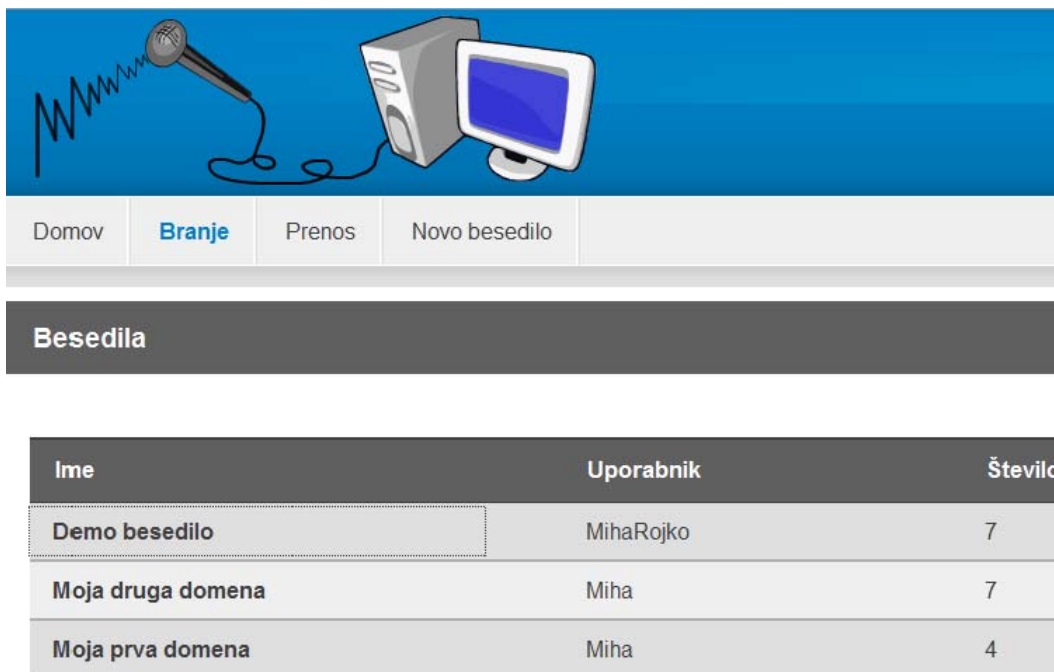
Poglavje prikazuje primere uporabe za navadnega obiskovalca, registriranega uporabnika in skrbnika spletne strani in lahko služi kot hiter priročnik za spoznavanje s spletno aplikacijo. Za vsako od navedenih vlog v sistemu je naveden po en primer uporabe.

5.1 Obiskovalec

Navaden obiskovalec je vsakdo, ki obiše spletno stran, a se ne vpiše v sistem. Kot takšen ima pravico do snemanja in prenašanja posnetkov. V nadaljevanju je podan primer uporabe za primer snemanja posnetkov.

