

**UNIVERZITET U BEOGRADU
FAKULTET ORGANIZACIONIH NAUKA**

ZAVRŠNI (MASTER) RAD

**RAZVOJ SERVISA ZA ADAPTACIJU SISTEMA
ELEKTRONSKOG OBRAZOVANJA**

Mentor:

Dr Marijana Despotović - Zrakić

Student:

Aleksandar Nastevski 854/2010

Beograd, januar 2013.

Komisija koja je pregledala rad
kandidata **ALEKSANDAR (MENDO) NASTEVSKI**
pod naslovom **RAZVOJ SERVISA ZA ADAPTACIJU SISTEMA ELEKTRONSKOG**
OBRAZOVANJA i odobrila odbranu:

Mentor: dr Marijana Despotović-Zrakić, docent

Član: dr Božidar Radenković, redovni profesor

Član: dr Gordana Milosavljević, redovni profesor

APSTRAKT

Završni rad se bavi razvojem mehanizama za adaptaciju sistema elektronskog obrazovanja. U prvom delu rada prikazan je teoretski osvrt na sistem elektronskog obrazovanja, koncept i uloge novih tehnologija u obrazovanju.

U drugom, tj. praktičnom delu rada razvijena je veb aplikacija za adaptaciju sistema elektronskog obrazovanja. Veb aplikacija omogućava adaptaciju sistema elektronskog obrazovanja za Moodle LMS sistem, studenata, grupa, kurseva, resursa, aktivnosti i kompletног sadržaja koji je kreiran u Moodle LMS sistemu, automatsko generisanje grupa studenata na osnovu zajedničkih karakteristika, ciljeva studenata, kao i prikaz adaptiranog sadržaja preko modula i bloka koji su napisani za Moodle LMS sistem. Aplikacija omogućava obaveštavanje studenata preko SMS servisa koji je realizovan preko Android aplikacije i veb servisa.

Opisane su koriшћene tehnologije kao i svi mehanizmi adaptacije u aplikaciji. Tehnologije koje će se koristiti prilikom realizacije praktičnog dela radu su NBphp framework, PHP, MySQL, AJAX, XML, Javascript i Android platforma za pisanje mobilne aplikacije.

Primarni cilj rada je adaptacija i prilagođavanje sadržaja studentima u zavisnosti od njihovog interesovanja, ciljeva, karakteristika, sposobnosti itd. Primenom razvijene veb aplikacije i mehanizama za adaptaciju postiže se bolja efikasnost prilikom učenja studenata i savladavanje određenog gradiva.

Ključne reči: servisi, mehanizmi, adaptacija, elektronsko obrazovanje, web aplikacija...

ABSTRACT

Završni rad se bavi razvojem mehanizama za adaptaciju sistema elektronskog obrazovanja. U prvom delu rada prikazan je teoretski osvrt na sistem elektronskog obrazovanja, koncept i uloge novih tehnologija u obrazovanju.

U drugom, tj. praktičnom delu rada razvijena je veb aplikacija za adaptaciju sistema elektroniskog obrazovanja. Veb aplikacija omogućava adaptaciju sistema elektroniskog obrazovanja za Moodle LMS sistem, studenata, grupa, kurseva, resursa, aktivnosti i kompletног sadržaja koji je kreiran u Moodle LMS sistemu, automatsko generisanje grupa studenata na osnovu zajedničkih karakteristika, ciljeva studenata, kao i prikaz adaptiranog sadržaja preko modula i bloka koji su napisani za Moodle LMS sistem. Aplikacija omogućava obaveštavanje studenata preko SMS servisa koji je realizovan preko Android aplikacije i veb servisa.

Opisane su koriшene tehnologije kao i svi mehanizmi adaptacije u aplikaciji. Tehnologije koje će se koristiti prilikom realizacije praktičnog dela radu su NBphp framework, PHP, MySQL, AJAX, XML i Javascript.

Primarni cilj rada je adaptacija i prilagođavanje sadržaja studentima u zavisnosti od njihovog interesovanja, ciljeva, karakteristika, sposobnosti itd. Primenom razvijene veb aplikacije i mehanizma za adaptaciju postiže se bolja efikasnost prilikom učenja studenata i savladavanje određenog gradiva.

Ključne reči: servisi, mehanizmi, adaptacija, elektronsko obrazovanje, web aplikacija...

Curriculum Vitae

Ime i prezime: Aleksandar Nastevski

Datum i mesto rođenja: 11.09.1986., Gnjilane

Adresa: Beograd – Braće Jerkovića 58/32

Mobilni tel.: 065 31 41 441

E-mail: acanastevski@gmail.com



Cilj: Lični i profesionalni razvoj u oblasti softverskog inženjerstva i internet tehnologija. Spreman za dinamičan timski rad i preuzimanje odgovornosti za dodeljene zadatke.

Obrazovanje

- Fakultet Organizacionih Nauka, Beograd, Master studije(2013).
 - Odsek: Elektronsko poslovanje i upravljanje sistemima
 - Prosek: 10,00
- Fakultet Organizacionih Nauka, Beograd, diplomirani inženjer organizacionih nauka (2010).
 - Odsek: Informacioni sistemi i tehnologije
 - Prosek: 8.10
- Srednja škola – Gimnazija „Bora Stanković“ Vranje (2005.).
 - Smer: Prirodno-matematički

Tehnička znanja i sposobnosti

- PHP, MySQL, JavaScript, AJAX, JQuery, XML, HTML, CSS
- Java, C#
- Objektno orijentisani princip programiranja uz korišćenje paterna.
- Projektovanje i razvoj informacionih sistema

Softver

- Softverski paketi: NetBeans i Eclipse razvojna okruženja
- Web paketi: Wordpress, OsCommerce, Joomla
- Framework: NBphp, CakePHP, Code Igniter
- Osnovno znanje: Photoshop
- Platforma: Microsoft Windows

Radno iskustvo

- 2012 – osnivač firme [**eSoftSolutions**](#) koja se bavi izradom specijalizovanih softvera i web aplikacija u obrazovanju
- 2010 – osnivač firme [**NB SOFT**](#) koja se bavi izradom web poslovnih aplikacija, održavanje i razvoj
- 2009 – PHP programer, Centar za strane jezike Kontext, Beograd
- 2008 – administrator, Spring DOO

Životni moto: “Verujte u ono što radite i dogodiće se. Kao profesionalac VAŠ cilj mora biti sledeći: Biti bolji danas nego juče, biti bolji sutra nego danas.”

Sadržaj

1.	Uvod.....	9
2.	Adaptivni sistemi elektronskog obrazovanja	12
2.1.	Elektronsko obrazovanje. Pojam i definicija	13
2.2.	Adaptivni sistemi elektronskog obrazovanja	17
2.2.1.	Adaptivni hipermehanika sistemi obrazovanja (AEHS)	18
2.2.2.	Adaptivni obrazovni sistemi zasnovani na veb-u (AWES)	23
2.2.3.	Adaptivnost u sistemima za upravljanje učenjem	24
3.	Tehnologije korišćene u razvoju rešenja	28
3.1.	PHP.....	29
3.2.	Objekto orijentisano programiranje (OOP)	32
3.3.	JavaScript	33
3.4.	AJAX	35
3.5.	Jquery	37
3.6.	Jqgrid	38
3.7.	HTML i CSS	39
3.8.	XML.....	40
3.9.	Android.....	41
3.10.	Moodle LMS.....	44
3.11.	Model View Controller (MVC)	48
4.	Razvoj servisa za adaptaciju sistema elektronskog obrazovanja.....	49
4.1.	Opis korisničkog zahteva aplikacije.....	50
4.2.	Analiza korisničkih zahteva	51
4.3.	Mehanizam za adaptaciju sistema elektronskog obrazovanja	52
4.3.1.	Prošireni model objekti-veze (PMOV).....	54
4.3.2.	Analiza objekata koje treba adaptirati	55
4.3.3.	Personalizovana tabla studenta.....	57
4.3.4.	Mehanizam za adaptaciju kursa	59
4.3.5.	Mehanizam za adaptaciju grupe	60
4.3.6.	Mehanizam za adaptaciju resursa	61
4.3.7.	Mehanizam za adaptaciju aktivnosti	62
4.3.8.	Mehanizam za definisanje ciljeva, kriterijuma, ograničenja	63
4.3.9.	Mehanizam za adaptaciju studenata	64
4.3.10.	Mehanizam za automatsko i ručno razmeštanje studenata po grupama	65
4.3.11.	Mehanizam za prikaz adaptiranih objekata	68

4.3.12.	Prikaz adaptiranih objekata (adaptivni režim).....	69
4.3.13.	Arhitektura mehanizma za adaptaciju	70
4.3.14.	Spisak tabela adaptivnog mehanizma.....	71
4.4.	Slučajevi korišćenja.....	74
4.4.1.	Slučaj korišćenja – Definisanje kriterijuma adaptacije	75
1.1.1.	Slučaj korišćenja - Kreiranje nove ankete.....	77
1.1.2.	Slučaj korišćenja – Dodavanje i uklanjanje sadržaja ankete.....	79
1.1.3.	Slučaj korišćenja – Adaptacija grupa.....	80
1.1.4.	Slučaj korišćenja – Adaptacija resursa i aktivnosti	82
4.5.	Uputstvo za instalaciju adaptivnog mehanizma.....	86
4.6.	Web servis za slanje SMS poruka (nbSMS)	87
4.6.1.	Arhitektura nbSMS servisa	88
4.6.2.	Android API	89
4.6.3.	Web2Sms API	94
4.6.4.	Pregled baze podataka web servisa za slanje SMS poruka	97
4.6.5.	Slučajevi korišćenja – nbSMS web aplikacija	100
4.6.6.	Slučajevi korišćenja – nbSMS android aplikacija	104
5.	Zaključak.....	107
6.	Prilog – Reprezentativni delovi koda	109
7.	Literatura i reference	121

LISTA SLIKA, TABELA I DIJAGRAMA

Slika 1. Model sistema elektronskog obrazovanja	16
Slika 2. Faze procesa učenja na daljinu	16
Slika 3. Model online učenja sa tipovima interakcije	17
Slika 4. Adaptive Hypermedia Application Model (AHAM)	19
Slika 5. Opšta arhitektura AEHS sistema	22
Slika 6. Obrada zahteva preko PHP interpretatora.....	31
Slika 7. Arhitektura Android platforme	43
Slika 8. Izgled kursa u Moodle LMS-u	47
Slika 9. Tok izrade, predaje i ocenjivanja domaćeg zadatka u Moodle LMS	47
Slika 10. Koncept Model-view-controller.....	48
Slika 11. Prošireni model MVC uzora	48
Slika 12. Arhitektura rešenja za razvoj mehanizma za adaptaciju	52
Slika 13. Prošireni model objekti-veze mehanizma za adaptaciju.....	54
Slika 14. Prikaz procesa koji mogu da uređuju personalizovanu tablu.....	58
Slika 15. Arhitektura mehanizma za adaptaciju sistema Moodle LMS	70
Slika 16. Lista tabela adaptivnog mehanizma	73
Slika 17. Lista kreiranih kriterijuma adaptacije	75
Slika 18. Forma za unos novog kriterijuma adaptacije.....	76
Slika 19. Izgled forme dok se čeka da se kriterijum snimi.....	76
Slika 20. Mehanizam za definisanje anketa.....	77
Slika 21. Forma za unos nove ankete	78
Slika 22. Prikaz dela koji dodaje i uklanja sadržaj ankete	79
Slika 23. Forma za prikaz sadržaja kriterijuma	79
Slika 24. Mehanizam za adaptaciju grupa	80
Slika 25. Prikaz kriterijuma adaptacije grupe i potencijalni kriterijumi	81
Slika 26. Mehanizam za adaptaciju resursa i aktivnosti.....	82
Slika 27. Prikaz kriterijuma adaptacije resursa i potencijalnih kriterijuma.....	83
Slika 28. Prikaz potencijalnog kriterijuma adaptacije	83
Slika 29. Dodeljivanje vrednosti cilja za resurs	83
Slika 30. Ručno i automatsko razmeštanje studenata po grupama	84
Slika 31. Prikaz bloka gde se prikazuje adaptiran sadržaj studentima.....	85
Slika 32. Adaptivni režim prikaza sadržaja	85
Slika 33. Root direktorijum aplikacije	86
Slika 34. Arhitektura nbSMS servisa	88
Slika 35. Spisak tabela nbSMS servisa za slanje SMS poruka.....	99
Slika 36. Prikaz liste korisnika servisa.....	100
Slika 37. Dodavanje korisnika	101
Slika 38. Pregled poslatih SMS poruka	102
Slika 39. Kreiranje i slanje nove SMS poruke.....	103
Slika 40. Ekran za logovanje (levi) i početni ekran aplikacije (desno)	104
Slika 41. Pregled liste poslatih poruka i poruka za slanje	105
Slika 42. Podešavanje nbSMS servisa	106
Tabela 1. Objašnjenje parametara za HTTP zahteva za slanje novih SMS poruka	90
Tabela 2. Lista parametara koji se prosleđuju za dobijanje statusa poslatih poruka	92
Tabela 3. Lista moguća grešaka prilikom slanja SMS poruke.....	93
Tabela 4. Opis parametara koje prosleđuje poslovna aplikacija web sms servisu	95
Tabela 5. Opis parametara koji prosleđuje poslovna aplikacija za proveru poslatih SMS poruka.....	96

1. Uvod

Koncept učenja preko Interneta i upotreba globalne mreže u organizovanju i realizaciji obrazovnog procesa poprima sve veći značaj. Ovome doprinose nove tehnologije i Internet, koje se koriste svakodnevno u prikupljanju informacija, komunikaciji, zabavi, pružanju različitih usluga i servisa. Dostupnost informaciono-komunikacionih tehnologija dovodi do promena načina i sadržaja koji se pruža na raspolaganje studentima.

Količina edukativnih sadržaja se rapidno uvećava. Različiti tipovi sadržaja kao što su: tutorijali, elektronske knjige, naučni članci i sl., su sada dostupne na vebu. Pronalaženje adekvatnih materijala za učenje postaje jedan od ključnih problema. Istovremeno, svaki pronađeni resurs ima određene specifičnosti, kada su u pitanju načini prezentacije, oblasti koje pokriva, struktura sadržaja i sl. U tom smislu, postavlja se pitanje: Kako pomoći studentima i drugim učesnicima u obrazovanju, da poboljšaju pronalaženje, organizaciju i upotrebu onih resursa, koji najviše odgovaraju njihovim potrebama, ciljevima, interesima i trenutnom znanju. Dakle, neophodno je obezbediti personalizovan pristup materijalima za učenje.

Potreba za pružanjem personalizovanog pristupa informacijama je prepoznata ne samo u obrazovanju, nego i u ostalim sferama ljudskog delovanja. U poslednje vreme, mnogobrojna istraživanja u oblastima elektronskog poslovanja, marketinga su dovela do razvoja solucija, koje pružaju kvalitetne sisteme personalizacije. Međutim, ta rešenja imaju dosta ograničenu primenljivost u kontekstu elektronskog obrazovanja. Osnovni razlog je velika diverzifikovanost interesa, veština, ciljeva i stilova učenja studenata. Korisnici servisa e-obrazovanja pripadaju heterogenim grupama, koje odlikuju različite karakteristike. Usled sve većeg značaja i zastupljenosti elektronskog obrazovanja, prilagođavanje obrazovnih sadržaja i aktivnosti prema pojedinačnom korisniku, predstavlja jedno od najčešćih problema, koji se danas razmatraju.

Osnovni cilj rada je da se obezbedi adaptivnost u postojećem sistemu za elektronsko obrazovanje u Moodle LMS-u, kreiranjem aplikacije koja će sama po

sebi biti generička i koja će omogućiti profesorima da automatski ili ručno adaptiraju sadržaj studentima u zavisnosti od ciljeva, kriterijuma i ograničenja. Sam cilj rada jeste da se razvije servis za adaptaciju sistema elektronskog obrazovanja, a da implementacija adaptivnog mehanizma ostaje da bude odluka profesora ili nekog drugog ekspreta. Osnovni cilj rada se, prema tome, može definisati kao potreba da se studentu prikaže adaptirani sadržaj koji je prilagođen njegovim potrebama, zahtevima, ciljevima, interesovanjima itd.

U drugom poglavlju su opisani osnovni pojmovi i definicije elektronskog obrazovanja, kao i koncepti i modeli elektronskog učenja i faze procesa učenja na daljinu. Pored toga u drugom poglavlju su opisani i već razvijeni sistemi adaptivnog učenja koji su zasnovani na internetu, kao i mogući tipovi adaptacije. U ovom delu opisani su opšti problemi adaptivnih sistema kao i problemi samog procesa adaptacije. Osvrt je dat i na značaju mobilnog obrazovanja, tj učenja putem mobilnih uređaja, što u današnjem svetu poprime sve veći značaj s obizoram da u praksi imamo sve veći broj korisnika koji koriste mobilne uređaje, a sa druge strane ti uređaji se stalno usavršavaju i sve su nalik klasičnim računarima.

U trećem poglavlju su opisane tehnologije koje su korišćene prilikom razvoja servisa za adaptaciju sistema elektronskog obrazovanja. U ovom poglavlju opisan je serverski programski jezik PHP (Hypertext Preprocessor) koji se koristi za razvoj Server-Side aplikacija i dinamičkog web sadržaja. Pored PHP-a u ovom poglavlju opisane su osnove savremenih tehnologija koje se koriste za razvoj web rešenja poput JavaScripta, AJAX-a, Jquery-a, JQGrid-a kao i osnovna arhitektura i struktura Moodle LMS sistema za elektronsko učenje, kao i osnovni koncepti objektnog orijentisanog programiranja. U okviru ovog poglavlja, opisana je relativno nova platforma za mobilne uređaje Android, koja se veoma brzo proširila širom sveta i sve veći broj mobilnih uređaja (telefoni, smartfn uređaji, tableti itd) koji su napravljeni na toj platformi.

U četvrtom poglavlju opisan je način i obrazloženje kako je i zbog čega razvijen servis za adaptaciju sistema elektronskog obrazovanja. Prikazan je detaljan opis korisničkog zahteva, kao i detaljan opis i razvoj servisa za adaptaciju koji su razvijeni u okviru Moodle LMS sistema. U ovom poglavlju opisani su i slučajevi korišćenja aplikacije za adaptaciju sistema elektronskog obrazovanja. Ovde je takođe dat prikaz

web servisa koji služi za obaveštavanje studenta putem SMS poruka, opisana je web aplikacija i android aplikacija.

U petom poglavlju opisano je do kojih se zaključaka došlo prilikom analize i razvoja rešenja za adaptaciju sistema elektronskog obrazovanja, opis mogućnosti koje pruža sama aplikacija kao i mogućnosti za nadogradnju i dalji razvoj aplikacije.

U šestom poglavlju su priloženi i opisani reprezentativni delovi koda koji su korišćeni u razvoju aplikacije.

2. Adaptivni sistemi elektronskog obrazovanja

Adaptivni sistemi služe da prilagođavaju određen sistem različitim okolnostima. Kompleksan proces prilagođavanja sistema zasniva se na određenim predefinisanih stanja korisnika i saznanja do kojih se došlo prilikom samog procesa prilagođavanja. Bitne karakteristike na osnovu kojih se radi adaptacija jesu kriterijumi, očekivanja i ciljevi korisnika, kao i određeni afiniteti i stilovi učenja.

Ovi sistemi imaju mogućnost da prave razliku između korisnika i samim tim različitim korisnicima da distribuiraju različit sadržaj.

U kontekstu elektronskog obrazovanja, adaptivni sistemi su više fokusirani i specijalizovani za adaptaciju sadržaja i prikazivanje sadržaja korisniku.

Svi adaptivni sistemi generalno prolaze kroz tri faze: 1) proces prikupljanja podataka o korisniku, 2) proces analize i pravljenje korisničkog modela, i 3) sam čin prikaza adaptiranog sadržaja.

Da bi se shvatili kompleksan proces adaptivnih sistema elektronskog obrazovanja, mora predhodno da se prikaže malo više informacija o samom konceptu elektronskog obrazovanja, a kasnije i da se prođe detaljno kroz kompletan proces adaptacije i adaptivnih sistema.

2.1. Elektronsko obrazovanje. Pojam i definicija

Elektronsko obrazovanje (*eLearning*) je način obrazovanja koje se zasniva na upotrebi savremenih tehnologija (*TSL – Technology supported education/learning*) u učenju. Jedno od prvih predstavljanja ovog koncepta bilo je 1993. godine.

William D. Graziadei je predstavio *online* razmenu tutorijala, predavanja i projekata koristeći elektronsku poštu, Gopher/Lynx iVAX Notes. Lynx je omogućavao korisnicima Unix-a i VMS platforme veoma prihvatljiv interfejs za razmenu podataka a da pritom zadrže kontrolu nad istim.

VAX Notes je sistem za konferencije koji se zasniva na računarima i koji omogućava korisnicima da na lak i jednostavan način razmenjuju informacije. Za razliku od elektronske pošte gde poruku može da vidi samo jedan korisnik, ovde su poruke dostupne svim učesnicima konferencije što bitno ubrzava proces komunikacije a i štedi resurse.

Ove tehnologije su omogućile profesorima da zajedno sa studentima formiraju virtualno okruženje koje služi obrazovanju VICES - *Virtual Instructional Classroom Environment in Science*.

Obrazovanje na daljinu ne podrazumeva obaveznu upotrebu modernih informaciono-komunikacionih tehnologija, za razliku od elektronskog obrazovanja, jedan vid obrazovanja na daljinu je korespondencija poštom. Dakle, kod elektronskog obrazovanja se radi o upotrebi računara, Interneta, mobilnih uređaja za razmenu informacija. Naravno da je elektronsko obrazovanje pogodan, verovatno i najčešće korišćen način za realizaciju obrazovanja na daljinu, a može se koristiti i kao dopuna klasičnom obrazovanju.

Proizvodi koji omogućavaju *eLearning* moraju biti laki za upotrebu, održavanje, skalabilni (ne smeju ograničavati broj korisnika, kurseva, materijala, itd.), jeftini u eksploataciji. Virtualne učionice, blogovi, kolaborativni softver (softver dizajniran tako da podrži i pomogne sve učesnike u ostvarivanju nekog zajedničkog cilja), ePortfolio (kolekcija elektronskih fajlova, slika, multimedije održavanih od strane korisnika) su

samo neke od tehnologija koje se koriste, najčešće zajedno, u elektronskom obrazovanju. Mnogo napora se ulaže u kreiranje standarda koji će omogućiti razmenu i ponovnu upotrebu obrazovnih materijala, nastavnih jedinica (*Learning Objects*).

Nastavna jedinica predstavlja jednu zaokruženu celinu, i ako se pravilno formatira, dokumentuje lako se može upotrebiti u nekom drugom kontekstu ili drugoj platformi za razvoj *elearning* portala. Za ovakvo dokumentovanje, formatiranje sadržaja se koriste meta tagovi, ključne reči i slično i to najčešće korišćenjem XML-a. Često se pojavljuje termin *mobile learning* tj. *m-learning* koji označava korišćenje mobilnih tehnologija u obrazovanju.

M-learning koristi sve prednosti mobilnih tehnologija (3G, Wi-Fi, GPRS) i ukida ograničenja vezana za lokaciju. Procenjuje se da *eLearning* tržište vredi oko 38 milijardi eura.

Elektronsko obrazovanje je kompleksan sistem koji uključuje sledeće elemente:

- Učenje na daljinu (distance learning) i predavanja na daljinu (distance teaching), koji su odvojeni vremenski i prostorno.
- Nastavne materijale koji mogu da budu u raznim formama (štampani materijali, audio vizuelni, ...).
- Proces učenja koji može da bude individualni i grupni.
- Tutorski rad kombinacijom raznovrsnih formi “face-to-face” komunikacija korišćenjem medija.
- Interaktivni rad i postizanje sinergijskog efekta grupe studenata.

Obrazovanje na daljinu se razvijalo paralelno sa razvojem tehnologije. Razvoj Internet tehnologija omogućilo je vremensku i prostornu odvojenost učenja i predavanja, a razvoj multimedijalnih tehnologija je omogućilo realizaciju nastavnih materijala sa interaktivnim elementima. Uspeh studenata u procesu elektronskog obrazovanja zavisi od kvaliteta kursa, prezentacije instruktora, interakcije unutar kursa, opreme i tehnologije.

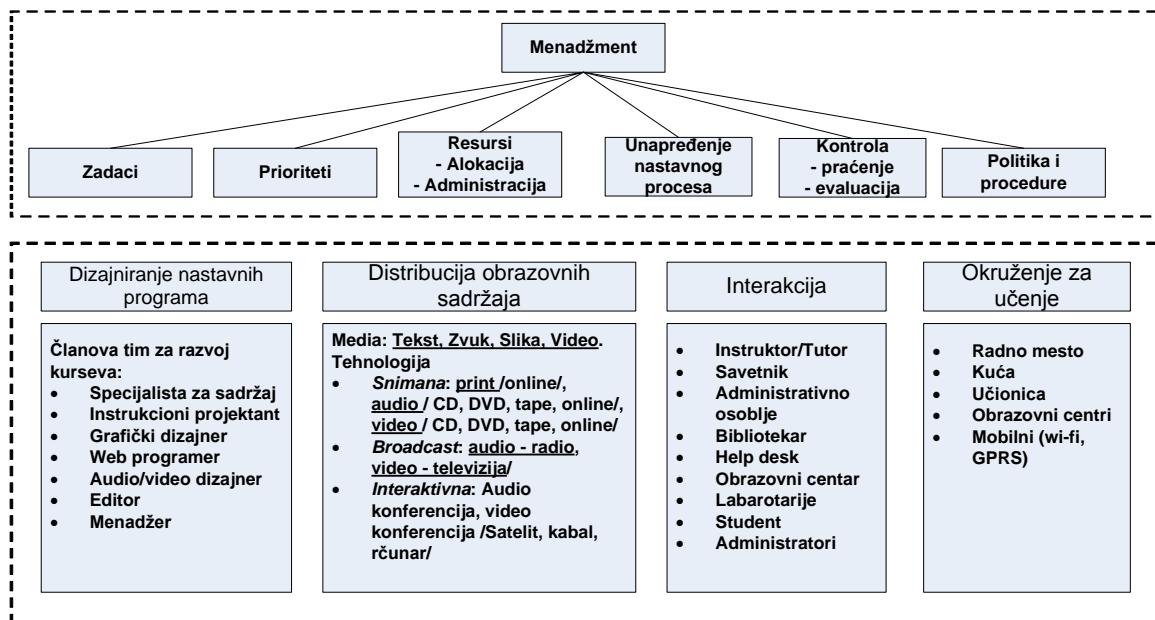
Koristeći ove nove mogućnosti, učenicima i studentima je omogućeno da samostalno savladavaju nastavne sadržaje, što zahteva novi didaktičko - metodički pristup, kao i obučenost svih učesnika u nastavnom procesu. Nove tehnologije omogućavaju da se proces učenja odvija bilo gde i bilo kada, bez geografskih ili vremenskih ograničenja.

U evropskom obrazovnom sistemu, ovakav vid učenja je alternativa tradicionalnom, dok kod nas najviše je zastavljen u okviru programa za obrazovanje odraslih i u okviru visokog obrazovanja.

Osnovni koncepti standarda e-obrazovanja obuhvataju:

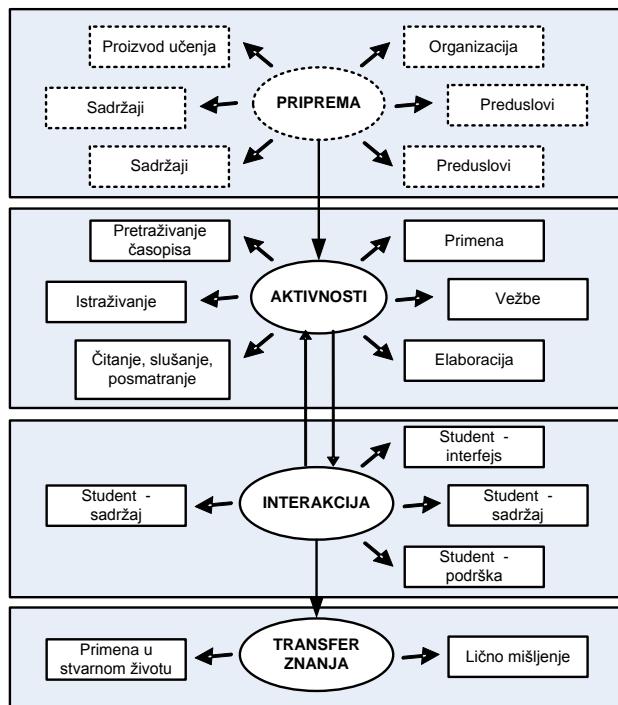
- **dostupnost (accessibility)** - daje autoru kursa mogućnost pristupa obrazovnim sadržajima sa bilo koje lokacije,
- **višestruko korišćenje obrazovnih materijala (reusability)** – ponovno korišćenje obrazovnih sadržaja u različitim situacijama,
- **interoperabilnost (interoperability)** – obrazovni sadržaji se mogu isporučivati bez obzira na alate kojima su kreirani, obrazovnu platformu na kojoj se isporučuju itd.
- **trajnost (durability)** - osigurava korišćenje obrazovnih sadržaja bez potrebe modifikovanja slučaju nadogradnje i unapređenja sistema.

Postoje tri različita pristupa identifikaciji forme elektronskog obrazovanja: sa aspekta medijuma na kome se nastavni materijali baziraju, organizacione strukture i vrste komunikacije učesnika u obrazovanju. Model elektronskog obrazovanja možemo da posmatramo sa konceptualnog (Slika 1) i sistemskog aspekta (Slika 2).



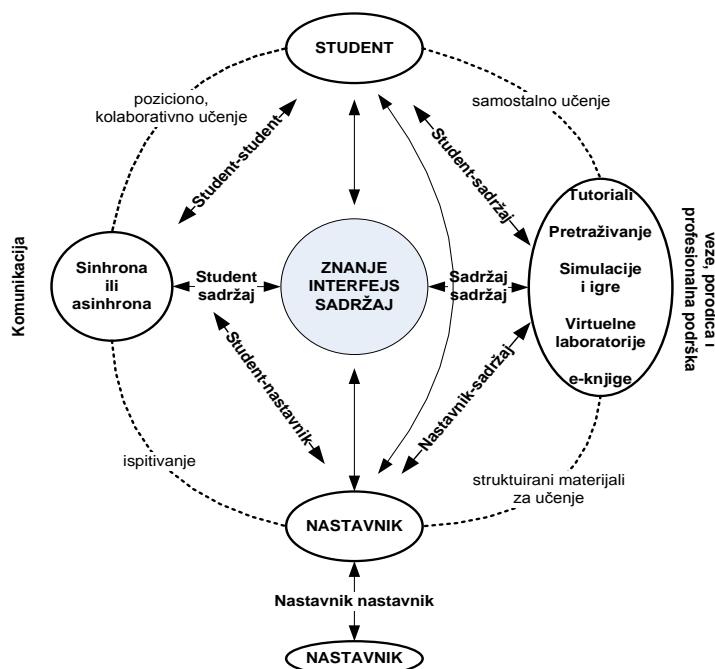
Slika 1. Model sistema elektroniskog obrazovanja

Ako posmatramo elektronsko obrazovanje sa vremenskog aspekta možemo da uočimo četiri faze (Slika 1): pripremu (organizacione pripreme, priprema sadržaja, materijala, i preduslove koje student mora da zadovolji), aktivnosti studenta u toku samog obrazovnog procesa, interakciju studenata sa nastavnim sadržajima, evaluaciju stečenog znanja i transfer znanja.



