

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

Model sustava za upravljanje razredom

Kristijana Paskvali

Mentorica: doc.dr.sc. Ani Grubišić

Split, rujan 2015

Sadržaj:

Uvod	7
1. Sustavi e-učenja	8
2. Primjeri sustava za upravljanje razredom i njihove osnovne karakteristike	12
2.1. Moodle	12
Proširivanje funkcionalnosti sustava	15
Dodatci	16
2.2. .LRN (dot learn)	16
2.3. Opensis	17
2.4. Fedena	20
2.5. GPA Teacher	21
3. Metode razvoja sustava za upravljanje razredom	24
3.1. Agile Scrum razvojna metoda sustava za upravljanje nastavom	27
4. Model sustava za upravljanje razredom	35
Zahtjevi korisnika	35
Dijagram slučajeva korištenja	36
4.1. Model baze podataka	39
4.2. Zahtjev za okruženjem potrebnim za realizaciju projekta	40
4.3. Prilagodba sučelja phpMyreservation sučelju sustava za administriranje nastavom	42
4.4. Kreiranje kolegija	43
4.5. Dodavanje stavki kolegiju	45
4.6. Pridruživanje studenata kolegiju	49
4.7. Evidencija izostanaka i testova	49
4.8. Prijava kolokvija	50

4.9. Prijava ispita	50
4.10. Grupiranje studenata u ispitne grupe	51
4.11. Rezerviranje dvorana i pridruživanje kolegija rezervacijama	58
Zaključak	60
5. Literatura	61
Sažetak.....	62
Summary.....	63
Skraćenice.....	64
Privitak	65

Popis slika:

Slika 1:1 LCMS- deploy-LMS	10
Slika 1:2 Tijek rada stvaranja multimedijskog nastavnog materijala.....	11
Slika 1:3 Grafički prikaz razmjene podataka između SIS LMS i LCMS sustava	11
Slika 2:1 Izgled Moodle sučelja	13
Slika 2:2 Razlike u privilegijama uloga	14
Slika 2:3 Dodatni moduli Moodle platforme.....	15
Slika 2:4 Moduli sustava DotLRN	17
Slika 2:5 Struktura OpenSIS sustava.....	18
Slika 2:6 Primjer prikaza organizacijske tablice iz sustava OpenSIS	18
Slika 2:7 Točke integracije između OpenSIS i Moodle sustava	19
Slika 2:8 Osnovni moduli koje nudi Fedena komercijalni sustav	20
Slika 2:9 Sučelje sustava GPA Teacher	21
Slika 3:1 Slijed aktivnosti u razvoju programske podrške	24

Slika 3:2 Osnovni modeli životnog ciklusa razvoja programske podrške.....	25
Slika 3:3 Unified Process	26
Slika 3:4 Faze Scrum razvojnog ciklusa.....	30
Slika 3:5 Sprint	32
Slika 3:6Primjer Backlog	33
Slika 3:7 Workflow	34
Slika 4:1 Prikaz dijagrama slučajeva korištenja	36
Slika 4:2 Prikaz razriješenih slučajeva korištenja instalacijom jqGrid kontrole	38
Slika 4:3 Model baze podataka.....	39
Slika 4:4 Ilustracija zahtjeva za okruženjem	40
Slika 4:5 Dijagram aktivnosti logiranje/registriranje	40
Slika 4:6 Izgled formi logiranje i registriranje	41
Slika 4:7 Ilustracija prilagodbe sučelja.....	42
Slika 4:8 Lista kreiranih kolegija.....	43
Slika 4:9 Funkcionalnost klase Prikaz.....	47
Slika 4:10 Prikaz nove stavke.....	48
Slika 4:11 Identifikacija korisnika od strane LMS sustava	48
Slika 4:12 primjer tablice generirane pomoću jqGrid kontrole	49
Slika 4:13 Forma za kreiranje ispitnih grupa.....	51
Slika 4:14 Grupiranje za pismeni ispit	52
Slika 4:15 Tablica grupiranja za usmeni ispit u bazi podataka	56
Slika 4:16 Prikaz grupiranja za pismeni ispit	57
Slika 4:18 Rezerviranje dvorana i pridruživanje kolegija rezervacijama.....	58

Popis kodova:

Kôd 5.3.1 – Podešavanje prikaza sučelja u datoteci cnfig/js/main.js	43
Kôd 5.4.1 – Kreiranje popisa kolegija sa pripadajućim kartisama i sadržajem.....	44
Kôd 5.5.1 – Dodavanje stavke u HTML preložak stavki	46
Kôd 5.5.2 – Dodavanje podataka putanji	47
Kôd 5.10.1 – Funkcija grupiranja za pismeni ispit.....	53
Kôd 5.10.2 – Funkcija grupiranja za usmeni ispit.....	56

Popis tablica:

Tabela 3-1 Osnovne funkcionalnosti navedenih sustava.....	23
---	----

Uvod

Pristup otvorenog koda (eng. open source) je zasnovan na programskom kodu koji je dostupan javnosti, dopuštajući joj slobodu modificiranja, integriranja i nadogradnje izvornog koda. Modificirani kod može biti podijeljen s javnošću na jednak način, ili po određenim uvjetima prodan [1]. Mnoge organizacije danas koriste proizvode iz sfere otvorenog koda prvenstveno stoga što prestaju biti ovisne o davatelju komercijalne usluge u svrhu održavanja, servisiranja ili podrške pri pojavi poteškoća.

„E-učenje je korištenje multimedije i Interneta u svrhu poboljšanja kvalitete učenja - omogućavanjem pristupa udaljenim izvorima i uslugama i omogućavanjem suradnje i komunikacije na daljinu.”[8]

U ovom radu razmotrit ćemo osnovna svojstva sustava namijenjenih administraciji nastave i njenom sadržaju, te Scrum metodologiju razvoja pojedinih dodataka LMS sustavima otvorenog koda.

Razmotrit ćemo neke funkcionalnosti sustava otvorenog koda orijentirane na obrazovanje kroz upravljanje kolegijima i administraciju nastave, te besplatne on-line sustave koji potpomažu u realizaciji spomenutih stavki.

Razmotrit ćemo specifične administrativne zahtjeve korisnika koji trenutno nisu obuhvaćeni u niti jednom LMS ni SIS rješenju, niti u e-Dnevniku i razviti sustav koji zadovoljava navedene zahtjeve uz integraciju i modifikaciju dviju kontrola otvorenog koda i široke podrške zajednice otvorenog izvora pri pronalaženju rješenja za specifične probleme.

Kroz projekt rada ćemo razviti sustav prema konceptualnom modelu objektno orijentirane analize i dizajna. Gotovu aplikaciju postaviti ćemo na besplatni servis za udomljavanje stranica gdje ju je moguće testirati.

1. Sustavi e-učenja

Arhitektura sustava e-učenja uključuje dizajn, opravdne funkcionalnosti i iskoristivosti, odmjerenu estetiku sučelja. Razumijevanje potreba takvog specifičnog sustava zahtijeva suradnju i učinkovito komuniciranje dizajnera, programera i krajnjih korisnika u ostvarenju zadanih ciljeva konkretnog sustava. Zbog složenosti suradničke mreže neophodni su jasno definirani zahtjevi, namjene procesa, funkcionalnosti, sučelje i objašnjenja korisnosti određenih intervencija.

Razumijevanje arhitekture sustava e-učenja omogućava dizajnerima i programerima uspješno integriranje širokog raspona funkcionalnosti koje krajnji korisnici zahtijevaju. Krajnji korisnici obuhvaćaju sve sudionike sustava : pokrovitelje kurikuluma, dizajnere, programere, instruktore (nastavnike), autore, stručnjake iz raznih područja sadržaja, studente i administratore.

Učinkovito dizajnirani sustav rješava zahtjeve svih krajnjih korisnika ako su jasno definirani prije projektiranja tehnološke infrastrukture , radnih procesa i omogućavanja takvih radnih procesa.

Dobro dizajnirani sustav e-učenja zahtijeva [10]:

- Otvorenu arhitekturu uz podršku uspostavljenih industrijskih standarda, prvenstveno u svrhu omogućavanja dodavanja novih budućih modula
- Skalabilnost, kako bi tehnička infrastruktura mogla rasti zajedno s brojem krajnjih korisnika i sofisticiranijim zahtjevima
- Potencijalni globalne dosege, kako bi sustavi e-učenja mogli biti korišteni u van obrazovnim okruženjima, u svrhu poboljšanja komunikacije i suradnje različitih grupacija.
- Mogućnost integracije novih modula koja zahtijeva jednakost postavljenih prednjih i pozadinskih veza s poslovnim resursima

- Prilagodljivost kao otvorenost prema novim tehnologijama i primjenu novih najboljih praksi
- Kontrolu složenosti sustava kako bi novi zahtjevi mogli biti realizirani brzo i pravovremeno

Web bazirani sustavi omogućuju pristup svakom

Sustav za upravljanje učenjem (eng. Learning management system LMS) je programska podrška koja omogućava administriranje, dokumentiranje, praćenje raznih aspekata procesa učenja, izvještavanje i dostavu sustava za e-učenje, ali ne služi kreiranju sadržaja učenja. LMS je okvir koji rukovodi svim aspektima procesa učenja. SCORM je standard dostave nastavnog materijala za LMS sustave.