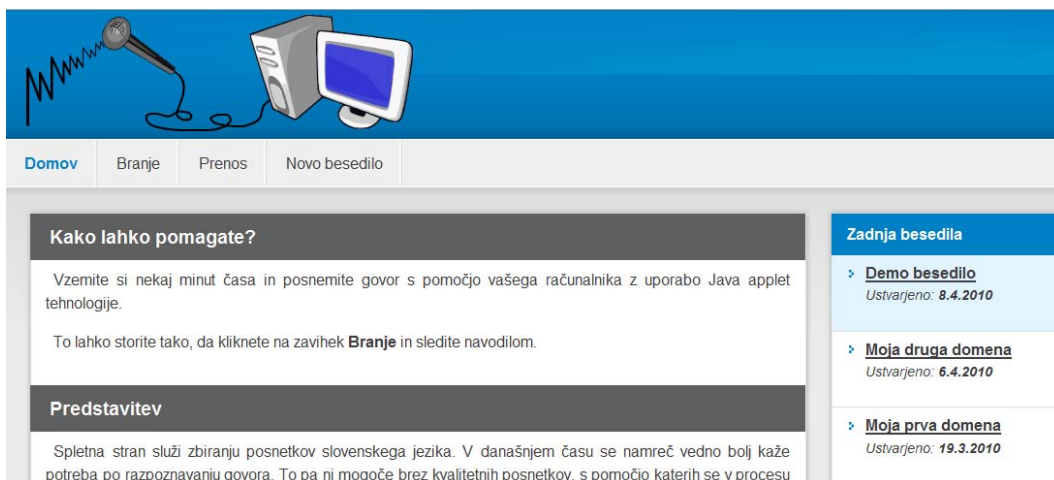


Slika 5.1: Obiskovalec na glavni strani spletne aplikacije izbere zavihek *Branje*.



Ime	Uporabnik	Število
Demo besedilo	MihaRojko	7
Moja druga domena	Miha	7
Moja prva domena	Miha	4

Slika 5.2: Na strani *Branje* v rubriki *Besedila* izbere želeno besedilo s klikom na naslov besedila. Sistem ga preusmeri na snemalno stran.



Kako lahko pomagate?

Vzemite si nekaj minut časa in posnemite govor s pomočjo vašega računalnika z uporabo Java applet tehnologije.

To lahko storite tako, da kliknete na zavihek **Branje** in sledite navodilom.

Predstavitev

Spletna stran služi zbiranju posnetkov slovenskega jezika. V današnjem času se namreč vedno bolj kaže potreba po razpoznavanju govora. To pa ni mogoče brez kvalitetnih posnetkov, s pomočjo katerih se v procesu

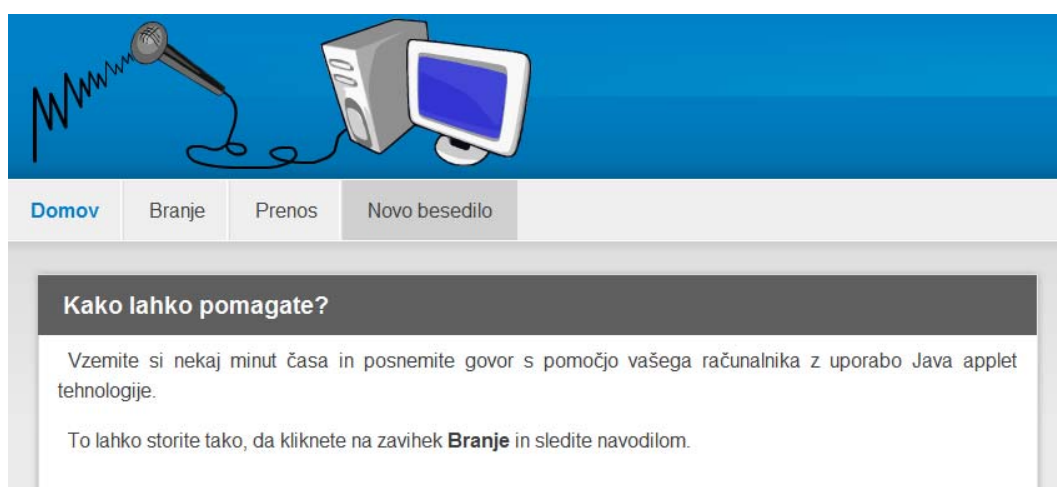
Zadnja besedila

- ▶ [Demo besedilo](#)
Ustvarjeno: 8.4.2010
- ▶ [Moja druga domena](#)
Ustvarjeno: 6.4.2010
- ▶ [Moja prva domena](#)
Ustvarjeno: 19.3.2010


Slika 5.3: Obiskovalec lahko izbere tudi katero od zadnjih ustvarjenih besedil kar iz rubrike *Zadnja besedila* na strani *Domov*. Sistem ga preusmeri direktno na snemalno stran.