Slika 2. Faze procesa učenja na daljinu

Na Slici 3 prikazan je model učenja sa aspekta tipova interakcije učesnika u procesu elektronskog obrazovanja. Model prikazuje interakciju studenta i profesora kao i interakciju sa online sadržajima.



Slika 3. Model online učenja sa tipovima interakcije

Kod modela za online učenje centralno mesto zauzima obrazovni sadržaj u formi interfejsa. Ovaj model se zasniva na međusobnoj interakciji nastavnika i studenata (kao glavnih učesnika u obrazovanju), zatim interakciji studenata sa nastavnim sadržajima (tutorijali, virtuelne laboratoriјe, elektronske knjige), kao i interakcija student – student, gde studenti komuniciraju (sinhrono i/ili asinhrono) posebno ili u grupama.

2.2. Adaptivni sistemi elektronskog obrazovanja

Kursevi elektronskog učenja veoma često imaju problem "univerzalne veličine", pošto se isti statički sadržaj prezentuje svim studentima bez uzimajući različite potrebe sutdenata, sa ciljem da se učenik dovede "on-line" i "u" samo tehnologiju. Veći broj istraživanja je dokazao da ovaj način organizacije sistema za elektronsko učenje završava neuspehom ili pak delimičnim uspehom.

Sistem elektronskog učenja se definije kao adaptivan, ako je u mogućnosti da:

- prati aktivnosti svojih učesnika;
- interpretira iste na osnovu oblasno-specifičnih modela;
- otkriva zahteve i preferencije u skladu sa prethodno uočenim aktivnostima i precizno ih reprezentuje u povezanim modelima.

Pošto se ponašanje sistema prilagođava učesniku, tj. osobi, ova vrsta adaptacije se naziva personalizacija. Prema tome, adaptivni sistem elektronskog učenja se može opisati kao *personalizovan* sistem, koji je pored kreiranja personalizovanih sadržaja, sposoban da obezbedi adaptivno "dostavljanje" kursa, interakciju, saradnju i podršku.

Personalizovano e-učenje koristi aktivnu strategiju učenja koja osposobljava učenika da kontroliše sadržaj, tempo i obim učenja. Ono podržava učenika omogućavajući mu alate i mehanizme pomoću kojih on može personalizovati svoje učenje. Adaptivno personalizovano elektronsko učenje predstavlja sastavni elemenat sledeće generacije sistema elektronskog obrazovanja. Ono nudi viziju dinamički kreiranih kurseva koji su prilagođeni specifičnim potrebama pojedinca, prethodnom znanju, računarskom okruženju, povezanošću i komunikacijskim prednostima.

Međutim, osnovnu barijeru u razvoju adaptivnog elektronskog učenja predstavljaju složenost, visoki troškovi i dugačak vremenski period koji su potrebni za razvoj ovakvog sistema. Ključni problem koji se razmatra u ovom radu je na koji način se projektuje i implementira jedan adaptivni sistem elektronskog učenja. U narednom tekstu je dat pregled postojećih rešenja dostupnih u referentnoj literaturi, kao i tehnologija pomoću kojih se sistemi realizuju.

2.2.1. Adaptivni hipermedia sistemi obrazovanja (AEHS)

Pojam adaptivnih hipermedia sistema se zasniva na terminima hipertekst i hipermedia. Hipertekst se odnosi na nesekvencijalni tekst, koji je povezan hyperlinkovima. Hipermedia proširuje koncept hiperteksta pomoću objekata kao što su: slike, audio, video zapisi, animacije i sl.

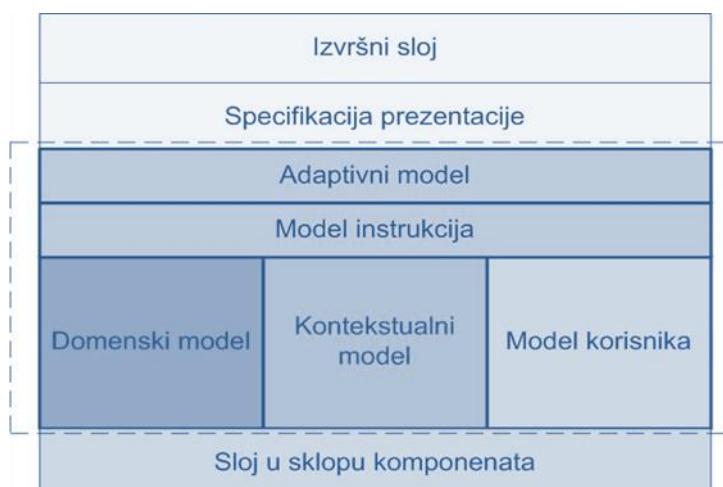
Adaptivni hipermedia sistem (AHS) se definiše kao hipertekst ili hipermedia sistem, koji reflektuje odgovarajuće osobine korisnika (učenika) u korisničkom modelu.

Osnovna karakteristika je sposobnost prilagođavanja osobinama korisnika (npr. ciljevi, teme, znanja, pozadina, iskustvo, preference, interesovanja i sl.) i okruženja (lokacija, protok, računarska platforma i dr.)

Adaptivni hipermedia sistem obrazovanja ima ulogu da:

- pomogne učeniku u pronalaženju najpogodnijeg resursa za učenje
- obezbedi razumevanje procesa učenja u određenoj oblasti
- vodi učenika kroz učenje
- obezbedi orientaciju učenika u mnoštvu materijala

Najzastupljeniji referentni model za aplikacije adaptivnog hipermedia sistema je AHAM (De Bra, P., Houben, G. and Wu, 1999). Prema ovom modelu AHS se sastoji iz elemenata prikazanih na slici 4:



Slika 4. Adaptive Hypermedia Application Model (AHAM)

Domenski model (Domain Model)

Definiše konceptualni dizajn adaptivne aplikacije, odnosno šta će biti adaptirano. Informaciona struktura domenskog modela u tipičnom hipermedia sistemu se može opisati preko dve povezane mreže objekata:

- Oblast znanja (Knowledge Space) – mreža koncepata
- Hiperoblast (Hyperspace) – mreža hiperdokumenata

U skladu sa tim, dizajn adaptivnog hypermedia sistema obuhvata tri ključna koraka: strukturiranje znanja, hiper oblasti i povezivanje ova dva područja.

Model sadržaja (Content model)

Definiše različite vrste objekata učenja i njihove komponenete. Opšti model sadržaja obuhvata:

- Fragmenti sadržaja – elementi sadržaja u osnovnoj formi (tekst, audio, video) koji reprezentuju pojedinačne izvore bez povezivanja sa drugim
- Objekti sadržaja – setovi fragmenata sadržaja; apstraktni tipovi □
- Objekti učenja – agregiraju i povezuju postojeće objekte sadržaja i dodaju ciljeve učenja. Standardi koji se koriste na ovom nivou:
 - ✓ IMS Content Packaging – opis materijala za učenje
 - ✓ IMS Question and Test Interoperability – XML jezik za opis pitanja i testova
 - ✓ IEEE Learning Object Matadata – opis izvora učenja

Model korisnika (User Model)

Model korisnika reprezentuje relevantne karakteristike korisnika kao što su: preference, kompetencije, oblasti i ciljevi. Model korisnika bi trebalo da sadrži lične informacije o korisniku, kontakt sa sistemom, servise koje je koristio, načine korišćenja i objašnjenja te interakcije. Većina adaptivnih hipermehdia sistema koristi tzv. *overlay model*, zasnovan na znanju korisnika.

Ključni princip *overlay modela* je da za svaki koncept domenskog modela, pojedinačni model znanja korisnika sadrži podatke, koji predstavljaju procenu nivoa znanja korisnika o tom konceptu. Ponderisani *overlay model* se može posmatrati kao niz parova koncept – vrednost, za svaki domenski koncept. Alternativni model je tzv. *historic model*, koji čuva informacije o posetama korisnika stranicama. Pojedini sistemi koriste ovaj model kao komplementarni izvor adaptacije.

Ciljevi učesnika se mogu modelovati kao set kompetencija koje se mogu predstaviti slično kao i *overlay model*. Kao dodatak dinamičkim dimenzijama modela korisnika, treba definisati i preferencije korisnika. Najčešće se opisuju kroz stilove učenja, motivaciju i jezik.

Kontekstualni model (Context Model)

Pojam kontekst se može definisati kao skup okolnosti u okviru kojeg se nešto odvija. U okviru adaptivnog učenja, ovaj termin se odnosi na karakteristike okruženja. Model korisnika i model konteksta određuju parametre na osnovu kojih se obavlja adaptacija. Osnovni cilj je generisanje meta podataka automatski na osnovu odgovarajućeg konteksta. Ovo bi omogućilo preciznije prikupljanje i izdvajanje podataka, kada se objekti učenja obrađuju od strane nastavnika i učenja.

Upravljanje kontekstom se odnosi na teme kao što su: automatizovana akvizicija kontekstualnih metapodataka, prezentovanje sadržaja, aktivnosti i usluga.

Model instrukcija (Instruction Model)

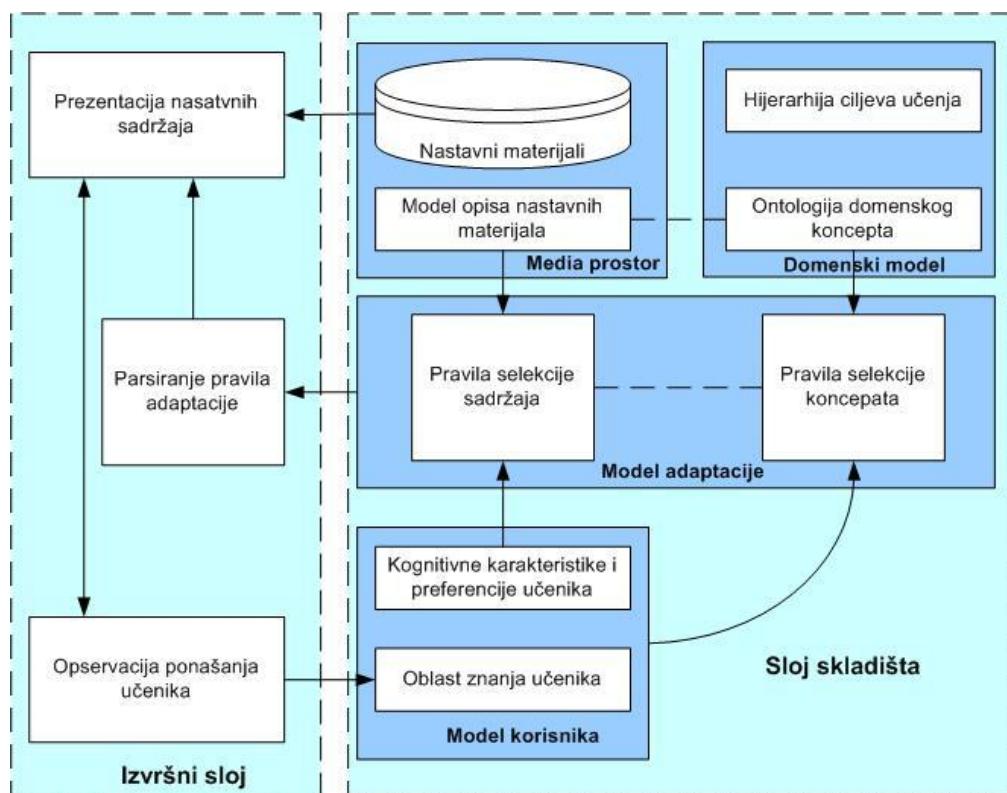
Instrukcioni model i modeli adaptacije specificiraju navigacioni dizajn aplikacije adaptivne hipermedia sistema. Istovremeno, pored specifikacije prezentacije, opisuju način kako bi trebalo izvršiti adaptaciju, odnosno opisuju dinamiku sistema. Dizajn elektronskog učenja predstavlja način modelovanja aktivnosti i scenarija i pristupa učenja.

Ključni aksiom, zajednički za sve pristupe organizovanja učenja, je «učenik obavlja aktivnosti učenja u okruženju sa određenim resursima». Osnovni ciljevi se odnose na opisivanje dinamičke interakcije između korisnika i sistema, uloga učesnika i servisa sistema, scenarija (ciljeva, tema, aktivnosti).

Model adaptacije (Adaptation Model)

Model specificira semantiku adaptacije. Adaptacija definiše status pojedinačnih objekata (objekti sadržaja ili delovi) na osnovu atributa metapodataka i trenutnih parametara modela korisnika i kontekstualnog modela. Adaptacija se najčešće realizuje prilagođavanjem sadržaja i linkova koristeći pogodne tehnike adaptacije. Na višem nivou prezentacije se definiše kako bi svaki pojedinačni objekat trebalo da bude prezentovan korisnicima (npr. prezentovanje, sortiranje, podvlačenje, objavljuvanje i sl.)

Na slici 5 je prikazan model arhitekture adaptivnog hipermedia sistema elektronskog obrazovanja :



Slika 5. Opšta arhitektura AEHS sistema

Tipovi adaptacije:

- 1) Adaptivna interakcija – odnosi se na adaptaciju koja se odigrava na interfejsu sistema i ima ulogu da poboljša i podrži interakciju korisnika sa sistemom, bez modifikacije samog sadržaja
- 2) Adaptivno dostavljanje kursa – tehnike adaptacije, koje se odnose na usklađivanje kursa prema pojedinačnom korisniku. Osnovni cilj je da se napravi veza između sadržaja kursa i karakteristika korisnika, tako da se postigne optimalan rezultat učenja. Tipični primeri: adaptivna navigacija, adaptivna selekcija sadržaja kursa, dinamičko restrukturiranje kursa, vremenski raspored, itd.
- 3) Pronalaženje i kreiranje sadržaja – odnosi se na primenu adaptivnih tehniku u otkrivanju i kreiranju onih sadržaja koji odgovaraju karakteristikama učenika, ali i ciljevima kursa.
- 4) Adaptivna kolaborativna podrška – podrazumeva podršku učenja u smislu organizacije komunikacije između učesnika i usmeravanja ka zajedničkim ciljevima.

2.2.2. Adaptivni obrazovni sistemi zasnovani na veb-u (AWES)

Adaptivni sistemi elektronskog učenja zasnovani na vebu (AWES) su razvijani na teorijskim konceptima AHES. U narednom tekstu su navedeni neki od primera AWES, objašnjeni sa aspekta funkcionalnosti i adaptivnih tehniku koje koriste.

Intelligent Distributed Environment for Active Learning (IDEAL) (Shang, Shi, and Chen, 2001) je adaptivni sistem, zasnovan na inteligentnim agentima sa ciljem obezbeđivanja aktivnog učenja. Materijali za učenje su adaptirani u smislu organizacije, selekcije i prezentacije u zavisnosti od predznanja, stilova učenja i dostupnosti.

Task-based Adaptive Learner Guidance On the Web (TANGOW) (Carro, Pulido, and Rodriguez, 2001) je adaptivni sistem dizajniran za razvoj veb zasnovanih kurseva, gde se adaptacija zasniva na tzv."taskovima učenja" i pravilima. Sadržaj je

organizovan kao lista media elemenata povezanih u "taskove". Adaptacija se realizuje modifikacijom "taskova" i elemenata u okviru njih.

Adaptive Hypermedia for All (AHA!) (AHA! 2007; de Bra and Calvi, 1998; Stash, Cristea, and de Bra, 2006) – adaptivni sistem koji se u potpunosti zasniva na AHA teoriji. Adaptacija se realizuje kroz adaptivnu selekciju tema, kreiranje različitih navigacionih putanja, raspored informacija.

(ELM Adaptive Remote Tutor) (ELM-ART) III (Brusilovsky, P., Schwarz, E. and Weber, G., 2001) organizovan kao interaktivna inteligentna elektronska knjiga sa hipermedialnim sadržajem. Metodi adaptacije: navigaciona podrška, kurs kao sekvenca, "problem based" pristup.

InterBook (Brusilovsky, P., Eklund, J., and Schwarz, E. 1998) – adaptivni sistem, koji se zasniva na kreiranju adaptivnih navigacionih putanja i linkova.

Adaptive Personalized eLearning Services (APeLS) (Trinity College Dublin) – sistem se zasniva na jasnoj separaciji modela sadržaja, učenika sa jedne, i adaptivnog mehanizma sa druge.

2.2.3. Adaptivnost u sistemima za upravljanje učenjem

Sistemi za upravljanje učenjem (Learning management systems) se uspešno primenjuju u elektronskom obrazovanju. Sistemi za upravljanje učenjem (*Learning Management Systems - LMS*) omogućavaju konstrukciju pogodnih okruženja za učenje komponovanjem resursa za učenje. LMS sistemi su orijentisani ka kreiranju i komponovanju različitih objekata učenja.

Korišćenjem alata za autorizaciju, upravljivost ITS je potpuno u rukama kreatora (autora), dok su LMS virtuelna okruženja koja u potpunosti upravljaju resursima. Nastavnici (kao autori kurseva) i učenici predstavljaju samo korisnike sistema (klijente). Administrativne funkcionalnosti su potpuno u rukama serverske strane. Pored navedenih odlika, LMS nude mogućnost različitih vidova kolaboracije u toku procesa učenja kao i veću transparentnost resursa, a time i "vidljivost" autora na

Web-u. Jedna od osnovnih tendencija u ovim sistemima je da obezbede višestrukost korišćenja objekata za učenje (*Learning Objects - LO*). Neki od najpoznatijih LMS su: WebCT, Blackboard, ATutor i Moodle.

Primarni cilj ovih sistema je kreiranje i organizovanje online kurseva, međutim, najčešće, ne uzimaju se u obzir razlike između učenika. Isti kurs može u potpunosti odgovarati jednom korisniku, ali istovremeno i predstavljati veliku poteškoću drugom. Korisnici LMS sistema pripadaju heterogenim grupama sa suprotnim osobinama i potrebama. Ogromne količine informacija, aktivnosti i zadataka, mogu prouzrokovati nesnalaženje i nepostizanje optimalnih rezultata učenja. Neophodno je postaviti učenika u centar obrazovnog procesa. S tim u vezi, i LMS bi trebalo da obezbede usklađenost kurseva sa potrebama i karakteristikama korisnika.

LMS podrazumevaju postojanje različitih korisničkih uloga (nastavnik, administrator, učenik, itd.). Selekcija sadržaja koji se prikazuje, zasniva se na definisanim ulogama. Dakle, LMS podržavaju pojedine forme adaptivnosti, ali neophodno je obezbediti sofisticirane mehanizme adaptacije, kao što je to slučaj u adaptivnim hipermedia sistemima elektronskog obrazovanja. Minimum adaptivnosti u veb aplikaciji elektronskog obrazovanja obuhvata sakupljanje podataka o interakciji studenta sa sistemom i kreiranje modela studenta.

Osnovne funkcionalnosti koje tipičan LMS obezbeđuje korisnicima obuhvata:

- upravljanje kursevima – kreiranje, administracija, adaptacija i supervizija kurseva
- upravljanje “učionicom” – alati za upravljanje nastavom, kreiranje grupa, zadataka, raspored
- komunikacioni alati – komunikacija i interakcija korisnika na efikasan način (email, chat, konferencije, obaveštenja, deljenje sadržaja i aplikacija)
- upravljanje sadržajem za učenje – kreiranje, skaldištenje i distribuiranje materijala za učenje
- alati za evaluaciju učenja – statistike, izveštaji i sl.
- administracija opštim podacima – performance nastavnog procesa, podaci o studentima, finansijska pitanja i sl.

Za svaki od ovih servisa je potrebno obezbediti interfejs koji se prilagođava zahtevima i potrebama korisnika.

Adaptivni sistemi elektronskog obrazovanja se fokusiraju na prilagođavanje kurseva individualnim karakteristikama korisnika. Međutim, ovi sistemi imaju i nedostatke.

Pre svega, kreiranje ovakvih sistema i njihovo uključivanje u obrazovne procese je dosta kompleksan i skup proces, koji zahteva visok nivo angažovanja svih korisnika. Ovi sistemi imaju zadatak da pruže adaptivne funkcionalnosti, koje se, međutim veoma često ne uklapaju u proces učenja. Jedan od čestih problema je i nemogućnost ponovnog korišćenja materijala učenja. Sistemi nisu interoperabilni, razvijeni su kao izolovane aplikacije i ne mogu da razmenjuju resurse niti podatke o učenicima. Istraživanja pokazuju da su ovi sistemi uglavnom prototipski i eksperimentalni sa ograničenim interfejsom. Osnovni servisi sistema elektronskog obrazovanja, kao što su administracija kurseva, kreiranje sadržaja i sl. su kompleksni u adaptivnim sistemima. Usled složenosti sistema, od korisnika se zahteva određeno predznanje o samom sistemu. Ne postoje osnovni servisi kada su u pitanju komunikacija i socijalna interakcija između učesnika u elektronskom obrazovanju.

Sa druge strane, LMS se fokusiraju na podršku svim procesima u elektronskom obrazovanju. Osnovni cilj je omogućiti korisnicima adekvatne usluge koje olakšavaju organizovanje učenja i samo učenje. Istovremeno, alati za komunikaciju kao što su forumi, chat, wiki, blogovi i sl. poboljšavaju interakciju tokom pohađanja kurseva. Dakle, ključni problem u adaptivnim sistemima se odnose na arhitekturu i realizaciju sistema, a ne na adaptivnost. LMS ne obezbeđuju adekvatan nivo adaptacije sadržaja i servisa. Jedna od ključnih ideja ovog rada je razvoj sistema e-obrazovanja preko integracije adaptivnih servisa i funkcionalnosti sa već postojećim LMS, pre nego razvoj posebnog adaptivnog sistema elektronskog e-obrazovanja.

Primeri dostupnih adaptivnih LMS, realizovanih kreiranjem i integracijom dodatnih adaptivnih funkcionalnosti na već postojećim LMS su: *WHURLE 2.0* i *E-class*.

3. Tehnologije korišćene u razvoju rešenja

Prilikom razvoja aplikacije za adaptaciju sistema elektronskog obrazovanja korišćen je serverski programski jezik PHP (Hypertext Preprocessor) koji se izvršava na strani servera. U PHP-u je odrađena kompletna poslovna logika aplikacije. U kombinaciji sa PHP-om korišćena je MySQL baza podataka kao i SQL upitni jezik za manipulaciju sa bazom.

Na ovoj tehnologiji je urađen i Web SMS servis koji je povezan sa android aplikacijom za slanje SMS poruka studentima.

Za obradu i povećanu interakciju sa korisnikom korišćen je klijentski skript jezik JavaScript kako bi se ubrzao rad same aplikacije kao i AJAX koji je omogućio da se obrađeni podaci u JavaScriptu, zatim posalati preko AJAX obrade i u PHP-u.

Od savremenih tehnologija za razvoj aplikacije još je korišćena JavaScript biblioteka Jquery(JavaScript Library) koje omogućava brži način kodiranja u JavaScriptu.

Za tabelarni prikaz podataka korišćena je JQGrid tehnologija koja omogućava veoma brzo i efikasno tabelarno prikazivanje podataka iz baze kao i manipulaciju sa tim podacima uz pomoć JavaScripta, Jquery kao i XML-a.

Za prikaz podataka na ekran internet pretraživača korišćen je HTML(HyperText Markup Language). Dizajn sajta je kompletno urađen u CSS(Cascading Style Sheets) uz poštovanje W3C standarda.

Prilikom razvoja aplikacije korišćeno je razvojno okruženje Netbeans, u kome jeписан PHP kod. Administracija baze podataka rađena je u phpMyadminu gde su i proveravani sql kodovi.

Mobilna aplikacija pisana je na Andoird platformi, koja putem web servisa i razmene podataka putem XML komunicira sa WEB SMS servisom za slanje poruka.

3.1. PHP

PHP (Hypertext Preprocessor) je open source server-side skriptni programski jezik za dinamičko generisanje HTML koda. PHP je skriptni jezik pomoću kojeg možete kreirati HTML stranicu na serveru pre nego što se ona, popunjena dinamičkim sadržajem, pošalje klijentu. Ovim načinom generisanja sadržaja klijent ne može videti kod (skriptu) koji je generisao sadržaj koji vidite, već ima pristup samo čistom HTML kodu.

Broj sajtova koji su urađeni u PHP-u raste iz dana u dan, i sve je veći broj firmi koje žele da svoje stare sisteme prebace na PHP.

PHP radi pod Apache i IIS web serverima. Radi na Windows, Linux i Mac OS X operativnim sistemima. Ono što PHP stavlja još više ispred ostalih web skriptnih tehnologija je njegova podrška za baratanje širokom paletom baza podataka. Podržava sve popularnije baze podataka kao što su MySQL, PostgreSQL, dBBase, Oracle, ODBC...

Tvorac PHP-a, Rasmus Lefdorf je 1995 napisao jednostavan skup Perl skriptova da bi sebi olakšao rad. Nazvao ih je Personal Home Page tools. Ubrzo su ga kontaktirali drugi ljudi, nakon čega ga je pretvorio u pravi parser izvornog koda.

U današnje vreme se pred interpretere koda postavlja nekoliko teških zadataka, koji se u engleskom jeziku mogu nazvati 4S (Speed, Stability, Security, Scalability), to znači da se mora obezrediti brzina izvršavanja, da aplikacije budu stabilne i da se što brže i efikasnije otklanaju bagovi, uz to sistem mora biti stabilan i siguran na napade spolja i uz sve to da se obezbedi skalabilnost web aplikacije, odnosno da se aplikacija može nesmetano prilagođavati povećanju i smanjenju opterećenja.

PHP je stekao popularnost zbog svoje jednostavnosti i sintakse nasleđene iz programskog jezika C. Tokom vremena jezik se proširivao i sticao mogućnosti za objektno orijentisano programiranje, naročito od verzije 5.0. od kada i postaje zvanično i objektno orijentisan.

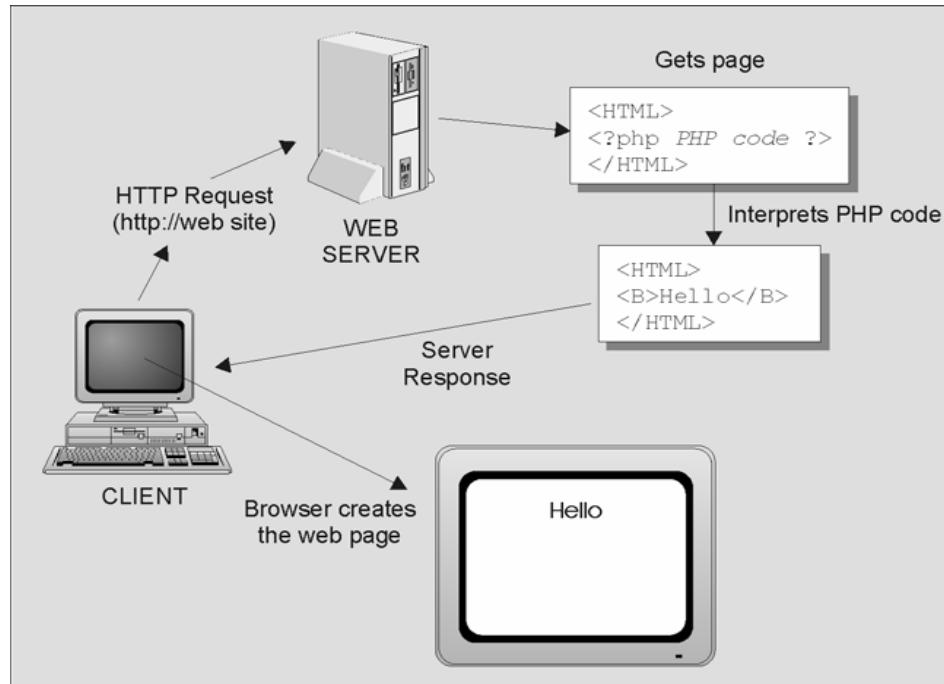
Neke od karakteristika i funkcija koje PHP obezbeđuje su:

- manipulacija nizovima, stringovima, datumima, preračunavanje datuma po različitim kalendarima...
- kontrola pravopisa
- pakovanje i raspakivanje fajlova (bzip i zip format)
- pristup COM objektima
- čitanje i kreiranje PDF fajlova
- pristup raznim bazama podataka direktno ili kroz ODBC interfejs
- manipulacija fajlovima i direktorijumima
- upravljanje XML dokumentima uz podršku za DOM model
- ubotreba .NET tehnologija
- ostvarivanje konekcija na FTP server i komunikacija FTP protokolom
- ostvarivanje HTTP konekcija na udaljene web servere
- komunikacija IMAP, POP3 i NNTP protokolima
- pristup Java objektima i pozivanje njihovih metoda
- komunikacija pomoću LDAP protokola
- kreiranje i čitanje Flash prezentacija (.swf fajlovi)
- implementacija SSL-a pomoću OpenSSL biblioteke
- upravljanje procesom štampanja
- upotreba socket-a, za direktnu komunikaciju na nekom TCP/UDP portu

Verzija 5 PHP-a donosi novu, snažniju objektno orijentisani implementaciju u kojoj su klase i objekti još korisniji. Sada PHP ima sve karakteristike koje se očekuju od objektno orijentisanog programskog jezika, gde je sada programerima olakšano da pišu kompleksne, modularne i zahtevne web aplikacije.

Sav PHP kod koji se piše mora se nalaziti unutar tagova `<?php i ?>` kako bi interpretator prepoznao da je to PHP kod i izvršio ga. Fajlovi sa .php ekstenzijom se "skeniraju" i obrađuju se kod. Promenljiva u PHP-u se obeležava sa \$.

Obrađivanje http zahteva uz pomoć PHP interpretatora može se videti na sledećoj slici:



Slika 6. Obrada zahteva preko PHP interpretatora

Primer klase koja je napisana u PHP-u:

```
<?php
class Osoba {
    protected $ime;
    protected $prezime;

    function __contructor($ime, $prezime){
        $this->ime = $ime;
        $this->prezime = $prezime;
    }
    function getIme(){
        return $this->ime;
    }
    function setIme($ime){
        $this->ime = $ime;
    }
}
?>
```

Ostale reprezentativne delove PHP i ostalog koda možete videti u prilogu u šestom poglavlju.

U sledećem tekstu biće objašnjeni najosnovniji koraci za manipulaciju sa PHP-om koji je objektno orijentisan jer je koncept objektnog programiranja korišćen i u razvoju aplikacije za adaptaciju sistema elektronskog obrazovanja.

3.2. Objekto orijentisano programiranje (OOP)

Ideja kod objektno orijentisanog (OO) koncepta programiranja jeste da se stvara modularan kod, koji će biti smešten u posebne PHP klase, a potom taj isti kod uvrstiti u normalnu PHP stranicu.

U kontekstu objektno orijentisanog softvera, objekt može biti skoro bilo koja stvar ili koncept - fizički objekt poput stola ili kupca, odnosno konceptualni objekt koji postoji samo u softveru, poput datoteke ili polja za unošenje teksta.