Skup normi SCORM (Sharable Content Object Reference Model) osigurava realizaciju skupa tehničkih standarda i specifikacija za web bazirani sustav e-učenja. Nastavni sadržaji kreirani različitim alatima mogu biti dostavljeni LMS sustavu po standardu SCORM interoperabilnosti.

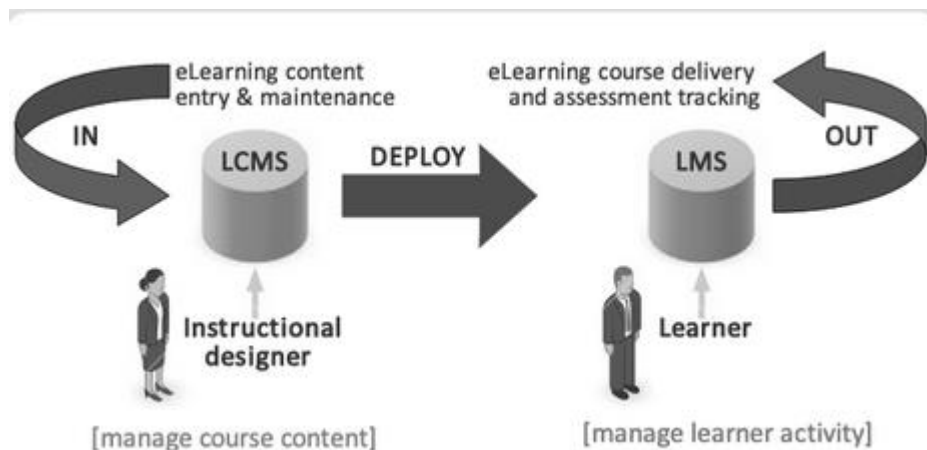
LMS sustavi otvorenog koda su potpuno prilagodljivi i lako mogu biti dizajnirani prema zahtjevima obrazovne ustanove, ili neke druge organizacije. Mogu se pretvoriti u društvene platforme za učenje. Može se stvoriti zasebna on-line zajednica putem LMS sustava koji sadrži chat, blog, spoj na društvene mreže.

Osnovna arhitektura LMS sustava se sastoji od portala za učenje, sustava za autorizaciju, autentifikaciju, programa za analizu i kreiranje izvještaja, akademske administracije koja obuhvaća dodavanje profila korisnika, dodjeljivanje uloga korisnicima, upravljanje akademskom strukturom, upravljanjem rasporedom nastave.

SIS (Student Information System) je informacijski sustav obrazovne ustanove i sadrži detaljne podatke o studentima, kao što su: alergije, cijepljenja, prethodno školovanje, podatci o roditeljima, izostancima, disciplini, raspored sati, podatke o zaposlenicima, računovodstvene podatke. SIS upravlja osnovnim osjetljivim podacima i zato zahtijeva visoku sigurnosnu zaštitu.

Sustav za upravljanje sadržajem učenja (eng. Learning content management system LCMS) je šire okruženje koje okuplja različite korisnike, razvojne programere, autore, oblikovatelje

nastavnih materijala (eng. instructional designer), eksperte iz raznih područja koji sudjeluju u stvaranju tehnologije e-učenja. LCMS je prvenstveno orijentiran na razvoj, upravljanje i objavljivanje sadržaja koji mogu biti dostavljeni putem LMS sustava.

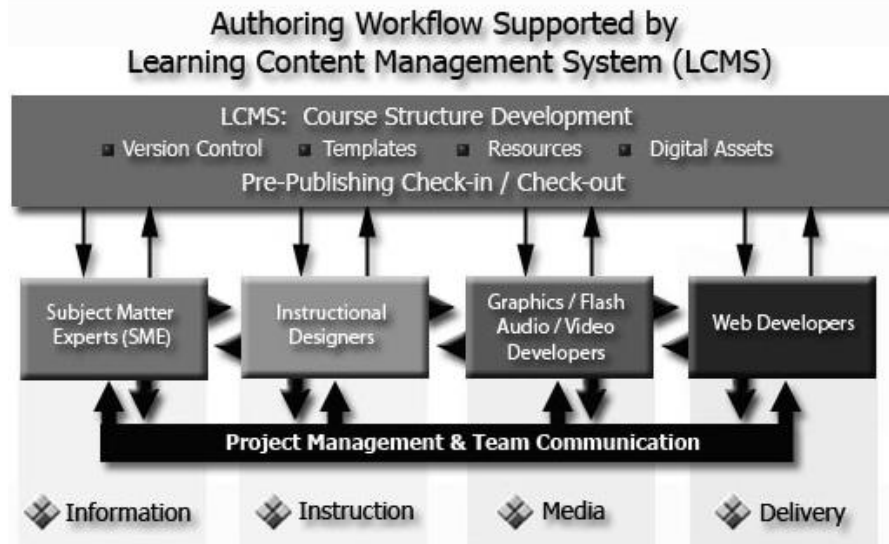


Slika 1:1 LCMS- deploy-LMS

Dakle za LCMS možemo reći da je multi-korisničko okruženje u kojem autori nastavnih sadržaja (eng. learning developers) mogu kreirati, pohraniti, ponovo koristiti, upravljati i dostavljati nastavni materijal iz središnjeg spremišta objekata učenja.

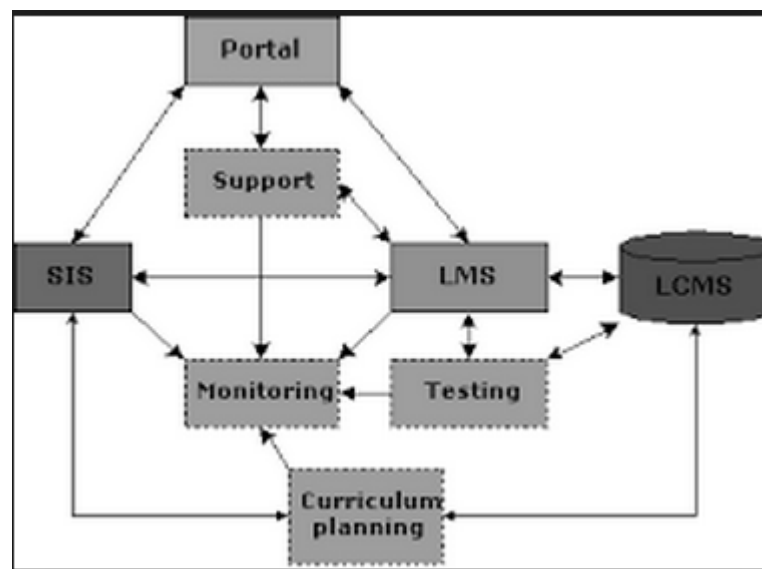
LCMS sustavi dopuštaju korisnicima, administratorima, autorima, ekspertima iz raznih područja (poglavito iz područja nastave) kreirati novi nastavni sadržaj (prezentacija, animacija, kviz..) , ili pohraniti postojeći sadržaj drugih autora. Sustav upravlja organizacijom sadržaja , odnosno kreiranjem, pohranom i dostavom nastavnog sadržaja.

Osnovne komponente LCMS sustava su editori, spremište metatagiranih objekata učenja, dinamičko sučelje dostave sadržaja. Ilustrirana je podrška LCMSa tijekom rada stvaranja, razvoja i upravljanja sadržajem učenja.



Slika 1:2 Tijek rada stvaranja multimedijskog nastavnog materijala

Ilustracija predstavlja logički okvir Internet ili Intranet školskog portala za e-učenje koji sadrži SIS, LMS za dostavu i LCMS za razvoj nastavnog sadržaja



Slika 1:3 Grafički prikaz razmjene podataka između SIS LMS i LCMS sustava

Pored navedenih sustava koji su u praksi često integrirani, spomenut ćemo i e-dnevnik koji je često sastavni dio SIS i LMS sustava, ali može biti i samostalna web aplikacija koja omogućava nastavnicima različitih ustanova da olakšano vode vlastitu evidenciju i postavljaju obavijesti za roditelje i učenike.

2. Primjeri sustava za upravljanje razredom i njihove osnovne karakteristike

Razred u ovom kontekstu predstavlja skupinu učenika ili studenata koji zajedno prate nastavni sadržaj vođen od zajedničkog nastavnika. Poseban aspekt vođenja nastave podrazumijevaju administrativni poslovi, odnosno rad na razrednoj evidenciji i administraciji.

Razmotrit ćemo nekoliko različitih tipova sustava sa njihovim osnovnim značajkama, koji kroz svoja programska rješenja olakšavaju vođenje, administraciju i prezentaciju podataka o nastavi i studentima.