5.2 Uporabnik

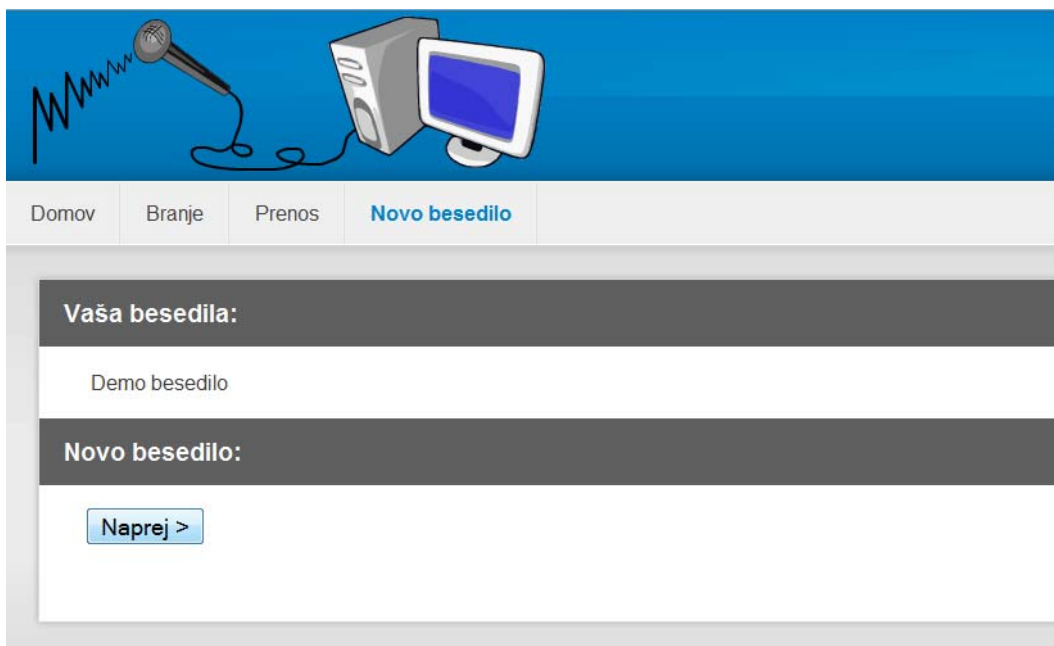
Uporabnik predstavlja vsakogar, ki je že izvedel postopek registracije v sistem in se lahko zato ob obisku strani vpiše v sistem. To mu poleg snemanja in prenašanja posnetkov omogoča tudi ustvarjanje novih besedil. V nadaljevanju je podan primer uporabe za primer ustvarjanja novega besedila.



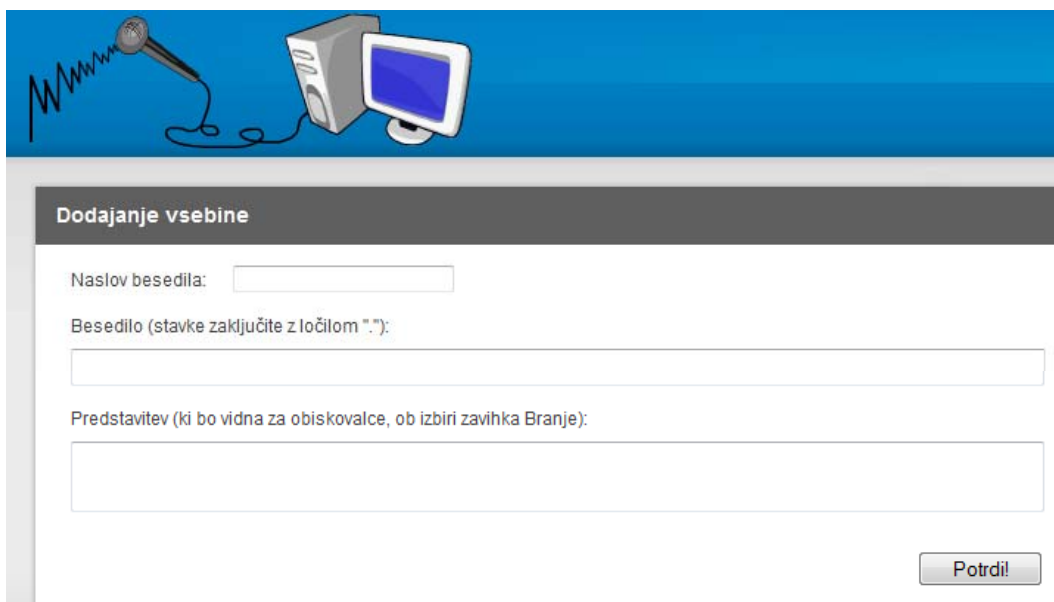
Slika 5.4: Uporabnik iz navigacijskega menija izbere opcijo *Novo besedilo*. Sistem preveri, če je uporabnik že vpisan.