Objektno orijentisani softver se projektuje i gradi kao skup samostalnih objekata sa atributima i operacijama koji međusobnom interakcijom ispunjavaju zahteve projekata. Atributi su svojstva ili promenljive koje opisuju objekat. Operacije su metode, radnje ili funkcije koje objekat može da izvršava da bi izmenio sebe ili nešto u svom spoljašnjem okruženju.

Jedna od glavnih prednosti objektno orijentisanog programiranja je u tome da podržava kapsuliranje, poznato i kao skrivanje podataka. To znači da je pristup podacima unutar datog objekta moguć samo pomoću operacija tog objekta, koji se nazivaju njegovim interfejsom. Funkcionalnost objekta zavisi od podataka sa kojima objekat radi.

Objektno orijentisani programski jezik mora da podržava **polimorfizam**, što znači da se istoimena operacija može potpuno različito ponašati u različitim klasama. Na primer operacija "oceni" klase "Student" i klase "Profesor" mogu da budu različite. Polimorfizam predstavlja karakteristiku ponašanja, a ne karakteristiku objekata.

Još jedan bitan koncept u objektnom programiranju predstavlja i **nasleđivanje klasa**. Nasleđivanje obezbeđuje mogućnost definisanja hijerarhijskih odnosa između objekata, konkretno između klasa i podklasa. Podklasa nasleđuje atribute i operacije svoje natklase. Primer, ako izdvojimo klase Student i Profesor, oni imaju nešto zajedničko što i jednu i drugu klasu možemo predstaviti kao Osobu. S tim da

Studenta od Profesora može razlikovati broj indeksa, odnosno službeni broj, ali ipak imaju zajedničke atribute poput imena, prezimena itd.

3.3. JavaScript

JavaScript je jedan od vodećih jezika koji se upotrebljavaju za programiranje Web stranica. Pošto spada u grupu skript jezika (jezici koji se interpretiraju tokom izvršavanja), a ne u grupu kompajliranih jezika, veoma je pogodan za razvijanje internet aplikacija.

Kada je u pitanju kompajlirani jezik, programer piše izvorni kod koji predstavlja tekst pisan sličnim jezikom kao što je Engleski, a zatim ga propušta kroz kompjajler koji će ga prevesti na mašinski jezik. Kompajlirani kod se zatim izvršava. Na taj način kod je veoma brz, ali je program vezan za određeni tip operativnog sistema.

Kod skript jezika međutim, ne postoje kompjajleri, tako da kod uvek ostaje u obliku izvornog koda. Kada dođe vreme za izvršavanje ovog koda, softver za izvršavanje skriptova ga interpretira tokom izvršavanja, čime se izvršavanje usporava. Drugim rečima, računar jedan red izvršnog koda prevodi na mašinski jezik i izvršava, pa zatim drugi itd.

Skript kod je u većoj meri prenosiv, pošto može da se izvršava na svakom računaru koji ima odgovarajuću mašinu za skriptovanje. Drugim rečima, ukoliko imamo skript koji je pisan u JavaScriptu, on se može izvršavati na bilo kom računaru, jedino što je potrebno jeste da računar ima instaliran JavaScript mašinu za skriptovanje.

Neke od mogućnosti koje *JavaScript* pruža svojim korisnicima/programerima su:

- *HTML* dizajnerima pruža programerski alat
- reagovanje na određene događaje (klik mišem, učitavanje stranice, ...)
- čitanje i pisanje *HTML* elemenata
- provera podataka pre predaje serveru
- prepoznavanje *browser-a*
- pravljenje *cookies-a* itd.

Skript jezici kao što je JavaScript se dobro uklapaju sa HTML-om pošto su i sami HTML dokumenti čisti tekst i pošto se mogu prikazati na svakom računaru bez obzira na operativni sistem.

JavaScript je Netscape-ova cross-platforma, objektno-zasnovan skriptni jezik. JavaScript je mali, lagani jezik; nije pogodan kao samostojeći jezik, nego je stvoren kao dodatak za druge proizvode i aplikacije kao što su web browseri. Unutar host okruženja, JavaScript može se povezati sa objektima njegova okružja kako bi omogućio programsku kontrolu nad njima.

Osnove JavaScripta sadrže osnovni set objekata kao što su Array, Date , i Math, ali i osnovni set jezičkih elemenata kao što su operatori, kontrolne strukture i naredbe. Osnove JavaScripta mogu se proširiti za razne namene dodavajući na te osnove dodatne objekte; na primer:

- *Client-side JavaScript* proširuje osnove JavaScripta dodavajući objekte za kontrolu browsera (Navigator ili neki drugi web browser) i njihov Document Object Model (DOM). Na primer, client-side proširenja dopuštaju aplikaciji da smesti elemente u HTML obrazac i odgovori na ono što napravi korisnik kao npr. klikovi mišem, unos u obrazac, ili navigaciji stranicama.
- *Server-side JavaScript* proširuje osnove jezika dodavajući objekte koji se tiču rada JavaScripta na serveru. Na primer, server-side proširenja dopuštaju aplikaciji da komunicira sa relacijskim bazama podataka, omogućava kontinuitet informacije od jednog poziva do drugog poziva aplikacije ili da izvodi manipulacije datotekama na serveru.

Netscape je kreirao JavaScript i JavaScript se prvo bitno koristio u Netscape browserima. Pa ipak, Netscape sarađuje sa [ECMA](#) (European Computer Manufacturers Association) da stvori standardan, internacionalni programski jezik na osnovama JavaScript-a. ECMA je međunarodno udruženje za standarde u informacionim i komunikacionim sistemima. Ova standardizovana verzija JavaScripta, zvana ECMAScript, ponaša se na isti način u svim aplikacijama koje podržavaju ovaj standard.

3.4. AJAX

AJAX je skraćenica od Asynchronous JavaScript And XML. Kratko rečeno AJAX je zasnovan na JavaScript i HTTP zahtevima i njegova glavna osobina je da omogući da se na nekom sajtu šalje/prima sadržaj bez ponovnog učitavanja stranice.

Pre nego što se kreće sa izučavanjem AJAX-a potrebno je predhodno da se zna:

- HTML / XHTML
- JavaScript
- Server-side scripting

AJAX nije novi jezik programiranja, već tehnika za kreiranje boljih, bržih i interaktivnijih web aplikacija. Sa AJAX-om, vaš JavaScript može da komunicira direktno sa serverom, koristeći JavaScript XMLHttpRequest objekat. Pomoću ovog objekta, vaš JavaScript može da razmenjuje podatke sa web serverom, bez ponovnog učitavanja stranice.

AJAX koristi asinhroni transfer podataka (HTTP zahtevi) izmenu pretraživača i servera, omogućujući web aplikaciji da obrađuje samo one podatke koji su zaista potrebni umesto čitavih stranica.

AJAX je zasnovan na sledećim web standardima:

- JavaScript
- XML
- HTML
- CSS

Web standardi koji se koriste u AJAX-u su dobro definisani i podržani od svih velikih web browsera.

Web aplikacije imaju mnogo prednosti nad desktop aplikacijama, osnovna je da one lako mogu da dopru do više korisnika, lakše su za instaliranje i obezbenivanje podrške, i lakše za razvijanje.

Sa AJAX-om, internet aplikacije mogu da budu bogatije i brže za korišćenje nego što su bile do sada.

U "klasičnom" programiranju, ako želite da dobijete bilo kakvu informaciju iz baze podataka ili fajla nekog koji se nalazi na serveru, ili pošaljete serveru informacije o korisniku, vi treba da napravite HTML formu i preko GET ili POST metode dobijete ili pošaljete podatke serveru. To bi značilo da nakon što se pošalje/dobije informacija od servera da se onda učita opet cela strana kako bi rezultati bili vidljivi i samim tim pošto server vraća novu stranicu svaki put kada korisnik pošalje neke podatke, tradicionalne web aplikacije znaju da budu spore.

Pomoću AJAX-a JavaScript komunicira direktno sa serverom, kroz JavaScript XMLHttpRequest objekat. Pomoću HTTP zahteva, web stranica šalje zahtev serveru, i dobije odgovor od njega - bez ponovnog učitavanja web stranice, i korisnik gotovo da i neće primetiti da je skript poslao zahtev za stranicom, ili da je u pozadini poslao podatke na server.

Koristeći XMLHttpRequest objekat, web programer može da ažurira stranicu sa podacima sa servera nakon što se stranica učitala sto je opet jedan veliki plus!

Kada se radi sa XMLHttpRequest-om treba voditi racuna i koji browser je u pitanju. Veliki broj browsera podržava XMLHttpRequest (Internet Explorer, sve verzije novije od 5.0 verzije, Firefox, Opera (što bi ukratko značilo skoro svi)) ali je malo drugačije slanje XMLHttpRequest-a svakom od njih.

3.5. Jquery

Jquery predstavlja brzu i efikasnu JavaScript biblioteku koja pojednostavljuje izradu HTML dokumenata, uspešno upravlja događajima, animacijom i interakcijom sa AJAX-om za brzi razvoj veb aplikacija. Jquery je smišljen da promeni način na koji se koristi JavaScript. Jquery je najpopularnija JavaScript biblioteka koja je u upotrebi danas.

Jquery je slobodan, open-source softver, licencirana pod MIT licencom i GNU General Public License.

Jquery sintaksa je osmišljena kako bi se programeri lakše kretali kroz dokument, kako bi se lakše manipulisalo sa elementima, kreirale animacije i razvijale AJAX aplikacije. Jquery takođe pruža mogućnost programerima da rade dodatke u okviru nadogradnje osnovne biblioteke. Korišćenjem ovim objekata programeri su u mogućnosti da kreiraju veoma kompleksne i dinamičke veb aplikacije.

Jquery sadrži sledeće funkcije:

- Izbor DOM elemenata
- DOM prelaz i izmene (uz podršku za CSS)
- Događaji
- CSS manipulacije
- Efekti i animacije
- AJAX
- Proširenje preko raznih dodataka (plugins)
- Utilities
- Cross-browser podrška

Zbog arhitekture Jquery-a, programeri mogu veoma lako kreirati svoje dodatke i tako proširiti funkcionalnost samog Jquery-a. Trenutno postoje na hiljade Jquery dodataka koje pokrivaju širok spektar funkcionalnosti kao što su AJAX pomagači, Web servisi, dodaci za tabelarni prikaz podataka, dinamične liste, HTML i XHTML alati, drag and drop funkcionalnosti, događaji, kuki itd.

Na zvaničnom sajtu postoji veoma dobra i obimna dokumentacija za proučavanje i mogućnosti koje pruža Jquery sa konkretnim primerima gde u realnom vremenu možete videti primer koji želite. Više informacija možete pročitati na zvaničnom sajtu: <http://www.jquery.com>.

3.6. Jqgrid

Jqgrid je ustvari JavaScript biblioteka koju je moguće koristiti uz AJAX i JavaScript kontrole koje obezbeđuju rešenje za predstavljanje i manipulaciju tabela podataka na internetu.

Jqgrid radi sa velikim brojem tehnologijama u pozadini na serverskoj strani kao što su PHP, ASP, Java Servlets, JSP, ColdFusion, i Perl.

Ideja Jqgrid jeste da na jednosten, brz i efikasan način predstavi tabelarno podatke iz baze podataka. Jedna od velikih prednosti Jqgrida jeste to što je nezavisan od server-side tehnologije kao i baze podataka u pozadini.

Jqgrid radi veoma efikasno sa velikim brojem podataka gde je u stanju da za veoma kratko vreme obradi 300 000 do 400 000 rekorda u bazi brzo i efikasno.

Jqgrid koristi Jquery JavaScript biblioteke i definisan je kao dodatak za Jquery. Jqgrid je razvijen od strane *Tony Tomov-a* iz firme *Trirand Inc* sa sedištem u Sofiji.

Više informacija o Jqgridu možete naći na internet adresi:

<http://www.trirand.com/blog/>, gde postoji kompletan dokumentacija za korišćenje i instalaciju, kao i veliki broj primera u kojima se može koristiti. Demo aplikacije urađene uz pomoć Jqgrida možete videti na sledećoj internet adresi <http://www.trirand.com/blog/jqgrid/jqgrid.html>

Postoje više verzija Jqgrida, aktuelna je poslednja 3.8 verzija. Postoje više paketa, osnovni paket je besplatan i ideja je bila da se doprinese open source zajednici, dok postoje i komercijalizovane verzije.

3.7. HTML i CSS

HTML, HyperText Markup Language, je standarizovani jezik koji se koristi pri strukturiranju tekstova, medija i ugrađenih objekata u web stranice i elektronsku poštu. Kao modifikovanu i pojednostavljenu verziju SGML jezika, HTML standarizuje i održava World Wide Web Consortium (W3C). Originalnu verziju HTML-a kreirao je Tim Berners-Lee, a prva zvanična verzija je izšla juna 1993 godine.

HTML jezik je sačinjen od običnog teksta i tagova. Iako se HTML znakovi (tags) često nazivaju kodom, tehnički HTML nije kod jer kompjuterski kod su instrukcije koje od računara traže da izvrši određenu operaciju. Svrha HTML znakova je da se struktura dokumenta "označi" tako da bi korisnikov agent (user agent), tj. internet pretraživač mogao da prepozna strukturu dokumenta i ispravno je prikaže u prozoru internet pretraživača kojeg korisnik koristi.

Za dodatno modifikovanje web stranica, do sada su izšli neki dodaci:

- CSS ili Cascading Style Sheets za izgled i položaj prezentacije
- Skriptni jezici (Javascript, VBScript) za omogućavanje dinamičnosti i interaktivnosti na web prezentacijama,
- DOM ili Document Object Model, koji označava vezu između skripte i elementa na stranici

Gornji dodaci, u paketu sa HTML jezikom, ponekad se nazivaju DHTML ili Dinamični HTML.

Cascading Style Sheets (CSS) je stilski jezik koji se koristi za opisivanje semantike prezentacije (izgleda i oblika). Najčešća primena je u stilu veb stranice pisane HTML-u ili XHTML-u ali CSS se može primeniti i na bilo koju vrstu HTML dokumenata. CSS je prvenstveno dizajniran da omogući razdvajanje sadržaja dokumenata iz dokumenata prezentaciji, uključujući elemente kao što su raspored, fontovi, boje itd. CSS je W3C standard.

Više informacija o CSS kao i HTML-u možete pogledati na sledećoj internet adresi:
<http://www.w3schools.com/css/> i <http://www.w3schools.com/html/>.

3.8. XML

XML je akronim od eXtensible Markup Language, a konzorcijum W3C prihvatio ga je kao standard za označavanje (engl.*markup*) dokumenata. On definiše opštu sintaksu za označavanje podataka jednostavnim, svima razumljivim oznakama.

XML je metajezik za označavanje tekstualnih dokumenata. Podaci se smeštaju u XML dokumente u obliku znakovnih nizova (engl.*strings*), između tekstualnih oznaka koje ih opisuju. Osnovne jedinice podataka i oznaka u XML-u nazivaju se *elementi*. XML specifikacija precizno definiše sintaksu koje se morate pridržavati pri pisanju oznaka: kako su elementi razgraničeni oznakama, kako oznaka izgleda, kakva imena elementi mogu imati, gde se stavljuju atributi itd.

XML je jezik za označavanje i ništa više od toga.

Pre svega, *XML nije programski jezik*. Ne postoji kompjajler XML-a koji čita XML datoteke i daje izvršni kod. Eventualno biste mogli definisati skript jezik koji koristi format XML-a kao matični, a interpretira ga neki binarni program, ali bi čak i ta primena bila neobična. XML se može upotrebiti kao format naredaba u programima koji nešto rade, kao što i tradicionalni programi mogu čitati tekstualne konfiguracijske datoteke i preuzimati razne akcije na osnovu pročitanog.

Primer XML 1:

```
<?xml version="1.0"?>
<student brojIndeksa="474/05">
<ime>Aleksandar</ime>
<prezime>Nastevski</prezime>
<jmbg>1109986970024</jmbg>
<adresa>Braće Jerkovića 58/32, 11000 Beograd</adresa>
<telefon>0653141441</telefon>
<godinaStudija>apsolvent</godinaStudija>
</student>
```

XML je *izuzetno fleksibilan* format podataka. Teorijski se svi podaci koji se uopšte mogu smestiti u računar, mogu snimiti u formatu XML.

3.9. Android

Android je mobilni operativni sistem prvo bitno razvijen od strane Android kompanije, koju je kupio Gugl 2005. godine. Nakon kupovine, Gugl je zatražio stvaranje Open Handset Alliance (OHA) udruženja, koja je sada angažovana u pružanju podrške i dalji razvoj sistema. Android je zasnovan na izmenjenoj verziji Linux jezgra. Po broju prodatih pametnih telefona, zasnovanih na Android OS tokom 2012 godine, Android se nalazi na prvom mestu sa udelom od 47 procenata.

Početak Android distribucije najavljen je 5. novembra 2007. godine zajedno sa osnivanjem Open Handset Alliance, udruženja 78 kompanija posvećene unapređenju otvorenih standarda za mobilne uređaje. Gugl je veći deo Android koda objavio pod Apache licencom, licencom slobodnog softvera i licencom otvorenog koda.

Android OS se sastoji od 12 miliona linija koda, uključujući 3 miliona linija XML koda, 2,8 miliona linija C koda, 2,1 miliona linija Java koda i 1,75 miliona linija C++ koda..

Android predstavlja softverski stek namenjen mobilnim telefonima koji u sebi uključuje operativni sistem, posrednički softver kao i ključne aplikacije za normalno funkcionisanje. Android SDK obezbeđuje alatke i API koji su neophodni za dalji razvoj aplikacija na Android platformi koristeći JAVA programski jezik.

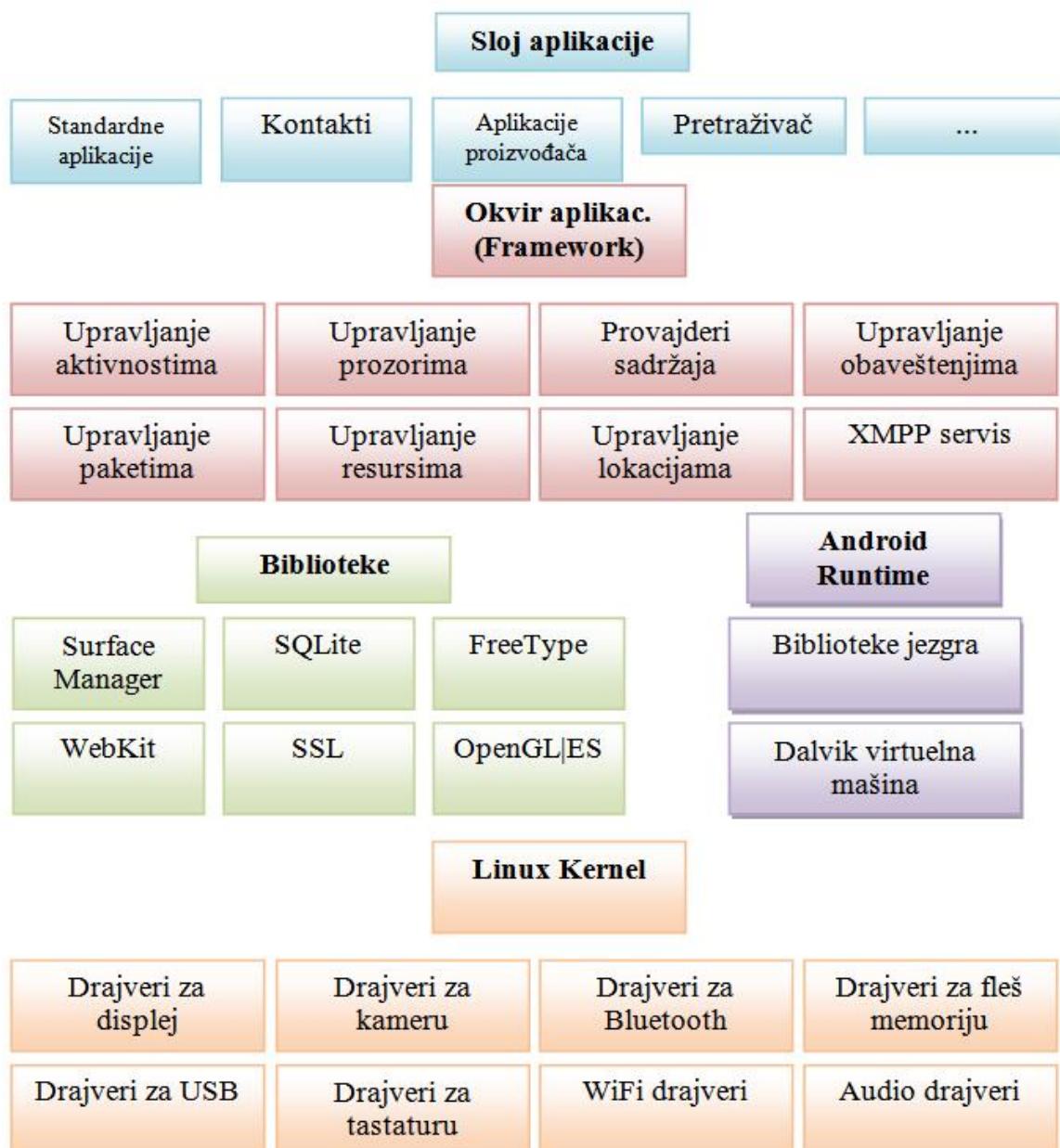
Prema istraživanjima iz poslednjeg kvartala 2012. godine Android ima 53% tržišnog udela što ga svrstava na prvom mestu na listi najpopularnijih platformi zamobilne telefone.

Shodno svojoj politici OHA ističe sledeće karakteristike Android programskog okruženja:

- **otvorenost** - programeru omogućava potpunu slobodu u razvoju novih i već postojećih aplikacija, a proizvođaču uređaja slobodno korišćenje i prilagođavanje platforme bez plaćanja autorskih prava;
- **sve aplikacije su ravnopravne** - što znači da ne postoji razlika između osnovnih jezgarnih aplikacija uređaja i dodatnih aplikacija. Svim aplikacijama omogućen je ravnopravni pristup resursima pokretnog uređaja što daje mogućnost prilagođavanja uređaja specifičnim potrebama individualnog korisnika;
- **automatsko upravljanje životnim ciklusom aplikacije** - omogućava nadzor pokretanja i izvršavanja aplikacija na sistemskom nivou optimizovanim korišćenjem memorije i snage uređaja. Krajnji korisnik više ne brine o gašenju određenih aplikacija pre pokretanja drugih;
- **rušenje granica "klasičnih" aplikacija** - mogućnost razvoja novih i inovativnih aplikacija zasnovanih na međusobnoj kolaboraciji tehnologija;
- **brz i jednostavan razvoj aplikacija** - omogućen je bogatom bazom korisnih programskih biblioteka (eng. libraries) i alata za izradu aplikacija;
- **visokokvalitetni grafički prikaz i zvuk** - podržana 2D vektorska i 3D OpenGL (engl. Open Graphics Library) grafika, te ugrađeni kodeci svih često korišćenih audio i video formata;
- **kompatibilnost sa većinom sadašnjeg i budućeg hardvera** - uključuje prenosivost Androidovih aplikacija na ARM, k86 i ostale arhitekture, te prilagodljivost sistema ulaznim i izlaznim komponentama.

Arhitektura Android sistema bazira se na Linux-u 2,6 jezgru (eng. kernel) koja se koristi kao sloj apstrakcije hardvera (HAL, eng. Hardvare Abstraction Laier). Razlog za korišćenje jezgra operativnog sistema Linux je dokazana pogonska podrška (eng. driver model), mogućnost upravljanja memorijom i procesima, sigurnosni model, mrežni sistem, te dokazana robusnost, konstantni razvoj i unapređivanje sistema.

Slika 7 prikazuje arhitekturu Android sistema:



Slika 7. Arhitektura Android platforme

3.10. Moodle LMS

Moodle (Modular Object Oriented Developmental Learning Environment) je sistem za pravljenje kurseva tj. softverski paket koji je dizajniran da pomogne predavačima da kreiraju kvalitetne "online" kurseve i da usmeravaju rezultate svojih učenika. Takvi elektronski sistemi za učenje ponekad se nazivaju „Sistem učenja na daljinu“, „Uslovi virtualnog učenja“ i „Menadžer sistema sadržajnog učenja“. Studentima je potreban jedino „pretraživač“ (npr. Firefox, IE) da bi učestvovali u kursu Moodle.

Moodle je „open source“ softver, što znači da slobodno možete da ga skinete sa interneta, koristite, modifikujete, čak i distribuirate (pod uslovima GNU - Opšte javne licence). Moodle se bez modifikacija primenjuje na Unix, Linux, Windows, MAC OS X, Netware i na bilo kojim drugim sistemima koji podržava PHP, uključujući većinu internet provajdera. Podaci se drže u samo jednoj bazi podataka: MySQL i PostgreSQL su najbolje opremljeni, ali se takođe može koristiti uz Oracle, Access, Interbase, ODBC i druge.

Osnivač i kreator Moodle-a je Australijanac Martin Dougiamas koji je predavač i naučnik što predstavlja veoma dobru kombinaciju za razvoj ovakog tipa softvera.

Moodle ima preko 70 jezičkih paketa, uključujući: arapski, katalan, kineski (pojednostavljen i tradicionalan), češki, danski, holandski engleski(britansku i američku verziju), finski, francuski (francuski i kanadsku verziju), nemački, grčki, mađarski, indonežanski, italijanski, japanski, norvežki, poljski, portugalski, brazilski, rumunski, ruski, slovački, španski, švedski, tai i turski. Od skora je dostupna i verzija prilagodjena za srpski jezik (ćirilična i latinična).

Dizajniranje i razvoj Moodle-a, vođeni su posebnom filozofijom učenja, načinom razmišljanja za koji možete pomisliti da se, ukratko rečeno, definiše kao "pedagogija društvene izgradnje".

U budućnosti, kako se tehnička infrastruktura Moodle-a bude stabilizovala, dalji napredak u pedagoškoj potpori biće glavni pravac njegovog razvoja.

Neke od karakteristika Moodle-a:

- Moodle projekat je aktivan i stalno se menja i razvija;
- Promoviše društvenu konstruktivnu pedagogiju;
- Pogodan za potpuno samostalne online kurseve ili kao dodatak kursevima Face-to-Face (Lice u lice);
- Jednostavan, malih hardverskih i softverskih zahteva, efikasan i pristupačan kroz pretraživač;
- Jednostavan za instalaciju na svakoj platformi koja ga podržava;
- Zahteva samo jednu bazu podataka i dozvoljava njen deljenje sa drugim sistemima;
- Spisak kurseva na web sajtu opisuje sve postojeće kurseve uključujući prava i pravila pristupa;
- Kursevi se mogu kategorizovati i pretraživati i itd.

Sadržaj glavnog Moodle direktorijuma:

- config.php – ovde se nalaze opšta podešavanja Moodle-a
- version.php – određuje verziju moodle-a
- indeks.php – početna stranica
- admin/ - ovde se nalazi kod za administraciju celog sajta
- auth/ - ovde je smešten kod za autorizaciju (autentifikaciju) korisnika
- blocks/ - blokovi koji se prikazuju sa strane
- course/ - kod za kreiranje, promenu i prikaz kurseva;
- doc/ - pomoćna dokumentacija Moodle-a;
- files/ - kod za promenu i prikaz sačuvanih datoteka;
- lang/ - podrška za različite jezike;
- lib/ - biblioteka (engl. library) osnovnog koda;
- login/ - kod za kreiranje i uključivanje novih korisničkih naloga;

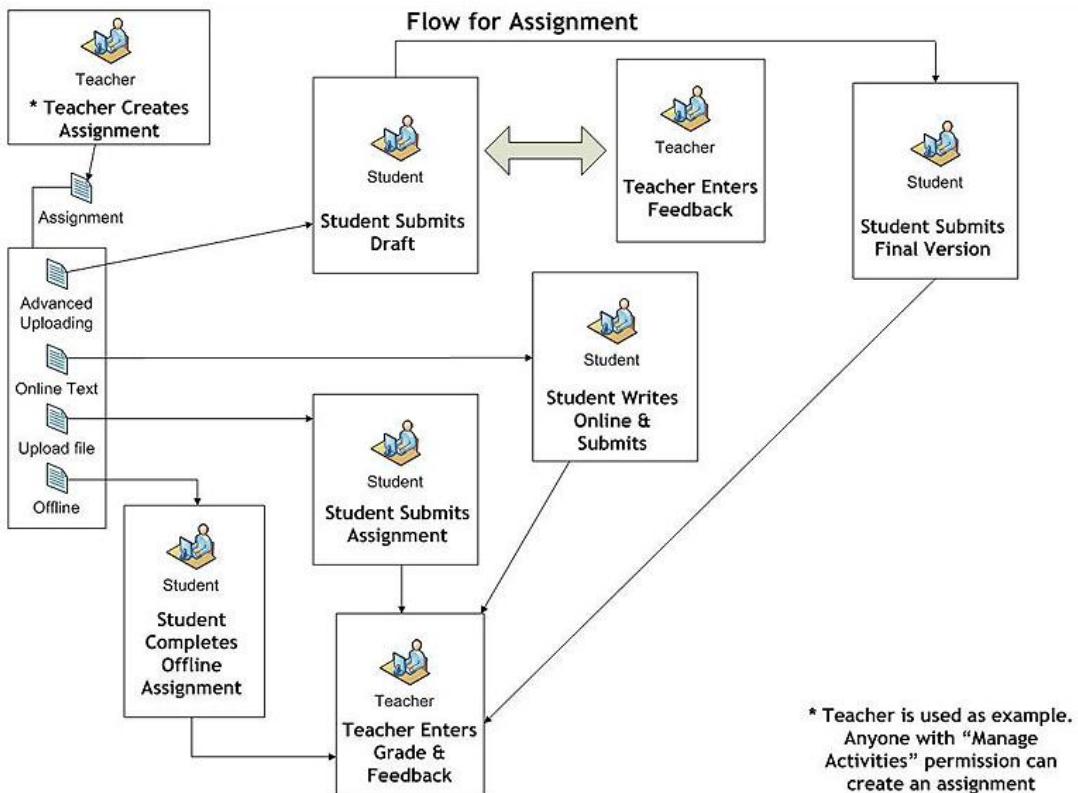
- mod/ - moduli Moodle kurseva;
- pix/ - osnovna grafika stranice, ovde su smeštene slike;
- theme/ - teme za promenu izgleda stranice;
- user/ - kod za promenu i prikaz podataka o korisnicima;

Kao dodatak prilikom razvoja aplikacije za adaptaciju sistema elektronskog obrazovanja za Moodle LMS, dodat je u okviru Moodle direktorijuma direktorijum mehanizmi/ u kome je smeštena aplikacija za adaptaciju, dok su modul i block adaptivnog režima dodati u okviru već postojećih direktorijuma.

Moodle je dizajniran tako da bude fleksibilan, stabilan, kompatibilan i lako izmenljiv, kreiran je korišćenjem serverskog programskog jezika PHP, koji mu time obezbeđuje nezavosnost od platforme na koju se izvršava. Moodle je sam po sebi pravljen na visokom nivou modularnosti i koristi veliki broj tehnologija kao što su: deljenje biblioteka, apstrakcije i kaskadni stilovi za definisanje izgleda, koji omogućavaju proširivost postojećeg sistema.