2.1. Moodle

Moodle (Eng. Modular Object-Oriented Dynamic Learning Environment). Naziv govori o svojstvima sustava:

- Modularnost označava odvojene funkcionalnosti sustava koje su međusobno neovisne i prilagodljive. Ponekad se za pojedini modul koristi termin „paket“.
- Objektna orijentiranost predstavlja programski stil baziran na konceptu objekata koji sadrže attribute i metode.
- Dinamičko ažuriranje sadržaja putem sučelja
- Okruženje za učenje (virtualno) koje sadrži niz alata za prezentaciju i evaluaciju znanja, te praćenje različitih aktivnosti sudionika

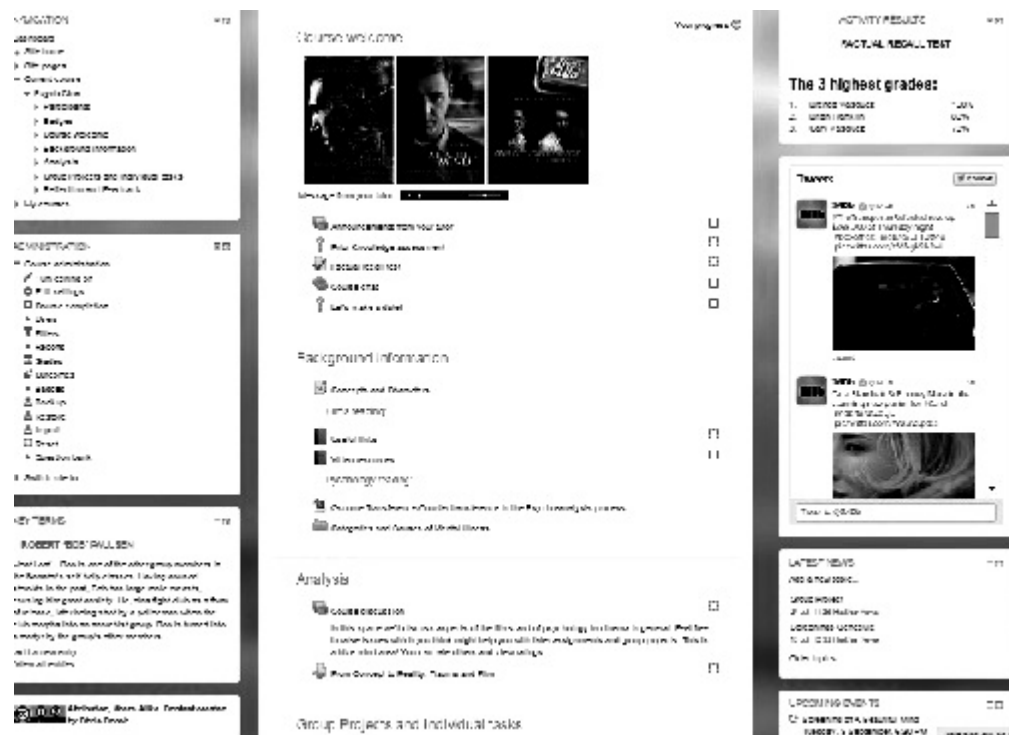
Moodle je projekt otvorenog koda i zaštićen je GNU GPL licencom, što korisnicima omogućava slobodno (besplatno) pokretanje programa, kopiranje, distribuiranje, proučavanje, mijenjanje, poboljšavanje programa uz uvjet da izmijenjeni kod temeljen na GPL licenci zadrži takvu licencu.

Moodle aplikacija je pisana PHP programskim jezikom i podržava više vrsta baza podataka. Može se integrirati na gotovo svaku popularnu platformu i povezati s raznim odvojenim sustavima.

Mnoštvo korisnika sudjeluje u stvaranju novih i mijenjanju postojećih modula, testiranju proizvoda, konzultiranju s ostalim korisnicima.

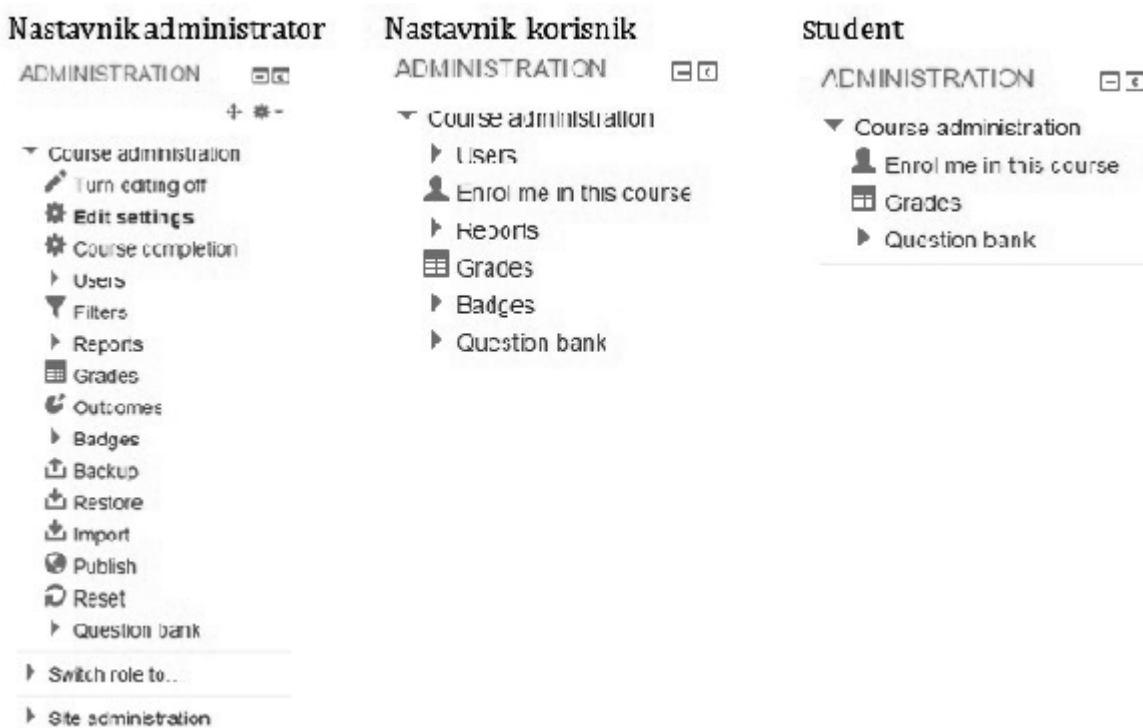
Moodle naslovnica ima koncept web portala, odnosno u središnjem stupcu prikazuje osnovni sadržaj, dok se u bočnim stupcima nalaze poveznice na različite module koji omogućavaju upravljanje sustavom. Izgled sučelja, kao i sadržaj se može promijeniti prema potrebi. Broj administracijskih mogućnosti ovisi o postavkama sustava, broju instaliranih modula i privilegijama korisnika.

Primjer sučelja iz perspektive nastavnika, studenta i gosta:



Slika 2:1 Izgled Moodle sučelja

Primjer razlike u privilegijama korisnika s obzirom na korisničku ulogu:



Slika 2:2 Razlike u privilegijama uloga

Postojeće uloge korisnika je moguće mijenjati, kao i privilegije određenih uloga. Postoji više načina dodavanja novih korisnika sustavu kao što su: samostalna registracija korisnika, pojedinačni unos podataka, uvoz iz Excel tablice, spajanjem na LDAP protokol itd.

Upravljanje kolegijem se može na temelju različitih konceptualnih predložaka. Tako kolegij može biti organiziran po tjednima nastave, poglavljima, po tempu savladavanja gradiva, itd. Moguće je kreirati i razne složene multimedijalne i interaktivne sadržaje za što je potrebno uključiti dodatne alate, koji nisu sastavni dio standardnog paketa. Takvi sadržaji moraju udovoljiti SCORM, Tin Can ili nekim drugim uvaženim standardima za učenje na daljinu. Moguće je kreirati različite izvještaje o aktivnostima sudionika sustava i rezultatima njihovih obrazovnih aktivnosti.

Proširivanje funkcionalnosti sustava

Po principima GNU GPL licence široka zajednica programera razvija razne pomoćne dodatke i module. Većina takvih dodataka je besplatna. Na službenim stranicama Moodle-a možemo pronaći izlist raznih verificiranih dodataka različitih autora, podijeljenih u kategorije kao na slici:

Categories

Activities (283) The main student activities in a course Assignment (25) Database (4) Workshop (3) Quiz (86) SCORM (1) Book (3)	Availability conditions (5) Availability conditions allow teachers to restrict an activity or section so that only certain users can access it.	Blocks (263) Useful tools to add to courses or home pages
Themes (120) Change the look and feel of your site or course	Users (69) Tools for managing users Authentication (32) Enrollment (28) User profile fields (9)	Course formats (25) Change the structure/layout of course pages
Filters (53) Filters can process and change text	Reports (26) Report plugins for administrators, teachers and general users. Admin reports (0) Course reports (0)	Gradebook (13) There are several ways to extend the gradebook Grade reports (6) Reports (4) Imports (0) Grading methods (3)
General plugins (Local) (90) General plugins go into the local directory and do not fit into any particular category.	Editors (46) Alternative text editors for 2.0 or later TinyMCE (14) Atto (28)	Cache (5) Contains plugin types for the Moodle Unified Cache (MUC) Locks (0) Stores (5)
Messaging outputs (4) Redirect messages to other places	Repositories (34) Connect to repositories of files	Portfolios (2) Export user content to other systems
Plagiarism (12) Connect to different plagiarism services	Web service protocols (1) Beyond REST, SOAP, AMF and XMLRPC...	Admin tools (16) Provides utility scripts useful for admins to examine and modify a Moodle site (2.2+) Log storage (1)
Calendars (3) Various types of calendar systems	Other (28) Useful code that does not conform to any standard plugin type Utilities (1) Experimental (4) Incubating (4)	

Slika 2:3 Dodatni moduli Moodle platforme

Nedostatak samog sustava je što nema dovoljno administrativnih modula za upravljanje studentima, stoga je za vođenje dokumentacije potrebno integrirati dodatni sustav koji time proširuje mogućnosti upravljanja, kao što su OpenSIS ili Fedena, iako niti navedeni sustavi ne

zadovoljavaju administrativne potrebe institucija visokog obrazovanja. Također je moguće razviti novi dodatak sustavu koji će upotpuniti specifične potrebe nekog korisnika.

