

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Filip Kocijan

RAZVOJ SUSTAVA ZA UPRAVLJANJE
KONFERENCIJOM

DIPLOMSKI RAD

Varaždin, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Filip Kocijan

Matični broj: 42598/13-R

Studij: Informacijsko i programsko inženjerstvo

RAZVOJ SUSTAVA ZA UPRAVLJANJE
KONFERENCIJOM

DIPLOMSKI RAD

Mentor:

Dr. sc. Tihomir Orehovački

Varaždin, studeni 2015.

Sadržaj

1. Uvod	4
2. Postojeća rješenja	6
2.1. OpenConf.....	8
2.2. COMS (Conference Management System)	12
2.3. <i>ConfTool (Conference Management Tool)</i>	17
2.4. Zaključak analize	21
3. Razvoj vlastitog sustava	23
3.1. Korištene tehnologije i alati.....	23
3.1.1. Java.....	24
3.1.2. JSF, PrimeFaces i Maven okviri(frameworks).....	25
JSF.....	25
Maven.....	27
PrimeFaces	28
3.1.3. MySQL.....	28
3.1.4. Ostale tehnologije.....	30
3.2. Dijagrami sustava	31
3.3. Algoritam za automatsku dodjelu dokumenata recenzentima	36
3.4. Razvoj i kôd sustava	39
4. Zaključak	49
5. Popis slika.....	50
6. Literatura	51

1. Uvod

Ovaj rad će se baviti razvojem sustava za upravljanje konferencijom (eng. *Development of a conference management system*). Sustav za upravljanje konferencijom je web-temeljen softver (eng. *Web-based software*) koji pruža organizaciju konferencije - naročito znanstvenih konferencija. Sustavi za upravljanje konferencijom obično olakšavaju najčešće aktivnosti za organizaciju konferencije. Općenito sustav ima interakciju sa korisnicima čije uloge se mogu podijeliti u tri glavne grupe:

- predsjednici konferencija (eng. *Chair*)
- članovi odbora i recenzenti
- autori

Osim tih tipova korisnika, sustav može sadržavati i druge uloge poput općeg sudionika i administratora sustava koji koordinira tim grupama korisnika. Osnovne aktivnosti, odnosno funkcionalnosti, koje sustav mora omogućavati tim korisnicima su sljedeće:

- kreiranja konferencije od strane predsjednika
- dodavanje članova odbora, recenzenta i autora u konferenciju
- dodavanje svojih radova i dokumenata u konferenciju od strane autora
- recenziranje radova od strane recenzenata
- pomoć pri odluci predsjednika od strane članova odbora
- donošenje odluke (prihvaćen, odbijen...) o predanom radu od strane predsjednika

Osim tih osnovnih funkcionalnosti koje sustav mora imati, on može sadržavati i neke dodatne - koje omogućuju lakše, brže i bolje upravljanje konferencijom. To su primjerice: višejezičnosti, anonimne autore za recenzente, slanje elektroničke pošte korisnicima, diskusije o radovima, anonimne recenzije, praćenje dovršenosti recenzija itd.[1]

U zadnjih deset godina sustavi za upravljanje konferencijom postali su svakom godinom sve popularniji. Razlog tome su gore navedene mogućnosti ali i smanjeni troškovi organizacije, dostupnost za sve korisnike (preko interneta), te sve veći broj konferencija na godišnjoj razini u svijetu. Ipak, najveći razlog je što organizatori konferencije imaju moćan i centraliziran alat za upravljanje koji im pruža pristup svim dokumentima i recenzijama u bilo koje vrijeme i na bilo kojem mjestu. [2]

Rast broja konferencija, a time i popularnost sustava za upravljanje potvrđuju i statistički podaci ICCA-e (eng. *International Congress & Convention Association*). ICCA je najviša globalna udruga u industriji za konferencije i jedna od najistaknutijih organizacija u svijetu međunarodnih skupova, sa preko tisuću članova organizacija i kompanija u više od 90 država. Zadnja objavljena statistika (2013.) pokazuje trend rasta broja događaja njihovih članova za 535 u odnosu na prošlu godinu. Sa 11,685 redovnih događaja udruga koji rotiraju najmanje u tri zemlje, iznosi porast od preko 100% u odnosu na 2006. godinu (5315 događaja). [3] Ti podaci ukazuju na popularnost i motivaciju za korištenjem sustava za upravljanje konferencijama sve češće i u sve većem broju u posljednjih nekoliko godina.

Postoji nekoliko rješenja sustava za upravljanje konferencijom koje su dostupne krajnjem korisniku. U prvom poglavlju će biti pregled tih postojećih sustava, njihova obilježja, nedostaci i sl. Uzet će se neki od popularnijih sustava kao što su *COMS*, *OpenConf*, *EasyChair*, *ConfTool* te će svaki od njih biti zasebno analiziran. Ovi sustavi ispunjavaju osnovne potrebe i znatno olakšavaju organizaciju i upravljanje konferencijom. Ipak, još uvijek postoje mnoge funkcionalnosti koje se mogu poboljšati na način da se poboljša kvaliteta i smanji opterećenje na sustav i korisnika. Jedna od takvih funkcionalnosti je proces dodjele dokumenata recenzentima koji će biti posebno obrađen u sljedećem poglavlju.

Nakon toga slijedi poglavlje za razvoj vlastitog rješenja u kojem će se u potpoglavljima opisati korištene tehnologije za implementaciju sustava, te sami razvoj sustava u kojem se vide dijagrami, kôd, funkcionalnosti i izgled. U potpoglavljju korištene tehnologije opisati će se tehnologije i alati poput *Java*, *NetBeans*, *MySQL*-a te okvira *JSF*, *PrimeFaces* i *Maven*-a. Također će se u jednom potpoglavljju obraditi algoritam i njegova implementacija za automatsku dodjelu dokumenata recenzentima. Za kraj slijedi zaključak u kojem se opisuju obilježja i prednosti sustava u odnosu na već spomenuta postojeća rješenja kao i nedostaci te promišljanja o budućem razvoju i korištenju aplikacije.

Sa gore spomenutim podacima ali i izazovima koji mogu nastupiti tokom izrade vlastitog rješenja, motivacije za ovaj rad svakako ne nedostaje. Osim teorijskog dijela rada koji sa sobom nosi nove informacije, podatke i spoznaje tu je i praktični dio koji također nosi svoje izazove. Teorijski dio rada zbog toga pruža dobar uvod za razvoj vlastitoga sustava jer uz to daje i do znanja na koji način je najbolje razvijati sustav, a da se pritom pazi na nedostatke ostalih sustava - ali i najvažnije što vlastiti sustav mora pružati i imati. Izrada sustava je zatim bazirana na vlastito odabranim tehnologijama sa kojima se trebaju ispuniti ti

uvjeti. Naravno, to nije uvijek lako, a naročito ako treba biti bolje od postojećeg. No, na kraju se dobije sustav koji je sposoban upravljati konferencijom na istoj razinu sa nekim poboljšanim mogućnostima. Sustav se kao takav dalje može koristiti od strane svih korisnika te naknadno dograđivati i poboljšavati što ga čini primamljivim za izradu ali i korištenje.

2. Postojeća rješenja

U ovom poglavlju analizirat će se tri rješenja za upravljanje konferencijama koja postoje na tržištu. Iako postoje mnoga rješenja uzeti će se samo neka koja se najčešće koriste i koja su preporučena od strane nekoliko uglednih organizacija i sveučilišta. Sva tri rješenja su na neki način komercijalna, odnosno treba platiti da bi se u potpunosti iskoristile neke ili sve funkcionalnosti i usluge. Budući da se radi o aplikacijama koje nisu sasvim besplatne te je u njih uloženo da bi se moglo zaraditi, na taj se način rješenja mogu uspoređivati. Rješenja koja će se analizirati su sljedeća:

- *OpenConf*[4]
- *COMS (Conference Management System)*[5]
- *ConfTool (Conference Management Tool)*[6]

Osnovni koncept okvira za usporedbe je taj od strane sustava. Sustav je skup dijelova koji tvore kompleksnu ili jedinstvenu cjelinu koja služi nekoj svrsi. U širem smislu to su organizacije, namještaj, računala, automobili ali i softver. Bit sustava je da dijelovi međusobno djeluju učinkovito i usklađeno na takav način da sustav kao cjelina ima korisnu funkciju za jednu ili više subjekata u okruženju sustava.

Da bi se skup međusobno interaktivnih dijelova računao kao sustav, njihova interakcija mora proizvesti koherentno i korisno sveukupno ponašanje. Interakcije se uvijek mogu podijeliti u dijelove koje su korisne za okruženje, zvane funkcije. Sustavne interakcije su po prirodi uvijek komunikacije sa jednom ili više osoba u okruženju sustava. Način na koji su funkcije posložene u vremenu zove se ponašanje sustava. Funkcije, komunikacije i ponašanje su svojstva sustava. Također postoje i druga svojstva sustava poput sigurnosti i razumljivosti (eng. *user-friendliness*), no za analizu spomenutih sustava koristi će se samo neka.[7]

Razlog tome je što se radi o rješenjima koje su implementirane u različitim tehnologijama te se nema pristup potpunom kôdu i funkcionalnostima da bi se moglo ići u

dublje tehničke usporedbe i analize. Zbog toga će se analizirati samo svojstva bitna za krajnjeg korisnika, što je i najvažnije. Na kraju će se napraviti zajednički zaključak od svih rješenja te opisati razlike i sličnosti. Za pojedinu analizu gledati će se na sljedeće mogućnosti korisnika:

- koristi sustav bez ili sa minimalnom podrškom (sustav nema grešaka)
- razumije kako koristiti sustav bez ili sa minimalno podrške i dokumentacije (dizajn, upotrebljivost...)
- ima korist od sustava koju želi (može upravljati konferencijom bez dodatnih problema)

Sve karakteristike će se proći za svaki sustav te uz njih prikazati i prikladna slika zaslona (eng. *Screenshot*).

Sustavi za upravljanje konferencijom obično vode cijeli proces upravljanja u manjim, vremenski ovisnim dijelovima zvanim fazama. Neke faze se mogu preklapati u vremenu dok druge moraju biti striktno sekvencijalne kako koriste podatke postavljene ili generirane tijekom prethodnih faza. Postoje aktivnosti vezane za organizaciju konferencije poput: pronalaska financijske potpore, uređenje dvorane za sjednice prezentacije, aranžiranje službenih večera i ostalih društvenih aktivnosti itd. One su manje-više neovisne o upravljanju sa dokumentima te se mogu izvoditi paralelno. U vrijeme dodjele svi dokumenti moraju biti postavljeni, svi recenzenti registrirani i sve kategorije odabrane tako da funkcionalnost automatske dodjele dokumenata može napraviti najbolje rezultate. Faze revizije i diskusije mogu malo odstupati od vremena budući da nekoji dokumenti mogu već biti recenzirani, dok ostali još čekaju. Faza prihvatanja/odbijanja zatim može trajati dok se svi dokumenti ne procijene te im se dodjeli status.[8]

Gore navedeni sustavi također se provode preko faza, a ovisno o sustavu one mogu varirati u vremenu izvršavanja. Neki, osim samog sustava za upravljanje, pružaju i ostale aktivnosti vezane uz organizaciju konferencije. Njih će se također dotaknuti budući da se radi o cjelokupnom rješenju organizacije konferencije koje dolazi uz sustav.

2.1. OpenConf

Sustav za upravljanje konferencijama *OpenConf* napravljen je od strane *Zakon Group LLC*. Pušten je u pogon 2004. godine i od onda pa do danas (kolovoz, 2015) imao je nekoliko nadogradnji i verzija - trenutna je 6.40. Koristi se u preko 100 zemalja, a neke od organizacija koje ga koriste više od pet godina su IEEE, NCSS, W3C, ACM itd. Sam sustav implementiran je pomoću PHP, JavaScript i MySQL tehnologija.

Karakteristike

Početna stranica sa koje se može pristupiti sustavu je jednostavna i čista, korisnik se može brzo snaći, te doći do željenih informacija i samog rješenja.

The screenshot displays the OpenConf website interface. The top navigation bar includes links for Home, Editions, Demo, Download, Services, Support, Translations, and Pricing & Purchase. The main content area features a large heading "Peer-Review, Abstract and Conference Management" followed by a description of the system's ease of use and its widespread adoption. Below this, there are four buttons: Download, Compare, Demo, and Purchase. To the right, a sample conference program for "Wednesday, 1 December" is shown, detailing sessions, breaks, and lunch. The bottom section contains a "Submission" sidebar with links like Assignment, Review, Bidding, Discussion, Selection, Program, Mobile Program, Proceedings, and Multilingual*. The main area below the sidebar has a "General Information" form for submission details, including fields for Submission Title, Submission Type (Paper, Poster, Short Paper), Author(s) information, and a "Current Release" section highlighting version 6.40.

OpenConf Home Editions Demo Download Services Support Translations Pricing & Purchase

Peer-Review, Abstract and Conference Management

Known for its ease of use and clean interface, OpenConf has powered thousands of events* in over 100 countries.

[Download](#) [Compare](#)
[Demo](#) [Purchase](#)

OpenConf Peer Review & Conference Management System

International Conference

OpenConf Peer Review & Conference Management System

[OpenConf Home](#) [Email Chair](#)

Full Program

Wednesday, 1 December

8:00-9:00 **Welcome & Keynote** (Rooms A & B)

9:00-9:30 **Break**

9:30-12:00 **CONCURRENT SESSION I**

Room A	Room B
Session 1 Chair: Vannevar Bush <i>On Distributed Communications</i> Paul Baran, RAND <i>Why Pascal is Not My Favorite Programming Language</i> Brian Kernighan, AT&T Bell Laboratories	Session 2 Chair: Steve Crocker <i>Alpanwocky</i> R. Merryman, UCSD-CC <i>Up Over Avian Carriers With Quality Of Service</i> D. Waltzman, IronBridge Networks

12:00-14:00 **Lunch**
Buffet lunch will be served in the hotel restaurant

14:00-17:00 **CONCURRENT SESSION II**

Room A	Room B
Session 3 Chair: Donald Davies <i>Communication Networks To Serve Rapid-response Computers</i>	Session 4 Chair: S. Christey <i>As We May Think</i> Vannevar Bush, Office of Scientific Research

Submission »
Assignment
Review
Bidding
Discussion
Selection
Program
Mobile Program
Proceedings
Multilingual*

See more features in our [Editions](#) comparison or try them in our online [demo](#)

General Information

Submission Title:

Submission Type: ☐ Paper ☐ Poster ☐ Short Paper

Author(s)

Author 1

First/Given Name:

Last/Family Name:

Organization:

Country:

Email:

Current Release

6.40

OpenConf News [Facebook](#) [Twitter](#)

Version 6.40 Released
3 August 2015

Version 6.30 Released
14 March 2015

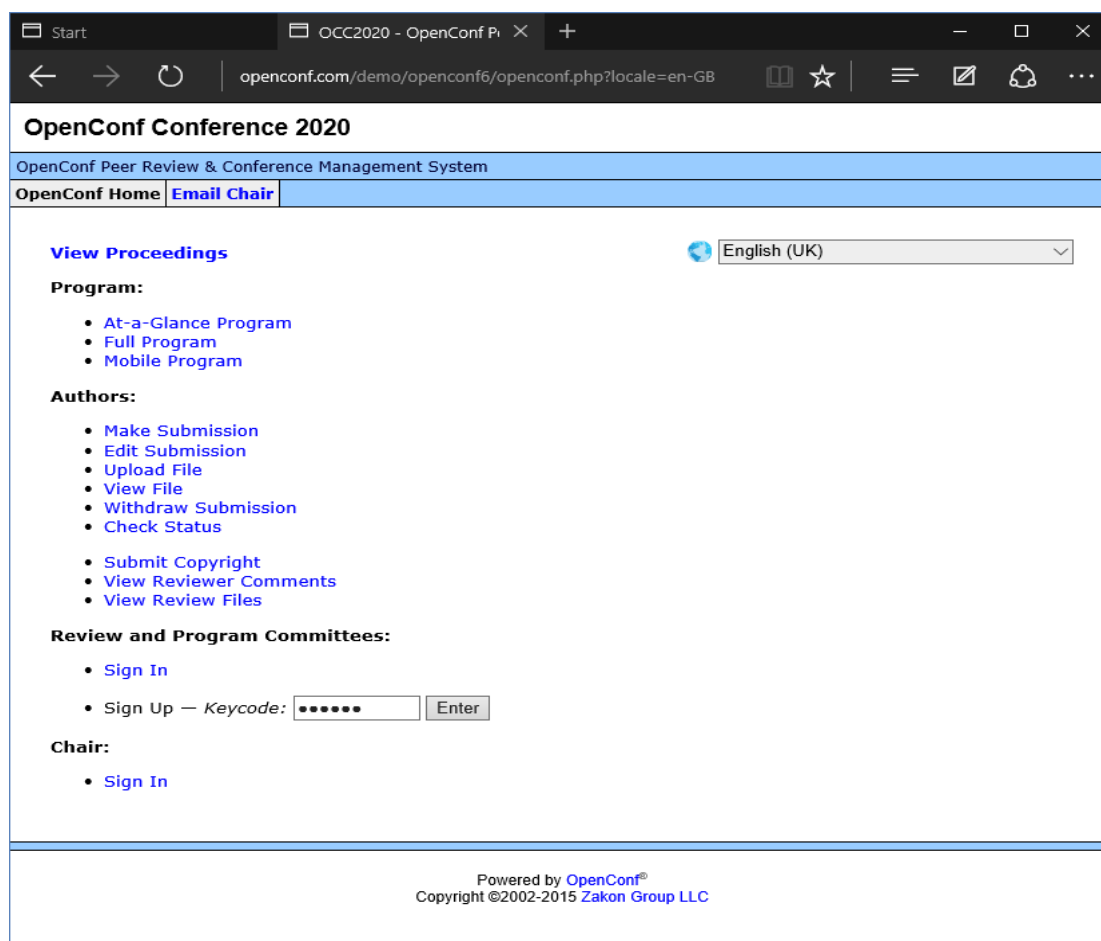
Version 6.20 Released
7 November 2014

Version 6.10 Released
16 July 2014

OpenConf Named IEEE Preferred Partner
13 June 2014

Slika 1: Prikaz početne stranice OpenConf

OpenConf nudi tri izdanja: besplatno, plus i profesionalno. Sa besplatnim izdanjem (eng. *Community edition*) korisnik ima osnovne funkcionalnosti te može voditi konferenciju u kojoj se mogu predati radovi, prenositi datoteke, dodijeliti revizije, pregledati revizije, dodijeliti članovi odbora, prihvatiti/odbiti radovi, izvršiti detekcija konflikata, manipulacija e-poštom te izvoz podataka. Dakle, sa navedenim funkcionalnostima korisnik komotno može upravljati konferencijom u kojoj ne zahtjeva dodatne module koje pružaju ostale verzije. Ako korisnik želi, može isprobati sve module iz profesionalne i plus verzije, koju nude kao demo na stranici popunjenu sa podacima na njihovom serveru. Prednost ovoga je da korisnik može odlučiti da li mu treba samo besplatna ili pa plus/profesionalna verzija. Plus verzija osim osnovnih funkcionalnosti nudi još i diskusiju članova odbora, mogućnost recenzenata da odaberu radove koje žele recenzirati, prijenos datoteke od recenzenata, mogućnost autora da opovrgnu recenziju te dodjeljivanje recenzija od strane članova odbora. Profesionalna verzija osim svega toga još nudi i podršku za prijenos višestrukih vrsta datoteka, *Docoloc* servis za provjeru plagijata, izrada prilagođenih radova, revizija i formi za članove odbora te još nekoliko ostalih modula.



Slika 2: Demo profesionalne verzije sustava

Demo profesionalne verzije sustava izgleda kao na slici 2, a korisnik se može logirati sa spremljenim korisničkim imenom i lozinkom ako želi isprobati uloge recenzenta, člana odbora ili predsjednika. Gumbići za spremanje ili izvršavanje akcije su onemogućeni ali korisnik može pregledati sve funkcionalnosti.

Izgled same aplikacije nije baš lijep i moderan što bi moglo smetati određenoj skupini krajnjih korisnika, primjerice onima koji žele kupiti plus/profesionalnu verziju. Jedina prednost takvog sučelja je u tome da je jednostavno i čisto te se korisnik može lako snaći. Funkcionalnosti su implementirane dobro, te izvršavaju svoje funkcije. Neke su se mogle riješiti na drugačiji način (da pojednostavne rad korisnika), no i ovako rade služe svrsi. Primjer tome je prihvaćanje/odbijanje radova autora od strane predsjednika (slika 3). Klikom na funkcionalnost otvara se nova stranica koja sadrži tabelu radova autora, te njihove ocjene od strane članova odbora; kao i broj revizija, te potvrdu da li je rad prihvaćen ili odbijen. Također sadrži i naziv rada kao link na njega, predsjednik prvo mora kliknuti na rad da bi ga on preusmjerio na drugu stranicu u kojoj vidi podatke o radu te revizore. Zatim ako želi vidjeti svaku reviziju mora posebno kliknuti na svaki od revizora koji ga opet preusmjerava na drugu stranicu sa podacima o reviziji.

