

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Rok Robida

Aplikacija za obvladovanje urnika poletne šole

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Rok Rupnik

Ljubljana, 2012

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



Št. naloge: 00286/2012

Datum: 12.04.2012

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

Kandidat: **ROK ROBIDA**

Naslov: **APLIKACIJA ZA OBVLADOVANJE URNIKA POLETNE ŠOLE**
APPLICATION FOR SCHEDULE MANAGEMENT OF SUMMER
SCHOOL

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Organizacija poletne šole zahteva med drugim tudi obvladovanje urnika. Zaradi različnih omejitev se urnik lahko spremeni tudi tik pred poletno šolo, spremeni pa se lahko tudi med potekom poletne šole.

Zasnujte spletno aplikacijo za celovito obvladovanje urnika poletne šole. Aplikacija naj izdelovalcu urnika omogoča predvsem učinkovitost pri spreminjanju urnika. Razvijte tudi skupino spletnih storitev, ki poljubnemu zunanjemu sistemu ali aplikaciji omogoča dostop do urnika.

Mentor:

doc. dr. Rok Rupnik

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU
diplomskega dela

Spodaj podpisani/-a ROK ROBIDA,

z vpisno številko 63070170,

sem avtor/-ica diplomskega dela z naslovom:

APLIKACIJA ZA OBVLADOVANJE URNIKA POLETNE ŠOLE

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

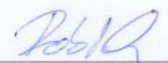
DOC. DR. ROK RUPNIK

in somentorstvom (naziv, ime in priimek)

- 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 12.09.2012

Podpis avtorja/-ice:



ZAHVALA

Zahvaljujem se mentorju doc. dr. Rok Rupniku za pomoč in nasvete v času nastajanja diplomskega dela.

Posebna zahvala gre družini za vzpodbude in podporo v času študija.

Kazalo vsebine

1. Uvod.....	1
2. Opis tehnoloških sredstev in orodij pri razvoju spletne aplikacije.....	3
2.1 Razvojna orodja	3
2.1.1 NuSphere PhpED.....	3
2.1.2 PowerDesigner	4
2.1.3 RestClient	5
2.2 Opis tehnologij.....	6
2.2.1 Označevalni jezik HTML.....	6
2.2.2 Stilski jezik CSS.....	7
2.2.3 Javascript (jQuery).....	7
2.2.4 PHP.....	8
2.2.5 CodeIgniter.....	8
2.2.2 Wamp	10
2.2.3 Apache strežnik	11
2.2.4 Podatkovna baza MySQL.....	11
3. Načrtovanje spletne aplikacije.....	13
3.1 Zajem zahtev aplikacije	13
3.2 Diagram primera uporabe	13
3.2.1 Diagram primera uporabe za administratorja.....	13
3.2.2 Diagram primera uporabe za skrbnika	14
3.2.3 Diagram primera uporabe za predavatelja	15
3.3 Načrtovanje podatkovna baze	16
4. Uporabe spletne aplikacije poletne šole	19
4.1 Namen.....	19
4.2 Praktična uporaba spletne aplikacije.....	19
4.2.1 Registracija.....	19
4.2.2 Prijava.....	20
4.2.3 Pozabljeno geslo.....	22
4.2.4 Odjava	23
4.2.5 Administracija konference	23
4.2.5.1 Dodaj nov dogodek	24

4.2.5.2 Uredi dogodek	25
4.2.5.3 Dodeljevanje pravic.....	29
4.2.5.4 Sponzorji	29
4.2.6 Administracija urnika konference.....	29
4.2.7 Administracija vprašalnika na temo predavanj.....	32
4.3 Prednosti.....	32
4.4 Pomanjkljivosti.....	35
4.5 Izboljšave spletne aplikacije.....	37
4.5.1 Razvoj spletne aplikacije v ogrodjem CodeIgniter.....	37
4.5.2 Integracija ACL	38
4.5.3 Sinhronizacija z Google koledarjem.....	39
4.5.4 Možnost dodajanja slik, videov in povzetkov	40
4.5.5 Prehod na CMS sistem.....	40
4.5.6 Uporaba spletne aplikacije preko SSL povezave:	41
5. Sklepne ugotovitve.....	43
6. Kazalo slik.....	45
7. Viri	47

Seznam uporabljenih kratic

AJAX - Asynchronous JavaScript and XML (tehnologija za ustvarjanje interaktivnih spletnih strani).

HTML - HyperText Markup Language (je protokol za prenos informacij po spletu)

CSS - Cascading Style Sheets (slogovna predloga, ki določa izgled spletnih strani)

PHP - Hypertext Preprocessor (odprtokodni skripni jezik za izdelavo dinamičnih spletnih strani)

REST - Representational State Transfer (programski arhitekturni stil za omrežne sisteme)

URL - Uniform Resource Locator (naslov spletne strani na svetovnem spletu)

JSON - JavaScript Object Notation (format za izmenjavo podatkov)

HTTPS - Hypertext Transfer Protocol Secure (protokol, ki je namenjen varni komunikaciji na svetovnem spletu)

POVZETEK

Vsako leto fakulteta za računalništvo in informatiko organizira poletno šolo za študente in dijake, ki želijo poglobiti znanje in pridobiti izkušnje iz novih tehnologij na področju računalništva. Namen diplomskega dela je bil razvoj spletne aplikacije, ki bi uporabnikom omogočala obvladovanje urnika poletne šole in vnos podatkov za lažje komuniciranje ter informiranje med uporabniki in študenti oziroma dijaki.

Za načrtovanje smo z orodjem PowerDesigner definirali model podatkovne baze ter diagrame primera uporabe, ki so nam služili v času razvoja. Jedro programske kode sestavljata strežniški jezik PHP in JavaScript knjižnica jQuery na uporabniški strani.

Spletno aplikacijo smo uspešno razvili. V fazi načrtovanja smo si koncept dobro zamislili, saj se med razvojem spletne aplikacije nismo soočali z večjimi težavami. Uporabniku smo omogočili uspešno splošno administracijo konference, pregledno urejanje terminov v koledarju in omogočili komunikacijo med poslušalci in predavatelji v obliki foruma.

Pri razvoju spletne aplikacije smo znanje, pridobljeno med študijem, uspešno uporabili za praktično reševanje problemov, ki so nastajali pri razvoju aplikacije. V primeru, da se bo spletna aplikacija uporabljala v namene organizacije poletne šole, bomo z razvojem nadaljevali in si prizadevali uporabnikom omogočiti čim bolj prijetno izkušnjo.

Ključne besede

spletna aplikacija, PHP, jQuery, komunikacija, uporabnik, strežnik

ABSTRACT

Each year the faculty of computer and information science organises a summer school where students are invited to gain knowledge of current technologies in their area of expertise. The goal of current thesis was to create a web application that enables easier organisation of events at the summer school. The aims were to create an easy to use web application offering easy scheduling of the events, ability to modify those events and lastly, to enable communication between speakers and listeners of the events scheduled at the summer school.

First, we have defined database model and use case diagrams with planning tool PowerDesigner which later helped us during developing phase. The core of the source code consists of server side scripting language PHP and of jQuery JavaScript library client side features.

The web application was successfully developed. In the planning phase, we created a firm concept which later led us through developing phase with minimum problems. We were able to offer easy administration of events, management of time slots and create discussion board where speakers and listeners can interact in a forum-like environment.

Developing this web application the knowledge gained at the faculty was well put to use and helped to form practical solutions to problems. In case developed web application is actually used for the purposes of managing summer school, our goal is to make sure we enable user friendly experience in the future.

Key words

web application, PHP, jQuery, communication, user, server

1. Uvod

V današnjem času se na svetovnem spletu pojavlja ogromna količina spletnih aplikacij, ki so namenjene tako širši publiki kot tudi ožjemu, specifičnemu krogu uporabnikov. Slednje so ponavadi bolj kompleksne in zahtevajo od uporabnika več predznanja, medtem ko so aplikacije namenjene širši populaciji relativno enostavne za uporabo.

Glede na vse hitrejši ritem življenja in potrebo po hitrem in lahkem dostopu do informacij so spletne aplikacije postale nuja sodobnega življenja. Lahkotnost osveževanja, vzdrževanja, možnost delovanja na več operacijskih sistemih in preprostost uporabe brez zasedanja prostora na uporabniških napravah so glavne prednosti uporabe takšnih informacijskih virov.

Našteti argumenti so razlog za porast pojava aplikacij. Spletne aplikacije so dostopne povsod, kjer je zagotovljen dostop do interneta. Ob zasnovi spletne aplikacije je potrebno upoštevati omejitve, poskrbeti za varnost in kakovostno oblikovati uporabniški vmesnik; preglednost in enostavna uporaba sta ključna za uspeh vsake aplikacije. Vsak razvijalec, ki načrtuje spletno aplikacijo teži k temu, da pospeš proces uporabe oziroma olajša uporabniško izkušnjo uporabniku.

Spletne aplikacije delimo na uporabniški ter strežniški del. Na uporabniškem delu poteka interakcija uporabnika s podatki in informacijami, ki jih posreduje spletna aplikacija. Strežniški del pa prisrubi oziroma shranjuje podatke, s katerimi je uporabnik manipuliral.

Na Fakulteti za računalništvo in informatiko v Ljubljani se vsako leto organizira poletna šola za študente in dijake, ki želijo poglobiti znanje in pridobiti izkušnje iz novih tehnologij na področju računalništva. Organizacija poletne šole zahteva med drugim tudi obvladovanje urnika. Zaradi različnih omejitev se urnik lahko spremeni tik pred začetkom izvajanja načrtovanega programa, do sprememb pa prihaja tudi med samo izvedbo.

Namen diplomskega dela je razvoj spletne aplikacije za obvladovanje urnika poletne šole, ki bi omogočala uporabnikom (administratorju, skrbniku, predavatelju) vnos časovnih in vsebinskih podatkov poletne šole za lažje komuniciranje ter informiranje med prej omenjenimi uporabniki ter študenti oziroma dijaki.

Spletna aplikacija za potrebe poletne šole je razdeljena na tri večje sklope:

- splošna administracija programa,
- urejanje terminov v koledarju,
- komunikacija med poslušalci in predavatelji v obliki foruma.

Vsaka oseba se mora pred uporabo aplikacije registrirati. Registrirana oseba prejme na svoj elektronski naslov kratko pozdravno sporočilo ter vstopne pravice za uporabo aplikacije, katere ji dodeli administrator. Administrator aplikacije lahko poljubno določi termin poletne

šole, doda logotip konference in dodeli skrbniške pravice ostalim sodelujočim osebam (skrbnik, predavatelj). Skrbnik si lahko zamisli urnik po dnevih. Na voljo mu je orodje v obliki koledarja, ki omogoča preglednejši nadzor nad vsemi termini urnika. Vsako predavanje ima svojo sekcijo, ki se loči po barvi. Barvo izbere skrbnik. Aplikacija ponuja možnost dodajanja sponzorjev urejenih po pomembnosti, določi organizacijski oziroma programski odbor oseb, predavatelji pa lahko vzpostavijo stik s poslušalci in tako dobijo takojšnjo povratno informacijo.

Cilji diplomske naloge so :

- prikaz modela primera uporabe,
- prikaz podatkovnega modela,
- praktična uporaba spletne aplikacije.

2. Opis tehnoloških sredstev in orodij pri razvoju spletne aplikacije

V tem poglavju bomo opisali vse tehnologije, ki so bile uporabljene v času razvoja. Delijo se na dve glavni veji:

- razvojna orodja,
- tehnološka sredstva.

2.1 Razvojna orodja

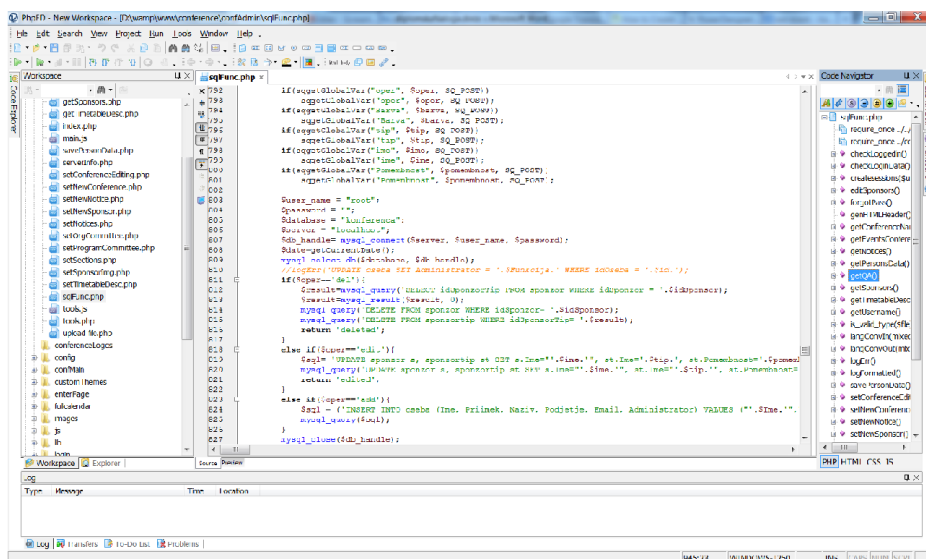
Tu bomo opisali razvojna orodja, ki so v pomoč oziroma zagotavljajo razvoj celovite programske rešitve.