Dodatci

Kao što je spomenuto, Moodle sadrži mnoge dodatke koji se mogu dograditi na jezgru sustava. Pozitivan aspekt takvog pristupa je prilagodljivost sustava, a negativan aspekt se može pokazati prilikom nadogradnje sustava, stoga je bitno da dodatci budu što samostalniji, neovisniji o jezgri sustava i da sa jezgrom komuniciraju putem API sučelja. Tako dizajnirani, testirani dodatak se pridružuje listi javnih Moodle dodataka koji stoje na raspolaganju ostalim korisnicima na prilagodbu i ne ometaju nadogradnju. Opseg osnovnih funkcionalnosti Moodle-a uzrokuje strogo tipiziranje svojstava dodataka koji su klasificirani prema tipovima koji zahtijevaju implementaciju specifičnog API sučelja, iako to u praksi često nije slučaj što dodatno komplicira implementaciju dodataka. Postoje nove težnje za izmjenom arhitekture sustava, na način da se smanji opseg srži, a razne funkcionalnosti prebace na dodatke, što bi ga činilo daleko prilagodljivijim, dok s druge strane trend gomilanja osnovnih specifičnih značajki proširuje jezgru. Kompromis bi se mogao naći u standardiziranju tipova dodataka. Tijek rada (Slika 3:7 Workflow) koji prolazi implementacija dodatka prije dodavanja njegovih funkcionalnosti u jezgru sustava.[3]

2.2. .LRN (dot learn)

.LRN je projekt OpenACS-a (Open Architecture Community System). Internacionaliziran je tako da podržava različite jezike, dijalekte i vremenske zone. Istaknute mogućnosti. LRN-a su dostave domaće zadaće putem drop-boxa, SCORM prikazi, podržanost upravljanjem različitim ulogama u sustavu.

OpenACS je napredan set alata i okvira za izgradnju skalabilne (prilagodljive smanjenju i opterećenju sistema), društveno orijentirane web aplikacije.

Instalacija i nadogradnja zahtijeva korištenje Tcl (Tool Command Language) programskog jezika, Oracle ili Postgres bazu za pohranu podataka, AOLserver za HTTP usluge, te Windows operacijski sustav.



Slika 2:4 Moduli sustava DotLRN

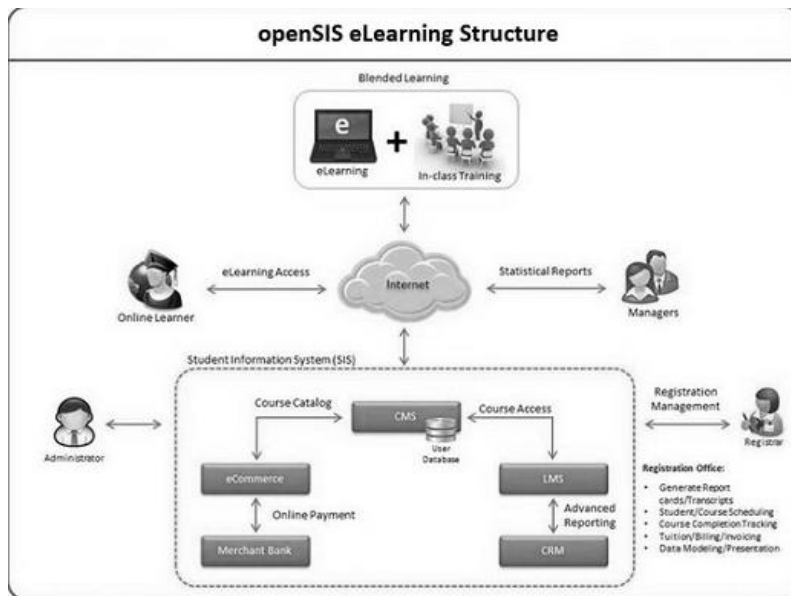
(Pruzeto sa dotlrn.org)

2.3. Opensis

OpenSIS je također besplatni sustav otvorenog koda prvenstveno namijenjen prikupljanju podataka o studentima i kreiranju različitih tabličnih izvještaja za različite korisnike koji mogu biti primjerice :administrator, nastavnik, student, roditelj.

Osnovni paket sadrži podatke o studentima koje je moguće nadopuniti željenim stupcima, raspored nastavnika po dvoranama i terminima, te kolegiju koji održavaju, e-imenik pojedinih kolegija s ugrađenim sustavom za ocjenjivanje, kreiranje izvještaja, evidenciju izostanaka, roditeljski portal koji omogućuje uvid u ocjene, ponašanje, izostanke učenika, jednostavno ponovno pokretanje sustava za kreiranje nove školske godine.

Struktura OpenSIS sustava za e-učenje



Slika 2:5 Struktura OpenSIS sustava

(Pruzeto sa opensis.com)

Course	Period - Teacher	Room	Term	Enrolled	Dropped
Home Room	Home Room - HR - Mary M Jones	100	Full Year	Aug/17/2009	N/A
Music	Period 1 - Q1 - Mu1 - Mary M Jones	Mu101	Quarter 1	Aug/17/2009	N/A
English	Period 2 - Writing - Mary M Teacher	lang 101	Full Year	Aug/17/2009	N/A
English	Period 3 - Reading - Derrick Holland	100	Full Year	Aug/17/2009	N/A
EveryDayMath	Period 4 - Q1 - EDM1 - Jacob A Donoldson	M101	Quarter 1	Aug/17/2009	N/A
EveryDayMath	Period 4 - Q2 - Algebra1 - Jacob A Donoldson	M101	Quarter 2	Oct/12/2009	N/A
EveryDayMath	Period 4 - Q3 - Algebra2 - Jacob A Donoldson	M101	Quarter 3	Jan/11/2010	N/A
EveryDayMath	Period 4 - Q4 - Calculus - Jacob A Donoldson	M101	Quarter 4	Mar/8/2010	N/A

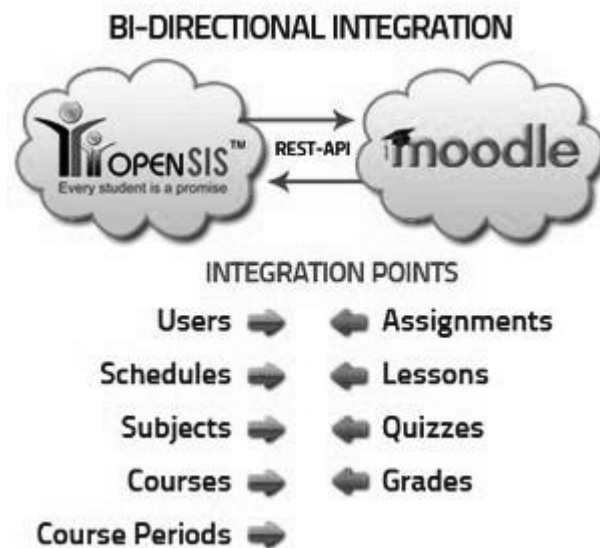
Slika 2:6 Primjer prikaza organizacijske tablice iz sustava OpenSIS

(Pruzeto sa opensis.com)

Sustav podržava integraciju s Moodle LMS-om i Moodlesroom

Navedene dvije aplikacije su integrirane u slijedećim područjima:

- Korisnici kreirani u openSIS-u mogu automatski biti kreirani i u Moodle-u. Postoji stavka za dodavanje korisnika u Moodle.
- Upravljanje kolegijem je u openSIS-u podijeljeno u tri razine :subjekt, Kolegij i Period Kolegija koje se mogu sinkronizirati u istovjetne razine u Moodle-u.
- Raspored studenata se automatski sinkronizira s istovjetnim rasporedom kreiranim u Moodle-u uz propisanu ovjeru. Studenti dobivaju pristup svim svojim kolegijima i njihovim sadržajima. Nastavnici koji upravljaju kolegijima dobivaju izlist svih upisanih studenata u njihov kolegi. Kopiraju se i postavke kalendara.
- Dnevnik ocjena sinkronizira aktivnosti učitelja u moodle-u u formi rasporeda, lekcija, kvizova.Svi stečeni bodovi se automatski povlače u openSIS. Tu učitelj može analizirati napredak studenata, prosječni rezultat koji su studenti ostvarili iz pojedine cjeline, dodijeliti ocjenu studenta, te zaključiti ocjenu.



Slika 2:7 Točke integracije između OpenSIS i Moodle sustava

(Pružeto sa opensis.com)

2.4. Fedena

Fedena je također programska podrška otvorenog koda osnovnih funkcionalnosti, dok se razne komercijalne implementacije plaćaju. Pisana je Ruby programskim jezikom uz korištenje okruženja Ruby on Rails.

Osnovni paket sadži slijedeće dodatke:



Slika 2:8 Osnovni moduli koje nudi Fedena komercijalni sustav

(Pruzeto sa fedena.com)

Fedena ima više realiziranih mogućnosti od OpenSIS-a, ugodnije korisničko sučelje i pojednostavljeno dodavanje modula uz nadoplatu, odnosno namijenjena je korisnicima bez posebnih vještina iz IT područja. OpenSIS je programski jednostavnije realiziran i otvoreniji idejama i nadogradnjama razvojnih programera.