Submission Scores

All Acceptance Types All Submission Types Filter

Links:
 Score – Show reviews and accept/reject submission
 Submission ID, Title – Show submission info

Definitions:
 Score = average score across reviews (if no score, ignored)
 Weight = Number of reviews with a score
 Weight* = May include reviews marked as incomplete (unassign)
 Range = Min - Max scores
 Adv Recom = Advocate Recommendation

Select all submissions

	Final Decision	Advocate Recom.	Score	Weight	Range	Submission ID, Title	Sub. Type	Advocate
☒	Accept	Accept	6.00	4	6	13. Computer Programming as an Art	Paper	Charles Babbage
☐	Accept	Accept	6.00	4	6	24. IP over Avian Carriers with Quality of Service	Paper	John von Neumann
☐	Accept	Accept	5.75	4	5-6	27. The Infinite Monkey Protocol Suite (IMPS)	Paper	Alan Turing
☐	Accept	Accept	5.67	3	5-6	15. Host Software	Paper	John von Neumann
☐	Accept	Accept	5.25	4	5-6	1. Man-Computer Symbiosis	Paper	Alan Turing
☐	Accept	Accept	5.25	4	5-6	7. On Distributed Communications	Paper	Charles Babbage
☐	Accept	Accept	5.00	4*	4-6	3. As We May Think	Paper	Alan Turing
☒	Accept	Accept	5.00	4*	5	5. Why Pascal is Not My Favorite Programming Language	Paper	Charles Babbage
☐	Accept	Accept	4.50	4	4-5	11. Communication networks to serve rapid-response computers	Paper	Alan Turing
☐	Accept	Accept	4.25	4	4-5	17. ARPWOCKY	Paper	John von Neumann
☐	Accept	Accept	4.25	4	4-5	19. Act one - the poems	Paper	John von Neumann
☐	Accept	Reject	4.00	4	1-6	21. Standard for the transmission of IP datagrams on avian carriers	Paper	Alan Turing
☐	Reject	Reject	4.00	4	4	25. Y10K and Beyond	Paper	John von Neumann
☐	Accept	Accept	3.75	4	3-4	9. Toward A Cooperative Network Of Time-Shared Computers	Paper	Alan Turing
☐	Accept	Accept	3.75	4	2-5	23. Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)	Paper	John von Neumann
☐	Reject	Accept	3.50	4	1-5	22. RITA -- The Reliable Internetwork Troubleshooting Agent	Paper	John von Neumann

Legend:
 Accept (15)
 Reject (13)
 Pending (0)
 Decision != Recom.

Slika 3: Jedna od funkcionalnosti OpenConf sustava, prihvaćanje/odbijanje radova

Ovo se moglo riješiti elegantnije upotrebom ajax-a i dijalog elemenata. Korisnik bi morao manje klikati, prije bi došao do željenih informacija te ne bi morao „skakati“ između stranica. Osim takvih „nedostataka“, funkcionalnosti same po sebi rade kako bi trebale te se s njima mogu komotno obavljati željene funkcije, te upravljati konferencijom.

Moglo bi se reći da korisnik ima potpunu korist i da nema dodatne troškove, no tu se javlja sljedeći nedostatak. Besplatna i plus verzija zahtjeva od korisnika posjedovanje vlastitog servera na kojem će pokretati sustav. U tom slučaju rješenje se preuzme sa njihove stranice u obliku .zip arhive koja sadrži datoteke sustava. Korisnik zatim mora instalirati sustav na server. Tu može doći do problema ako se korisnik ne razumije u to, te ne može natjerati sustav da proradi. Na stranici ipak nude instrukcije za instalaciju te dodatnu dokumentaciju, no to može biti naporno te zahtijevati više vremena. Još jedan problem u tome je ako korisnik nema vlastiti server ili pristup istome. U tom slučaju korisnik može imati dodatne troškove da bi mogao pokretati sustav. Također server mora biti *Apache* te podržavati *PHP 5.3.7+* i *MySQL 5+* verzije da bi sustav mogao raditi. Druga opcija koju nude korisniku je da se sustav nalazi na njihovom serveru, no treba platiti godišnju pretplatu.

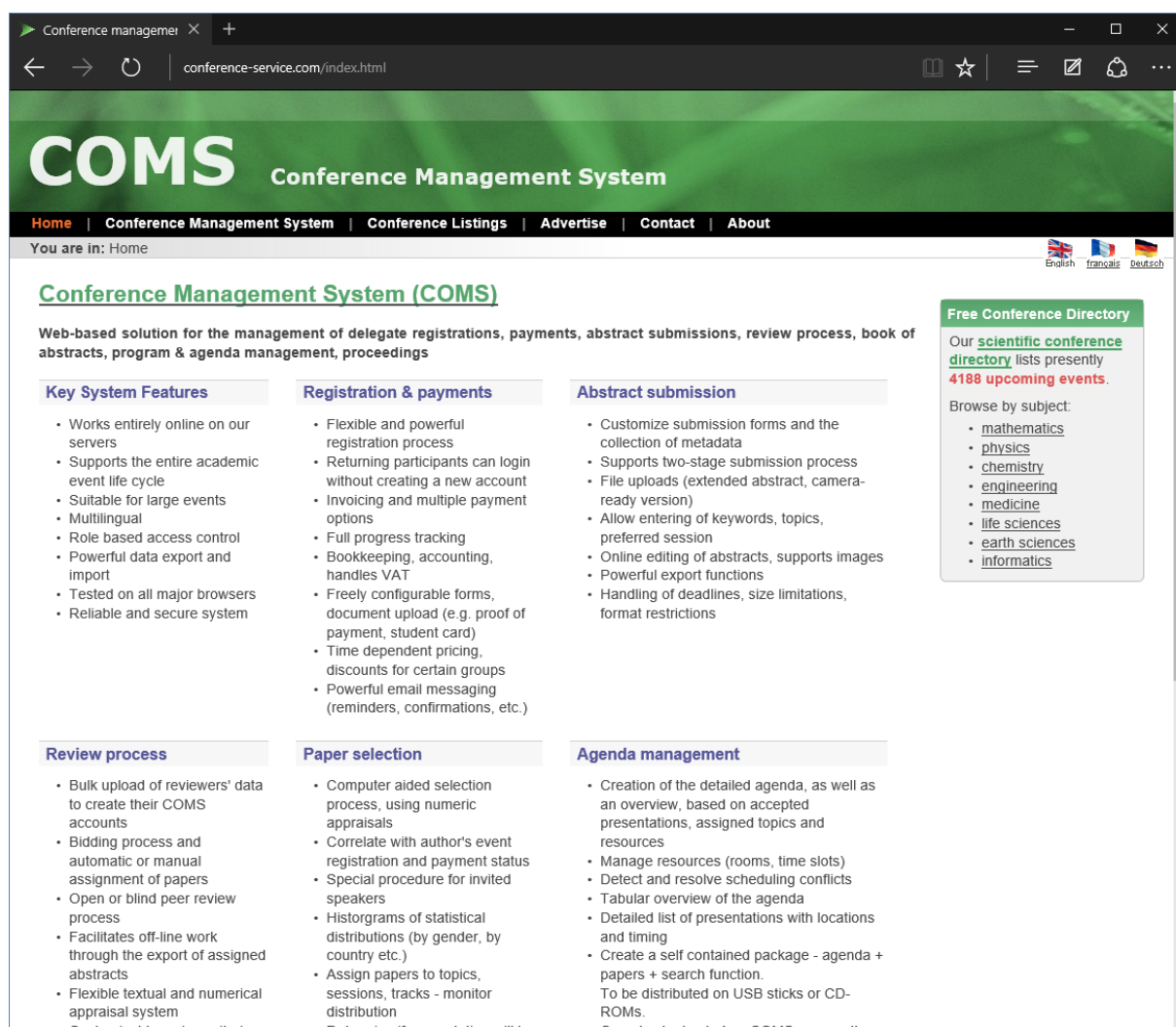
Sustav dakle funkcionira dobro te se sa njime može lako upravljati konferencijom. Sve funkcionalnosti rade kako bi trebale iako ima mjesta za poboljšanje, pogotovo kod izgleda. Korisnik sam odlučuje da li mu je dovoljna besplatna verzija ili treba dodatne module plus/profesionalnih verzija. Najveći nedostatak je ako korisnik ne zna instalirati sustav ili pa nema server za to. U tom slučaju može imati dodatne troškove, a pogotovo ako koristi hosting uslugu od strane *OpenConf*-a.

2.2. COMS (Conference Management System)

COMS je u vlasništvu *Marianne Mandl* iz Austrije, koja sa njime i upravlja. U pogonu je od rujna 2006. godine. Na stranici se nalazi broj pregleda stranica na mjesečnoj bazi iz njihove server log datoteke. Po njoj se može zaključiti da se radi o prilično aktivnom i posjećenom rješenju jer brojka varira od pola milijuna do 900 000 pregleda mjesečno. Stranica također sadrži reference organizacija/sveučilišta iz cijelog svijeta koja su koristila sustav.

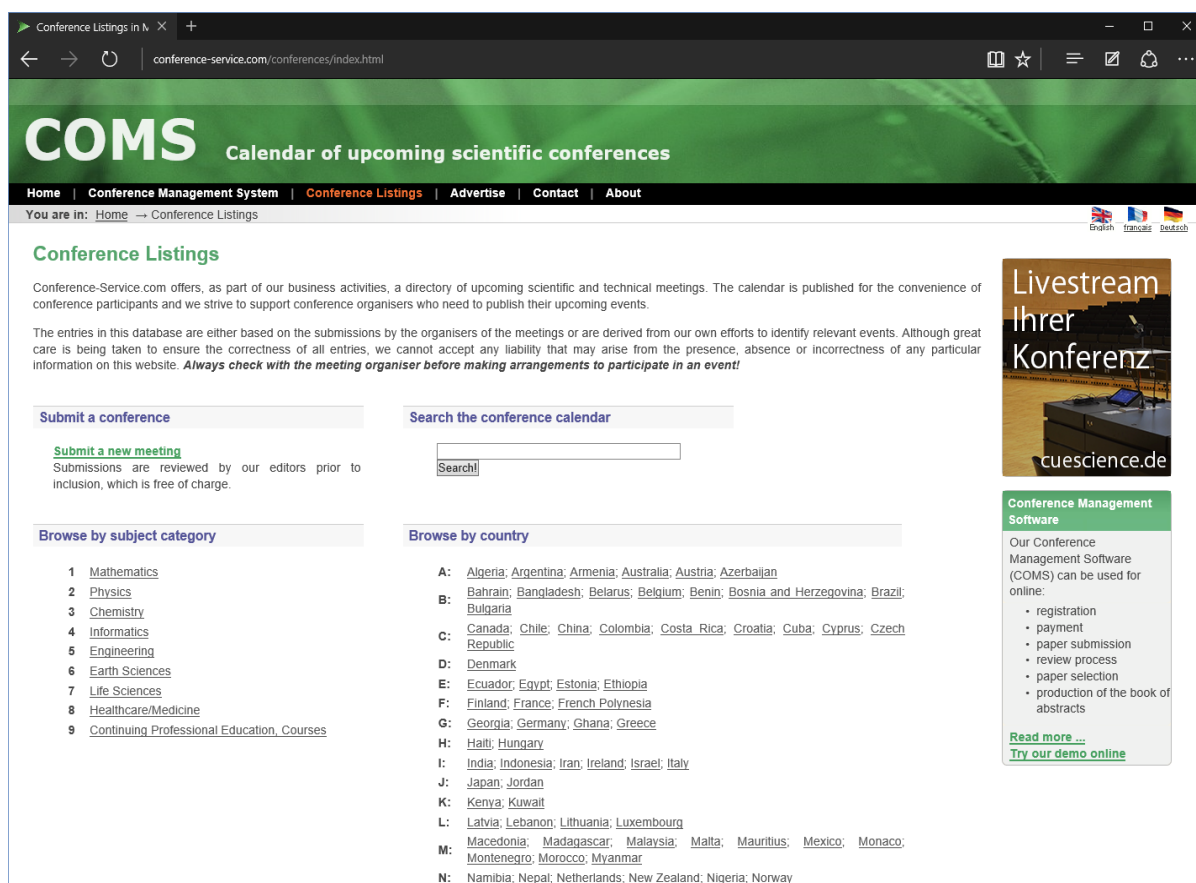
Karakteristike

Sa početne stranice se na početku može odmah vidjeti koje usluge pruža sustav. Za razliku od stranice prijašnjeg sustava, korisnici mogu birati između tri jezika, te se tako možda lakše informirati.



Slika 4: Početna stranica COMS

Stranica nudi dvije usluge, jedna je sami sustav za upravljanje konferencijom, a druga je objava te popis konferencija. Lista konferencije sadrži sve konferencije koje će se održati, sortirane prema predmetu i zemlji konferencije (slika 5). Korisnik može besplatno kreirati konferenciju na način da ispuni obrazac kojeg administrator zatim mora prihvatiti. Nakon što je konferencija prihvaćena ona je vidljiva svima u već spomenutim kategorijama. Ostali korisnici mogu zatim kontaktirati organizatora te konferencije za dogovor oko prisustvovanja. Ovo je dobar način za organizatora konferencije da možda privuče neke vanjske sudionike te ostalim korisnicima da prisustvuju nekoj željenoj konferenciji.



Slika 5: Usluga kreiranja i traženja konferencija

Objavljivanjem konferencije, treba istom i upravljati. Zbog toga se nudi i sustav za upravljanje konferencijom COMS. Za razliku od usluge objavljivanja i traženja konferencije, sustav nije besplatan. Cijena sustava ovisi o veličini konferencije (broju korisnika) te odabranim modulima. Ovisno o broju korisnika sustava postoje šest kategorija: do 100, 101-500, 501-1000, 1001-2000, 2001-4000 te preko 4000 (slika 6). Za svaku od tih kategorija cijena za sustav i pojedini modul se povećava proporcionalno sa brojem korisnika. Troškovi se računaju samo jedanput, a instalacija se može koristiti i nakon završetka konferencije do jedne godine.

COMS 2015 - Demo Site Price List: COMS | Abstr...
conference-service.com/conference-support/coms-price-list.html

COMS Conference Management System

Home | Conference Management System | Conference Listings | Advertise | Contact | About

You are in: Home → Conference Services → Online Conference Management System → Price List

English Français Deutsch

Our services

- Conference Management System
- Features
- Demo
- References
- Price List**
- Order COMS

Price List: Conference Management System (COMS)

Costs are calculated on a per-conference basis, i.e. there is only a single one-time up-front fee.

Included in the price are:

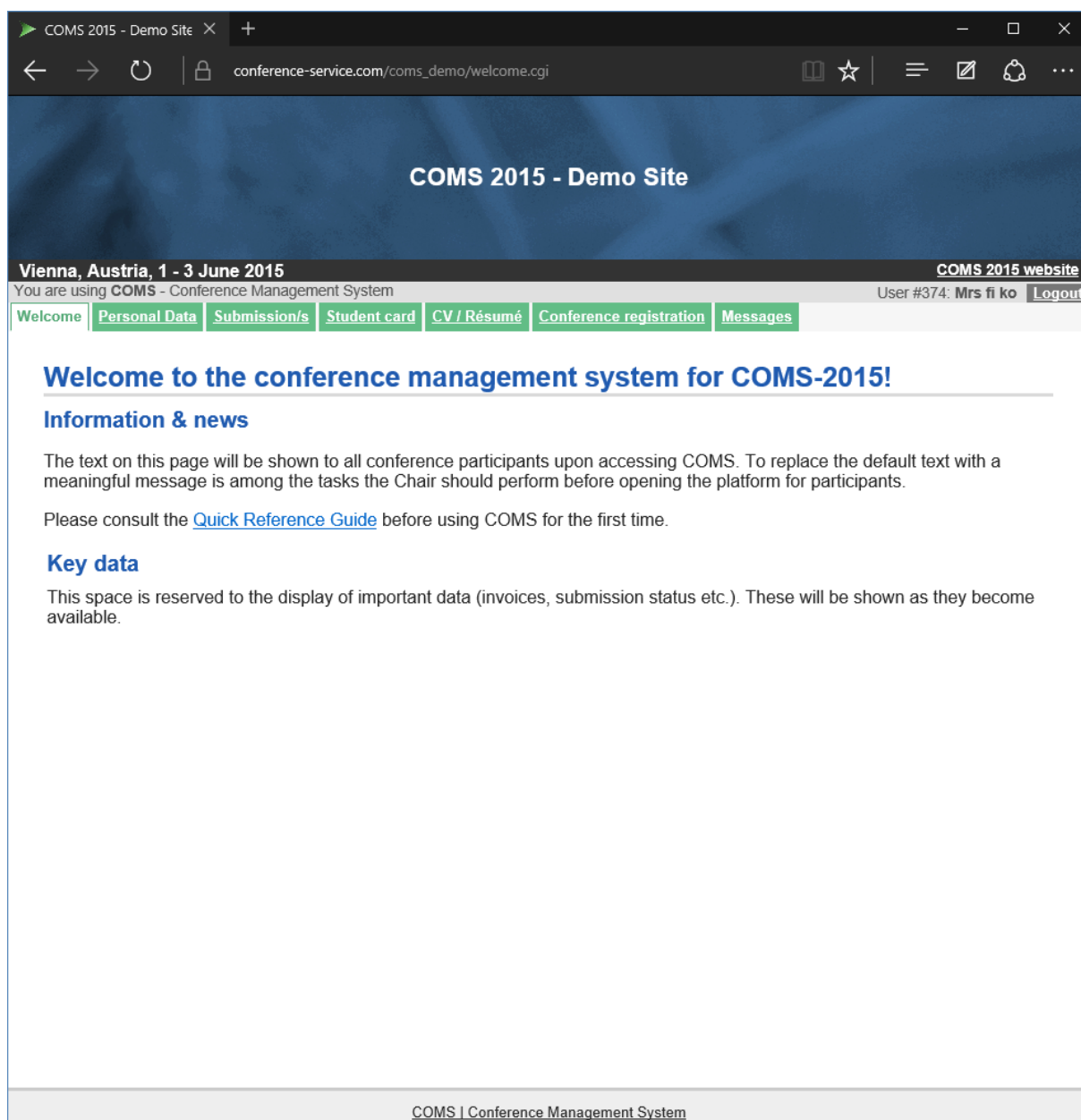
- Usage of the installation from the time of the order and as long as needed after the event has finished (up to 1 year).
- Support by email or phone.
- Daily backups.
- Styling of the banner to match your event website.
- Free placement of a banner ad for your event on our event calendar (optional).

The price depends on the size of the conference and on the modules selected (setup fee plus relevant add-ons). The quoted prices do not include the Austrian value added tax (20% where applicable).

Options	Number of user accounts on COMS					
	up to 100	101 - 500	501 - 1000	1001 - 2000	2001 - 4000	over 4000
Setup fee for a new COMS instance - Basic COMS installation with your custom banner. - Simple event registration (up to two forms with unlimited number of items) - Delegate management; progress tracking; data exports - Targeted, personalised emails to registered users - Additional forms (document upload, freely configurable forms)	€ 150	€ 255	€ 360	€ 470	€ 580	Please enquire
Add-on 1 - Payments and invoicing	€ 150	€ 200	€ 270	€ 380	€ 490	Please enquire

Slika 6: Lista troškova sa opisom pojedinog modula i cijenom za broj korisnika

Demo stranica za testiranje sustava i modula nije javna, te se korisnik prvo mora registrirati da dobi testne podatke za prijavu. To ne bi bio problem da se zatim može testirati cijeli sustav, no nakon prijave na raspolaganju je samo dio sustava za autora. Za testiranje kompletnog sustava sa svim modulima i popunjenog sa testnim podacima potrebno je kontaktirati administratora za podatke za prijavu. Ni to ne bi bio problem ali nakon višekratnog kontaktiranja administratora te čekanja nekoliko tjedana, odgovor se nije dobio. Iako je stranica aktivna te se dnevno vrte konferencije, to može utjecati na korisnika da odustane od samog sustava zbog manjka tehničke podrške ali i nemogućnosti testiranja cijelog sustava.

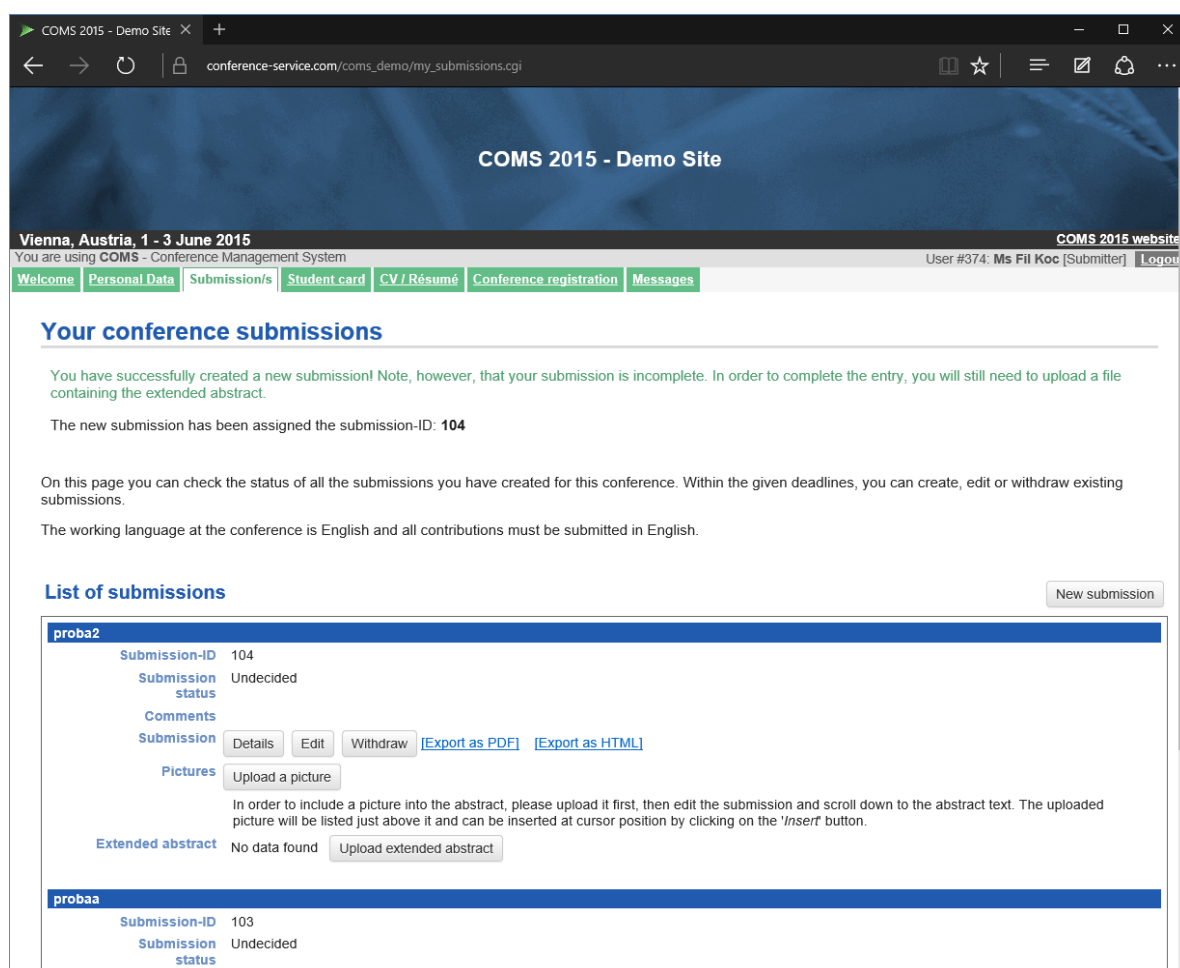


Slika 7: Demo početne stranice sustava, prijavljenog kao autor

Na prije spomenutoj listi cijena (slika 6) uz osnovni sustav i za svaki modul piše opis njegovog sadržaja. Tako postoje osim osnovne instance sustava, deset dodatnih modula. Osnovni sustav sadrži mogućnost korisnika da se registriraju za dolazak na konferenciju, upravljanje konferencijom osnovnim funkcijama, praćenje napretka, slanje e-pošte korisnicima te dodatnim obrascima. Osnovnim paketom korisnik bi mogao upravljati konferencijom i ne bi imao dodatne troškove. No, iz opisa ostalih modula mora odlučiti trebaju li mu neki od njih. Tako postoji modul za plaćanje i fakture koji služi za upravljanje uplatama korisnika za troškove konferencije, modul za dodatne funkcionalnosti kod upravljanja konferencijom, modul za anketiranje korisnika nakon konferencije itd. Korisnik

potom prema opisu modula odlučuje ako mu treba koji od modula te se zatim sustav naručuje pomoću obrasca.