The screenshot shows the Moodle LMS interface. At the top, it says 'Prijavljeni ste kao Aleksandar Nastevski (Odjava)'. The main title is 'Placement testovi'. On the left, there's a sidebar with 'Prisutni korisnici' (Učesnici), 'Aktivnosti' (Forumi, Skype, Testovi), 'Pretraga foruma' (Dalje), 'Napredno pretraživanje', and 'Administracija' (Uključi uređivanje, Podešavanja, Dodeli uloge, Ocene, Grupe). The central area is titled 'Pregled teme' and contains a section for 'Placement testovi' with five topics: 1. HTML i CSS, 2. Javascript i Jquery, 3. PHP, 4. PHP XML, and 5. PHP - Objektno orijentiran koncept programiranja. On the right, there are sections for 'Adaptirani resursi' (Java Script - praktikum, Zadatak 1 (za studente osnovnih studija), Principi razvoja aplikacija u Internet okruženju, PHP skripta), 'Uključi adaptivni režim' (Pričak preporučenih resursa na osnovu adaptacije), 'Najnovije vesti' (Dodajte novu temu... (Trenutno nema novosti)), and 'Predstojeći događaji' (Nema predstojećih događaja).

Slika 8. Izgled kursa u Moodle LMS-u



Slika 9. Tok izrade, predaje i ocenjivanja domaćeg zadatka u Moodle LMS

3.11. Model View Controller (MVC)

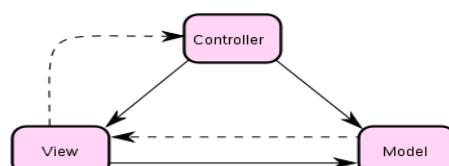
Prilikom razvoja mehanizma za adaptaciju sistema elektronskog obrazovanja korišćen je patern Model View Controller (MVC) kako bi aplikacija bila stabilnija, sigurnija i veoma jednostavna za nadogradnju i kasnije proširivanje.

MVC predstavlja uzor za arhitekturu softvera (Architectural pattern) koji se koristi u programskom inžinjerstvu. MVC uzor organizuje aplikaciju u 3 odvojena modula. Uspešna upotreba uzora razdvaja poslovnu logiku od korisničkog interfejsa što se kao rezultat dobija da se program može nezavisno modifikovati, bilo da je to vizuelni efekat web aplikacije ili pak poslovna logika, bez uticaja pomenute modifikacije na ostale delove aplikacije.

Model-view-controller istovremeno predstavlja i arhitektonski i dizajn uzor, zavisno od toga gde se koristi.

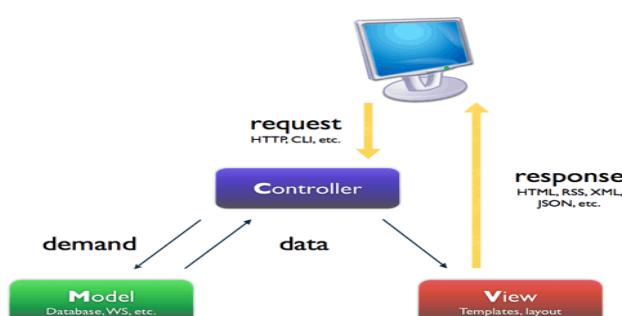
MVC uzor se često koristi prilikom razvoja web aplikacija gde poglede predstavljaju stvarne HTML ili XHTML stranice, dok Controller je kod koji prikuplja podatke i generiše dinamički sadržaj unutar HTML ili XHTML stranica. Na kraju model predstavlja stvarni sadržaj koji se čuva u nekoj bazi podataka ili XML dokumentu.

Osnovnu arhitekturu MVC uzora možete videti na sledećoj slici:



Slika 10. Koncept Model-view-controller

Model MVC uzora koji je korišćen u razvoju aplikacije izgleda kao na sledećoj slici:



Slika 11. Prošireni model MVC uzora

4. Razvoj servisa za adaptaciju sistema elektronskog obrazovanja

Servisi za adaptaciju sistema elektronskog obrazovanja su veoma kompleksni što sa teoretskog stanovišta isto tako i sa praktičnog. Međutim, ako se određene ključne celine dobro apstrahuju, nakon dobre analize, može se doći do rešenja koje je sastavljeno od velikog broja malih delova, gde svi ti mali delovi čine jedno kompleksnu celinu koja funkcioniše kao homogena celina.

U ovom poglavlju na nrednih stranama biće objašnjeni konkretni koraci koji su korišćeni prilikom razvoja rešenja i kao i same aplikacije. Na početku će biti opisani korisnički zahtevi aplikacije, u drugoj fazi biće urađena analiza korisničkih zahteva, kao i nakon toga prikazati detaljan opis rešenja, prikazati rezultate istraživanja koje je sprovedeno na eksperimentalnim grupama, kao i detaljno objašnjeni svi slučajevi korišćenja aplikacije.

Servis za adaptaciju sistema elektronskog obrazovanja kao kompleksna celina sastoji se iz sledećih mini aplikacija koje nezavisno funkcionišu kao jedinstvena celina:

1. Aplikacija za adaptaciju sadržaja (Moodle Plugin) - namenjen profesorima da rade adaptaciju sadržaja koji kreiraju na Moodle LMS sistemu
2. Mechanizam za analizu podataka i zaključivanje – Nezavisna PHP aplikacija
3. Moodle blok za prikaz top adaptiranog sadržaja
4. Moodle modul za prikaz adaptiranog sadržaja na nivou kompletног курса
5. Moodle modul za realizaciju Personalizovane table studenta, odnosno grupe ako se radi adaptacija na nivou grupe
6. Web servis za slanje SMS poruka putem interneta
7. Android aplikacija koja komunicira sa web servisom i prosleđuje SMS poruke kranjim korisnicima, odnosno studentima

4.1. Opis korisničkog zahteva aplikacije

Potrebno je da se razviju servisi za adaptaciju sistema elektronskog obrazovanja, koji bi omogućili adaptaciju sadržaja nekog sistema za elektronsko obrazovanje, to je Moodle LMS.

Ideja je da se na osnovu nekih ulaznih podataka kao što su popunjavanje ankete studenata, praćenje logova studenata, praćenje ostvarenih rezultata studenata na određenim specijalizovanim testovima, kao i rezultata na nekim od predhodnih kurseva koje je student pohadiao, adaptira određeni sadržaj i da se prilagođeni sadržaj prikaže studentu. Tako bi se studentima prikazivao sadržaj koji odgovara njihvim potrebama, očekivanjima, ciljevima, ograničenjima itd.

Potreno je da se kreira generička aplikacija koja će ostaviti prostor profesoru (ekspretu iz određene oblasti obrazovanja) da na svoj način može prilagođavati sadržaj oderđenoj grupi studenata i tako uradi adaptaciju sistema elektronskog obrazovanja. Ovo se drugim rečima može reći da se na različitim kursevima, sadržaj može različito prilagođavati u zavisnosti od potreba i specifičnosti oblasti obrazovanja gde se adaptacija primenjuje, gde ekspert zaključuje uz pomoć mehanizma za adaptaciju kako će adaptirati određeni sadržaj.

Jedan od ključnih zahteva je da se ne menja jezgro Moodle LMS sistema, već da se uradi aplikacija koja će biti nezavisna od samog LMS sistema.

Pored navedenih zahteva, potrebno je da se napravi mehanizam koji bi automatski omogućio kreiranje grupa studenata u okviru kursa. Automatsko razmeštanje studenata po grupama radilo bi se na osnovu podataka koje se prikupe preko mehanizama za adaptaciju sadražaja, dodatno obrade i donese zaključak koje studente u kojim grupama treba rasporediti.

Potreno je da se unapredi i poboljša informisanja studenata o novostima i dešavanjima na kursu putem SMS poruka. U tom smislu potrebno je da se razvije servis koji će omogućiti jeftino slanje SMS poruka sa web-a na telefon, potrebno je da se servis integriše sa Moodle LMS sistemom, ali da ipak servis ostane nezavistan i da se lagano može primeniti i na druge online sisteme.

4.2. Analiza korisničkih zahteva

Nakon detaljno definisanih korisničkih zahteva krenulo se u analizu istih, gde se došlo do zaključka da je potrebno da se kreira aplikacija koja će biti generička i koja će biti nezavisna od samog LMS sistema, u ovom slučaju Moodle LMS sistema.

Za realizaciju adaptivnog mehanizma postojala su dva moguća načina na koja su se mogli realizovati pomenuti korisnički zahtevi.

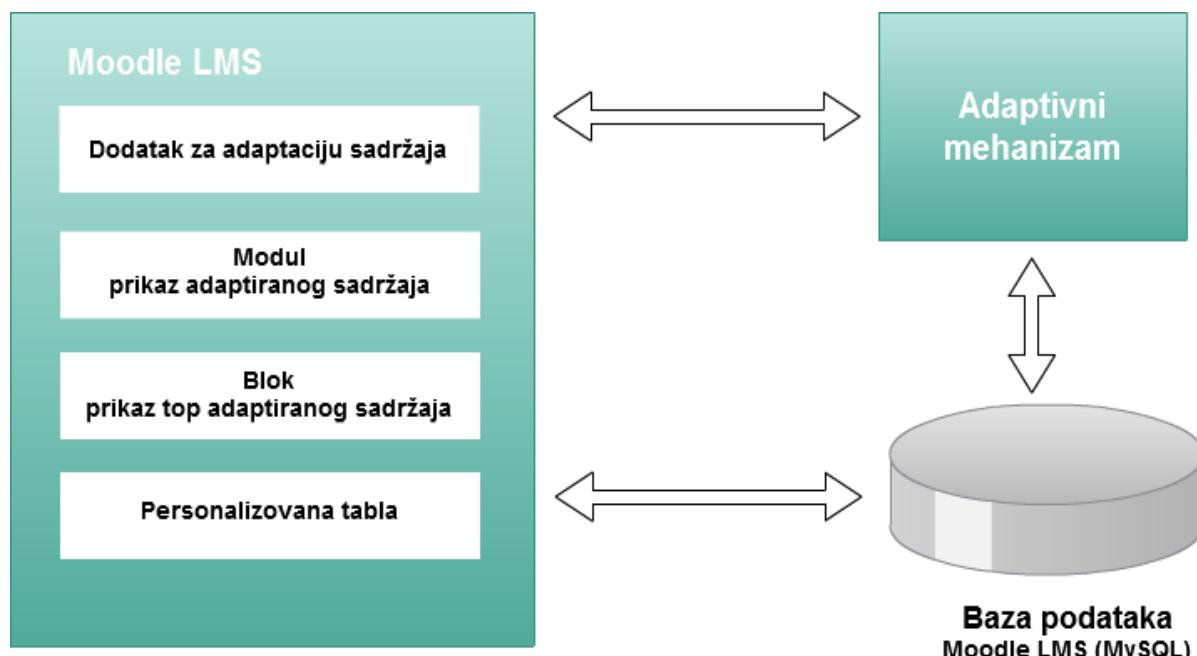
1. Da se dogradi već postojeći LMS sistem, da se promene njegove već postojeće funkcionalnosti i tako obezvbedi adaptivnost koja se zahteva. Ovo rešenje nije bilo pogodno za implementaciju, zbog toga što bi se narušilo samo jezgro LMS sistema i nakon promene samog jezgra dovelo bi do pitanja da li bi mogao LMS sistem da se bez grešaka dogradi u Update-uje sa nekim novijim verzijama, što samo po sebi na duže staze ne predstavlja dobro rešenje.
2. Drugi način, koji je i implementiran, jeste da se napiše zasebna PHP aplikacija, koja će preko adaptera koji bi se realizovao za konketan LMS sistem povezati sa tim LMS sistemom. Adapter koji predstavlja Mehanizam za adaptaciju biće povezan na bazu podataka koju koristi LMS sistem i biće realizovan kao dodatak koji obezbeđuje konkretan LMS sistem, u vidu Plugin-a, bloka, modula.

Ovako rešenje će biti nezavisno od samog LMS sistema, gde bi se vršila adaptacija studenata, resursa, sadržaja, grupe itd, dok za sam prikaz adaptiranih objekata potrebno je da se kreira plugin, modul i blok koji su napisani za Moodle LMS sistem i koji bi nesmetano i bez ikakvih promena samog jezgra mogli efiksano funkcionišati. Pomenuti plugin, modul kao i blok koji će biti kreirani i prilagođeni Moodle LMS sistemu, urađeni su na način na koji zahteva Moodle LMS.

Za realizaciju web servisa za slanje sms poruka, biće razvijana kompletno odvojena PHP aplikacija koje će predstavljati web servis, kao i androird aplikacija koja bi služila za komunikaciju sa web servisom za slanje sms poruka.

4.3. Mehanizam za adaptaciju sistema elektronskog obrazovanja

Mehanizam za adaptaciju sistema elektronskog obrazovanja je zasebna PHP aplikacija koja je povezana sa Moodle LMS sistemom i bazom podataka. Na sledećoj slici prikazana je arhitektura dobijeog rešenja za adaptaciju sistema elektronskog obrazovanja:



Slika 12. Arhitektura rešenja za razvoj mehanizma za adaptaciju

Na slici se može videti da je mehanizam za adaptaciju izdvojen iz samog Moodle LMS sistema, gde komunicira sa samim LMS sistemom preko plugin-a, modula i bloka koji su kreirani za Moodle LMS sistem i zajedničke baze podataka. Na ovaj način ostavljen je prostor za kasniju nadogradnju mehanizma za adaptaciju kao i povezivanje samog mehanizma sa nekim drugim LMS sistemima.

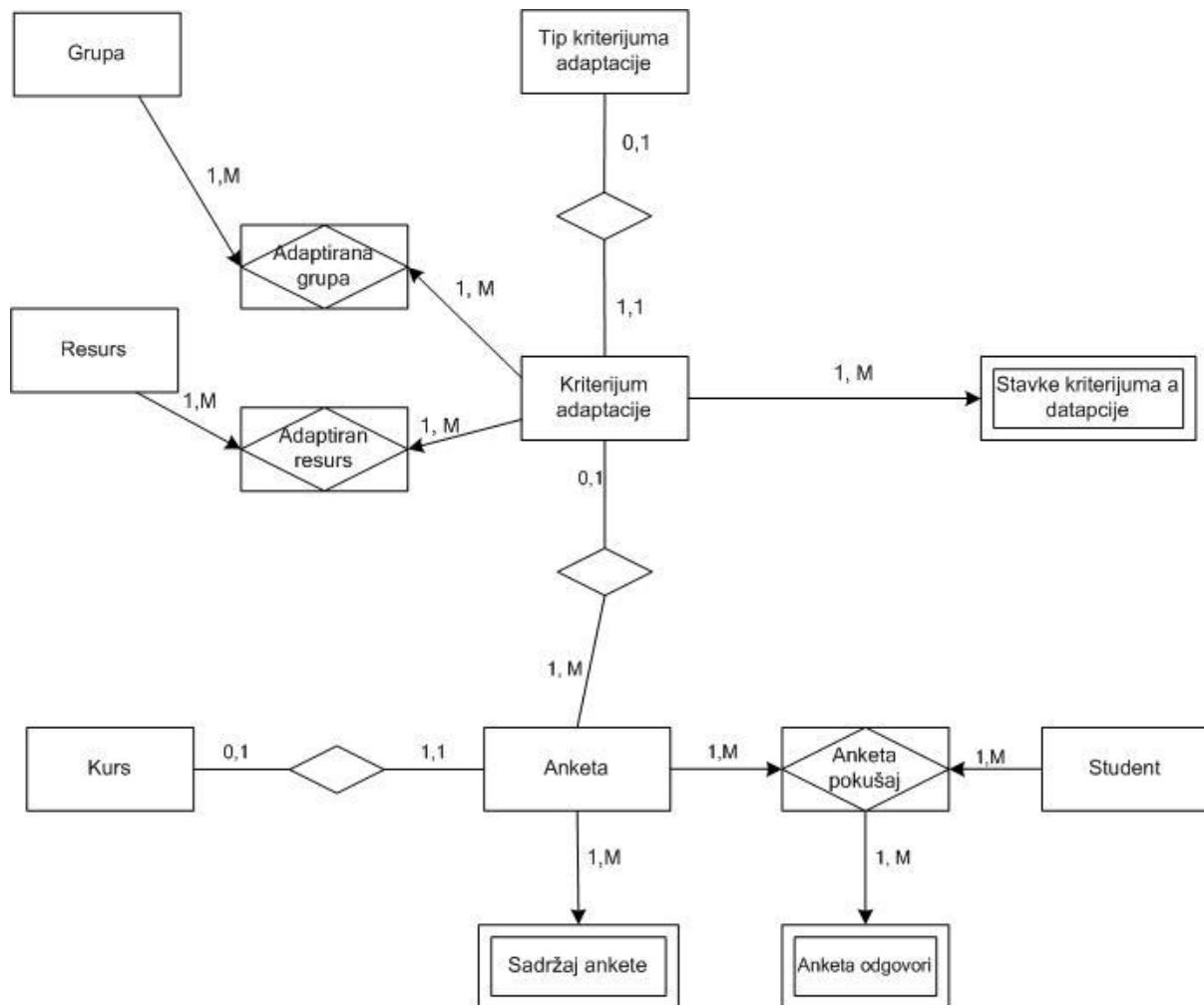
U ovom slučaju adaptivni mehanizam ima zadatku da adaptira objekte iz LMS sistema i bez ikakvih promena samog jezgra LMS sistema prikaže adaptirane objekte krajnjem korisniku, u ovom slučaju studentu.

Na osnovu predhodnog može se grubo reći da se treba razviti sledeće:

- PHP aplikacija kao nezavisan adaptivni mehanizam,
- Dodatak (Plugin) napisan za Moodle LMS sistem koji će omogućiti profesorima da urade adaptaciju sadržaja
- Modul koji je napisan za Moodle LMS sistem u kome će se prikazivati adaptirani objekti iz LMS sistema,
- Blok koji je napisan za Moodle LMS sistem u kome će se isto kao i u modulu prikazivati na nivou kursa adaptirani objekti,
- Potrebno je da se proširi osnovna baza podataka Moodle LMS sistema sa dodatnim odgovorajućim tabelama koje će kasnije biti detaljino objašnjene

Plugin, modul i blok koji će biti napisani za Moodle LMS predstavljaju vezu između adaptivnog mehanizma i samog LMS sistema.

4.3.1. Prošireni model objekti-veze (PMOV)



Slika 13. Prošireni model objekti-veze mehanizma za adaptaciju

4.3.2. Analiza objekata koje treba adaptirati

Prvi korak u razvoju servisa za adaptaciju jeste da se analizira LMS sistem za koji se radi adaptacija. Nakon analize strukture i arhitekture Moodle LMS sistema urađen je popis objekata koje treba adaptirati.

Kako bi mehanizam za adaptaciju ispravno funkcionisao potrebno je predhodno adaptirati sledeće objekte:

- Kurs,
- Grupa,
- Resurs,
- Aktivnost i
- Korisnik, odnosno student.

Kurs može imati više grupa u kojima su smešteni studenti, potrebno je razviti mehanizam koji će adaptirati kurs korisnicima.

Grupa predstavlja skup više studenata na jednom mestu koje karakterišu neke zajedničke osobine. Adaptacija sadržaja se radi na nivou grupe, studenti u istoj grupi imaće mogućnost pristupa istim materijalima. Adaptacijom na nivou grupe se otišlo korak dalje, gde zavisno od afiniteta, karakteristika, ciljeva i mogućnosti grupe, prikazuje se adaptiran sadržaj za konkretnu grupu.

Resurs je jak objekat u Moodle LMS sistemu i pod resursom razlikujemo nekoliko različitih tipova kao što su: natpis, tekstualna stranica, web stranica, link ka datoteci ili web sajtu, direktorijum itd. Za sve navedene tipove resursa potrebno je da se uradi adaptacija u kontekstu da se svim resursima dodele neke osobine, karakteristike, ciljevi, ograničenja kako bi resurs mogao da postane kandidat za ulazak u servis za adaptaciju sadržaja.

Aktivnost je takođe jak objekat u Moodle LMS sistemu. U okviru aktivnosti razlikujemo: bazu podataka, forum, izbor, lekciju, pričaonicu, rečnik, test, upitnik, wiki, domaći zadak, anketa i još mnogo drugih tipova aktivnosti. Ovaj skup aktivnosti nije ograničen jer generički napravljen tak da se lako uvode novi tipovi

aktivnosti. Za aktivnost se analogno kao i resursu adaptacija radi na potpuno isti način.

Student je akter i aktivan učenik u procesu elektronskog obrazovanja koji koristi materijale, resurse koji su dostupni na kursu, ima interaktivnost sa ostalim studentima kao i sa profesorima. Mehanizam ima zadatku da proširi funkcionalnost i karakteristike studenata koristeći personalizovanu tablu studenta koja se koristi za adaptaciju sadržaja i prikaz adaptiranog sadržaja studentu.

Na osnovu navedenog spiska objekata koje treba adaptirati dolazimo do zaključka koje sve mehanizme treba razviti kako bi se u potpunosti kompletirao mehanizam za adaptaciju sadržaja.

Spisak mehanizama za adaptaciju objekata koje je potrebno isprojektovati i razviti:

- Mehanizam za adaptaciju kursa
- Mehanizam za adaptaciju grupe
- Mehanizam za adaptaciju resura
- Mehanizam za adaptaciju aktivnosti
- Mehanizam za adaptaciju studenata (Personalizovana tabla studenta)
- Mehanizam za definisanje ciljeva, kriterijuma, ograničenja
- Mehanizam za definisanje, pregled i obradu anketa, odnosno ciljeva, kriterijuma i ograničenja koja će se koristiti prilikom adaptacije (Mehanizam za adaptaciju studenata)
- Mehanizam za automatsko i ručno razmeštanje studenata po grupama i
- Mehanizam za prikaz adaptiranih objekata

Na narednim stranama biće detaljno objašnjeni svi navedeni mehanizmi.

Kao rezultat razvoja svih ovih mehanizama adaptacije dobila bi se personalizovana tabla studenta koja je detaljno opisana u narednoj sekciji ovog rada.

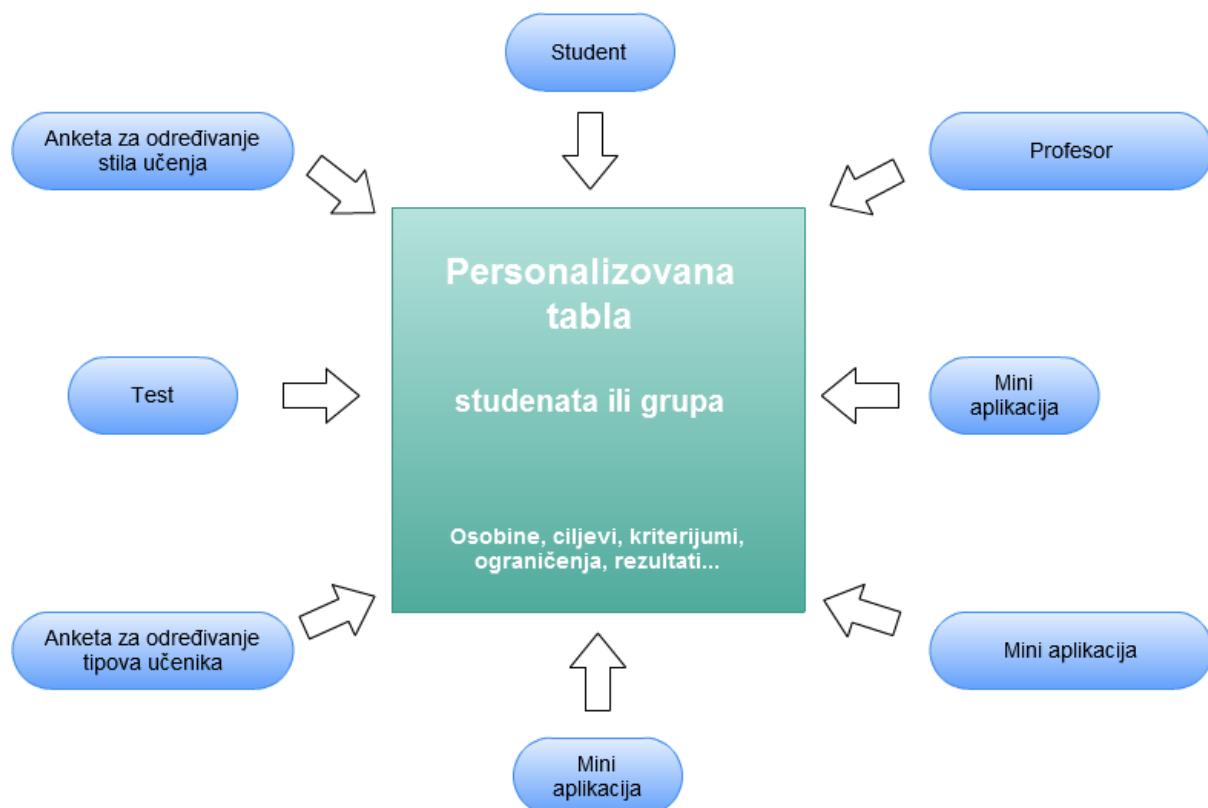
4.3.3. Personalizovana tabla studenta

Personalizovana tabla studenta je zapravo rezultat adaptacije studenta. Na ovoj personalizovanoj tabli biće zapisane sve bitne karakteristike studenta na osnovu kojih će se kasnije raditi adaptacija sadržaja. Ideja ove personalizovane table jeste da rešenje koje radi adaptaciju bude fleksibilno i da ne postoje ograničenja adaptacije.

Personalizovanu tablu mogu da uređuju

- studenti, tako što sami sebi dodeljuje određene karakteristike, potrebe, ciljeve, želje, mogćnosti itd.
- Profesori, oni mogu da beleže određene bitne karakteristike studenata, beleže subjektivne i objektive ocene, razne zaključke do kojih su došli posredno ili neposredno u kontaktu sa studentima i još mnogo drugih informacija za koje procene da su bitne za adaptaciju sadržaja
- Neograničen broj mini aplikacija koje bi određivale određene karakteristike studenata i kao rezultat upisavale na personalizovanu tablu, na osnovu koje se kasnije radi adaptacija sadržaja. Kao primer može da se navede neka specijalizovana anketa koja složenim zaključivanjem dolazi do saznanja i određivanja tipova učenika ili stilova učenja, a na personalizovanoj tabli, upisuje samo kao rezultat zaključivanja. Ovim se dobija fleksibilno rešenje i omogućava da razni eksperti iz raznih oblasti personalizaciju rade na način kako je potreban u dатој oblasti.
- Kao još jedan primer dodatne mini aplikacije može da se odnosi na bilo koju aktivnost koju podržava LMS sistem, npr može da se razvije određen mini aplikacija koja upoređuje rezultat onih testova, domaćih zadataka koje ekspert izabere i kaže da bi trebalo da se nađu na personalizovanoj tabli studenta. Nakon toga, aplikacija sama analizira i upisuje rezultate na personalizovanoj tabli.

Na sledećoj slici predstaljena je ideja koji sve procesi mogu da pišu po personalizovanoj tabli, gde se ovakvim rešenjem dobija fleksibilnost da se kasnije može veoma lako uključivati u sam proces adaptacije veliki broj dodatnih mini aplikacija.



Slika 14. Prikaz procesa koji mogu da uređuju personalizovanu tablu

Personalizovana tabla je neograničena i fleksibilna. Postoji minimalni skup elemenata koje personalizovana tabla treba da ima kako bi mehanizam za adaptaciju mogao da funkcioniše. Personalizovana tabla ima mogućnost da se konstantno nadogradjuje sa novim podacima i informacijama za koje ekspert ili profesor misli da bi bile značajne za adaptaciju sadržaja.

4.3.4. Mehanizam za adaptaciju kursa

Mehanizam za adaptaciju kursa služi za definisanje ciljeva, kriterijuma i ograničenja kursa. Ovaj mehanizam ima zadatak da adaptira kurs, tj da pruži mogućnost da se proširi informacija o kursu, njegovim zahtevima, ciljevima, ograničenjima i slično.

Mehanizam ima mogućnost da izlista sve kreirane kurseve u okviru Moodle LMS sistema, da prikaže detalje kursa, odnosno pogled trenutno dodeljenih ciljeva, kriterijuma, karakteristika i ograničenja kursa, kao i mogućnost da se svi ti podaci promene i dodaju novi.

Da bi se ovaj mehanizam realizivao potrebno je:

- Da se kreira nova tabela u bazi pod nazivom *mdl_am_ciljevi_kursa*. U ovoj tabeli se čuvaju informacije koji ciljevi, kriterijumi, karakteristike i ograničenja su dodeljene kursevima;
- pravo pristupa mehanizmu ima korisnik koji ima ulogu profesora ili viši nivo privilegija u okviru Moodle LMS sistema,
- preduslov za korišćenje ovog mehanizma jeste da je kreiran i definisan mehanizam za definisanje ciljeva, kriterijuma i ograničenja koji će biti kasnije detaljnije objašnjen.

4.3.5. Mehanizam za adaptaciju grupe

Mehanizam za adaptaciju grupe služi za definisanje ciljeva, kriterijuma i ograničenja grupe. Ima zadatak da adaptira postojeću grupu u Moodle LMS sistemu.

Mehanizam ima mogućnost da izlista sve kreirane grupe za izabrani kurs koje je kreiran u Moodle LMS sistemu, zatim mehanizam ima mogućnost da se prike detalje grupe, odnosno koji sve ciljevi, kriterijumi, karakteristike i ograničenja su dodeljena grupi, da se date informacije promene ili pak dodaju nove.

Mehanizam za adaptaciju grupe, služi da se ručno dodele neke osobine, karakteristike, ciljevi grupe i kao takav omogućava da se kasnije razviju dodatni mehanizmi koji bi proces adaptacije automatizovali.

Ako je na nivou mehanizma za adaptaciju sadržaja izabran režim rada adaptacija na nivou grupe, onda svaka grupa dobija svoju personalizovanu tablu gde se prilikom prezentovanja adaptiraog sadržaja zaključivanje donosi na osnovu uporedjivanja karakteristika, kriterijuma, ograničenja, ciljeva grupe i resursa odnosno aktivnosti sa druge strane. Svim studentima u okviru jedne grupe prikazuje identičan sadržaj.

Da bi se pomenuti mehanizam realizivao potrebno je:

- Da se kreira nova tabela u bazi *mdl_am_ciljevi_grupa* koja spaja grupu sa ciljevima. U ovoj tabeli se čuvaju informacije koji ciljevi, kriterijumi, karakteristike i ograničenja su dodeljene određenim grupama;
- pravo pristupa mehanizmu ima korisnik koji ima ulogu profesora ili viši nivo privilegija,
- preduslov za korišćenje ovog mehanizma jeste da je kreiran i definisan mehanizam za definisanje ciljeva, kriterijuma i ograničenja koji će biti kasnije detaljnije objašnjen.

4.3.6. Mehanizam za adaptaciju resursa

Mehanizam za adaptaciju resursa služi za definisanje ciljeva, kriterijuma i ograničenja resursa, odnosno pruža mogućnost da se definiše šta se postiže ako student koristi dati resurs, kojim tipovima studenata je namenjen dati resurs ili pak da li student može da koristi resurs ako su definisana neka ograničenja.

