Aplikacija za rješavanje problema rasporeda

Macan, Doroteo

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:812038

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

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

Image not found or type unknown

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

zir.nsk.hr

Image not found or type unknown

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

DOROTEO MACAN

APLIKACIJA ZA RJEŠAVANJE PROBLEMA RASPOREDA

Završni rad

Pula, 2016. godine

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

DOROTEO MACAN

APLIKACIJA ZA RJEŠAVANJE PROBLEMA RASPOREDA

Završni rad

JMBAG: 0303045994, redoviti student
Studijski smjer: Informatika

Predmet: Strukture podataka i algoritmi

Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Tihomir Orehovački

Pula, rujan 2016. godine

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Doroteo Macan, kandidat za prvostupnika informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, 19.9.2016. godine

IZJAVA
o korištenju autorskog djela

Ja, Doroteo Macan dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom „Aplikacija za rješavanje problema rasporeda“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.
Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 19.9.2016. godine

Potpis

_____

Sadržaj

1. UVOD 3

2. PROBLEM RASPOREDA SATI 5

2.1 OPIS PROBLEMA 5

2.2 POSTOJEĆE APLIKACIJE ZA RJEŠAVANJE PROBLEMA RASPOREDA 6

2.2.1 ASC RASPOREDI 6

2.2.2 EMS CLASSROOM SCHEDULING 10

2.2.3 MIMOSA SCHEDULING SOFTWARE 10

2.2.4 PRIME TIMETABLE 12

2.2.5 USPOREDBA APLIKACIJA PO ZAHTJEVIMA 16

3. KORIŠTENE TEHNOLOGIJE 17

3.1 ODABIR TEHNOLOGIJE 17

3.2 PHP 18

3.3 MYSQL 19

3.4 HTML5 20

3.5 CSS 21

4. PROGRAMSKO RJEŠENJE 22

4.1 HODOGRAM KORIŠTENJA 22

4.2 IZBORNIK APLIKACIJE 23

4.3 NASLOVNA 24

4.4 PROFESORI 24

4.5 KOLEGIJI 24

4.6 STUDENTI 25

4.7 DVORANE 26

4.8 NASTAVA 27

4.9 GENERIRAJ 29

4.10 RASPOREDI 30

4.10.1 RASPORED GRUPA 30

4.10.2 RASPORED PROFESORA 32

4.10.3 RASPORED DVORANA 33

5. BAZA PODATAKA 34

5.1 OPIS BAZE PODATAKA 34

5.2 TABLICA DVORANA 35

5.3 TABLICA PROFESOR 35

5.4 TABLICA KOLEGIJ 35

5.5 TABLICA TIP_NASTAVE 35

5.6 TABLICA GRUPA_NAZIV 36

5.7 TABLICA GRUPA_GODINA 36
5.8 TABLICA GRUPA 36
5.9 TABLICA SEMESTAR 36
5.10 TABLICA NASTAVA 36
5.11 TABLICE NASTAVA_ZIMSKI I NASTAVA_LJETNI 37
6. FUNKCIONALNOSTI 38
6.1 UVOD U FUNKCIONALNOSTI 38
6.2 PREGLED PODATAKA 38
6.3 DODAVANJE PODATAKA 40
6.4 IZMJENA PODATAKA 41
6.5 BRISANJE PODATAKA 43
6.6 TABLICA NASTAVE 43
6.7 GENERIRAJ 48
6.8 FUNKCIJA RASPORED 49
6.9 FUNKCIJA GENERIRAJ 52
7. ZAKLJUČAK 57
LITERATURA 58
POPIS SLIKA 60
POPIS PRIMJERA 61
POPIS TABLICA 62

1. UVOD

U obrazovnim ustanovama, od osnovnih škola, srednjih škola do fakulteta, javlja se na početku svake školske godine potreba za kreiranjem novog rasporeda sati. Cilj je kreirati raspored sati koji omogućava da se nastava odvija prema zadanom planu i programu. Potrebno je obuhvatiti sve nastavne grupe, predmete i profesore, te isključiti subjektivni pristup odgovorne osobe zadužene za stvaranje rasporeda. Kako bi se eliminirala mogućnost greške, pristupa se programskim rješenjima.

Ovaj rad opisuje web aplikaciju kojoj je zadatak riješiti taj problem. Web aplikacija je prvobitno namijenjena fakultetima zbog mogućnosti koje su sadržane u njoj, no moguće ju je koristiti i u drugim sličnim ustanovama. Kako bi raspored sati bio dobar za uporabu potrebno je postaviti nekoliko uvjeta kojih se algoritam sortiranja nastave u raspored mora držati. Najvažniji uvjeti su: da se ne mogu održavati dva različita predavanja u isto vrijeme u istoj dvorani, da jedan profesor i studentska grupa ne mogu biti prisutni na više predavanja odjednom, te da broj studenata ne smije nadilaziti broj mjesta u dvorani, posebno u slučajevima kada su u nastavi potrebna računala za rad na njima. Uz takve uvjete se mogu postaviti i neki drugi, ovisno o potrebi ili želji, poput toga da određena studentska grupa nema više od određenog broja sati dnevno ili da veći broj nastave istog kolegija bude uzastopno u bloku sati.

Aplikacija je namijenjena za svakodnevno pregledavanje od strane korisnika. Jednom kreirani raspored moguće je mijenjati svaki puta kad se za to pojavi potreba, odnosno kad dođe do promjena uvjeta koje je potrebno zadovoljiti. Svrha aplikacije je generiranje rasporeda pomoću algoritma. Za korištenje aplikacije ne postoji potreba za identifikacijom i autorizacijom, već samo za autentifikacijom. Obzirom da ne postoji korisnički profili, aplikacija se koristi od strane jednog sveučilišnog tijela pomoću lozinke za njeno korištenje. Generiranje rasporeda, kao glavne mogućnosti koju ova aplikacija pruža, moguće je tek nakon što se ubace svi željeni podaci u bazu podataka, a iste je moguće kasnije dodavati, uklanjati i izmjenjivati po potrebi. Nakon što je raspored generiran moguće ga je pregledavati po različitim elementima. Podijeljen je na zimski i ljetni semestar, sa mogućnošću pregleda po dvoranama, profesorima i studentskim grupama.

Završni rad se sastoji od uvoda, pet glavnih poglavlja i zaključka. Drugo poglavlje opisuje problem rasporeda i njegove zahtjeve, te daje osvrt na postojeće aplikacije, njihove prednosti i nedostatke. Treće poglavlje se osvrće na tehnologije koje su korištene za izradu aplikacije. Procedura i opis rada u web aplikaciji prikazani su u četvrtom poglavlju. Peto poglavlje opisuje bazu podataka, koja sadržava sve podatke koji su potrebni aplikaciji. U šestom poglavlju je opisana funkcionalnost, uz priložene primjere koda.

2. PROBLEM RASPOREDA SATI

2.1 OPIS PROBLEMA

Raspoređivanje je proces u kojem se skup elemenata dodjeljuje skupu aktivnosti u određenom vremenskom razdoblju sa ciljem optimizacije rezultata. (Gantt, 2008)

Kako bi se generirao raspored sati moraju biti zadovoljeni svi strogi zahtjevi i pravilno postavljene sve fizičke mogućnosti vezano uz relevantne elemente.

Elementi koje raspored sadržava su: studentske grupe, profesori, dvorane, kolegiji i vrijeme. (Bedek, 2008)

Strogi zahtjevi koje svaki raspored mora zadovoljiti su:

- Profesor u jednom nastavnom satu može:
  - Predavati samo jednoj grupi studenata
  - Predavati u samo jednoj dvorani
  - Predavati samo jedan kolegij
- Studentske grupe u jednom nastavnom satu mogu:
  - Slušati samo jedan kolegij
  - Prisustvovati predavanju u samo jednoj dvorani
- Dvorane u jednom nastavnom satu mogu primiti:
  - Manji ili jednak broj studenata s obzirom na broj mjesta
  - Samo jednu grupu studenata
  - Održavanje samo jednog kolegija
- Kolegiji:
  - Kojima su potrebna računala moraju se održavati u dvorani koja ima računala
- Vrijeme:
  - Nastava se mora održati unutar petnaest školskih sati, u koje spada sedam sati jutarnje i osam sati popodnevne smjene

Osim tih strogih zahtjeva postoje i blaži zahtjevi koji nisu fizički određeni, ali im je cilj postići da raspored i nastava budu ugodniji za polaznike. Time je omogućeno da se studenti rasterete obveza koje imaju u jednom danu.

Blagi zahtjevi su:

- Ako postoji više sati nastave jednog kolegija istog tipa (predavanja, seminari, vježbe), koje studenti moraju slušati u jednom tjednu, tada je ta nastava smještena u blok sate koji se moraju održati u jednom danu;
- Maksimalan broj sati koje studenti mogu imati u jednom danu je sedam, osim ako u tih sedam sati ulazi i početak održavanja bloka sati, tada se nastava održava do kraja trajanja bloka sati.

Ispunjavanje tih zahtjeva osigurava uvjete za generiranje rasporeda. Iako su blagi zahtjevi manje važni nego strogi, za program su oni isto bitni jer aplikacija neće u cijelosti generirati raspored ako se krši bilo koji od zahtjeva. Problem generiranja može se riješiti korištenjem raznih postupaka koji ovise o zahtjevima problema. (Đurasević, 2015) Ako se unutar samog koda isključe blagi zahtjevi, algoritam generira raspored koji je upotrebljiv, no nije toliko prihvatljivo posložena nastava kao u rasporedu koji ima obuhvaćene sve zahtjeve. Isključivanjem strogih uvjeta posve se gubi upotrebljivost stvorenih rasporeda, jer fizički nije moguće na taj način održavati nastavu. Moguće je dodavanje naknadnih blagih zahtjeva, ali nije preporučljivo dodavati veću količinu, jer se time povećava vjerojatnost da raspored neće podržati sve zahtjeve.

Raspored sati koji zadovoljava stroge zahtjeve može se izvršiti u polinomnom vremenu, dok optimizacija rasporeda sati spada u MP težak problem. Izrada rasporeda se ubraja u razred najtežih problema koje definira računalna znanost. Bit problema je u njegovoj složenost i činjenici da ima toliko mogućih rješenja da ih niti računalo nije u mogućnosti sve pretražiti i pronaći najbolje. (Pribil, 2012)

2.2 POSTOJEĆE APLIKACIJE ZA RJEŠAVANJE PROBLEMA RASPOREDA

2.2.1 aSc RASPOREDI[1]

aSc Rasporedi (eng. aSc Timetables) je trenutno najveća komercijalna aplikacija koja se bavi isključivo problemom organiziranja rasporeda sati. Prva verzija aplikacije je izašla prije više od 20 godina. Koristi se u 173 zemlje svijeta, u preko 150 000 škola.

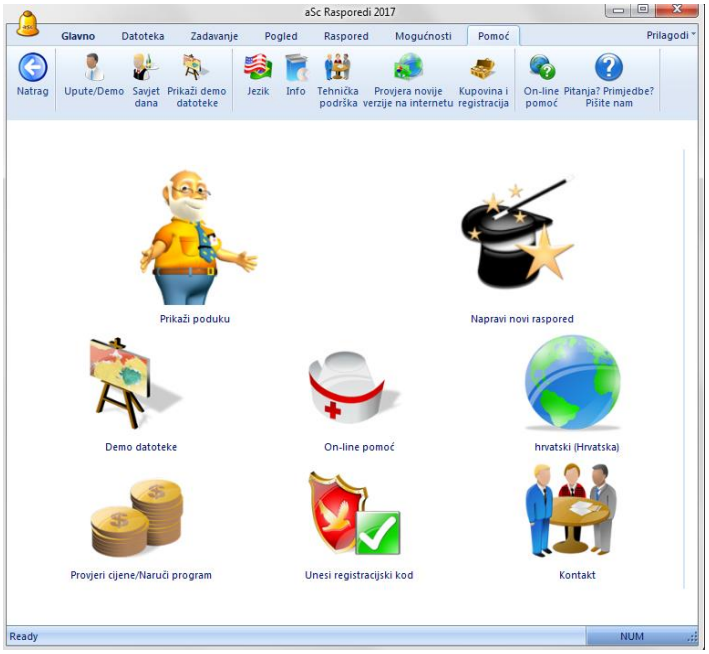
[1] Web stranica aplikacije: http://www.asctimetables.com/

Dostupna je na mnogim jezicima, a jedan od njih je i hrvatski. Cijena aplikacije se kreće od 399$ za osnovne i srednje škole do 3 995$ za izradu individualnih rasporeda sati za učenike sa posebnim potrebama, dostupna je i besplatna, probna verzija. Aplikacija se može koristiti na Windows i MAC računalima. Nakon stvaranja rasporeda, aplikacija pruža mogućnost ispisa za svaki razred (studentsku grupu), dvoranu i profesora. Ispisani dokument sadržava vodeni žig aplikacije.

Funkcije koje pruža aSc Rasporedi (Applied Software Consultants, 2016):

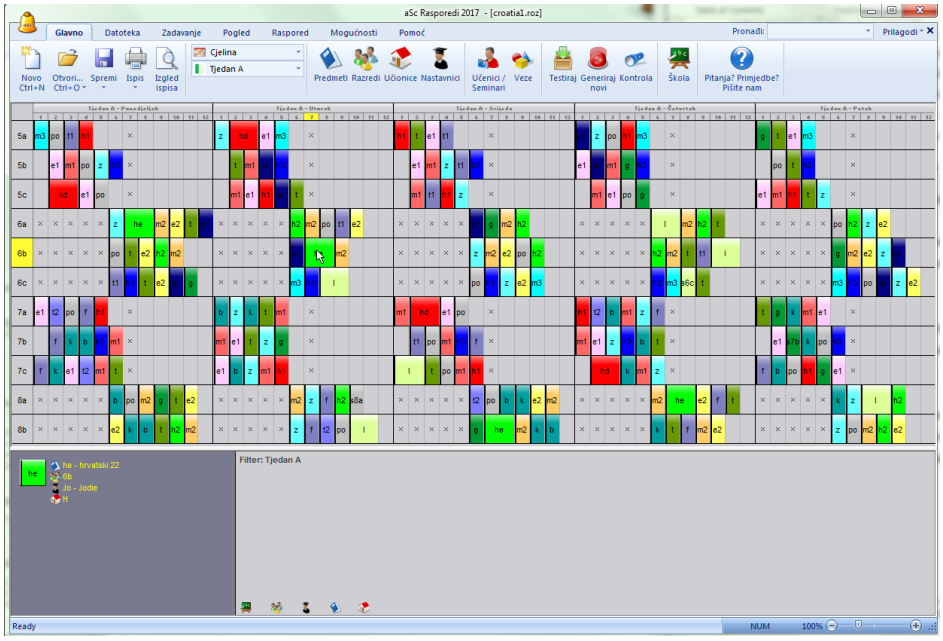
- Automatsko podešavanje
- Ručno podešavanje
- Pregled rasporeda sati
- Jednostavan unos podataka
- Mobilni telefon
- Uvoz podataka
- Zgrade (škole, objekti...)
- Prilagodba na korisnika
- Asc-ove zamjene
- Asc-ov Edupage
- Regionalne specifičnosti
- Asc-ov Raspored sati online
- Elektronski \"Dnevnik\"
- Izborni predmeti

Aplikacija je u potpunosti napisana u odabranom jeziku. Funkcionalnosti se nalaze u različitim karticama koje su vrlo pregledne. Funkcionalnosti imaju jasne nazive i uz njih pripadajuće ikone, što se može vidjeti na slici 1.



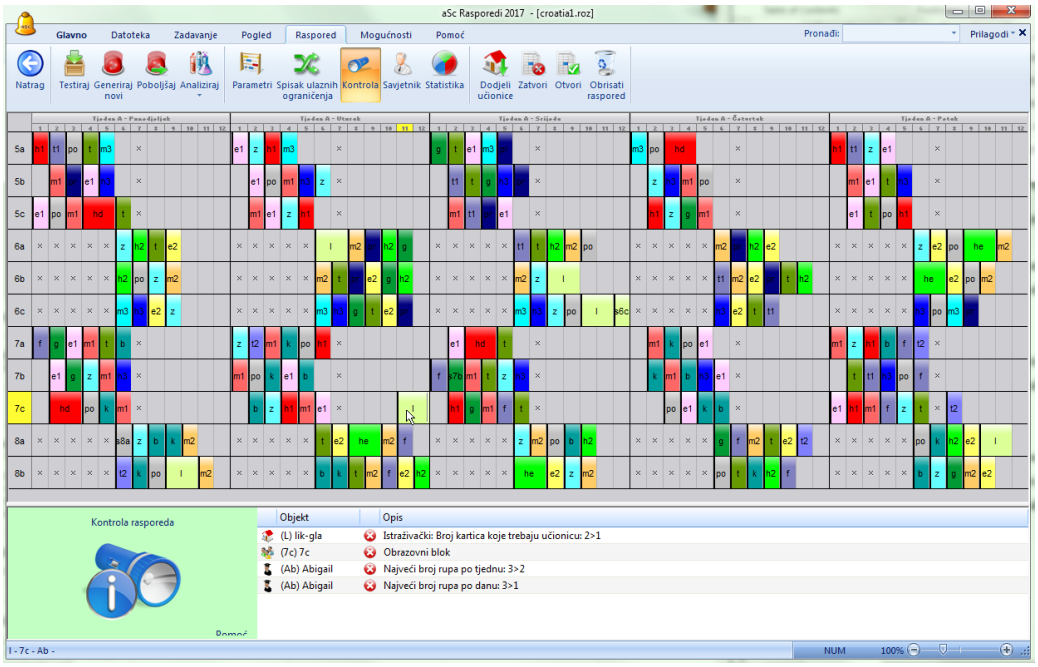
Slika 1. aSc Rasporedi - Naslovna stranica

Prvi prozor nudi nekoliko osnovnih opcija poput: prikaza poduke, pokretanje izrade rasporeda, pregleda demo datoteka, on-line pomoći, odabira jezik i informacija o kontaktu. Kad su uneseni svi podaci i kriteriji raspored se generira i prikaže po razredima poredanim u redove. Polje sadrži predmet nastave, informacija o istome je prikazana u donjem lijevom kutu, što je prikazano na slici 2.