Testiranjem sustava kao autor u demo verziji (slika 7) može se zaključiti da se radi o sustavu sa jednostavnim sučeljem gdje se sve nalazi na jednom mjestu te se može lako navigirati kroz funkcionalnosti preko menija za odabir. Korisnik se brzo i lako prilagodi sučelju i funkcionalnostima koje sustav nudi. Dizajn je jednostavan ali i ljepši nego kod prijašnjeg sustava. Polja za unos se ističu drugačijom bojom, a neki elementi za unos kao *listbox*-evi i *radiobutton*-i imaju dodatne animacije prilikom odabira. Testiranjem sustava kao autor može se lako napraviti novi dokument ispunjavanjem forme za unos novog dokumenta. Nakon toga dobije se lista podnesenih dokumenata za konferenciju. Problem je, kao i u prijašnjem sustavu, nekorištenje *ajax*-a. Tako primjerice kod editiranja dokumenta nakon spremanja promjena ne dolazi do nikakve obavijesti da je dokument uspješno promijenjen. To je moglo biti uspješno i elegantno riješeno sa *ajax* tehnologijom, a korisnik bi znao da je došlo do promjene.



Slika 8: Funkcionalnost dodavanja novog dokumenta kao autor a

Sustav radi kako bi trebao i što se od njega očekuje, barem za dio oko autora. Nažalost, funkcionalnosti za ostale uloge nisu mogle biti testirane zbog već spomenutog razloga ne dobivanja korisničkih podataka za detaljno testiranje. To je jedan zabrinjavajući nedostatak, pogotovo ako korisnik želi isprobati sustav i module da bi ga kupio. No, prednost sustava za razliku od prijašnjeg je da korisnik nakon kupnje nema dodatne troškove zbog servera. Sustav se kompletno sa svim modulima nalazi na njihovim serverima te korisnik nema nikakve brige sa instalacijom.

2.3. *ConfTool (Conference Management Tool)*

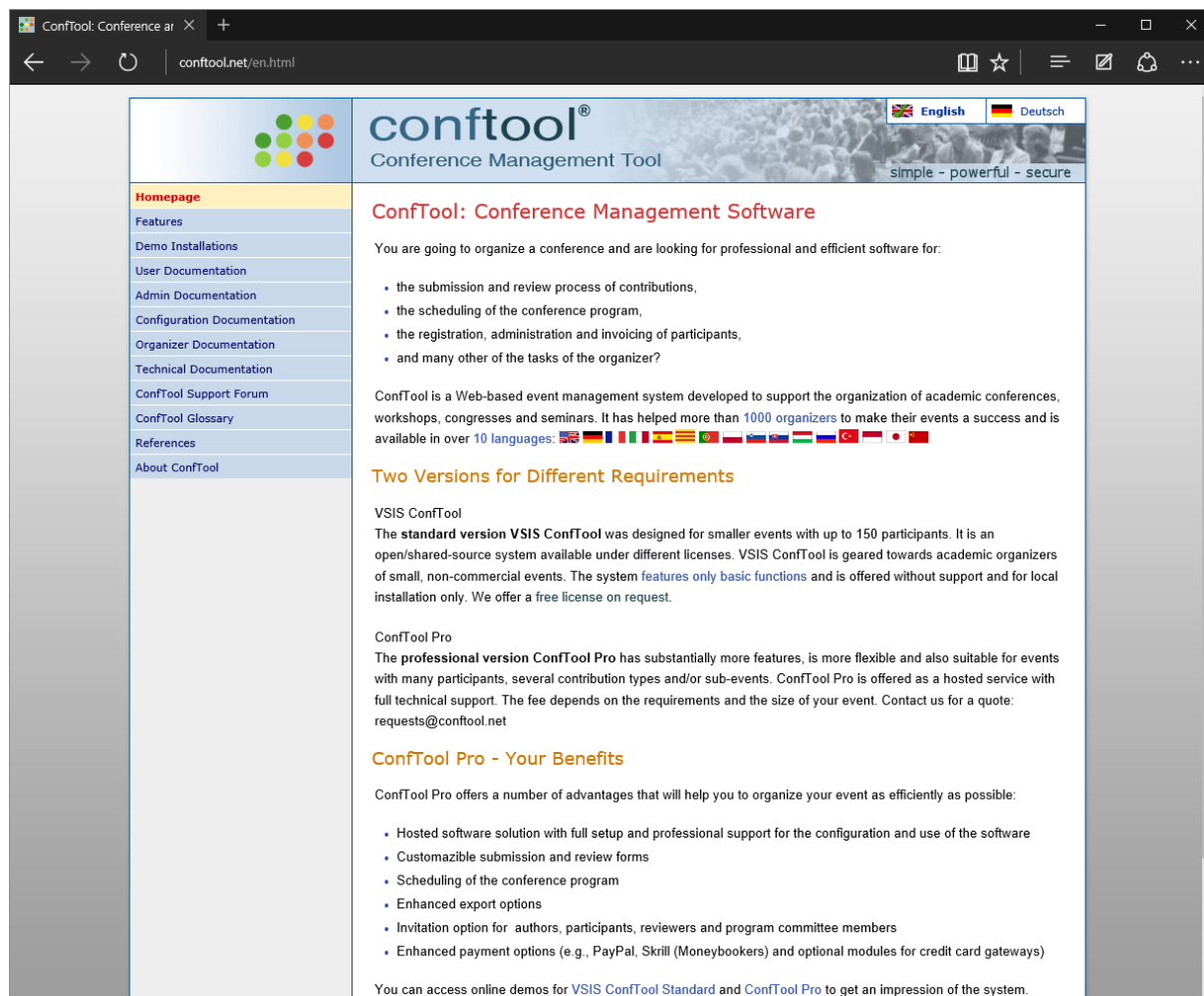
Alat za upravljanje konferencijama *ConfTool* razvijen je i upravljan od tvrtke iz Njemačke, ConfTool GmbH, Hamburg. Sustav je aktivan od 2003. godine dok se par godina prije koristio prototip istog. Do danas (kolovoz, 2015.) sustav je korišten više od tisuću puta, a neke od organizacija/sveučilišta koje su ga koristile su UNIDO, World Bank Group, University of Cambridge, Harvard Medical School itd. Trenutna verzija sustava je 1.8.4. od siječnja 2015. godine.

Karakteristike

Kao i kod prijašnjih sustava, i ovdje je početna stranica vrlo jednostavna i čista te korisnik može brzo doći do informacija i samog sustava. Stranica nudi dva jezika, engleski i njemački, a sve informacije su vrlo dobro i detaljno opisane.

ConfTool nudi dvije verzije za različite zahtjeve korisnika. Prva verzija je VSIS ConfTool, koja je dizajnirana za manje događaje do 150 sudionika. To je besplatna verzija sustava (eng. *open/shared-source*) dostupna pod različitim licencama. VSIS ConfTool je usmjerena prema akademskim organizatorima malih nekomercijalnih događaja. Sustav ima samo osnovne funkcije i nudi se bez potpore i samo lokalne instalacije. Besplatna licenca se može dobiti na zahtjev. Profesionalna verzija ConfTool Pro ima znatno više mogućnosti, fleksibilnija je i stoga pogodna za događaje sa mnogo sudionika, nekoliko vrsta doprinos i/ili poddogađaja. ConfTool Pro se nudi kao hosting usluga uz tehničku podršku. Naknada ovisi o zahtjevima i veličini događaja. ConfTool Pro tako osim osnovnih funkcionalnosti nudi još i višejezično sučelje, opsežne sigurnosne postavke, uvoz podataka, funkciju nadmetanja za recenzente, prilagodljivi obrasci za recenziju, više mogućnosti plaćanja za korisnika (trošak održavanja konferencije), pozivanje sudionika itd. Za ConfTool Pro nije specificirana cijena

već se mora kontaktirati administrator da se dobije ponuda. Isto vrijedi i za besplatnu verziju. Instalacijski paket odnosno besplatna licenca se dobije nakon odobrenja administratora.

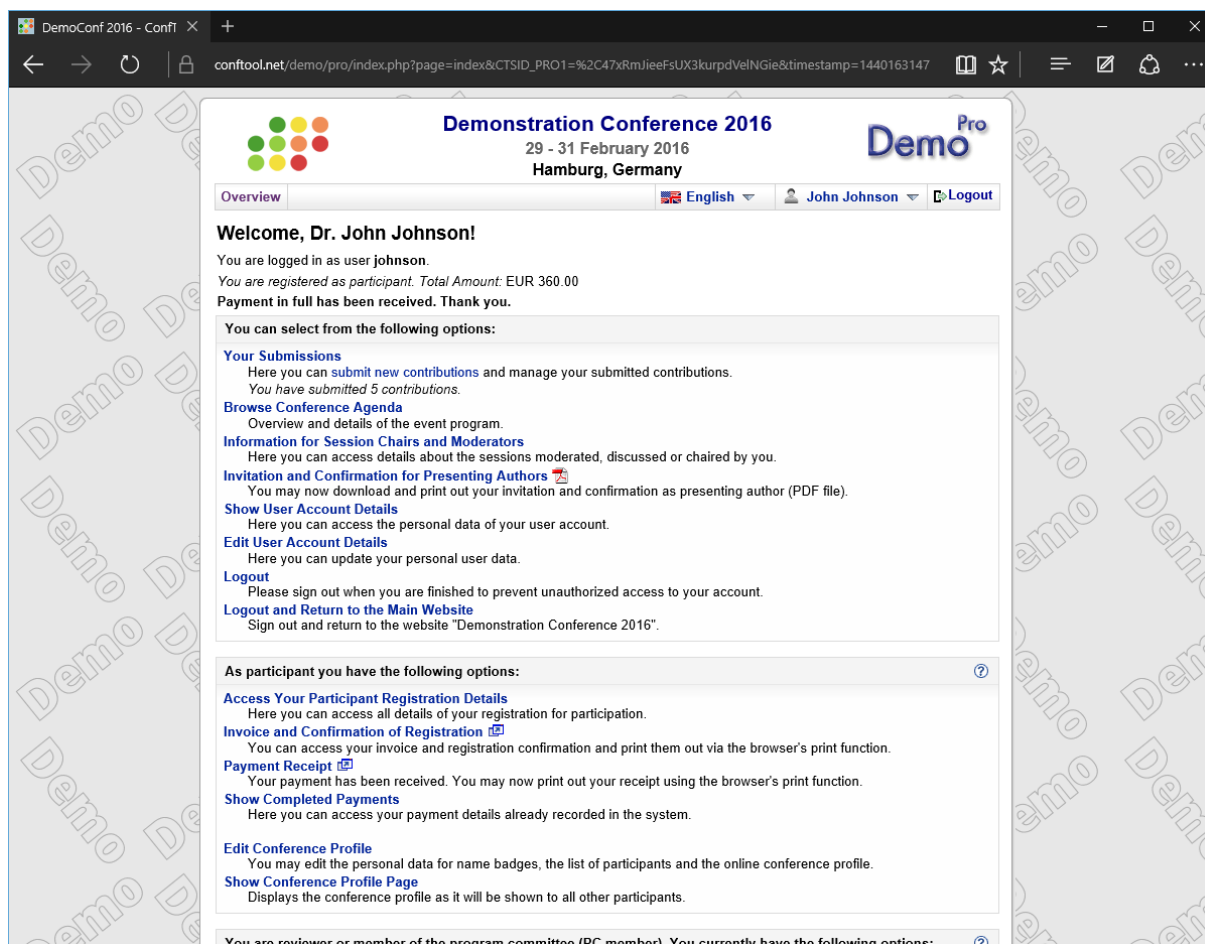


Slika 9: Početna stranica ConfTool-a

Za testiranje sustava nude se besplatna i pro demo verzija kao linkovi na stranici. Besplatna verzija se može testirati u deset različitih jezika ali se u samoj aplikaciji ne mogu mijenjati jezici. Pro verzija može imati više jezika koji se mogu mijenjati. Sve demo verzije popunjene su sa testnim podacima, a korisnik se može prijaviti u aplikaciju sa svojim podacima (nakon registracije) ili koristiti njihove korisničke podatke koji se dobiju na samoj stranici. Također nude detaljnije testiranje sustava sa vlastitim konfiguracijama, u tom slučaju se treba kontaktirati administrator te pričekati par dana da se instalira te dobije osobna demo verzija.

Prijavom u pro verziju demo sustava sa njihovim podacima dobije se uloga recenzenta i člana programskog odbora. Pro verzija se u izgledu malo razlikuje od besplatne verzije dok funkcionalnosti izgledaju i rade isto (koje postoje u obje verzije) te se na njih dolazi klikom na pojedini link (slika 10). Slično kao i kod prijašnjih sustava, sučelje nije baš moderno te je u

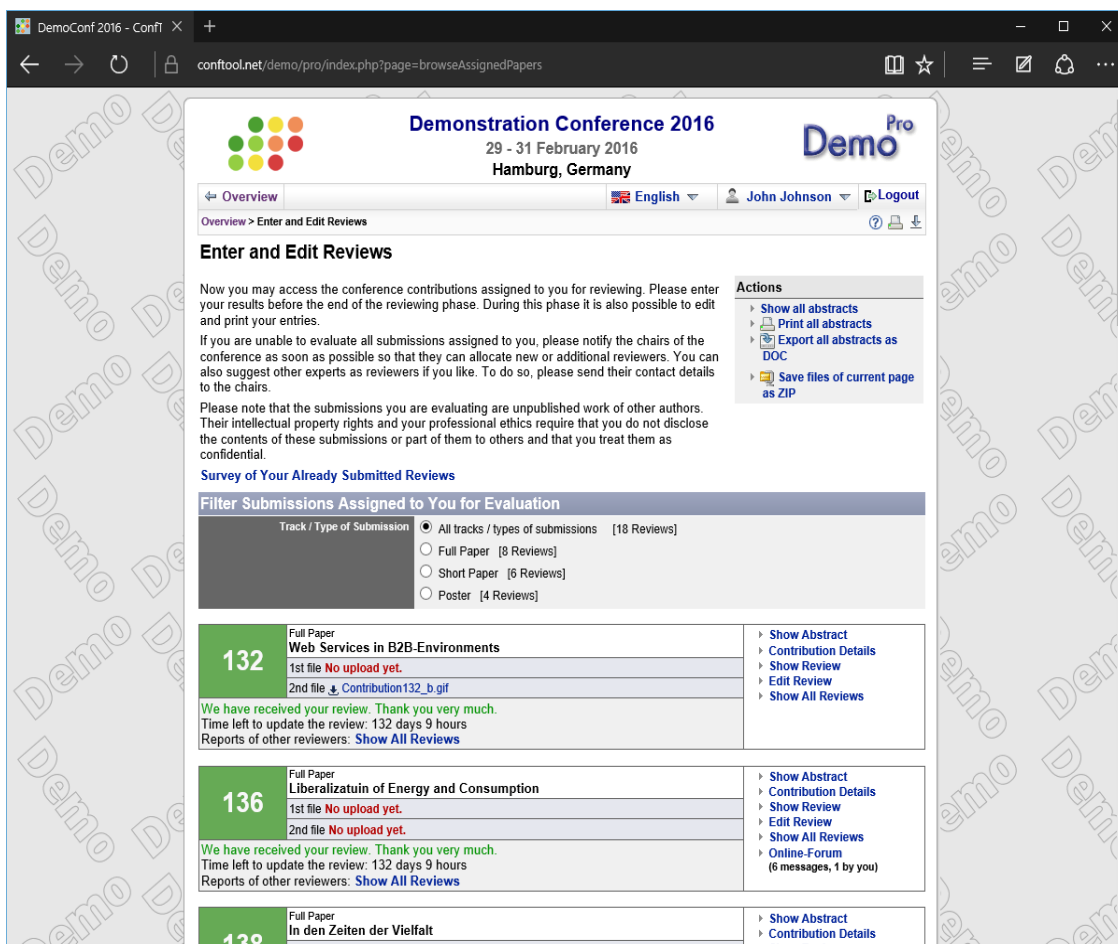
ovom slučaju čak malo nepregledno zbog opisa ispod svakog linka. Iako taj opis pomaže korisniku to se moglo riješiti na način da se prilikom prijelaza miša na link pojavi njegov tekst opisa kao informacija. Tako bi sučelje bilo preglednije, a korisnik bi imao opis za svaki link.



Slika 10: Početno sučelje ConfTool Pro sustava

Funkcionalnosti rade dobro i bez grešaka, imaju detaljan opis te se korisnik može lako snaći. Kao i u prijašnjim sustavima i ovdje nedostaje *ajax* tehnologija. Za sve funkcionalnosti treba puno klikati i navigirati, čak i za jednostavne stvari poput opisa dokumenta treba klikom ići na drugu stranicu da bi se pročitali detalji te ponovno vratilo nazad (slika 11). To može biti naporno za korisnika, pogotovo ako želi na brzinu proći više dokumenata pa se u nekom trenutku može i izgubiti. Osim tih nedostataka u kojima su se neke stvari mogle riješiti bolje i efikasnije za korisnika, sustav radi kako bi trebao te ima puno dodatnih funkcionalnosti (pro verzija) koje su dobro opisane. U globalu je sve dobro opisano i dokumentirano od same konfiguracije i funkcionalnosti do svih uloga koje pruža sustav. Uloga ima jako puno a administrator sustava može birati koje uloge želi imati tokom konferencije; pa tako osim osnovnih uloga (recenzent, autor...) postoje još i uloge editora, moderatora, prezentera,

asistenta itd. Možda čak ima i malo previše uloga i dokumentacije za nekog korisnika koji tek počinje koristiti sustav, no to može lako ignorirati, a kasnije ako je potrebno se vratiti. Budući da za besplatnu verziju ne postoji tehnička podrška, dodatna i detaljna dokumentacija dobro dođe - pa čak i za korisnike pro verzije jer mogu naći nešto konkretno bez kontaktiranja podrške.



Slika 11: Funkcionalnost recenziranja dokumenata

Nažalost besplatna verzija ne nudi hosting, pa se sustav mora instalirati na vlastitom serveru. To je nedostatak kao i kod prvog sustava, gdje korisnik može imati dodatne troškove ako nema vlastiti server ili pristup istom. Server također mora podržavati i zahtjeve sustava. Pro verzija se nalazi na njihovim serverima pa korisnik nema tih problema. Testiranjem demo verzija korisnik može odlučiti da li mu je potrebna pro verzija ili može upravljati konferencijom sa besplatnom verzijom. Ako ima mogućnosti instaliranja sustava na server to ne bi trebao biti problem jer sve osnovne funkcionalnosti rade kako bi trebale te se jednostavnom konferencijom može upravljati bez problema.

2.4. Zaključak analize

U skladu sa trenutnim trendom u računalnom softveru, dva sustava se ne nude samo kao tradicionalni prodajni softverski proizvod nego kao potpuna hosting usluga. Softver kao usluga (eng. *SaaS-Software as a Service*) smanjuje ukupne troškove tako da oslobađa organizatora konferencije da kupuje dodatni hardver i softver, unajmljuje administratora sustava, programera za instalaciju i podršku itd. Međutim javlja se drugi problem za ta cloud bazirana rješenja, a to je – sigurnost i privatnost analizirana od *Mark Ryan*[9]. Za organizatore koji preferiraju sustav na vlastitim serverima te žele preuzeti potpunu odgovornost za njihove podatke mogu koristiti *OpenConf* kao individualno rješenje.