The image shows a screenshot of a login form titled 'Prijava'. The form has two input fields: 'Uporabniško ime:' and 'Geslo:'. Below the input fields is a button labeled 'Prijava'. At the bottom of the form, there is a link that says 'Registrirate se lahko [tukaj!](#)'.

Slika 5.5: V primeru, da uporabnik še ni vpisan, ga sistem preusmeri na vpisno stran.



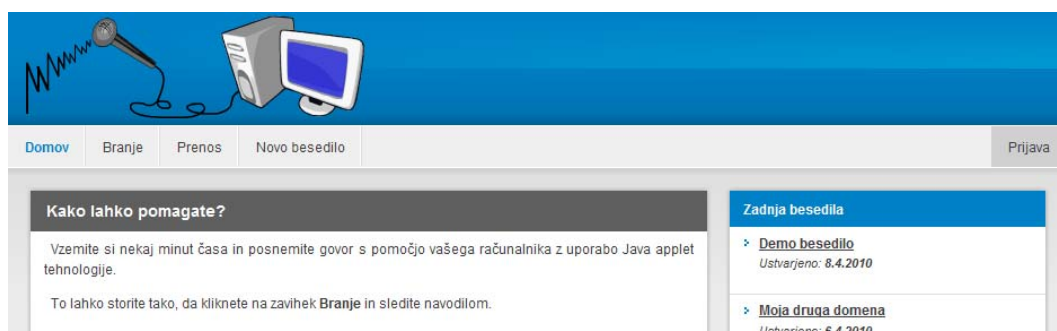
Slika 5.6: Uporabnika sistem usmeri na spletno stran, kjer lahko s klikom na gumb *Naprej >* začne z kreiranjem novega besedila.



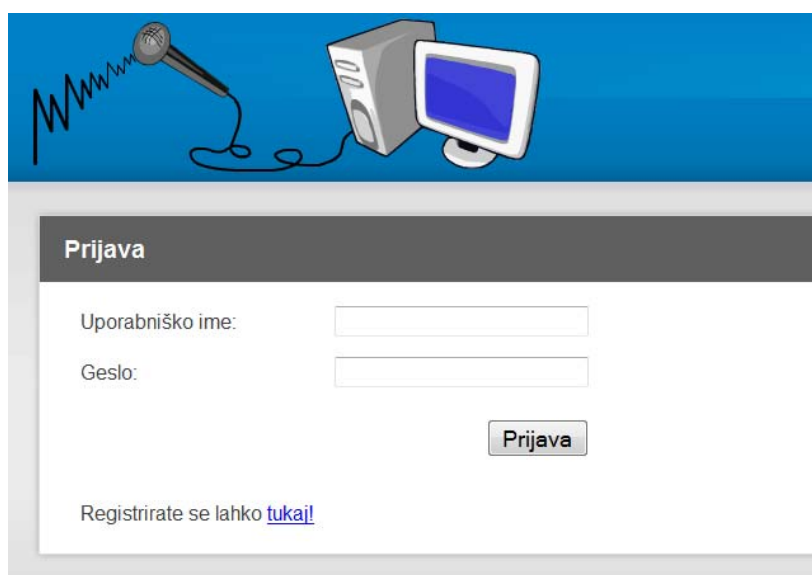
Slika 5.7: Uporabnik izpolni vsa ponujena polja in po končanem vnosu leta tega potrdi s klikom na gumb Potrdi!.