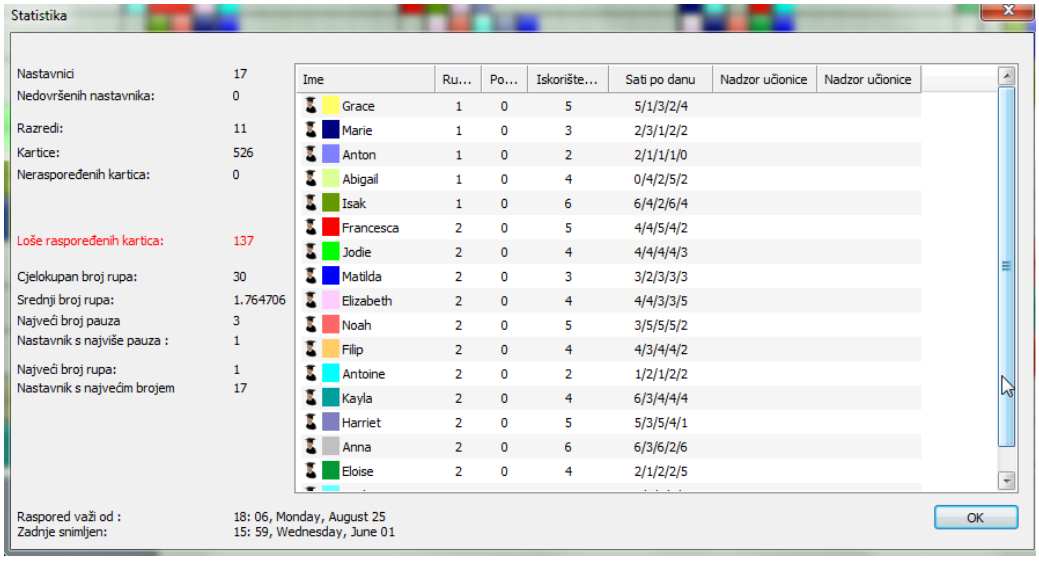


Slika 2. aSc Rasporedi - Prikaz rasporeda

Predmeti se mogu pomoću miša prenositi na druga polja u retku. Aplikacija momentalno provjerava je li to dopuštena radnja, ako nije ispiše obavijest o tome. Korisnik može nakon što je posložio raspored kako njemu odgovara, ili ga ostavio u stanju kakvom ga je aplikacija ponudila, pokrenuti funkciju kontrole, kao na slici 3. ili pogledat statistiku rasporeda, kao na slici 4. Krajnji raspored se tada može isprintati i fizički podijeliti učenicima i profesorima.



Slika 3. aSc Rasporedi - Kontrola



Slika 4. aSc Rasporedi - Statistika

2.2.2 EMS CLASSROOM SCHEDULING[2]

EMS je aplikacija koja se ne bavi samo rasporedom nastave, nego pruža i mogućnost stvaranja rasporeda sastanaka, rasporeda dvorana za sastanke, rasporeda događaja, rasporeda ureda i prikaza statistike za sve navedeno. Aplikacija je za desktop računala. Nema ponuđenu cijenu, niti probnu verziju. Potencijalni korisnik za početak mora na web stranici odabrati iz kojeg razloga želi koristiti njihove usluge, unesti osobne podatke, e-mail adresu, te mora pričekati povratni mail od autora, čime započinje dogovor za kupnju njihovih usluga.

Funkcije koje pruža aplikacija (EMS Software, 2016):

- Raspored po dvoranama, ovisno o veličini i opremi
- Raspored po preferencijama profesora
- Automatizirano raspoređivanje
- Mogućnost izmjene po potrebi
- Uvoz informacija o kolegijima i profesorima
- Brza objava stvorenih rasporeda za profesore i studente
- Zakazivanje ispita
- Odabir željenog postotka popunjenosti dvorane
- Prikaz neispunjenih zahtjeva
- Prikaz dvije ili više nastava koje mogu biti spojene

2.2.3 MIMOSA SCHEDULING SOFTWARE[3]

Mimosa je neovisna o tome je li raspored namijenjen za osnovne škole, srednje škole, fakultete ili nešto slično. Prva verzija aplikacije je izašla 1996. godine. Besplatna probna verzija traje 60 dana, doživotna besplatna verzija ima ograničenje od 80 unosa, potpuna verzija ima cijenu ovisno o broju potrebnih unosa od 800€ do 8 000€, cijena raste 500€ za svakih 500 unosa. Aplikacija je dostupna na svim verzijama Windowsa od Windowsa 98, a za Mac OS i Linux pružene su mogućnosti rada preko VirtualBox-a i sličnih rješenja.

[2] Web stranica aplikacije: https://www.emssoftware.com/
[3] Web stranica aplikacije: http://www.mimosasoftware.com/

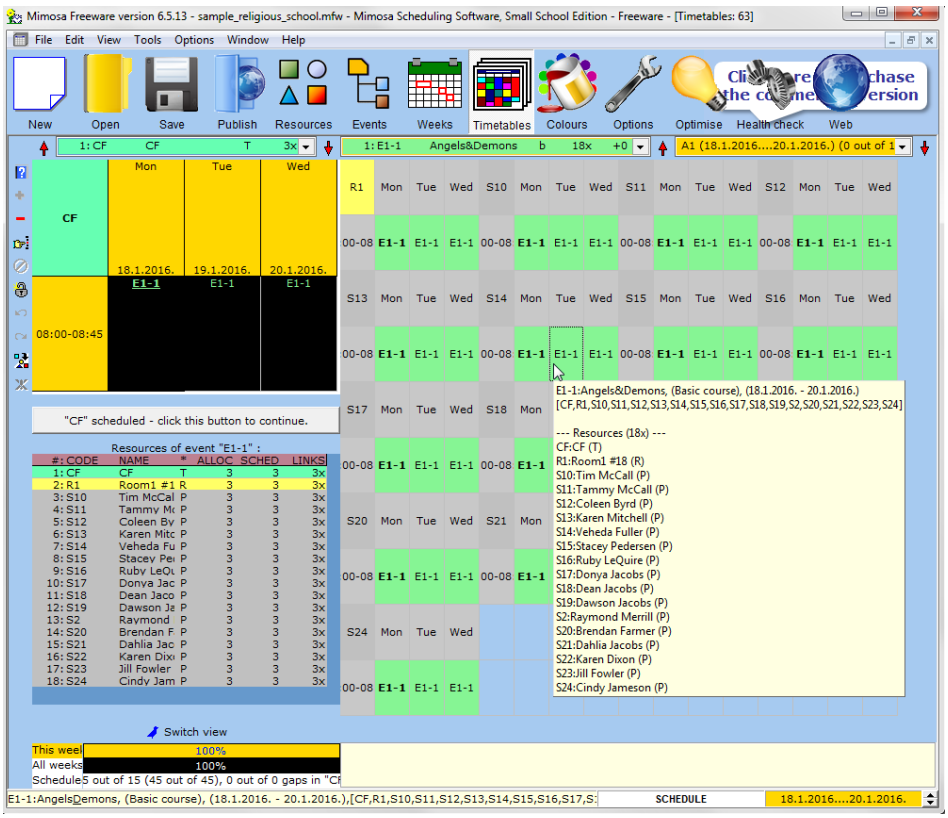
Funkcije koje pruža Mimosa (Mimosa Software Ltd., 2016):

- Spajanje učitelja sa razredom
- Raspored po smjerovima
- Raspored po razredima
- Dodjeljivanje predmeta učitelju
- Raspored po učionicama

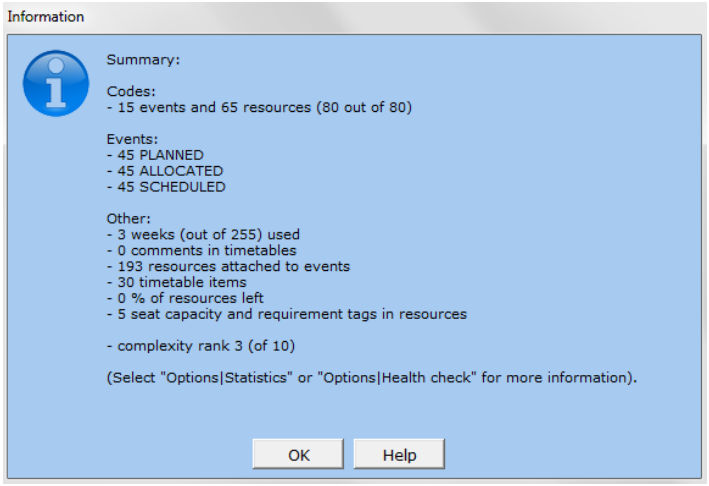
Aplikacija ima prikazane opcije u alatnoj traci sa nazivima uz velike ikone (slika 5). Korisnici koji žele vidjeti demo verzije rasporeda ograničeni su otvaranjem istih, jer se svi podaci preko 80 unosa, koje dopušta besplatna verzija, nasumično brišu.

Prikaz rasporeda, koje se može vidjeti na slici 5., je kaotičan, sa jakim bojama. Bez čitanja uputa za korištenje prikazane informacije su nejasne. Polja su nepregledna i ne sadrže potpune informacije.

Optimizacija rasporeda provjerava samo je li sva nastava raspoređena u raspored. Na alatnoj traci postoji opcija *Health check* (hrv. „provjera ispravnosti“) koja pruža sažete informacije o stvorenom rasporedu, prikazane na slici 6.



Slika 5. Mimosa - Raspored



Slika 6. Mimosa - Provjera ispravnosti rasporeda

2.2.4 PRIME TIMETABLE[4]

Prime Timetable je moderna web aplikacija za upravljanje rasporedom nastave. Neovisna je o platformi i uređaju na kojem se koristi. Prilagođena je radu preko desktop računala, tableta ili pametnog telefona. Aplikacija generira raspored ovisno o zadanim zahtjevima, stvoreni rasporedi se mogu dijeliti online i printati. Za korištenje je dovoljno samo se registrirati u sustav. Pruženo je 30 dana besplatnog rada u aplikaciji sa svim mogućnostima. Cijena aplikacije je nakon toga 149€ godišnje za ustanove do 40 profesora ili 299€ godišnje za ustanove do 100 profesora. U slučaju da je potrebna veća količina profesora treba se kontaktirati autora oko dogovora cijene. Ako se sklapa ugovor na više od godinu dana, godišnja cijena pada ovisno o vremenu za koje je sklopljen ugovor.

Funkcije koje pruža ova aplikacija (PrimeTimetable.com, 2016):

- Uvoz podataka koristeći kopiraj/zalijepi
- Postavljanje dana i vremena trajanja sata
- Upravljanje kolegijima, dvoranama, profesorima, razredima, grupama
- Upravljanje nastavom sa više od jednog profesora, razreda, grupe ili dvorane
- Uređivanje rasporeda koristeći miša
- Izbjegavanje sukoba pomoću master, pojedinačnih i dodatnih pogleda
- Zabrana pozicija markerom, označavanje sa pin alatom i brisanje sa spužvom

[4] Web stranica aplikacije na: http://www.primetimetable.com/

- Spajanje kartica pomoću spajalica za zakazivanje zajedničke nastave
- Uređivanje, spajanje/razdvajanje, kopiranje/lijepljenje i prilagođavanje izgleda kartica
- Označavanje neželjene pozicije s žutim markerom i obavezne s plavim markerom
- Rješavanje ograničenja i sukoba klikom na „čarobnu tipku“
- Čitanje savjeta i trikova kako bi se dobio maksimum iz automatski generiranog rasporeda
- Zvono pokazuje u stvarnom vremenu obavijesti o trenutnom stanju rasporeda
- Optimizacija opterećenja nastavnika, lekcija u nizu, praznina, promjena zgrada
- Optimizacija opterećenje za nastavu i studente, zabrana ili dopuštanje praznina
- Raspoređivanje kolegija ravnomjerno tijekom tjedna
- Optimiziranje korištenja dvorana i opreme
- Vračanje neograničenog broja urađenih akcija i pregled povijesti
- Ispis majstor rasporeda i svih pojedinačnih odjednom
- Podešavanje veličine kartica
- Dijeljenje s osobljem, dodjeljivanje uloga vlasnika, tvorca i gledatelja
- Suradnja s kolegama, sinkronizacija promjene na različitim uređajima
- Objava na webu, dijeljenje na društvenim mrežama ili dodavanje na web stranici škole
- Prilagođavanje boja rasporeda
- Izvoz u Excel, lokalnu datoteku, CSV, sliku , HTML, XML i PDF
- Konfiguracija opcije aplikacije da odgovaraju specifičnim potrebama škole

Prijavom korisnika u sustav otvara se prikaz rasporeda, koji je u početku prazan. Stanica je logična za pregled. U rasporedu stupci predstavljaju dane od ponedjeljka do petka. Postoji mogućnost pregled po razredima, profesorima, dvoranama i kolegijima, u svakom od tih pregleda red u tablici predstavlja jednog od njih. Razredi u poljima imaju kartice kolegija, dok sve ostale tablice u poljima imaju kartice razreda. Kartice se razlikuju po boji i lako su pregledne, što je vidljivo na slici 7. Slika 8. prikazuje mogućnosti aplikacije nalaze se u ikoni aplikacije, u gornjem lijevom kutu, kako ne bi zauzimale prostor za prikazivanje rasporeda. Jasno su napisane i lako ih je koristiti bez uputa o korištenju. Unosi podataka su jasni, pregledni i lako prilagodljivi.



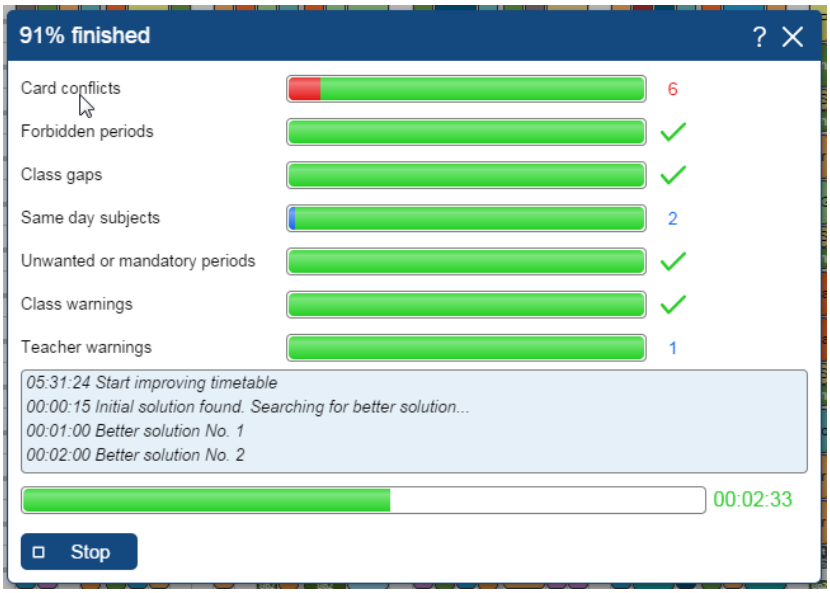
Slika 7. Prime Timetable - Raspored



Slika 8. Prime Timetable - Izbornik

Generiranje ima neodređeno vrijeme trajanja, ovisno o kompleksnosti zahtjeva. Aplikacija nakon 15 sekundi ponudi prvu verziju rasporeda i ispiše koliko zahtjeva ta verzija krši, strogi zahtjevi su ispisani crvenom bojom, a blagi plavom. Svakih sljedećih 60 sekundi aplikacija ponudi novu verziju rasporeda, dok ne zadovolji sve uvjete ili dok korisnik ne stopira taj rad jer je zadovoljan zahtjevima koji su do tad ispunjeni, što je

prikazano na slici 9. Generirani raspored korisnik može lako uređivati, premještajući kartice mišem. U slučaju da je određena kartica stavljena na polje u kojem krši određene zahtjeve, aplikacija to dopusti, ali zvono, koje se vidi na slici 10. u gornjem desnom kutu, prikaže obavijest koji su zahtjevi prekršeni.



Slika 9. Prime Timetable - Generiranje rasporeda



Slika 10. Prime Timetable - Obavijest o greškama

2.2.5 USPOREDBA APLIKACIJA PO ZAHTJEVIMA

Nakon obrade odabranih postojećih rješenja za problem rasporeda nastave izvršena je usporedba zahtjeva koje zadovoljavaju, što je prikazano u tablici 1. Iz dobivenih rezultata može se zaključiti da je Prime Timetable aplikacija zadovoljila najviše zahtjeva i jedina je dostupna u web pregledniku. Aplikacija Mimosa je zadovoljila minimalan broj zahtjeva, dok su aplikacije aSc Rasporedi i EMS zadovoljile podjednak broj zahtjeva.

APLIKACIJE / ZAHTJEVI	aSc Rasporedi	EMS classroom scheduling	Mimosa Scheduling Software	Prime Timetable
Zadovovoljavanje strogih zahtjeva	ima	ima	ima	ima
Uređivanje rasporeda koristeći miša	ima	ima	nema	ima
Rješavanje ograničenja i sukoba	ima	ima	ima	ima
Uvoz podataka	ima	ima	nema	ima
Postavljanje dana i vremena trajanja sata	nema	nema	nema	ima
Upravljanje kolegijima, dvoranama, profesorima, razredima, grupama	ima	ima	ima	ima
Upravljanje nastavom sa više od jednog profesora, razreda, grupe ili dvorane	nema	ima	nema	ima
Označavanje neželjene i obavezne pozicije	nema	nema	nema	ima
Savjeti i trikovi kako bi se dobio maksimum iz automatski generiranog rasporeda	nema	ima	nema	ima
Obavijest o trenutnom stanju rasporeda	ima	ima	ima	ima
Optimizacija opterećenja nastavnika, lekcija u nizu, praznina, promjena zgrada	ima	ima	nema	ima
Optimizacija opterećenje za nastavu i studente, zabrana ili dopuštanje praznina	nema	ima	nema	ima
Raspoređivanje kolegija ravnomjerno tijekom tjedna	ima	ima	nema	ima
Optimiziranje korištenja dvorana i opreme	ima	ima	nema	ima
Prikaz nastava koje mogu biti spojene	nema	ima	nema	nema
Zakazivanje ispita	nema	ima	nema	nema
Ispis rasporeda	ima	ima	ima	ima
Dijeljenje s osobljem	nema	nema	nema	ima
Objava na webu, dijeljenje na društvenim mrežama ili dodavanje na web stranici škole	nema	ima	ima	ima
Izvoz u Excel, lokalnu datoteku, CSV, sliku , HTML, XML i PDF	nema	nema	nema	ima
Web aplikacija/desktop aplikacija	desktop aplikacija	desktop aplikacija	desktop aplikacija	web aplikacija

Tablica 1. Usporedba aplikacija po zahtjevima

3. KORIŠTENE TEHNOLOGIJE

3.1 ODABIR TEHNOLOGIJE

Rasporede sati potrebno je riješiti na bilo koji način koji zadovoljava zahtjeve. Postoji više načina za rješavanje ovog problema, pa tako i više različitih tehnologija pomoću kojih se dolazi do programskog rješenja.