Med pomembnimi uporabljenimi razvojnimi orodji so:

- NuSphere PhpEd,
- PowerDesigner,
- RestClient.

2.1.1 NuSphere PhpED

Programsko orodje PhpED (slika 2.1) je bilo razvito v podjetju NuSphere in je relativno mlado orodje, saj je bilo uporabnikom dostopno od leta 2001 dalje. Je plačljivo integrirano razvojno orodje, ki omogoča razvijanje v jezikih PHP, Javascript, HTML, CSS... Potrebno je omeniti, da ob enem nudi tudi podporo za zgoraj omenjene jezike. Urejevalnik PhpED deluje na Microsoft Windows platformi, Mac OSX in Linux.



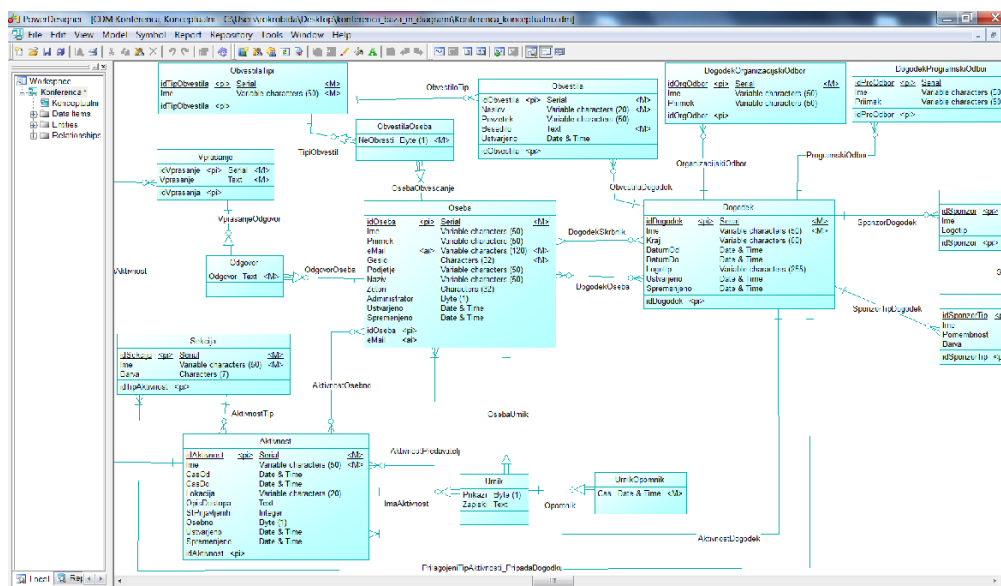
Slika 2.1: Orodje NuSphere PhpED.

Orodje PhpEd ponuja razvijalcem mnogo dodatnih funkcionalnosti za hitrejši in učinkovitejši razvoj:

- naprednejši izgled urejevalnika,
- omogoča vidnost sklopov programske kode, po predhodnem izboru razvijalca,
- dinamično osvetljevanje sintakse v delih kode,
- možnost osvetljevanja začetnih in končnih značk v HTML kodi,
- dopolnjevanje kode,
- možnost hitrega iskanja po kodi,
- kontrolo lokalne ter ostalih prejšnjih verzij,
- zmogljiv razhroščevalnik,
- validacija PHP, HTML ter CSS kode,
- testiranje enot,
- možnost vključevanja več tipov podatkovnih baz.

2.1.2 PowerDesigner

PowerDesigner je plačljivo orodje, ki je bilo razvito v podjetju Sybase. Deluje na operacijskem sistemu Microsoft Windows. Orodje je namenjeno modeliranju podatkov, modeliranju programske opreme in procesov v zgodnjih začetnih fazah projektov. Izdelava diagramov je edinstvenega pomena v fazi načrtovanja, saj lahko načrtamo celotno strukturo podatkovne baze ter primere funkcionalnosti, ki naj bi jih podpirala aplikacija. Ti diagrami so nam v pomoč v kasnejši fazi razvijanja programske logike (slika 2.2).



Slika 2.2: Prikaz načrtovanja konceptualnega modela podatkovne baze za spletno aplikacijo v orodju PowerDesigner.

Namestitev ter uporaba orodja PowerDesigner je enostavna. Omogoča nam poslovno modeliranje, modeliranje podatkov ter objektno modeliranje. Orodje teži predvsem k osredotočenosti na sam model, ne pa toliko na orodja pri razvoju celovite rešitve.

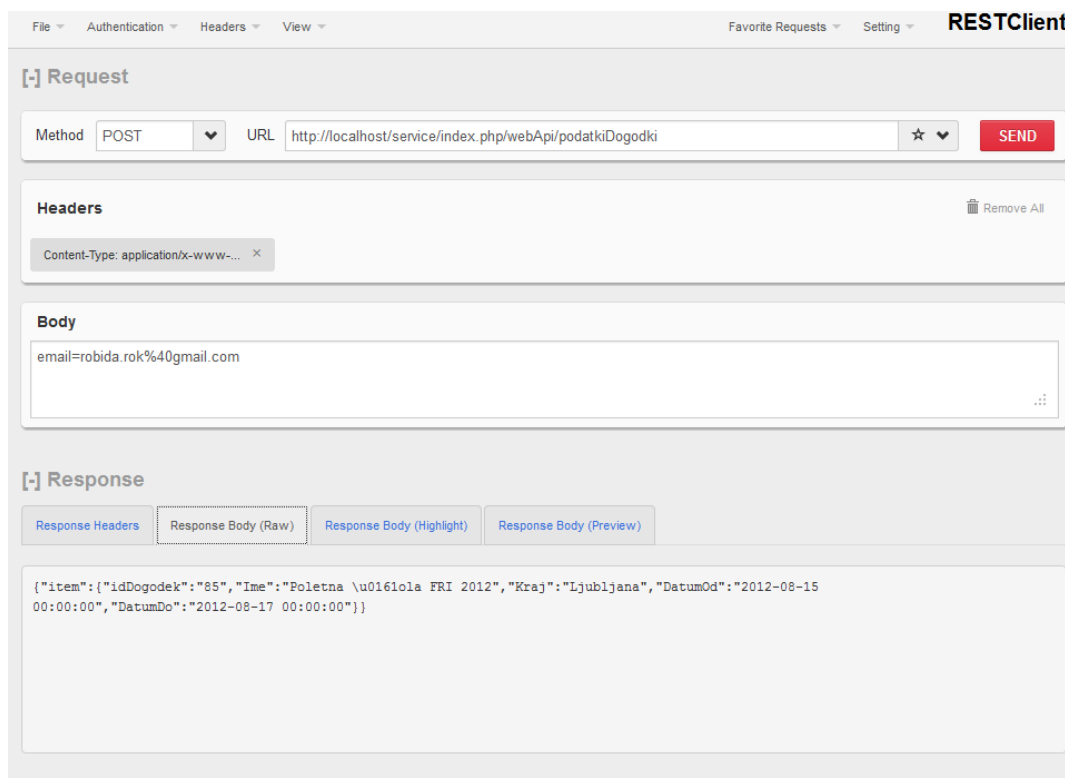
Lastnosti orodja PowerDesigner:

- večja produktivnost,
- podpira več različnih okolij,
- ponuja široko paleto orodij za analizo,
- zasnovan za kompleksna podjetja.

To orodje smo uporabili v začetni fazi načrtovanja samega poteka delovanja spletne aplikacije. Z njim smo si pomagali tudi pri načrtovanju podatkovnega modela (konceptualni, logični, fizični).

2.1.3 RestClient

RestClient je dodatek, ki ga namestimo v spletni brskalnik Mozilla Firefox. Z aplikacijo imamo možnost direktne komunikacije s strežnikom preko REST storitev. Vmesnik omogoča poljubno nastavitvev REST metode, URL naslov spletne storitve, na katerem je storitev dosegljiva, glavo ter parametre.



Slika 2.3: Primer klica REST storitve.

Slika 2.3 prikazuje primer odgovora na klic spletne storitve z orodjem RestClient, ki vrača tabelo konferenc glede na podan parameter (email) v JSON formatu. Spletna storitev je bila razvita v ogrodju CodeIgniter.

2.2 Opis tehnologij

V tem poglavju bomo opisali tehnologije, ki so nam bile v pomoč v času razvoja.

Med pomembnejšimi uporabljenimi tehnologijami so:

- označevalni jezik HTML,
- CSS,
- Javascript (jQuery),
- PHP,
- Codeigniter,
- Wamp,
- podatkovna baza MySQL,
- Apache strežnik.

2.2.1 Označevalni jezik HTML

HTML je označevalni jezik za prikaz spletnih strani ter ostalih informacij, ki jih je mogoče prikazati v spletnem brskalniku. Jezik HTML je standardiziran in izhaja iz družine SGML (Standard Generalised Markup Language). Zanj skrbi skupnost W3C, ki razvija standarde in zagotavlja rast in razvoj spleta na dolgi rok.

Uporaba jezika je razmeroma enostavna. Posebnost jezika HTML je, da uporablja značke kot so <html>. Značke jezika so lahko enodelne ali pa dvodelne. Enodelne se zaključujejo znotraj iste značke z uporabo znaka />. Dvodelne pa objemajo željen vstavljen objekt, npr. tabelo, kjer značka <table> ponazarja začetek, značka </table> pa zaključek. Končna poševna črta nam sporoči, da se vsebina značke na tem mestu zaključi [3]. Znotraj značke lahko uporabimo dodatne attribute, ki opišejo lastnosti kot so na primer poravnanaost, akcije, slogovni stil in podobno.

2.2.2 Stilski jezik CSS

Kaskadne stilske podloge ali CSS (angl. Cascading Style Sheets) so podloge, ki določajo obliko ter videz strani, napisane v označevalnem jeziku HTML. CSS so se razvile kot odgovor na potrebo po privlačnejših in stilsko oblikovanih spletnih straneh [3].

Z uporabo CSS se slogovni del spletne strani loči od ostalega dela. S tem vplivamo na lažje urejanje ter dodajanje slogov ter večjo preglednost kode, ki je napisana v HTML jeziku. Med drugim vplivamo tudi na samo optimizacijo kode, saj lahko uporabimo iste podloge na več različnih spletnih straneh.

2.2.3 Javascript (jQuery)

Javascript je skriptni programski jezik, ki omogoča razvijalcu razvoj interaktivnih spletnih strani na uporabniški strani (client-side). Javascript nam omogoča razvoj dodatnih funkcionalnosti in pomaga narediti spletno stran bolj funkcionalno [1]. Javascript ima med izvajanjem spletne aplikacije vpliv na HTML kodo in se običajno izvede ob sprožitvi nekega dogodka, kot na primer ob nalaganju spletne strani, ob kliku na element, premiku miške.

jQuery je brezplačna in obsežna javascript knjižnica, ki omogoča razvijalcu dodatne funkcionalnosti za hitrejši razvoj spletnih aplikacij. Bistvo te knjižnice je, da z manj kode naredimo več. Je preprosta za uporabo. Ima zelo obširno dokumentacijo, podprto s praktičnimi primeri, in veliko skupnost razvijalcev, ki skrbijo za redno osveževanje na tem področju.

Omogoča nam tudi reševanje težav, ki se pojavijo, ko enako spletno stran odpremo v različnih spletnih brskalnikih, pri tem pa se poruši oblika spletne strani. Knjižnica omogoča razvijalcem, da razvijejo lastne vtičnike in jih nato po potrebi vključijo v to knjižnico.

Znak \$ ima privzete ukaze jQuery funkcije. Omenjenemu znaku sledi selektor. Ta selektor je lahko bodisi sama HTML značka, značkin identifikator ali pa značkin razred. Po določitvi selektorja lahko uporabljamo katerokoli funkcijo, ki se nahaja v jQuery knjižnici.

Prednost uporabe te knjižnice je proženje AJAX (asinhroni javascript in xml) klicev. S pomočjo AJAX klicev si lahko aplikacija s strežnikom izmenjuje podatke v ozadju brez ponovnega nalaganja spletne strani. S tem močno pospešimo ter izboljšamo delovanje spletne aplikacije.