Dodatci :

- DoCircuits je online simulator strujnih krugova. Dizajnirane krugove online sustav analizira.
- Moodle
- BigBlueButton omogućuje video konferenciju. Otvorenog koda je i prvenstveno namijenjen učenju na daljinu.
- Tally upravlja financijskim sustavom organizacije
- Biometric podupire razne biometrijske sklopovske uređaje

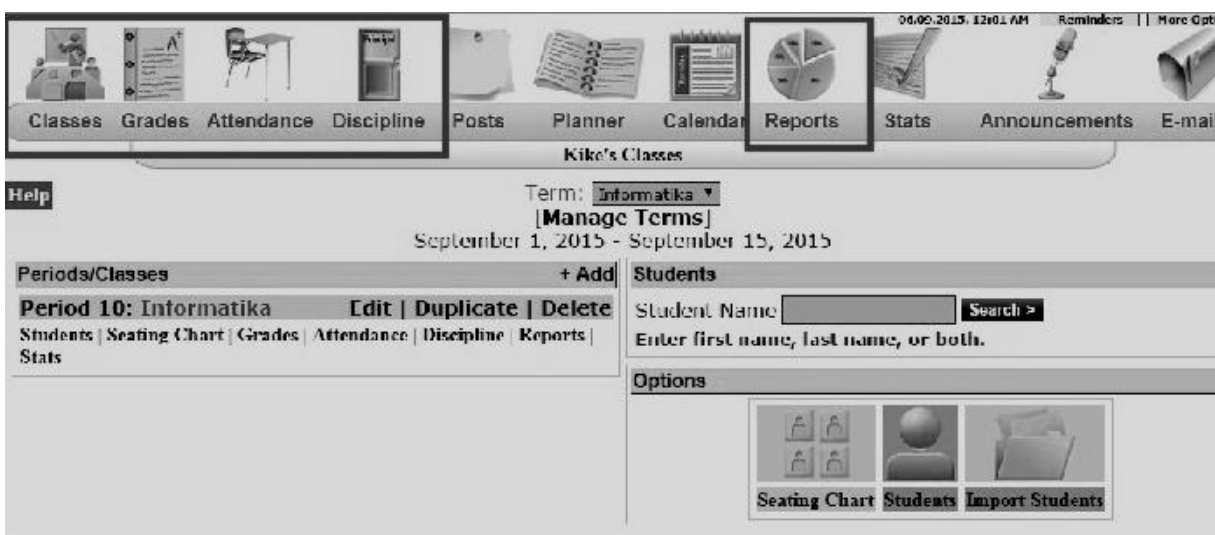
- RFID tehnologija koristi radio frekvenciju kako bi se razmjenjivale informacije između prijenosnih uređaja/memorija i host računala.

2.5. GPA Teacher

Je on-line aplikacija kojoj se pristupa jednostavnim kreiranjem korisničkog računa, te nije potrebna nikakva dodatna instalacija. Sadrži besplatne module i pro module koji se plaćaju na godišnjoj razini.

Besplatni paket sadrži :

- Classes modul definira početak i kraj održavanja kolegija, njegovu podjelu na poglavlja i periode njihovih trajanja. Tako kreiranim kolegijima možemo dodavati studente pojedinačno i iz Excel datoteke, te ih organizirati u grupe.
- Grades modul omogućava unos bodova iz testova znanja pojedinih poglavlja i uz definiranu skalu studentima automatski pridodjeljuje ocjene i računa prosjek.
- Attendance omogućava evidenciju o izostancima.
- Discipline omogućava upisivanje komentara u vidu pohvala i opaski
- Reports kreira izvještaje o pojedinim stavkama u vidu izlista ili grafikona.



Slika 2:9 Sučelje sustava GPA Teacher

Gpa Teacher omogućava nastavnicima pojednostavljeno vođenje evidencije o učenicima, u Pro verziji i postavljanje obavijesti za studente te kreiranje izvještaja za roditelje čak i u slučaju kad njihova matična institucija ne raspolaže ikakvim on-line administrativnim alatima. Nastavnici jednostavno kreiraju korisnički račun i vode vlastitu evidenciju, neovisnu o nekim drugim vanjskim faktorima, dok se svi podaci pohranjuju na serveru aplikacije. Definitivna prednost ovog sustava je jednostavnost upotrebe i zadovoljava potrebe nastavnika koji nemaju potrebu za korištenjem složenijih sustava.

Zajedničke administrativne značajke navedenih sustava su :

- Dodavanje korisnika
- Dodjeljivanje korisničkih uloga
- Kreiranje kolegija
- Vođenje evidencije o izostancima
- Vođenje evidencije o ocjenama
- Kreiranje izvještaja

Zajednički nedostatak u sigurnosnoj ranjivosti sustava proizlazi iz: javne dostupnosti njihovih otvorenih kodova, prilagodljivom razvoju prema potrebama organizacije ovisnom o podršci zajednice čija dobronamjernost nije zajamčena.[4]

Tablični prikaz osnovnih funkcionalnosti navedenih sustava:

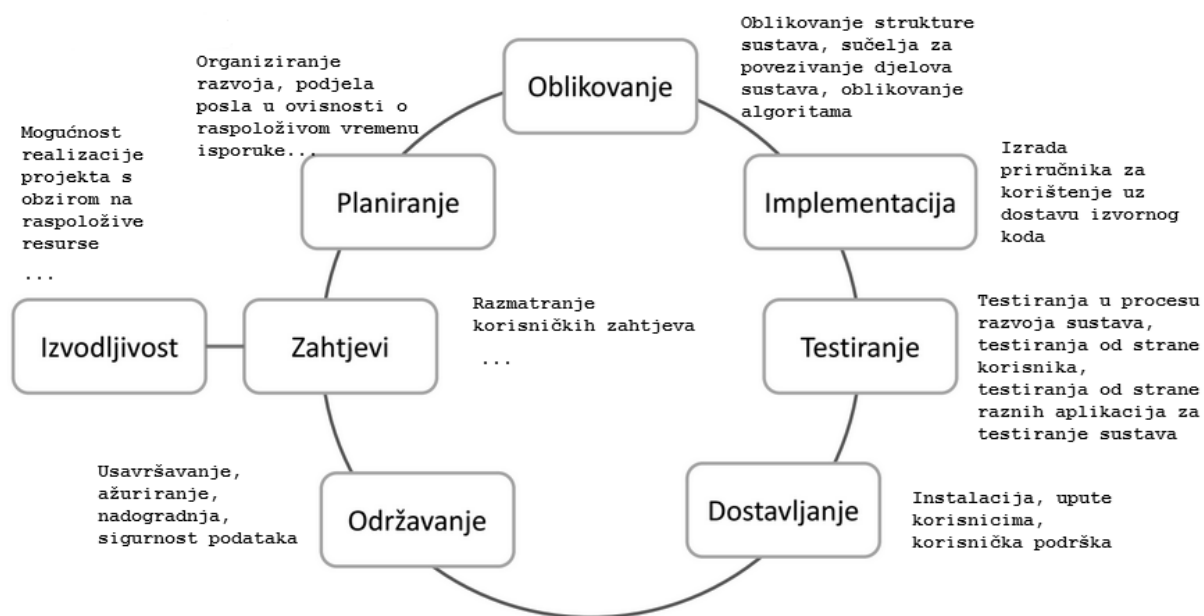
	Tip sustava	Dodavanje korisnika	Korisničke uloge	Evidencija Izostanaka i ocjena	Praćenje	Kreiranje izvještaja	Isporuka nastavnih sadržaja
Moodle	LMS	Korisnici ne mogu samostalno kreirati vlastiti korisnički račun	Ovlasti sudionika su ovisne o dodijeljenoj korisničkoj ulozi	Svi sustavi omogućavaju evidenciju ocjena i izostanaka	Osim praćenja evidencije, prate se sve aktivnosti korisnika sustava. posjećenje stranice, logiranja, aktivnosti nad sadržajem, duljina on-line perioda...	Izvještaj o evidencijama, bodovima, logiranjima korisnika, posjećivanju sadržaja, duljini online perioda..	Po SCORM standardu, sadržaji mogu biti dizajnirani od LMCS sustava, po standardima za osmišljavanje nastavnih sadržaja: ADDIE + Bloomova taksonomija (Error! Reference source not found.)
.LRN							
OpenSIS	SIS				Praćenje podataka iz baze podataka	Izvještaji na temelju SQL upita	
Fedena							
GPATeacher	e-Dnevnik	Kreiranje vlastitog korisničkog računa	Vlastita evidencija i ovlasti pregleda objava za studente i roditelje		Praćenje evidencije i kalendara sadržaja (kurikuluma)	Kreiranje izvještaja na temelju vlastitih zabilješki	Ne isporučuju se nastavni sadržaji

Tabela 2-1 Osnovne funkcionalnosti navedenih sustava

3. Metode razvoja sustava za upravljanje razredom

Metodologija razvoja programske podrške je okvir koji se koristi za strukturiranje, planiranje i kontrolu procesa razvoja informacijskog sustava od strane razvojnog tima prema korisničkim zahtjevima naručitelja.