Prije odabira tehnologije potrebno je utvrditi slijedeće:

Kriterije koji utječu na odluku koja tehnologija će se koristiti:

- Platforma na kojoj se aplikacija koristi
- Oblik rješenja
- Obujam korisnika koji koriste aplikaciju
- Učestalost korištenja aplikacije
- Lokacija podataka

Svojstva aplikacije za raspored nastave:

- Aplikacija se rijetko koristi
- Aplikacijom je dovoljno da upravlja jedna osoba
- Gotove rasporede može pregledati bilo tko
- Poželjno je da bude neovisna o platformi
- Poželjno je da se rad izvršava na strani poslužitelja, a ne na uređaju na kojem se koristi
- Podaci se spremaju u bazu podataka

Uz ove činjenice izbor se suzio na javu ili kombinaciju PHP-a i HTML-a. Iz razloga dostupnosti aplikacije odabrana je kombinacija PHP-a i HTML-a, obzirom da ne postoji nikakva potreba za instalaciju aplikacije na uređaj, već joj se može pristupiti sa bilo kojeg uređaja i bilo koje lokacije putem web browsera. Za korištenje aplikacije potrebno je da korisnik bude online, te da ima pristup aplikaciji sa lozinkom. Kompletan proces izrade rješenja se obrađuje na poslužitelju, te korisnik nije ovisan o performansama uređaja na kojemu koristi aplikaciju. Kao gotovo rješenje, aplikacija pruža raspored nastave na web stanici. Obzirom na činjenicu da je u današnje doba pristup internetu na dohvat ruke, online raspored nastave na stranici škole ili fakulteta omogućuje svim

njenim polaznicima pristup i pregled u realnom vremenu bez potrebe za instaliranjem ikakve dodatne aplikacije.

3.2 PHP

PHP (*Personal Home Page*) je skriptni programski jezik namijenjen za web skriptiranje. Pomoću njega se kreira HTML stranica sa dinamičnim sadržajem na strani poslužitelja i šalje korisniku. Takav način stvaranja HTML datoteka osigurava da korisnik nema pristup PHP kodu i da stranica koju je primio ne postoji nigdje na poslužitelju. Jezik *je open-source*, svatko može skinuti izvorne PHP kodove i uređivat ih po želji. Sintaksom je sličan nekim drugim program jezicima, poput C-a ili Pearl-a. Iste radnje se mogu izvoditi korištenjem različitih funkcija. Funkcije PHP-a mogu manipulirati mnogim tipovima sadržaja, u što spadaju: grafika, učitavanje .NET, rad sa XML-om i drugi. Podržane su i baze podataka poput: MySQL, PostgreSQL, dBase, Oracle, ODBC i druge. Koristi se na Unix/Linux i Windows/Apache platformama. Kombinira se sa JavaScript-om, jQuery-em, CSS-om i sa HTML5. (php.com.hr, 2008)

Za pokretanje PHP datoteke potrebno je imati poslužitelja koji podržava PHP, datoteke sa nastavkom .php i kod unutar njih koji poštuje sintakse ovog programskog jezika. Kod se piše unutar tagova „<?php“ i „?>“, što omogućava konstantno preskakanje iz HTML-a u PHP unutar PHP datoteke. Naredbe završavaju sa znakom „;“. Podaci se spremaju u varijable koje započinju sa znakom „$“. Za razliku od mnogih drugih jezika PHP ne zahtjeva definiranje tipa varijable, već sam zaključuje o kojem se tipu radi (ako je podatak u navodnicima onda je string, ako je cjelobrojni broj je integer, ako je decimalni broja je float, itd.). Funkcije *require* i *include* ubacuju datoteku u sadržaj skripte koja poziva tu funkciju. Petlje *if* i *if/else* ne razlikuju se od tih petlja u drugim programskim jezicima. Ako je uvjet zadovoljen petlja se izvršava, inače se skripta nastavlja ili prelazi na *else* naredbu. Moguće je dopisati više uvjeta petlji sa veznicima &&(i) i ||(ili). *Switch* petlja provjerava više uvjeta odjednom, brža je, korisnija i jednostavnija od *if* petlji. U kombinaciji sa HTML-om mogu se ispunjavati formulari. HTML forma šalje upisane podatke PHP-u koje on može obrađivati. Obrađivanje podataka u PHP varijablama može se, osim pomoću funkcija, vršiti i pomoću operatora. Operatori su slični kao i kod ranije navedenih programskih jezika. Postoji nekoliko vrsta operatora: pridružujući (=), matematički (+,-,*,/,%), uspoređivački (==,

===, !=, <, >, <=, >=), znakovni („.“ za spajanje stringova), kombinirani matematički i pridružujući (+=, -=, *=, /=, %=. .=), povećanje(++) i smanjenje (--). *Array* predstavlja višedimenzionalno polje, tj. varijablu u koju se može spremiti više vrijednosti. Kako bi se tako nešto postiglo polja su indeksirana. *For* petlja se koristi kako bi se zadane naredbe ponavljale dok je uvjet zadovoljen. *For* uzima tri izraza odvojena sa „;“. U prvom izrazu se mogu stvoriti nove varijable, u drugom izrazu se postavlja uvjet koji mora biti poštovan da bi se petlja odvijala, dok treći izraz može sadržavati naredbe koje će se izvršiti na kraju svakog prolaza kroz petlju. (Robin, 2014)

PHP je odabran iz razloga što je kompatibilan sa svim platformama. Sličnost sa mnogim drugim programskim jezicima pokazuje logiku programiranja u njemu i moć koju je pokupio od tih jezika. Mogućnost spajanja sa HTML-om i MySQL-om koristi za izradu dinamične web stranice i za upravljanje velikom količinom podataka, što je za ovu aplikaciju vrlo potrebno.

3.3 MYSQL

MySQL je poslužitelj baza podataka. Pristupa mu se preko mreže na isti način na koji se pristupa i web poslužiteljima, uz pomoću korisničkog imena i lozinke koja je dobivena od poslužitelja. Poslužitelj može imati mnogo baza podataka, sa kojima se može raditi unutar jednog projekta. Svaki korisnički račun može imati drugačiju razinu administratorskih prava za određenu bazu podataka ili cijeli poslužitelj. Kućna upotreba je besplatna. Postoje verzije za sve veće operacijske sustave. Najčešće se koristi za izradu manjih ili srednjih web stranica. U kombinaciji sa PHP-om, naredbe se spremaju u varijablu koja se zatim poveže na bazu i sprema povratne informacije u varijablu u obliku liste kroz koju se prolaziti PHP petljama. (Welling & Thomson, 2008)

Jezik naredbi je sličan engleskom jeziku, zadavanje zvuči logično. Naredbe se upisuju u klijent. Stvaranje baze započinje jednom jednostavnom naredbom nad poslužiteljem. Piše se u obliku „mysql> CREATE DATABASE naziv_baze;“. Naredbe završavaju sa znakom „;“. Na isti način se stvaraju i nove tablice sa naredbom u obliku „CREATE TABLE naziv_tablice;“. Unos podataka u bazu se izvršava naredbom u obliku „INSERT INTO naziv_tablice (naziv_stupca1, naziv_stupca2) VALUES (1, „vrijednost2“);“. Za prikaz podataka iz tablice se koristi naredba *SELECT FROM*, koja uz neke druge naredbe pruža široku paletu različitih kombinacija podataka, koje se mogu pročitati iz

tablica baze. Naredba „SELECT * FROM naziv_tablice“ dohvaća sve podatke koji se nalaze u tablici. Znak „*“ označava da upit traži sve podatke iz tablice. Ako se želi dohvatiti podatke samo iz jednog ili nekoliko stupaca, tada se piše upit poput „SELECT naziv_stupca2 FROM naziv_tablice“. Na tu naredbu se mogu dopisati uvjeti poput „…WHERE naziv_stupca1 = 1“. Podaci se mogu sortirati ako se na tu naredbu dopiše upit u obliku „…ORDER BY naziv_stupca2“. Podaci iz tablice se mogu brisati i izmjenjivati. Naredba DELETE omogućuje brisanje podataka iz tablice. Naredba se piše u obliku „DELETE FROM naziv_tablice“, čime se brišu svi podaci iz zadane tablice. Ako se žele obrisati samo neki podaci moguće je na to dopisati uvjet sa naredbom WHERE. Uređivanje podataka se vrši sa naredbama u obliku „UPDATE naziv_tablice SET naziv_stupca2 = 'vrijednost3' WHERE naziv_stupca1 = 1;“ (php.com.hr, 2008)

Dostupnost, brzina i lakoća pisanja naredbi u MySQL su zaslužne za njeno korištenje u izradi programskog rješenja za generiranje rasporeda sati.

3.4 HTML5

HTML5 (*HyperText Markup Language*) je dorađena verzija HTML-a. To je jezik za strukturiranje i prezentiranje sadržaja na internet. Njime se oblikuje sadržaj na web stranicama. Jednostavan je i besplatan. Kako bi se mogli prikazivati hipertext dokumenti potrebno ga je otvoriti u jednom od web preglednika.

HTML elementi se pišu unutar znakova „<“ i „>“. Svaka HTML datoteka bi trebala započeti sa <html> i završiti sa </html>. Unutar tog elementa trebali bi biti <head> i <body>. U <head> se nalaze podaci koji se ne prikazuju na stranici, već ju opisuju. Jedan od elemenata koji se nalazi u <head> je <title> unutar kojeg je napisan naziv koji se ispisuje na kartici stranice u web pregledniku. <body> sadrži HTML kodove koji određuju sadržaj za prikazivanje na web stranici. To mogu biti tekst, slike, popisi, poveznice, tablice, strukturni elementi i elementi za video i audio zapise. Ulomci teksta se pišu unutar ili <p> elemenata. Naslovi ulomaka se pišu unutar elementa <h1> za glavni naslov ili unutar elemenata <h2> do <h6> za podnaslove. Na web stranicu se mogu ispisivati liste poredane po brojevima, za što služe element <ol> ili bez brojeva za što služi element <ul>. Tablice se pišu unutar <table> elementa, redovi unutar <tr> i podaci unutar <td>. Poveznice se pišu obliku „

Google“. Ono što je najvažnije za kombinaciju PHP-a i HTML-a su podaci koji se među njima izmjenjuju. To omogućavaju HTML forme. Forme se pišu unutar elementa <form>. Forme mogu imati method=“POST“ i method=“GET“. method=“POST“ predstavlja nevidljive podatke koje izmjenjuje sa PHP-om. method=“GET“ predstavlja podatke koji se ispisuju u adresi stranice. Akcija forme prikazuje lokaciju na koju forma vodi korisnika. Postoji nekoliko vrsta upisa u element, a pišu se u obliku „<input type="text">“. *Text* daje prostor za upis teksta, *number* daje prostor za upis broja, *radio* daje više mogućnosti od kojih je moguće odabrati samo jednu, *date* omogućava izbor datuma, password daje polje za upis teksta koji je sakriven, a *submit* daje tipku koja završava formu.

Jezik je kontinuirano u razvoju i dovodi nove mogućnosti. Niti jedan web preglednik za sad još ne prihvaća sve mogućnosti HTML-a, te je potrebno biti oprezan pri pisanju koda kako bi što veća količina preglednika mogla prikazati stranicu u njenom pravom obliku. HTML omogućava okupljanje dosad svih navedenih tehnologija i njihovo komponiranje kako bi se stvorile mnoge web aplikacije, pa tako i ova. (Robin, 2014)

3.5 CSS

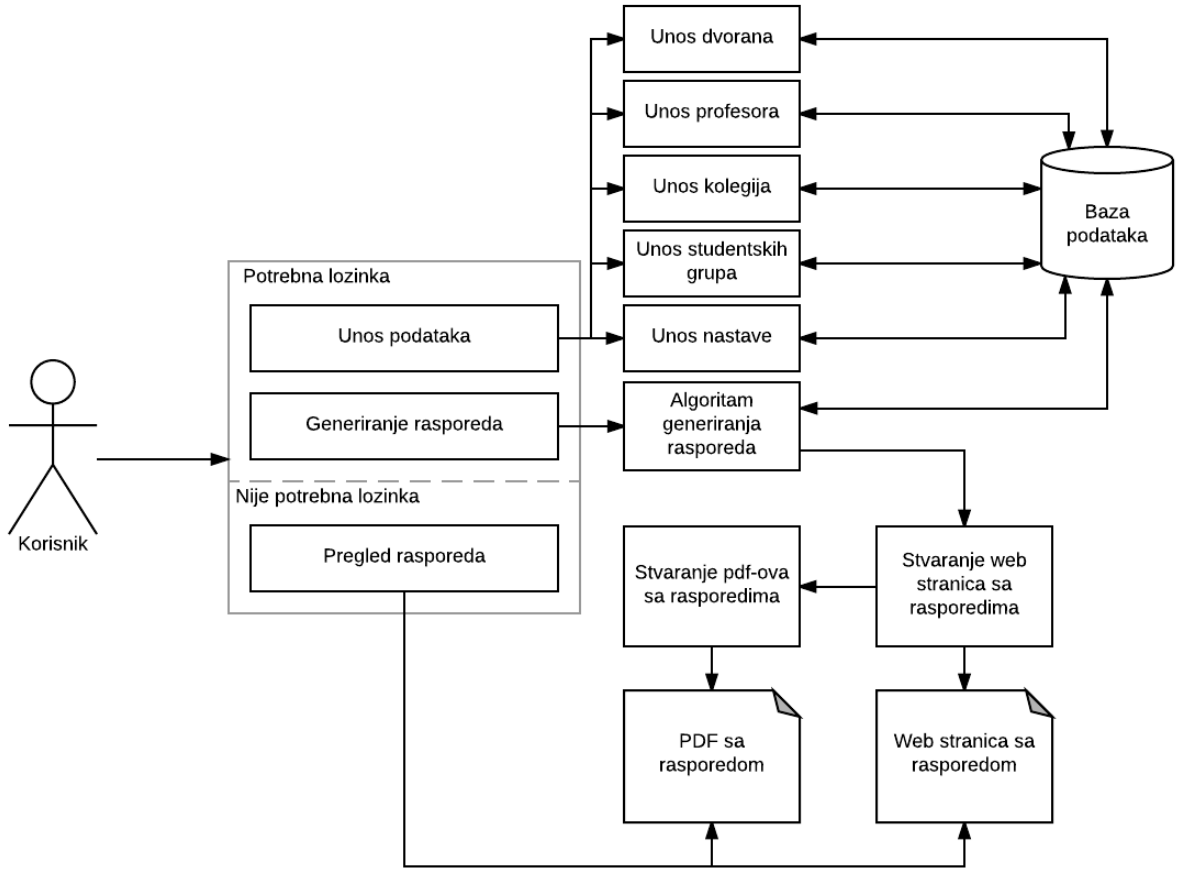
CSS (Cascading Style Sheets) je stilski jezik koji definira kako prikazati HTML elemente. Napravljen je sa svrhom uštede vremena i poboljšanja izgleda i funkcionalnosti web stranica. lako se može pisati unutar tagova HTML-a najčešće se piše u eksternim CSS datotekama. Eksterne datoteke sadržavaju klase ili id-ove sa stilom koji se može primijeniti u HTML-datotekama, koje se pozivaju na tu CSS datoteku. Razlika između klasa i id-ova je ta što se id primjenjuje samo prvi puta kad se na njega pozove u HTML datoteci, dok se klasa može primijeniti beskonačno mnogo puta. CSS datoteka može sadržavati i stilove koje se odnose na sve elemente HTML-a koji se nalaze pod zadanim elementima poput <body>, <a> i sl.. Stilovi za klase pišu u obliku: „.klasa1 { background: blue; }“; za id-ove u obliku „#id1 { background: yellow; }“ i za HTML elemente „body { background: red; }“. Stil može utjecati na pozadinu, boju i veličinu fonta, širinu visinu i poziciju ulomaka i drugo.

CSS je neizbježan jezik koji se koristi pri uređivanju web stranica, njegovo korištenje je neupitno kad se krene sa radom u HTML-u. (Lynch & Horton, 2009)

4. PROGRAMSKO RJEŠENJE

4.1 HODOGRAM KORIŠTENJA

Za lakše snalaženje korisnika, napravljen je hodogram korištenja, koji je prikazan na slici 11. Na njemu su prikazane mogućnosti koje korisnik ima u aplikaciji.



Slika 11. Hodogram korištenja

Korisnik može izabrati bilo koju od ponuđenih opcija. Nema potrebe da svaki put obavi sve što može, npr. korisnik može uči u aplikaciju i ne napravit ništa već samo pregledati rasporede. Iz tog razloga su prikazane tri mogućnosti koje korisnik može, ali ne mora odradit. Unos podataka i generiranje rasporeda su mogućnosti koje su zaštićene lozinkom, dok je pregled rasporeda dostupan svima. Unosom podataka o dvoranama,

profesorima, kolegijima i studentskim grupama podaci se spremaju u bazu podataka. Pomoću tih podataka se stvaraju podaci o nastavi koji se također spremaju u bazu. Kad su podaci već uneseni korisnik može odmah preći na generiranje rasporeda. Pokretanjem generiranja rasporeda, algoritam dohvaća podatke iz baze, obrađuje podatke, stvara raspored, vraća informacije u bazu i unosi ih u HTML verziju rasporeda, naknadno se HTML verzija pretvara u PDF. Kreirani rasporedi u HTML ili PDF verziji se mogu pregledati i bez potrebe za lozinkom kako bi svi učenici ili studenti uvijek imali pristup najnovijim verzijama rasporeda.

4.2 IZBORNIK APLIKACIJE

Grafičko sučelje za korištenje ove aplikacije napravljeno je u svrhu lakšeg unosa podataka u bazu, pokretanja algoritma i pregleda stvorenih baza podataka. Dok korisnik nije prijavljen ima samo opciju pregleda rasporeda i mogućnost prijave u sustav kao što je prikazano na slici 12., dok slika 13. prikazuje izbornik koji se tokom cijelog rada u aplikaciji nalazi na vrhu prozora. Mogućnosti koje korisnik može odabrati su Naslovna, Profesori, Kolegiji, Studenti, Dvorane, Nastava, Generiraj, Rasporedi i Odjava, tj. Prijava. Svaki od tih izbora vodi na svoju stranicu. Korisnik zna na kojoj se stranici trenutno nalazi po tome što je ona napisana podebljanim slovima. Ponovnim klikom na stranicu na kojoj se korisnik već nalazi stranica se osvježi. Pritiskom na Odjava, korisnik prelazi na naslovnu stranicu i nema mogućnost pregleda ostalih stranica osim rasporeda.