Osim razlika u grafici i korisničkom sučelju sva rješenja nude osnovne kao i dodatne funkcionalnosti za upravljanje konferencijom poput procesa podnošenja dokumenata, dodjela recenzenata, revizija dokumenata itd. Također nude i funkcionalnosti poput provjere plagijata upotrebom vanjskih servisa poput *Docoloc-a*, automatsko generiranje programa konferencije i sl. Nedostatak im je naravno izgled sustava ali i ne korištenje nekih modernih tehnologija poput *ajax-a* što čini neke funkcionalnosti manje elegantnima i efikasnima.

Najveća razlika tih sustava bi bio način na koji se automatski dodjeljuju dokumenti recenzentima, odnosno u metodama za opis dokumenata koje kompetencije imaju revizori te algoritmima koji se koriste za to.

Generalno metode za opis se mogu podijeliti u dvije glavne skupine:

- *eksplicitne metode* - zahtijevaju da korisnik eksplicitno navede koje dokumente želi i/ili koje kompetencije ima (da pruža neke opisne metapodatke)
- *implicitne metode* - inteligentne metode koje automatski preuzimaju potrebne opisne podatke za dokumente i revizore preko prijašnjih publikacija sa Interneta

Implicitne metode većinom rade tekst analizu sadržaja dokumenta i/ili sadržaja online digitalnih knjižnica, bibliografskih indeksa ili ostalih resursa poput - *DBLP*, *Google Scholar*, *ACM* itd. Implicitne metode oslanjaju se na vanjske izvore podataka na internetu koji su više-manje inercijalni i sadrže rijetke podatke. Novi radovi i članci su indeksirani uz mjesec kašnjenja i ne od strane jednog te istog bibliografskog indeksa. Predlagatelji implicitnih

metoda priznaju da nisu bili u stanju pronaći relevantne informacije na internetu za oko 20% članova konferencija koje provode ove metode. Vjerojatno zbog tog razloga svi komercijalno dostupni sustavi za upravljanje konferencijom se oslanjaju na samo eksplicitne metode. Međutim, implicitne metode mogu biti vrlo korisne za automatsko otkrivanje sukoba interesa.

Postojeći sustavi koriste sljedeća tri načina za eksplicitni opis dokumenata i revizora:

- nadmetanje/ocjena dokumenta (recenzentovi iskazuju svoj interes za svaki dokument posebno)
- biranje sa liste predefiniranih tema konferencije
- oboje, spoj prvog i drugog

Nadmetanje zahtjeva da recenzenti prođu sve predane dokumente te daju svoje želje, odnosno da li žele ili ne žele recenzirati. To se obično radi na način da recenzent odabere opcije nadmetanja u padajućem izborniku. Njegove želje se smatraju apsolutno točnima, budući da nitko osim recenzenta ne zna da li je on dovoljno kompetentan da recenzira specifični dokument. Nadmetanje međutim ne opisuje dokumente ili recenzente već eksplicitno pokazuje na veze između njih. Ako dokument nije poželjan od dovoljno recenzenata, oni će mu biti nasumično dodijeljeni budući da sustav ne zna koja je domena dokumenta ili recenzentova kompetencija. U slučaju da je broj prijavljenih radova velik, recenzenti se nikad ne nadmeću za sve njih što rezultira nedostajanjem puno faktora sličnosti. *Philippe Rigaux* predlaže pametno rješenje zvano Iterativna Metoda Ocjene (eng. *IRM-Iterative Rating Method*) [10]. Ona pokušava prevladati slučajno dodjelu dokumenata na način da predviđa odabire koji nedostaju tako što iterativno nanosi kolaboracijski algoritam filtriranja na ocjene koje su eksplicitno dane od strane recenzenta ili prethodno predviđene (u prijašnjoj iteraciji).

Opisujući dokumente te kompetencije recenzenata sa listom tema područja, pretpostavlja se da sustav ima predefiniranu listu tema koje najbolje opisuju područje pokrivenosti konferencije. Tijekom predaje dokumenata autori moraju odabrati teme područja koje opisuju njihov rad. Recenzenti tokom registracije moraju odabrati teme koje odgovaraju područjima znanosti u kojima su kompetentni. Za razliku od nadmetanja, opis sa liste predefiniranih tema daje nezavisni, samostalan opis svakog dokumenta i recenzenta. Definiranje odgovarajuće i dovoljno dobre liste tema je međutim težak zadatak. Ne bi ih smjelo biti previše ali moraju pokriti cijelo područje znanosti u kojoj se nalazi konferencija te osigurati dovoljno pojedinosti za preciznim opisom radova i kompetencija. Za široko područje multidisciplinarnih konferencija to je nemoguć zadatak.

Kombinacija oba načina nudi prednosti za svakog od njih. Teme područja bi se trebale koristiti prvo za identificiranje male grupe dokumenata koja će biti predložena za nadmetanje

za svakog recenzenta. Ako recenzent stavi želju na dokument metoda će eksplicitno odrediti faktore sličnosti između određenog dokumenta i recenzenta. U protivnom će faktori sličnosti biti izračunati prema temama područja. Sva tri postojeća rješenja koriste metodu nadmetanja kao i odabir tema područja. Međutim, postoje značajne razlike u algoritmima dodjele. Ovdje ih se neće dotaknuti jer je to tema nekog drugog područja ali će se jedna vrsta algoritama koristiti u razvoju sustava.[8]

3. Razvoj vlastitog sustava

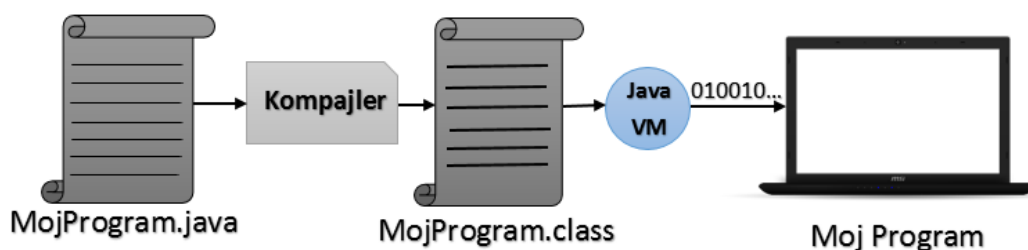
Glavni dio ovog rada bio je razvoj vlastitog sustava za upravljanje konferencijama. Sustav mora imati osnovne funkcionalnosti potrebne za upravljanje te mogućnost nadogradnje sa verzijama na neke dodatne. Cilj kod razvoja sustava je da se probaju ukloniti nedostaci i mane gore spomenutih sustava te da se napravi novi, vlastiti i bolji sustav. Za tu svrhu koristiti će se neke nove i moderne tehnologije. U ovom poglavlju opisati će se korištene tehnologije, te razvoj sustava pomoću dijagrama kao i kôd, funkcionalnosti te izgled. Također će se proći i spomenuti algoritam za automatsku dodjelu dokumenata recenzentima. Opisati će se zašto je on tako važan te na koji načine je implementiran odnosno kako funkcionira.

3.1. Korištene tehnologije i alati

Za razvoj ove web-aplikacije kao pozadinski (eng. *back-end*) jezik koristi se Java. Kao razvojni alat koristi se Netbeans IDE 8.0.2 (eng. *IDE-Integrated Development Environment*) te MySQL Workbench 6.3 za kreiranje i rad sa bazom podataka. Samabaza podataka je MySQL 5.6. Za kreiranje web-aplikacije u Netbeans-u korišten je Maven arhitektonski tip odnosno Maven okvir (eng. *framework*). Kao okvir za razvoj izgleda(eng. *front-end*) aplikacije koristi se JSF (eng. *Java Server Faces*) i PrimeFaces. Za pokretanje aplikacije koristi se GlassFish 4.1 Server. Ostale tehnologije koje se mogu naći u projektu su standardne web-tehnologije: HTML5, JavaScript, AJAX (eng. *Asynchronous Java Script and XML*), CSS (eng. *Cascading Style Sheet*). Sve tehnologije i alati su besplatni odnosno *open-source*, te zajedno odlično "harmoniraju" , pa se sa njima mogu napraviti odlične moderne web-aplikacije.

3.1.1. Java

Kada se govori o Java tehnologiji misli se na programski jezik i platformu. Zbog toga treba razlikovati Java programski jezik, Java Virtualni Stroj (eng. *JVM-Java VirtualMachine*) te Java platformu. Java programski jezik je jezik u kojem se pišu Java aplikacije, apleti, serveri i komponente. Kada se Java program (.java) kompajlira pretvara se u bajt kôd (.class) koji je prijenosni strojni jezik SPJ (eng. *CPU-Central processing unit*) arhitekture zvane Java Virtualni Stroj (slika 12). JVS se može implementirati direktno u hardver ali se najčešće implementira kao softverski program koji interpretira i izvršava bajtni kôd. Java programi se bez JVS za prikladni hardver i OS platformu ne mogu pokretati. Na sreću JVS je napravljen da radi na različitim okruženjima, od *Blue-ray playera* pa sve do velikih *mainframe* računala. Ta značajka omogućuje da se Java program izvršava na bilo kojoj platformi neovisno gdje je napisan. To znači da se Java program može razviti na Apple Mac računalu koristeći OS X te zatim prebaciti na Linux ili Windows te pokretati bez dodatnih promjena. Java platforma je predefinirani set Java klasa koje se nalaze na svakoj Java instalaciji. Java platforma se dakle sastoji od Java Virtualnog Stroja te Java API-eve (eng. *API-Application Programming Interface*). API pruža osnovnu funkcionalnost Java programskog jezika. Nudi široki spektar korisnih klasa spremnih za upotrebu u aplikacijama. [11]



Slika 12: Pregled procesa razvoja softvera

Java programski jezik je moderan objektno orijentirani jezik čija je sintaksa slična C-u. Dizajneri jezika težili su ka moćnom programskom jeziku ali u isto vrijeme pokušali su izbjeći prekompleksne značajke koje su zaustavile ostale OO jezike poput C++. Učinivši jezik jednostavnim, dizajneri su omogućili programerima da pišu robusne kodove bez grešaka. Kao posljedica elegantnog dizajna i značajki iduće generacije, Java jezik postao je vrlo popularan među programerima što ga trenutno čini brojem jedan programskim jezikom na svijetu.

Najnovije izdanje Java 8, koje se koristi i u razvoju ovog sustava, predstavlja dosad možda najveću promjenu u jeziku od Java 5 (2004.). Radi se u uvođenju lambda izraza (eng. *Lambda expressions*) koji obećavaju znatno povećati produktivnost programera. Razlog tome je što su kolekcije (eng. *Collections*) ažurirane da mogu koristiti lambde, a mašinerija potrebna za ostvarivanje toga pruža fundamentalnu promjenu objektno orijentiranom pristupu u Java-i. U svojih dvadeset godina postojanja (1995-2015), Java je konstantno napredovala, no njezin osnovni kostur i koncept ostao je isti. Tako bi se kôd napisan u Java 1 (1995) mogao pokrenuti na današnji Java 8 bez promjena i rekompajliranja. [12]

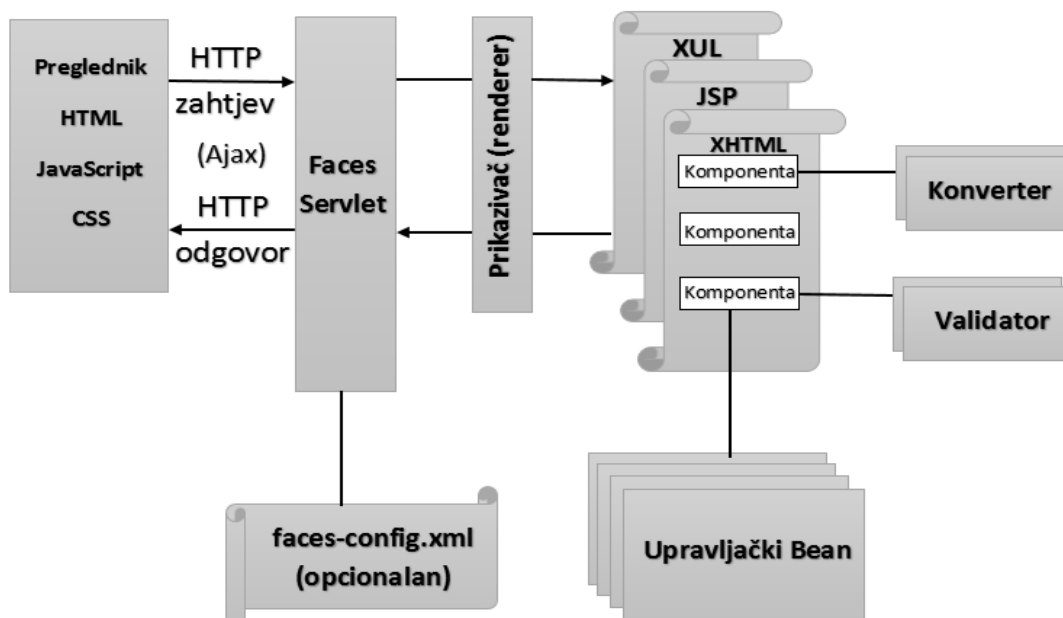
3.1.2. JSF, PrimeFaces i Mavenokviri(frameworks)

Iako ima još par odličnih i često korištenih okvira poput Spring, Hibernate i Struts koji su se mogli upotrijebiti na razvoju ovog projekta, ostalo se je na ovima u naslovu. Razlog tome je što se ne radi o velikom projektu sa većim brojem članova tima.

Softverski okvir je univerzalni, ponovno iskoristivo softversko okruženje koje pruža određenu funkcionalnost kao dio veće softverske platforme za olakšanje razvoja softverskih aplikacija, proizvoda i rješenja. Softver okviri mogu uključivati programe podrške, prevodioce, biblioteke, alatne setove i sučelja za programiranje aplikacija (eng. *API*) koje okupljaju sve različite komponente te omogućuje lakši razvoj projekta ili rješenja.[13]

JSF

JavaServerFaces (JSF) je java-temeljen web aplikacijski okvir namijenjen da pojednostaviti razvoj web-baziranih korisničkih sučelja. JSF arhitektura je jednostavna za razumjeti ako je korisnik upoznat sa web okvirima. Aplikacije sa JSF-om su standardne web aplikacije koje presijecaju HTTP (eng. *Hypertext Transfer Protocol*) preko Faces servleta te stvaraju HTML. Arhitektura dozvoljava upotrebu bilo kojeg PDL-a (eng. *page declaration language*) ili VDL-a (eng. *view declaration language*) koji se zatim renderira za različite uređaje (web preglednike, mobilne uređaje, itd.) te razvoj stranica upotrebom komponenti, slušača i događaja.



Slika 13: JSF arhitektura

Gornja slika reprezentira nekoliko važnih dijelova JSF-a koji ga čine bogatim i fleksibilnim:

- *FacesServlet i faces-config.xml*: FacesServlet je glavni servlet (manji Java program koji se pokreće unutar web servera) aplikacije, može se konfigurirati preko faces-config.xml deskriptora.
- *stranice i komponente*: JSF dozvoljava više vrsta PDL-ova, poput JSP-a, Facelets-a i XHTML-ova.
- *prikazivači (eng. Renderers)*: Zaslužni za prikazivanje komponenti i prevođenja korisničkog unosa u vrijednosti svojstva komponenti.
- *konverteri*: Oni konvertiraju vrijednost komponenti (Date, Boolean, itd.) u i od markup vrijednosti (String).
- *validatori*: Odgovorni su da unesene vrijednosti od strane korisnika budu ispravne.
- *Upravljački bean i navigacija*: Poslovna logika se nalazi u upravljačkim bean-ovima, koja također može upravljati navigacijom između stranica.
- *Ajax podrška*: JSF 2.0 dolazi sa ugrađenom podrškom za Ajax tehnologiju.

Većina web okvira koriste Model-Pogled-Kontroler (eng. *MVC-Model View Controller*) uzorak dizajna, tako i JSF. Uzorak se koristi da bi se odvojio pogled (stranice) i model (podaci koji se moraju prikazivati). Kontroler upravlja akcijama korisnika koje rezultiraju

promjenama u modelu te pogledu. U JSF-u taj kontroler je servlet zvani FacesServlet. Svi korisnički zahtjevi prolaze kroz FacesServlet koji provjerava zahtjev te poziva različite akcije na modelu koristeći upravljačke bean-ove. Taj servlet je interni te dio JSF-a. Jedini način na koji se može konfigurirati je koristeći vanjske metapodatke. Do JSF 1.2, jedini izvor za njegovu konfiguraciju bila je faces-config.xml datoteka. Od JSF 2.0 ta datoteka je izborna a većina metapodataka se konfigurira preko anotacija.

JSF okvir mora poslati stranicu za pregled na korisnikov uređaj (npr. preglednik) te joj je za to potrebna neka vrsta tehnologije za prikaz. Ta tehnologija je PDL (eng. *page declaration language*). JSF aplikacija može koristiti nekoliko tehnologija za svoj PDL, poput JSP-a i Facelets-a. Stranice napravljene od tih tehnologija sastoje se od stabla komponenti koje pružaju specifične funkcionalnosti za interakciju sa krajnjim korisnikom (tekst polja, gumbi, listboksevi itd.). Komponente su zatim povezane sa upravljačkim bean-ovima koji koriste anotacije te se na taj način korisnički zahtjevi prenose u model ali i obrnuto. JSF sve to rješava interno (JSF arhitektura) pa je na korisniku samo da razmišlja o logici i izgledu aplikacije. Na razvoju projekta koristila se JSF verzija 2.2.[12]

Maven

Na službenoj stranici Maven-a stoji da je Maven alat za uključivanje i upravljanje softverskim projektima. Baziran na konceptu projektno objektnog modela (eng. *POM-Project object model*), Maven može upravljati gradnjom projekta, izvješćivanjem i dokumentacijom iz središnjeg dijela informacija. Iako ovo daje ideju o tome što je Maven, ne objašnjava točno o čemu se radi. Dakle, Maven pruža bogatu razvojnu infrastrukturu (kompilacija, testiranje, izvještavanje, kolaboraciju i dokumentaciju) do jednostavnog opisa bilo kojeg Java projekta. To je jednostavan način za gradnju projekta bez da se gradi gradbeni sustav.

Maven je počeo kao pokušaj da se pojednostavni i standardizira kompleksni, Ant-baziran gradbeni proces korišten za Jakarta Turbine. Danas Maven hvata najbolju praksu i kodificira zajednički gradbeni proces na način da se može dijeliti na svim projektima. Kada se projekt „mavenizira“, kreira se projektni deskriptor koji opisuje sadržaj i formu projekta. Taj deskriptor se zatim koristi od strane zajedničkih plug-inova koji kompiliraju, testiraju, pakiraju i razvijaju projektne artefakte. Sa drugim riječima, Maven-u se pokaže na problem i on ga riješi. Maven-u se veli gdje se kôd nalazi, a on se pobrine za kompilaciju. Kaže se mu da napravi WAR datoteku ili JUnit izvještaj, a on pribavi potrebne biblioteke automatski.

Maven je dio automatizacijski, dio gradbeni i dio opisni. Ako se Maven koristi kako treba, manje vremena se provodi radeći na sistemu za gradnju projekta, a više vremena na samom kodu projekta. [15] Trenutno aktualna verzija (26.8.2015.) je 3.3.3., dok se za razvoj projekta koristila verzija 3.0.5.

PrimeFaces

PrimeFaces je vodeći lagani besplatni korisnički okvir za JSF-bazirane web aplikacije. Pruža bogati izbor sa preko sto UI (eng. *User Interface*) komponenti u samo jednoj JAR datoteci, bez potrebe konfiguracije i bez preduvjeta. Namjera PrimeFaces-a je izrada komponenti sa ugrađenom Ajax tehnologijom koje su bazirane na standardnim JSF 2.0 Ajax API-evima te bogatim izgledom sa mehanizmom za mijenjanjem tema. Komponente u PrimeFaces-u su razvijene sa izjavom „Dobra UI komponenta mora sakriti svoju kompleksnost ali zadržati fleksibilnost“. Na taj način razvijatelj dobiva moćne ali jednostavne komponente za izradu modernog i lijepog sučelja za krajnjeg korisnika. Trenutna besplatna verzija PrimeFaces-a je 5.2, koja se ujedno koristi i u razvoju projekta. Uskoro bi trebala doći verzija 5.3 koja sa sobom donosi još više mogućnosti, poput responzivnih i dodatnih komponenti.[17]

3.1.3. MySQL

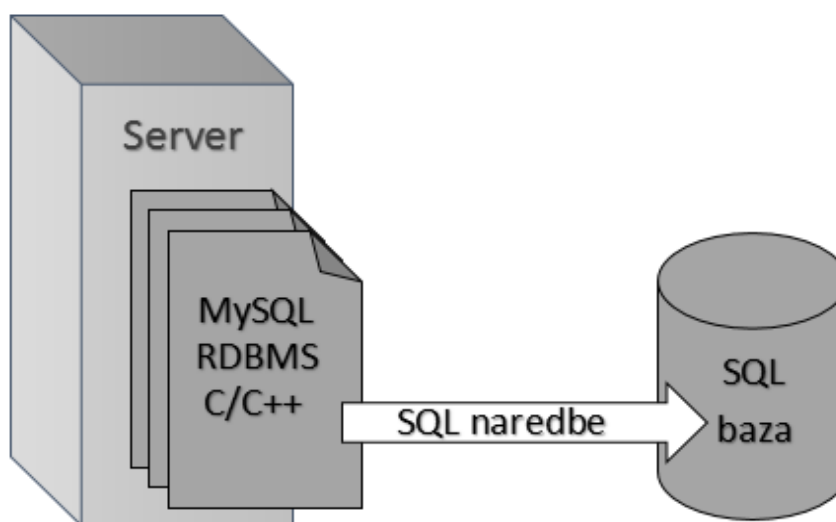
MySQL je najpopularniji besplatni (eng. *open-source*) sistem za upravljanje relacijskim bazama podataka (eng. *RDBMS-relational database management system*). Kao i sve ostale komercijalne verzije RDBMS-a, MySQL pruža bogati set karakteristika koje osiguravaju sigurno okruženje za spremanje, održavanje i pristupanje podacima. Dakle MySQL je brza, pouzdana i skalabilna alternativa mnogim komercijalnim RDBMS-ima. Sljedeća lista pruža pregled svih važnih karakteristika MySQL-a:

- *skalabilnost*: MySQL može upravljati velikim bazama, što je dokazano implementacijom u organizacijama poput Google, Yahoo!, NASA itd. Prema MySQL dokumentaciji, neke baze koje se koriste imaju preko 50 milijuna zapisa i preko 60,000 tabela sa 5 bilijuna redaka.