Životni ciklus programske podrške je slijed aktivnosti koje se javljaju tijekom razvoja programske podrške. (Slika 3:1 Slijed aktivnosti u razvoju programske podrške)

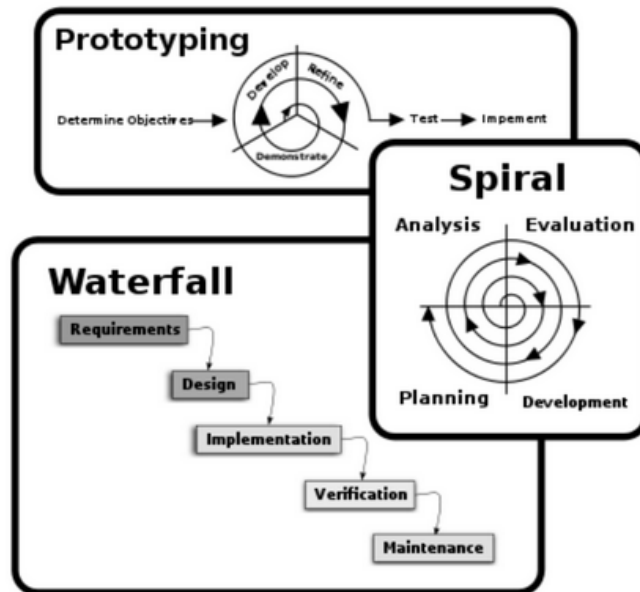


Slika 3:1 Slijed aktivnosti u razvoju programske podrške

Postoje razni modeli životnog ciklusa programske podrške, koji se mogu i kombinirati u skladu s potrebama:

Prototipni model služi za razvijanje prototipa u skladu sa zahtjevima korisnika, Prototip se razvija izvan sustava i služi uspostavi boljes sporazumijevanja između naručitelja i izvođača. Nakon prihvaćenog prototipa slijede uobičajene faze vodopadnog modela.

Wodopadni model predstavlja linearni slijed aktivnosti, odnosno prelazak na novu aktivnost kada je prethodna potpuno završila.



Spiralni model je kombinacija vodopadnog i prototipnog modela. Oblikovanje, implementacija i dostavljanje su linearni dijelovi, dok svaki puni krug spirale predstavlja izraden prototip

Slika 3:2 Osnovni modeli životnog ciklusa razvoja programske podrške

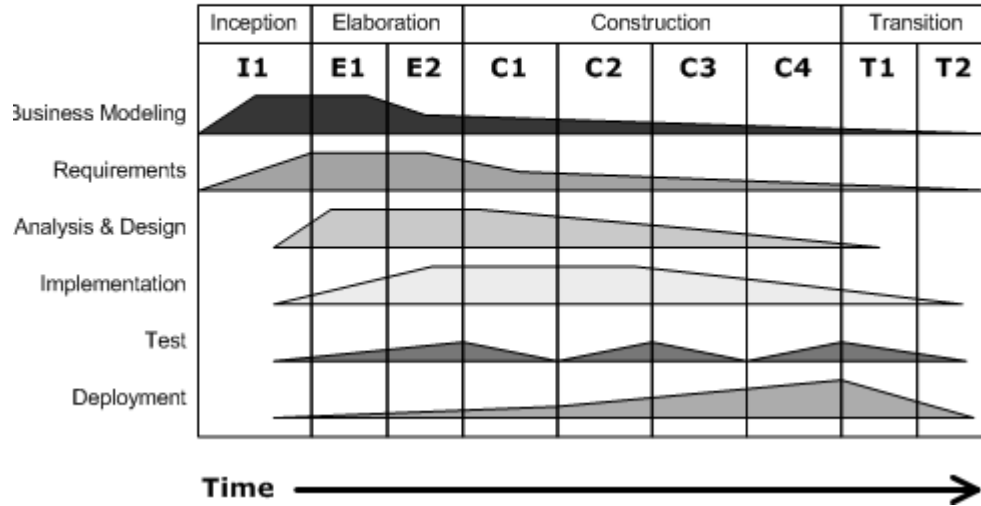
Pored navedenih važno je spomenuti najčešće korišteni iterativni-inkrementalni model kojeg karakterizira:

- Više iteracija životnog ciklusa programske podrške se može odvijati istovremeno
- Razvoj ponavljanjem životnih ciklusa (iteracija) u manjim vremenskim periodima (inkrement)
- Inkrementalni razvoj dijeli funkcionalnosti sustava u inkremente
- Svaki inkrement (odnosno dio funkcionalnosti) se isporučuje nakon iteracije životnog ciklusa [2]

Unified Process je poznati iterativno-inkrementalni model koji se sastoji od faza početka, razrade, izgradnje i prijlaza (inception, elaboration, construction, transition) koje su podijeljene u vremenski uokvirene iteracije. Svaka iteracija rezultira inkrementom koji predstavlja ostvarenje nekih funkcionalnosti sustava.(Slika 3:3 Unified Process)

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



Slika 3:3 Unified Process

Osnovne karakteristike Unified Process modela su :

- Naglasak na arhitekturi sustava, koja mora biti dobro definirana, kako bi se uz objektno orijentirani pristup razvoju omogućila gradnja sustava od već gotovih komponenti (faza Elaboration)
- Naglasak na adresiranju najkritičnijih rizika pri razvoju sustava, prije poduzimanja značajnijih aktivnosti

Rational Unified Process (RUP) je okvir za razvoj sustava po modelu životnog ciklusa Unified Process, kreiran od strane „Rational Software“ korporacije.

Agile Unified Process (AUP) je pojednostavljeni RUP.

U sljedećem poglavlju ćemo objasniti osnove Agile Scrum modela životnog ciklusa kakav se koristi prilikom dodavanja novih funkcionalnosti već razvijenom sustavu (npr. dodavanje novih modula, dodataka, različitih funkcionalnosti prilagodljivim LMS sustavima prilikom izgradnje novih verzija od strane razvojnih timova pojedinih sustava). Realizacija novih funkcionalnosti je uveliko uvjetovana iskustvom i potrebama široke zajednice korisnika.

3.1. Agile Scrum razvojna metoda sustava za upravljanje nastavom

Scrum (engl. ološ) je Agile metoda za razvoj programske podrške nazvana po situaciji u ragbiju kada igrači igraju ispočetka nakon slučajne povrede. Agile modeliranje je osnovano na metodologiji prakse. Scrum metoda je prvi put primijenjena '93 u Easel Korporaciji od Jeff Sutherland, John Scumniotales i Jeff McKenna.[6]

Agile se razlikuje od Rational Unified Process (RUP) procesa s kojim smo se često susretali u načinu upravljanja procesom razvoja. RUP proces ima prediktivnu, a Agile adaptivnu razvojnu strategiju.

Agile zahtijeva sastavljanje tima kao početnu točku, bazira se na stalnoj interakciji s korisnicima i dijeljenju znanja.[5] Scrum metoda je zasnovana na pretpostavci da razvojni proces programske podrške složen i nepredvidljiv i tretira ga kao „crnu kutiju“ (nasuprot potpuno definiranim procesima) i definira razvojne procese kao labave skupove aktivnosti uz kombinaciju poznatih i djelotvornih alata i tehnika. Ova metodologija iz navedenih razloga nosi visok rizik od pogreške, ali je svojevrsno poboljšanje u odnosu na uobičajene iterativno inkrementalne razvojne cikluse.

Ova metodologija upravljanja, unapređenja i održavanja nalazi svoju funkcionalnost u već postojećim sustavima prilikom dodavanja novih funkcionalnosti, modula prototipa.

Navedeni LMS sustavi koriste ovu metodologiju pri razvoju novih modula i „zakrpa“.

Proces razvoja nije linearan kao što je slučaj u vodopadnim, spiralnim i iterativnim metodologijama koje nužno slijede četiri aktivnosti :analizu, dizajn, implementaciju i testiranje i on preskače iz aktivnosti u aktivnost prema potrebi.

Plan puštanja proizvoda programske podrške u promet je baziran na slijedećim varijablama:

- Zahtjevi kupaca(korisnika) – kako sadašnji sustav treba nadograditi .
- Vremenski pritisak – u kojem je vremenskom okviru potrebno steći konkurentsku prednost.
- Natjecateljstvo – na što je konkurentski tim spreman i što je potrebno za pobjedu
- Kvaliteta – kakva je zahtijevana kvaliteta, s obzirom na gore navedene varijable
- Vizija – koje su promjene potrebne u ovoj fazi da bi se ispunila vizija sustava .
- Izvori – kakvo osoblje i kakva sredstva stoje na raspolaganju .

Pošto je razvoj sustava kompleksan, složen proces, od njega je zahtijevana najveća moguća prilagodljivost i odgovarajuća kontrola nad prilagodbama. Evolucija sustava poništava one sustave koji su izolirani od promjenjivog kaotičnog okoliša. Potrebno je timovima omogućiti adaptivan rad unutar kompleksnog okruženja koristeći neprecizne procese.