Mehanizam ima mogućnost da izlista sve kreirane resurse u Moodle LMS sistemu, resursi se mogu pretraživati po kursevima i tipovima, kao i slobodnom tekstu kako bi se lakše pronašao resurs koji se adaptira. Pored toga, moguće je pregledati detalje resursa, odnosno videti listu ciljeva, kriterijuma, karakteristika i ograničenja koja su već dodeljena resursu, mogućnost da se te informacije promene ili pak dodaju nove.

Napomena: Ovde je usko grlo predstavlja nedovoljna angažovanost profesora na adaptaciji sadržaja zbog količine posla koji treba uraditi. Svaki profesor koji kreira sadržaj u okviru Moodle LMS sistema, potrebno je da taj sadržaj odmah i adaptira, što zahteva dodatno angažovanje profesora a i postavlja se pitanje kompetentnosti profesora da li ima dovoljna znanja da proceni koji sadržaj kojim karakteristikama studenata odgovara. Sa druge strane, ovaj mehanizam je od ključnog značaja za kompletan proces adaptacije, a ako adaptacija resursa nije urađena onako kako treba, onda i rezultat adaptacije neće biti adekvatan!

Da bi se pomenuti mehanizam realizivao potrebno je:

- Da se kreira nova tabela u bazi *mdl_am_ciljevi_resursa* koja spaja resurs sa njegovim ciljevima, kriterijumima, karakteristikama i ograničenjima.
- pravo pristupa mehanizmu ima korisnik koji ima ulogu profesora ili viši nivo privilegija,
- preduslov za korišćenje ovog mehanizma jeste da je kreiran i definisan mehanizam za definisanje ciljeva, kriterijuma i ograničenja koji će biti kasnije detaljnije objašnjen.

4.3.7. Mehanizam za adaptaciju aktivnosti

Mehanizam za adaptaciju aktivnosti služi sa definisanje ciljeva, kriterijuma i ograničenja aktivnosti, odnosno pruža mogućnost da se definiše šta se postiže ako student radi odredjenu aktivnost i da li može da koristi ako su predhodno definisana neka ograničenja nad datom aktivnošću.

Ovaj mehanizam ima mogućnost da izlista sve kreirane aktivnosti u okviru Moodle LMS sistema, aktivnosti se mogu pretraživati po kursevima i tipovima aktivnosti, kao i slobodnom unosu teksta, odnosno po nazivu i opisu aktivnosti. Pored toga, moguće je pregledati detalje aktivnosti, odnosno videti listu ciljeva, kriterijuma, karakteristika i ograničenja koja su već dodeljena, kao i mogućnost da se te informacije promene ili pak dodaju nove.

Napomena: Ovde je usko grlo predstavlja nedovoljna angažovanost profesora na adaptaciji sadržaja zbog količine posla koji treba uraditi. Svaki profesor koji kreira sadržaj u okviru Moodle LMS sistema, potrebno je da taj sadržaj odmah i adaptira, što zahteva dodatno angažovanje profesora a i postavlja se pitanje kompetentnosti profesora da li ima dovoljna znanja da proceni koji sadržaj kojim karakteristikama studenata odgovara. Sa druge strane, ovaj mehanizam je od ključnog značaja za kompletan proces adaptacije, a ako adaptacija resursa nije urađena onako kako treba, onda i rezultat adaptacije neće biti adekvatan!

Da bi se pomenuti mehanizam realizivao potrebno je:

- Da se kreira nova tabela u bazi *mdl_am_ciljevi_aktivnosti* koja spaja aktivnost sa njegovim ciljevima, kriterijumima, karakteristikama i ograničenjima.
- pravo pristupa mehanizmu ima korisnik koji ima ulogu profesora ili viši nivo privilegija,
- preduslov za korišćenje ovog mehanizma jeste da je kreiran i definisan mehanizam za definisanje ciljeva, kriterijuma i ograničenja koji će biti kasnije detaljnije objašnjen.

4.3.8. Mehanizam za definisanje ciljeva, kriterijuma, ograničenja

Mehanizam za definisanje ciljeva, kriterijuma, ograničenja je osnova kompletног mehanizma za adaptaciju sadržaja sistema elektronskog obrazovanja, jer se svi ostali mehanizmi oslanjaju na njega, odnosno koriste ovaj mehanizam kao osnovu prilikom adaptacije određenih objekata koji su potrebni za funkcionisanje adaptivnog mehanizma.

Ovaj mehanizam ima mogućnost pregleda svih do sada kreiranih ciljeva, kriterijuma, ograničenja, kao i mogućnost kreiranja novih i promene postojećih. Svi ovde kreirani podaci su osnava za kreiranje personalzovane table studenata odnosno grupa na osnovu kojih se kasnije radi adaptacija sadržaja. Ciljevi, kriterijumi, ograničenja koja su ovde kreirana se dodeljuju svim objektima koji učestvuju u procesu adaptacije kao što su kurevi, grupe, resursi, aktivnosti, studenti.

Da bi se pomenuti mehanizam realizirao potrebno je:

- Da se kreiraju nove tabele u bazi *mdl_am_cilj*, *mdl_am_cilj_stavke*, *mdl_am_tip_cilja* *mdl_cilj_kurs_info*, gde će biti smešteni svi kreirani ciljevi, kriterijumi i ograničenja kao i inofrmacije koje dodatno određuju određeni cilj.
- pravo pristupa mehanizmu ima korisnik koji ima ulogu profesora ili viši nivo privilegija.

Postoje 4 tipa ciljeva koje se mogu definisati i koje mehanizam za adaptaciju trenutno podržava, s tim da je ostavljena mogućnost za kasnije proširenje.

Cilj može biti sledeće:

- Otvoreno tekstualno pitanje, na koje kasnije u anketi student daje proizvoljan odgovor, a mehanizam zaključuje koliko se pomenuti odgovor poklapa sa ciljevima resursa, aktivnosti, grupa itd.
- Zatvoreno pitanje sa jednim tačnim odgovorom, gde je moguće navesti više očekivanih odgovora od kojih je samo jedan tačan.
- Zatvoreno pitanje sa više tačnih odgovora, gde je moguće da student izabere više od jednog ponuđenog odgovora prilikom popunjavanja ankete,a pritom

mehanizam za adaptaciju zaključuje koji skup odgovora treba da analizira prilikom adaptacije objekata.

- Ostvareni rezultat na nekom od izbranih kurseva. Mehanizam sam analizira uspešnost studenta na izabranom kursu.

Na osnovu tipa cilja, mehanizam za adaptaciju zna kako da adaptira određeni objekat.

4.3.9. Mehanizam za adaptaciju studenata

Mehanizam za adaptaciju studenata ili mehanizam za definisanje, pregled i obradu anketa, odnosno ciljeva, kriterijuma i ograničenja koja će se koristiti prilikom adaptacije objekata vrši adaptaciju studenta. Ovaj mehanizam se drugačije može nazvati i personalizovana tabla studenata gde se kao mini aplikacija koristi anketa gde student sam daje odgovore na postavljena pitanja.

Adaptacija studenata se radi tako što se popunjava anketa koja je predhodno kreirana u ovom mehanizmu koristeći mehanizam za definisanje ciljeva, zatim se ta popunjena anketa pregledava i obrađuje automatski od strane sistema ili ručno od strane profesora. Prilikom kreiranja ankete profesor će imati mogućnost pregledavanja ankete, da bi video izgled kako će se ta anketa prikazati studentu. Mehanizam će imati obezbeđenu mogućnost ručnog pregledavanja ankete.

Da bi se pomenuti mehanizam realizivao potrebno je:

- Da se kreiraju nove tabele u bazi *mdl_am_anketa*, *mdl_am_anketa_sadrzaj*, *mdl_am_anketa_pokusaj*, *mdl_am_anketa_odgovori*
- pravo pristupa mehanizmu imaće korisnik koji ima ulogu studenta kao i profesor i korisnici koji imaju viši nivo privilegija

U ovom mehanizmu moći će da se kreiraju nove ankete, importuju već definisani ciljevi u anketi, aktiviranje i deaktiviranje anketa kao i pregledavanje samog sadržaja ankete.

4.3.10. Mehanizam za automatsko i ručno razmeštanje studenata po grupama

Mehanizam ima zadatak da obezbedi automatsko i ručno razmeštanje studenata po grupama. Ručno razmeštanje studenata po grupama je omogućeno uz pomoć dodatnog modula koji je realizovan preko Drag & Drop funkcionalnosti za lako manipulisanje.

Automatsko razmeštanje studenata po grupama je kompleksan proces koji se odvija analizirajući definisane ciljeve, kriterijume, ograničenja studenata sa jedne strane i definisane ciljeve, kriterijume i ograničenja grupa sa druge strane. Pored toga, postoje i dodati kriterijumi za rasoređivanje studenat po grupama, a to je i vremenski aspekt, odnosno termini kada studenti mogu da imaju časove nastave (vremenski aspekt).

Ovaj mehanizam ima mogućnost izlistanja svih grupa u okviru jednog kursa sa leve strane, i izlistanje svih upisanih studenata na tom kursu sa desne strane. Profesor jednostavim prevlačenjem studenata u jednu od grupa, radi raspoređivanje studenata. Ako se raspoređivanje radi automatski, potrebno je samo da se klikne na dugme koje automatski rasporedi studente po grupama.

Postoje 2 režima rada na osnovu kojih se studenti razmeštaju po grupama:

1. **Predefinisane grupe:** Ciljevi, kriterijumi, ograničenja i karakteristike grupa su već definisana i potrebno je samo pronaći studente koji odgovaraju datim opisima. Ovo je takodje veoma kompleksan proces zaključivanja. Sistem prvo pronalazi one studente koji su najmanje fleksibilni pa tim putem sužava rezultat.
2. **Novokreirane grupe:** U trenutku se analiziraju svi potencijalni studenti koji treba da budu razmešteni po grupama, shodno dobijenim prelaznim rezultatima kreiraju se moguće grupe studenata koje kasnije profesor (ekspert) treba samo da odobri. Sistem sam zaključuje koliki broj različitih grupa treba da se kreira uzimajući u obzir minimalni i maksimalni broj grupa koji sme da se kreira, veličina grupe (kapacitet učionica), određene karakteristike studenata, ciljevi, porube, ograničenja. Još jedan bitan faktor koji učestvuje u razmeštanju po grupama je i podatak koji se beleži o

studentu a to su termini kada student može da pohađa nastavu, odnosno vremenska dimenzija, vremenska ograničenja.

Dodatak:

- pravo pristupa mehanizmu ima korisnik koji ima ulogu profesora ili viši nivo privilegija.
- Preduslov za uspešno funkcionisanje mehanizma je da su adaptirani objekti (student da je popunio anketu, odradio predefinisani kurs itd.) kao i da su adaptirani objekti kurs i grupa

Razmeštanje studenta po grupama

Algoritam za razmeštanje studenta po grupama je veoma kompleksan, gde je u prilogu data samo osnova tog algoritma:

- 1) Sistem prolazi kroz grupe kojima su već definisani ciljevi, kriterijumi, ograničenja
- 2) Za date ciljeve, karakteristika, ograničenja grupe, sistem analizira studente i uzima one studente na osnovu sledećih nivoa poklapanja
 - a. Prvo se uzimaju studenti koji zadovoljavaju sve uslove, nivo poklapanja je 100% (u realnim slučajevima ovde spada veoma mali broj studenata)
 - b. U drugoj interaciji uzimaju se oni studenti gde je nivo poklapanja iznad 90%
 - c. U trećeoj iteraciji uzimaju se oni studenti gde je nivo poklapanja iznad 80%
 - d. U poslednjoj interaciji uzimaju se oni studenti gde je nivo poklapanja iznad 75%
 - e. Ako nisu zadovoljeni onda se u narednoj interaciji uzimaju svi studenti gde je nivo poklapanja iznad 50%, s tim da se ovi studenti specifičnim oznakama obeležavaju, kako bi profesor, ekspert mogao ručno da ih razmesti
- 3) Nivo poklapanja se određuje na sledeći način:

- a. Upoređuju se prvo oni ciljevi, kriterijumi, ograničenja sa najvišm nivoem prioriteta
 - b. Svaki od ciljeva, kriterijuma, ograničenja ima definisan nivo odstupanja
 - c. Svako od studenata prilikom razmeštanja po grupama dobija koeficijent poklapanja, gde viši nivo koeficijenta garantuje da će student do završetka procesa raspoređivanja ostati u istoj grupi. Ovo govori i to da može da se desi situacija da jedan student bude privremeno razmešten u jednoj grupi, ali da se do kraja procesa ipak rasporedi u nekoj od drugih grupa sa drugim ciljevima, kriterijumima i ograničenja.
- 4) Studenti koji nisu zadovoljili nijedan uslov grupe, označavaju se crvenom oznakom i dobijaju crveni status, odnosno takvi studenti moraju se ručno razmestiti.
 - 5) Uspešnost algoritma se svodi na nekih 75 do 85 % automatske realizacije, u najboljim slučajevima može da bude i do 95%.

Kreiranje grupa studenta i razmeštanje po grupama

Za razliku od algoritma gde se studenti razmeštaju u već kreirane grupe, u ovom algoritmu se kreiranje nove grupe radi automatski, odnosno sam algoritam zaključuje koliko novih grupa treba kreirati. Ovo je veoma kompleksan proces koji u razmatranje uzima određene ulazne podatke koje je definisao ekspert za adaptaciju, a tu su:

- minimalan broj grupa na kursu
- maksimalan broj grupa na kursu
- stepen poklapanja u rasponu od 0 do 100%
- u razmatranje ulaze samo oni ciljevi zapisani na personalizovanim tablama koji su specijalno označeni da služe i za automatsko razmeštanje

Nakon definisanih ulaznih podataka, algoritam kreće sa analizom personalizovanih tabli studenata, pravi pomoćne grupe gde privremeno smešta određene studente sa njihovima karakteristikama i iste upisuje na privremenu personalizovanu tabelu grupe. Kompletan proces se kasnije nastavlja kao kod algoritma kod predefinisanih grupa.

4.3.11. Mehanizam za prikaz adaptiranih objekata

Mehanizam za prikaz adaptiranih objekata koristi adaptivni algoritam za zaključivanje kako bi studentima prikazao adaptiran sadržaj, odnosno onaj sadržaj za koji je student zainteresovan ili je pak njemu namenjen. Ovaj mehanizam se drugačije i naziva i adaptivni režim rada.

Svaki ulazak studenta u adaptivni režim iniciraće pokretanje ovog mehanizma koji će inteligentno doneti odluku, koristeći sve dosad opisane mehanizme, da prikaže onaj sadržaj koji je po svim kriterijumima, ciljevima, karakteristika i ograničenjima namenjen studentima.

Adaptivni algoritam, prilikom svakog ulaska studenta u adaptivni režim rada, upoređuje podatke upisane na personalizovanu tablu studenta i upoređuje sa sadržajem koji ima u svojoj bazi znanja i nakon analize i upoređivanja, izdvaja skup resursa i aktivnosti koji treba da prikaže studentu. Pomenuti skup resursa i aktivnosti se usklađuje na vremenoskom nivou kako je student izabrao u svojim podešavanjima sistema. Moguće je usklađivanje sadržaja u trenunutku (malo je sporije vreme izvršavanja zbog kompleksnosti algoritma, ali se svaki naredni korak pristupa podacima iz keš memorije, dok svako novo pokretanje ovog režima rada ponovo inicira pokretanje algoritma), usklađivanje na 12h, usklađivanje na 24h ili pak usklađivanje na 7 dana.

Ovaj mehanizam predstavlja početak i kraj svakog zahteva za prikaz adaptiranog sadržaja.

- Uslov za izvršavanje ovog mehanizma je da su adaptirani objekti poput kursa, grupe, resursa, aktivnosti i studenta, zatim potrebno je da su studenti razmešteni po grupama koja je adaptirana, donosno definisani su njeni ciljevi.
- pravo pristupa mehanizmu ima korisnik koji ima ulogu studenta koji je popunio anketu, u suprotnom studentu u adaptivnom režimu biće prikazana informacija da mora prvo da popuni odgovorajuću anketu ili uradi odgovorajući kurs.

4.3.12. Prikaz adaptiranih objekata (adaptivni režim)

Prikaz adaptiranih objekata realizovaće se kreiranje modula i bloka u okviru Moodle LMS sistema. Novokreirani modul i blok predstavljaju adaptivni režim Moodle LMS sistema u kome će se prikazivati adaptirani i prilagođeni objekti, odnosno resursi i aktivnosti studentu, poput testova, domaćih zadataka, rečnika, foruma, pričaonica itd.

U okviru bloka biće prikazani top 10 linkova (linkovi ka sadržaju kursa, resursa ili aktivnosti) koji se najviše poklapaju sa studentovim potrebama. Modul predstavlja kompletan adaptivni režim rada, gde je sadržaj kursa prikazan potpuno identično kao i na običnom kursu, koji je organizovan po nedeljama odnosno temama, s tim da će tu biti prikazan samo onaj sadržaj koji je namenjen konkretno tom studentu.

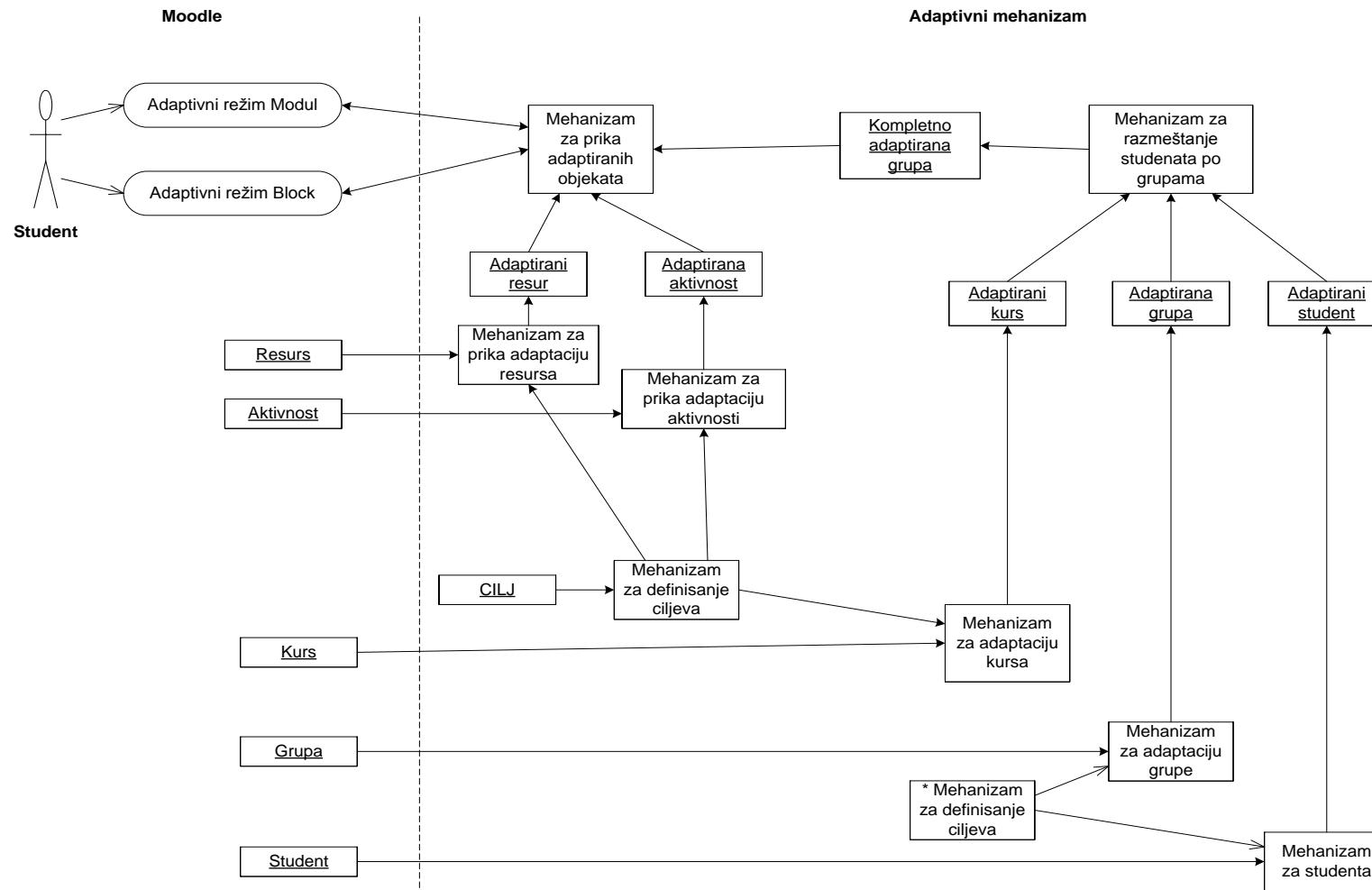
Resurse i aktivnosti koji će biti prikazani u ovom modulu odnosno bloku obezbediće mehanizam za prikaz adaptiranih objekata koji će na osnovu unapred definisanih kriterijumima, kao i logova korisnika i svih pravilno popunjениh ulaznih podataka generisati adaptirani sadržaj.

Sadržaj koji će se prikazivati u ovom modulu i bloku biće prilagodjen studentu, njegovim potrebama, odnosno ciljevima grupa u kojoj je predhodno student smešten.

Dodatno:

- pravo pristupa mehanizmu ima korisnik koji ima ulogu studenta koji je popunio anketu,
- mogućnost da profesor pregleda prikaz adaptiranog sadržaja za unete kriterijume, odnosno simulacija popunjavanja ankete i ostalih ulaznih podataka,
- da bi student pristupio adaptivnom režimu potrebno je da predhodno popuni anketu i pošalje zahtev za razmeštanje u određenu grupu, sistem sam pokušava da razmesti studenta u grupu koje ima slične karakteristike, ako razmeštanje bude neuspešno, profesor će imati mogućnost da ručno smesti studenta u određenu grupu.

4.3.13. Arhitektura mehanizma za adaptaciju



Slika 15. Arhitektura mehanizma za adaptaciju sistema Moodle LMS

4.3.14. Spisak tabela adaptivnog mehanizma

Table: mdl_am_cilj

	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	id	int(11)			No	None	auto_increment
<input type="checkbox"/>	tip_cilja_id	int(11)			No	1	
<input type="checkbox"/>	naziv	varchar(255)	utf8_general_ci		No	None	
<input type="checkbox"/>	text_pitanja	text	utf8_general_ci		No	None	
<input type="checkbox"/>	user_dodao	int(11)			No	-1	
<input type="checkbox"/>	datum_unosa	timestamp			No	CURRENT_TIMESTAMP	
<input type="checkbox"/>	user_promenio	int(11)			No	-1	
<input type="checkbox"/>	datum_promene	timestamp			No	0000-00-00 00:00:00	
<input type="checkbox"/>	status	int(11)			No	1	

Table: mdl_am_cilj_stavke

	Field	Type	Collation	Attributes	Null	Default	Extra	
<input type="checkbox"/>	id	int(11)			No	None	auto_increment	 
<input type="checkbox"/>	cilj_id	int(11)			No	-1		 
<input type="checkbox"/>	text_stavke	text	utf8_general_ci		No	None		 
<input type="checkbox"/>	status	int(11)			No	1		 

Table: mdl_am_tip_cilja

	Field	Type	Collation	Attributes	Null	Default	Extra	
<input type="checkbox"/>	id	int(11)			No	None	auto_increment	 
<input type="checkbox"/>	naziv	varchar(255)	utf8_general_ci		No	None		 
<input type="checkbox"/>	opis	text	utf8_general_ci		No	None		 
<input type="checkbox"/>	status	int(11)			No	1		 

Table: mdl_am_cilj_kurs_info

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	id	int(11)			No	None	auto_increment	  
<input type="checkbox"/>	cilj_id	int(11)			No	-1		  
<input type="checkbox"/>	course_id	int(11)			No	-1		  

Table: mdl_am_anketa							
	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<u>id</u>	int(11)			No	None	auto_increment
<input type="checkbox"/>	<u>naziv</u>	varchar(255)	utf8_general_ci		No	None	
<input type="checkbox"/>	<u>opis</u>	text	utf8_general_ci		No	None	
<input type="checkbox"/>	<u>user_uneo</u>	int(11)			No	-1	
<input type="checkbox"/>	<u>datum_unosa</u>	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
<input type="checkbox"/>	<u>course_id</u>	int(11)			No	-1	
<input type="checkbox"/>	<u>status</u>	int(11)			No	1	

Table: mdl_am_anketa_sadrzaj							
	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<u>id</u>	int(11)			No	None	auto_increment
<input type="checkbox"/>	<u>anketa_id</u>	int(11)			No	None	
<input type="checkbox"/>	<u>cilj_id</u>	int(11)			No	None	

Table: mdl_am_anketa_pokusaj							
	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<u>id</u>	int(11)			No	None	auto_increment
<input type="checkbox"/>	<u>anketa_id</u>	int(11)			No	None	
<input type="checkbox"/>	<u>user_id</u>	int(11)			No	None	
<input type="checkbox"/>	<u>datum_pokusaja</u>	timestamp			No	CURRENT_TIMESTAMP	

Table: mdl_am_anketa_odgovori							
	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<u>id</u>	int(11)			No	None	auto_increment
<input type="checkbox"/>	<u>pokusaj_ankete_id</u>	int(11)			No	-1	
<input type="checkbox"/>	<u>odgovor</u>	text	utf8_general_ci		No	None	

Table: mdl_am_ciljevi_grupe

	Field	Type	Collation	Attributes	Null	Default	Extra	
	<code>id</code>	int(11)			No	None	auto_increment	
	<code>grupa_id</code>	int(11)			No	-1		
	<code>cilj_id</code>	int(11)			No	-1		
	<code>vrednost</code>	text	utf8_general_ci		No	None		
	<code>status</code>	int(11)			No	1		

Table: mdl_am_ciljevi_kursa

	Field	Type	Collation	Attributes	Null	Default	Extra	
	<code>id</code>	int(11)			No	None	auto_increment	
	<code>kurs_id</code>	int(11)			No	-1		
	<code>cilj_id</code>	int(11)			No	-1		
	<code>vrednost</code>	text	utf8_general_ci		No	None		
	<code>status</code>	int(11)			No	1		

Table: mdl_am_ciljevi_resursa

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	<code>id</code>	int(11)			No	None	auto_increment	
	<code>resurs_id</code>	int(11)			No	-1		
	<code>cilj_id</code>	int(11)			No	-1		
	<code>vrednost</code>	int(11)			No	None		
	<code>status</code>	int(11)			No	1		

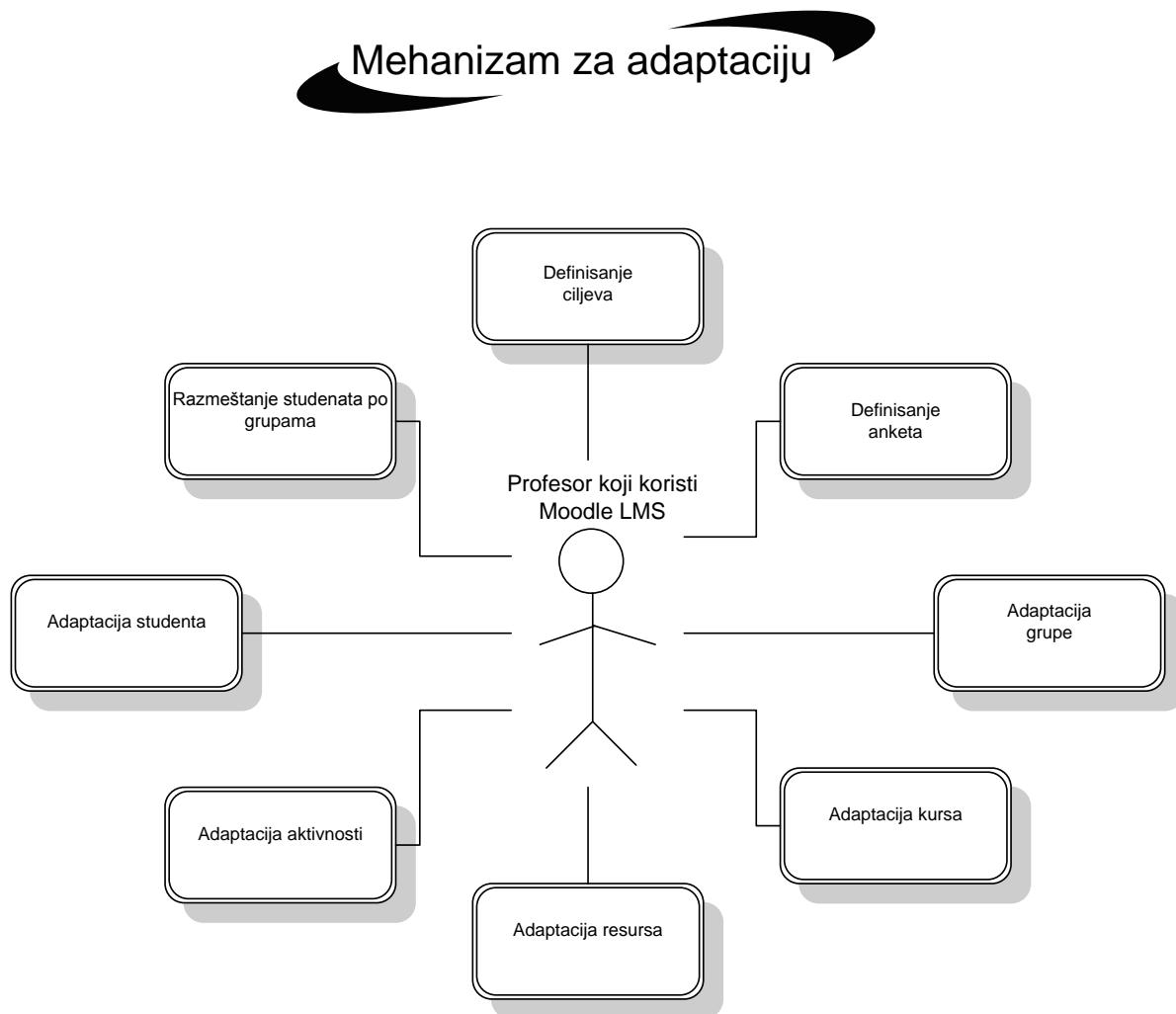
Table: mdl_am_ciljevi_aktivnosti

	Field	Type	Collation	Attributes	Null	Default	Extra	
	<code>id</code>	int(11)			No	None	auto_increment	
	<code>aktivnost_id</code>	int(11)			No	-1		
	<code>cilj_id</code>	int(11)			No	-1		
	<code>vrednost</code>	text	utf8_general_ci		No	None		
	<code>status</code>	int(11)			No	1		

Slika 16. Lista tabela adaptivnog mehanizma

4.4. Slučajevi korišćenja

Na narednoj slici možete videti dijagram slučajeva korišćenja od strane korisnika, odnosno profesora koji radi adaptaciju. Na slici nije prikazan mehanizam za prikazivanje adaptiranih objekata jer pomenuti mehanizam, bez ikakvih intervencija profesora sam dolazi do zaključka šta treba prikazati studentu.



Dijagram 1. Dijagram slučajeva korišćenja adaptivnog mehanizma

4.4.1. Slučaj korišćenja – Definisanje kriterijuma adaptacije

Osnovni scenario za definisanje novog kriterijuma adaptacije.