Slika 12. Neprijavljeni korisnik



Slika 13. Prijavljeni korisnik

4.3 NASLOVNA

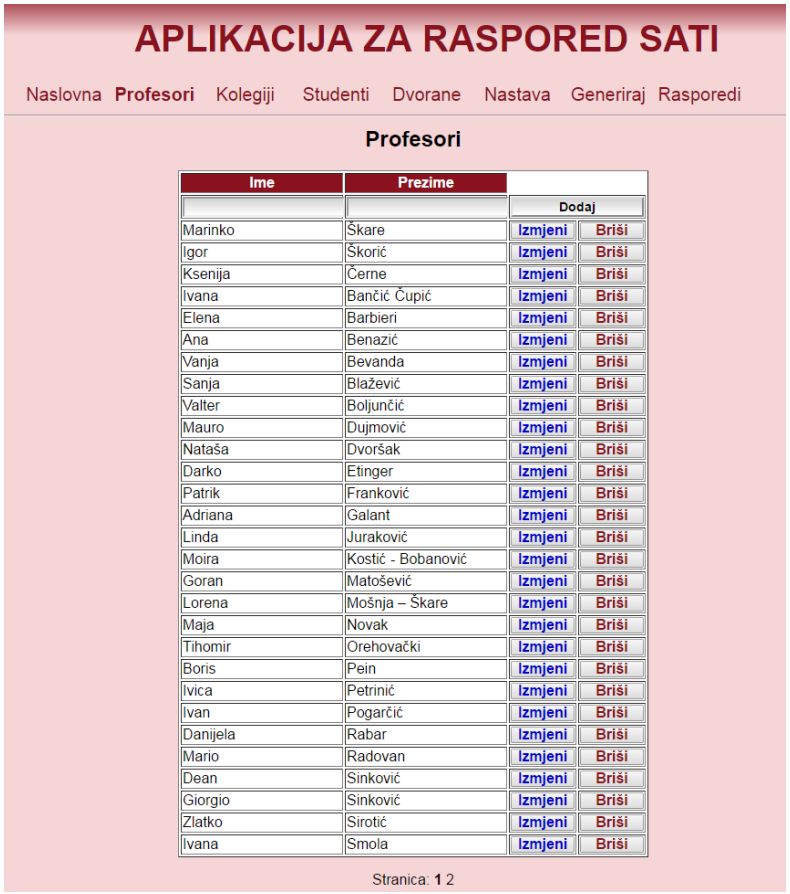
Naslovna stranica je prva stranica koja se otvori kad korisnik uđe u aplikaciju. Prikazuje kraći opis aplikacije i informacije o tome tko ju je izradio. Stranica sadržava i kratke upute za korištenje aplikacije napisane u nekoliko točaka. Nakon početka rada korisnik nema veću potrebu vraćati se na istu.

4.4 PROFESORI

Stranica Profesori prikazuje, kao na slici 14., tablicu unesenih profesora. U tablici se prikazuje do 30 podataka, poredanih abecedno po prezimenu, zatim po imenu. Za sve ostale podatke potrebno je prijeći na ostale stranice te tablice koje su ponuđene ispod same tablice. Podebljanim slovima je prikazana stranica na kojoj se korisnik nalazi. Prvi red tablice je prazan i pruža mogućnost unosa novog podatka. Nakon što se podaci ispune i pritisne tipka Dodaj, stranica se osvježi, podatak se doda u baznu tablicu i u prikaz, a pregled ostaje na stranici na kojoj je korisnik bio kad je unio podatke. Aplikacija radi po istom principu i za izmjenu podataka. Nakon pritiska na tipku Izmjeni omogućeno je uređivanje podatka o profesoru, te kad su podaci uređeni pritiskom na tipku Spremi, podaci se spremaju u baznu tablicu. Tipka Briši automatski izbriše taj podatak iz bazne tablice i iz pregleda.

4.5 KOLEGIJI

Stranica Kolegiji prikazuje, slično kao i tablica Profesori na slici 14., tablicu unesenih kolegija. Na tablici se prikazuje do 30 podataka, poredanih abecedno po kolegiju, za sve ostale podatke potrebno je prijeći na ostale stranice tablice koje su ponuđene ispod tablice. Podebljanim slovima je prikazana stranica na kojoj se korisnik trenutno nalazi. Prvi red tablice je prazan i pruža mogućnost unosa novog podatka. Nakon što se podaci ispune i pritisne tipka Dodaj, stranica se osvježi, podatak se doda u baznu tablicu i u prikaz, a pregled ostaje na stranici na kojoj je korisnik bio kad je unio podatke. Kao i kod profesora, u tablici kolegija postoje tipka Izmjeni i Briši koje rade na identičan način.



Slika 14. Tablica Profesori

4.6 STUDENTI

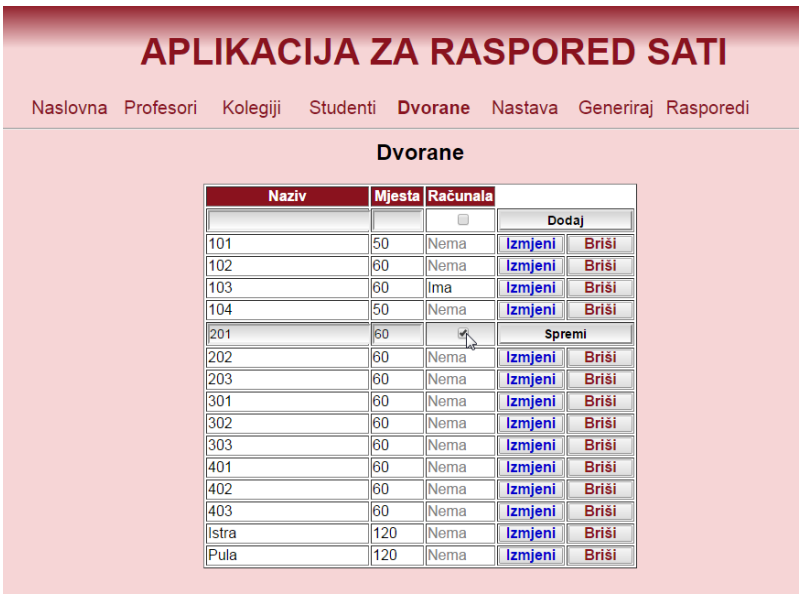
Stranica Studenti sastoji od dvije tablice koje ovise jedna o drugoj. To su tablica Smjerovi i tablica Studentske grupe. Tablica Smjerovi će biti rjeđe korištena i u njoj se nalaze podaci o smjerovima koji postoje na određenom sveučilištu. U njoj se ne očekuje previše podataka, te su uvijek svi prikazani neovisno o njihovoj količini. Smjerove je moguće dodavati, izmjenjivati i brisati. Tablica Studentske grupe ovisi o tablici Smjerovi. Kako bi se stvorio novi podatak o studentskoj grupi potrebno je da prvo postoji željeni smjer u tablici smjerova. Smjer se može odabrati u padajućem izborniku, kao na slici 15., koji je poredan abecednim redom, zatim se odabere godina studija, te se upisuje broj studenata za pojedinu godinu na određenom smjeru. Ukoliko se u tablici nalazi veći broj podataka isti se raspoređuju na veći broj stranica. koje je moguće pregledavati opcijom izbora stranice, koja je u tom slučaju ponuđena ispod tablice. U ovoj tablici se ne očekuje velika količina novih studentskih grupa, no očekuju se promjene broja studenata u studentskim grupama na godišnjoj bazi.



Slika 15. Tablice Studentske grupe i Smjerovi

4.7 DVORANE

Stranica Dvorane obuhvaća nekoliko podataka raspoređenih u tablicu. Tablica sadrži naziv dvorane, broj sjedećih mjesta i podatak ima li računala. Isto je opisano sa riječima „Ima“, odnosno sivim slovima „Nema“, kako je i prikazano na slici 16. Podrazumijeva se da je količina računala jednaka broju sjedećih mjesta u dvorani. U slučaju da količina podataka, odnosno broj redaka pređe 30, formiraju se nove stranice tablice. Unos podataka za stupce Naziv i Mjesta sličan je dosadašnjim, dok se kod stupca Računala može i ne mora stavit kvačica u ponuđeni prozorčić, ovisno o tome ima ili nema dvorana računala. Po istom principu kao i u prethodnim tablicama vrši se izmjena, odnosno brisanje podataka.



Slika 16. Tablica Dvorane

4.8 NASTAVA

U svim prethodno opisanim stranicama vrši se unos podataka. U stranici Nastava se svi dosad uneseni podaci objedinjavaju, osim podataka o dvoranama, koje služe za generiranje rasporeda. Tablica se sastoji od nekoliko stupaca koji su prikazani na slici 17., a to su: Semestar, Grupa, Kolegij, Tip, Sati/tjedan, Računala i Profesor. Kao i kod većine prijašnjih tablica prikazuje se do 30 podataka po stranici.



APLIKACIJA ZA RASPORED SATI

Naslovna Profesori Kolegiji Studenti Dvorane **Nastava** Generiraj Rasporedi

Nastava

Semestar	Grupa	Kolegij	Tip	Sati/tj	Računala	Profesor	Dodaj	
Zimski	Informatika 1	Engleski jezik I	Predavanje	2		Mauro Dujmović	Izmjeni	Briši
Zimski	Informatika 1	Engleski jezik I	Vježbe	1		Ivana Bančić Čupić	Izmjeni	Briši
Zimski	Informatika 1	Matematika za informatičare I	Predavanje	3		Valter Boljunčić	Izmjeni	Briši
Zimski	Informatika 1	Matematika za informatičare I	Vježbe	2		Danijela Rabar	Izmjeni	Briši
Zimski	Informatika 1	Osnove ekonomije	Predavanje	2		Marinko Škare	Izmjeni	Briši
Zimski	Informatika 1	Osnove ekonomije	Seminari	2		Giorgio Sinković	Izmjeni	Briši
Zimski	Informatika 1	Osnove IKT	Predavanje	2		Vanja Bevanda	Izmjeni	Briši
Zimski	Informatika 1	Osnove IKT	Vježbe	3		Marko Turk	Izmjeni	Briši
Zimski	Informatika 1	Programiranje	Predavanje	2	Potrebna	Tihomir Orehovački	Izmjeni	Briši
Zimski	Informatika 1	Programiranje	Vježbe	3	Potrebna	Boris Pein	Izmjeni	Briši
Zimski	Informatika 1	Uvod u metodologiju stručnog i znanstvenog rada	Predavanje	2		Sanja Blažević	Izmjeni	Briši
Zimski	Informatika 1	Uvod u metodologiju stručnog i znanstvenog rada	Seminari	1		Sanja Blažević	Izmjeni	Briši
Zimski	Informatika 2	Baze podataka II	Predavanje	2		Vanja Bevanda	Izmjeni	Briši
Zimski	Informatika 2	Baze podataka II	Seminari	2		Nataša Dvoršak	Izmjeni	Briši
Zimski	Informatika 2	Matematika za informatičare II	Predavanje	3		Valter Boljunčić	Izmjeni	Briši
Zimski	Informatika 2	Matematika za informatičare II	Seminari	2		Danijela Rabar	Izmjeni	Briši
Zimski	Informatika 2	Operacijski sustavi	Predavanje	3		Valter Boljunčić	Izmjeni	Briši
Zimski	Informatika 2	Operacijski sustavi	Seminari	2		Walter Stemberger	Izmjeni	Briši
Zimski	Informatika 2	Poslovno računovodstvo	Predavanje	2		Lorena Mošnja – Škare	Izmjeni	Briši
Zimski	Informatika 2	Poslovno računovodstvo	Vježbe	2		Robert Zenzerović	Izmjeni	Briši
Zimski	Informatika 2	Računalne mreže	Predavanje	3		Mario Radovan	Izmjeni	Briši
Zimski	Informatika 2	Računalne mreže	Seminari	2		Branko Velimirović	Izmjeni	Briši
Zimski	Informatika 3	Informacijska tehnologija i društvo	Predavanje	3		Mario Radovan	Izmjeni	Briši
Zimski	Informatika 3	Informacijska tehnologija i društvo	Seminari	2		Ivica Petrinić	Izmjeni	Briši
Zimski	Informatika 3	Operacijska istraživanja	Predavanje	3		Valter Boljunčić	Izmjeni	Briši
Zimski	Informatika 3	Operacijska istraživanja	Vježbe	2		Valter Boljunčić	Izmjeni	Briši
Zimski	Informatika 3	Poslovni informacijski sustavi	Predavanje	3		Giorgio Sinković	Izmjeni	Briši
Zimski	Informatika 3	Poslovni informacijski sustavi	Vježbe	2		Igor Škorić	Izmjeni	Briši
Zimski	Informatika 3	Softversko inženjerstvo	Predavanje	2		Tihomir Orehovački	Izmjeni	Briši

Stranica: **1** 2

Slika 17. Tablica Nastava

Stupac Semestar predstavlja semestar u kojem se nastava održava. Mogući izbori su zimski i ljetni semestar. Podaci su sortirani tako da se prvo prikazuje nastava u zimskom semestru, te zatim nastava u ljetnom semestru. Nakon toga, poštujući prethodno sortiranje, vrše se daljnja sortiranja unutar stupaca Grupa, Kolegij i Tip.

Stupac Grupa predstavlja studentsku grupu kojoj je namijenjena nastava. Pri unosu podataka, pritiskom na padajuću listu pojavljuju se sva polja koja se nalaze u tablici Studentske grupe na stranici Studenti.

Stupac Kolegij predstavlja kolegij koji grupa pohađa. Pri unosu podataka, pritiskom na padajuću listu pojavljuje se popis svih unesenih kolegija sa stanice Kolegiji, te korisnik svojevoljno bira jedan od ponuđenih.

Stupac Tip predstavlja tip nastave. Za kolegije koji imaju predavanja, vježbe i/ili seminare svaki tip nastava se zasebno unosi u tablicu. Predavanja, vježbe i seminari su jedini tipovi nastave koji trenutno postoje u aplikaciji, ne mogu se brisati ni dodavati novi pomoću aplikacije. Korisnik mora odabrati jedan od zadanih tipova sa padajuće liste.

Sati/tjedan je stupac u koji se, za razliku od ostalih stupaca koji imaju padajuću listu, ručno upisuje broj sati koji se mora održati u jednom tjednu za nastavu kolegija određenog tipa.

Stupac Računala, slično kao i u tablici Dvorane, prikazuje jesu li potrebna računala za održavanje nastave. Pri unosu tog podatka potrebno je u ponuđeni prozorčić staviti kvačicu ako su računala potrebna, nakon čega se ispisuje „Potrebna“, odnosno bez kvačice polje ostaje prazno.

Stupac Profesor prikazuje profesora određene nastave. Pri unosu podataka padajuća lista prikazuje popis svih profesora sa imenom i prezimenom poredanih abecednim redom po prezimenu.

Tipke Dodaj, Izmjeni, Spremi i Briši rade kao i kod svih dosadašnjih stranica. Pritiskom na jednu od tipki ostaje se na stranici tablice koja se trenutno koristi i aplikacija odradi zadanu funkciju. Aplikacija neće dodati ili spremiti podatke ako nisu potpuni ili korektni, tj. ako nedostaje neki od podataka ili ako su podaci redundantni i već postoji ta informacija u tablici. Za zabranu unosa određenih podataka nije potrebno da svi podaci budu identični već se ona pokreče ako postoji redak u kojem se poklapaju grupa, kolegij i tip. Npr. ako se razlikuje profesor u novom i starom podatku, nema smisla da dva različita profesora drže nastavu jednoj grupi, istog kolegija, istog tipa te je takav unos zabranjen. Brisanje kojeg od podataka u tablicama profesora, kolegija, smjerova ili studentskih grupa rezultira automatskim brisanjem redaka u tablici nastave koji su sadržavali obrisani podatak. Zato, ako npr. neki od profesora odlazi sa sveučilišta, savjetuje se da se za nastavu kojoj je on predavao najprije odredi drugi profesor, a tek

naknadno briše profesor koji odlazi, kako se ne bi obrisali svi redovi sa studentskim grupama i kolegijima kojima je on predavao.

4.9 GENERIRAJ

Stranica Generiraj sadrži dvije tipke:

- Generiraj raspored za zimski semestar
- Generiraj raspored za ljetni semestar

Na taj način se omogućava da se podaci za svaki semestar generiraju odvojene, te se ne gubiti vrijeme čekajući generiranje rasporeda koji se možda ne planira koristiti. Ispod tipki je obavijest da se nakon pritiska na jednu od tipki pričeka da aplikacija odradi generiranje rasporeda. Vrijeme potrebno za generiranje ovisi o broju podataka u tablici nastave. Nakon što je program završio zadane radnje ispiše obavijest o tome da je završio sa generiranjem, sa rasporedom podataka dvorana, studentskih grupa i profesora, te je li ubacivanje bilo uspješno, što se može vidjeti na slici 18. Ako ubacivanje podataka nije bilo uspješno, znači da nisu svi podaci ubačeni u rasporede, jer nije bilo moguće zadovoljiti jedan ili više zahtjeva. Preporuča se da se najprije pregledaju rasporedi po studentskim grupama, kako bi se pronašla nastava koja nije ubačena. Bez obzira jesu li rasporedi uspješno ili bezuspješno popunjeni pojavljuje se i obavijest da se rasporedi mogu pronaći u izborniku ili pod poveznicom Rasporedi.



Slika 18. Obavjest o stvaranju rasporeda

4.10 RASPOREDI

Stranica Rasporedi sadržava dvanaest poveznica, šest za zimski, šest za ljetni semestar, odnosno za svaki raspored dvije verzije: HTML i PDF. Pomoću zadanih lista na stranici moguće je pregledati rasporede za studentske grupe, profesore i dvorane. Stranica se ne nalaze pod lozinkom, jer se na nju direktno ne može utjecati, ni mijenjati podatke. Svrha rasporeda je da svi kojima je to potrebno imaju pristup putem službenih stranica sveučilišta.