V času razvoja smo uporabili vtičnik JqGrid, ki uporablja jQuery knjižnico. JqGrid je rešitev, s katero je možno opisati večji tok podatkov v obliki seznama, omogoča pa tudi vrsto ostalih funkcionalnosti. Ker se rešitev izvaja na uporabniški strani, podatki pa se osvežujejo

dinamično (AJAX), lahko vtičnik integriramo s katerikoli programskim jezikom za načrtovanje dinamičnih spletnih strani (PHP, ASP, Perl, JSP).

Za potrebe uporabniku lažjega vnosa datuma ter ure smo uporabili jQuery gradnik DatePicker. Gradnik omogoča vnos datuma ali časa preko slikovnega vmesnika. Uporaba le tega je prinesla dve bistveni koristi:

- uporabniku prijaznejši ter enostavnejši vnos,
- preko programske logike zagotovljen enoten format vnosa.

2.2.4 PHP

PHP (angl. Personal Home Page Tools) je razširjen odprtokodni programski jezik, ki je namenjen razvoju spletnih dinamičnih strani. PHP se lahko uporablja na večini spletnih strežnikov in podpira skoraj vse operacijske sisteme in platforme. Za delovanje ne potrebuje licence. PHP programski jezik običajno teče na spletnem strežniku, pri čemer PHP koda kot vhod oblikuje spletno stran, kot izhod pa se lahko uporablja za ustvarjanje scenarijev z aplikacijami v ukazni vrstici za uporabniški vmesnik na strani odjemalca [5]. Spletni strežnik se v tem primeru obnaša kot interpreter PHP kode. Ta strežniška oprema je pogoj, da lahko izvajamo spletno stran na računalniku.

PHP je eden izmed prvih strežniku razvitih skriptnih jezikov, ki jih je moč vstaviti v HTML dokument.

Program je bil napisan kot skupina CGI programov v programskem jeziku C. Napisal ga je dansko-kanadski razvijalec Rasmus Lerdorf, da bi zamenjal nekaj skript napisanih v programskem jeziku Perl, ki jih je uporabljal za upravljanje svoje spletne strani [6].

Najbolj poznana orodja, ki so bila razvita v programskem jeziku PHP, so MediaWiki, Joomla, Wordpress, Drupal.

Za razvoj naše spletne aplikacije smo uporabili 5.3.13 verzijo PHP programske opreme.

2.2.5 CodeIgniter

CodeIgniter je odprto kodno spletno programsko ogrodje (angl. Framework), namenjeno razvoju dinamičnih spletnih aplikacij v programskem jeziku PHP. Razvito je bilo v podjetju ElisLab. Prva verzija opisane programske opreme je izšla leta 2006. Danes je na njihovi uradni spletni strani dostopna verzija 2.1. Z uporabo ogrodja se količina programske kode močno zmanjša, pridobimo na strukturiranosti kode pa tudi samo delovanje aplikacije je

hitrejše v primerjavi z osnovnim PHP programskim jezikom. Namestitev programske opreme in uporaba je enostavna.

Prednosti uporabe opisanega ogrodja je več. Na spletu najdemo obširno dokumentacijo, ki nam je v pomoč pri razvoju aplikacij. Poleg tega obstaja tudi velika skupnost razvijalcev, ki pripomorejo pri reševanju problemov. Obstajajo številni preprosti dokumentirani primeri, ki opisujejo rabo ogrodja na preprost, razvijalcu razumljiv način.

Platforme, ki jih CodeIgniter podpira za delo s podatkovno bazo so:

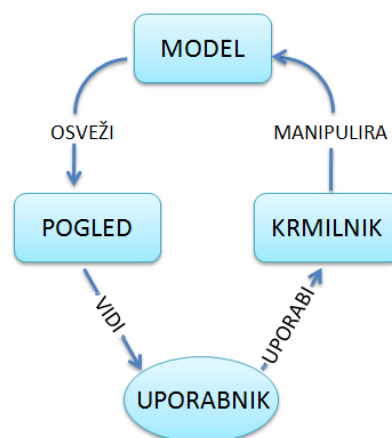
- MySQL
- MySQLi
- MS SQL
- Postgres
- Oracle
- SQLite
- ODBC

CodeIgniter temelji pretežno na razvojni arhitekturi model-pogled-krmilnik, angleško MVC (model-view-controller) (slika 2.4).

Krmilnik je kot nekakšen posrednik med modelom oziroma poslovno logiko in pogledom. Obravnava HTTP zahteve in na podlagi teh zahtev kliče ustrezne modele poslovne logike ter pripadajoče poglede.

V modelu se nahaja poslovna logika celotne aplikacije. Tu se nahaja programska koda, ki služi komuniciranju aplikacije z bazo podatkov.

Pogled vsebuje programsko kodo uporabniškega vmesnika, preko katerega poteka interakcija z uporabniki.



Slika 2.4: Shema MVC arhitekture.

2.2.3 Apache strežnik

Vsaka spletna aplikacija za delovanje potrebuje strežnik, ki upravlja z njeno logiko ter skrbi za hranjenje podatkov. Za gostovanje naše spletne aplikacije smo uporabili Apache strežnik.

Apache je zmogljiv strežniški program za izmenjavo in usmerjanje podatkov po protokolu HTTP. Zgrajen je bil leta 1994 na temeljih propadlega strežnika NCSA. Takrat se je skupina razvijalcev s področja strežniške programske opreme združila in iz popravkov NCSA strežnika sestavila celoto, ki so jo kasneje preimenovali v Apache. Bil je prva alternativa Netscapeovemu spletnemu strežniku. Od leta 1996 je napredoval na vseh področjih in njegova priljubljenost z vzponom na vodilno mesto strmo narašča.

Spletni strežniški program Apache danes domuje v več kot polovici vseh spletnih strežnikov po svetu. Priljubljen je predvsem zaradi vzdržljivosti, prilagodljivosti in zanesljivosti. Uporaba Apache-a je brezplačna (General Public Licence), kar v praksi pomeni, da so vse napake zelo hitro odpravljene, na spletu pa obstaja nešteto spletnih mest namenjenih podpori in izobraževanju. Apache med drugim podpira tudi skriptni jezik PHP, ki smo ga uporabili za razvoj programske logike pri naši spletni aplikaciji [7].

Aplikacija je prvotno podpirala le operacijska sistema Unix ter FreeBSD, danes pa je na voljo tudi za širše uporabljene operacijske sisteme kot so Microsoft Windows, Solaris in Mac OS X.

V času razvoja smo spletno aplikacijo gostili na strežniku Apache.

2.2.4 Podatkovna baza MySQL

MySQL je svetovno najpogosteje uporabljen relacijsko podatkovni sistem, ki deluje kot strežnik in omogoča večuporabniški dostop do podatkovnih baz. MySQL je klasična (relacijska) zbirka podatkov, ki lahko deluje kot samostojen podatkovni strežnik ali pa v kombinaciji z drugimi tehnologijami. Poizvedbe v podatkovni bazi MySQL se vršijo z jezikom SQL (Structure Query Language – strukturiran jezik za poizvedbe) [8]. Podatkovni sistem MySQL uporablja večina strani na spletu ter vsi večji odprtokodni projekti, ki potrebujejo sistem za upravljanje z bazo. Trenutno je nameščen na več kot osem milijonov sistemov in deluje na več kot dvajsetih operacijskih sistemih.

Podatkovni sistem MySQL je bil razvit leta 1995 v švedskem podjetju MySQL AB. Že od samega začetka pa vse do danes se ponuja pod pogoji uporabe javne licence GNU (General Public License) [8].

V obdobju razvoja smo aplikacijo gostili na WampServer-ju. Ta nam je služil kot testni strežnik za gostovanje, ki je imel integriran tudi podatkovni sistem MySQL.

3. Načrtovanje spletne aplikacije

V tem poglavju bomo predstavili tri pomembnejše faze pred samim razvojem aplikacije:

- zajem zahtev aplikacije,
- diagram primera uporabe,
- bazni model.

3.1 Zajem zahtev aplikacije

Pred samim načrtovanjem in razvojem smo določili zahteve, ki naj bi jih aplikacija izpolnjevala :

- brezhibno delovanje na spletnem brskalniku Mozilla Firefox ter na kateremkoli operacijskem sistemu,
- možnost hkratne uporabe večjega števila uporabnikov,
- preprosta in enostavna uporaba aplikacije,
- oblika v smislu barv naj ne bo preveč vpadljiva.

3.2 Diagram primera uporabe

Diagrami primera uporabe so interakcija med akterjem in sistemom z namenom doseganja skupnih ciljev. Akter je lahko oseba ali naprava, ki se nahaja izven sistema. V primeru uporabe lahko prikažemo zaporedje akcij, ki jih izvede sistem in daje samemu akterju nek rezultat.

Z orodjem PowerDesigner smo načrtovali diagrame primerov uporabe za naslednje akterje:

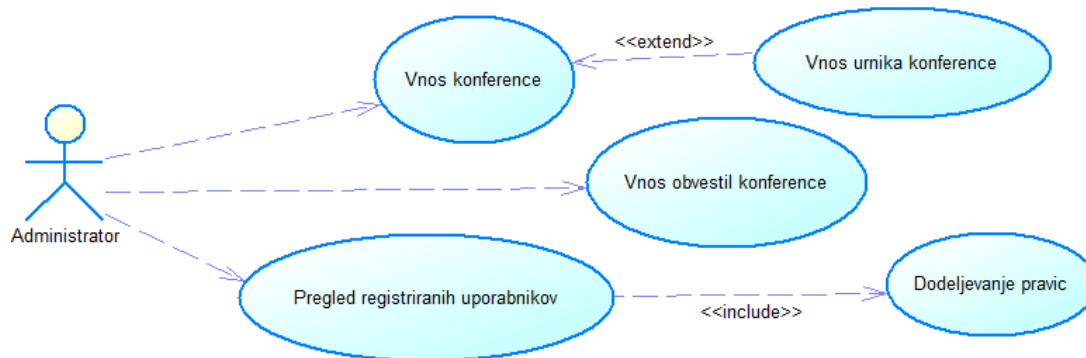
- administrator,
- skrbnik,
- predavatelj.

3.2.1 Diagram primera uporabe za administratorja

Administrator lahko uporablja aplikacijo na sledeče načine (slika 3.1):

- vnaša osnovne podatke o konferenci (ime, kraj, termin in logotip konference),
- dodaja in popravlja urnike predavanj za posamezne konference,
- dodaja, popravlja in briše obvestila, ki so namenjena udeležencem,

- ima možnost vpogleda v seznam vseh oseb, ki so registrirane in uporabljajo spletno aplikacijo,
- dodeli pravico novo registriranim osebam.

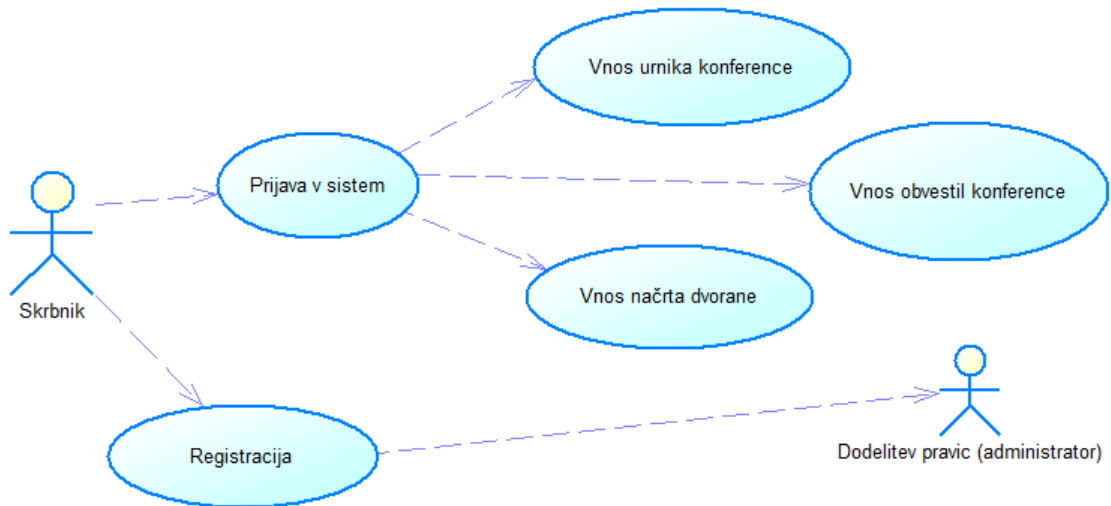


Slika 3.1: Diagram primera uporabe za uporabnika z administratorskimi pravicami.