Slučaj korišćenja: Definisanje novog kriterijuma adaptacije.

Akter: korisnik koji ima ulogu profesora na Moodle LMS sistemu

Preduslov: Pokrenut mehanizam za definisanje kriterijuma adaptacije

Osnovni scenario:

- 1) Akter pokreće formu za unos novog kriterijuma klikom na dugme „Novi kriterijum“;

Kriterijum	Tip kriterijuma	Text pitanja	Kreirao	Datum unosa	
1 + rezultat na uvodnom testu	Rezultat	rezultat na uvodnom kursu	Aleksandar Nastevski	2011-10-08 12:18:29	<input checked="" type="checkbox"/>
2 + Koji je Vas stil učenja?	Više tačnih	Koji je Vas stil učenja?	Aleksandar Nastevski	2010-10-11 19:38:56	<input checked="" type="checkbox"/>
3 + Vrsta materijala?	Jedan tačan	Vrsta materijala koja najčešće koristite?		2010-10-11 19:29:36	<input checked="" type="checkbox"/>
4 + Odaberite vreme kada ti najviše odg	Jedan tačan	Odaberite vreme kada ti najviše odgovoraju	Aleksandar Nastevski	2010-10-11 15:12:09	<input checked="" type="checkbox"/>
5 + Tekst pitanje	Tekst	Text pitanje pitanje?	Aleksandar Nastevski	2010-10-11 11:25:51	<input checked="" type="checkbox"/>
6 + Ovo su materijali cilj	Više tačnih	jel da?	Aleksandar Nastevski	2010-10-09 15:50:24	<input checked="" type="checkbox"/>
7 + Koji se vas stil učenja?	Jedan tačan	Koji je Vas cilj učenja?	Aleksandar Nastevski	2010-10-06 08:59:08	<input checked="" type="checkbox"/>
8 + Rezultat na kursu Moodle Kurs za	Rezultat	Uspešnost rezultata na kursu Moodle Ku		2010-10-01 01:27:43	<input checked="" type="checkbox"/>

Slika 17. Lista kreiranih kriterijuma adaptacije

- 2) Akter popunjava formu, može da odabere jedan od 4 ponuđena tipa kriterijuma, zavisno od tipa, forma se prilagođava za popunu podataka;

Definisanje kriterijuma adaptacije...

Zatvori

Unesite kriterijum

Podaci o kriterijumu adaptacije

Tip kriterijuma: Rezultat

Naziv: Rezultat na kursu Moodle Kurs za Profesore

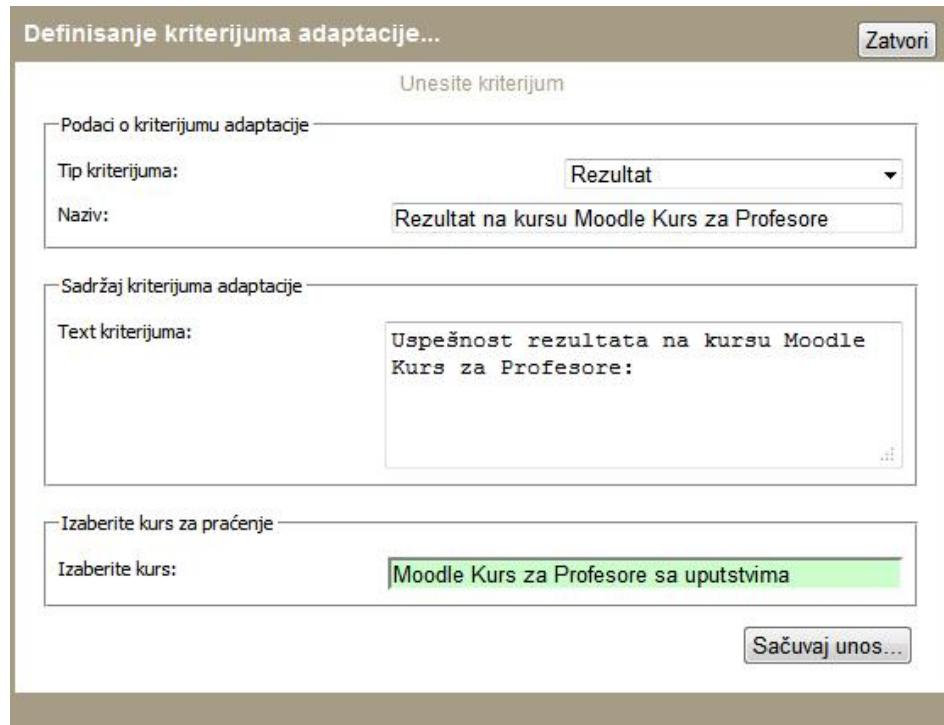
Sadržaj kriterijuma adaptacije

Text kriterijuma: Uspešnost rezultata na kursu Moodle Kurs za Profesore:

Izaberite kurs za praćenje

Izaberite kurs: Moodle Kurs za Profesore sa uputstvima

Sačuvaj unos...



Slika 18. Forma za unos novog kriterijuma adaptacije

- 3) Klikom na dugme „Sačuvaj unos“ pokreće se snimanje novog kriterijuma;

Definisanje kriterijuma adaptacije...

Zatvori

Podaci se snimaju, molimo Vas sačekajte...

Podaci o kriterijumu adaptacije

Tip kriterijuma: Rezultat

Naziv: Rezultat na kursu Moodle Kurs za Profesore

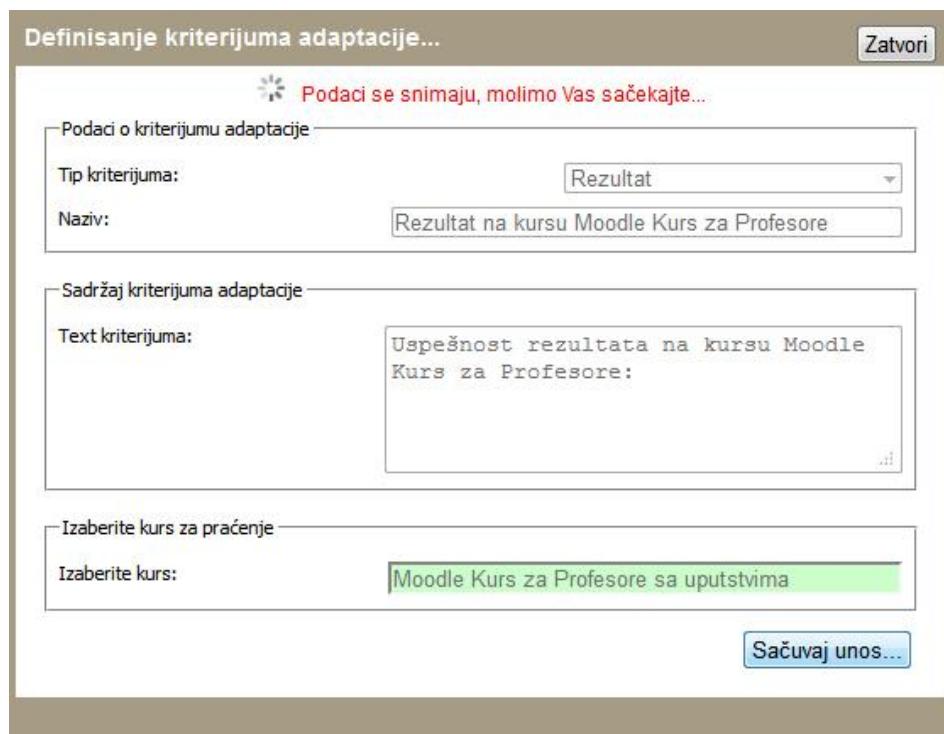
Sadržaj kriterijuma adaptacije

Text kriterijuma: Uspešnost rezultata na kursu Moodle Kurs za Profesore:

Izaberite kurs za praćenje

Izaberite kurs: Moodle Kurs za Profesore sa uputstvima

Sačuvaj unos...



Slika 19. Izgled forme dok se čeka da se kriterijum snimi

- 4) Prikazuje se poruka o uspešnosti snimanja novog cilja.

Aleterantivni scenario:

- 4a) Ako nije moguće snimiti novi cilj, sistem prikazuje poruku na ekranu.

4.4.2. Slučaj korišćenja - Kreiranje nove ankete

Osnovni scenario za kreiranje nove ankete.

Slučaj korišćenja: Kreiranje nove ankete

Akter: korisnik koji ima ulogu profesora na Moodle LMS sistemu

Preduslov: Pokrenut mehanizam za definisanje ankete

Osnovni scenario:

- 1) Akter pokreće formu za kreiranje nove ankete klikom na dugme „Nova anketa“;

Naziv	Opis	Kurs	Kreirao	Datum
Anketa nova mala je promena	sta da tti kazem bre	Profesori Valjevo	Aleksandar Nastevski	2010-10-09
Anketa broj 1 promena	Opis ankete broj 1	Opšti kurs B2 (privi deo)	Aleksandar Nastevski	2010-10-10
Anketa nova mala je	sta da tti kazem bre	Profesori Valjevo	Aleksandar Nastevski	2010-10-09
Anketa koja utvrđuje stil	stil bate		Aleksandar Nastevski	2010-10-10
Anketa broj 2	Opis ankete broj 2	Profesori Valjevo	Aleksandar Nastevski	2010-10-09
Anketa broj 1 promena	Opis ankete broj 1	Opšti kurs B2 (privi deo)	Aleksandar Nastevski	2010-10-09

Naziv	Tip
Tekst pitanje	Tekst
rezultat na uvodnom testu	Rezultat
Vrsta materijala?	Jedan tačan
Koji je Vas stil učenja?	Više tačnih
Rezultat na kursu Moodle Kurs za F	Rezultat
Koji se vas stil učenja?	Jedan tačan

Naziv	Tip
rezultat na uvodnom testu	Rezultat
Koji je Vas stil učenja?	Više tačnih
Vrsta materijala?	Jedan tačan
Odaberi vreme kada ti najviše odgov	Jedan tačan
Tekst pitanje	Tekst
Ovo su materijali cilj	Više tačnih
Koji se vas stil učenja?	Jedan tačan

Slika 20. Mehanizam za definisanje anketa

- 2) Akter popunjava formu za unos nove ankete, pritom izabira kurs za koji je anketa namenjena;

Kreiranje nove ankete... Zatvori

Unesite podatke za anketu:

Podaci o anketi

Naziv:

Opis:
Ova anketa sadrži skup pitanja koja određuju stil učenja studenata.

Izaberite kurs: Moodle Kurs za Profesore sa uputstvima

Sačuvaj unos...

Slika 21. Forma za unos nove ankete

- 3) Klikom na dugme „Sačuvaj unos“ pokreće se snimanje ankete;
- 4) Prikazuje se poruka o uspešnosti snimanja ankete.

Aleterantivni scenario: Ako nije moguće snimiti novi cilj, sistem prikazuje poruku na ekranu.

4.4.3. Slučaj korišćenja – Dodavanje i uklanjanje sadržaja ankete

Slučaj korišćenja: Dodavanje i uklanjanje sadržaja ankete

Akter: korisnik koji ima ulogu profesora na Moodle LMS sistemu

Preduslov: Pokrenut mehanizam za definisanje ankete

Osnovni scenario:

- 1) Akter sa desne strane izabira potencijalne kriterijume koje želi da ubaci u anketu;

Sadržaj ankete: Anketa nova mala je promena		
	Naziv	Tip
1	Tekst pitanje	Tekst
2	rezultat na uvodnom testu	Rezultat
3	Vrsta materijala?	Jedan tačan
4	Koji je Vas stil učenja?	Više tačnih
5	Rezultat na kursu Moodle Kurs za F	Rezultat
6	Koji se vas stil učenja?	Jedan tačan

Potencijalni sadržaj ankete		
	Naziv	Tip
1	rezultat na uvodnom testu	Rezultat
2	Koji je Vas stil učenja?	Više tačnih
3	Vrsta materijala?	Jedan tačan
4	Odaberite vreme kada ti najviše odgov	Jedan tačan
5	Tekst pitanje	Tekst
6	Ovo su materijali cilj	Više tačnih
7	Koji se vas stil učenja?	Jedan tačan

Slika 22. Prikaz dela koji dodaje i uklanja sadržaj ankete

- 2) Nakon toga klikom na dugme „Dodaj“ izabrani kriterijumi se dodaju anketi;
- 3) Akter sa leve strane vidi sadržaj ankete i odabira kriterijume koje želi da izbaci iz sadržaja ankete;
- 4) Nakon odabira ciljeva, klikom na dugme „Ukloni“ kriterijumi se izbacuju iz sadržaja ankete.
- 5) U svakom trenutku akter može da vidi kompletan sadržaj kriterijuma klikom na ikonicu u obliku lufe pored kriterijuma.

Prikaz kriterijuma adaptacije...

Uspešno učitani podaci!

Kriterijum adaptacije

Vrsta materijala koje najčešće koristite?

- elektronske knjige
- audio zapisi
- video klipovi

Sačuvaj unos

Slika 23. Forma za prikaz sadržaja kriterijuma

Aleterantivni scenario: Ako nije moguće dodati ili uklonite izabrane kriterijume, sistem će prijaviti informaciju o tome. Kriterijumi koji su dodati, ne mogu se po drugi put ubaciti u istu anketu.

4.4.4. Slučaj korišćenja – Adaptacija grupe

Slučaj korišćenja: Adaptacija grupe

Akter: korisnik koji ima ulogu profesora na Moodle LMS sistemu

Preduslov: Pokrenut mehanizam za adaptaciju grupe

Osnovni scenario:

- 1) Akter pretražuje grupe po kursu za koje želi da uradi adaptaciju;

Naziv	Tip	Vrednost
DODAJ		
UKLON		

Naziv	Tip
1 Rezultat na kursu Moodle Kurs za profesore	Rezultat
2 rezultat na uvodnom testu	Rezultat
3 Koji je Vas stil učenja?	Više tačnih
4 Vrsta materijala?	Jedan tačan
5 Odaberite vreme kada ti najviše odg	Jedan tačan
6 Tekst pitanje	Tekst
7 Ovo su materijali cilj	Više tačnih

Slika 24. Mehanizam za adaptaciju grupe

- 2) Klikom na ikonicu dodaj u poslednjoj koloni vrši se odabir grupe kojoj se želi dodati kriterijumi adaptacije;
- 3) U na levoj strani prikazana je tabela koja sadrži kriterijume adaptacije izabrane grupe, dok se sa desne strane nalaze potencijalni kriterijumi adaptacije koji se mogu dodeliti

The screenshot shows two tables side-by-side. The left table is titled 'Kriterijumi adaptacije grupe: Š-P3/2 1g' and lists three items: 'Vrsta materijala?' (Jedan tačan), 'rezultat na uvodnom testu' (Rezultat), and 'Koji je Vas stil učenja?' (Više tačnih). The right table is titled 'Potencijalni kriterijumi adaptacije grupe' and lists seven items: 'rezultat na uvodnom testu' (Rezultat), 'Koji je Vas stil učenja?' (Više tačnih), 'Vrsta materijala?' (Jedan tačan), 'Odaberi vreme kada ti najvise odgovor' (Jedan tačan), 'Tekst pitanje' (Tekst), 'Ovo su materijali cilj' (Više tačnih), and 'Koji se vas stil učenja?' (Jedan tačan). Both tables have columns for 'Naziv', 'Tip', and 'Vrednost'. Buttons for 'DODAJ' (Add) and 'UKLONI' (Delete) are visible above the left table. Navigation buttons for 'Strana 1 od 1' and '20' are at the bottom of both tables.

Kriterijumi adaptacije grupe: Š-P3/2 1g			
	Naziv	Tip	Vrednost
1	Vrsta materijala?	Jedan tačan	
2	rezultat na uvodnom testu	Rezultat	
3	Koji je Vas stil učenja?	Više tačnih	

Potencijalni kriterijumi adaptacije grupe			
	Naziv	Tip	
1	rezultat na uvodnom testu	Rezultat	
2	Koji je Vas stil učenja?	Više tačnih	
3	Vrsta materijala?	Jedan tačan	
4	Odaberi vreme kada ti najvise odgovor	Jedan tačan	
5	Tekst pitanje	Tekst	
6	Ovo su materijali cilj	Više tačnih	
7	Koji se vas stil učenja?	Jedan tačan	

Slika 25. Prikaz kriterijuma adaptacije grupe i potencijalni kriterijumi

- 4) Klikom na dugme „Dodaj“ izabrani kriterijumi se dodaju grupi;
- 5) Klikom na dugme „Ukloni“ izabrani kriterijumi se uključuju iz grupe;
- 6) U svakom trenutku akter može da vidi kompletan sadržaj kriterijuma klikom na ikonicu u obliku luke.
- 7) Vrednost definisanog kriterijuma grupe se dodaje klikom na ikonicu „Dodaj“ u tabeli gde se prikazuju kriterijumi grupe.
- 8) Rasporedjivanje studenata po grupama vrši se klikom na ikonicu „Rasporedi“, gde mehanizam sam rasporedi studente i prikaže obaveštenje akteru.
- 9) Klikom na ikonicu „Vidi“ u tabeli spisak grupa, akter može da vidi raspored studenata po grupama.

4.4.5. Slučaj korišćenja – Adaptacija resursa i aktivnosti

Slučaj korišćenja: Adaptacija resursa i aktivnosti

Akter: korisnik koji ima ulogu profesora na Moodle LMS sistemu

Preduslov: Pokrenut mehanizam za adaptaciju resursa ili aktivnosti

Osnovni scenario:

- 1) Akter pretražuje resurse ili aktivnosti po kursu za koje želi da uradi adaptaciju. Prilikom pretraživanja akter mora da izabere tip resursa ili aktivnosti kao i kurs u kome su smešteni resursi odnosno aktivnosti;

Naziv	Tip	Vrednost
1 Vrsta materijala?	Jedan tačan	
2 rezultat na uvodnom testu	Rezultat	
3 Odaberite vreme kada ti najvise odgovorite.	Jedan tačan	
4 Koji je Vas stil učenja?	Više tačnih	

Naziv	Tip
rezultat na uvodnom testu	Rezultat
Koji je Vas stil učenja?	Više tačnih
Vrsta materijala?	Jedan tačan
Odaberite vreme kada ti najvise odgovorite.	Jedan tačan
Tekst pitanje	Tekst
Ovo su materijali cilj	Više tačnih
Koji je Vas stil učenja?	Jedan tačan

Slika 26. Mehanizam za adaptaciju resursa i aktivnosti

- 2) Klikom na dugme „Traži“ u tabeli ispod se prikazuju resursu koji su kreirani u Moodle LMS sistem za izabrane filtere;
- 3) Klikom na ikonicu lupa, akter može da vidi resurs u originalnom prikazu koji je uradjen u Moodle LMS sistemu;
- 4) Adaptacija resursa radi se tako što se klikne na ikonicu „Dodaj“ za željeni resurs i potom dodaju kriterijumi adaptacije;

- 5) Na levoj strani prikazana je tabela koja sadrži kriterijume adaptacije resursa, dok se sa desne strane prikazuju potencijalni kriterijumi;

	Naziv	Tip	Vrednost
1	Vrsta materijala?	Jedan tačan	<input type="text" value="Vrsta materijala?"/>
2	rezultat na uvodnom testu	Rezultat	<input type="text" value="rezultat na uvodnom testu"/>
3	Odaberite vreme kada ti najvise odgovorite	Jedan tačan	<input type="text" value="Odaberite vreme kada ti najvise odgovorite"/>
4	Koji je Vas stil ucenja?	Više tačnih	<input type="text" value="Koji je Vas stil ucenja?"/>

	Naziv	Tip
1	rezultat na uvodnom testu	Rezultat
2	Koji je Vas stil ucenja?	Više tačnih
3	Vrsta materijala?	Jedan tačan
4	Odaberite vreme kada ti najvise odgovorite	Jedan tačan
5	Tekst pitanje	Tekst
6	Ovo su materijali cilj	Više tačnih
7	Koji se vas stil ucenja?	Jedan tačan

Slika 27. Prikaz kriterijuma adaptacije resursa i potencijalnih kriterijuma

- 6) Klikom na dugme „Dodaj“, izabrani kriterijumi se dodeljuju datom resursu;
 7) Klikom na dugme „Ukloni“ izabrani kriterijumi se uklanjuju;
 8) U svakom trenutku akter može da vidi kompletan sadržaj kriterijuma klikom na ikonicu u obliku lufe.

Prikaz kriterijuma adaptacije...

Uspešno učitani podaci!

Kriterijum adaptacije

Koji je Vas stil ucenja?

- vizuelni
- globalni
- sekvensijalni
- intuitivni

Sačuvaj unos

Slika 28. Prikaz potencijalnog kriterijuma adaptacije

- 9) Vrednost dodeljenog cilja resursa se može dodati klikom na ikonicu „Dodaj“ u tabeli gde se prikazuju ciljevi resursa.

Prikaz kriterijuma adaptacije...

Uspešno učitani podaci!

Kriterijum adaptacije

rezultat na uvodnom kursu

Rezultat:

Rezultat veći od unetog broja Rezultat manji od unetog broja

Sačuvaj unos

Slika 29. Dodeljivanje vrednosti cilja za resurs

4.4.6. Slučaj korišćenja – Razmeštanje studenata po grupama

Slučaj korišćenja: Ručno i automatsko razmeštanje studenata po grupama

Akter: korisnik koji ima ulogu profesora na Moodle LMS sistemu

Preduslov: Pokrenut mehanizam za ručno i automatsko razmeštanje studenta po grupama. Pored toga potrebno je da na datom LMS sistemu ima kreiran kurs kao i upisani studenti.

Osnovni scenario:

- 1) Akter ima mogućnost da izabere sa desne strane studente koji su neraspoređeni po grupama i da ih jednostavnim prevlačenjem smesti u jednu od ponuđenih grupa na datom kursu.

Smeštanje polaznika po grupama

Parna	Pon	Uto	Sre	Cet	Pet	Sub	Ned
11h15 - 12h45							
12h45 - 13h30							
15h15 - 16h45							
17h00 - 18h30							
18h45 - 20h15							
20h30 - 22h00							

Nedeljni	Pon	Uto	Sre	Cet	Pet	Sub	Ned
11h15 - 12h45							
12h45 - 13h30							
15h15 - 16h45							
17h00 - 18h30							
18h45 - 20h15							
20h30 - 22h00							

Slika 30. Ručno i automatsko razmeštanje studenata po grupama

- 2) Akter ima mogućnost jednostavnim prevlačenjem da prebacuje polaznike iz jedne u drugu grupu, ili pak da raspoređenog studenta izdvoji u delu za neraspoređene studente.
- 3) Akter ima mogućnost da kliknom na dugme Rasporedi automatski, da pokrene servis za automatsko razmeštanje studenata po grupama, shodno predhodno definisanim kriterijumima grupa u studenata.
- 4) Klikom na dugme Sačuvaj promene, podaci se snimaju u bazu, akteru se prikazuje poruka o uspešnosti izvršene operacije.
- 5) Klikom na dugme Odustani, akter ima mogućnost da prekine proces razmeštanja studenata po grupama.

4.4.7. Slučaj korišćenja – Prikaz adaptiranog sadržaja studentima

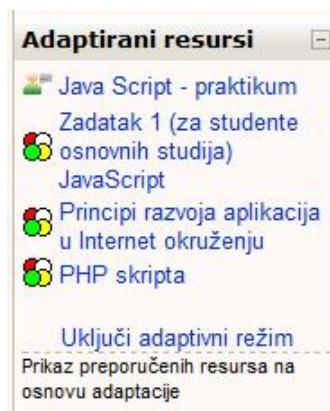
Slučaj korišćenja: Prikaz adaptiranog sadržaja studentima

Akter: korisnik koji ima ulogu studentna na Moodle LMS sistemu

Preduslov: Akter mora da je upisan na kursu koji pristupa, mora da ima definisanu svoju personalizovanu tablu, definisane ciljeve, kriterijume, karakteristike i ograničenja.

Osnovni scenario:

- 1) Akter se sa desne strane u okviru bloka prikazuje sadržaj koji po svim kriterijumima najviše odgovara njegovim potrebama



Slika 31. Prikaz bloka gde se prikazuje adaptiran sadržaj studentima

- 2) Akteri klikom na link pristupa sadržaju i odražuje odreženu aktivnost
- 3) Klikom na link ulazak u adaptivni režim, akteru se prikazuje kompletno adaptiran sadržaj usklađen sa njegovom personalizovanom tablom

Slika 32. Adaptivni režim prikaza sadržaja

- 4) Akter klikom na neki od resursa ili aktivnosti, normalno koristi LMS platformu

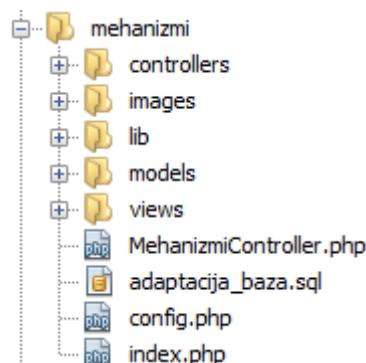
4.5. Uputstvo za instalaciju adaptivnog mehanizma

Mehanizam za adaptaciju sistema elektronskog obrazovanja u okviru Moodle LMS sistema, kao aplikacija koja je urađena u okviru diplomskog rada je veoma jednostavna za instalaciju.

Koraci za instalaciju su sledeći:

- Potrebno je prvo da root direktorijum aplikacije pod nazivom „**mehanizmi**“ kopirate u root direktorijum Moodle-a,
- Potom da importujete tabele koje su dodatno kreirane za potrebne adaptivnog mehanizma. Tabele se importuju tako što se importuje sql fajl **adaptacija_baza.sql** koji se nalazi i root direktorijumu mehanizma za adaptaciju.
- Aplikacija koristi istu bazu koju koristi i Moodle, pristup bazi se vrši sa parametrima kao i za pristup Moodle-ovoj bazi,
- Aplikaciju možete pokrenuti kada u brauzeru upišete adresu <http://localhost/moodle/mehanizmi/> ili možete je pokrenuti u okviru administrativnog menija u Moodle-u pod nazivom Adaptivni mehanizmi.

Na sledećoj slici predstavljen je root direktorijum aplikacije:



Slika 33. Root direktorijum aplikacije

4.6. Web servis za slanje SMS poruka (nSMS)

Web servis za slanje SMS poruka predstavlja kombinaciju Android aplikacije koja se pokreće na telefonu, kao i web aplikacije koja služi kao web servis za povezivanje sa ostalim poslovnim sistemima.

Ideja za razvoj ovako jednog servisa, potekla je odatle da se iskoriste lični resursi za slanje SMS poruka i kako bi se zaobišli razni operateri koji prodaju kredit za slanje SMS poruka.

Kod ovog servisa, ideja je da svako od korisnika koji ima neki poslovni sistem na internetu i želi da na svom sistemu integriše sistem za komunikaciju preko SMS-a, pritom da mogu da koriste svoje resurse, odosno ako već imaju telefon na Android platformi, da iskoriste svoj telefon, instaliraju nSMS aplikaciju, kreiraju besplatno nalog na webNSMS servisu i koriste svoj lični broj koji se nalazi u telefonu za slanje SMS poruka gde se troše ona sredstva koja su rasoloživa na toj sim kartici. Kartica može da bude postpejd, pripejd, da ima neki paket, kredit, određen broj besplatnih sms poruka itd i da na taj način obaveštavaju svoje korisnke poslovnih sistema automatizovano.

Kompletan servis sastoji se iz sledeći delova:

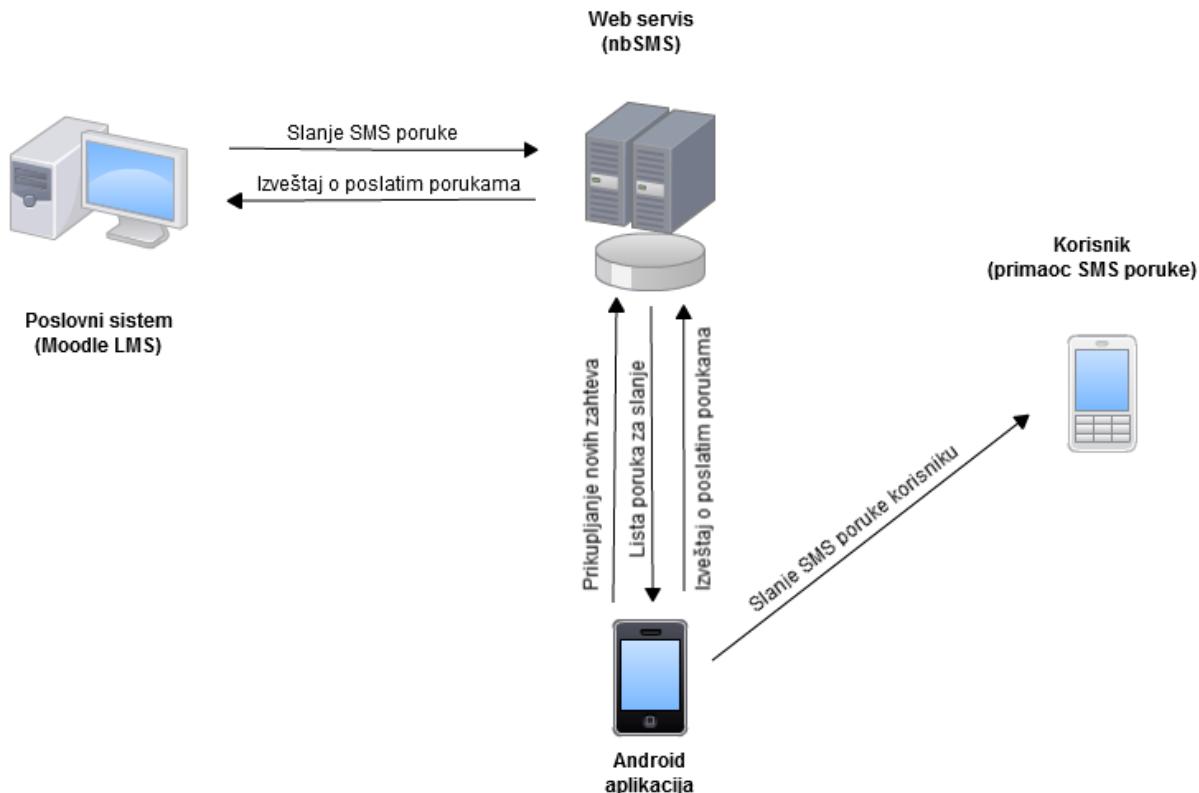
- Android aplikacija
- Web servis, PHP aplikacija
- Adapter za povezivanje sa poslovnim sistemom

Web servis omogućava sledeće mogućnosti

- Kreiranje korisnika, odnosno naloga, koji mogu da koriste preko web servisa za slanje sms poruka.
- Pregled poslatih sms poruka sa njihovim statusima
- Ručno slanje sms poruka preko web aplikacije
- Android API koji omogućava da se android aplikacija poveže sa ovim servisom
- Web2Sms API koji omogućava da se neka poslovna aplikacija poveže sa ovim servisom

4.6.1. Arhitektura nbSMS servisa

Kao što može da se vidi na narednoj slici, arhitektura ovog sistema jasno pokazuje da se sistem fleksibilan i da omogućava da se povežu nekompatibilni sistemi.