Karakteristike Scrum metodologije su slijedeće:

- Prva i posljednja faza (planiranje i zatvaranje) sastoje se od definiranih procesa, gdje su svi procesi, ulazi i izlazi dobro definirani. Znanje o tome kako realizirati te procese je eksplicitno. Protok je općenito linearan, s nekim ponavljanja u fazi planiranja.
- Sprint faza je empirijski postupak. Mnogi procesi u fazi sprinta su nepoznati ili nekontrolirani, pa ih se tretira kao crnu kutiju koja zahtijeva vanjsku kontrolu. Zbog toga su kontrole, uključujući i upravljanje rizicima, postavljeni na svakoj iteraciji Sprint faze, kako bi se izbjegao kaos prilikom maksimiziranja fleksibilnosti.
- Sprintovi su nelinearni i fleksibilni. Gdje god je dostupno, koristi se eksplicitno poznavanje procesa ; inače se koristi prešutno znanje i metoda pokušaja i pogreške za izgradnju. Sprint se koristi prilikom razvoja konačnog proizvoda.
- Projekt je otvoren za okruženje do faze zatvaranja. Konačna dostava se može mijenjati u bilo kojem trenutku tijekom planiranja i u Sprint fazama projekta. Projekt ostaje otvoren složenosti okruženja, uključujući i natjecateljstvo, vrijeme, kvalitetu i financijske pritiske tijekom ove faze

Tehnike razvojnih faza uključuju sastavljanje tima, slično kao sportskog tima: svaki član tima treba raditi samostalno, ali usmjerenom prema zajedničkom cilju. Scrum sugerira da ekipa treba imati najviše 6 - 7 članova. Voditelj tima se naziva Scrum master. Njegov posao je provedba i upravljanje Scrum procesima u projektu. Scrum tim kao cjelina definira praksu, sastanake, artefakte i terminologiju momčadi, a Scrum majstor jamči poštivanje tih identificiranih normi. Scrum majstori imaju predstavničku ulogu i njihova vlast je uglavnom neizravna, no većinu svog vremena posvećuju vanjskim upravljanjima, ne ometajući tim i rješavanju vanjskih prepreka, koji se ne može riješiti u Scrum tim. Majstor je također usredotočen na osiguravanje transparentnosti u procesu razvoja

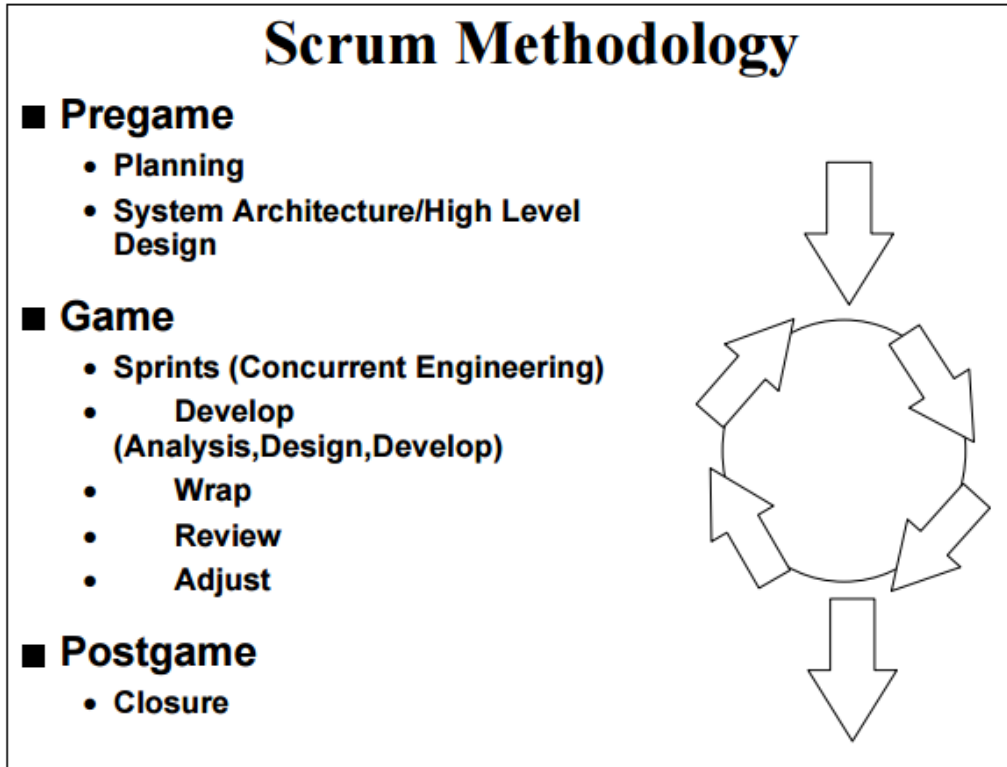
Kreiranje projektnih dokumenata tzv zaostataka (engl. Backlogs)

Postoje 3 vrste zaostataka :

- **Proizvoda** - Djeluje kao spremište ciljanih izmjena i dogradnji namijenjenih puštanju u promet u nekom trenutku . To su obično zahtjevi visokoe razine, s visokom razinom procjena i proračuna od strane interesnih skupina.
- **Otpuštanja** - Zahtjevi izvučeni i identificirani iz zaostataka proizvoda te postavljeni kao prioritet za nadolazeće izdanje. Takvi dokumenti sadrže više pojedinosti o zahtjevima , a manje o proračunima koji su obično procijenjeni od strane samog tima koji obavlja posao.
- **Sprinta** - Na početku svakog sprinta, tim ima sprint planiranja uz krajnji rezultat definiranja zaostalih zahtjeva / pod- zahtjeva i predviđanja njihovih rješavanja i završetak sprinta. Završen Sprint je u potpunosti kodiran, testiran i dokumentiran .

Scrum generalno sadrži tri faze [7]

: Pred igra (eng. Pregame), Igra (eng. Game) i Post igra (eng. Postgame)



Slika 3:4 Faze Scrum razvojnog ciklusa

(Preuzeto sa: http://navegapolis.net/files/Scrum_Development_Process.pdf)

- Pred igra

Definiranje sastava razvojnog tima koji djeluje kao sportski tim u kojem svaki pojedinac radi neovisno u ostvarenju zajedničkog cilja. Svaki tim ima voditelja koji je zadužen za provedbu zajednički dogovorene strategije realizacije Scrum-a, upravljanje i održavanje vanjskog sučelja Scrum tima.

Obrada zahtjeva sljedeće nadogradnje, obrada zaostataka iz prethodnog ciklusa ili fiksirane liste zahtjeva koji trebaju biti ispunjeni u predodređenom vremenskom periodu. Niz takvih vremenskih perioda je sastavnica Scrum razvojnog ciklusa.

Segmentacija projekta u vremenske periode razvoja koje ne traju dulje od četiri tjedna, a nazivaju se „sprints“. Svaki sprint sadrži susrete tima, a započinje planiranjem.

- Igra

Razvoj novih funkcionalnosti nadogradnje, uz poštovanje predodređenog vremenskog ciklusa, zahtjeva, cijene, konkurentnosti, kvalitete. Primjena strategija dogovorenih u Pred igri. Igra je serija sprintova:

- Sprint razvoja - Definiranje promjena potrebnih za provedbu zahtjeva backloga u paketima, otvaranje paketa, analiziranje domene, projektiranje, razvoj, provedba, testiranje i dokumentiranje promjena. Razvoj se sastoji od mikro procesa otkrivanja, izuma i implementacije
- Sprint pakiranja - Stvaranje izvršnih datoteka obuhvaćenih promjenama uz objašnjenja promjena
- Sprint revizije – svi timovi prisustvuju sastanku uz preglede i prezentacije promjena, diskutiraju o nastalim problemima i strategiji rješavanja, definiraju sljedeće backloge
- Sprint prilagodbe – Primjena zaključaka sa prethodnih sastanka

Svaki sprint na kraju ima posebnu reviziju.

Tijekom sprinta , tim provodi svakodnevne Scrum sastanke .

Sastanci se održavaju na istom mjestu, u isto vrijeme, svaki radni dan i ne traju dulje od 30 minuta. Na prvom sastanku je imenovan „Scrum master“ (voditelj tima) koji će na svakom sastanku članovima tima postaviti sljedeća tri pitanja:

Što ste učinili nakon posljednjeg Scrum sastanka ?

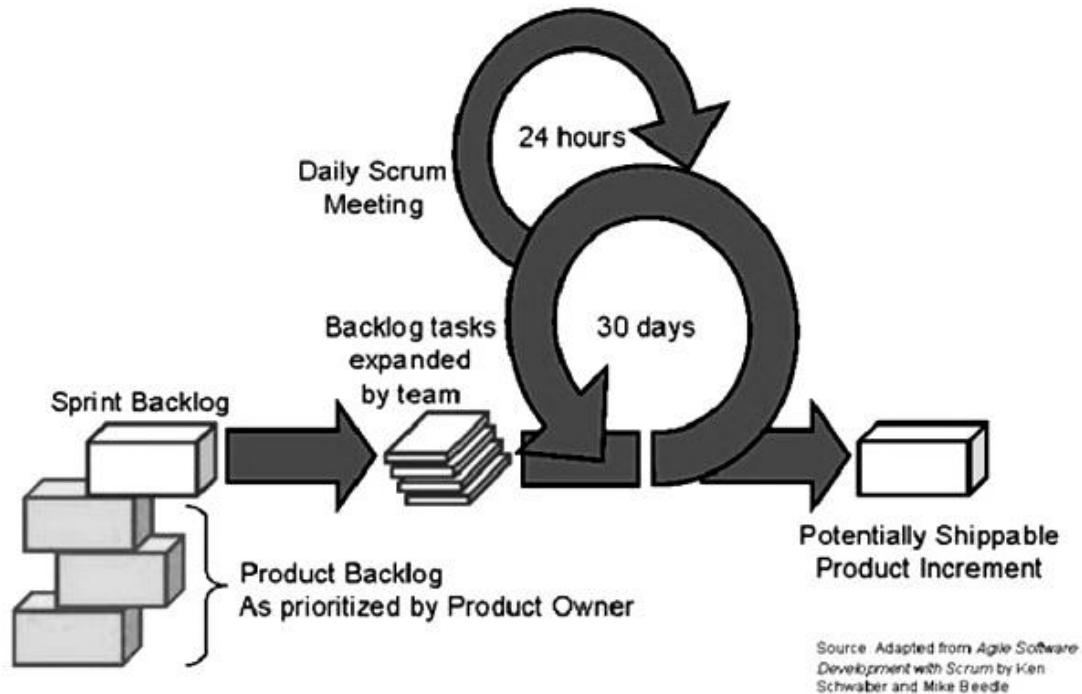
Što je otežalo vaš rad ?