3.2.2 Diagram primera uporabe za skrbnika

Primeri uporabe za skrbnika aplikacije so naslednji (slika 3.2):

- Registracija v sistem - skrbnik ob prijavi vnese podatke kot so ime, priimek, uporabniško ime ter izbrano geslo. Uspešno registriran uporabnik dobi povratno elektronsko sporočilo, ki mu pove, da je prišlo do registracije s prejemnikovim naslovom in ga opomni, da mu bo administrator dodelil pravice v najkrajšem možnem času,
- Ko so skrbniku dodeljene pravice, se mora pred urejanjem konference uspešno prijaviti v sistem.
- Tako kot administrator ima tudi skrbnik možnost dodajanja urnika konference, vendar le znotraj okvirov termina, ki mu ga je določil administrator.
- Skrbnik lahko vnaša obvestila za konferenco, ki mu je bila dodeljena s strani administratorja (sam je ne more ustvariti).
- Skrbnik ima tudi možnost dodajanja dodatnega opisa za načrt dvorane.

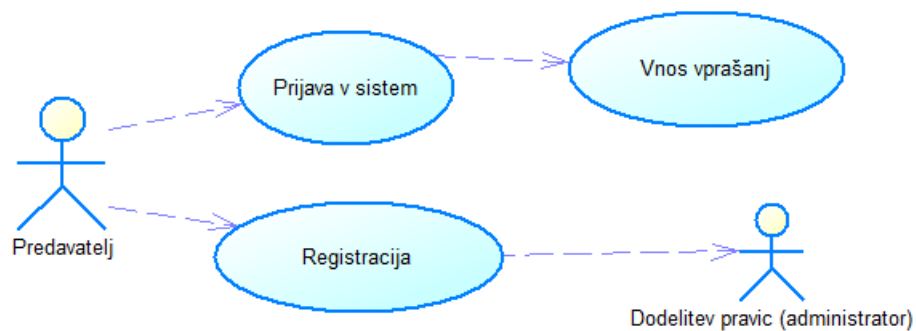


Slika 3.2: Diagram primera uporabe za uporabnika s skrbniškimi pravicami.

3.2.3 Diagram primera uporabe za predavatelja

Primeri uporabe za predavatelja so naslednji (slika 3.3):

- Postopek registracije in prijave v sistem je enak kot pri skrbniku.
- Predavatelj lahko uporablja aplikacijo tudi tako, da kot nosilec predavanj komunicira s študenti. Tako ima na primer možnost postavljanja vprašanj poslušalcem v zvezi s predavano snovjo in podobno.

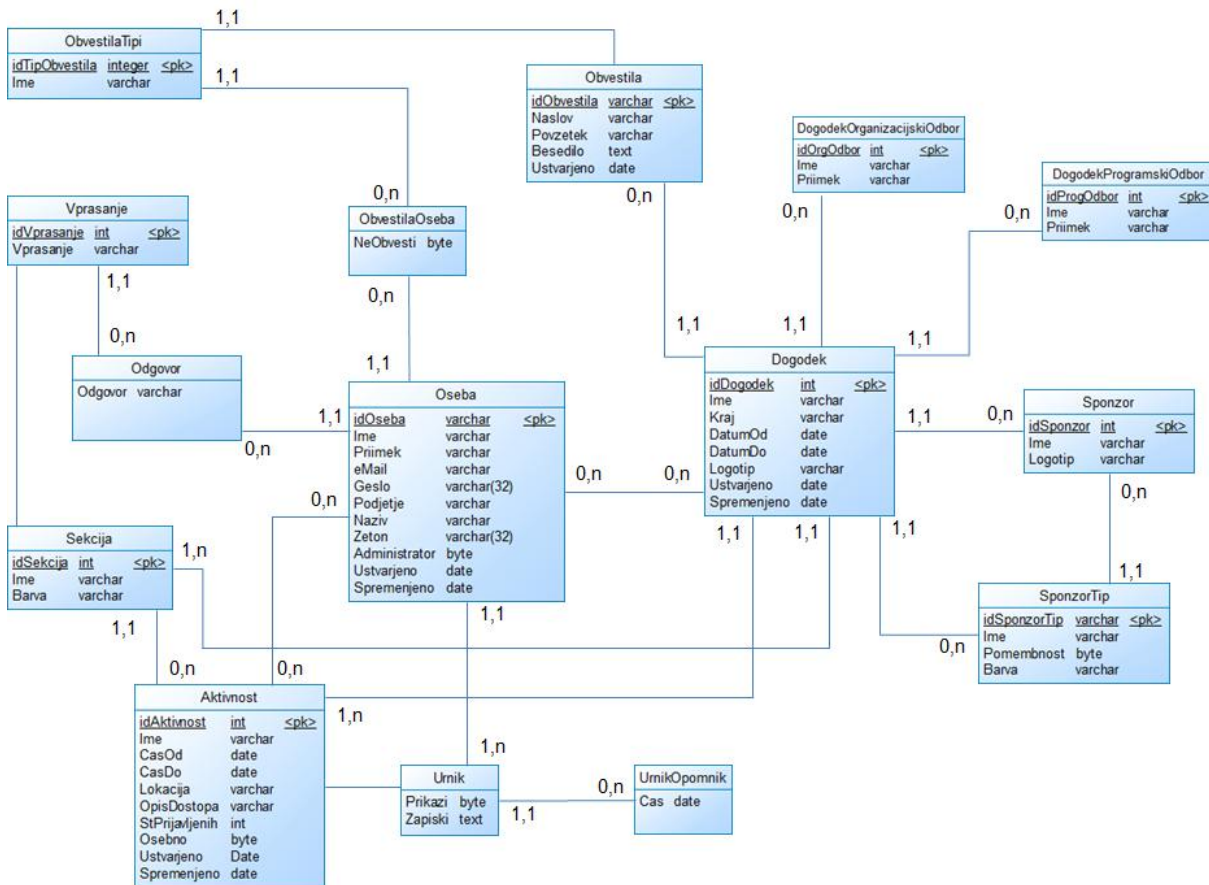


Slika 3.3: Diagrame primera uporabe za uporabnika s pravicami predavatelja.

3.3 Načrtovanje podatkovna baze

Podatkovni model smo realizirali z orodjem PowerDesigner. Podatkovni model je prikazan z:

- entitetami, ki vsebujejo attribute,
- povezavami, ki ponazarjajo medsebojna razmerja med entitetami v smislu števnosti.



Slika 3.4: Podatkovni model spletne aplikacije.

Kratek opis 15 entitetnih tipov podatkovnega modela (slika 3.4):

- *ObvestilaTipi*: šifrant obvestil, ki določa stopnjo pomembnosti posameznega obvestila (visoka, srednja, nizka),
- *Vprasanje*: podatki o vprašanjih, ki so vezani na sekcije (predavanja),
- *Odgovor*: podatki o odgovorih na zastavljena vprašanja,
- *Sekcija*: vsako predavanje oziroma aktivnost ima svojo sekcijo; sekcije se za boljšo preglednost ločijo po barvah,
- *Aktivnost*: osnovni podatki o aktivnostih oziroma predavanjih na konferenci,

- *ObvestilaOseba*: entiteta je namenjena uporabnikom mobilne aplikacije oziroma gostom poletne šole,
- *Oseba*: podatki o uporabnikih, ki se registrirajo preko spletne aplikacije,
- *Urniki*: entiteta je namenjena uporabnikom mobilne aplikacije oziroma gostom poletne šole,
- *UrnikiOpomniki*: entiteta je namenjena uporabnikom mobilne aplikacije oziroma gostom poletne šole,
- *Dogodek*: glavna entiteta poletne šole, katera vsebuje vse osnovne podatke programa,
- *DogodekOrganizacijskiOdbor*: podatki o članih, ki sestavljajo organizacijski odbor poletne šole,
- *DogodekProgramskiOdbor*: podatki o članih, ki sestavljajo programski odbor poletne šole,
- *Sponzor*: osnovnih podatki o sponzorju konference,
- *SponzorTip*: šifrant sponzorja, ki določa stopnjo pomembnosti posameznega sponzorja konference.

4. Uporabe spletne aplikacije poletne šole

V tem poglavju bomo opisali namen razvoja spletne aplikacije ter demonstrirali njeno praktično uporabo. Na koncu poglavja bomo tudi kritično ocenili uporabnost le-te.

4.1 Namen

Na fakulteti za računalništvo in informatiko se vsako leto organizira poletna šola za študente in dijake, kateri želijo poglobiti znanje in pridobiti izkušnje iz novih tehnologij na področju računalništva. Organizacija poletne šole zahteva med drugim tudi obvladovanje urnika. Zaradi različnih omejitev se urnik lahko spremeni tik pred začetkom poletne šole, sprembe pa so možne tudi med potekom programa.

Namen diplomskega dela je razvoj spletne aplikacije za pomoč pri obvladovanju urnika poletne šole, ki bi omogočala uporabnikom (organizatorju, administratorju, skrbniku, predavatelju) vnos podatkov poletne šole za lažje komuniciranje ter informiranje med prej omenjenimi uporabniki ter študenti oziroma učenci.

Sam projekt rešitve problema sestoji iz dveh ločenih delov. Diplomsko delo obsega samo administracijo konference, medtem ko drugi projekt ponuja rešitev prikaza informacij konference na mobilnih naravah.

4.2 Praktična uporaba spletne aplikacije

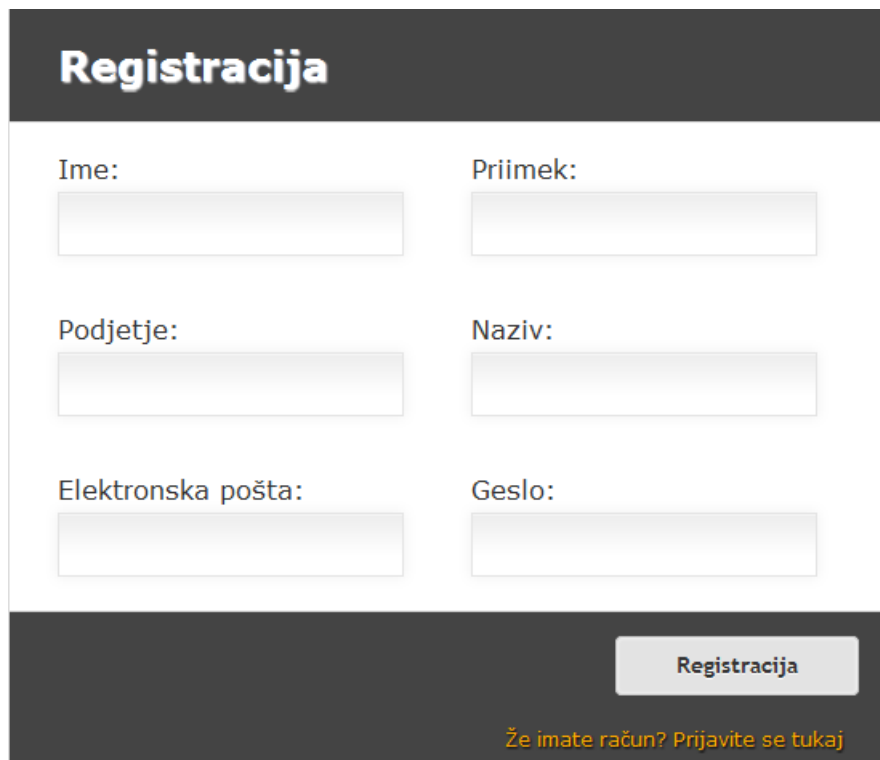
Tu bomo opisali vse faze dejanske uporabe z vidika strani uporabnika spletne aplikacije.

4.2.1 Registracija

S postopkom registracije si uporabnik zagotovi dostop do storitev spletne aplikacije na podlagi dodeljenih pravic administratorja kot nosilca aplikacije.

Ob vsakem obisku uporabnika spletne aplikacije se izriše vnosna maska (slika 4.1). V primeru, da uporabnik še ni registriran in želi postati uporabnik aplikacije, mora izbrati možnost »Nimate računa? Registrirajte se«. Z izbiro te možnosti se uporabniku prikaže obrazec za izpolnjevanje podatkov. Vsa vnosna polja na obrazcu so obvezna. Vnosna polja zahtevajo naslednje podatke: elektronski naslov, ki ob enem velja kot uporabniško ime ob prijavi, zeleno geslo, ime, priimek, podjetje ter naziv. S potrditvijo gumba »Registracija« potrdimo svojo registracijo. V primeru, da uporabnik pozabi izpolniti dotično polje ga ob

potrditvi registracije sistem na to avtomatsko opozori. Ob uspešni registraciji oseba na svoj elektronski naslov prejme sporočilo, ki opomni uporabnika, da se je uspešno registriral ter da mu bodo v najkrajšem časovnem okviru dodeljene pravice, katere mu pripadajo v okviru poletne šole. Ob dodelitvi pravic s strani administratorja se uporabnik lahko uspešno prijavi v aplikacijo.