Slika 34. Arhitektura nbSMS servisa

Poslovni sistem može da bude bilo koja aplikacija na internetu, može da bude pisana na bilo kom programskom jeziku, bitno je da ima mogućnost da može da izvrši HTTP request i da ima podršku za obradu XML dokumenata.

Poslovni sistem, izvršava HTTP request, gde prosleduje SMS poruke na slanje. Web servis, prima zahteve i skladišti zahteve u lokalnoj bazi podataka. Android aplikacija na periodično pristupa web servisu i proverava da li ima novih zahteva za slanje, ako ima web servis vraća tu listu zahteva koje treba da se pošalju. Android aplikacija nakon toga šalje SMS poruku krajnjem korisniku putem standardnog Android kanala. Nakon prijema izveštaja o poslatim porukama od strane provajdera, android aplikacija šalje izveštaj o poslatim porukama web servisu, koji dalje te izveštaje prosleđuje Web poslovom sistemu.

4.6.2. Android API

Web servis preko kontrole za Android API omogućava android aplikaciji da proverava zahteve za slanje konkretnih sms poruka.

Prilikom instalacije android aplikacije, potrebno je da se urade i određena podešavanja u samoj aplikaciji kako bi se android aplikacija uspešno povezala za web servisom.

Osnovno podešavanje jeste da se unese korisničko ime, broj telefona koji je registrovan na web servisu kao i lozinka za pristup. Ovaj deo je osnovni deo podešavanja, kako bi android aplikacija mogla da prikupljuje zahteve za slanje sms poruka sa web servisa i koristi operacije Android API-a.

Osnovna podešavanja:

- Broj telefona koji je u telefonu i koji je registrovan na web servisu
- Korisničko ime
- Lozinka za pristup

Pored toga u samoj android aplikaciji moguće je podesiti i ostala podešavanja:

- Broj sms poruka koje mogu da se pošalju u jednoj interaciji/sesiji
- Broj sms poruka koje mogu da se pošalju na dnevnom nivou
- Vreme na koje se pokreće servis za slanje novih sms poruka, podrazumevano je da se servis pokreće na svakih 15 minuta

Aplikacija pored toga ima još mogućnosti:

- Pregled poslatih sms poruka sa statusima isporuke primaocu
- Pregled izveštaja koliko je do sada ukupno poslata poruka od instalacije aplikacije
- Operacija koja proverava da li ima novih zahteva za slanje, ali se ne šalju sms poruke
- Operacija koja omogućava da se ručno pokrene servis za proveru zahteva za slanje sms poruka i slanje sms poruka

Prikupljanje novih zahteva za slanje

Primer kako se šalje zahtev da se pokupe nove sms poruke za slanje sa web servisa je dat u uprilogu sa objašnjenjem šta znače koji parametri.

HTTP request

http://nsms.nbsoft.rs/?view=android&task=getnewmessage&clientid=2&username=nastevski&password=nastevski&phoneFrom=+381653141441&limit=15&attempt=2&no_html=1

Parametri

Naziv	Vrednost	Opis
view	android	Naziv kontrolora kome se pristupa
task	getnewmessage	Naziv operacije kojoj se pristupa
no_html	1	Označava da neće ispisati html već vraća XML
clientid	2	Id klijenta koji je registrovan na web servisu kako bi mogla da se izvrši autorizacija
username	Nastevski	Korisničko ime za proveru pristupa
password	Nastevski	Lozinka za proveru pristupa servisu
phonefrom	+381653141441	Telefon sa kojeg se šalje sms poruka
limit	15	Broj novih zahteva koji se preuzimaju (ako je na čekanju već broj sms poruka, ostatak će se preuzeti u narednoj interaciji)
attempt	2	Označava da uzme samo one poruke koje su pokupale da se pošalju najviše 2 puta

Tabela 1. Objašnjenje parametara za HTTP zahteva za slanje novih SMS poruka

Response

Primer odgovora kada se pošalje zahtev za prikupljanje novih zahteva za slanje sms poruka izgleda ovako:

```
<?xml version="1.0" encoding="utf-8"?>
<messages size="3" clientid="2">
  <message id='19'>
    <id>19</id>
    <subject>naslov poruke</subject>
    <textMessage>telo poruke</textMessage>
    <priority>1</priority>
    <status>-1</status>

    <sendToPhone>3816565700</sendToPhone>
    <attempt>0</attempt>
  </message>
  <message id='18'>
    <id>18</id>
    <subject>naslov poruke</subject>
    <textMessage>telo poruke</textMessage>

    <priority>1</priority>
    <status>-1</status>
    <sendToPhone>381653141441</sendToPhone>
    <attempt>0</attempt>
  </message>
  <message id='17'>
    <id>17</id>
    <subject>naslov poruke</subject>
    <textMessage>telo poruke</textMessage>
    <priority>1</priority>
    <status>-1</status>
    <sendToPhone>3816565700</sendToPhone>
    <attempt>0</attempt>
  </message>
</messages>
```

Ako nema poruka za slanje, parametar size dobija vrednost 0 (size=0)

Response kada korisnik nema pravo pristupa

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <id>1</id>
  <info>No have premision! Check your username and password! Contact our administration!</info>
</error>
```

Slanje izveštaja o poslatim porukama i njihovim statusima

Druga operacija jeste da kada se pošalje prouka sa andoird telefona da se web servisu vrati odgovor odnosno izveštaj o poslatim porukama sa njihovim statusima.

Mogući statusi:

-1: novi zahtev, 1: poslata, 2: poništena, 3: šalje se, 4: nije poslata

HTTP Request

http://nsms.nbsoft.rs/?view=android&task=status&requeststatus=8:2:123124141414,9:1:21312414&no_html=1&clientid=2&username=nastevski&password=nastevski&phoneFrom=381653141441

Naziv	Vrednost	Opis
view	android	Naziv kontrolora kome se pristupa, odosno android drajver
task	status	Naziv operacije
no_html	1	Označava da neće ispisati html već vraća XML
clientid	2	Id klijenta koji je registrovan na web servisu kako bi mogla da se izvrši autorizacija
username	Nastevski	Korisničko ime za proveru pristupa
password	Nastevski	Lozinka za proveru pristupa servisu
requeststatus	8:2:123124141414, 9:1:21312414	ovde se prosleđuje id poruke, odnosno requesta, to je ID zahteva koji se dobio iz XML id message Primer: 12:1:time(),15:-1:time(),16:3:time() time: predstavlja vreme u formatu 123454546 12 je id poruke, 1 je status, time() je vreme u int formatu. Kada se šalje za više poslatih sms poruka, onda se zapis odvaja zarezom
phoneFrom	381653141441	Sa kog broja su poslate sms poruke

Tabela 2. Lista parametara koji se prosleđuju za dobijanje statusa poslatih poruka

Response ako je uspešno snimljen status

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <flag>1</flag>
  <info>Saved status request!</info>
  <error>0</error>
</error>
```

Ako nastane greska onda flag ima vrednost -1, u deo info je opis greske, u deo error je id greske.

Spisak mogućih grešaka koje mogu da nastanu:

Šifra greške	Opis
21	nisu prosleđeni zahtevi, No selected request!
22	No corect data send! ako nisu podaci poslati u ispravom formatu
23	No corect data send! ako nisu podaci poslati u ispravom formatu
1	nema pravo pristupa
0	znači da nema greške

Tabela 3. Lista moguća grešaka prilikom slanja SMS poruke

4.6.3. Web2Sms API

Web servis omogućava putem API-a, odnosno putem HTTP requesta i XML-a da se razmene informacije između nekomptabilnih sistema. Ovaj deo omogućava da se poveže poslovni sistem sa web servisom za sanje sms poruka.

Razmena informacija se svodi na izvršavanju nekog HTTP zahteva (request) i dobijanja nekog odgovora (response).

Poslovni sistem preko HTTP zahteva može da odradi operaciju slanje zahteva za slanje SMS poruka, pored toga, poslovni sistem može i da proveri status konkretnih poruka. Prilikom podešavanja i implementacije, na strani poslovnog sistema, poslovni sistem treba da obazbedi da Web servis može da izvrši HTTP request i vrati izveštaj o poslatim porukama poslovnom sistemu. Prilikom kreiranja naloga na web servisu podešava se URL koji će biti izvršen kada web servis dobije informaciju o statusu poslate SMS porke.

Slanje SMS poruka

Da bi poslovni sistem poslao SMS poruke, mora da se izvrši HTTP request koji ima sledeći potpis:

HTTP request

`http://nsms.nbsoft.rs/?view=web2sms&task=send&clientid=2&username=nastevski&password=nastevski&subject=nasov+poruke&body=telo+poruke&phones=+381653141441,+381653141441&no_html=1`

U prilogu je dato objašnjenje za svaki od parametara koji se prosleđuju:

Naziv	Vrednost	Opis
view	web2sms	Naziv kontrolora kome se pristupa, odosno adaptera za slanje sms poruke
task	send	Naziv operacije
no_html	1	Označava da neće ispisati html već vraća XML
clientid	2	Id klijenta koji je registrovan na web servisu kako bi mogla da se izvrši autorizacija
username	Nastevski	Korisničko ime za proveru pristupa
password	Nastevski	Lozinka za proveru pristupa servisu
subject	Naslov poruke	Polje gde se upisuje naslov poruke
body	telo+poruke	Sadržaj sms poruke koja se šalje
phones	+381653141441,+3 81653141441	Lista brojeva mobilnih telefona kome trebaju biti isoručene SMS poruke. Ako se ista poruka šalje na više brojeva telefona, onda se brojevi odvajaju zarezom

Tabela 4. Opis parametara koje prosledjuje poslovna aplikacija web sms servisu

Response

Kao odgovor nakon izvršavanja HTTP zahteva dobija se sledeći XML:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <flag>1</flag>
  <info>Saved status request!</info>
  <request>Id zahteva</request>
  <message>Id poruke</message>
  <error>0</error>
</error>
```

Provera statusa poslatih SMS poruka

Poslovni sistem ima mogućnost da implementira operaciju provere izveštaja o poslatim porukama. Da bi se dobio izveštaj, potrebno je da se izvrši HTTP request i kao odgovor se dobija XML za prikazom traženih informacija.

HTTP request

http://nsms.nbsoft.rs/?view=web2sms&task=status&clientid=2&username=nastevski&password=nastevski&messagesIds=123,124,125&no_html=1

Naziv	Vrednost	Opis
view	web2sms	Naziv kontrolora kome se pristupa, odosno adaptera za slanje sms poruke
task	send	Naziv operacije
no_html	1	Označava da neće ispisati html već vraća XML
clientid	2	Id klijenta koji je registrovan na web servisu kako bi mogla da se izvrši autorizacija
username	Nastevski	Korisničko ime za proveru pristupa
password	Nastevski	Lozinka za proveru pristupa servisu
messagesIds	123,124,125	Id poruke, odnosno zahteva koji u poslati. Ova informacija se dobija prilikom slanja prvog zahteva. Ako se želi da se istovremeno proveri status za više poslatih poruka, potrebo je da se id poruke odvoji zarezom.

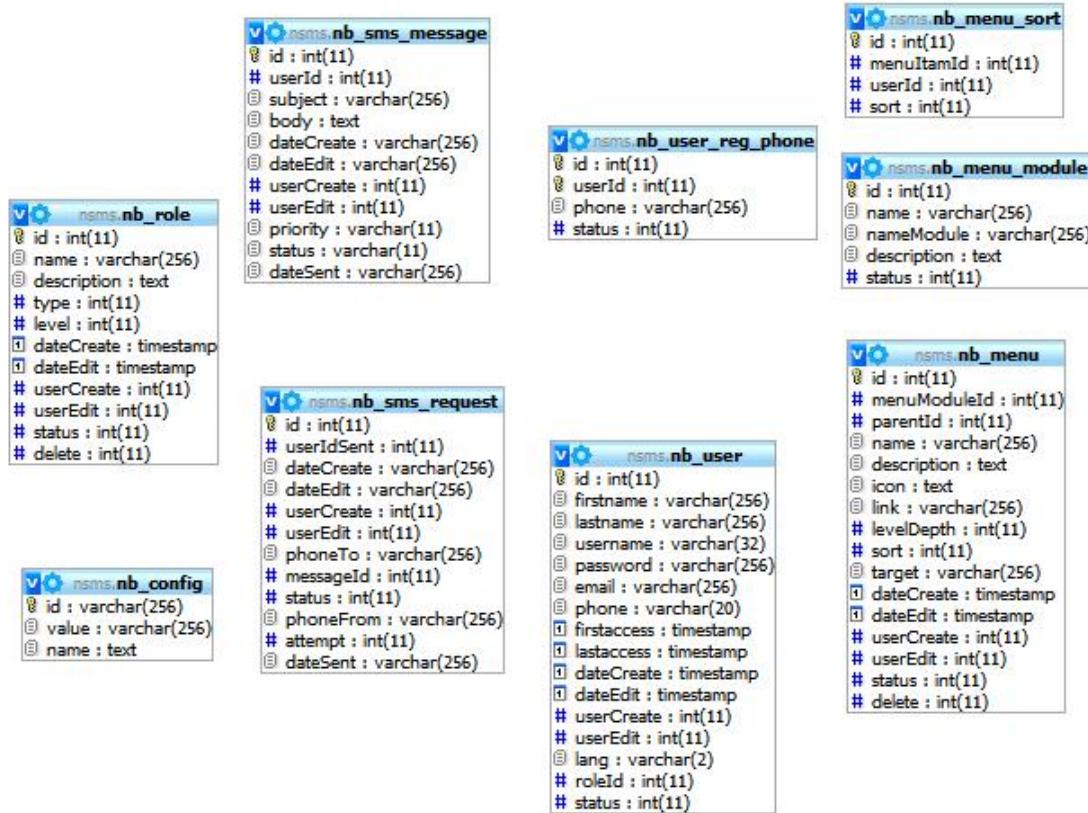
Tabela 5. Opis parametara koji prosledjuje poslovna aplikacija za proveru poslatih SMS poruka

Response

```
<?xml version="1.0" encoding="utf-8"?>
<messages clientid="2">
    <message id='123'>
        <id>123</id>
        <status>1</status>
        <info>delevery</info>
    </message>
    <message id='124'>
        <id>124</id>
        <status>1</status>
        <info>delevery</info>
    </message>
</messages>
```

4.6.4. Pregled baze podataka web servisa za slanje SMS poruka

U prilogu se nalazi prikaz tabela koje predstavljaju bazu web servisa za slanje SMS poruka.



nb_user								
	Browse	Structure	SQL	Search	Insert	Export	Import	Operations
	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	id	int(11)			No	None	AUTO_INCREMENT	    
<input type="checkbox"/>	firstname	varchar(256)	utf8_unicode_ci		Yes	NULL		    
<input type="checkbox"/>	lastname	varchar(256)	utf8_unicode_ci		Yes	NULL		    
<input type="checkbox"/>	username	varchar(32)	utf8_unicode_ci		Yes	NULL		    
<input type="checkbox"/>	password	varchar(256)	utf8_unicode_ci		Yes	NULL		    
<input type="checkbox"/>	email	varchar(256)	utf8_unicode_ci		Yes	NULL		    
<input type="checkbox"/>	phone	varchar(20)	utf8_unicode_ci		No	None		    
<input type="checkbox"/>	firstaccess	timestamp			Yes	NULL		    
<input type="checkbox"/>	lastaccess	timestamp			Yes	NULL		    
<input type="checkbox"/>	dateCreate	timestamp			Yes	NULL		    
<input type="checkbox"/>	dateEdit	timestamp			Yes	NULL		    
<input type="checkbox"/>	userCreate	int(11)			No	-1		    
<input type="checkbox"/>	userEdit	int(11)			No	-1		    
<input type="checkbox"/>	lang	varchar(2)	utf8_unicode_ci		No	en		    
<input type="checkbox"/>	roleId	int(11)			No	3		    
<input type="checkbox"/>	status	int(11)			No	-1		    

nb_user_reg_phone								
Browse		Structure	SQL	Search	Insert	Export	Import	Operations
	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	<u>id</u>	int(11)			No	None	AUTO_INCREMENT	     
<input type="checkbox"/>	<u>userId</u>	int(11)			No	-1		     
<input type="checkbox"/>	<u>phone</u>	varchar(256)	utf8_unicode_ci		No	None		     
<input type="checkbox"/>	<u>status</u>	int(11)			No	1		     

nb_sms_message								
Browse		Structure		SQL	Search	Insert	Export	Import
	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	id	int(11)			No	None	AUTO_INCREMENT	     
<input type="checkbox"/>	userId	int(11)			No	None		     
<input type="checkbox"/>	subject	varchar(256)	utf8_unicode_ci		No	None		     
<input type="checkbox"/>	body	text	utf8_unicode_ci		No	None		     
<input type="checkbox"/>	dateCreate	varchar(256)	utf8_unicode_ci		No	None		     
<input type="checkbox"/>	dateEdit	varchar(256)	utf8_unicode_ci		No	None		     
<input type="checkbox"/>	userCreate	int(11)			No	-1		     
<input type="checkbox"/>	userEdit	int(11)			No	-1		     
<input type="checkbox"/>	priority	varchar(11)	utf8_unicode_ci		No	1		     
<input checked="" type="checkbox"/>	status	varchar(11)	utf8_unicode_ci		No	-1		     
<input type="checkbox"/>	dateSent	varchar(256)	utf8_unicode_ci		Yes	NULL		     

nb_sms_request								
Browse		Structure		SQL	Search	Insert	Export	Import
	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	<u>id</u>	int(11)			No	None	AUTO_INCREMENT	     
<input type="checkbox"/>	<u>userIdSent</u>	int(11)			No	-1		     
<input type="checkbox"/>	<u>dateCreate</u>	varchar(256)	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	<u>dateEdit</u>	varchar(256)	utf8_unicode_ci		Yes	NULL		     
<input checked="" type="checkbox"/>	<u>userCreate</u>	int(11)			No	-1		     
<input type="checkbox"/>	<u>userEdit</u>	int(11)			No	-1		     
<input type="checkbox"/>	<u>phoneTo</u>	varchar(256)	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	<u>messageId</u>	int(11)			No	-1		     
<input type="checkbox"/>	<u>status</u>	int(11)			No	-1		     
<input type="checkbox"/>	<u>phoneFrom</u>	varchar(256)	utf8_unicode_ci		Yes	NULL		     
<input type="checkbox"/>	<u>attempt</u>	int(11)			No	0		     
<input type="checkbox"/>	<u>dateSent</u>	varchar(256)	utf8_unicode_ci		Yes	NULL		     

The screenshot shows two database tables in MySQL Workbench:

nb_config Table:

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> id	varchar(256)	utf8_unicode_ci		No	None		
<input type="checkbox"/> value	varchar(256)	utf8_unicode_ci		No	None		
<input type="checkbox"/> name	text	utf8_unicode_ci		Yes	NULL		

nb_role Table:

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> id	int(11)			No	None	AUTO_INCREMENT	
<input type="checkbox"/> name	varchar(256)	utf8_unicode_ci		No	None		
<input type="checkbox"/> description	text	utf8_unicode_ci		No	None		
<input type="checkbox"/> type	int(11)			No	2		
<input type="checkbox"/> level	int(11)			No	60		
<input type="checkbox"/> dateCreate	timestamp			Yes	NULL		
<input type="checkbox"/> dateEdit	timestamp			Yes	NULL		
<input type="checkbox"/> userCreate	int(11)			No	-1		
<input type="checkbox"/> userEdit	int(11)			No	-1		
<input type="checkbox"/> status	int(11)			No	-1		
<input type="checkbox"/> delete	int(11)			No	-1		

Slika 35. Spisak tabela nbSMS servisa za slanje SMS poruka

4.6.5. Slučajevi korišćenja – nbSMS web aplikacija

Slučaj korišćenja: Pregled liste korisnika web servisa

Akter: korisnik koji ima pravo pristupa web aplikaciji

Preduslov: Potreno je da korisnik ima kreiran nalog na web servisu za pristup aplikaciji.

Osnovni scenario:

- 1) Akter iz glavnog naviogacionog menija bira opciju Korisnici;
- 2) Nakon toga dobija se ekran sa prikazom liste korisnika
- 3) Akter ima mogućnost da pretražuje korisnike tako što u predviđeno polje unese deo imena ili pezimene i klikne na dugme traži
- 4) Akter ima mogućnost da zabrani pristup nekom korisniku klikom na semafor ikonicu (crveno označava da korisnik nema prava pristupa, zelena ikonica označava da korisnik ima prava pristupa aplikaciji)

ID	Ime i prezime	E-mail	Telefon	Username	Pristup
1	Korisnik 1 Korisnik 1	korisnik1@nbsoft.rs	0653141441	korisnik1	-
2	Korisnik 2 Korisnik 2	korisnik2@nbsoft.rs	0653141441	korisnik2	-
3	Marko Markovic	marko@nbsoft.rs	0653141441	marko	-
4	Aleksandar Nastevski	aleksandar@nbsoft.rs	381653141441	nastevski	07.01.2013, 16:59:18
5	Blagoje Nastevski	acanastevski@hotmail.com	381653141441	blagoje	-

Slika 36. Prikaz liste korisnika servisa

Slučaj korišćenja: Dodavanje novog korisnika servisa

Akter: korisnik koji ima pravo pristupa web aplikaciji

Preduslov: Potreno je da korisnik ima kreiran nalog na web servisu za pristup aplikaciji.

Osnovni scenario:

- 1) Akter iz glavnog naviogacionog menija bira opciju Korisnici;
- 2) Nakon toga dobija se ekran sa prikazom liste korisnika
- 3) Akter bira opciju Novi korisnik, nakon toga otvara se ekran kao na narednoj slici

Evidentiranje korisnika sistema

Uspešno učitani podaci!

Osnovne informacije

Ime: Aleksandar
Prezime: Nastevski
Telefon: 381653141441
E-mail: aleksandar@nbsoft.rs

Podaci o nalogu

Uloga: Client
Jezik: Srpski
Korisničko ime: nastevski
Lozinka:

Podaci o telefonu

Brojevi: +381653141441

Example (+381653141441, +38163222333)

Sačuvaj promene

Slika 37. Dodavanje korisnika

- 4) Akter unesi sve potrebne podatke u ispravnom formatu
- 5) Kada su podaci unet, potrebno je da se klikne na dugme Sačuvaj promene i time je korisnik kreira. Na vrhu frme akteru se prikazuje poruka o usšenosti kreiranja korisnika.

Slučaj korišćenja: Pregled poslatih SMS poruka

Akter: korisnik koji ima pravo pristupa web aplikaciji

Preduslov: Potreno je da korisnik ima kreiran nalog na web servisu za pristup aplikaciji.

Osnovni scenario:

- 1) Akter iz glavnog naviogacionog menija bira opciju SMS poruke;
- 2) Nakon toga dobija se ekran sa prikazom liste SMS poruka kao na sledećoj slici

The screenshot shows the nSMS service interface. At the top, there is a header with the logo 'nSMS service' and a user profile 'Aleksandar Nastevski Odjava'. Below the header is a navigation bar with links: Početna, Korisnici, SMS poruke, and Podešavanja. A search bar says 'Pretraga: Unesite ime i prezime...'. The main content area has a title 'SMS poruke za slanje :: Pretraga i podešavanja'. Below it is a search form with fields 'Naziv', 'Traži', and 'Nova SMS poruka'. The main part of the page displays two tables: 'SMS poruke' and 'Poruka broj 4'. The 'SMS poruke' table lists six messages with columns: ID, Naslov, Tekst poruke, Prioritet, Kreirano, Poslato, and actions. The 'Poruka broj 4' table shows two entries with columns: Telefon, Pokušaj, and Status. At the bottom, there are pagination controls for both tables.

ID :	Naslov	Tekst poruke	Prioritet	Kreirano	Poslato	
1	17 Naslov poruke	Ovo je sadržaj poruke!	1	07.01.2013. 17:22:10	07.01.2013. 17:22:10	
2	16 Poruka broj 4	Ovo je tekst poruke koji će biti poslat	1	07.01.2013. 17:21:44	07.01.2013. 17:21:44	
3	15 Nova poruka	Ovo je tekst poruke koji će biti poslat	1	07.01.2013. 17:21:31	07.01.2013. 17:21:31	
4	14 Novi poruka	Ovo je tekst poruke koji će biti poslat	1	07.01.2013. 17:21:22	07.01.2013. 17:21:22	
5	13 Nova poruka	Sadržaj nove poruke, sadržaj	1	07.01.2013. 17:12:55	07.01.2013. 17:12:55	
6	12 Test poruka za slanje	Ovo je prva sms poruka poslat	1	07.01.2013. 17:12:05	07.01.2013. 17:12:05	

	Telefon	Pokušaj	Status
1	+38163222333	0	poslata
2	+381653141441	0	poslata

Slika 38. Pregled poslatih SMS poruka

- 3) U tabeli sa leve strane se nalazi lista poruka koje su poslate. Klikom na neku SMS poruku, u tabeli sa desne strane se prikazuje lista primaoca te poruke.
- 4) Akter ima mogućnost pretrage poslatih poruka po korisnicima, po naslovu ili sadržaju poruke, po datumu slanja itd. Nakon unosa kriterijuma, potrebno je da se klikne na dugme Traži, gde se nakon toga u tabli ispod prikazuje rezultat pretrage.

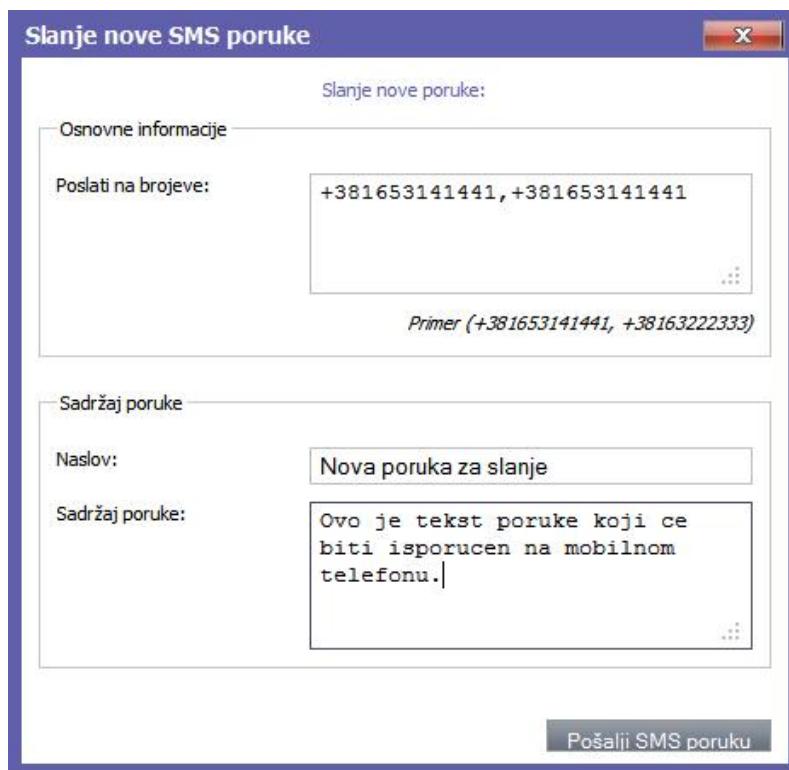
Slučaj korišćenja: Kreiranje nove SMS poruke i slanje poruke

Akter: korisnik koji ima pravo pristupa web aplikaciji

Preduslov: Potreno je da korisnik ima kreiran nalog na web servisu za pristup aplikaciji.

Osnovni scenario:

- 1) Akter iz glavnog naviogacionog menija bira opciju SMS poruke;
- 2) Nakon toga dobija se ekran sa prikazom liste SMS poruka i opcija Nova SMS poruka
- 3) Nakon što se klikne na opciju Nova SMS poruka dobija se ekran kao na sledećoj slici
- 4) Akter unosi sve potrebne podatke za slanje SMS poruke



Slika 39. Kreiranje i slanje nove SMS poruke

- 5) Nakon što akter unese sve potrebne podatke, klikom na dugme Pošalji SMS poruku, SMS poruka je poslata. Informacija o uspešnosti slanja SMS poruke biće ispisana na vrhu ekrana.

4.6.6. Slučajevi korišćenja – nbSMS android aplikacija

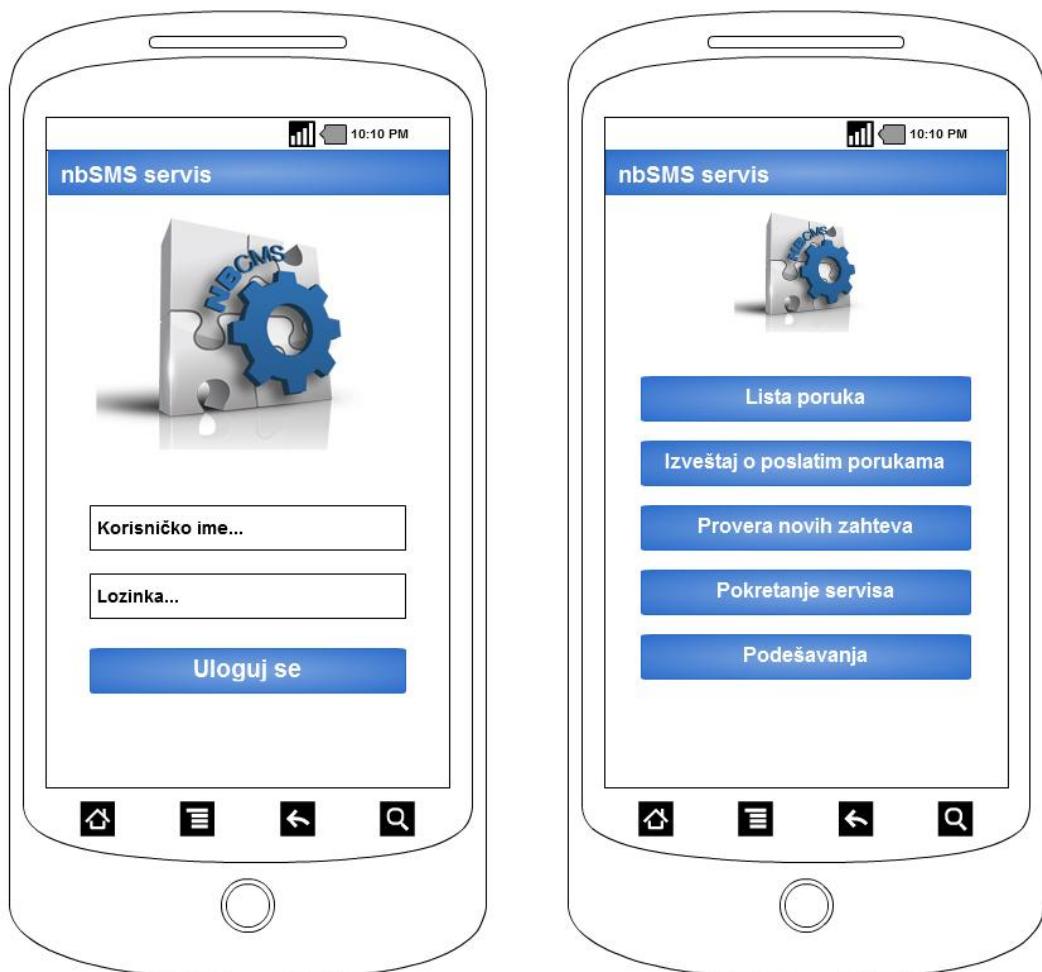
Slučaj korišćenja: Pokretanje aplikacije i početni ekran

Akter: korisnik koji ima instaliranu android aplikaciju na svom telefonu

Preduslov: Potreno je da korisnik ima kreiran nalog na web servisu za pristup aplikaciji.

Osnovni scenario:

- 1) Akter unosi podatke za pristup android aplikaciji, nakon unosa se potrebno je da se klikne na dugme Uloguj se;
- 2) Ako su uneti pogrešni podaci, akteru će biti prikazana poruka da unese ispravne podatke
- 3) Ako je akter uneo ispravne podatke, prikazuje se početni ekran aplikacije
- 4) Akter može da bira jednu od 5 operacija koje postoji u aplikaciji



Slika 40. Ekran za logovanje (levi) i početni ekran aplikacije (desno)

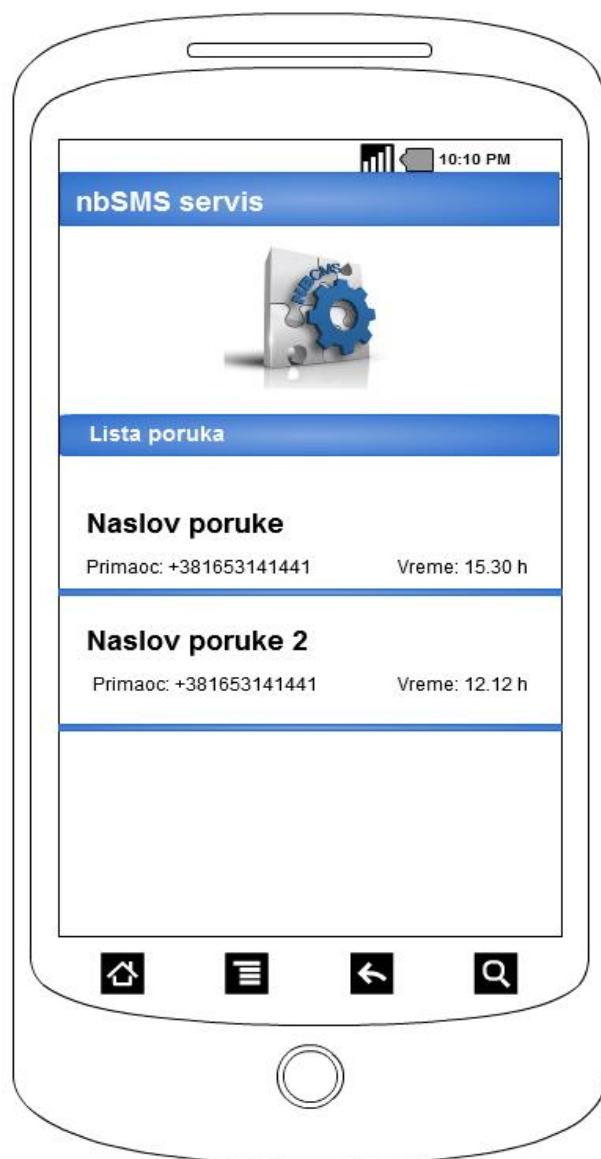
Slučaj korišćenja: Pregled liste poslatih SMS poruka

Akter: korisnik koji ima instaliranu android aplikaciju na svom telefonu

Preduslov: Potrebno je da se akter predhodno ulogovao na aplikaciju.

Osnovni scenario:

- 1) Izborom u glavnom meniju opciju Lista poruka, prikazuje se ekran kao na sledećoj slici
- 2) Akter ima mogućnost da pregleda listu poruka, vidi detalje poruke kao što je kome je poruka poslata, vreme slanja, status poruke.



Slika 41. Pregled liste poslatih poruka i poruka za slanje

Slučaj korišćenja: Podešavanje nbSMS servisa

Akter: korisnik koji ima instaliranu android aplikaciju na svom telefonu

Preduslov: Potrebno je da se akter predhodno ulogovao na aplikaciju.

Osnovni scenario:

- 1) Izborom u glavnom meniju opciju Podešavanja, prikazuje se ekran kao na predhodnoj slici sa desne strane
- 2) Akter unosi želje vrednosti u polja koja su poredviđena za to
- 3) Nakon unosa novih vrednosti akter ima mogućnost da snimi podešavanja
- 4) Ako su podešavajna uspešno snimljena, akteru se prikazuje poruka da su podaci uspešno snimljeni, u suprotnom se prikazuje informacija greški koja je nastala.



Slika 42. Podešavanje nbSMS servisa

5. Zaključak

U današnje vreme kada smo preplavljeni velikim brojem kojekakvih informacija, sve je teže i teže iz gomile podataka izvući onaj pravi, što ponekad može biti veoma veliki problem. Pomenuti problem se poslednjih godina mogao uočiti i u oblasti elektronskog obrazovanja, gde su web sajtovi preplavljeni velikim brojem informacija i gde sve više i više dolazi do izražaja pitanje kako da se pronađe adekvatan materijal s obzirom na potrebe, ciljeve i mogućnosti.

Usled sve većeg značaja i zastupljenosti elektronskog obrazovanja, prilagođavanje obrazovnih sadržaja i aktivnosti prema pojedinačnom korisniku, predstavlja jedno od najčešćih problema, koji se danas razmatraju.

S tom idejom, možemo govoriti o razvoju novih mehanizama za adaptaciju adekvatnog sadržaja, koji bi bio prilagođen potrebama, interesovnjima, mogućnostima itd. Mehanizam ima zadatak da iz gomile podataka koji se nalaze u bazi, izvuče onaj pravi i studentu u prvi plan istakne one materijale za koje je on zainteresovan, koji bi odgovorali njegovom stilu učenja, koji su adekvatni za njegovo trenutno znanje itd.

Aplikacija Mehanizam za adaptaciju sistema elektronskog obrazovanja u Moodle LMS razvijena korišćenjem uzora poput MVC i drugih, gde je jasno razdvojena poslovna logika od samog interfejsa aplikacije, gde bilo koje modifikacije na interfejsu ne utiču na samu poslovnu logiku kao i obrnuto. Aplikacija ima zadatak da adaptira objekte iz Moodle LMS sistema i da u adaptivnom režimu prikaže adaptirane objekte poput domaćih zadataka, testova, lekcija, pričaonica itd nad kojima je izvršena predhodna adaptacija. Kao neki od kriterijuma koji se koriste za zaključivanje su praćenje logova korisnika, ulaznih podataka koje student unosi popunjavanjem ankete kao i nekih rezultata na nekom od meta kurseva.

Mehanizam za adaptaciju sistema elektronskog obrazovanja u Moodle LMS-u je generički, što dalje govori da je profesorima ostavljena mogućnost da sami mogu kreiraju razne implementacije adaptivnog režima zavisno od trenutnih ciljeva.

Napravljena je jasna razlika od samog razvoja mehanizma za adaptaciju i njegove implemetacije, čime je stvorena podloga za dalje poboljšanje same aplikacije.

Aplikacija je pisana korišćenjem objektno orijentisanog koncepta programiranja, pisanjem modula, gde je svaki modul nezavisan od drugih, čime je obezbeđeno da se kasnije lako može mehanizam proširivati i nadograđivati. Trenutna verzija mehanizma ima mogućnost da izvrši adaptaciju kursa, grupe, studenta, resursa, aktivnosti, inteligentno razmesti studente po grupama na osnovu raznih ulaznih podataka i prikaže adaptirane sadržaje studentu.

Jedan od pravaca proširivanja aplikacije mogao bi da bude automatsko kreiranje i adaptacija grupe na osnovu analize i drugih ulaznih podataka, gde bi mehanizam imao zadatku da analizira pomenute informacije i sam dođe do zaključka kako da adaptira grupu shodno kakvim studentima raspolaze. S obzirom na modularnost aplikacije, nadogradnja modula ne bi uticala na rad samog mehanizma.

6. Prilog – Reprezentativni delovi koda

Kontroler kompletne aplikacije:

```
<?php
defined( 'MEHANIZAM_ADAPTACIJE' ) or die ( 'Zabranjen pristup' );
/**
 * Description of ControllerAM
 * @author Nastevski Aleksandar
 */
class ControllerAM {

    protected $naziv;

    function __construct($naziv) {
        global $CFG, $COURSE, $USER, $SESSION;
        $this->setNaziv($naziv);
        require_login($COURSE);
    }

    function getNaziv(){
        return $this->naziv;
    }

    function setNaziv($naziv){
        $this->naziv = $naziv;
    }

    //f-ja koja izvrsava kontrolera
    function execute(){
        $task    = strtolower(optional_param('task', "", PARAM_RAW));
        $no_html = optional_param('no_html', 0, PARAM_INT);
        $tmpl    = optional_param('tmpl', "", PARAM_RAW);

        ob_start();

        if (method_exists($this, $task))
            $ret = $this->$task();
        else
            $ret = $this->show();

        if ($ret !== null)
            echo json_encode($ret);

        $output = ob_get_contents();
        ob_end_clean();

        if ($no_html) {
            echo $output;
        } elseif ( $tmpl == 'component' ) {
            echo "<html>\n<head>\n";
            echo "<title>".$this->title."</title>\n";
            echo "</head><body>\n";
            echo $output;
            echo "</body>\n</html>";
        } else {
            print_header(setNaslov(), setNaslov(), "");
            print_heading(setNaslov());
            echo $output;
            print_footer();
        }
    }
}
```

```

}

//f-ja koja prikazuje prikaz korisniku, odnosno pokreće view controllera
function show($viewController){
    if (file_exists('controllers/'.$viewController.'Controller.php')) {
        require_once('controllers/'.$viewController.'Controller.php');

        $controllerName = ucfirst($viewController).'Controller';
        if (class_exists($controllerName)) {
            $controller = new $controllerName();
            return $controller->show();
        } else {
            die( 'Class not found: '.$controllerName);
        }
    } else {
        die( "Controller $controllerName doesn't exist" );
    }
    return false;
}

}
?>

```

Glavni kontroler mehanizma za adaptaciju:

```

<?php
defined( 'MEHANIZAM_ADAPTACIJE' ) or die ( 'Zabranjen pristup' );
/**
 * Description of MehanizmiController
 *
 * @author Nastevski
 */
class MehanizmiController extends ControllerAM{

    var $naziv;

    function __construct() {
        $this->naziv = 'Mehanizmi';
    }

    //prikaz mehanizma za adaptaciju
    function show(){

        global $CFG;
        $viewController = strtolower(optional_param('view', 'error', PARAM_RAW));

        switch ($viewController) {
            case 'ciljevi':
                setNaslov('Mehanizam za definisanje ciljeva');
                break;
            case 'anketa':
                setNaslov('Mehanizam za definisanje anketa');
                break;
            case 'grupa':
                setNaslov('Mehanizam za adaptaciju grupe');
                break;
            case 'kurs':
                setNaslov('Mehanizam za adaptaciju kursa');
                break;
            case 'resurs':
                setNaslov('Mehanizam za adaptaciju resursa');
                break;
            case 'aktivnost':
                setNaslov('Mehanizam za adaptaciju aktivnosti');
                break;
            default:

```

```
$viewController = 'error';
setNaslov('Pokušavate da pristupite nepostojećoj stranici!');
break;
}

return parent::show($viewController);
}

?>
```

Pokretanje mehanizma za adaptaciju:

```
//ucitavanje podesavanja mehanizma za adaptaciju
require_once('config.php');

$mehanizamZaAdaptaciju = new MehanizmiController();
$mehanizamZaAdaptaciju->execute();
```

Prikaz config.php fajla koji učitava osnovne fajlove aplikacije za adaptaciju:

```
<?php
//definisanje promenljive za proveru pristupa fajlovima
define( 'MEHANIZAM_ADAPTACIJE', 1 );

//ucitavanje glavnog config fajla Moodle-a
require_once('../config.php');

//ucitavanje osnovnih fajlova mehanizma za adaptaciju
require_once('lib/ObjectAM.php');
require_once('lib/ControllerAM.php');
require_once('lib/ViewControllerAM.php');
require_once('lib/ModelAM.php');

require_once('MehanizmiController.php');

//ucitavanje main_liba mehanizma za adaptaciju
require_once('lib/main_lib.php');
?>
```

Prikaz glavnog modela aplikacije za adaptaciju:

```
<?php
defined( 'MEHANIZAM_ADAPTACIJE' ) or die ( 'Zabranjen pristup' );
/** 
 * Description of ModelAM
 *
 * @author Nastevski
 */
class ModelAM {
    protected $nazivTabele;

    function __construct($nazivTabele) {
        $this->setNazivTabele($nazivTabele);
    }

    protected function unos($objekat){
        $id = insert_record($this->getNazivTabele(), $objekat);
        if(!$id){
            return false;
        }
        return $id;
    }

    protected function unosUTabelu($nazivTabele = "", $objekat = null){
```

```
if($nazivTabele == ""){
    return false;
}
if($objekat == null){
    return false;
}
$id = insert_record($nazivTabele, $objekat);
if(!$id){
    return false;
}
return $id;
}

protected function promena($objekat){
    $id = $objekat->id;
    if(!update_record($this->getNazivTabele(), $objekat)){
        return false;
    }
    return $id;
}

protected function promeniStatus($objekat){
    return $this->promena($objekat);
}

protected function getNazivTabele(){
    return $this->nazivTabele;
}

protected function setNazivTabele($nazivTabele){
    $this->nazivTabele = $nazivTabele;
}

protected function obrisiRekorde($atribut, $vrednost){
    return delete_records($this->getNazivTabele(), $atribut, $vrednost);
}

protected function obrisiUTabelu($nazivTabele = "", $atribut, $vrednost){
    if($nazivTabele == ""){
        return false;
    }
    return delete_records($nazivTabele, $atribut, $vrednost);
}
?>
```

Prikaz kontrolera interfejsa, odnosno kontrolera pogleda:

```
<?php
defined( 'MEHANIZAM_ADAPTACIJE' ) or die ( 'Zabranjen pristup' );
/**
 * Description of ViewControllerAM
 *
 * @author Nastevski
 */
class ViewControllerAM extends ObjectAM{

    var $type = 'html';

    //setuje promenljivu na ekran
    function setVar($name, $value){
        $this->$name = $value;
    }

    //vraca ime ViewControlera odnosno klase
    function getName(){
        $res = explode('Controller', get_class($this));
        return strtolower($res[0]);
    }

    //prikan podrazumevanog html-a, ucitavanje osnovnih elemenata
    function show($tmpl=null){
        if (!$tmpl) $tmpl = $this->getName();

        $tmpl_file = 'views/'.$this->getName().'.'.$tmpl.'.'.$this->type.'.php';
        $css_file = 'views/'.$this->getName().'.'.$tmpl.'.css';
        $js_file = 'views/'.$this->getName().'.'.$tmpl.'.js';

        if (file_exists($tmpl_file)) {
            require( $tmpl_file );

            if ($this->type == 'xml'){
                if ( stristr($_SERVER["HTTP_ACCEPT"],"application/xhtml+xml") ) {
                    header("Content-type: application/xhtml+xml; charset=utf-8");
                } else {
                    header("Content-type: text/xml; charset=utf-8");
                }
            }else {
                if (file_exists($css_file)) import_css($css_file);
                if (file_exists($js_file)) import_js($js_file);
            }
        } else {
            echo 'Template "'.$tmpl_file.'" doesn\'t exist!';
            exit(1);
        }
    }
?>
```

Prikaz kontrolera za manipulaciju sa mehanizmom za definisanje ciljeva:

```
<?php
defined( 'MEHANIZAM_ADAPTACIJE' ) or die ( 'Zabranjen pristup' );
/**
 * Description of CiljeviController
 * CiljeviController nasledjuje klasu controllora aplikativnog nivoa
 * @author Nastevski
 */
class CiljeviController extends ViewControllerAM{

    var $naziv = 'ciljevi';
    protected $cijeljModel;
    protected $kursModel;

    function __construct() {
        $this->importModel();
        $this->cijeljModel = new Cilj();
        $this->kursModel = new Kurs();
    }

    function getNazivViews(){
        return $this->naziv;
    }

    function show(){

        $no_html = optional_param( 'no_html', 0, PARAM_INT );
        $task = optional_param( 'task', 'xml', PARAM_ALPHA );

        if ( !$no_html ) {

            insert_jquery_js_and_css();
            insert_jqgrid_js_and_css();
            insert_brze forme_and_css();
            insert_autocomplete();
            insert_html_lib_and_css();

            $tipovi_ciljeva = $this->cijeljModel->getTipoveCilja();
            $this->setVar( 'tipovi_ciljeva', $tipovi_ciljeva );

            $formaCiljUnos = $this->cijeljModel->kreirajFormuZaUnos();
            echo $formaCiljUnos;

        } else {

            switch($task) {
                case 'getlistukurseva':
                    $search = optional_param( 'q', "", PARAM_RAW );
                    $records = $this->kursModel->getKurseve($search);
                    //echo json_encode($records);
                    return $records;
                    break;
                case 'snimicilj':
                    $tip = optional_param( 'am_cilj_tip_cilja', -1, PARAM_INT );
                    $naziv = optional_param( 'am_cilj_naziv', "", PARAM_RAW );
                    $text = optional_param( 'am_cilj_text', "", PARAM_RAW );
                    $stavke = optional_param('am_cilj_moguci_odgovori', "", PARAM_RAW );
                    $kurs_id = optional_param( 'am_cilj_kurs_id', -1, PARAM_INT );

                    return $this->snimiCilj($tip, $naziv, $text, $stavke, $kurs_id);
                    break;
                case 'status':
                    $cijeljId = optional_param( 'id', -1, PARAM_INT );
                    $statusId = optional_param( 'status', -1, PARAM_INT );
            }
        }
    }
}
```

```

        return $this->promeniStatusCilja($cijId, $statusId);
        break;
    case 'getxmlciljevi':
        $page    = optional_param( 'page', 1, PARAM_INT );
        $limit   = optional_param( 'rows', 10, PARAM_INT );
        $sidx    = optional_param( 'sidx', 'id', PARAM_RAW );
        $sord    = optional_param( 'sord', 'DESC', PARAM_RAW );
        $search   = optional_param( 'search', "", PARAM_RAW );

        $filter = array(
            'c.tip_cilja_id' => optional_param( 'tip_cilja', 0, PARAM_INT ),
        );

        $records = $this->ciljModel->get_xml_list($page, $limit, array($sidx => $sord), $filter, $search);

        $this->setVar( 'records', $records );
        $this->setVar( 'ciljModel', $this->ciljModel );
        $this->type = 'xml';
        $tmpl = $this->getNazivViews();
        break;
    default:
        $page    = optional_param( 'page', 1, PARAM_INT );
        $limit   = optional_param( 'rows', 10, PARAM_INT );
        $sidx    = optional_param( 'sidx', 'id', PARAM_RAW );
        $sord    = optional_param( 'sord', 'DESC', PARAM_RAW );
        $search   = optional_param( 'search', "", PARAM_RAW );

        $filter = array(
            '/t.id' => $code,
        );

        $records = $this->ciljModel->get_xml_list($page, $limit, array($sidx => $sord), $filter, $search);

        $this->setVar( 'records', $records );
        $this->type = 'xml';
        $tmpl = 'ciljevi';
        break;
    }
}

return parent::show($tmpl);
}

function importModel(){
    import_model('Cilj');
    import_model('Kurs');
}

function snimiCilj($tip , $naziv, $text, $stavke = "", $kurs_id = -1){
    global $USER;

    if($tip == -1){
        return false;
    }

    $noviCilj = new object();
    $noviCilj->tip_cilja_id = intval($tip);
    $noviCilj->naziv      = $naziv;
    $noviCilj->text_pitanja = $text;
    $noviCilj->user_dodao  = $USER->id;

    if(!$this->proveriValidnostZaNoviCilj($tip, $stavke, $kurs_id)){
        return false;
    }

    $cijId = $this->ciljModel->unos($noviCilj);

    if(!$cijId){

```

```

        return false;
    }

$stavke = $this->snimiStavkeCilja($ciljId, $tip, $stavke, $kurs_id);

if(!$stavke){
    $this->ciljModel->obrisi($ciljId);
    return false;
}

return true;
}

function snimiStavkeCilja($ciljId, $tip, $stavke = "", $kursId = -1){

switch($tip) {
    case 1:
        if($ciljId == -1){
            return false;
        }
        $novaStavka = new object();
        $novaStavka->cilj_id = intval($ciljId);
        $novaStavka->text_stavke = "Odgovor";
        return $this->ciljModel->unosStavkeCilja($novaStavka);
        break;
    case 2:
        if($ciljId == -1){
            return false;
        }
        $textStavkeNiz = explode(getTipOdvajanjaStavke(), $stavke);
        $stavkaldString = "";
        foreach($textStavkeNiz as $id => $stavka){
            $novaStavka = new object();
            $novaStavka->cilj_id = intval($ciljId);
            $novaStavka->text_stavke = $stavka;
            $stavkald = $this->ciljModel->unosStavkeCilja($novaStavka);
            if(!$stavkald){
                $this->obrisiStavkeCiljaString($stavkaldString);
                return false;
            }
            $stavkaldString = $stavkaldString.$stavkald. ",";
        }
        return true;
        break;
    case 3:
        if($ciljId == -1){
            return false;
        }
        $textStavkeNiz = explode(getTipOdvajanjaStavke(), $stavke);
        $stavkaldString = "";
        foreach($textStavkeNiz as $id => $stavka){
            $novaStavka = new object();
            $novaStavka->cilj_id = intval($ciljId);
            $novaStavka->text_stavke = $stavka;
            $stavkald = $this->ciljModel->unosStavkeCilja($novaStavka);
            if(!$stavkald){
                $this->obrisiStavkeCiljaString($stavkaldString);
                return false;
            }
            $stavkaldString = $stavkaldString.$stavkald. ",";
        }
        return true;
        break;
    case 4:
        if($ciljId == -1){
            return false;
        }
}

$novaStavka = new object();

```

```

$novaStavka->cilj_id = intval($ciljId);
$novaStavka->text_stavke = "Rezultat";
$stavkald = $this->ciljModel->unosStavkeCilja($novaStavka);

if(!$stavkald){
    return false;
}

$kursCilja = new object();
$kursCilja->cilj_id = intval($ciljId);
$kursCilja->course_id = intval($kursId);

$kursCiljald = $this->ciljModel->unosKursStavkeCilja($kursCilja);

if(!$kursCiljald){
    $this->ciljModel->obrisiStavkuCilja($stavkald);
    return false;
}
return true;
break;
}
return true;
}

function proveriValidnostZaNoviCilj($tip, $stavke = "", $kursId = -1){

switch($tip) {
case 2:
if($stavke == ""){
    return false;
}
break;
case 3:
if($stavke == ""){
    return false;
}
break;
case 4:
if($kursId == -1){
    return false;
}
break;
}
return true;
}

function obrisiStavkeCiljaString($stavkeString){
$stavkeString = substr($stavkeString, 0, -1);
$stavke = explode(',', $stavkeString);
foreach($stavke as $id => $stavkald){
    $this->ciljModel->obrisiStavkuCilja($stavkald);
}
}

function promeniStatusCilja($id, $status){
$ciljStatus = new object();
$ciljStatus->id = intval($id);
$ciljStatus->status = intval($status);
return $this->ciljModel->promeniStatus($ciljStatus);
}

}
?>
```

Prikaz modela za Cilj:

```
<?php
/**
```

```

/* Description of Cilj
 * @author Nastevski
 */
class Cilj extends ModelAM{

    var $nazivTabele = 'am_cilj';

    function __construct() {
        parent::__construct($this->getNazivTabele());
    }

    function getTipoveCilja(){
        return get_records('am_tip_cilja');
    }

    function getNazivTabele(){
        return $this->nazivTabele;
    }

    function get_xml_list($page = 1, $limit = 10, $orderby = array(), $filter = array(), $search = ""){
        global $CFG;
        $sql = "SELECT
            c.*,
            tc.naziv as naziv_tipa_cilja,
            CONCAT(u.firstname, ' ', u.lastname ) as ime_prezime

        FROM
            ".$CFG->prefix.$this->getNazivTabele()." c
            left JOIN ".$CFG->prefix."am_tip_cilja tc ON tc.id = c.tip_cilja_id
            left JOIN ".$CFG->prefix."user u ON u.id = c.user_dodao
        ";

        foreach($filter as $id => $value){
            if (!$value) unset($filter[$id]);
        }

        if ($filter or $search) {
            $first = true;
            $sql .= "WHERE ";
            if ($filter) {
                foreach($filter as $col => $value) {
                    if ($first) $first = false;
                    else $sql .= " AND ";
                    if(strpos($value, ' '))
                        $cond = $value;
                    else
                        $cond = "= " . mysql_real_escape_string($value) . "";
                    $sql .= "(".$col . " ".$cond.")";
                }
            }

            if ($search) {
                if (!$first) $sql .= " AND ";
                $sql .= " (CONCAT(c.naziv, ' ', c.text_pitanja ) LIKE '%" . mysql_real_escape_string($search) . "%') ";
            }
        }

        $spisak = pagedQuery($sql, $page, $limit, $orderby);

        return $spisak;
    }

    function unos($noviCilj){
        return parent::unos($noviCilj);
    }

    function obrisi($id){
        return parent::obrisiRekorde('id', $id);
    }
}

```

```

function unosStavkeCilja($novaStavkaCilja){
    return parent::unosUTabelu('am_cilj_stavke', $novaStavkaCilja);
}

function obrisiStavkuCilja($stavkald){
    return parent::obrisiUTabelu('am_cilj_stavke', 'id', $stavkald);
}

function unosKursStavkeCilja($noviKurs){
    return parent::unosUTabelu('am_cilj_kurs_info', $noviKurs);
}

function getIkonaFormaPromenaCilja($ciljId){
    return '<a style="background-image:url('.getWwwRootMehanizma()."images/ikone/edit.png"); width="30" height="25" href="javascript:otvoriFormuPromenaCilja('.intval($ciljId).')"> </a>';
}

function getIkonaStatusCilja($status, $ciljId){
    return getIkonuZaStatus($status, 'javascript:promeniStatusCilja('.$.ciljId.', '.getSuprotniStatus($status).)');
}

function promeniStatus($ciljStatus){
    return parent::promeniStatus($ciljStatus);
}
?>

```

Deo koda koji zavisno od setovanih podatka kupi iz baze delimično zadovoljene odgovorom, odnosno odgovore koje sadrže deo zadatog. Ovaj deo koda se koristi kada mehanizam analizira tekst odgovora koji je student dao u slobodnom formatu prilikom popunjavanja ankete.

```

if ($search) {
    if (!$first) $sql .= " AND ";
    $sql .= " (CONCAT(c.naziv, ' ', c.text_pitanja ) LIKE '%" . mysql_real_escape_string($search) . "%' )";
}

```

Deo koda koji se koristi u mehnizmi za adaptaciju kada odgovori moraju u potpunosti da se poklapaju ili da upadaju u neki od skupova:

```

if ($filter) {
    foreach($filter as $col => $value) {
        if ($first) $first = false;
        else $sql .= " AND ";
        if(strpos($value, ' '))
            $cond = $value;
        else
            $cond = "=" . mysql_real_escape_string($value) . "";
        $sql .= (".$col . ".$cond.");
    }
}

```

Prikaz dela koda koji kreira XML fajl koji se koristi za tabelarni prikaz podataka u Jqgridu kao i prilikom učitavanja podataka iz baze preko AJAX-a:

```
<?php
defined( 'MEHANIZAM_ADAPTACIJE' ) or die ( 'Zabranjen pristup' );
echo '<?xml version="1.0" encoding="utf-8"?>.\n';
?>
<rows>
<page><?=$this->records->page;?></page>
<total><?=$this->records->total;?></total>
<rows><?=$this->records->rows;?></rows>
<records><?=$this->records->records;?></records>
<? foreach($this->records->rows as $row): ?>
<row id='<?=$row->id;?>'>
    <cell><?='<![CDATA['.$row->naziv.']]>';?></cell>
    <cell><?='<![CDATA['.$row->naziv_tipa_cilja.']]>';?></cell>
    <cell><?='<![CDATA['.$row->text_pitanja.']]>';?></cell>
    <cell><?='<![CDATA['.$row->ime_prezime.']]>';?></cell>
    <cell><?='<![CDATA['.$row->datum_unosa.']]>';?></cell>
    <cell><?='<![CDATA['.$this->ciljModel->getIkonaStatusCilja($row->status, $row->id).']]>';?></cell>
    <cell><?='<![CDATA['.$this->ciljModel->getIkonaFormaPromenaCilja($row->id).']]>';?></cell>
</row>
<? endforeach; ?>
</rows>
```

Deo koda koji je napisan u JavaScriptu koji služi da preko AJAX-a snimi novi cilj u bazu podataka:

```
function snimiNoviCilj(data){
    $("#form_am_cilj :input").attr( 'disabled', 'disabled' );
    setPorukuNaFormuCilj(getTextPodaciSeSnimaju());
    $.ajax({
        type: "POST",
        url: urlSniMiCilj,
        data: data,
        dataType: 'json',
        success: function(ret) {
            $("#form_am_cilj :input").removeAttr( 'disabled' );
            if(ret) {
                setPorukuNaFormuCilj(getTextUnosOk());
                display_msg(getTextUnosOk());
                gridReload();
                resetujFormuCilj();
            } else {
                setPorukuNaFormuCilj(getTextUnosError());
                display_msg(getTextUnosError(), 'error');
            }
        }
    });
}
```

7. Literatura i reference

- [1] Doc. Dr Marijana Despotović-Zrakić, Prof. Dr Božidar Radenković, Zvanična literatura i materijali sa predmeta Internet tehnologije na četvrtoj godini studija na Fakultetu Organizacionih Nauka, 2010.
- [2] Dušan Barać, „Razvoj modela i servisa portala za adaptivno elektronsko obrazovanje“, 2010.
- [3] Jason Cole and Helen Foster, Using Moodle, Teaching with the Popular Open Source Course Management System, Second edition, November 2007;
- [4] David hunter, Kurt Cagle, Dave gibbson, Nikola Ozu, Jon Pinnock, Paul Spencer, XML od početka, CET 2001;
- [5] Web portal, Moodle, <http://www.moodle.org>, poslednja poseta 30.09.2010.;
- [6] Web portal, PHP, <http://www.php.net>, poslednja poseta 30.09.2010.;
- [7] Web portal, W3Schools, www.w3schools.com, poslednja poseta 30.09.2010.;
- [8] Web portal, Wikipedia, <http://www.wikipedia.org>, poslednja poseta 30.09.2010.;
- [9] Web portal, AJAX, <http://www.ajax.org>, poslednja poseta 30.09.2010.;
- [10] Jquery biblioteka, <http://docs.jquery.com>, poslednja poseta 30.09.2010.;
- [11] Jqgrid biblioteka, <http://www.trirand.com/blog/>, poslednja poseta 30.09.2010.;
- [12] Demo verzija Jqgrid biblioteke, <http://www.trirand.com/blog/jqgrid/jqgrid.htm>, poslednja poseta 30.09.2010.;
- [13] JavaScript, <http://www.javascript.com>, poslednja poseta 30.09.2010.;
- [14] Stefan Mischook, Learn Object Oriented Programming (OOP) in PHP, 2007.