Unosom novih podataka i generiranjem novih rasporeda, automatski se mijenjaju rasporedi i svi imaju pristup novim rasporedima. Na vrhu stranice rasporeda ispisan je podatak kad su rasporedi stvoreni kako bi se mogla procijeniti njihova relevantnost. Za svaki raspored postoji web stranica u HTML obliku i PDF datoteka. PDF datoteka je stvorena iz stranice rasporeda i gotovo su identične. Stvorena je pomoću mPDF[5] klase koja je besplatni softver preuzet sa njenih službenih stranica i implementiran u kod ove aplikacije. mPDF nije na niti jedan način izmjenjivan već su samo korištene njegove funkcije za potrebe stvaranja PDF datoteka rasporeda.

4.10.1 RASPORED GRUPA

Stranica rasporeda Grupe nema nikakvu funkcionalnost već služi samo za pregled rasporeda po studentskim grupama. Rasporedi su poredani abecednim redom i postoje samo za grupe koje se nalaze u tablici nastave. Nema potrebe imati pregled rasporeda za grupe koje nemaju u sebi nikakvu nastavu. U slučaju da postoji tablica određene studentske grupe koja je prazna, to znači da postoje podaci za tu grupu u tablici nastave, ali ne zadovoljavaju uvjete zadane u algoritmu, npr. u studentskoj grupi ima više studenata nego što ih ijedna dvorana može primiti. U tom slučaju se moraju napraviti određene izmjene podataka. Pregled rasporeda je napravljen tako da bi podaci bili što potpuniji, a raspored što pregledniji. Stupci predstavljaju dane od ponedjeljka do petka, dok redovi predstavljaju sate, od 1. do 15. Svaki od redaka koji predstavlja sat, unutar svakog dana ima još tri retka. Tri retka služe kako bi prikaz bio potpun, prvo polje (sive pozadina) predstavlja kolegij koji grupa ima u tom terminu, drugo polje (svjetlije sive pozadine) predstavlja profesora koji predaje taj kolegij i treće polje (bijele pozadine) predstavlja dvoranu u kojoj se nastava održava. Kako neki

[5] Službena stranica besplatnog softvera mPDF: http://www.mpdf1.com/

podaci zauzimaju veći prostor, zbog duljine svoga naziva, ispisan je početak koji se nastavlja sa „…“ radi estetike pregleda rasporeda. Da se ne bi gubila potpunost informacije, na podatke koji nisu stali u za to predviđen prostor moguće je kliknut mišem, te dok se tipka drži ispisan je potpuni podatak koji se nalazi u tom polju, primjer se može vidjeti na slici 19. U PDF datotekama su ispisani potpuni podaci kao što je prikazano na slici 20.

Raspored stvoren: 6. 9. 2016.

Grupe: Zimski semestar

Informatika 1

	Pon	Uto	Sri	Čet	Pet
1	Engleski jezik I: P Mauro Dujmović Istra	Osnove ekonomije: P Marinko Škare Istra	Uvod u metodologiju struč… Sanja Blažević Istra		
2	Engleski jezik I: P Mauro Dujmović Istra	Osnove ekonomije: P Marinko Škare Istra	Uvod u metodologiju struč… Sanja Blažević Istra		
3	Engleski jezik I: V Ivana Bančić Čupić Istra	Osnove ekonomije: S Giorgio Sinković Istra	Uvod u metodologiju struč… Sanja Blažević Istra		
4	Matematika za informatič… Valter Boljunčić Istra	Osnove ekonomije: S Giorgio Sinković Istra	Programiranje: P Tihomir Orehovački 103		
5	Matematika za informatič… Valter Boljunčić Istra	Osnove IKT: P Vanja Bevanda Istra	Programiranje: P Tihomir Orehovački 103		
6	Matematika za informatič… Valter Boljunčić Istra	Osnove IKT: P Vanja Bevanda Istra	Programiranje: V Boris Pein 103		
7	Matematika za informatič… Danijela Rabar Istra	Osnove IKT: V Marko Turk Istra	Programiranje: V Boris Pein 103		
8	Matematika za informatič… Danijela Rabar Istra	Osnove IKT: V Marko Turk Istra	Programiranje: V Boris Pein 103		
9		Osnove IKT: V Marko Turk Istra			
10					
11					
12					
13					
14					
15					

Slika 19. HTML prikaz rasporeda grupe

Grupe: Zimski semestar

Informatika 1

	Pon	Uto	Sri	Čet	Pet
1	Engleski jezik I: P	Osnove ekonomije: P	Uvod u metodologiju stručnog i znanstvenog rada: P		
	Mauro Dujmović	Marinko Škare	Sanja Blažević		
	Istra	Istra	Istra		
2	Engleski jezik I: P	Osnove ekonomije: P	Uvod u metodologiju stručnog i znanstvenog rada: P		
	Mauro Dujmović	Marinko Škare	Sanja Blažević		
	Istra	Istra	Istra		
3	Engleski jezik I: V	Osnove ekonomije: S	Uvod u metodologiju stručnog i znanstvenog rada: S		
	Ivana Bančić Čupić	Giorgio Sinković	Sanja Blažević		
	Istra	Istra	Istra		
4	Matematika za informatičare I: P	Osnove ekonomije: S	Programiranje: P		
	Valter Boljunčić	Giorgio Sinković	Tihomir Orehovački		
	Istra	Istra	103		
5	Matematika za informatičare I: P	Osnove IKT: P	Programiranje: P		
	Valter Boljunčić	Vanja Bevanda	Tihomir Orehovački		
	Istra	Istra	103		
6	Matematika za informatičare I: P	Osnove IKT: P	Programiranje: V		
	Valter Boljunčić	Vanja Bevanda	Boris Pein		
	Istra	Istra	103		
7	Matematika za informatičare I: V	Osnove IKT: V	Programiranje: V		
	Danijela Rabar	Marko Turk	Boris Pein		
	Istra	Istra	103		
8	Matematika za informatičare I: V	Osnove IKT: V	Programiranje: V		
	Danijela Rabar	Marko Turk	Boris Pein		
	Istra	Istra	103		
9		Osnove IKT: V			
		Marko Turk			
		Istra			
10					
11					
12					
13					
14					
15					

Slika 20. PDF prikaz rasporeda grupe

4.10.2 RASPORED PROFESORA

Stranica rasporeda Profesori nema nikakvu funkcionalnost već služi samo za pregled rasporeda po profesorima. Rasporedi su poredani abecednim redom prezimena profesora. Kao što vrijedi i za grupe tako vrijedi i za profesore, ako se ne nalaze u tablici nastave, njihov raspored se neće ispisati, a ako je ispisan, ali je prazan, znači da nastava koju je trebao držati ne zadovoljava određene uvjete zadane u algoritmu. Raspored se i u ovom slučaju sastoji od stupaca koji predstavljaju dane i redaka koji predstavljaju sate. U rasporedu profesora, prvo polje (sive pozadine) predstavlja studentsku grupu koja sluša nastavu, drugo polje (svijetlo sive boje) predstavlja kolegij koji profesor održava i treće polje (bijele pozadine) predstavlja dvoranu u kojoj se

nastava održava. Kao i u rasporedu grupa, klikom miša na polje koje nije bilo moguće u potpunosti ispisati prikažu se cjeloviti podaci.

4.10.3 RASPORED DVORANA

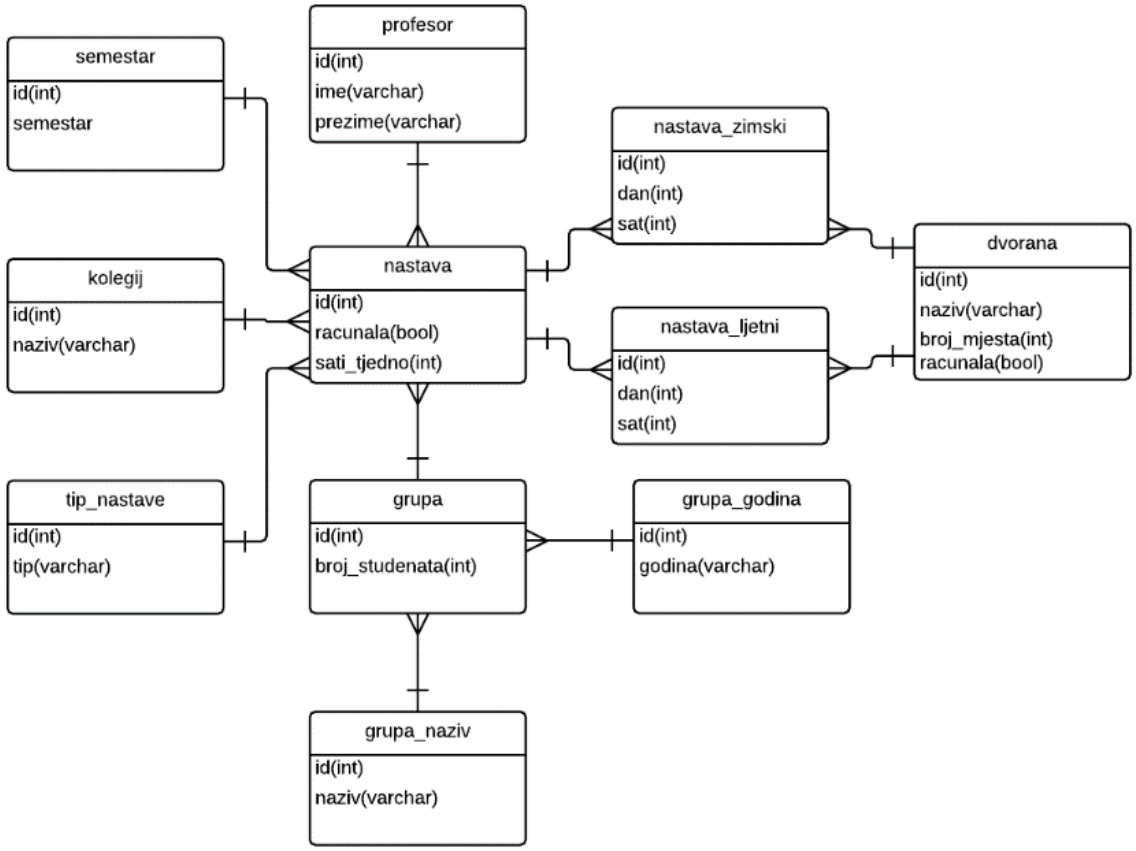
Stranica rasporeda Dvorane nema nikakvu funkcionalnost već služi samo za pregled rasporeda po dvoranama. Dvorane su poredane abecednim redom, te se ne prikazuju ako se nijedna nastava ne održava u njima. Prazne dvorane se nikad ne prikazuju. Raspored se sastoji od stupaca koji predstavljaju dane i redaka koji predstavljaju sate. Imaju tri polja za svaku kombinaciju dana i sata, gdje prvo polje (sive pozadine) predstavlja studentsku grupu koja u toj dvorani ima nastavu u tom terminu, drugo (svijetlo sivo polje) predstavlja kolegij koji se sluša i treće polje (bijele pozadine) predstavlja profesora te nastave. Klikom miša na polje koje nije bilo moguće u potpunosti ispisati prikažu se cjeloviti podaci.

5. BAZA PODATAKA

5.1 OPIS BAZE PODATAKA

Uspješno sortiranje nastave u rasporede i njihov prikaz omogućuje baza podataka. U bazi podataka koju koristi ova web aplikacija nalaze se tablice koje su ispunjene svim potrebnim podacima kako bi se mogli provjeriti zadani uvjeti pri generiranju rasporeda, te točno, pravovremeno i potpuno prikazati svi podaci koji opisuju nastavu sortiranu u rasporede.

U nastavku je prikazan detaljan opis tablica sa podacima koji se nalaze u njima, te svrha i namjena tih podataka. Baza se sastoji od 11 tablica. Tablice su međusobno povezane. Postoji sedam tablica roditelja i četiri djece, kao što se vidi na slici 21.



Slika 21. Baza podataka

5.2 TABLICA DVORANA

Tablica dvorana, kao i sve slijedeće tablice koje su opisane u nastavku imaju jedinstveni, automatski stvoreni id koji služi za rad sa podacima u bazi.

Atributi koje sadrži tablica dvorana su: naziv, broj_mjesta, racunala. Tip naziva je varchar, broj_mjesta je integer, te racunala boolean. Naziv predstavlja naziv dvorane, koji je većinom broj, no kako bi se osiguralo da prepozna i riječ, stupac je tipa varchar. Naziv mora biti jedinstven, te nije dopušteno unositi dvorane sa nazivom koji već postoje. Broj_mjesta je količina mjesta, tj. broj studenata koje dvorana može primiti, tipa integer, jer mora biti cjelobrojni broj. Svrha ovog stupca je da se zadovolje zahtjevi da svi studenti imaju svoje mjesto za vrijeme održavanja nastave. Racunala prikazuje ima li dvorana računala na kojima studenti mogu raditi, tipa boolean, jer postoje samo dvije mogućnosti, ima ili nema. Ima je prikazano sa „1“, a nema sa „0“. Svrha je da se studenti koji moraju za vrijeme nastave radit na računalima smjeste u takvu dvoranu.

5.3 TABLICA PROFESOR

Tablica profesor sadrži atribute ime i prezime. Kako su oba podatka riječi ti atributa su tipa varchar. Ime i prezime predstavljaju profesora koji održavati nastavu, dopušteno je imati više istih imena ili prezimena, ali nije dopušteno imati kopije istih kombinacija imena i prezimena. Ako se dva ili više profesora isto zovu i prezivaju potrebno je prilikom unosa napraviti razliku među njima.

5.4 TABLICA KOLEGIJ

Tablica kolegij se sastoji od samo jednog atributa, a to je naziv koji je tipa varchar. Naziv predstavlja naziv kolegija koji se održava na nastavi. Naziv je jedinstven i nisu dopuštene kopije.

5.5 TABLICA TIP_NASTAVE

Tablica tip_nastave se sastoji od atributa tip, koji je tipa varchar. Tip predstavlja tip nastave koji se održava. Postoje samo tri tipa, predavanje, vježbe i seminari, pa ne postoji potreba da korisnik dodaje, mijenja ili briše redove iz tablice.

5.6 TABLICA GRUPA_NAZIV

Tablica grupa_naziv se sastoji od atributa naziv, koji je tipa varchar. Tablica je detaljnije opisana u poglavlju 6.8.

5.7 TABLICA GRUPA_GODINA

Tablica grupa_godina se sastoji od atributa godina, koji je tipa integer. Tablica je detaljnije opisana u poglavlju 6.8.

5.8 TABLICA GRUPA

Tablica grupa se sastoji od atributa broj_studenata, koji je tipa integer. Kako bi se izbjegla redundancija podataka, tablica grupa je dijete tablicama grupa_naziv i grupa_godina, jer se u njima nalaze atributi naziv i godina. Naziv predstavlja smjer obrazovanja, a godina predstavlja godinu obrazovanja. Pri unosu je dopušteno dodavati, mijenjati i brisati smjerove, tj. dodavati naziv u tablici grupa_naziv, te ih naknadno kombinirati u tablici grupa sa godinama koje su fiksno unesene u grupa_godina, od godine 1 do godine 5. Kombinacija tih podataka mora biti jedinstvena i nije dopušteno raditi kopije već postojećih. Osim toga potrebno je u tablicu grupa upisati broj studenata koji pohađaju odabrani smjer na odabranoj godini.

5.9 TABLICA SEMESTAR

Tablica semestar se sastoji od atributa semestar, koji je tipa varchar. Semestar sadrži informaciju održava li se nastava u zimskom ili ljetnom semestru. Ne postoji potreba za izmjenom tih podataka, već su oni fiksni.

5.10 TABLICA NASTAVA

Tablica nastava se sastoji od atributa sati_tjedno, koji je tipa integer, i racunala, koji je tipa boolean, te je ona dijete tablica profesor, kolegij, tip_nastave, grupa i semestar. Kombinacijom vanjskih ključeva iz tablica roditelja u tablici nastave stvara se ciljana informacija koja se ubacuje u sam raspored nastave. Kako ne bi došlo do redundancije baza ne dopušta stvaranje novih redova koji imaju istu kombinaciju kolegija, tipa

nastave i studentske grupe. Sati_tjedno predstavljaju količinu odabrane nastave koja se mora održati unutar jednog tjedna, ta nastava je u raspored unesena u obliku blok sati. Racunala daju informaciju o tome jesu li studentima potrebna računala za rad na nastavi, broj „0“ predstavlja da nisu, a „1“ da su potrebna. Pomoću toga se provjerava može li se nastava održati u određenoj dvorani. Dopušteno je da se nastava održi u dvorani koja ima računala iako ona nisu potrebna, ali nije dopušteno da se nastava za koju su potrebna računala održava u dvorani koja ih nema. Pri ispisu rasporeda, pomoću vanjskih ključeva koji se nalaze u tablici nastava vuku se podaci iz tablica njenih roditelja.

5.11 TABLICE NASTAVA_ZIMSKI I NASTAVA_LJETNI

Tablice se sastoje od atributa dan i sat, koji su oboje tipa integer. Obije tablice imaju za roditelje nastava i dvorana. Dan predstavlja dan u tjednu koji je označen brojevima od 1 do 5. Gdje je 1 ponedjeljak, a 5 petak. Sat predstavlja sat u danu koji je označen brojem od 1 do 15. Gdje je 1 prvi sat jutarnje smjene, 7 je zadnji sat jutarnje smjene, 8 je prvi sat popodnevne smjene i 15 je zadnji sat popodnevne smjene.

Vanjskim ključem dohvaća se podatak koja se nastava održava u kojoj dvorani, za zadani dan i sat. Na početku algoritma brišu se svi podaci iz tablice za koji je semestar algoritam pokrenut, te se unose novi generirani podaci. Algoritam za svaki semestar izvršava se zasebno, kao i brisanje podataka iz tablice. Da je postojala samo jedna tablica bilo bi potrebno brisati podatke samo za određeni semestar iz te tablice, ali se id-ovi ne bi praznili, već bi se svakim generiranjem nastavljali id-ovi, što bi eskaliralo nakon nekoliko korištenja aplikacije.

6. FUNKCIONALNOSTI

6.1 UVOD U FUNKCIONALNOSTI

Vizualan izgled web aplikaciji daju HTML i CSS, a njenu svrhu i funkcionalnosti obavljaju programski jezici koji se nalaze iza toga. Ova aplikacija koristi PHP u svrhu obavljanja funkcionalnosti, koji upravlja podacima iz baze podataka korištenjem MySQL-a. U sljedećim poglavljima prikazani su programski kodovi koji su korišteni u tim programskim jezicima, te su detaljno objašnjeni kako bi se pobliže moglo vidjeti koje sve funkcije aplikacija obavlja.