The image shows a registration form with a dark header containing the title "Registracija" in white. Below the header, there are six input fields arranged in three rows and two columns. The first row contains "Ime:" and "Priimek:". The second row contains "Podjetje:" and "Naziv:". The third row contains "Elektronska pošta:" and "Geslo:". At the bottom right of the form, there is a button labeled "Registracija" and a link "Že imate račun? Prijavite se tukaj" in orange text.

Slika 4.1: Vnosna maska za registracijo.

4.2.2 Prijava

Ob začetnem vstopu v aplikacijo se prikaže prijavna maska z dvema vnosnima poljema (slika 4.2). Prvo je uporabniško ime, ki je enako naslovu elektronske pošte, drugo pa je polje, v katerega uporabnik vpiše svoje geslo. Oba podatka sta enaka podatkom, ki si ju je uporabnik izbral v postopku registracije. V primeru, da pride do napake pri vnašanju polj, sistem avtomatično opozori uporabnika o neuspešni prijavi v aplikacijo. V nasprotnem primeru je uporabnik prijavljen v aplikacijo.

Prijava

Uporabniško ime:

Geslo: [Ste pozabili geslo?](#)

Zapomni se me

Prijava

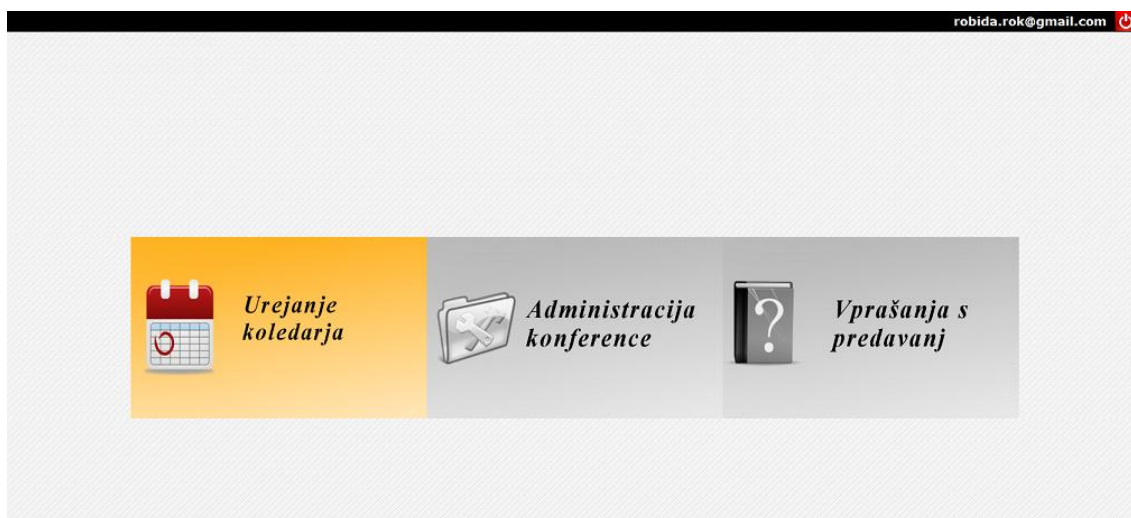
[Nimate računa? Registrirajte se](#)

Slika 4.2: Vnosna maska za prijavo v spletno aplikacijo.

Ob uspešni prijavi se uporabniku prikažejo trije glavni skopi (slika 4.3):

- urejanje koledarja,
- administracija konference,
- vprašanja s predavanj.

Sklop, ki ga lahko uporabnik uporablja, je zapisan v uporabniškem profilu, ki je določen z dodeljenimi pravicami.



Slika 4.3: Uporabniški vmesnik za izbiro sklopa.

4.2.3 Pozabljeno geslo

Ob prijavi v sistem je dodatna možnost tudi povezava nai koni»Ste pozabili geslo?«. Ta možnost je namenjena že registriranim uporabnikom, ki so pozabili svoje geslo. S pritiskom na povezavo se nam odpre maska z vnosnim poljem uporabniškega imena, v našem primeru elektronskega sporočila (slika 4.4). Ob potrditvi poslanega uporabniškega imena sistem najprej preveri ali elektronski naslov res obstaja. V primeru, da ne obstaja, sistem uporabnika opozori, da poslani elektronski naslov ne obstaja, v nasprotnem primeru pa sporoči uporabniku, da je geslo prejel na poslani elektronski naslov. V primeru, da uporabnik med krajšim obdobjem (v dvanajstih urah od zadnjega zahtevka) odda zahtevek za vnovično ponastavljanje pozabljenega gesla, ga sistem opozori na že poslano elektronsko pošto. V ta namen preprečimo nepotrebno obremenjevanje strežnika.

Slika 4.4: Vnosna maska za ponastavitev pozabljenega gesla.

Sistem uporabniku, ki je pozabil geslo, dodeli naključno geslo, ki je veljavno 12 ur po zahtevku za ponastavitev gesla. Začasno geslo je shranjeno v obliki piškotka, ki se pobriše predčasno v primeru, da je uporabnik zamenjal geslo v prej kot dvanajstih urah.

S prejetim geslom po elektronski pošti se uporabnik prijavi v sistem kot s pravim geslom. Ob uspešni prijavi se prikažeta dve vnosni polji, ki uporabnika opomnita na ponovno nastavitev gesla. Polja za geslo imajo zakrit pogled vnosa znakov, dodatno pa mora uporabnik ponovno vpisati isto geslo v drugo polje, da se sistem prepriča, da je le-to pravo. Z dvojnimi vpisi gesla se uporabnik dokončno prepriča, da je vnesel natanko takšno geslo, kot si ga je zamislil. S tem postopkom je ponastavitev pozabljenega gesla končana.

4.2.4 Odjava

Med uporabo aplikacije imamo v zgornjem robu ekrana orodno vrstico. V skrajnem desnem robu imamo gumb, ki omogoča odjavo iz aplikacije, ko končamo z delom. S tem se prekine seja in uporabnik ne more več dostopati do aplikacije oziroma je potrebna ponovna prijava v sistem.

Poleg gumba za odjavo se nahaja še uporabniško ime trenutno prijavljenega uporabnika. V primeru, da se uporabnik nahaja v enem izmed treh glavnih sklopov aplikacije, se mu v orodni vrstici prikaže povezava »Nazaj«, ki ga usmeri na stran, ki ponuja izbiro sklopov.

4.2.5 Administracija konference

Ena od možnosti izbire ob uspešni prijavi v aplikacijo je administracija konference. Vstop v ta sklop je mogoč samo v primeru, če so uporabnikovememu profilu dodeljene pravice administratorja oziroma skrbnika. Vsebina sklopa sestoji iz štirih zavihkov:

- Dodaj nov dogodek,
- Uredi dogodek,
- Dodeljevanje pravic,
- Sponzorji.

Medtem ko so administratorju vidni vsi štirje zavihki, ima skrbnik možnost dostopa le do urejanja dogodkov.

4.2.5.1 Dodaj nov dogodek

Slika 4.5: Uporabniški vmesnik za vnos novega dogodka.

Kot je razvidno iz slike 4.5, mora administrator najprej ustvariti novo konferenco, šele zatem je možno nadaljnje urejanje ustvarjene konference. Ob uspešnem vnosu vseh polj sistem uporabniku potrdi, da je bila konferenca uspešno dodana. Omeniti je potrebno, da so vsa polja obvezna pri vnosu nove konference, v nasprotnem primeru se podatki konference ne shranijo v podatkovno bazo.

Za izpolnjevanje terminov smo uporabili jQueryev vtičnik za vnos datumov, ki pripomore k lažjemu in hitrejšemu vnosu podatkov. Format zapisa datuma je standardiziran, vtičnik preprečuje vnos napak pri izpolnjevanju obrazca s strani uporabnika.

Pri dodajanju logotipov se ob potrditvi preveri format logotipa. V primeru neustreznega formata slike sistem opozori uporabnika. Trenutno so podprti formati slik: jpg, jpeg, bmp, gif ter png.

4.2.5.2 Uredi dogodek

Zavihek »Uredi dogodek« je edini zavihek znotraj sklopa administracije konference, ki je viden obema tipoma uporabnikov, tako administratorju kot tudi skrbniku.

Slika 4.6: Vnosni obrazec ob urejanju dogodka.

Primer (slika 4.6):

Skrbnik si izbere kreirano konferenco za katero ima pravice. Ob izbiri imena poletne šole se izpišejo podatki, katere je predhodno vnesel administrator. Ime, kraj, začetek ter zaključek dogodka so fiksni in jih ni mogoče spreminjati. V ostala polja lahko skrbnik vnese poljubno vsebino. Programski ter organizacijski odbor je sestavljen iz imen in priimkov oseb, ki sodelujejo na konferenci poletne šole. Uporabnik ima možnost dodajanja obvestil, ki so namenjena udeležencu poletne šole (slika 4.7).

Obvestila se izpisujejo v seznamu. Ta seznam se imenuje jqGrid in omogoča uporabniku pregleden prikaz vseh obvestil konference, možnost urejanja že obstoječih obvestil ali dodajanje novih. Vsako obvestilo vsebuje naslov, povzetek, besedilo in tri izraze za

označevanje pomembnosti obvestila: visoko, srednje ter nizko. Ob vsaki spremembi vnosa obvestila se shrani datum spremembe.

Slika 4.7: Vnosna maska za dodajanje novega obvestila v seznam.

Shranjen vnos se nato prikaže v seznamu (slika 4.8).

Naslov	Povzetek	Besedilo	Tip obvestila	Ustvarjeno
Poletna šola	Predavanja odpadejo.	Zaradi službenih obveznosti bo profesor XY v petek, dne 17.8.2012 odsoten.	Visoka	05.09.2012 05:48:26

Slika 4.8: Seznam trenutnih obvestil izbrane konference.

Skrbniku je na voljo tudi ogled trenutnega logotipa, v primeru da ga je administrator ob kreiranju konference nastavil. V primeru, da logotip še ni nastavljen oziroma želimo sliko zamenjati, pritisnemo na tipko »browse«, kjer se odpre okno za izbiro datotek shranjenih na lokalnem disku. Ob vsaki izbiri datoteke se preverja pravilnost formata slike. Če smo uspešno prenesli sliko na strežnik, se nam ob pritisku na gumb z lupo odpre pojavno okno z logotipom (slika 4.9).



Slika 4.9: Pojavno okno z logotipom.

Konferenca lahko vsebuje eno ali več sekcij. Sekcija je v tem primeru definirana kot vrsta ali tip predavanja. Vsaki sekciji skrbnik ali administrator dodeli poljubno barvo (slika 4.10). Izbran ton barve se bo prikazoval ob izbiri tipa sekcije, ko bo uporabnik vnesel urnik konference v koledar. Brez predhodnega vnosa sekcije je možnost dodajanja terminov v koledar onemogočena.

Slika 4.10: Vnosna maska za dodajanje sekcij konferenci.

4.2.5.3 Dodeljevanje pravic

V tem zavihku ima administrator pregled nad vsemi uporabniki spletne aplikacije. V seznamu so na voljo osnovni podatki oseb ter njihove pravice. Tudi tu ima administrator možnost urejanja, dodajanja ali brisanja oseb.

V tem sklopu administrator upravlja tudi s pravicami registriranih oseb. Za pravice so po dogovoru možna štiri stanja:

- *Stanje 0:* ima na novo registriran uporabnik, ki še nima pravice dostopa oziroma se še ne more prijaviti v spletno aplikacijo,
- *Stanje 1:* ima administrator, ki naj bi bil za potrebe te spletne aplikacije le en uporabnik,
- *Stanje 2:* ima uporabnik s pravicami skrbnika,
- *Stanje 3:* ima uporabnik s pravicami predavatelja.

4.2.5.4 Sponzorji

Zadnji sklop administracije je delo s sponzorji. Tudi tu lahko administrator dodaja, ureja ali izbriše sponzorje, ki so urejeni v seznamu.

Sponzorji se bodo pojavljali na oglasnem prostoru mobilne aplikacije, preko katere bodo dostopali učenci ter študentje poletne šole.

Slika 4.11: Vnosna maska za dodajanje novih sponzorjev.