5.3 Skrbnik

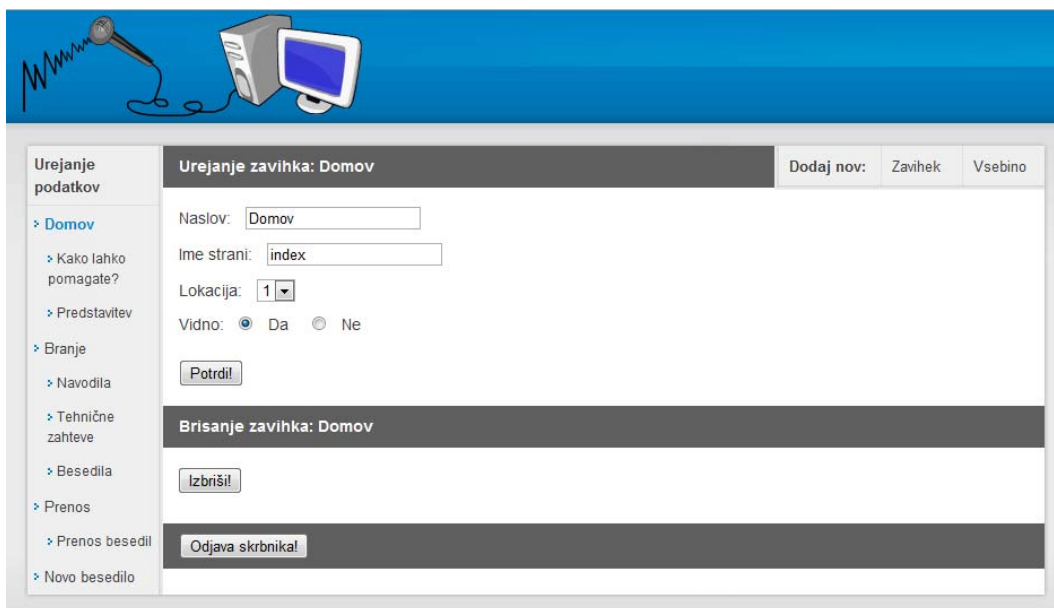
Skrbnik predstavlja prijavljenega uporabnika, ki ima poleg pravic do snemanja posnetkov, prenašanja posnetkov in ustvarjanja novega besedila tudi posebne pravice za spreminjanje vsebine spletne strani.



Slika 5.8: Skrbnik se mora najprej prijaviti v sistem. To stori tako, da na strani *Domov* pritisne gumb *Prijavi*.



Slika 5.9: Skrbnik se prijavi s svojim uporabniškim imenom ter geslo. Sistem ga ob uspešni prijavi preusmeri na skrbniške strani.



Slika 5.10: Skrbnik lahko v levem navigacijskem meniju izbere zavihek ali pa naslov, ki ga želi urediti. S klikom na enega od gumbov v zgornjem desnem kotu lahko prične z dodajanjem novega zavihka ali pa novega naslova znotraj neke strani. Spremembe lahko potrdi s pritiskom gumba *Potrdi!*. Če želi skrbnik izbrisati zavihek ali naslov znotraj le-tega, lahko to naredi s pritiskom gumba *Izbrisi!*. Po končanem urejanju vsebin se lahko skrbnik odjavi iz sistema s pritiskom gumba *Odjava skrbnika!*.