6.2 PREGLED PODATAKA

Pregled podataka na stranicama u svakoj od tablica omogućavaju podaci koji su dohvaćeni iz baze podataka. MySQL upit se piše unutar PHP-a i sprema se u jednu varijablu. Varijabla se poziva upitom „query“ i pozvani podaci se spremaju unutar nove varijable. U slučaju da je došlo do greške pri pozivanju podataka „die“ daje obavijest o pogrešci. Iako „die“ najčešće ne pruži nikakvu pomoć korisniku, koristan je pri prijavljivanju greške administratorima. Primjer 1. prikazuje dohvaćanje podataka za tablicu profesora.

Primjer 1. Dohvaćanje podataka iz baze

```
$query = "SELECT * FROM profesor ORDER BY prezime, ime;";
$res_prof = $conn->query($query);
if (!$res_prof) die($conn->error);
$rows_prof = $res_prof->num_rows;
```

U primjeru se prvo upit sprema u varijablu $query. Upit je napisan tako da se dohvate svi podaci iz tablice profesor sortirani prvo po prezimenu, a zatim po imenu. Tada se pokreće upit, podaci se spremaju u varijablu $res_prof. Slijedi provjera je li upit uspješno izvršen i na kraju se u varijabli $rows_prof sprema broj redaka, tj. podataka, koji je upit dohvatio. U nastavku poglavlja je prikazano daljnje korištenje dohvaćenih podataka, a u primjeru 2. prikazan je način na koji je riješen problem količine podataka na stranici.

Primjer 2. Rješenje problema količine podataka

```
$poc=0;
$kraj=29;
$get="";
if(isset($_GET['pg'])) {
  $pg=$_GET['pg'];
  $poc=($pg-1)*30;
  $kraj=$pg*30-1;
  $get="&pg=$pg";
}
```

Korisnik pritiskom na broj stranice koju želi, sprema taj podatak u varijablu $_GET['pg'] koja se puni HTML GET formom. Kad korisnik prvi put dođe na web stranicu koja sadrži tablicu prikaza podataka, ta varijabla je prazna, te je potrebno postaviti vrijednosti relevantne za tu stranicu. Prikaz tada započinje sa 0, kako je prvo polje u listi indeksirano sa 0, koja je spremljena u varijablu $poc, i završava sa 29, što je spremljeno u varijablu $kraj, to je sveukupno 30 podataka. Varijabla $get se koristi kasnije u slučajevima dodavanja, uređivanja ili brisanja podataka iz tablice. Prilikom odabira stranice koju se želi pregledati pokreče se if petlja koja ovisno o odabranoj stranici novi početak sprema u $poc, npr. ako je odabrana druga stranica tablice, sprema broj 30, a novi kraj koji sprema u $kraj, za drugu stranicu je 59. $get ispuni sa string-om koji prikazuje na kojoj se stranici tablice nalazi, npr. „&pg=2“. Nakon što je odabrana stranica, kreće ispis podataka u tablicu koji je prikazan u primjeru 3.

Primjer 3. Ispis podataka u tablicu

```
for($i=$poc; $i<$kraj && $i<$rows_prof; ++$i) {
            $res_prof->data_seek($i);
            $row_prof = $res_prof->fetch_array(MYSQLI_ASSOC);
            ?>
                <tr>
                    <td><?= $row_prof['ime'] ?></td>
                    <td><?= $row_prof['prezime'] ?></td>
                    <td>
     <a href="profesori.php?izmjeni=<?= $row_prof['id'] ?>
     <?=$get ?>">Izmjeni</a>
                   </td>
                    <td>
     <a href="profesori.php?brisi=<?= $row_prof['id'] ?>
     <?=$get ?>">Briši</a>
                     </td>
                </tr>
<?php } ?>
```

U prikazanom primjeru pokreče se petlja koji sa vrti maksimalno 30 puta, do kraja liste podataka iz baze ili dok nije prikazano 30 podataka na stranici tablice. Prikaz se radi kombinacijom HTML-a i PHP-a. HTML svakim prolaskom kroz petlju stvara novi red u tablici, te nova polja. U ovom slučaju, u tablici profesora, stvaraju se četiri stupca, prvi stupac je popunjen imenom, a drugi prezimenom profesora koji su dohvaćeni iz baze podataka. Kako bi se lakše prepoznalo koji se podatak dohvaća naredba fetch_array(MYSQLI_ASSOC) omogućava pristup podatku pomoću naziva stupca iz baze. Posljednja dva stupca postoje i slična su u svim tablicama, jedan stupac sadrži tipke Izmjeni, a drugi Briši. Tipka Izmjeni je adresa koja ponovo pokreče stranicu, sa GET podatkom u adresi o retku u kojem je ta tipka stisnuta u varijabli $_GET['izmjeni'], te se na to nadoveže varijabla $get. U slučaju da stranica nije odabrana od strane korisnika varijabla $get je kako smo ranije vidjeli prazna i ne utječe na adresu, no ako je korisnik odabrao stranicu tablice varijabla $get sadrži informacije o stranici, te se pri izvršavanju izmjene, koja je pokrenuta pritiskom na tipku, korisnik i dalje nalazi na željenoj stranici. Pritiskom na tipku Briši dešava se slična radnja kao i kod izmjene. Podatak o retku u kojem je odabrano puni se u varijablu $_GET['brisi'] i pokreće se petlja brisanja.

6.3 DODAVANJE PODATAKA

Aplikacija ima mogućnost dodavanja novih podataka u bazu. Ta mogućnost se nalazi na vrhu tablice u obliku forme u prvom redu, koji je moguće popuniti i zatim te podatke dodati u bazu. Primjer 4. prikazuje na koji način je to odrađeno.

Primjer 4. Red za unos podataka

```
<form id="dodajF" action="profesori.php<?php if(isset($_GET['pg']))
      {$s_pg=$_GET['pg']; echo "?pg=$s_pg";} ?>" method="POST"></form>
<tr>
  <td><label><input form="dodajF" type="text"  name="dime"></label></td>
  <td><label><input form="dodajF" type="text" name="dprezime"></label></td>
  <td colspan="2"><input form="dodajF" type="submit" value="Dodaj""></td>
</tr>
```

U primjeru je prikazano dodavanje novog podatka za profesore. Stvaranjem nove forme omogućava se da se podaci ispunjeni u nju prenesu na petlju koja dodaje

podatke u bazu podataka. U „action“ dijelu forme, nakon što je forma aktivirana, u GET dio adrese ispisuje se stranica na kojoj je forma pokrenuta kako bi se na istoj mogao nastaviti rad nakon što se unese podatak. „Method“ forme je POST kako adresa ne bi bila prenapučena podacima. Otvara se redak u tablici, prva dva podatka služe za upis imena i prezimena u input forme tipa text. Posljednja dva stupca tablice su spojena u jedan i sadrže tipku Dodaj koja aktivira formu. Nakon što je tipka Dodaj pritisnuta stranica se ponovo pokreče. A time se pokreće petlja prikazana u primjeru 5.

Primjer 5. Dodavanje podataka u bazu

```
if(isset($_POST['dime']) && isset($_POST['dprezime'])){
    $ime = $_POST['dime'];
    $prezime = $_POST['dprezime'];
    if ($ime=="" || $prezime=="") {echo "Nisu uneseni svi potrebni
      podaci!";} else {
        $query = "INSERT INTO profesor (ime, prezime) VALUES ('$ime',
            '$prezime');";
        $res_insert = $conn->query($query);
        if (!$res_insert) echo "INSERT failed: $query" .
            $conn->error . "";
    }
}
```

Petlja se u ovom primjeru pokreće ako postoje podaci u varijablama $_POST['dime'] i $_POST['dprezime'] koje su ispunjene formom iz prošlog primjera. Te varijable se spremaju u $ime i $prezime. Provjerava se ako je koja od njih prazna, ako je program ispisuje obavijest o pogrešci da nisu svi potrebni podaci uneseni u formu. Ako nisu prazni, stvara se upit za bazu, u kojem je upisano da se želi unesti novi podatak u bazu sa zadanim imenom i prezimenom. Upit se tada šalje na bazu i u slučaju greške vraća se obavijest. Uspješno uneseni podaci odmah su vidljivi u tablici na mjestu predviđenom za njih.

6.4 IZMJENA PODATAKA

Pritiskom na tipku Izmjeni za određeni red ponovo se pokreće stranica sa podatkom, u $_POST['izmjeni'], o retku u kojem je to zadano. Tablica se ponovo puni istim podacima, ali kad dođe do retka u kojem je pritisnuto dugme više ne bude samo ispis podataka već forma popunjena tim podacima, što je postignuto kodom koji je napisan u primjeru 6.

Primjer 6. Red za izmjenu podataka

```
<form id="izmjeniF" action="profesori.php<?php if(isset($_GET['pg']))
      {$s_pg=$_GET['pg']; echo "?pg=$s_pg";} ?>" method="POST"></form>
<tr>
   <input form="izmjeniF" type="hidden" name="izmjeni"
      value=<?=$row_prof['id'] ?>>
   <td><label><input form="izmjeniF" type="text" name="iime"
      value="<?=$row_prof['ime'] ?>"></label></td>
   <td><label><input form="izmjeniF" type="text" name="iprezime"
      value="<?=$row_prof['prezime'] ?>"></label></td>
   <td colspan="2"><input form="izmjeniF" type="submit"
      value="Spremi"></td>
</tr>
```

Polja koja su ranije bila ispunjena tipkama Izmjeni i Briši sada su jedna tipka koja služi za spremanje promjene. Tipka je dio forme koja za podatke uzima ime i prezime koje se nalazi u formi, bilo da su to prvobitni podaci ili posve novi podaci koje korisnik unese. Osim toga stvori se i varijabla $_POST['izmjeni'] u kojoj je spremljen id polja nad kojim se vrše izmjene. Tipkom Spremi stranica se ponovo pokreće, a time i petlja koja je ispisana u primjeru 7.

Primjer 7. Izmjena podataka iz baze

```
if(isset($_POST['izmjeni'])){
    $id = $_POST['izmjeni'];
    $ime = $_POST['iime'];
    $prezime = $_POST['iprezime'];
    if ($ime=="" || $prezime=="") {echo „Nisu uneseni svi potrebni
      podaci!";} else {
        $query = "UPDATE profesor SET ime='$ime', prezime='$prezime' WHERE
            id=$id;";
        $res_insert = $conn->query($query);
        if (!$res_insert) echo "UPDATE failed: $query".$conn->error;
    }
}
```

Postojanjem varijable $_POST['izmjeni'] pokreće se petlja, te se u varijable $id, $ime i $prezime spremaju za to potrebni podaci. Ako ime ili prezime nisu upisani aplikacija daje obavijest da nisu uneseni svi potrebni podaci i petlja završava. Sa svim potrebnim podacima stvara se upit za bazu koji pokazuje da želi spremiti podatke $ime i $prezime u za to postojeća polja u bazi podataka, u tablici profesora gdje je id jednak $id. Upit se šalje u bazu i u slučaju greške aplikacija ispisuje obavijest, ukoliko nema greške podatak se sprema u bazu i petlja završava.

6.5 BRISANJE PODATAKA

Brisanje podataka je jednostavnije nego njihova izmjena. Pritiskom na tipku Briši adresa se ispunjava informacijom, tj. id-em podatka, u kojem je redu tipka pritisnuta, te se pokreće kod prikazan u primjeru 8.

Primjer 8. Brisanje podatka iz baze

```
if(isset($_GET['brisi'])){
    $brisi = $_GET['brisi'];
    $query = "DELETE FROM profesor WHERE id = $brisi";
    $res_insert = $conn->query($query);
    if (!$res_insert) echo "DELETE failed: $query".$conn->error;
}
```

Varijabla $_GET['brisi'] sadrži id polja u kojem je tipka stisnuta. Taj podatak se sprema u varijablu $brisi koja ispunjava upit koji se treba poslati bazi. Upit traži od baze da izbriše redak u tablici profesori gdje je id jednak $brisi. Pogreška u upitu izaziva da aplikacija ispiše upozorenje o pogrešci. Ako nema greške podatak više ne postoji u bazi, a time niti u tablici prikaza.

6.6 TABLICA NASTAVE

Funkcije svih stranica sa podacima su slične, te je njihov opis pojašnjen u prethodnim poglavljima.

U nastavku poglavlja prikazan je ispis i dodavanje podataka nastave, iz razloga što on objedinjuje sve ostale podatke u jednu tablicu. Izmjena je slična dodavanju samo što forma u sebi sadrži podatke koji su već bili upisani u tablici, dok brisanje radi na istom principu kao i kod drugih tablica.

Kako bi se tablica popunila potrebno je dohvatiti podatke ne samo iz tablice nastave već iz svih tablica koja su njeni roditelji, a u nekim slučajevima čak i podatke njihovih roditelja. Kod koji se nalazi u primjeru 9. prikazuje upit na bazu za dohvaćanje svih potrebnih podataka.

Primjer 9. Dohvaćanje podataka iz tablice nastave

```
$query = "SELECT nastava.id AS id,
                 grupa.id AS grupaID,
                 kolegij.id AS kolegijID,
                 profesor.id AS profID,
                 tip_nastave.id AS tipID,
                 semestar.id AS semestarID,
                 nastava.sati AS sati,
                 nastava.potrebna_racunala AS racunala,
                 semestar.semestar AS semestar,
                 profesor.ime AS ime,
                 profesor.prezime AS prezime,
                 kolegij.naziv AS kolegij,
                 tip_nastave.tip AS tip,
                 grupa.broj_studenata AS broj,
                 grupa_naziv.naziv AS smjer,
                 grupa_godina.godina AS godina
          FROM nastava
          JOIN semestar
            ON nastava.semestar_id=semestar.id
          JOIN profesor
            ON nastava.prof_id=profesor.id
          JOIN kolegij
            ON nastava.kolegij_id=kolegij.id
          JOIN tip_nastave
            ON nastava.tip_id=tip_nastave.id
          JOIN grupa
            ON nastava.grupa_id=grupa.id
          JOIN grupa_naziv
            ON grupa.naziv_id = grupa_naziv.id
          JOIN grupa_godina
            ON grupa.godina_id = grupa_godina.id
          ORDER BY semestarID, smjer, godina, kolegij, tipID;";
$res_nastava = $conn->query($query);
if (!$res_nastava) die($conn->error);
$rows_nastava = $res_nastava->num_rows;
```

Dohvaćanje podataka je u ovom slučaju kompliciranije, nije moguće koristiti * za dohvaćanje svih podatak iz tablice nastave jer se dohvaćaju i podaci iz tablica roditelja. Svakom stupcu iz svake tablice koji je pozvan je dato alternativno ime kako bi se lakše čitalo budući kod. Tablica nastave spojena sa drugim tablicama korištenjem JOIN-a pomoću vanjskih ključeva koji su sadržani u tablici nastave. Radi lakšeg snalaženja podaci su sortirani najprije po semestrima, zatim po smjeru, godini, kolegiju i na kraju tipu kolegija. Aplikacija ispisuje upozorenje u slučaju pogreške. Varijabli $rows_nastava je dana vrijednost o tome koliko je redova dohvaćeno. Veći dio tih podataka koji su dohvaćeni prikazani su u primjeru 10.

Primjer 10. Prikaz podataka u tablici nastave

```
for($j=$poc; $j<$kraj && $j<$rows_nastava; ++$j) {
    $res_nastava->data_seek($j);
    $row_nastava = $res_nastava->fetch_array(MYSQLI_ASSOC);
<tr>
    <td><?=$row_nastava['semestar'] ?></td>
    <td><?=$row_nastava['smjer']?> <?=$row_nastava['godina']?></td>
    <td><?=$row_nastava['kolegij'] ?></td>
    <td><?=$row_nastava['tip'] ?></td>
    <td style="text-align: center"><?= $row_nastava['sati'] ?></td>
    <td><?= $row_nastava['racunala'] ? "Potrebna" : "" ?></td>
    <td><?= $row_nastava['ime'] ?> <?= $row_nastava['prezime'] ?></td>
    <td><a href="nastava.php?izmjeni=<?= $row_nastava['id'] ?>&igrupa=<?=
$row_nastava['grupaID']      ?>&isemestar=<?= $row_nastava['semestarID']
?>&ikolegij=<?=   $row_nastava['kolegijID'] ?>&itip=<?=
$row_nastava['tipID']   ?>&iracunala=<?= $row_nastava['racunala']
?>&isati=<?=      $row_nastava['sati'] ?>&iprof=<?= $row_nastava['profID']
?><?=$get   ?>">Izmjeni</a></td>
     <td><a href="nastava.php?brisi=<?=  $row_nastava['id'] ?><?=$get
?>">Briši</a></td>
</tr>
<?php } ?>
```

Iz priloženog se vidi da je ispis podataka jednostavan pomoću alternativnih imena koja su data stupcima iz različitih tablica baze. Adresa stranice na koju vodi tipka Izmijeni sadrži podatke koji se nalaze u tom retku i koji će biti prikazani u formi izmjene.

U primjeru 11. obrađeno je dodavanje novog reda u tablicu nastave.