- *prenosivost*: MySQL se može pokretati na više vrsta operacijskih sustava, poput Unix, Linux, Windows, MacOS itd. MySQL se također može pokretati na različitim arhitekturama, uključujući mikro računala pa sve do mainframe računala.
- *konektivnost*: MySQL je potpuno umreživ te podržava TCP/IP sokete, Unix sokete i imenovane cjevovode. Također, MySQL-u je moguće pristupiti sa Interneta od strane više korisnika istodobno. MySQL pruža i nekoliko API-ova koji daju mogućnost spajanja na aplikacije napisane u C, C++, Java, PHP itd.
- *sigurnost*: MySQL uključuje moćni sistem za upravljanje kontrolom pristupa podacima. Sistem koristi host i korisnik bazirane strukture koje kontroliraju tko može pristupiti određenim informacijama kao i nivo pristupa istima. Za sigurnosne konekcije podržan je i SSL(eng. *Secure Sockets Layer*) protokol.
- *brzina*: MySQL je razvijen imajući brzinu na umu. Vrijeme potrebno bazi u MySQL-u da se javi zahtjevu za podatke jednaka je ili brža od mnogih komercijalnih RDBMS-ova. Službena stranica sadrži rezultate testiranja koji demonstriraju brzinu koja se dobije MySQL implementacijom.
- *jednostavnost*: MySQL je jednostavan za instalirati i implementirati. Korisnik može imati instalirani i pokrenuti MySQL samo minutama nakon skidanja potrebnih datoteka. Čak i na administratorskoj razini je relativno lako za optimizirati.
- *besplatan*: MySQL AB čini MySQL izvorni kod dostupan svima za skidanje i upotrebu.

Kao skoro svi RDBMS-ovi tako i MySQL koristi SQL za upravljanje relacijskom bazom. SQL je iznad svega, računalni jezik koji se koristi za upravljanje i interakciju sa podacima u relacijskim bazama podataka. No ne smije se zamijeniti SQL sa programskim jezicima korištenim za razvoj RDBMS-ova. Tako je MySQL razvijen sa C i C++ jezikom. Dakle primarna funkcija SQL-a je da dopušta RDBMS interakciju sa podacima. Dok programski jezici sa kojima su napravljeni RDBMS pružaju okruženje u kojima se koristi SQL za upravljanje sa podacima.

Slika ispod demonstrira kako SQL ima interakciju sa MySQL RDBMS. MySQL se pokreće na nekom serveru sa Linux ili Windows platformom, dok baza pohranjena interno ili eksterno ovisno o konfiguraciji, sadrži stvarne datoteke baze.[18]



Slika 14: Interakcija SQL-a sa MySQL RDBMS

Kao što se može vidjeti, MySQL pruža brzu, pouzdanu soluciju za potrebe baza podataka. Za razvoj sustava korištena je MySQL verzija 5.6. sa standardnim SQL naredbama.

3.1.4. Ostale tehnologije

Ostale tehnologije korištene na projektu su standardne tehnologije za web, a to su (X)HTML, JavaScript i CSS. Iako se one već koriste sa okvirima poput PrimeFaces-a, ukratko će se svaka opisati jer su se koristile i pojedinačno izvan njega.

HTML (eng. *Hypertext Markup Language*) je jezik korišten za kreiranje dokumenata web stranica. Postoje nekoliko verzija HTML-a, no danas standardna je HTML5. Mnogo stroža implementacija te verzije je XHTML (eng. *eXtensible HTML*), koja je ustvari isti jezik ali sa mnogo strožim pravilima sintakse. Za razvoj projekta koristi se XHTML. HTML nije programski jezik, već opisni (eng. *Markup*) jezik što znači da je to sistem za prepoznavanje i opisivanje raznovrsnih komponenti dokumenta, poput zaglavlja, paragrafa itd. Opis pokazuje na temeljnu strukturu dokumenta (to se može zamisliti kao detaljna skica koju mašina može čitati).

Dok se (X)HTML koristi za opisivanje sadržaja web stranice, CSS (eng. *Cascading Style Sheets*) opisuje kako taj sadržaj mora izgledati. U web poslovnom okruženju način

na koji stranice izgledaju zove se prezentacija. To znači da font, boja, pozadinske slike, razmak između linija, raspored stranice itd. su kontrolirani sa CSS-om. Sa današnjom verzijom CSS3 se također mogu dodavati specijalni efekti i osnovne animacije na stranicu. CSS također pruža metode sa kojima se može kontrolirati kako će se dokumenti prezentirati na uređajima sa manjim ekranima. Također su dobar alat za automatsku produkciju jer se može specificirati kako će element izgledati na različitim stranicama sa samo jednim CSS dokumentom. [16]

JavaScript je skriptni jezik koji se koristi za dodavanje interaktivnosti i ponašanja web stranicama, kao što su:

- provjera ispravnosti unosa
- promjena stila elementa ili stranica
- mogućnost preglednika da zapamti informacije o korisniku za sljedeći put kada posjeti stranicu
- izgradnja widget-a za sučelje, poput proširivog izbornika
- akcije nakon klikanja na gumb ili neke druge elemente
- itd.

Dakle JavaScript se može koristiti za manipulaciju elemenata na stranici, stilova ili samog preglednika. Iako ima još mnogo mogućnosti i koristi od JavaScripta, pogotovo sa upotrebom Ajax-a, JSON-a ili nekih radnih okvira poput jQuery-a, ovdje se neće ići u te detalje.

3.2. Dijagrami sustava

Pod pojmom dijagrami sustava misli se na UML dijagrame te ERA model koji služe za lakšu i jednostavniju implementaciju rješenja, na način da vizualiziraju i prikazuju rješenja. Za razvoj ovog rješenja koristio se samo *Use Case* tip UML dijagrama da prikaže ponašanje sustava za svaku vrstu korisnika, odnosno šta koji tip korisnika vidi te može napraviti. Također se koristio i ERA model da bi se razvila potrebna baza podataka (tabele, relacije) prema izrađenom modelu.

UML (eng. *Unified Modeling Language*) je grafička notacija za crtanje dijagrama softverskih koncepata. Može se koristiti za crtanje dijagrama za problemsku domenu, ciljani softverski dizajn ili pak već implementiranog softverskog rješenja. Fowler (*UML Distilled*, 2d. ed. Martin Fowler, AddisonWesley, 1999) te tri razine opisuje kao *Konceptualna*, *Specifikacijska* i *Implementacijska*. *Specifikacijski* i *Implementacijski* razinski dijagrami imaju

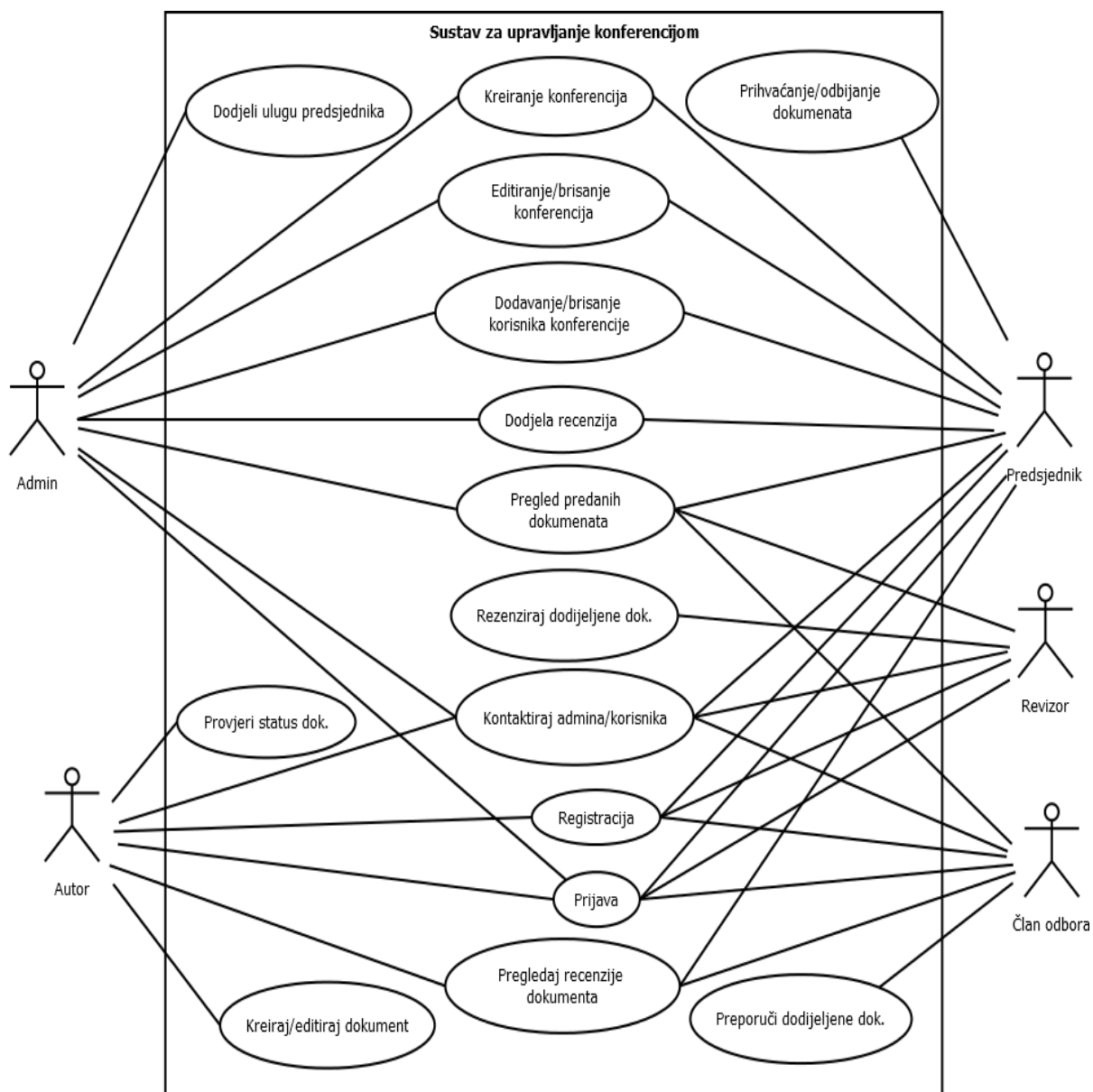
snažnu vezu prema izvornom kodu. Namjera dijagrama na *Specifikacijskoj* razini je da se i pretvore u izvorni kôd. Dok su dijagrami na *Implementacijskoj* razini namijenjeni da opisuju postojeći izvorni kôd. Kao takvi imaju pravila i semantiku koju moraju pratiti na toj razini. Takvi dijagrami imaju jako malo dvoznačnosti a mnogo formalnosti. Sa druge strane dijagrami na *Konceptualnoj* razini nisu nužno vezani za izvorni kôd. Oni se koriste da bi opisali koncepte i apstrakcije koje se nalaze u čovjekovoj problemskoj domeni. Ne prate stroga semantička pravila te zato njihovo značaj može biti dvosmislen.

Postoje nekoliko glavnih primarnih tipova dijagrama korištenih u UML-u. Oni su svrstani u tri kategorije. Statički dijagrami koji opisuju nepromjenjivu logičku strukturu softverski elemenata, prikazujući klase, objekte, strukture podataka i veze koje postoje između njih. Dinamički dijagrami prikazuju kako se softverski entiteti mijenjaju tijekom izvršavanja, prikazujući tok izvršavanja ili način kako entiteti mijenjaju stanje. Fizički dijagrami pa prikazuju nepromjenjivu psihičku strukturu softverskih entiteta, prikazujući psihičke entitete poput izvornih datoteka, biblioteka, binarnih datoteka, podatkovnih datoteka te vezu između njih. Ti glavni tipovi dijagrama su dijagram klasa, dijagram objekata, dijagram sekvence, kolaboracijski dijagram i dijagram stanja.

Crtanje dijagrama može biti veoma korisna aktivnost, no može biti i strašno gubljenje vremena. Odluka o korištenju UML-a može biti dobra stvar ali i loša. Ovisi o tome kako i koliko često se misli koristiti. Pravilo da sve mora biti napravljeno sa dijagramima je gore nego pogrešno. Mnogo projektnog vremena se može potrošiti na dijagrame koji neće biti potrebni i nitko ih neće čitati. Ispod se nalazi lista koja sadrži opis kada je najbolje crtati dijagrame:

- kada više ljudi treba razumjeti strukturu specifičnog dijela dizajna jer će svi raditi na istom istodobno. Treba prestati kada svi vele da razumiju.
- kada dvoje ili više ljudi imaju suprotna mišljenja kako bi specifični element trebao biti dizajniran te žele mišljenje tima.
- kada se želi igrati sad dizajnerskom idejom, a dijagrami pomažu u tome. Treba prestati kada se dobije rješenje te se to može preusmjeriti u kôd.
- kada se treba objasniti struktura nekog djela ili koda nekome drugome ili samome sebi. Treba se prestati kada bi objašnjenje bilo bolje gledanjem kôda.
- kada je projekt pri kraju, a kupac je zatražio dijagrame kao dio dokumentacije za ostale.

Sa gore navedenim pravilima za projekt se odlučilo koristiti dijagram koji će samo pojednostavniti i objasniti kako funkcioniра interakcija pojedinih tipova korisnika sa sustavom, odnosno što pojedini tip korisnika može i mora moći raditi. Da se to lakše protumači upotrijebili su se scenariji korištenja (eng. *Use Cases*). Scenariji korištenja opisuju ponašanje sustava. Opis se piše sa gledišta korisnika koji je rekao sustavu da napravi nešto određeno. Scenarij korištenja zatim hvata vidljivu sekvencu događaja koje sustav prolazi za pojedinog korisnika. Pod vidljivim događajem misli se na događaj kojeg korisnik vidi. Scenarij korištenja ne opisuje skrivena ponašanja i mehanizme sustava. On samo opisuje one stvari koje korisnik može vidjeti i koristiti. Određeni scenarij se umjesto pisanja može i na neki način pretvoriti u dijagram korištenja. Tu dolazi takozvani Dijagram granica sustava (eng. *System Boundary Diagram*) pomoću kojeg se mogu predočiti naši tipovi korisnika te njihove scenarije korištenja u sustavu.[19] Za ostale spomenute dijagrame nije bilo potrebe, budući da se ne radi o projektu koji se razvija u timu, a samo zbog dokumentacije bi bio gubitak vremena. Nakon napravljenih dijagrama korištenja bilo je lakše napraviti ERA model ali i sama logika kôda je nakon njih bila zamišljena, pa opet nije bilo potrebe za drugim dijagramima.

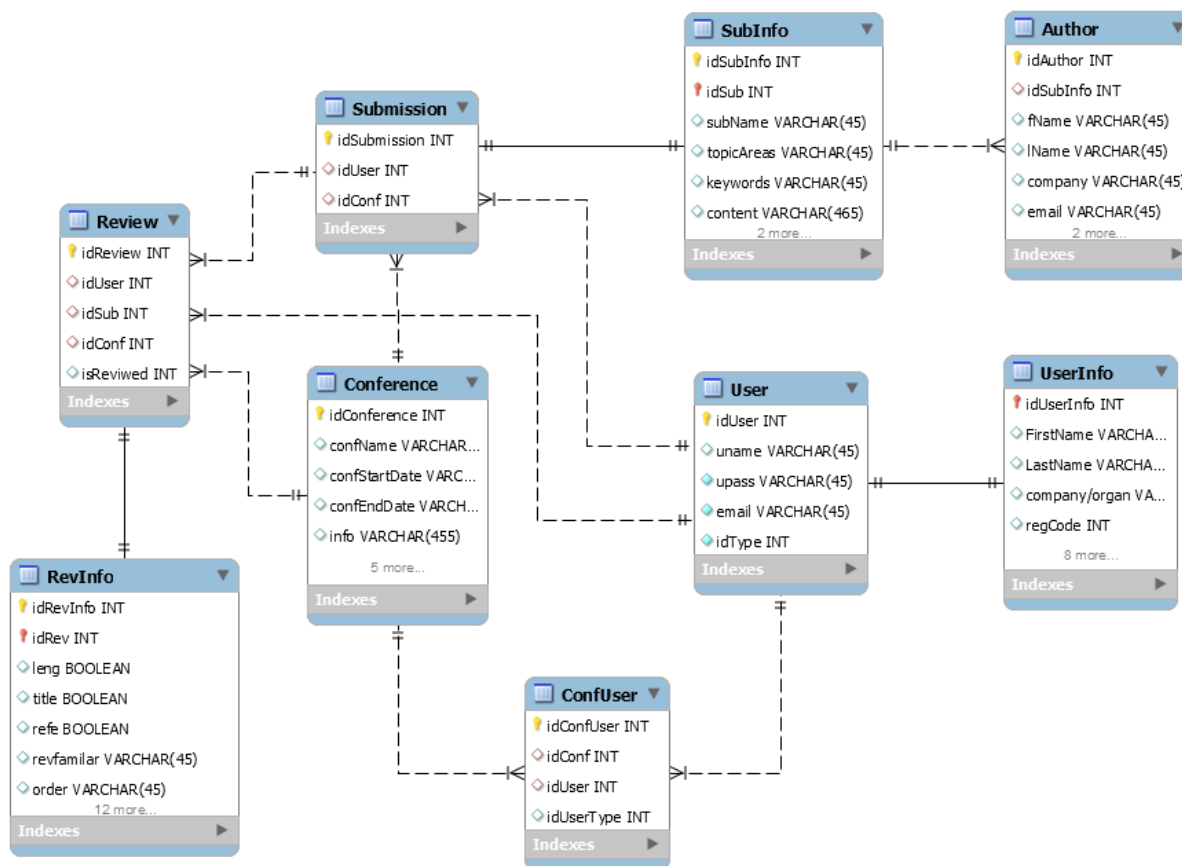


Slika 15: Slučajevi korištenja sustava za upravljanje konf. u System BoundaryDiagram-u

Veliki pravokutnik su granice sustava. Sve unutra pravokutnika je dio sustava koji se razvija. Izvan pravokutnika su korisnici koji djeluju na sustav. Unutar pravokutnika se nalaze slučajevi korištenja. Linije zatim povezuju korisnike sa slučajem kojeg oni mogu koristiti.

U slučaju našeg sustava za upravljanje konferencijom, imamo pet tipova korisnika koji mogu upravljati sa 15 slučajeva, ovisno o kojem tipu korisnika se radi. Svi korisnici osim admina moraju se prvo registrirati. Ako korisnik želi biti predsjednik; da bi mogao kreirati konferencije, dodavati korisnike u njih, dodijeliti recenzije, pregledavati te prihvaćati/odbijati dokumente mora kontaktirati admina koji mu daje ulogu predsjednika. Ostale uloge dobiju se

od strane predsjednika ovisno o konferenciji kojoj se pristupa, a mogu biti različite. Tako autor može predati svoj dokument, provjeriti da li je prihvaćen/odbijen i pregledati recenzije svojeg dokumenta. Revizor može pregledavati predane dokumente korisnika koji su mu dodijeljeni od strane predsjednika te recenzirati iste. Član odbora osim pregledavanja, može preporučiti, ocijeniti te vidjeti recenzije dodijeljenih dokumenata. Te funkcionalnosti daju predsjedniku mogućnost upravljanja konferencijom dok neke ostale koje se mogu naknadno dodati pojednostavnjuju to upravljanje.



Slika 16: ERA model sustava

ERA (eng. *Entity-Relationship-Attribute*) model sa slike napravljen je sustav prema slučajevima korištenja korisnika. Prikazuje skup stavki podataka i odnosa između njih u smislu entiteta (tablica), veza i atributa koji su uključeni. Entiteti sadrže attribute te su povezani sa drugim entitetima preko veza. Atributi su polja koja imaju svoj naziv, tip i druge karakteristike te sadrže vrijednosti. Gornji model sastoji se od devet entiteta od kojih svaki ima svoj broj atributa te različite vrste veza ovisno o entitetu. Entitet *User* sadrži korisničke podatke za prijavu te je povezan sa entitetom *UserInfo*, koji sadrži dodatne podatke o korisniku, vezom jedan-jedan što znači da jedan korisnik može imati jedne dodatke podatke.

Također je povezan sa entitetom *Conference*, koji sadrži podatke o konferencijama, vezom više-više sa pomoćnim entitetom *ConfUser*. Veza više-više znači da korisnik može biti u više različitih konferencija a konferencija može imati više različitih korisnika. Povezan je još sa entitetima *Review* i *Submission* vezom jedan-više, što znači da jedan korisnik može imati više revizija i dokumenata. Od ostalih entiteta tu su još *RevInfo*, *SubInfo* i *Author* koji sadrže dodatne podatke za druge entitete sa kojima su spojeni. Na ovaj način spremaju se svi podaci koji se koriste od strane sustava.