6 Zaključek

V diplomskem delu je opisana spletna aplikacija za zajem, hrambo in prenos posnetkov slovenskega jezika. Spletna aplikacija implementira sistem za upravljanje vsebin, ki omogoča spremembo tekstovnih vsebin spletnih strani. Sistem omogoča tudi kreiranje novih besedil, katerih stavki so v poljubnem vrstnem redu ponujeni obiskovalcem spletne strani. Tem spletna aplikacija omogoča tudi, da lahko posnetke tega ali kateregakoli drugega besedila prenesejo na svoj računalnik.

V diplomskem delu so bile omenjene spletne tehnologije, ki se danes največkrat uporabljajo. Izpostavljene so bile njihove prednosti in slabosti. Predstavljen je bil tudi postopek namestitve vseh potrebnih paketov za vzpostavitev razvojnega okolja. Omenjeni so bili tudi urejevalniki kode IDE, ki lahko olajšajo razvijanje spletnih aplikacij.

Navedene so bile tudi nekatere dodatne zahteve, ki naj jih izpolnjuje spletna aplikacija. Glede na to je bil izdelan načrt le-te, ki je služil kot vodilo pri implementaciji. Za potrebe morebitnega nadaljnjega razvoja smo opisali datotečno hierarhijo spletne aplikacije. Podani so tudi načini izvedbe nekaterih pomembnejših delov oziroma postopkov, ki sestavljajo spletno aplikacijo.

Opisane so bile tudi spletne tehnologije Java Applet, Web Start in JNLP, ki so uporabljene pri implementaciji snemalnika govora. Navedene so bile tudi vse zahteve, ki jih mora izpolnjevati snemalnik govora in podane shemo paketov, iz katerih je le-ta sestavljen. Opisani so tudi postopke implementacije nekaterih sestavnih delov snemalnika.

Podani so bili primeri uporabe spletne aplikacije za več različnih vlog uporabnikov. Omenjen je bil primer narekovanja besedila za obiskovalca, primer ustvarjanja novega besedila za uporabnika in primer upravljanja vsebine spletne strani za skrbnika sistema.

Spletna aplikacija je trenutno prilagojena za prikazovanje v brskalnikih Google Chrome in Mozilla Firefox.

Za nadaljnji razvoj omenimo le nekaj možnosti, ki zadevajo tako spletno aplikacijo kot tudi snemalnik govora. Prva takšna razširitev je implementacija iskalnika, ki bi poiskal in prikazal posnetke, ki bi ustrezali iskalnim kriterijem. Naslednja razširitev bi bila implementacija predvajalnika posnetkov, ki bi obiskovalcem omogočala poslušanje posnetkov, pre-

den bi jih prenesli k sebi. Sedanja različica snemalnika govora je implementirana tako, da se posnetki izbrišejo iz obiskovalčevega računalnika tudi, če ti predhodno niso bili uspešno preneseni na strežnik. Zato je ena odboljšav v prihodnosti lahko tudi implementacija boljšega načina prenosa posnetkov, ki jih posname obiskovalec. Ena od možnih rešitev je, da bi se podatki o morebitnem neuspehu zapisali v piškotek (cookie). Ob ponovnem obisku snemalne strani pa bi se postopek pošiljanja ponovil.