Za uspešno dodanega sponzorja je potrebno izpolniti vsa polja obrazca (slika 4.11). Pred vsakim vnosom novega sponzorja je potrebno predhodno vnesti konferenco. Najprej si izmed danih opcij izberemo ime konference, kateri želimo dodati sponzorja. Nato

določimo ime sponzorja in dodamo njegov logotip. Sponzorju določimo pomembnost na lestvici od 1 do 5, pri čemer 1 pomeni manj pomemben in 5 bolj pomemben. Na podlagi te ocene se bodo oglasi na mobilnih napravah prikazovali bolj pogosto ali v večjem formatu. V naslednjem polju ocenimo tip sponzorja, ocene pa izbiramo na ponujeni lestvici: platinasti, zlati ali srebrni. V zadnjem polju določimo prikazano barvo sponzorju. Tudi tukaj je izbira barv ponujena v naprej.

4.2.6 Administracija urnika konference

Ta sklop je namenjen skrbniku spletne aplikacije. Uporabnik ima možnost urejanja urnika za konferenco, katere skrbnik je. Sekcije preko urnika doda v časovnem okviru, katerega mu je določil administrator. Skrbnik lahko poljubno dodaja sekcije na koledar. Glavna prednost tega sklopa je, da uporabnik lahko interaktivno spreminja termine sekcij in mu je tako omogočeno poenostavljeno ter preglednejše planiranje urnika za izbrano konferenco.



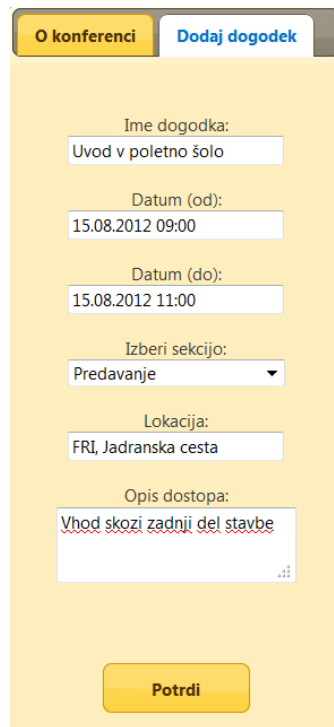
Slika 4.12: Uporabniški vmesnik za administracijo urnika.

Uporabnik mora v polju z izbiro izbrati konferenco, katero želi urejati. Ob izbiri pod poljem za izbiro dogodka se v prvem zavihku »O konferenci« prikažejo osnovni podatki ter časovni okviru poteka konference. Ob izbiri drugega zavihka »Dodaj dogodek«, se uporabniku odpre obrazec za dodajanje sekcij na koledar (slika 4.12). V primeru, da uporabnik ni dodal barve sekcije ob administraciji konference, je dodajanje sekcij na koledarju onemogočeno.

Za uspešno dodano sekcijo na koledarju je potrebno izpolniti vseh šest vnosnih polj (slika 4.13). Najprej je potrebno izpolniti ime dogodka, nato vpišemo začetek in konec sekcije. Ob

potrditvi sistem preveri časovni okvir dodane sekcije. V primeru, da je sekcija izven intervala, ki ga je določil administrator, je dodajanje sekcije na koledar neuspešno.

Sledi izbor tipa sekcije, ki je bil predhodno vnešen. Ostali dve polji, lokacija ter opis dostopa, pa sta koristni predvsem za poslušalca te sekcije. Ko izpolnimo vsa polja, potrdimo z gumbom »Potrdi« in sekcija je uspešno dodana. Lahko dodamo tudi dve sekciji znotraj istih časovnih okvirjev, npr. dve predavanji ki se odvijata hkrati, vendar na različnih lokacijah (slika 4.14).



The image shows a web form for adding a conference session. At the top, there are two tabs: "O konferenci" (highlighted in yellow) and "Dodaj dogodek". The form fields are as follows:

- Ime dogodka: Uvod v poletno šolo
- Datum (od): 15.08.2012 09:00
- Datum (do): 15.08.2012 11:00
- Izberi sekcijo: Predavanje (dropdown menu)
- Lokacija: FRI, Jadranska cesta
- Opis dostopa: Vhod skozi zadnji del stavbe

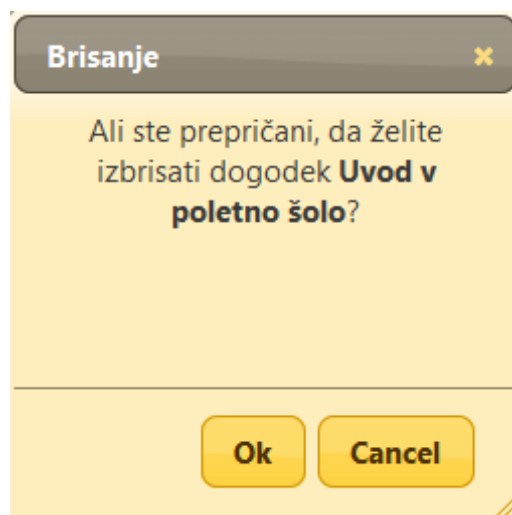
At the bottom of the form is a yellow button labeled "Potrdi".

Slika 4.13: Uporabniški vmesnik za dodajanje sekcije.

	Ned, 12.8.	Pon, 13.8.	Tor, 14.8.	Sre, 15.8.	Čet, 16.8.	Pet, 17.8.	Sob, 18.8.
Ce odnevi							
8:45							
8:50							
8:55							
9							
9:05				9:00 - 1:00 Uvod v poletno	9:00 - 10:00 Razovanje praktičnih nalog		
9:10							
9:15							
9:20							
9:25							
9:30							
9:35							
9:40							
9:45							
9:50							
9:55							
10							
10:05							
10:10							
10:15							
10:20							
10:25							
10:30							
10:35							
10:40							
10:45							
10:50							
10:55							
11							
11:05							
11:10							
11:15							
11:20							
11:25							

Slika 4.14: Primer prikaza sekcij na urniku koledarja.

Ko smo sekcije uspešno dodali, jih lahko urejamo po želji. V primeru, da želimo zamenjati termin določene sekcije, jo le povlečemo na nov željen termin. Ob spustu miške se bo termin sekcije samodejno posodobil v podatkovni bazi. V primeru, da želimo sekcijo odstraniti s seznama, pa uporabimo dvoklik z miško na sekcijo. Pojavi se nam dialog z vprašanjem namere (slika 4.15).



Slika 4.15: Dialog za brisanje sekcije.

4.2.7 Administracija vprašalnika na temo predavanj

Komunikacija med predavateljem in poslušalcem je zelo pomembna, še posebej na tipu konference kot je poletna šola. Ker morajo študentje in dijaki pridobiti veliko količino informacij v kratkem časovnem obdobju, je bil dodatno razvit sklop vprašalnika na temo predavanj. Služil naj bi kot nekakšen mini forum z namenom hitrejše komunikacije med predavatelji ter poslušalci.

Ob vstopu v sklop programska logika najprej preveri ali je oseba predavatelj. V primeru, da je uporabnik predavatelj in je že predhodno postavil vprašanje na temo svojih predavanj, se mu v seznamu prikaže dialog v obliki drevesne strukture vprašanj in odgovorov. Drevesna struktura je tro-nivojska. Prvi nivo predstavlja ime sekcije. V drugem nivoju je postavljeno predavateljevo vprašanje vezano na to sekcijo. V tretjem nivoju drevesne strukture pa se nahajajo vsi odgovori poslušalcev na zastavljeno vprašanje. Tako predavatelju zagotovimo dober pregled nad zastavljenimi vprašanji ter odgovori.

Sekcija/Vprašanje/Odgovor	Oseba
▼ Uvod v poletno šolo	
▶ Kako se vam je zdelo uvodno predavanje letošnje poletne šole?	Janez Novak
○ Pogrešal sem več slikovnega gradiva, drugače je bilo OK.	Peter Klepec
○ Bilo je super! Komaj čakam jutrišnji dan...	Miha Škrat
▶ Reševanje praktičnih nalog	
▶ Kaj pa zahtevnost nalog na delavnicah?	Janez Novak

Slika 4.16: Prikaz dialoga med predavateljem in poslušalci v obliki drevesnega seznama.

Na sliki 4.16 sta prikazani dve sekciji, katerih nosilec je predavatelj Janez Novak. Za vsako sekcijo je predavatelj zastavil po eno vprašanje. Na prvo vprašanje je že dobil dva odgovora.

4.3 Prednosti

V nadaljevanju bomo opisali prednosti uporabe spletne aplikacije v splošnem smislu kot tudi uporabnost dotične aplikacije za poletno šolo.

Uporabnost spletne aplikacije je zelo obsežen pojem. Kadar govorimo o uporabnosti spletnih strani se osredotočimo le na uporabnika, pozabimo pa, da mora biti aplikacija uporabna tudi za lastnika, vzdrževalca ali razvijalca s pogleda upravljanja ter vzdrževanja.

Okvirno lahko uporabnost aplikacije opišemo z naslednjimi dejavniki:

- podpira zastavljene cilje, ki so bili dogovorjeni v fazi načrtovanja,

- zadovolji uporabnikove potrebe kot so enostavno učenje, koristnost pridobljenega znanja za ponovne uporabe, zadovoljstvo ter udobje pri uporabi,
- zadovolji lastnikove potrebe kot na primer majhni stroški za vzdrževanje.

Prednosti uporabe spletne aplikacije za poletno šolo so naslednje:

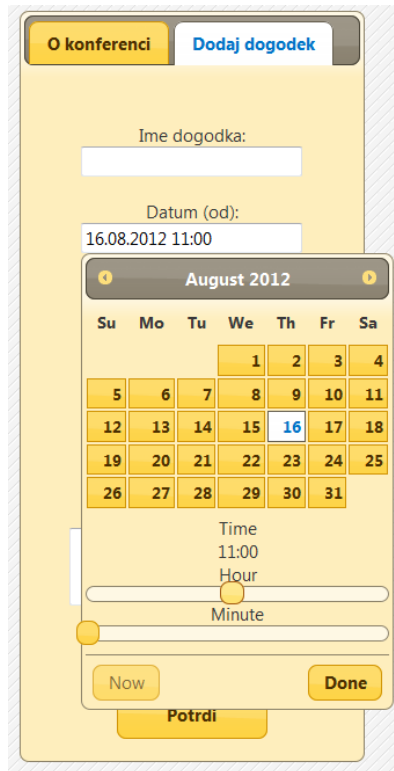
- **Podpora več brskalnikov**

Spletni brskalnik je predpogoj za delovanje vsake spletne aplikacije. Kljub začetnemu dogovoru aplikacijo podpira več kot en spletni brskalnik. V fazi načrtovanja smo določili za primarni brskalnik, ki ga aplikacija podpira, Mozilla Firefox. Ostali spletni brskalniki omogočajo večino spletnih funkcionalnosti, ki jih aplikacija nudi. V današnjih časih ima vsak operacijski sistem svoj brskalnik oziroma si ga uporabnik lahko po želji namesti. S tem, da naša aplikacija podpira večino spletnih brskalnikov smo omogočili uporabnikom platformno neodvisnost pri uporabi spletne aplikacije.

- **Enostavnost uporabe**

Uporaba spletnih brskalnikov je enostavna. Sodobni brskalniki podpirajo funkcije, ki olajšajo delo na spletu. Mednje lahko štejemo navigacijo med stranmi ter shranjevanje ali tiskanje vsebin.

Tudi razvoj naše spletne aplikacije je ves čas težil k enostavnosti vmesnika. Ker aplikacija vsebuje številna vnosna polja obstaja precejšnja verjetnost, da se bodo pojavljale napake pri vnosih. Da bi se v čim večji meri izognili napakam človeškega faktorja, smo nekaterim poljem dodali gradnike, ki omogočajo lažje izpolnjevanje polj ter tako zmanjšajo vpliv napak človeškega faktorja (slika 4.17). Predvidevali smo, da bodo aplikacijo uporabljali različni tipi uporabnikov, zato smo poskušali podobo aplikacije čim bolj standardizirati. Zaradi njene preprostosti aplikacija ne potrebuje uporabnikovega predznanja.



Slika 4.17: Prikaz pomoči pri vnosu datuma.

- **Enostavnost vzdrževanja**

Spletne aplikacije za razliko od namiznih aplikacij zahtevajo manj napora za vzdrževanje ter uporabo ne le za razvijalca programske opreme, ampak tudi za končnega uporabnika. Spletna aplikacija se nahaja le na enem mestu (spletnem strežniku), medtem ko je namizna aplikacija nameščena na fizičnih napravah uporabnika in zasede več prostora na lokalnem disku.

Pri spletni aplikaciji vzdrževanja s strani uporabnika ni. Uporabnik ne potrebuje namestitve ali posodabljanja (z izjemo posodabljanja spletnega brskalnika) in tako prihrani čas.