3.3. Algoritam za automatsku dodjelu dokumenata recenzentima

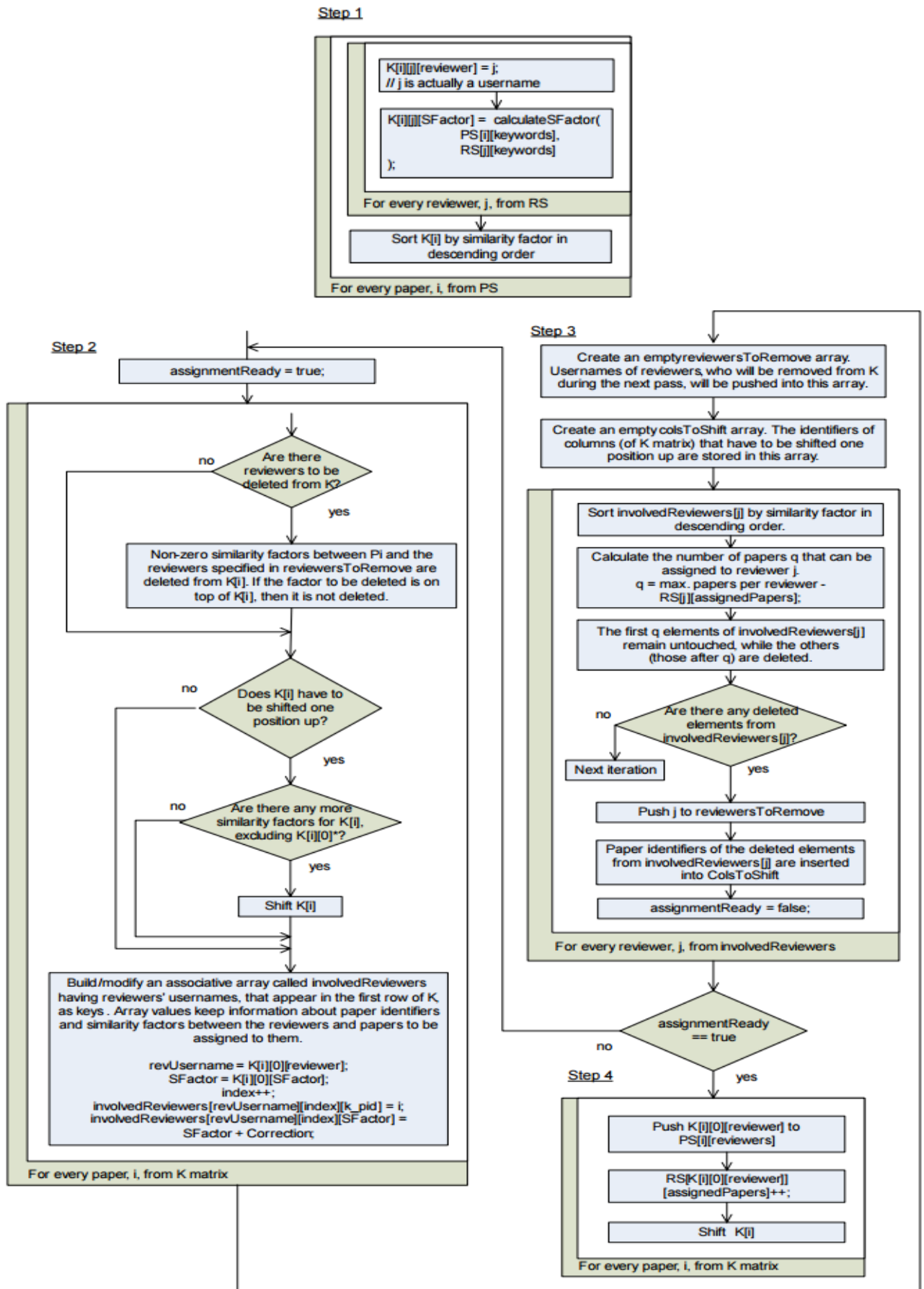
Proces dodjele dokumenata recenzentima je vjerojatno najvažniji i najizazovniji zadatak. Njegova točnost izravno utječe na kvalitetu konferencije i njezine slike. Za konferencije visoke razine, koje imaju nisku stopu prihvatanja, presudno je da dokumente evaluiraju najkompetentniji recenzenti za to područje. Dodjela može biti ručna i automatska. Ručna dodjela moguća kod konferencija koje imaju malo predanih dokumenata. No sa povećanjem broja dokumenata i recenzenata, ručna dodjela postaje sve manje i manje točna zbog ograničenja koje mora zadovoljiti - visoka točnost, sukob interesa i balansiranje opterećenja (tj. da svi recenzenti imaju otprilike isti broj radova).[8] Zbog toga je funkcionalnost automatske dodjele važan dio svakog sustava za upravljanje konferencijom.

U poglavlju 2.4 opisale su se dvije glavne skupine metoda za opis. Ovaj algoritam koristi eksplicitni način odnosno algoritam se oslanja na ključne riječi odabrane od strane autora i recenzenata. Koristeći padajući izbornik recenzenti i autori mogu birati između **konačnog** broja ključnih riječi za danu konferenciju. Na primjer neka je konferencija kategorije informatika. Moguće ključne riječi za odabir su baze podataka, mreže, sigurnost informacijskih sustava itd. Korisnik mora odabrati n ključnih riječi. Autor zatim odabire 3-5 ključnih riječi vezanih uz temu dokumenta dok recenzenti većinom biraju i više. Zatim je pronalazak najprikladnijeg dokumenta za recenzenta lak zadatak. Zapravo automatska dodjela je zatim vrlo brza i jednostavna ako nema ograničenja po broju dokumenata za recenzenta. Ako ima ograničenja algoritam automatski izračunava, a svaki recenzent zatim dobije jednak broj dokumenata. Za pronalazak najprikladnijeg recenzenta za svaki dokument izračunava se faktor sličnosti.

Primjer:

Pretpostavimo da konferencija ima sljedeće ključne riječi $KU=\{A,B,C,D,E,F,G,H,I\}$, autor je odabrao: $KR_D = \{A,C,F\}$ dok je recenzent jedan odabrao: $KR_{R1} = \{B,C,F,G\}$, a recenzent dva: $KR_{R2} = \{A,F\}$. Recenzent jedan i recenzent dva su jednako sposobni da recenziraju dokument budući da oba pokrivaju dvije od tri ključne riječi koje opisuju dokument. Tu dolazi do upotrebe izračun faktora sličnosti. Za prvog revizora on je: $FS_{DR1} = 2/5$ dok je za drugog on: $FS_{DR2} = 2/3$. Dakle recenzent dva je prikladniji za recenziranje tog dokumenta od recenzenta jedan. Razlog tome je što recenzent jedan ima više ključnih riječi što znači da je sposoban recenzirati više dokumenata. Drugim riječima, vjerojatnost pronalaska drugog dokumenta kojeg će recenzent jedan mogući evaluirati je veća nego vjerojatnost za recenzenta dva. Budući da postoji ograničen broj dokumenata po recenzenti dodjela dokumenta recenzentu jedan bila bi gubljenje resursa.

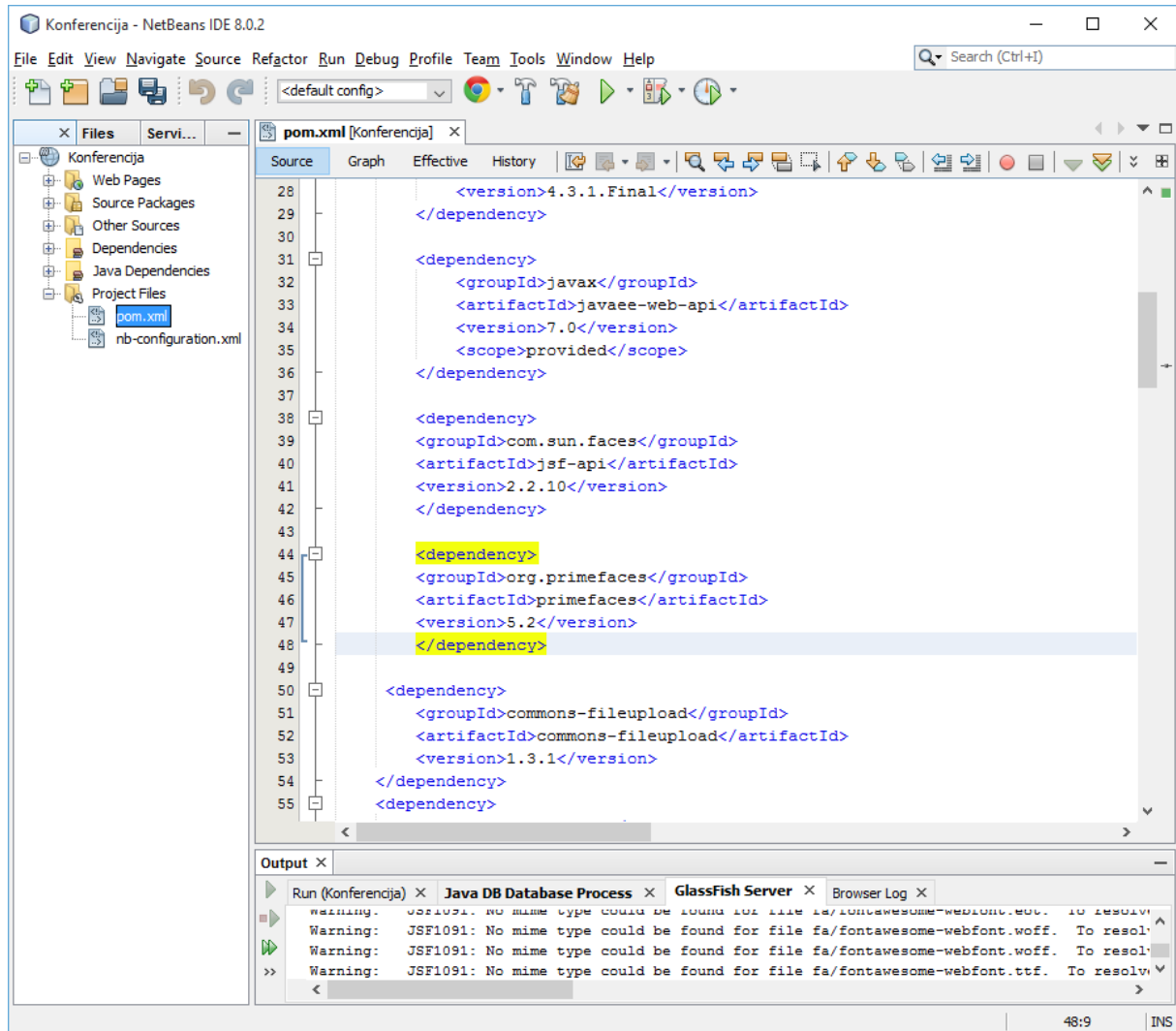
Algoritam se može podijeliti u četiri glavna koraka. Na početku treba znati spomenute faktore sličnosti. Zbog toga u koraku jedan počinje sa izračunom matrice sličnosti K . U drugom koraku algoritam uzima prvi red sortirane matrice K . U prvom redu matrice nalaze se najprikladniji recenzenti za svaki dokument. Zatim se izračunava koje recenzente treba izbaciti ako ih ima previše za isti dokument, a broj dokumenata po recenzentu je manji te koji redovi matrice se moraju pomaknuti. Pomoću korekcija napravljenih u koraku dva, u trećem se koraku određuje koji dokumenti se moraju dodijeliti recenzentima koji su se nalazili u prvom redu matrice. Ti koraci se ponavljaju dok se ne dobi željena završna matrica. Ona sadrži finalne recenzente koji će se u koraku četiri dodijeliti dokumentima. Na kraju koraka četiri svi dokumenti imaju jednog ili nula dodijeljena recenzenta. Ako dokumente treba recenzirati od više recenzenata koraci dva do četiri se ponavljaju koliko treba. Na slici 17. prikazani su koraci jedan do četiri. Svaki korak prikazuje što algoritam u pojedinom dijelu točno treba raditi da bi se dobio konačan rezultat automatske dodjele. [20]



Slika 17: Koraci 1-4, algoritma za automatsku dodjelu dokumenata recenzentima(izvor:[20])

3.4. Razvoj i kôdsustava

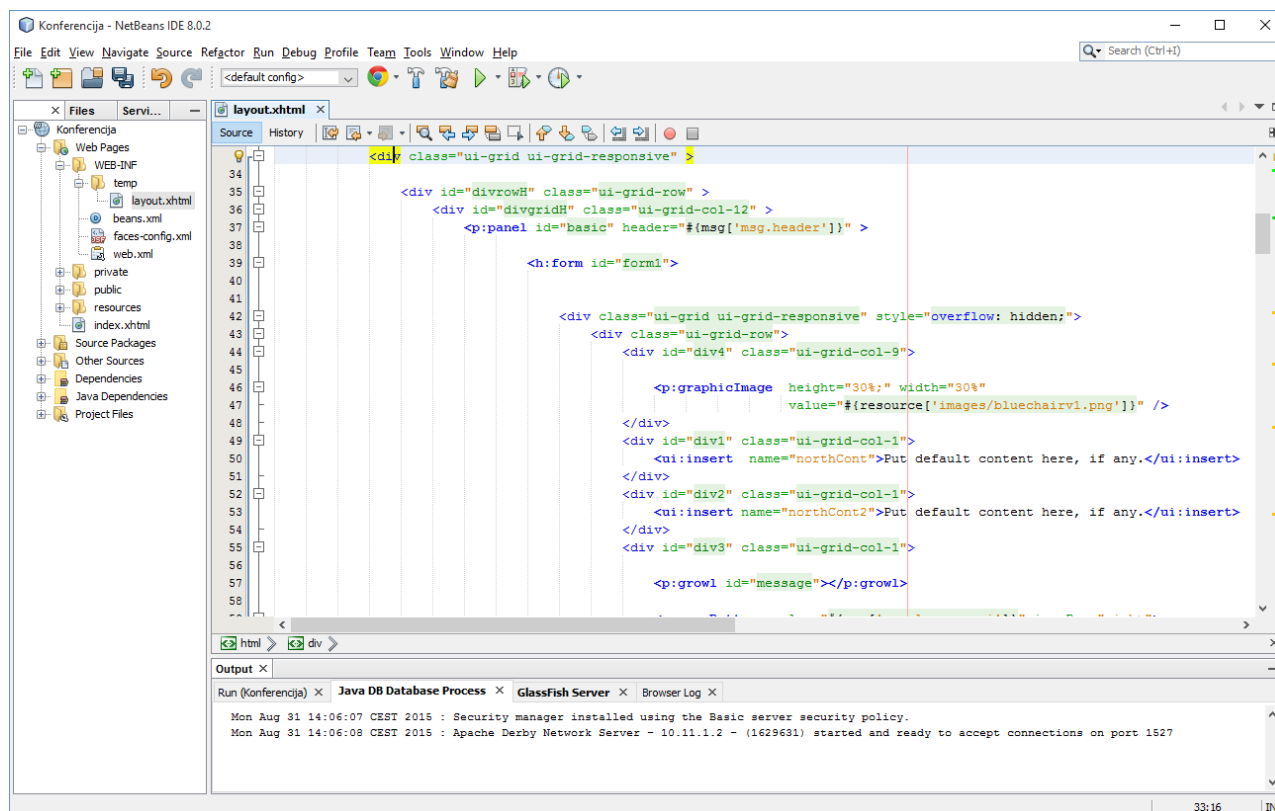
Nakon što se kreirao projekt u NetBeans IDE-u pomoću Maven-a, moglo se početi sa programskim dijelom. Za početak su se trebale dodati ovisnosti koje će se koristiti za razvoj sustava, poput PrimeFaces-a, MySQL-a itd. Sa Maven-om je to lako jer treba samo u *pom.xml* datoteci dodati ovisnosti koje su potrebne, a Maven zatim preuzme i spremi potrebne biblioteke.



Slika 18: Dodavanje ovisnosti pomoću Maven-a

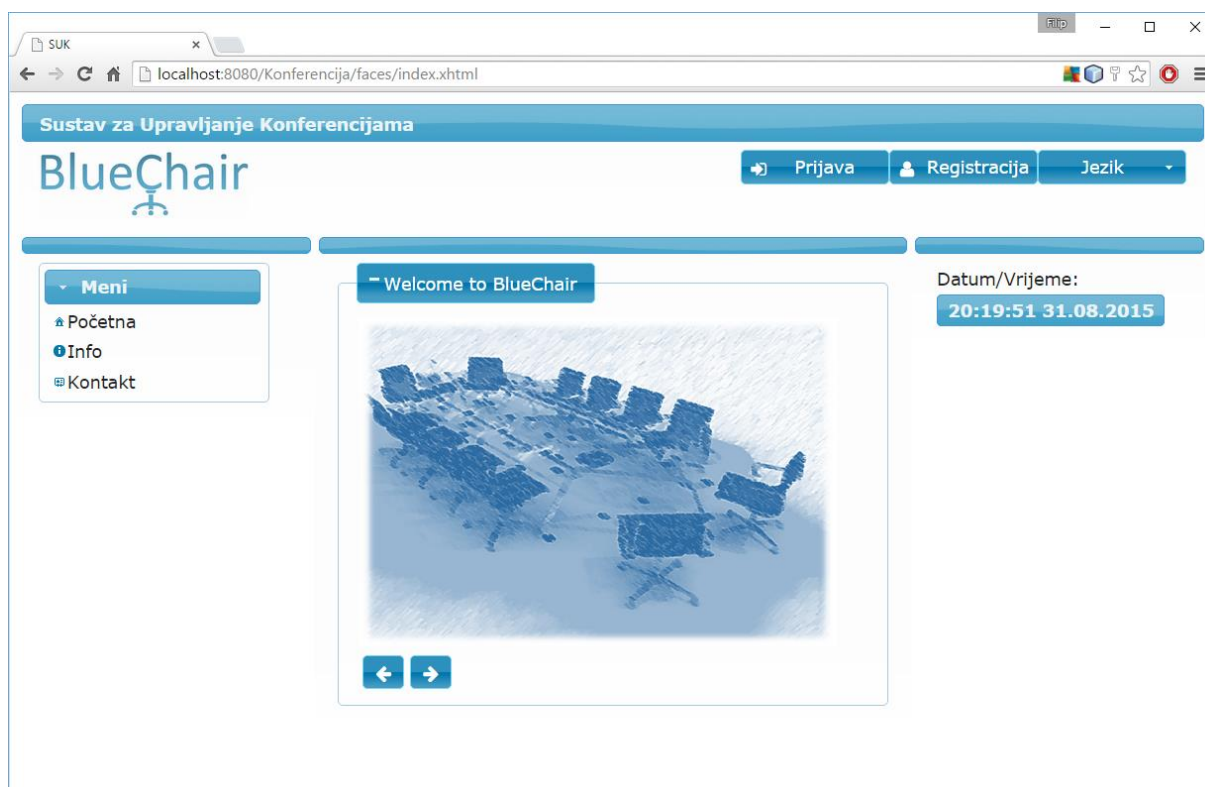
Da bi izgled i funkcionalnosti bilo lakše i bolje implementirati, napravila se jedna *layout.xhtml* datoteka koja služi kao predložak. Ona se dalje koristi za sve ostale stranice, što znači da svaka druga stranica samo „ubačena“ u nju. PrimeFaces komponente koriste oznaku za biblioteku *p*, standardne JSF oznake biblioteka su *f*, *h*, *c*, *ui*, *fn* i *ez*. U *layout.xhtml* datoteci koristilo se nekoliko glavnih komponenti koje čine predložak. Jedna od tih je *ui:insert* pomoću koje se može sa bilo koje druge stranice ubaciti sadržaj u nju. Za grupiranje

elementa koristila se *p:panel* komponenta, dok se sve to sa ostalim komponentama poput gumbića, slika i teksta nalazi u PrimeFaces-ovoj responzivnoj mreži. Ona se sastoji od običnih *div* elemenata koji imaju nazive klasa takve da čine responzivnu mrežu pomoću PrimeFaces CSS-a. Sve unutar mreže je responzivno, dakle mijenja veličinu i poziciju ovisno o veličini ekrana na kojim se prikazuje.