Seznam slik

Slika 3.1: Shema sistema za zajem in hrambo posnetkov slovenskega jezika	13
Slika 3.2: Načrt spletne aplikacije za zajem in hrambo posnetkov	15
Slika 3.3: Datotečna hierarhija - 1.del	16
Slika 3.4: Datotečna hierarhija - 2.del	17
Slika 3.5: Prikaz notranje zgradbe paketa <i>includes</i> in opis nalog posameznih skript	18
Slika 3.6: Prikaz osnovnega postopka generiranja spletne strani	19
Slika 3.7: Primer datoteke XML, v kateri so za vsak posnetek zapisani podatki o govorncu, besedilo, ki ga je bralec posnel, in ime datoteke s posnetkom	21
Slika 3.8: Konceptualni model podatkovne baze sistema, ki omogoča zajem, hrambo in prenos govornih posnetkov	22
Slika 4.1: Prikaz programskih paketov, ki sestavljajo aplikacijo snemalnika govora	28
Slika 4.2: Grafični vmestnik snemalnika govora	29
Slika 4.3: Primer JavaScript funkcije, ki izvede klic funkcije znotraj applet aplikacije	31
Slika 4.4: Shema pošiljanja posnetkov z bralčevega računalnika na strežnik	32
Slika 5.1: Obiskovalec - primer uporabe 1	33
Slika 5.2: Obiskovalec - primer uporabe 2	34
Slika 5.3: Obiskovalec - primer uporabe 3	34
Slika 5.4: Uporabnik - primer uporabe 1	35
Slika 5.5: Uporabnik - primer uporabe 2	35
Slika 5.6: Uporabnik - primer uporabe 3	36
Slika 5.7: Uporabnik - primer uporabe 4	36

Slika 5.8: Skrbnik - primer uporabe 1	37
Slika 5.9: Skrbnik - primer uporabe 2	37
Slika 5.10: Skrbnik - primer uporabe 3	38

Seznam tabel

Tabela 2.1: Prikaz najpomembnejših kriterijev primerjave spletnih tehnologij	10
Tabela 3.1: Tabela uporabnika	23
Tabela 3.2: Tabela vloge	23
Tabela 3.3: Tabela besedila	24
Tabela 3.4: Tabela posnetka	24
Tabela 3.5: Tabela naslova znotraj strani ali podstrani	25
Tabela 3.6: Tabela posamezne spletne strani	25

Literatura

- [1] (2010) Acoustic Model. Dostopno na:
http://en.wikipedia.org/wiki/Acoustic_Model.
- [2] (2010) Apache. Dostopno na:
<http://www.apache.org/>.
- [3] (2010) Apache Commons. Dostopno na:
<http://commons.apache.org/net/>.
- [4] D. Jurafsky, J. H. Martin, Speech and Language Processing,
New Jersey: Prentice Hall, 2008, pogl. 1.3.
- [5] (2010) FileZilla: The free FTP solution. Dostopno na:
<http://filezilla-project.org/>.
- [6] J. Valada, PHP and MySQL for Dummies, 2nd Edition,
Indianapolis: Wiley Publishing, 2004, str. 7 - 36.
- [7] (2010) Java Web Start. Dostopno na:
http://en.wikipedia.org/wiki/Java_Web_Start.
- [8] L. Rabiner, B. H. Juang, Fundamentals of Speech Recognition,
United States: Prentice Hall, 1993, str. 483.
- [9] (2010) Lynda.com. Dostopno na:
<http://www.lynda.com/>.
- [10] (2010) MySQL: World's most popular open source database.
Dostopno na: <http://www.mysql.com/>.
- [11] (2010) NetBeans. Dostopno na:
<http://netbeans.org/>.
- [12] (2010) PHP: Hypertext Preprocessor. Dostopno na:
<http://php.net/>.

- [13] (2010) Speech Recognition. Dostopno na:
http://en.wikipedia.org/wiki/Speech_recognition.
- [14] (2010) The Java Tutorials. Dostopno na:
<http://java.sun.com/docs/books/tutorial/deployment/applet/>.
- [15] (2010) VoxForge. Dostopno na:
<http://www.voxforge.org/>.
- [16] (2010) Website Design and Marketing Articles – .Net Comparison Chart.
Dostopno na:
<http://www.promoteware.com/Module/Article/ArticleView.aspx?id=10>.
- [17] (2010) W3School Online Web Tutorials. Dostopno na:
<http://www.w3schools.com/>.
- [18] (2010) XAMPP. Dostopno na:
<http://www.apachefriends.org/en/xampp.html>.