Razvijalci spletne aplikacije imajo dnevni nadzor nad uporabniki. Tudi v našem primeru ima administrator stalen nadzor nad uporabniki ter možnost pregledovanja njihove aktivnosti v aplikaciji. Ker imamo vse na enem mestu, lahko na hitro naredimo popravek, osvežimo podatke ali pa uporabniku zagotovimo zadnjo različico, ne da bi ga motili pri delu. Hkrati na ta način zmanjšamo stroške vzdrževanja.

- **Enoten podatkovni sistem**

Obstajajo sistemi, ki uporabljajo lokalno ločene baze ali pa celo baze, ki so na oddaljenih lokacijah. V tem primeru lahko pride do težav, kadar uporabnik čaka na sinhronizacijo vseh podatkov, saj se uporaba aplikacije močno upočasni.

V sodobnem informacijskem svetu je ključnega pomena, da ima uporabnik ažurno posodobljene podatke vedno pri roki. V ta namen smo uporabili centralno spletno bazo, ki bo skupna vsem uporabnikom. S tem smo odpravili čas čakanja na informacije in tako uporabniku prihranili čas.

4.4 Pomanjkljivosti

Na žalost imajo spletne aplikacije tudi nekaj pomanjkljivosti, ki se odražajo tako na strani uporabnikov kot tudi razvijalcev. V prvem delu bomo na splošno opisali pomanjkljivosti spletnih aplikacij, v drugem delu pa bomo naštetili pomanjkljivosti naše spletne aplikacije, ki je namenjena poletni šoli.

Splošne pomanjkljivosti spletnih aplikacij so:

- **Varnost**

Zadnja leta je varnost spletnih aplikacij močno napredovala, vendar še zmeraj obstajajo nevarnosti, ki lahko ogrozijo spletno aplikacijo. Zavedati se moramo, da s pojavom aplikacije na medmrežju le-ta postane na voljo vsem uporabnikom, med drugim tudi tistim, ki bi nam hoteli škodovati v smislu nepotrebne obremenjevanja strežnika ali s krajo občutljivih podatkov. Poleg napadalcev pa obstaja tudi možnost, da strežnik podatke objavlja na spletu zaradi napačnih nastavitev ali drugih vzrokov.

- **Dostop**

Predpogoj vsake spletne aplikacije je aktivna povezava do spleta. V primeru, da je povezava počasnejša kakor zahteva aplikacija, znajo biti uporabniške izkušnje okrnjene. Zakup hitrejših širokopasovnih povezav je rešitev, ki pa nas stroškovno bolj obremeni.

Za gostovanje ter obdelavo informacij ter zahtev s strani uporabnika je zadolžen strežnik. Ta ob preseženi količini prometa enostavno odpove in ne izvaja vseh zahtev uporabnikov. Tudi tu je možna rešitev v nakupu zmogljivejše opreme, ki je načeloma spet dražja.

- **Razvoj**

Čeprav smo omenili, da spletne aplikacije za razliko od namiznih zahtevajo manj napora za vzdrževanje ter uporabo in so ponavadi tudi cenejše, je v sam razvoj spletnih aplikacij potrebno vložiti precej več truda in znanja. Če vzamemo primerjavo z namiznimi aplikacijami, je razvoj spletne aplikacije dosti težji oziroma v nekaterih primerih še nemogoč do mere, kot so namizne aplikacije. Razvoj spletnih aplikacij nam dodatno otežujeta problem zmogljivosti ter omejenost spletnega brskalnika. Žal so nekatere namizne aplikacije še vedno nujne, saj jim spletne še ne morejo konkurirati v uporabniških izkušnjah ter določenih funkcionalnostih. Kljub modernim tehnologijam, ki dandanes omogočajo razvoj spletnih, nivo razvoja še vedno zaostaja za namiznimi aplikacijami. Toda trend se spreminja in tudi dovršenost spletnih aplikacij je vedno bolj podobna namiznim.

Na kratko bomo povzeli še tri pomanjkljivosti naše spletne aplikacije, na katere smo naleteli v fazi testiranja oziroma se je v času razvoja izkazalo, da bi bilo potrebno uporabiti drugačno tehnologijo za razvoj določenih rešitev:

- **Problem dodeljevanja pravic**

Ob registraciji uporabnika je takojšnja možnost uporabe aplikacije nemogoča, saj ima uporabnik v podatkovni bazi nastavljeno pravico 0, kar za določen čas pomeni, da je še brez pravic. Da se nov uporabnik lahko prijavi v aplikacijo, potrebuje predhodno potrditev s strani administratorja. Le-ta mu pravice ročno ponastavi, kar pa je lahko zamudno ter moteče s strani administratorja v primeru, da več oseb opravi registracijo v enakem časovnem obdobju.

Tudi samo preverjanje pravic v logiki aplikacije je tako, da se ob vsakem vstopu v izbran sklop preverja uporabniške pravice, na podlagi le teh pa se omogoči oziroma zavrne vstop.

- **Problemi varnosti**

Kot vsaka spletna aplikacija tudi ta ni odporna na zlorabo. Tekom razvoja smo opazili, da se ponekod pojavljajo problemi s sejo. Tudi sama povezava ob prijavi v spletno aplikacijo je nezavarovana. To sta glavni pomanjkljivosti, katere bi znali zli uporabniki izkoristiti in nam s tem škodovati.

- **Ni podpore za mobilne naprave**

Dandanes je s porastom pametnih mobilnih naprav mogoče opaziti trend naraščanja razvoja mobilnih aplikacij. Z razvojem le teh pa narašča tudi število uporabnikov. Večino spletnih aplikacij ima zato posebno različico, ki je na voljo za uporabo na mobilnih napravah v vsej svoji funkcionalnosti.

Naša spletna aplikacija še ni prijazna za uporabo na mobilnih napravah v vsej svoji funkcionalnosti, zato tu izgubimo na fleksibilnosti dostopa. V prihodnosti bi bilo vsekakor potrebno nadaljevati razvoj v smeri podpore za mobilne naprave.

4.5 Izboljšave spletne aplikacije

Na tem mestu bomo opisali morebitne izboljšave spletne aplikacije, ki bi jih z dodatnim časom, izkušnjami ter poglobljenim tehničnim znanjem lahko realizirali v prihodnosti. Bistvene izboljšave v smislu funkcionalnosti ter varnosti aplikacije so:

- razvoj spletne aplikacije v ogrodju CodeIgniter,
- integracija ACL,
- sinhronizacija z Google koledarjem,
- možnost dodajanja slik, videov, povzetkov,
- prehod na CMS sistem,
- uporaba spletne aplikacije preko SSL povezave.

4.5.1 Razvoj spletne aplikacije v ogrodjem CodeIgniter

Ena izmed izboljšav trenutno razvite spletne aplikacije bi vsekakor bila, da se trenutna arhitektura aplikacije spremeni na trodelno arhitekturo model-pogled-krmilnik. To bi nam omogočila uporaba ogrodja CodeIgniter, ki bazira na tem tipu arhitekture.

Razvoj, ki bazira na MVC arhitekturi, bi bil velika prednost za spletno aplikacijo, saj bi koda postala bolj pregledna ter bolj strukturirana, s tem pa bi bilo tudi vzdrževanje lažje. Če bi se v prihodnosti izkazale potrebe po dodatnem razvoju spletne aplikacije, bi si s tem tipom arhitekture prihranili ogromno časa in truda. V primeru, da bi želeli trenutno aplikacijo razširiti tudi na mobilne naprave, bi bilo tako potrebno spremeniti le poslovno logiko, logika modela pa bi ostala nespremenjena oziroma je ne bi bilo potrebno podvajati [4].

Ogrodje CodeIgniter ima med drugim implementiran tudi razred Seja. Omenjen razred omogoča vzdrževanje trenutnega uporabniškega stanja ter sledenje uporabniškim aktivnostim, ki se izvajajo v času uporabe spletne aplikacije. Ogrodje omogoča tudi shranjevanje piškotkov v bazi, ki se ob naslednji uporabi primerja z uporabnikovim trenutnim piškotkom. S tem bi pridobili na varnosti in zaščiti spletne aplikacije. Na tem mestu je potrebno omeniti, da ta razred ne operira s PHP izvornimi sejami, ampak kreira svoje podatkovne seje, ki so razvijalcu bolj prijazne.

4.5.2 Integracija ACL

Med razvojem spletne aplikacije smo naleteli na probleme z dodeljevanjem pravic uporabnikom. Možna rešitev bi bila integracija seznama kontrole dostopa ali ACL (angl. Access Control List).

ACL je del programske logike, ki skrbi za področje upravljanja pravic uporabnikov spletne aplikacije oziroma določa dostop do funkcionalnosti, ki jih lahko izvedejo znotraj aplikacije (slika 4.18). Opisan tip logike je pomembna lastnost PHP jezika na področju varnosti. Ponavadi je to seznam s podatki, ki vsebujejo pravice uporabnikov do programov, procesov ali datotek. Tudi mrežni administratorji uporabljajo to vrsto sistema, da zagotovijo kontrolo nad prometom in s tem varnejše omrežje.

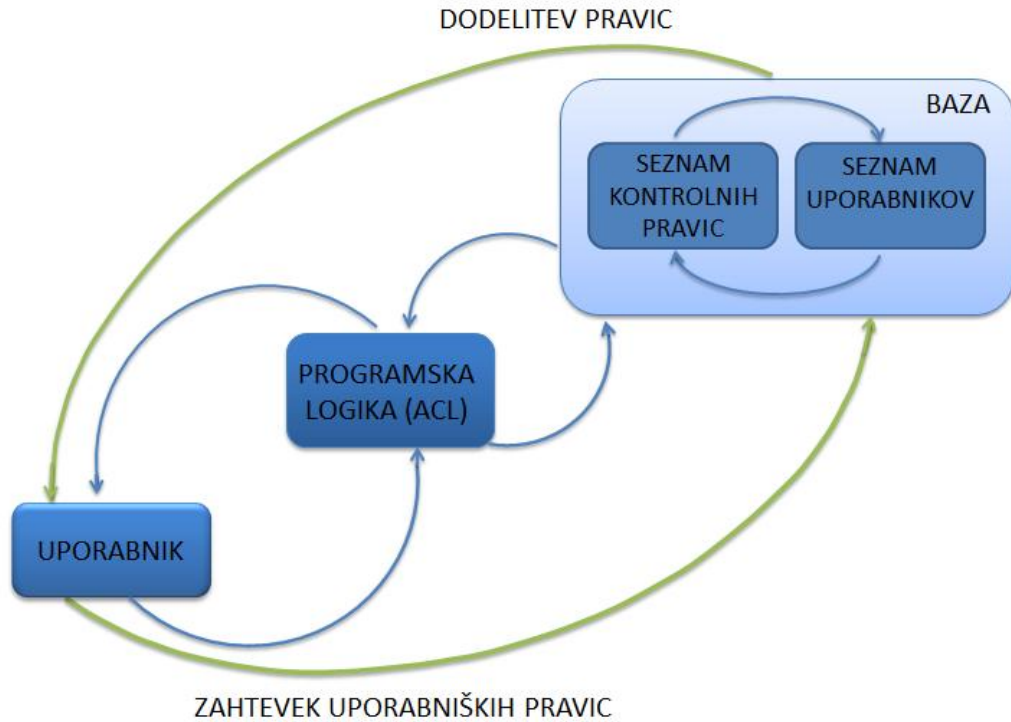
Da pa ACL model zaživi tudi v praksi, je potrebno spremeniti logiko podatkovne baze in programsko logiko. V podatkovni bazi je potrebno ustvariti dva ločena entitetna tipa. Prvi entitetni tip bi bila baza z osebami, ki je že obstoječa v naši podatkovni bazi. Druga entiteta pa je tabela z vsemi tipi pravic uporabnikov, ki bi jo bilo potrebno na novo ustvariti.

V programski logiki bi morali dodati razred, ki bi vseboval metodo za preverjanje pravic uporabnikov s podatkovne baze.

Razred za preverjanje praviic uporabnikov

```
<?php
Class Acl {
    private $baza;
    //incializacija podatkovnega objekta
    function __construct() {
        $this->baza = new baza;
    }
    function check($permission,$userid,$group_id) {
        //preverjanje uporabniških pravic
    }
}
?>
```

V našem primeru bi bila možna implementacija seznama tipa uporabnikov (administrator, skrbnik, predavatelj) katerim bi bile dodeljene posamezne pravice. S tem bi imeli rešitev v primeru, da bi se v prihodnje pojavil nov tip uporabnika. Preprosto bi ga vstavili v seznam ter mu določili pravice. Brez pretiranega posega v kodo bi struktura ostala še naprej pregledna.