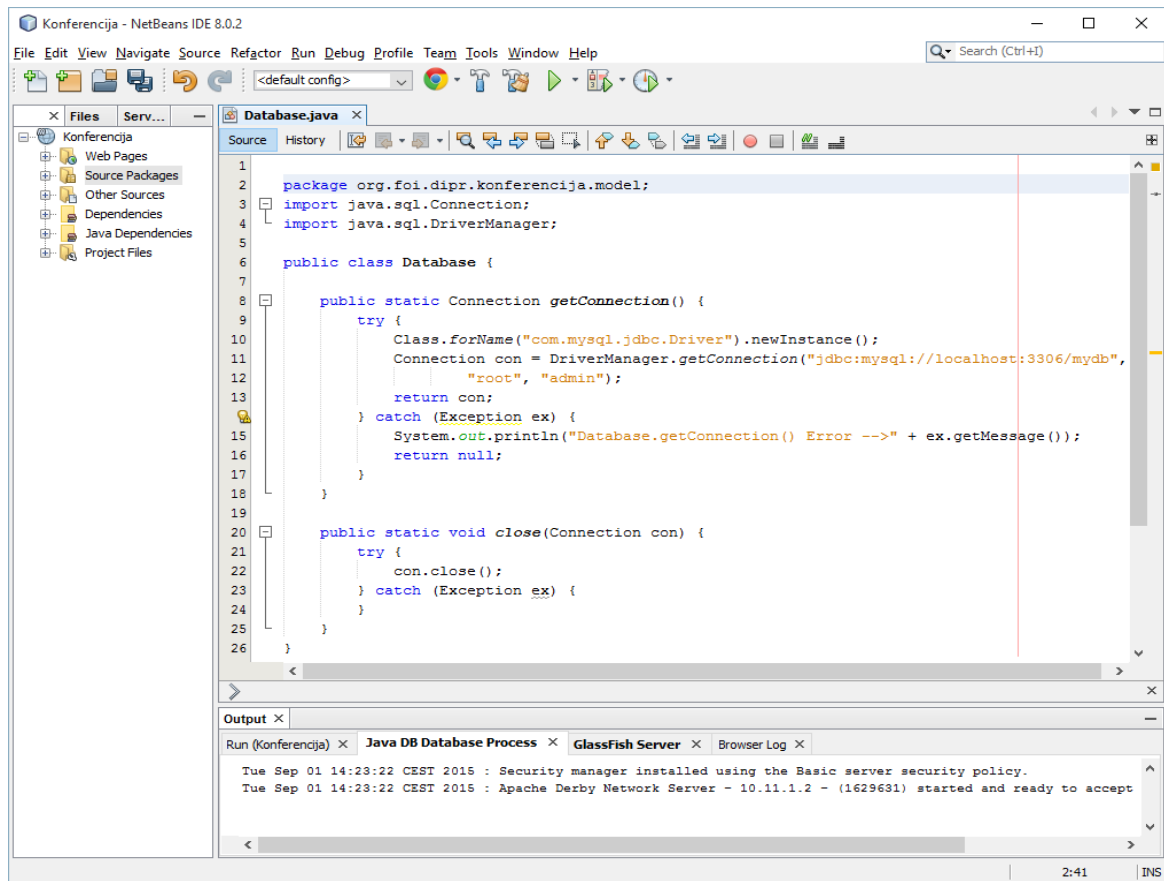
Slika 19: `layout.xhtml` datoteka, sa spomenutim komponentama

Dodatno tome, sve komponente i elementi se nalaze unutar *f:view* kontejnera, on sadrži atribut *locale* te pomoću njega i pozadinske funkcije se mijenja jezik sustava. Komponente koje se još koriste u predlošku, a samim time i na ostalim stranicama su *p:ajaxStatus*, *p:clock* i *p:menu*. Komponenta *p:ajaxStatus* je globalni indikator koji pruža povratnu informaciju kada je u tijeku Ajax zahtjev, koji se u sustavu koristi za prikaz grafičke animacije kada se to događa. Glavni meni je napravljen sa *p:menu* komponentom, koja sadrži elemente na koje se može kliknuti te izvršiti neku akciju ili preusmjerene. Sa *p:clock* se prikazuje vrijeme i datum servera. Na slici 19 se vidi početna stranica, odnosno `index.xhtml` datoteka. Ona sadrži *ui:composition* komponentu sa kojom može koristiti `layout.xhtml` predložak. Za umetanje svojeg sadržaja u *ui:insert* komponente predloška koristi *ui:define* komponentu sa određenim nazivom, ovisno o kojem *ui:insert*-u se radi. Za ljepši izgled stranice sadrži u središnjem panelu *p:imageSwitch* koji automatski mijenja dodijeljene slike.

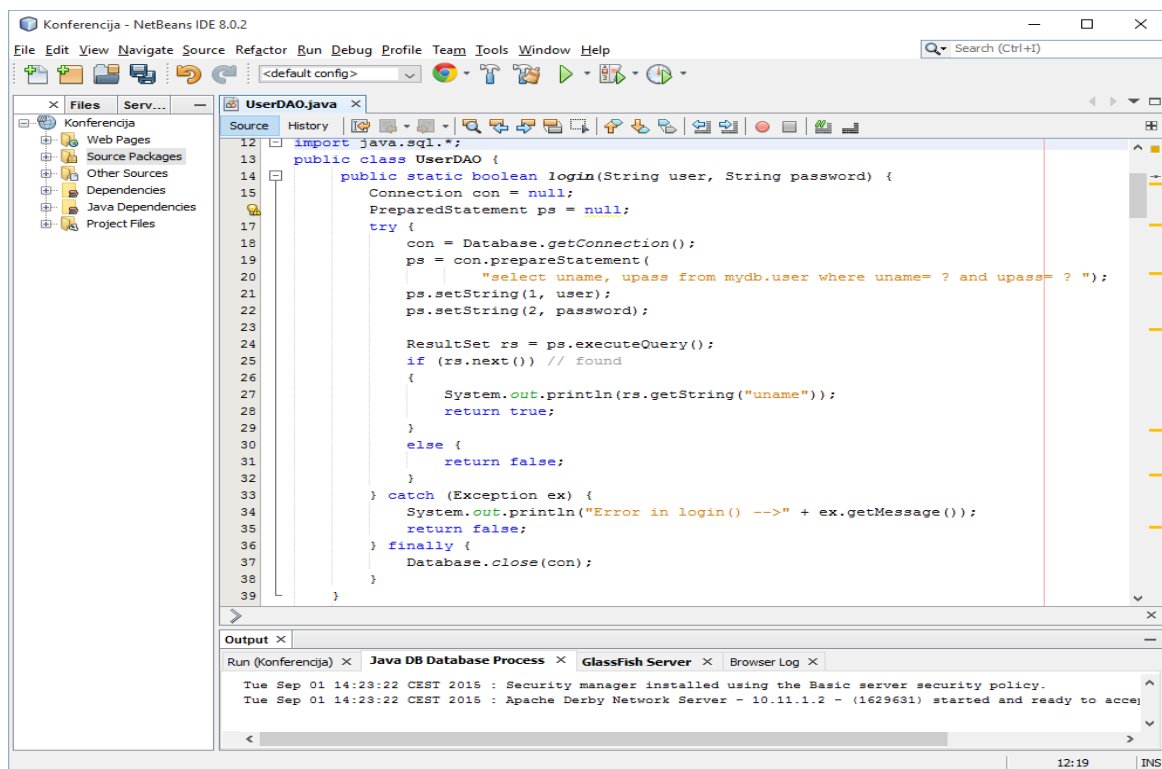


Slika 20: Izgled početne stranice sustava

Ostale stranice napravljene su na isti način, samo sadrže još svoje komponente i sadržaj. Prva takva bila je stranica za registraciju. Ona sadrži *PrimeFaces Extension* biblioteku koja ima komponentu *pe:masterDetail*. Sa njom se korisnik registrira korak po korak, odnosno može ići na sljedeći nivo tek kada ispuni potrebne podatke. Korisnik također nakon što unese ispravnu e-poštu dobije mail u kojim je šifra potvrde za nastavak registracije. Tijekom unosa šifre koristi se *p:ajax* komponenta koja sa Ajax-om provjerava nakon svakog unosa da li je unesena šifra ispravna te ako je - prikazuje gumbić za dalje. Nakon uspješno unesenih svih podataka i potvrđivanja registracije, oni se moraju spremiti u bazu podataka. Spajanje u bazu vrši se preko *Database.java* klase koja sadrži metodu za spajanje i zatvaranje veze na bazu. Metoda za spajanje koristi JDBC (eng. *Java Database Connectivity*) driver za spajanje na bazu. Ta klasa se zatim koristi u drugim DAO (eng. *Data Access Object*) klasama za spajanje na bazu te izvršavanje SQL naredba. Na slici 20 se vidi klasa *Databasete* metoda za spajanje na bazu, dok na se na slici 21 vidi kako se ona koristi za jednu od metoda koja služi za korisničku prijavu.



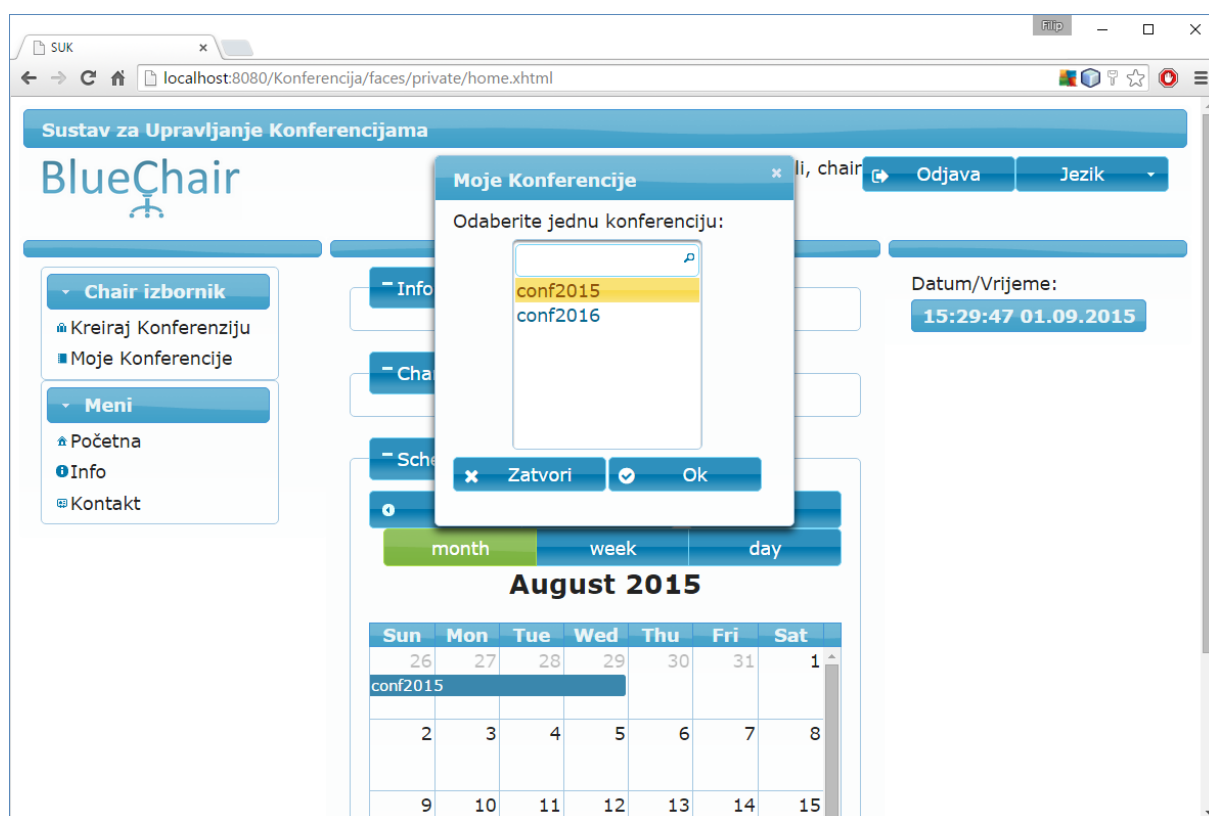
Slika 21: Database klasa sa metodom za spajanje na bazu podataka



Slika 22: Jedna od klasi i metoda koja koristi Database klasu za spajanje te izvršavanje SQL naredbi

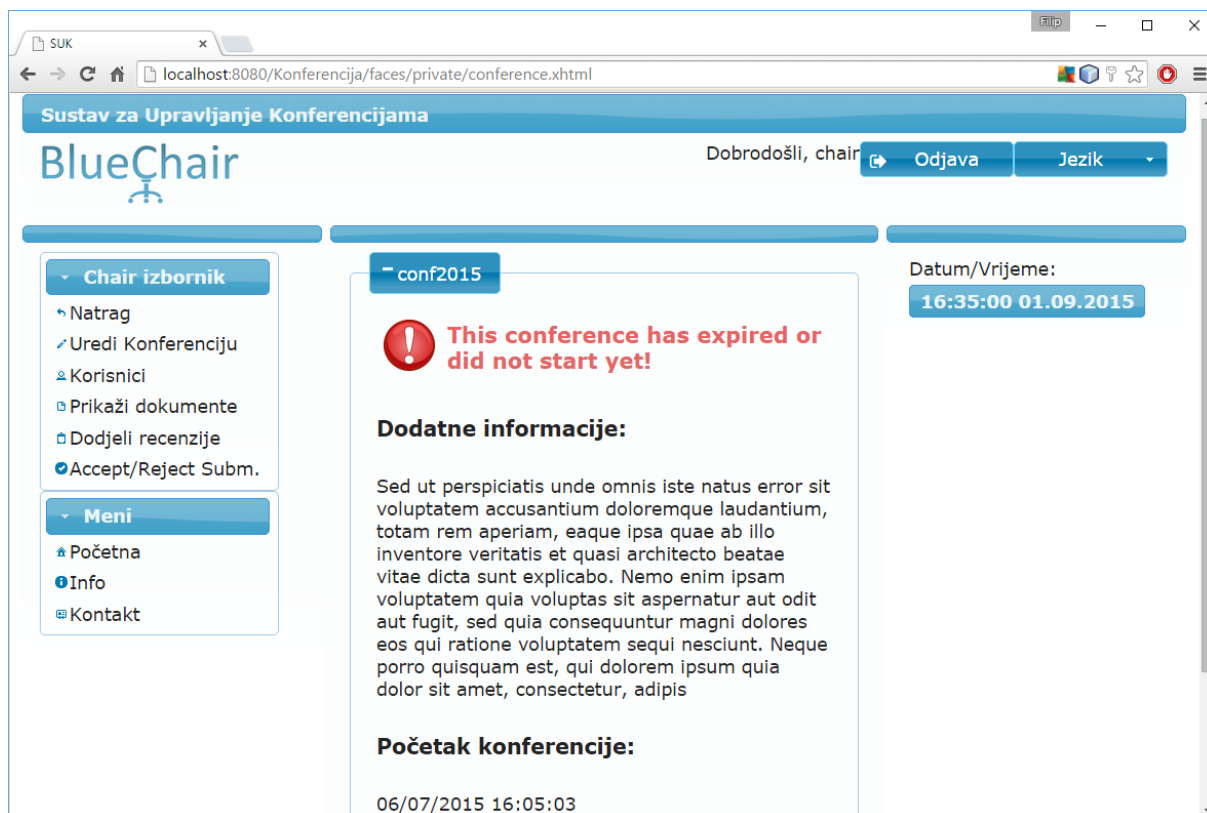
Metoda za prijavu korisnika prima dvije varijable koje korisnik unese, a to su korisničko ime i lozinka. Njih zatim ubaci u upit koji se šalje na bazu da provjeri postoji li korisnik sa tim podacima. Ako upit vrati nula rezultata prijava je neuspješna te metoda vraća *false* vrijednost - u suprotnom prijava je uspješna i vraća *true*. Na sličan način rade i sve ostale metode za rad sa bazom, ovisno koja se za šta koristi. Korisnik svoje podatke unosi preko *p:dialog* komponente sa početne stranice. Ona se pojavi kada korisnik pritisne gumbić za prijavu bez da se korisnika preusmjerava na drugu stranicu. Osim za prijavu ta komponenta se koristi na više mjesta u sustavu baš zbog toga što ne preusmjerava korisnika te čuva preglednost i jednostavnost. Također može mijenjati veličinu, poziciju, minimizirati se, zatvarati i animirati što je dodatno čini fleksibilnom i ugodnom za koristiti.

Nakon prijave ili odabira konferencije treba prikazivati samo one komponente i sadržaj koji su namijenjeni za tog korisnika. Tu "dolazi u igru" jedan atribut kojeg imaju većinom sve komponente - *rendered*. Ako je njegova vrijednost postavljena na *false*, komponenta se neće renderirati tj. prikazivati. Na slici ispod se tako prikazuju samo one komponente i sadržaj koji su nakon prijave dostupni predsjedniku. Također se vidi i prije spomenuta, otvorena *p:dialog* komponenta u kojoj se nalazi lista konferencija koje korisnik može odabrati.



Slika 23: Stranica sustava nakon prijave predsjednika sa otvorenom *p:dialog* komponentom

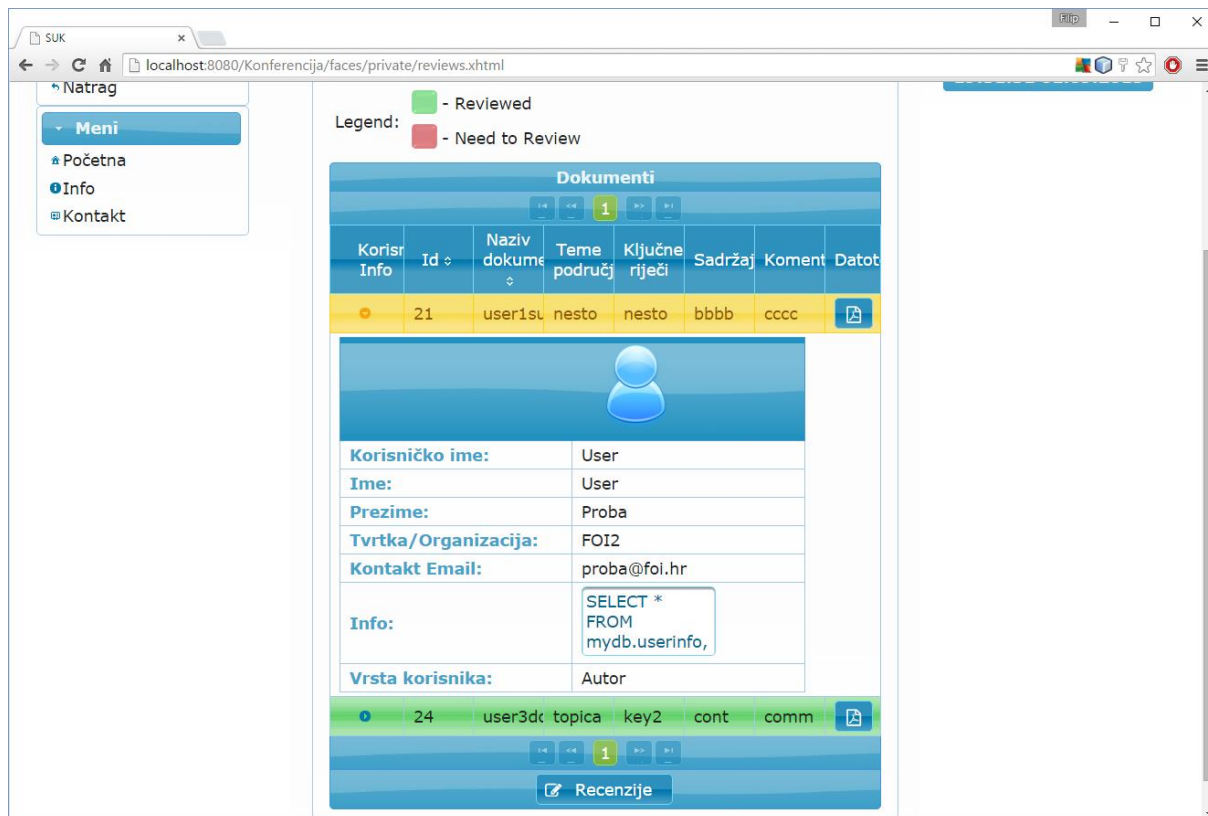
Na stranici predsjednik može i kreirati novu konferenciju te vidjeti početak i završetak svojih konferencija na kalendaru pomoću *p:calendar* komponente. U popisu konferencija korisnik može imati različite uloge za pojedinu konferenciju, ovisno o kojoj konferenciji se radi. Nakon odabira konferencije u kojoj je korisnik predsjednik, otvara se nova stranica sa izbornikom i sadržajem za predsjednika.



Slika 24:Stranica konferencije predsjednika

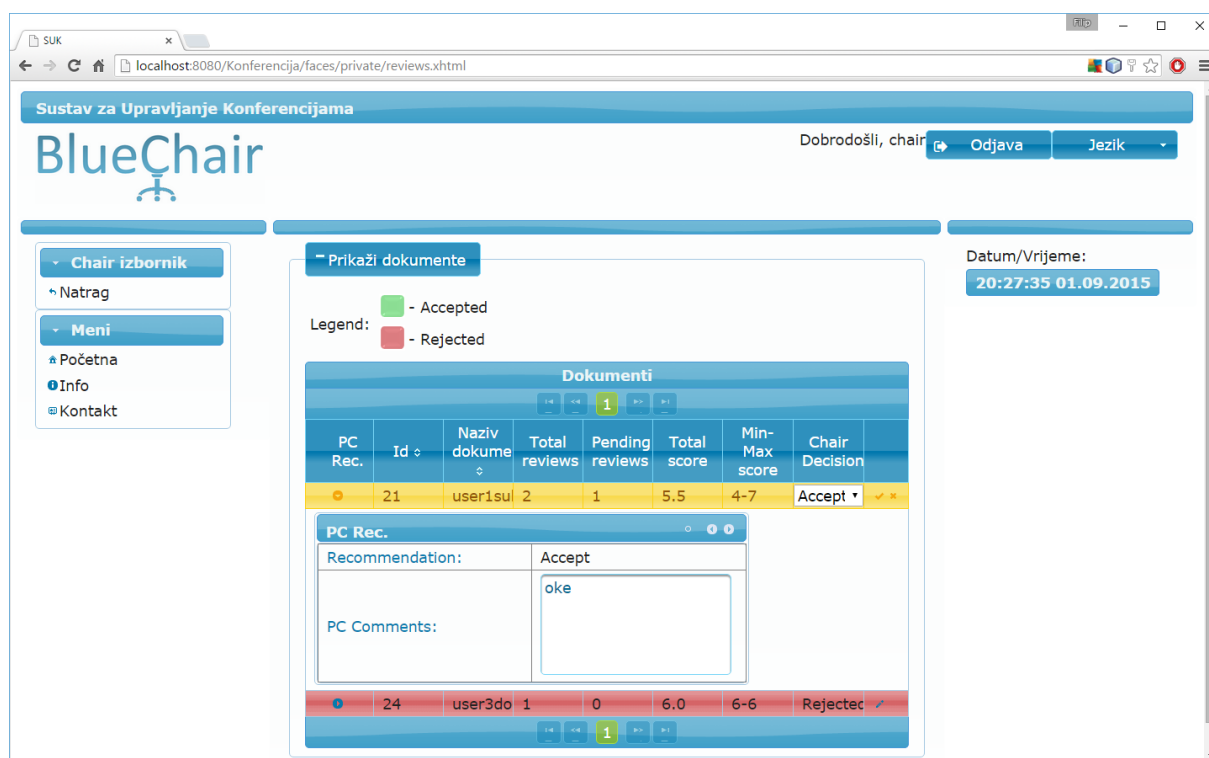
U izborniku se nalaze funkcionalnosti kojima predsjednik može upravljati konferencijom. "Uredi konferenciju" otvara stranicu istu kao i kod kreiranja konferencije no popunjenu sa podacima koji se zatim mogu promijeniti i spremiti. Pod "Korisnici" se otvara novi *p:dialog* koji sadrži *p:carousel* komponentu te akcije za dodavanje/brisanje jednog ili više korisnika kao i njihov tip. U *p:carousel* komponenti na lijepi su način prikazani svi korisnici i njihovi tipovi koji se nalaze u konferenciji. Za prikaz dokumenata koje su objavili autori kao i njihovih recenzija koristi se *p:dataTable* komponenta. To je tablica koja je popunjena sa informacijama o predanim dokumentima kao i podacima o autorima/recenzentima te revizijama i predanim datotekama. Prednost ovoga je da je sve na jednom mjestu te se korisnika ne preusmjerava na mnogo različitih stranica što omogućuje brže i lakše pregledavanje. Način na koji se to implementiralo, a da sve bude pregledno i

funkcionalno je sljedeći. Tablica koja se dobije sa *p:dataTable* popunjena je sa podacima o dokumentu iz baze. Ona za jedan od svojih stupaca ima gumbić koji proširuje red i prikazuje podatke o autoru. To radi pomoću *p:rowToggler* komponente koja se nalazi unutar tog stupca. Ona generira taj gumbić te pritiskom na njega otvara *p:rowExpansion* komponentu koja sadrži podatke o određenom autoru. Način na koji se zna koje podatke, odnosno kojeg korisnika treba prikazati je taj što se koristi *p:ajax* komponenta unutar *p:dataTable* koja ima atribut *event="rowToggle"*. To znači da se šalje Ajax zahtjev kada je red proširen sa podacima reda. Na taj način se zna koji dokument je proširen odnosno koje podatke za autora treba prikazati. Kada neki dokument ima recenzije on je obojan zeleno u protivnom je crveni. Klikom na red koji sadrži recenzije pojavljuje se gumb *Recenzije*, a red postaje žuti. Pritiskom na gumb otvara se *p:dialog* koji sadrži *p:carosuel* sa podacima o recenzijama i recenzentu. Da bi znao koje recenzije otvoriti, koristi se slični način kao i kod podataka o autoru. No ovaj puta je atribut u *p:ajax* komponenti *event="rowSelect"*, što znači da se šalje Ajax zahtjev kada je pritisnut određeni red. Tako sustav zna koje podatke o recenzijama mora prikazati. Kako autor i recenzent moraju predati dokument u PDF-u (eng. *Portable Document Format*), predsjednik ga može pregledati pritiskom na gumbić sa ikonom datoteke. On zatim otvara *p:dialog* koji prikazuje određeni PDF dokument pomoću *p:media* komponente.