Primjer 11. Red za dodavanje novog podatka u tablicu nastave

```
<form id="dodajF" action="nastava.php<?php if(isset($_GET['pg']))
{$s_pg=$_GET['pg']; echo "?pg=$s_pg";} ?>" method="POST"></form>
<tr>
   <td>
     <label><select form="dodajF" name="dsemestar">
        <option disabled selected value></option>
        <?php for($i=0; $i<$rows_semestar; ++$i){
              $res_semestar -> data_seek($i);
              $row_semestar = $res_semestar->fetch_array(MYSQLI_ASSOC);?>
                  <option value="<?=$row_semestar['id'] ?>">
                        <?=$row_semestar['semestar'] ?>
                  </option>
            <?php } ?>
     </select></label>
  </td>
  <td>
    <label><select form="dodajF" name="dgrupa">
       <option disabled selected value></option>
       <?php for($i=0; $i<$rows_grupa; ++$i){
             $res_grupa -> data_seek($i);
             $row_grupa = $res_grupa->fetch_array(MYSQLI_ASSOC);?>
                  <option value="<?=$row_grupa['id'] ?>">
                        <?=$row_grupa['naziv'] ?> <?=$row_grupa['godina']?>
                  </option>
```

```
          <?php } ?>
      </select></label>
   </td>
   <td>
       <label><select form="dodajF" name="dkolegij">
            <option disabled selected value></option>
            <?php for($i=0; $i<$rows_kol; ++$i){
                $res_kol -> data_seek($i);
                $row_kol = $res_kol->fetch_array(MYSQLI_ASSOC);?>
                  <option value="<?=$row_kol['id']?>">
                        <?=$row_kol['naziv']?>
                  </option>
            <?php } ?>
        </select></label>
   </td>
   <td>
       <label><select form="dodajF" name="dtip">
             <option disabled selected value></option>
             <?php for($i=0; $i<$rows_tip; ++$i){
                 $res_tip -> data_seek($i);
                 $row_tip = $res_tip->fetch_array(MYSQLI_ASSOC);?>
                 <option value="<?=$row_tip['id'] ?>" >
                       <?=$row_tip['tip'] ?>
                  </option>
             <?php } ?>
         </select></label>
   </td>
   <td>
      <label><input form="dodajF" type="number" name="dsati" min="1"
      max="6" ></label>
   </td>
   <td>
      <label><input form="dodajF" type="checkbox"
      name="dracunala" ></label>
   </td>
   <td>
       <label><select form="dodajF" name="dprofesor">
           <option disabled selected value></option>
           <?php for($i=0; $i<$rows_prof; ++$i){
              $res_prof -> data_seek($i);
              $row_prof = $res_prof->fetch_array(MYSQLI_ASSOC);?>
                 <option value="<?=$row_prof['id'] ?>" >
                       <?=$row_prof['ime'] ?> <?=$row_prof['prezime'] ?>
                 </option>
           <?php } ?>
       </select></label>
   </td>
   <td colspan="2"><input form="dodajF" type="submit" value="Dodaj"></td>
</tr>
```

Za razliku od ostalih stranica gdje se podatak dohvaća iz jedne bazne tablice u ovom slučaju se podaci nalaze u više njih. Za odabir već postojećih podataka potrebno je za svako polje forme izvršiti petlju. Petlje unutar forme omogućavaju da se ispišu sve mogućnosti u padajućoj listi iz tražene tablice u bazi. Prije nego su ispisani svi podaci stvoreno je jedno prazno polje.

Prvi podatak koji korisnik mora odabrati je semestar. Po zadanom postoje samo dva semestra, zimski i ljetni. Sljedeći odabir je grupa. Grupa ne samo što je roditelj tablice nastave u bazi, veći sama ima dva roditelja, u jednom je sadržan naziv, a u drugom godina. Primjer 12. prikazuje dohvaćanje podataka za grupu. Kodu za dohvaćanje podataka grupe su slični svi ostali kodovi za dohvaćanje podataka za pripadajuće padajuće liste.

Primjer 12. Dohvaćanje podataka grupe

```
$query = "SELECT grupa.id AS id,
                 grupa.broj_studenata AS broj_studenata,
                 grupa_naziv.naziv AS naziv,
                 grupa_godina.godina AS godina
          FROM grupa
          JOIN grupa_naziv
            ON grupa.naziv_id = grupa_naziv.id
          JOIN grupa_godina
            ON grupa.godina_id = grupa_godina.id
          ORDER BY naziv, godina;";
$res_grupa = $conn->query($query);
if (!$res_grupa) die($conn->error);
$rows_grupa = $res_grupa->num_rows;
```

Iz priloženog se vidi da se kod za dohvaćanje podataka grupe ne razlikuje mnogo od kodova za dohvaćanje podataka tablica roditelja. Razlikuje se način ispisa u tablici nastave, gdje su u ovom slučaju smjer i godina, koji čine jednu grupu, ispisani jedno uz drugo u istom polju. Slijede Kolegij i Tip nastave za koje se podaci dohvaćaju iz baze. Nakon toga dolazi stupac Sati/tjedan. Taj stupac ne dohvaća nikakve podatke iz baze veće je potrebno samo upisati broj koji mora biti minimalno 1. Sljedeće polje su Računala za koje, ako su potrebna na nastavi, treba kliknut na prozorčić kako bi se u njega ispisala kvačica. Posljednji stupac je Profesor koji sadrži ime i prezime profesora iz tablice profesor u bazu. Nakon što je korisnik odabrao sve podatke koji želi potrebno je kliknuti na tipku Dodaj. Stranica se ponovo pokreće i izvršava se petlja iz primjera 13.

Primjer 13. Unos podataka u baznu tablicu nastave

```
if(isset($_POST['dgrupa']) &&
    isset($_POST['dsemestar']) &&
    isset($_POST['dkolegij']) &&
    isset($_POST['dtip']) &&
    isset($_POST['dprofesor']) &&
    isset($_POST['dsati'])){
    $grupa = $_POST['dgrupa'];
```

```
    $semestar = $_POST['dsemestar'];
    $kolegij = $_POST['dkolegij'];
    $tip = $_POST['dtip'];
    $racunala=0;
    if(isset($_POST['dracunala'])) if($_POST['dracunala']=="on")
      $racunala=1;
    $prof = $_POST['dprofesor'];
    $sati = $_POST['dsati'];
    if ($grupa=="" || $semestar=="" || $kolegij=="" || $tip=="" ||
      $prof=="" || $sati=="") {echo "<br><br>Nisu uneseni svi potrebni
      podaci!";
    } else {
      $query = "INSERT INTO nastava (grupa_id, semestar_id, kolegij_id,
      tip_id, prof_id, sati, potrebna_racunala)
      VALUES ($grupa, $semestar, $kolegij, $tip, $prof, $sati, $racunala)";
      $res_insert = $conn->query($query);
      if (!$res_insert) echo "INSERT failed: $query".$conn->error;
    }
}
```

Kao i kod ostalih dodavanja pregledava se jesu li uneseni svi podaci, ako nisu ispisuje se obavijest da podaci nisu potpuni, ako su podaci potpuni spremaju se u tablicu nastave u obliku vanjskih ključeva za tablice roditelja sa kojima je ova tablica povezana.

Nakon što su svi potrebni podaci ubačeni u tablicu nastave koja je osnova algoritma za generiranje možemo završiti sa tablicama podataka. U primjerima kodova koji su ispisani u proteklih nekoliko poglavlja nisu prikazani potpuni kodovi koji se nalaze u aplikaciji, već su adaptirani kako bi bili pregledniji u ovom dokumentu. Stvaran kod pored prikazanih funkcionalnih naredbi sadrži i dodatne koje služe estetskom prikazu tablica.

6.7 GENERIRAJ

Na stranici Generiraj se nalaze pozivi funkcija, koje ispunjavaju raspored i ispisuju obavijest da je generiranje završeno uz poveznicu na stranicu Rasporedi. Funkcija za ispunjavanje rasporeda zove se „raspored“. Kao argument funkcije dan je broj „1“ koji predstavlja zimski semestar, tj. broj „2“ koji predstavlja ljetni semestar.

6.8 FUNKCIJA RASPORED

Funkcija raspored služi za ispunjavanje rasporeda, no ona sama ga ne generira. Raspored generira funkcija pod nazivom „generiraj“, koja također ima semestar za argument. Osim te funkcije, u funkciji raspored se pojavljuje još nekoliko funkcija koje služe za popunjavanje polja rasporeda podacima o profesoru, studentskoj grupi, kolegiju ili dvorani. Sve te funkcije su objašnjene u slijedećih nekoliko potpoglavlja.

Funkcija raspored započinje sa dohvaćanjem podataka o dvoranama iz bazne tablice dvorana, te vanjskih ključeva dvorana koje se nalaze u tablicama nastava_zimski i nastava_ljetni. Tablice nastava_zimski i nastava_ljetni postoje iz razloga kad se ispisuju rasporedi po dvoranama da se ne ispisuju rasporedi za sve dvorane nego samo za one koje se nalaze u tim tablicama. Te dvije tablice sadržavaju vanjske ključeve nastave i dvorane, te vrijeme održavanja nastave, tj. dan i sat. Funkcija dohvaća podatke iz tablice nastave, te zasebno podatke o studentskoj grupi i profesorima koji se nalaze u tablici nastave. Takvo dohvaćanje podatka koji se nalaze u tablici nastave, služi da se ne ispisuju rasporedi za sve grupe, tj. profesore, već samo za one koji slušaju, tj. predaju nastavu. U varijablu $raspored se sprema višedimenzionalno polje koje je vraćeno pozivanjem funkcije generiraj. O funkciji generiraj piše detaljnije u nastavku poglavlja, dovoljno je znati da višedimenzionalno polje koje ona vraća sadrži sve podatke o tome kada se održava koja nastava, u kojoj dvorani. Ovisno o tome koji je semestar započinje se spremati HTML kod u varijable. Iako se stvaraju datoteke za dvorane, profesore i grupe sve su izvedene na isti način, za primjer je prikazano samo stvaranje rasporeda dvorana. Stvaranje započinje upisivanjem otvaranja HTML koda, postavljanjem head-a, otvaranjem body-a i pisanjem naslova u varijablu $text_dvorane, tj. $text_grupe, tj. $text_profesori, kako je prikazano u primjeru 14.

Primjer 14. Početak stvaranja stranice rasporeda

```
if($semestar==1) $text_dvorane = <<<_ISP
        <html>
        <head>
            Raspored stvoren: $datum
            <title>Dvorane: Zimski</title>
            <meta charset="utf-8">
            <link href="style_raspored.css" rel="stylesheet"
                  type="text/css">
        </head>
        <body>
        <h1>Dvorane: Zimski semestar</h1>
_ISP;
```

Kod dvorana zatim popunjava varijable $rows_dvorana i $res_dvorana sa podacima o dvoranama koje su ranije dohvaćene iz tablica nastava_zimski ili nastava_ljetni. Pokreće se petlja, koja se izvršava onoliko puta koliko ima dvorana u $rows_dvorana. Petlja započinje sa ispisivanjem naziva dvorane, za koju slijedi raspored, te otvaranjem tablice koja služi kao raspored i ispisivanjem naziva dana u prvi red tablice, što se sprema u varijablu $ispis_dvorane, kao što je prikazano u primjeru 15.

Primjer 15. Početak stvaranja tablice rasporeda

```
for($dv=0, $table=1; $dv<$rows_dvorana; ++$dv, ++$table) {
        $res_dvorana->data_seek($dv);
        $row_dvorana = $res_dvorana->fetch_array(MYSQLI_ASSOC);
        $tablica=$row_dvorana['naziv'];
        $ispis_dvorane = <<<_ISP
        <div class="stranica">
        <span class="naslov_t">$tablica</span>
        <table border="1">
            <tr  class="dan">
                <td></td>
                <td class="ras_dan">Pon</td>
                <td class="ras_dan">Uto</td>
                <td class="ras_dan">Sri</td>
                <td class="ras_dan">Čet</td>
                <td class="ras_dan">Pet</td>
            </tr>
_ISP;
```

Prvi redak sadrži naziv dana, a nakon njega se izvršava petlja koja se vrti 15 puta, jednom za svaki školski sat u danu. Prvi stupac prikazuje sat u danu, a sljedećih 5 se prvo provjerava postoji li podatak o njima u varijabli $raspored, u koju je ranije spremljeno višedimenzionalno polje. Ako je za kombinaciju dvorane, dana i sata, podatak jednak -1 to znači da se tada ne ispisuje ništa i polje ostaje prazno. U slučaju da se u toj kombinaciji nalazi id nastave tada se u tri polja, koja se nalaze za svaku kombinaciju dana i sata, ispisuju određeni podaci. Za dvorane ta tri polja sadržavaju grupu, kolegij i profesora, za grupu sadržavaju kolegij profesora i dvoranu, te za profesora sadržavaju grupu, kolegij i dvoranu. Kako je podatak koji se nalazi u toj kombinaciji samo id nastave potrebno je dohvatiti točno tu informaciju o grupi, kolegiju, tipu kolegija, profesoru ili u drugim rasporedima dvoranama, pomoću funkcija koje vraćaju dohvaćene podatke iz baze. Funkcije dv_gru, dv_kol, dv_tip, dv_prof i dv_dvo kao argument uzimaju id nastave koji se nalazi u $raspored te vraćaju željenu

informaciju u višedimenzionalnom polju koje se sprema u varijablu za određeni dan, kako je prikazano u primjeru 16.

Primjer 16. Ispis podataka u raspored

```
if($raspored['grupa'][$row_grupa['grupaID']][0][$sat]['zauzeta'] != -1)
$pon=dv_kol($raspored['grupa'][$row_grupa['grupaID']][0][$sat]['zauzeta']).
": ".dv_tip($raspored['grupa'][$row_grupa['grupaID']][0][$sat]['zauzeta']);
else $pon="";
if($raspored['grupa'][$row_grupa['grupaID']][1][$sat]['zauzeta'] != -1)
$uto=dv_kol($raspored['grupa'][$row_grupa['grupaID']][1][$sat]['zauzeta']).
": ".dv_tip($raspored['grupa'][$row_grupa['grupaID']][1][$sat]['zauzeta']);
else $uto="";
if($raspored['grupa'][$row_grupa['grupaID']][2][$sat]['zauzeta'] != -1)
$sri=dv_kol($raspored['grupa'][$row_grupa['grupaID']][2][$sat]['zauzeta']).
": ".dv_tip($raspored['grupa'][$row_grupa['grupaID']][2][$sat]['zauzeta']);
else $sri="";
if($raspored['grupa'][$row_grupa['grupaID']][3][$sat]['zauzeta'] != -1)
$cet=dv_kol($raspored['grupa'][$row_grupa['grupaID']][3][$sat]['zauzeta']).
": ".dv_tip($raspored['grupa'][$row_grupa['grupaID']][3][$sat]['zauzeta']);
else $cet="";
if($raspored['grupa'][$row_grupa['grupaID']][4][$sat]['zauzeta'] != -1)
$pet=dv_kol($raspored['grupa'][$row_grupa['grupaID']][4][$sat]['zauzeta']).
": ".dv_tip($raspored['grupa'][$row_grupa['grupaID']][4][$sat]['zauzeta']);
else $pet="";

            $ispis_grupe .= <<<_ISP
                    <tr class="ras_row">
                        <td rowspan="3">$tsat</td>
                        <td class="ras_col1">$pon</td>
                        <td class="ras_col1">$uto</td>
                        <td class="ras_col1">$sri</td>
                        <td class="ras_col1">$cet</td>
                        <td class="ras_col1">$pet</td>
                    </tr>
_ISP;
```

Završetkom petlja koje spremaju pojedini raspored u HTML kodu u za to zadane varijable $ispis_dvorane, tj. $ispis_grupe, tj. $ispis_profesora, te varijable se ispisuju u glavnu varijablu $ text_dvorane, tj. $text_grupe, tj. $text_profesori, koja je ranije započeta. Ta glavna varijabla zatim zatvara body i HTML. Otvara se nova ili se piše preko već postojeće datoteke, tipa HTML, za dvorane, grupe ili profesore za odabrani semestar. Te HTML datoteke se pretvaraju pomoću već spomenutih klasa mPDF besplatnog softvera u za to zadane PDF datoteke, kako je prikazano u primjeru 17. Time završava funkcija raspored, ona nema povratne podatke već nakon svake generirane stranice ispiše obavijest da je raspored uspješno popunjen ili da neke podatke nije bilo moguće ubaciti u raspored.

Primjer 17. Stvaranje PDF-a rasporeda

```
if($semestar==1) $fh = fopen("grupe_zimski.html",'w') or die("Failed1");
    if($semestar==2) $fh = fopen("grupe_ljetni.html",'w') or
      die("Failed1");
    fwrite($fh, $text_grupe) or die("Failed2");
    fclose($fh);
    include('mpdf.php');
    $mpdfd=new mPDF();
    $fhd = file_get_contents("dvorane_zimski.html") or die("File
      dvorane_zimski doesn not exist.");
    $mpdfd->WriteHTML($fhd);
    $mpdfd->Output('dvorane_zimski.pdf', 'F');
```

6.9 FUNKCIJA GENERIRAJ

Svrha funkcije generiraj je već prikazana sa korištenjem njenih povratih podataka u funkciji raspored. Da bi se bolje shvatilo kako ona radi isto je u nastavku detaljnije prikazano. Funkcija je ispunjena algoritmom koji popunjava višedimenzionalno polje $raspored koje sadrži određenu količinu podataka. Osim tog višedimenzionalnog polja u funkciji također postoji i višedimenzionalno polje koje za prvi indeks, tj. prvu dimenziju, ima dvoranu, grupu ili profesora, te polje koje služi za nastavu. Funkcija započinje sa dohvaćanjem podatak iz baze, ovog puta iz tablice nastave, tablice grupa koje se nalaze u tablici nastave, tablice profesora koji se nalaze u tablici nastave i iz tablice dvorana. Funkcija u potpunosti na početku čisti tablice nastava_zimski i nastava_ljetni od svih podataka koje su one do tad sadržavale. To čišćenje se obavlja zato što su tablice nastava_zimski i nastava_ljetni dinamične, što znači da ovise o funkcij, dok su ostali podaci u bazi fiksni, ovise o korisniku i ostaju onakvi kakvi su bili pri pozivanju funkcije. Sa dohvaćenim podacima se stvaraju višedimenzionalna polja.

Za višedimenzionalno polje nastave prvi indeks je broj koji može biti od 0 do broja koliko ima redova nastave, drugi indeks je riječ id, blok, racunala, mjesta, profesor ili grupa. Indeks id sadrži id nastave. Indeks blok sadrži broj sati koji se mora odraditi u jednom tjednu i koji će biti ubačen u raspored u obliku blok sati. Indeks racunala sadrži podatak o tome jesu li potrebna računala na nastavi. Indeks mjesta sadrži broj mjesta koje studenti popunjavaju u dvorani. Indeks profesor sadrži id profesora koji predaje nastavu. Indeks grupa sadrži id grupe koja pohađa nastavu.

Višedimenzionalno polje za dvorane za drugi indeks uzima id dvorane. Za svaki od njih se stvara novi indeks, koji sadrži dan, zatim novi indeks koji sadrži sat, te svaki od njih riječ zauzeto, racunala i mjesta. Polje koje ima indeks zauzeto je ispunjeno brojem

-1 koje označava da ta dvorana još nije zauzeta. Polje koje ima indeks racunala sadržava podatak ima li ta dvorana računala. Polje koje ima indeks mjesta sadrži podatak o tome koliko mjesta ima dvorana.