Slika: 4.18: Shema ACL logike.

4.5.3 Sinhronizacija z Google koledarjem

Skrbnik konference ima možnosti vnosa urnika v koledar. Obstaja možnost, da obstoječi spletni koledar sinhroniziramo z Google koledarjem.

Za sinhronizacijo bi moral obstoječi spletni koledar omogočati generiranje urnika v format, ki je prijazen Google koledarju, npr. XML. Z opisano funkcionalnostjo bi aplikacijo obogatili za:

- sinhronizacijo koledarja na uporabniških mobilnih napravah,
- izvoz koledarja, ki bi služil kot projekcija na konferencah. V ta namen bi imeli tako dijaki kot študentje možnost dodatnega vpogleda na urnik, ki bi jih opozoril na spremembe, ki so se zgodile v realnem času, zaradi različnih vzrokov (na primer: nedostopnost dvorane).

4.5.4 Možnost dodajanja slik, videov in povzetkov

Trenutno je komunikacija med študenti in predavatelji zelo okrnjena. Preprosto obliko komunikacija bi lahko razširi ter jo izpopolnili.

Koristno bi bilo, da bi predavatelju omogočili možnost dodajanja gradiv, ki so se uporabljala na predavanjih ali ostalih elektronskih virov kot so slike in videi. Dodatno, bi lahko predavatelj dodal kratke kvize, ki bi preverjali znanje študentov ter dijakov. Na tem področju ostaja še veliko možnosti razvoja funkcionalnosti, ki bi obogatile uporabniško izkušnjo in naredile aplikacijo bolj uporabno.

4.5.5 Prehod na CMS sistem

Sistem za upravljanje vsebin ali CMS (angl. Content Management System) omogoča urejanje ter vzdrževanje spletnih vsebin brez znanja označevalnega jezika HTML.

Oseba, ki vzdržuje stran, lahko samostojno in brez programerskih veščin osvežuje vsebino spletne strani ter spreminja in dodaja besedila, slike in ostale večpredstavnostne vsebine. Podjetja ter posamezniki želijo redno ažurirane strani, zato je uporaba CMS sistema zelo razširjena.

Sistemi so namenjeni različnim tipom strani, saj nudijo zelo raznolik spekter funkcionalnosti.

Prednosti uporabe spletne aplikacije poletne šole s CMS sistemom so:

- Omogoča relativno dobro ter preprosto vzdrževanje spletne strani širši množici uporabnikov, tudi tistim, ki nimajo poglobljenega znanja na področju spletnih tehnologij.
- Lažje urejanje vizualne podobe besedil na spletnih straneh, možnost integracije različnih CSS stilov, bogata ponudba že narejenih spletnih strani z dizajnom pa je brezplačna.
- Enostavna namestitev ter upravljanje sistema.
- Princip urejanja vsebine preko WYSIWYG (angl. what you see is what you get) urejevalnika, ki omogoča predogled končne vsebine spletne strani. Ta princip omogoča razvoj tudi novim skrbnikom, brez poglobljenega znanja na področju razvoja spletnih aplikacij.
- Možnost urejanja kjerkoli ter kadarkoli.
- Upravljanje vseh sprememb poteka preko uporabniškega vmesnika (poseg v programsko logiko ni potreben).

- Strukturiranost programske kode sistema je pregledna, zato razvijalci nimajo težav pri razvoju dodatnih funkcionalnosti oziroma komponent.
- Sistem omogoča vodenje evidence pravic uporabnikov. Z uporabo evidence lahko ustvarimo hierarhični red uporabe CMS sistema.
- CMS nudi tudi širok spekter dodatnih funkcionalnosti oziroma modulov, ki jih enostavno namestimo.

Najbolj razširjeni CMS sistemi med uporabniki so:

- Wordpress,
- Joomla,
- Drupal,
- SilverStripe.

4.5.6 Uporaba spletne aplikacije preko SSL povezave:

V poglavju pomanjkljivosti spletne aplikacije smo omenili, da bi bil potreben še nadaljnji razvoj na področju varnosti. K izboljšanju varnosti bi v večji meri prispevala uporaba tehnologije SSL.

SLL (angl. Secure Layer Socket) je protokol, ki šifrira podatke, ki se prenašajo s komunikacijo med brskalnikom ter spletnim strežnikom. Spletni naslov zaščiten z SSL protokolom spoznamo po začetnem URL naslovu spletnega mesta, ki se začne s HTTPS. Protokol za varno komuniciranje omogoča:

- kriptografsko zaščito, ki varuje pred prestopanjem občutljivih podatkov,
- identifikacijo tako na strani strežnika kot tudi odjemalca.

V primerjavi z nešifrirano spletno stranjo protokol SSL zagotavlja večjo varnost ter zasebnost pri uporabi spleta in zmanjša tveganje, da bi tretja oseba imela možnost, da prestreže in zlorabi podatke v zle namene. V primeru, da spletna aplikacija operira z občutljivimi podatki kot so osebni podatki ter plačilne transakcije, tveganje zlorabe občutno zmanjšamo z uporabo SSL protokola [2].

V nadaljevanju bomo opisali kratek povzetek vzpostavitve komunikacije med strežnikom ter odjemalcem preko SSL protokola. Strežnik in odjemalec se dogovorita med seboj o tipu šifriranja. Vsak s svojim elektronsko podpisanim potrdilom (certifikatom) izkažeta drug drugemu svojo identiteto. Potrdili sta izdani (overjeni) s strani certifikatne agencije, kateri ji obe strani zaupata. Ključ za šifriranje seje med strežnikom in odjemalcem se vsakič znova izmenja s pomočjo algoritma za izmenjavo ključev. Po uspešni izmenjavi se SSL povezava obnaša kot običajna TCP povezava.

Čeprav naša spletna aplikacija ne operira s plačilnimi transakcijami, bi vseeno lahko izboljšali varnost prijave v spletno aplikacijo na sledeč način:

- Pridobitev elektronskega potrdila s strani certifikatne agencije, ki potrjuje našo identiteto kot lastnika spletnega strežnika.
- Namestitev elektronskega potrdila ali certifikata v spletni strežnik, kjer je nameščena naša spletna aplikacija. S tem omogočimo strežniku, da bo vzpostavil varno šifrirano povezavo z uporabnikovim brskalnikom ob obisku naše aplikacije.
- Na koncu moramo določiti strani, katere želimo zaščiti s protokolom SSL. V našem primeru bi bila to stran, kjer se uporabnik prijavi v spletno aplikacijo.
- Za uspešno prijavo mora uporabnik predhodno namestiti digitalno potrdilo na svoj spletni brskalnik.

5. Sklepne ugotovitve

Namen diplomskega dela je bil razvoj spletne aplikacije za obvladovanje urnika poletne šole, ki bi omogočala uporabnikom (organizatorju, administratorju, skrbniku, predavatelju) vnos za lažje komuniciranje ter informiranje med uporabniki ter študenti oziroma dijaki.

Spletno aplikacijo smo uspešno razvili. V fazi načrtovanja smo si koncept dobro zamislili, saj se tekom razvoja spletne aplikacije nismo soočali z večjimi težavami. Uporabniku smo uspešno omogočili splošno administracijo konference, pregledno urejanje terminov v koledarju in omogočili komunikacijo med poslušalci in predavatelji v obliki foruma.

Uporaba kombinacije programskih jezikov PHP ter JavaScript knjižnice jQuery se je izkazala za uspešno ne le pri realizaciji razvoja funkcionalnosti spletne aplikacije, ampak tudi pri sami vizualni podobi. Čeprav je bila v preteklosti uporaba JavaScripta pri razvoju spletnih aplikacij močno okrnjena zaradi omejitev spletnih brskalnikov in slabše zmogljivosti računalnikov, je knjižnica jQuery dokaz za vse večjo priljubljenost uporabe med razvijalci.

Pri naši spletni aplikaciji, smo prednosti te knjižnice uporabili za manipulacijo podatkov večjega obsega, ki smo jih prikazovali v obliki urejenega seznama (jqGrid). Hitrejše delovanje aplikacije smo dosegli z uporabo tehnike AJAX, ki omogoča razbremenitev komunikacije med aplikacijo in strežnikom in s tem pohitri delovanje. Hitro delovanje aplikacije je ključnega pomena za sodobnega uporabnika.

V ogrodju CodeIgniter smo razvili spletne storitve (angl. web service), ki nam omogočajo komunikacijo s spletnim strežnikom. Prednost uporabe spletnih storitev je v sami univerzalnosti le teh. V primeru, da razvijalec želi enako storitev uporabiti še enkrat, mu je dosegljiva na predhodno določenem URL naslovu. Dodatna prednost razvoja spletnih storitev je, da uporabljajo trnivojsko arhitekturo MVC. Tu smo si pridobili bogate izkušnje z logiko razvoja aplikacij v tem tipu arhitekture. V prihodnosti bi morali za izboljšanje aplikacije prilagoditi celotno logiko tako, da bi ustrezala modelu MVC. S tem bi omogočili večjo preglednost in strukturiranost programske kode ter omogočili lažje vzdrževanje aplikacije.

Pri razvoju spletne aplikacije smo utrdili znanje, ki smo ga dobili v času študija na fakulteti za računalništvo in informatiko s področij spletnih tehnologij, računalniških komunikacij ter podatkovnih baz.

V primeru, da se bo spletna aplikacija uporabljala v namene organizacije poletne šole, bomo z razvojem nadaljevali v smeri, da omogočimo uporabnikom čim bolj prijetno izkušnjo.

6. Kazalo slik

Slika 2.1: Orodje NuSpherePhpED.	3
Slika 2.2: Prikaz načrtovanja konceptualnega modela podatkovne baze za spletno aplikacijo v orodju PowerDesigner	4
Slika 2.3: Primer klica REST storitve	5
Slika 2.4: Shema MVC arhitekture	9
Slika 2.5: Prikaz vmesnika phpMyAdmin.....	10
Slika 3.1: Diagrame primera uporabe za uporabnika z administratorskimi pravicami	14
Slika 3.2: Diagrame primera uporabe za uporabnika s skrbniškimi pravicami.....	15
Slika 3.3: Diagrame primera uporabe za uporabnika s pravicami predavatelja.	15
Slika 3.4: Podatkovni model spletne aplikacije.....	16
Slika 4.1: Vnosna maska za registracijo.....	20
Slika 4.2: Vnosna maska za prijavo v spletno aplikacijo	21
Slika 4.3. Uporabniški vmesnik za izbiro sklopa	21
Slika 4.4: Vnosna maska za ponastavitev ozabljenega gesla.	22
Slika 4.5: Vnosni obrazec za vnos novega dogodka.	24
Slika 4.6: Vnosna ob urejanju dogodka.....	25
Slika 4.7: Vnosna maska za dodajanje novega obvestila v seznam	26
Slika 4.8: Seznam trenutnih obvestil izbrane konference.....	26
Slika 4.9: Pojavno okno z logotipom	27
Slika 4.10: Vnosna maska za dodajanje sekcij konferenci.....	27
Slika 4.11: Uporabniški vmesnik za dodajanje novih sponzorjev.....	28
Slika 4.12: Uporabniški vmesnik za administratorja urnika	29
Slika 4.13: Uporabniški vmesnik za dodajanje nove sekcije.....	30
Slika 4.14: Primer rikaza sekcij na urniku koledarja.....	31
Slika 4.15: Dialog za brisanje sekcije.....	31
Slika 4.16: Prikaz dialoga med predavatelji in poslušalci	32
Slika 4.17 Prikaz pomoči pri vnosu datuma	34
Slika 4.18 Shema ACL logike	39

7. Viri

- [1] M. McGrath, Mike, *JavaScript in easysteps*, UnitedKingdom:Computer Step, 2006, pogl. 1.
- [2] N. Krawetz, *Introduction to networksecurity*, Boston: Charles RiverMedia, 2007, pogl. 19.
- [3] D. Oliver, M. Morrison, *HTML and CSS in 24 hours (7th edition)*, USA: Sams Publishing, 2006, pogl. 1,12, 13.
- [4] K. Hadlock, *AjaxforWebApplicationDevelopers*, USA: Sams Publishing, 2007.Pogl. 1,3,4.
- [5] (2012) Web dizajn. Dostopno na: <http://sl.napraisajt.com/web-razvoj/kaj-je-php.html>.
- [6] (2012). Presentia. Dostopno na: <http://www.presentia.si/baza-znanja-helpdesk/2008/kaj-je-php>.
- [7] (2012). Wikipedia. Apachec server. Dostopno na: http://en.wikipedia.org/wiki/Apache_HTTP_Server.
- [8] (2012). MySQL. Dostopno na: www.mysql.com.