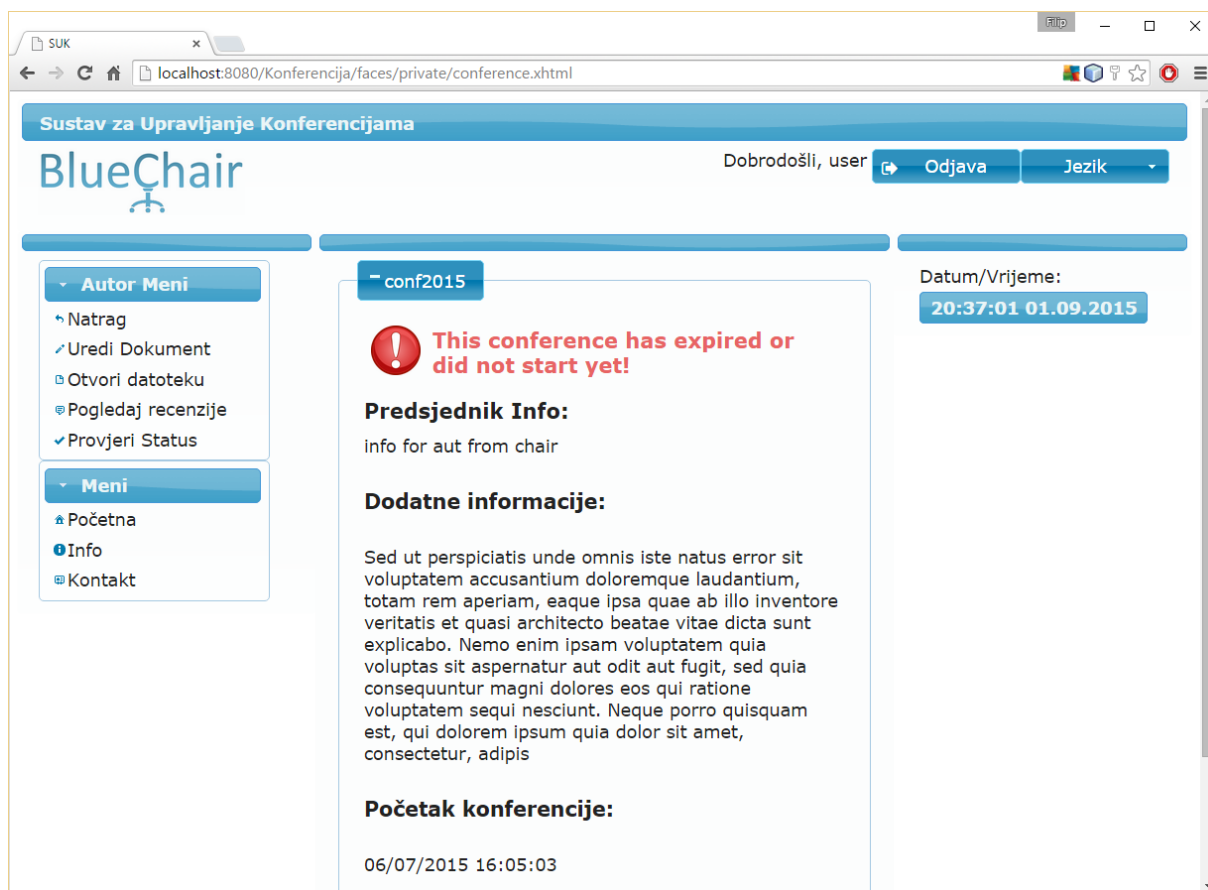
Slika 25: Prikaz podataka o dokumentima, njihovim autorima te recenzijama i recenzentima

Da bi predsjednik mogao gledati recenzije dokumenata prvo im mora dodijeliti recenzente koji će ih recenzirati. To može pomoću funkcionalnosti dodijeli recenzije koja otvara novu stranicu sa listom recenzenata. Pritiskom na recenzenta u listi pojavljuje se *p:pickList* komponenta koja sadrži dostupne i dodijeljene recenzije za tog recenzenta. Predsjednik zatim može birati želi li dodati ili maknuti dodijeljene recenzije za pojedinog recenzenta. U slučaju da se radi o više recenzenata i dokumenata, predsjednik može odabrati automatsku dodjelu. Za to treba samo odabrati broj dokumenata po recenzentu te pritisnuti gumb "dodijeli". U pozadini se zatim pokreće algoritam za automatsku dodjelu, te ako je sve u redu dodjeljuje dokumente recenzentima. Predsjednik te dodjele može vidjeti u gornjoj listi za svakog recenzenta te naknadno modificirati ako je potrebno. Za kraj predsjednik mora birati koju vrstu statusa želi dodijeliti pojedinim predanim dokumentima. U tom slučaju otvara se nova stranica slična onoj za pregled dokumenata, no ona sadrži *p:dataTable* sa drugačijim podacima. Sadrži podatke o dokumentu poput naziva, ukupnih revizija, ukupne ocjene itd, te podatke o preporukama članova odbora za taj dokument. Također za pojedini red (dokument) predsjednik može u stupcu odluka dodijeliti status tom dokumentu.



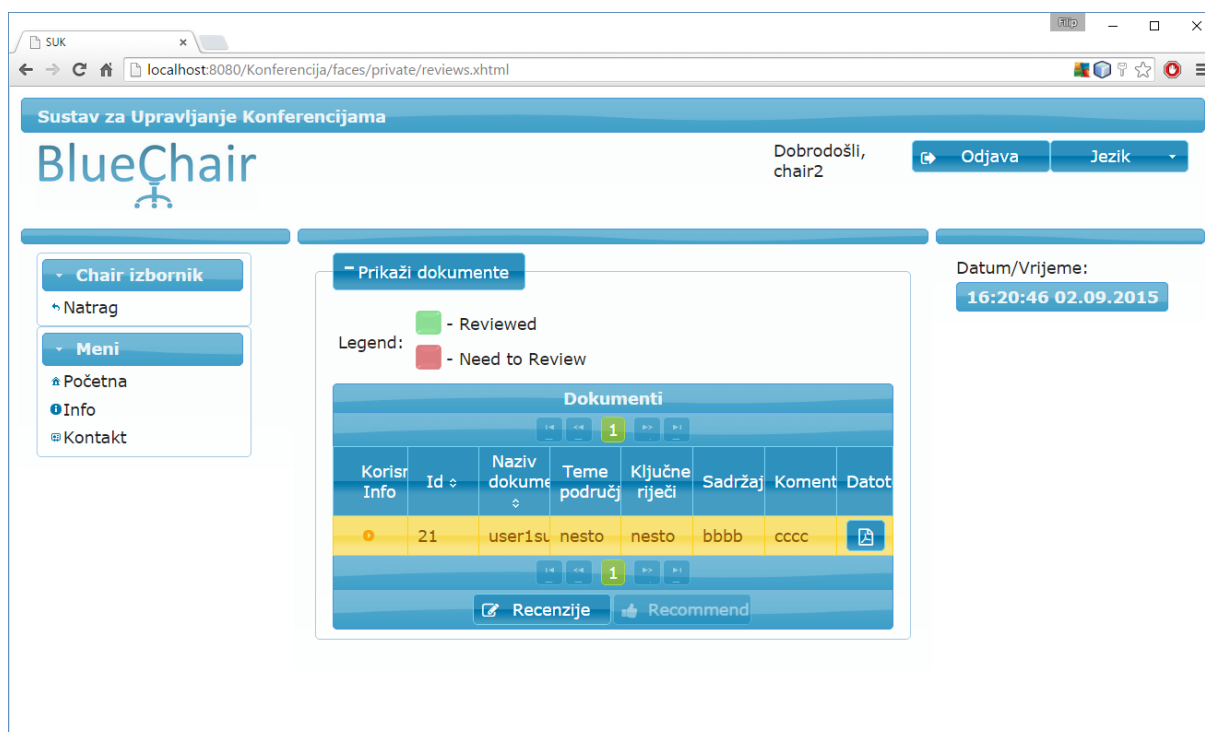
Slika 26: Odluka o dodijeli statusa dokumentu autora

Dakako da bi sve te akcije mogle postojati, moraju postojati i dokumenti autora. Tu dolazi uloga autora. Korisnik koji nema dodijeljenu ulogu predsjednika od strane admina se nakon prijave nalazi na istoj stranici kao i predsjednik, no bez mogućnosti kreiranja novih konferencija. No, može birati između konferencija te također imati različite uloge ovisno o konferenciji koju odabere. Kod uloge autora otvara se stranica sa izbornikom i sadržajem za autora. Glavna funkcionalnost je dakako kreiranje/uređivanje dokumenta. Na stranici za to nalazi se gotova forma koju autor mora ispuniti sa obaveznom priloženom PDF datotekom. Za postavljanje te datoteke koristi se *p:fileUpload* komponenta. Ona prima određenu veličinu i tip datoteke te aktivira slušač koji sprema datoteku u varijablu tipa *UploadedFile*. Nakon što korisnik ispuni formu te pritisne "Spremi", datoteka se kopira sa *IOUtils.copy()* u prije određeni direktorij sa nazivom datoteke *idKorisnika-idKonferencije.pdf*. Na taj način je datoteka fizički spremljena na disku te se može pristupiti preko jedinstvenog naziva. Autor može otvarati datoteku preko funkcionalnosti u meniju na način kako se otvaraju sve datoteke u sustavu preko *p:dialog* komponente koja sadrži *p:media* sa prikazom datoteke. Za kraj autor može pregledati sve recenzije svog dokumenta sa popisa recenzija koje se nalaze u *p:selectOneListbox* komponenti te provjeriti status dokumenta od strane predsjednika.



Slika 27:Stranica za autora

Funkcionalnost za recenziranje kod recenzenata izgleda isto kao i pregled dokumenata kod predsjednika, uz razliku što recenzenti ne mogu vidjeti ostale recenzije za pojedine dokumente već samo napraviti/urediti svoju recenziju. Recenziranje radi na istom principu kao i pregled recenzija kod predsjednika. Klikom na pojedini red odnosno dokument, prikazuje se gumb za recenziranje koji otvara *p:dialog* na kojem se nalazi forma za recenziranje. Kada recenzent spremi recenziju red postaje zelene boje što znači da je dokument recenziran, u protivnom je red crveni. Recenzent može vidjeti autore dokumenata samo ako je predsjednik aktivirao tu mogućnost tijekom kreiranja konferencije - u protivnom je recenziranje anonimno. Na isti način članovi odbora mogu preporučiti dokumente predsjedniku. Razlika je u tome što oni mogu vidjeti sve recenzije dokumenata, informacije o korisniku i recenzentu kao i predsjednik - ako su im te opcije omogućene. Također, nakon što preporuče dokument mijenja se boja reda, tako da se u svakom trenutku zna koje dokumente treba još obraditi. Ako konferencija nije počela ili je istekla, svi tipovi korisnika više ne mogu izvršiti svoje funkcije. Primjer je slika ispod u kojoj član odbora ne može više preporučiti dokument jer je gumb onemogućen zbog isteka konferencije.



Slika 28: Onemogućeno preporučanje dokumenata članova odbora zbog isteka konferencije

Ovo je bio kratak opis kako sustav funkcionira i izgleda. On se, naravno, može dalje nadograđivati i poboljšavati - no osnovne funkcionalnosti za upravljati konferencijom su tu. U radu nije detaljno opisan pozadinski kod, ali se može dobiti dobar uvid u način funkcioniranja sustava.

4. Zaključak

Proučivši informacije vezane uz sustave za upravljanje konferencijama može se zaključiti da se radi o rastućem trendu uporabe istim, naročito zbog njihovim funkcionalnosti kao i rasta broja konferencija na godišnjoj razini. Zbog tog se razloga proučilo i opisalo nekoliko sustava za upravljanje konferencijama, te se moglo vidjeti koje su im prednosti, a koji nedostaci. Osim što neki nude sustav kao uslugu (što smanjuje troškove ali povećava rizik sigurnosti) glavne prednosti tih sustava su njihove funkcionalnosti za upravljanje konferencijama, kojih ima mnogo, a rade bez greški. Naravno, to dolazi uz neku cijenu pa oni nisu potpuno besplatni. Najveći nedostatak proučenih sustava bi bio nekorištenje modernih tehnologija poput Ajaxa. Zbog toga su neke funkcionalnosti nepregledne i naporne, a i sam izgled sustava nije elegantan, moderan i lijep. Također se moglo zaključiti da im je glavna razlika, osim izgleda, način na koji se automatski dodjeljuju dokumenti recenzentima. Ovo je bitno budući da se radi jednoj od važnijih funkcionalnosti pogotovo za veće i ozbiljnije konferencije. Razlika se nalazi u algoritmima koji se koriste za tu funkcionalnost odnosno kako se ona implementira. No, sva rješenja koriste isti način za prikupljanje podataka potrebnih za njezinu izvedbu - a to je eksplicitno. To znači da autori i recenzenti sami daju potrebne informacije. Zbog toga je za razvoj vlastitog sustava jedan dio rada posvećen samo za algoritam koji se implementira u vlastito rješenje da bi se dobila željena funkcionalnost. Algoritam pruža ujednačenu raspodjelu radova recenzentima, te jamči da ako dokument ima barem jednu ključnu riječ zajedničku sa recenzentom, da će biti dodijeljen. Također radi brže u odnosu na druge algoritme što je praktički dokazano u radu od izvora algoritma. Osim toga, sustav je trebalo napraviti bez nedostataka koji su spomenuti. Na taj način se dobije poboljšani sustav dok funkcionalnosti ostaju iste. Zbog tog razloga su se za razvoj sustava upotrijebile tehnologije koje će osigurati da sustav radi kako treba, no da ujedno ima moderan i lijep dizajn te lakše i bolje implementirane funkcionalnosti. Pritom se misli na slučaj u praksi gdje korisnik može bez puno napora, klikanja i razmišljanja obavljati iste funkcionalnosti brže, bolje i efikasnije. Također, zbog svojeg responzivnog dizajna korisnik sve to može raditi i preko drugih uređaja (mobitela, tableta), a da pritom sustav ne izgubi na funkcionalnosti i efikasnosti, što za opisane sustave ne vrijedi. Za sad jedina mana sustava je što nema puno naprednih funkcionalnosti, a najveći razlog tome je što se sustav razvijao u kratkom vremenskom roku te od strane jedne osobe. No, on se lako može nadograditi sa puno novih funkcionalnosti i poboljšanja - naročito ako će se sustav dalje koristiti.

5. Popis slika

Slika 1: Prikaz početne stranice OpenConf.....	8
Slika 2: Demo profesionalne verzije sustava	9
Slika 3: Jedna od funkcionalnosti OpenConf sustava, prihvaćanje/odbijanje radova.....	10
Slika 4: Početna stranica COMS	12
Slika 5:Usluga kreiranja i traženja konferencija	13
Slika 6:Lista troškova sa opisom pojedinog modula i cijenom za broj korisnika	14
Slika 7: Demo početne stranice sustava, prijavljenog kao autor	15
Slika 8:Funkcionalnost dodavanja novog dokumenta kao autor.....	16
Slika 9:Početna stranica ConfTool-a.....	18
Slika 10:Početno sučelje ConfTool Pro sustava.....	19
Slika 11:Funkcionalnost recenziranja dokumenata	20
Slika 12:Pregled procesa razvoja softvera.....	24
Slika 13: JSF arhitektura	26
Slika 14: Interakcija SQL-a sa MySQL RDBMS	30
Slika 15:Slučajevi korištenja sustava za upravljanje konf. u System Boundary Diagram-u ...	34
Slika 16: ERA model sustava	35
Slika 17: Koraci 1-4, algoritma za automatsku dodjelu dokumenata recenzentima	38
Slika 18:Dodavanje ovisnosti pomoću Maven-a.....	39
Slika 19:layout.xhtml datoteka, sa spomenutim komponentama	40
Slika 20: Izgled početne stranice sustava	41
Slika 21:Database klasa sa metodom za spajanje na bazu podataka	42
Slika 22:Jedna od klasi i metoda koja koristi Database klasu za spajanje te izvršavanjeSQLa	42
Slika 23:Stranica sustava nakon prijave predsjednika sa otvorenom p:dialog komponentom.	43
Slika 24:Stranica konferencije predsjednika	44
Slika 25: Prikaz podataka o dokumentima, njihovim autorima te recenzijama i recenz.	45
Slika 26:Odluka o dodijeli statusa dokumentu autora.....	46
Slika 27:Stranica za autora	47
Slika 28: Onemogućeno preporučanje dokumenata članova odbora zbog isteka konferencije	48

6. Literatura

- [1] Pongsathorn N., Yachai L., "*Towards a RESTful Process of Conference Management System*", Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong
- [2] Andreas P., Rudolf M., Andreas R., "*Improving Scientific Conferences by enhancing Conference Management System within information mining capabilities*", Vienna University of Technology, 2006.
- [3] ICCA Statistics Report 2013, Dostupno 1.9.2015. na <http://iccardata.com/newsarchives/archivedetails.cfm?id=3537>
- [4] Zakon Group, "*Open Conf Conference Management System*", Dostupno 1.8.2015. na <http://openconf.org>
- [5] Marianne Mandl, "*COMS Conference Management System*", Dostupno 1.8.2015 na <http://conference-service.com>
- [6] ConfTool GmbH, "*ConfTool Conference Management System*", Dostupno 1.8.2015 na <http://conftool.net>
- [7] Md. J. H., Tribhuvan K. T., Sandeep K. S., "*Survey of Conference Management Systems*", International Journal of Computer Applications (0975 – 8887) Volume 2 – No.2, May 2010
- [8] Yordan K., "*Architecture of a Conference Management System Providing Advanced Paper Assignment Features*", International Journal of Computer Applications (0975 – 8887) Volume 34 – No.3, November 2011
- [9] Ryan, M. "*Cloud Computing Privacy Concerns on Our Doorstep*", Communications of the ACM vol. 54 (1), 2011
- [10] Rigaux Ph. "*An Iterative Rating Method: Application to Web-based Conference Management*", Proceedings of the 2004 ACM symposium on applied computing
- [11] Benjamin J. Evans, David Flanagan (2014) *Java in a Nutshell*. USA
- [12] Java Dokumentacija, Oracle, Dostupna 1.8.2015 na <http://docs.oracle.com/javase/tutorial>
- [13] Dirk (2000), *Framework Design: A Role Modeling Approach* (PDF), Swiss Federal Institute of Technology – Dostupno na 1.8.2015. <http://dirk.kriehle.com/computer-science/research/dissertation/diss-a4.pdf>
- [14] Vincent M., Timothy O'B. (2005) *Maven: A Developer's Notebook*, USA
- [15] Sudheer J., Ramkumar P. (2014), *Prime Faces Blueprints*, Packt Publishing Ltd., Birmingham UK

- [16] Jennifer N.R. (2012), *Learning Web Design: A beginner's guide to HTML, CSS, JS and Web graphics*, O'Reilly, California USA
- [17] PrimeFaces dokumentacija, PrimeTek, Dostupna 1.8.2015 na
<http://www.primefaces.org/documentation>
- [18] Steve S.(2002), *MySQLBible* , WileyPublishing, Inc., New York
- [19] Robert C.M.(2002), *UML for Java Programmers*, Prentice-Hall, Inc. USA
- [20] Yordan K., "An algorithm for automatic assignment of reviewers to papers"
International Conference on Computer Systems and Technologies - CompSysTech', 2006