Višedimenzionalno polje grupe sadrži za drugi indeks id grupe, za treći dan, za četvrti ili sat ili riječ granica, indeks koji sadrži sat ima još jedan indeks u kojem piše zauzeta. Indeks zauzeta ispunjen je podatkom o tome je li grupa toga dana u tom satu zauzeta i svi podaci su na početku -1 što znači da je grupa slobodna. Indeks granica služi kao brojač nastave koju je grupa slušala u tom danu.

Višedimenzionalno polje profesora sadrži za drugi indeks id profesora, zatim dan, pa sat i na kraju riječ zauzet, što sadrži informaciju je li profesor zauzet u tom terminu. Za početak su sva polja zauzet, jednaka -1, što označava da je slobodan.

Stvaranjem svih tih višedimenzionalnih polja započinje algoritam koji stvara novo višedimenzionalno polje koje je povratni podatak te funkcije. Prije nego se to polje počne ispunjavat potrebno je provjerit zadovoljavaju li dohvaćeni podaci zahtjeve kako bi mogli biti ubačeni u raspored. Kako bi se došlo do svih tih podataka algoritam prolazi kroz nekoliko ukomponiranih petlji koje su prikazane u primjeru 18.

Primjer 18. Petlje algoritma

```
for ($nas = 0; $nas < $rows_nastava; ++$nas) {
    if(isset($ar_nas[$nas])){
       for ($dvo = 0; $dvo < $rows_dvorana; ++$dvo) {
          $row_dvorana = $res_dvorana->fetch_array(MYSQLI_ASSOC);
          for ($dan = 0; $dan < 5; ++$dan) {
              $bs=15-$ar_nas[$nas]['blok'];
              for ($sat = 0; $sat <= $bs; ++$sat) {
                  $slobodno=0;
                  $grupa_slobodna=0;
                  $prof_slobodan=0;
                  $mjesta=0;
                  $racunala=0;
                  $granica=0;
                  for ($blok = 0; $blok < $ar_nas[$nas]['blok']; ++$blok){…
```

Prva for petlja ide kroz sve redove nastave, jedan po jedan, redoslijedom prvo ide sva nastava kojoj su potrebna računala, zatim nastava sa najviše studenata, abecedno po smjeru, rastuće po godini, abecedno po kolegiju, predavanja, vježbe i seminari.

Provjerava se postoji li za nastavu pod tim brojem višedimenzionalno polje, koje je moguće kasnije u funkciji izbrisano ako je nastava ubačena u raspored. Potvrdom da

postoji, kreće se u novu petlju koja prolazi kroz sve dvorane, redoslijedom da prvo idu dvorane bez računala, te zatim abecedno po nazivu.

Takav redoslijed dvorana i naziva je napravljen sa svrhom da raspored po dvoranama bude što optimalniji za održavanje nastave. Kako bi se osiguralo da grupe koje ne trebaju dvorane sa računalima ne zauzmu iste, prvo se raspoređuju sve grupe kojima su računala potrebna. Da bi se osiguralo da male grupe ne zauzmu velike dvorane, kreće se od većih grupa. Smjer i godina su radi estetike poredani tim redom. Poželjno je da se prvo održe predavanja, a tek nakon toga vježbe ili seminari za pojedini kolegij. Redoslijed kojim su sortirane dvorane je takav da prvo idu dvorane bez računala kako bi se u njih smjestile grupe kojima računala nisu potrebna. Zatim se započinje od većih dvorana iz razloga udobnosti dvorane. Na kraju je abecedno sortirano, iz razloga estetike pregleda rasporeda. Petlja koja slijedi izvršava se 5 puta, jednom za svaki radni dan u tjednu. Tada se stvara varijabla $bs koja provjerava da blok sat može najkasnije završit na 15. satu nastave.

Slijedeća petlja za svaki dan ide sat po sat, koji je spremljen u varijabli $bs. Tu su stvorene i varijable $slobodno, $grupa_slobodna, $prof_slobodan, $mjesta, $racunala, $granica. U te varijable je spremljen broj 0 koji označava da ne zadovoljavaju zahtjeve. Kada varijabla zadovolji zahtjev u nju se sprema broj 1 kao što je prikazano u primjeru 19. čiji se kod nalazi u posljednjoj petlji.

Primjer 19. Provjera zahtjeva

```
if ($ras['dvorana'][$row_dvorana['id']][$dan][$sat+$blok]['zauzeto']== -1){
      $slobodno=1;} else {$slobodno=0; break 1;
}
if ($ras['grupa'][$ar_nas[$nas]['grupa']][$dan][$sat+$blok]['zauzeta']==-1)
      {$grupa_slobodna=1;} else {$grupa_slobodna=0; break 1;
}
if ($ras['profesor'][$ar_nas[$nas]['profesor']][$dan][$sat+$blok]['zauzet']
      == -1){
      $prof_slobodan=1;} else {$prof_slobodan=0; break 1;
}
if ($ras['dvorana'][$row_dvorana['id']][$dan][$sat+$blok]['mjesta'] >=
      $ar_nas[$nas]['mjesta']){
      $mjesta=1;} else {$mjesta=0; break 1;
}
if ($ras['dvorana'][$row_dvorana['id']][$dan][$sat+$blok]['racunala'] >=
      $ar_nas[$nas]['racunala']){
      $racunala=1;} else {$racunala=0; break 1;
}
if ($ras['grupa'][$ar_nas[$nas]['grupa']][$dan]['granica'] <= 7){
      $granica=1;} else {$granica=0; break 1;
}
```

Primjer prikazuje provjeru zahtjeva. Njihov redoslijed je nebitan. Ako algoritam dođe do saznanja da određeni zahtjev nije zadovoljen prekida se petlja i ide se na slijedeći sat. Kad su provjereni svi sati u danu ide se na slijedeći dan , zatim na slijedeću dvoranu, ako su provjerene sve dvorane ide se na novu nastavu. U slučaju da algoritam ne uspije smjestiti nigdje nastavu, nastavlja se niz, ali je kasnije dato do znanja da nije sva nastava uspješno smještena u raspored.

Zahtjevi koje je potrebno zadovoljiti, redom kojim su prikazani u primjeru su: je li dvorana zauzeta, je li grupa zauzeta, je li profesor zauzet, ima li dvorana dovoljno mjesta za sve studente, ima li dvorana računala ako su ona potrebna i provjera jesu li studenti dostigli maksimalan broj sati koji mogu slušati u jednom danu. Ako su svi zahtjevi zadovoljeni, kako je već rečeno, sve varijable se ispunjavaju brojem 1, te započinje nova petlja koja obavlja nekoliko radnji koje su prikazane u primjeru 20.

Primjer 20. Popunjavanje rasporeda

```
if($slobodno==1 && $racunala==1 && $mjesta==1 && $prof_slobodan==1 &&
      $grupa_slobodna==1 && $granica==1){
        for ($blok = 0; $blok < $ar_nas[$nas]['blok']; ++$blok){
           $sb=$sat+$blok;
            $ras['dvorana'][$row_dvorana['id']][$dan][$sb]['zauzeto'] =
                  $ar_nas[$nas]['id'];
            $ras['grupa'][$ar_nas[$nas]['grupa']][$dan][$sb]['zauzeta'] =
                  $ar_nas[$nas]['id'];
            $ras['grupa'][$ar_nas[$nas]['grupa']][$dan][$sb]['dvorana'] =
                  $row_dvorana['id'];
            $ras['grupa'][$ar_nas[$nas]['grupa']][$dan]['granica'] =
                  $ras['grupa'][$ar_nas[$nas]['grupa']][$dan]['granica']+1;
            $ras['profesor'][$ar_nas[$nas]['profesor']][$dan][$sb]
                  ['zauzet'] = $ar_nas[$nas]['id'];
            $ras['profesor'][$ar_nas[$nas]['profesor']][$dan][$sb]
                  ['dvorana'] = $row_dvorana['id'];
            $insert_nastava = $ar_nas[$nas]['id'];
             $insert_dvorana = $row_dvorana['id'];
            if ($semestar==1) $query = "INSERT INTO nastava_zimski
                  (dan, sat, nastava_id, dvorana_id) VALUES
                  ($dan+1, $sb+1, $insert_nastava, $insert_dvorana);";
            if ($semestar==2) $query = "INSERT INTO nastava_ljetni
                  (dan, sat, nastava_id, dvorana_id) VALUES
                  ($dan+1, $sb+1, $insert_nastava, $insert_dvorana);";
           $res_insert = $conn -> query($query);
           if (!$res_insert) echo "INSERT failed: $query<br>" .
                  $conn->error . "<br>";
                              }
                              unset($ar_nas[$nas]);
                              break 3;
                        }
```

Kad su zadovoljeni svi zahtjevi program ispunjava višedimenzionalno polje rasporeda. Polja zauzeto za dvorane, profesore i grupe su zamijenile broj -1 sa id-om nastave koja se održava. Stvorene su nove dimenzije za profesore i grupe u kojima je ispisan id dvorane u kojoj se ta nastava održava. Kad je granica u dimenziji grupe povećana za +1, znači da je ta grupa odradila još jedan sat taj dan. Stvorene su dvije varijable koje sadrže id dvorane, tj. nastave kako bi se mogle dodati u baznu tablicu nastava_zimski ili nastava_ljetni, u obliku vanjskih ključeva. U tablicu se dodaje podatak o danu i satu. U slučaju da upit nije uspješno izvršen program ispisuje obavijest. Kad je sve izvršeno briše se lista te nastave iz višedimenzionalnog polja nastave i prekidaju se tri prethodno otvorene petlje kako bi se moglo preći na sljedeću nastavu.

Nakon što algoritam prođe kroz cijelu nastavu preostaje mu u višedimenzionalno polje rasporeda spremiti količinu lista nastave preostalu u višedimenzionalnom polju nastave. Nakon toga program vraća to polje funkciji raspored. Funkcija raspored ga tada može provjeriti, ako je više od 0 lista nastave ostalo u polju nastave ispiše se obavijest da nije sva nastava uspješno raspoređena u raspored, te je potrebno prilagoditi podatke da zadovoljavaju kriterije.

7. ZAKLJUČAK

Aplikacija sa ovakvom primjenom pridonosi obrazovnom sustavu i olakšava rad određenim članovima njegova društva. Odlike aplikacije su dostupnost, jednostavnost i preglednost u korištenju. Aplikacija je online i moguće joj je pristupiti sa bilo kojeg računala, tableta ili pametnog telefona uz pomoć lozinke. Uređaj koji je pristupio aplikacije ne mora imati bolje performanse, jer se sav rad odrađuje na strani poslužitelja. Jedino što je korisniku potrebno je aplikacija za pregled PDF datoteka, no i u slučaju da ju nema korisnik može pregledati rasporede online u obliku HTML stranice. Podaci koji su raspoređeni na nekoliko stranica se jednostavno unose u bazu, čitljivi su i lako se mogu mijenjati po potrebi. Generiranjem rasporeda izmjenjuju se prijašnje web stranice i PDF-ovi koji su sadržavali stare rasporede. Pogoduje tome da ne postoji potreba za ponovnim učitavanjem poveznica za rasporede na stranice ustanove koje ga koriste, već svi posjetitelji mogu odmah vidjeti novi raspored. Korisnici nemaju slobodu unošenja i izmjene zahtjeva kako oni žele. Prilagođavanje algoritma sa programerske strane nije teško izvedivo. Baza je jednostavna i sve podatke okuplja u jednoj tablici. Potreba za novim elementima se rješava na način da se stvori nova tablica u bazi, te da se njen vanjski ključ unese u glavnu tablicu po kojoj se vrši sortiranje. Dio algoritma za provjeru zahtjeva je pregledan i jednostavan je unos ili promjena zahtjeva.

Nastava u aplikaciji se sastoji od obaveznih kolegija, ne i od izbornih, jer je onda grupa podijeljena na više kolegija, a aplikacija u ovom stadiju još ne prihvaća tu mogućnost. U aplikaciji nije obrađena mogućnost unosa više profesora za jednu nastavu, što je u praksi moguće. U slučaju da se iz nekog razloga korisniku ne sviđa stvoreni raspored, koji je zadovoljio sve uvjete, on ga ne može izmijeniti, jer aplikacija generira po principu da prihvati prvo ispravno rješenje kako ne bi trošila previše vremena. U budućnosti je planirano da profesori i vanjski suradnici mogu odabirati vrijeme kada mogu ili žele raditi. Profesori i suradnici bi bili zadovoljniji rezultatom, no takav zahtjev bi smanjivao mogućnosti i vjerojatnosti da aplikacija uspješno stvori raspored. Moguće je napraviti da se raspored nakon što je generiran izmjeni na način da se polje u rasporedu doda, tj. izbriše po potrebi. Takva mogućnost zahtjeva posve novi algoritam koji provjerava zahtjeve na drugačiji način, a to stvora potrebu pregledavanja nastave po dvorani, profesoru ili studentskoj grupi.

LITERATURA

Applied Software Consultants, a., 2016. *aSc Timetables.* [Mrežno]
Available at: http://www.asctimetables.com/
[Pokušaj pristupa 25. kolovoz 2016.].

Bedek, V., 2008. *Stvaranje rasporeda sati genetskim,* Zagreb: Fakultet elektorinike i računarstva.

Đurasević, M., 2015. *Razvoj pravila rasporedivanja korištenjem genetskog programiranja.* [Mrežno]
Available at: https://www.fer.unizg.hr/_download/repository/Marko_Djurasevic_-_KDI.pdf
[Pokušaj pristupa 20. kolovoz 2016.].

EMS Software, 2016. *ems.* [Mrežno]
Available at: https://www.emssoftware.com/
[Pokušaj pristupa 25. kolovoz 2016.].

Gantt, H. L., 2008. *Scheduling: Theory, Algorithms, and Systems.* 3rd ur. New York: Prentice Hall.

Lynch, P. J. & Horton, S., 2009. *Web Style Guide.* 3rd ur. New Haven: Yale University Press.

Mimosa Software Ltd., 2016. *Mimosa Scheduling Software.* [Mrežno]
Available at: http://www.mimosasoftware.com/
[Pokušaj pristupa 25. kolovoz 2016.].

php.com.hr, 2008. */php.com.hr.* [Mrežno]
Available at: http://php.com.hr
[Pokušaj pristupa 25. kolovoz 2016.].

Pribil, S., 2012. *Algoritmi evolucijskog računanja primijenjeni na problem izrade školskog rasporeda sati,* Zagreb: Fakultet rlektrotehnike i računarstva.

PrimeTimetable.com, 2016. *Prime Timetable.* [Mrežno]
Available at: http://www.primetimetable.com/
[Pokušaj pristupa 25. kolovoz 2016.].

Robin, N., 2014. *Learning PHP, MySQL & JavaScript.* 4th ed. Cambridge: O'Reilly Media.

Welling, L. & Thomson, L., 2008. *PHP and MySQL Web Development.* 4th ur. Boston: Addison-Wesley Professional.

POPIS SLIKA

Slika 1. aSc Rasporedi - Naslovna stranica 8
Slika 2. aSc Rasporedi - Prikaz rasporeda 8
Slika 3. aSc Rasporedi - Kontrola 9
Slika 4. aSc Rasporedi - Statistika 9
Slika 5. Mimosa - Raspored 11
Slika 6. Mimosa - Pregled zdravlja rasporeda 12
Slika 7. Prime Timetable - Raspored 14
Slika 8. Prime Timetable - Izbornik 14
Slika 9. Prime Timetable - Generiranje rasporeda 15
Slika 10. Prime Timetable - Obavjest o greškama 15
Slika 11. Hodogram korištenja 22
Slika 12. Neprijavljeni korisnik 23
Slika 13. Prijavljeni korisnik 23
Slika 14. Tablica Profesori 25
Slika 15. Tablice Studentske grupe i Smjerovi 26
Slika 16. Tablica Dvorane 26
Slika 17. Tablica Nastava 27
Slika 18. Obavjest o stvaranju rasporeda 29
Slika 19. HTML prikaz rasporeda grupe 31
Slika 20. PDF prikaz rasporeda grupe 32
Slika 21. Baza podataka 34

POPIS PRIMJERA

Primjer 1. Dohvaćanje podataka iz baze 38
Primjer 2. Rješenje problema količine podataka 39
Primjer 3. Ispis podataka u tablicu 39
Primjer 4. Red za unos podataka 40
Primjer 5. Dodavanje podataka u bazu 41
Primjer 6. Red za izmjenu podataka 42
Primjer 7. Izmjena podataka iz baze 42
Primjer 8. Brisanje podatka iz baze 43
Primjer 9. Dohvaćanje podataka iz tablice nastave 44
Primjer 10. Prikaz podataka u tablici nastave 45
Primjer 11. Red za dodavanje novog podatka u tablicu nastave 45
Primjer 12. Dohvaćanje podataka grupe 47
Primjer 13. Unos podataka u baznu tablicu nastave 47
Primjer 14. Početak stvaranja stranice rasporeda 49
Primjer 15. Početak stvaranja tablice rasporeda 50
Primjer 16. Ispis podataka u raspored 51
Primjer 17. Stvaranje PDF-a rasporeda 52
Primjer 18. Petlje algoritma 53
Primjer 19. Provjera zahtjeva 54
Primjer 20. Popunjavanje rasporeda 55

POPIS TABLICA

Tablica 1. Usporedba aplikacija po zahtjevima 16

SAŽETAK

Ovaj rad se bavi rješavanjem problema rasporeda nastave pomoću web aplikacije. Kako bi se moglo pristupiti kreiranju aplikacije, obrađeni su: problem rasporeda, njegovi strogi i blagi zahtjevi, te neka od postojećih programskih rješenja, koja su međusobno uspoređena. Do programskog rješenja se došlo koristeći PHP, MySQL, HTML i CSS koje su opisane i obrazložen je razlog njihova odabra. Kako bi se bolje shvatio princip rada aplikacije ovaj rad sadržava opis funkcionalnosti i baze podataka uz primjere koda pisanog u PHP-u i MySQL-u.

Ključne riječi: raspored nastave, web aplikacija, PHP, MySQL, HTML, CSS

ABSTRACT

This thesis deals with solving the university class scheduling problem using web application. In order to start creating application, some topics had to be addresssed: the scheduling problem, its strict and mild requirements and some of the existing software solutions, which were compared. The program solution is implemented by means of PHP, MySQL, HTML and CSS, which are described together with the reasoning of their choice. In order to facilitate understanding of the working principle of the application this thesis contains a description of the functionalities and database with code examples written in PHP and MySQL.

Keywords: class shedule, web application, PHP, MySQL, HTML, CSS