Što planirate raditi do sljedećeg Scrum sastanka?

Razgovor je striktno ograničen na odgovaranje na postavljena pitanja od strane članova tima.

Novi sastanci mogu biti uspostavljeni odmah nakon Scrum sastanka na temelju odgovora na postavljena pitanja .

Voditelj tima je odgovorna za brzo donošenje odluka, ako je to nužno za uklanjanje prepreka prema napretku .

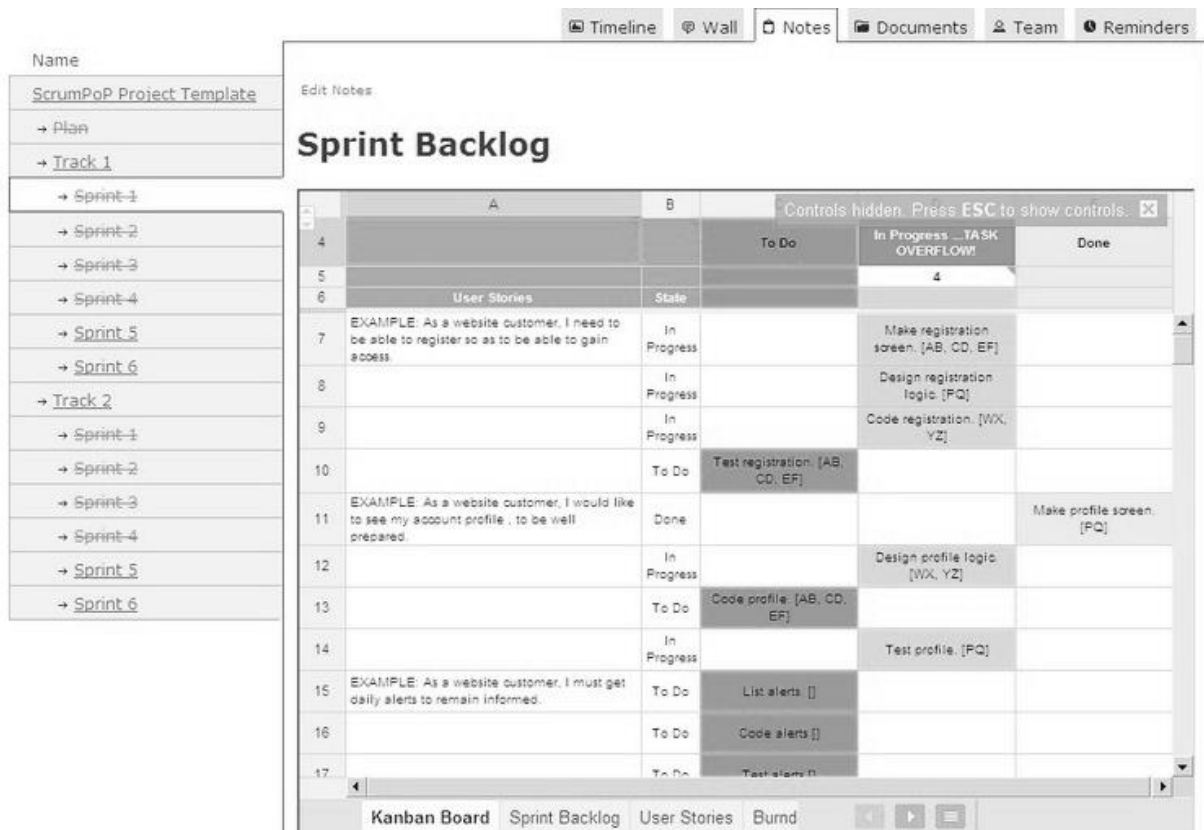


Slika 3:5 Sprint

Osnovni projektni dokument je “Product backlog” koji sadrži listu zahtjeva u prvom stupcu zahtjeva. Zahtjevi se nazivaju “user stories” i zbilja su ponekad predstavljeni kao kratke priče.

U drugom stupcu se nalazi trenutni status pojedinog zahtjeva, odnosno To do, Done ili In progress. U trećem stupcu se nalazi To do lista, odnosno kratki opis akcije kojime se namjerava razriješiti određeni zahtjev. U stupcu In progress je također kratki opis akcije koja se trenutno obavlja npr. kodiranje profila, ili testiranje registracije. U zadnjem stupcu Done su označeni zahtjevi koji su ispunjeni. Ako je potrebno, tom dinamičkom dokumentu koji se u procesu sprinta mijenja možemo dodati i dodatne stupce primjerice s određenim napomenama kao što je npr. prioritet određenog zahtjeva koji se u toku razvojnog procesa može i mijenjati.(Sl.1:3)

- Post igra predstavlja zatvaranje, odnosno pripremu za objavu projekta uz posljednja testiranja i pripremu dokumentacije objave.



Slika 3:6 Primjer Backlog

(Preuzeto sa: <http://www.paradigmpop.com/>)

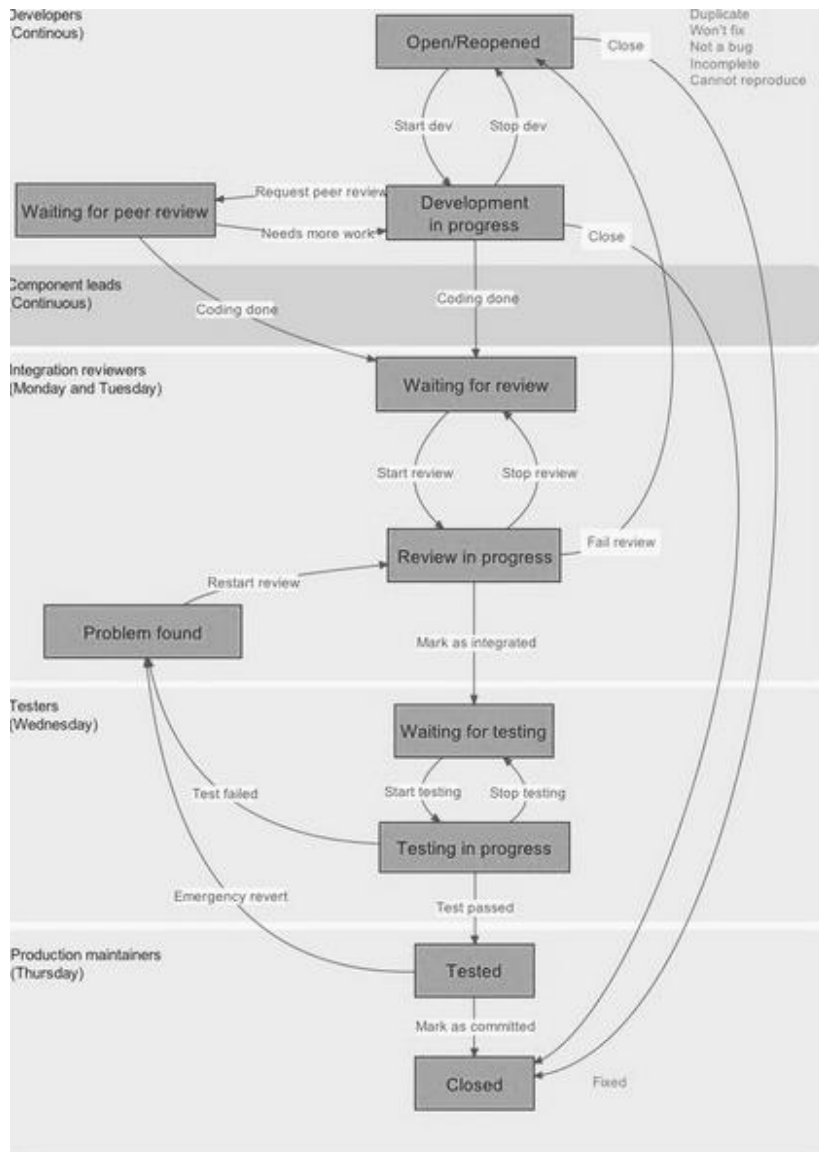
Iz navedenog je jasno da je za ovakav način razvoja programske podrške nužno sastaviti tim eksperata koji dobro poznaju sustav koji nadograđuju, sa svim njegovim značajkama, funkcionalnostima, datotekama i da se međusobno dobro sporazumijevaju. Na dnevnoj razini u sprint fazi imaju kratki sastanak od 30-tak min u kojem revidiraju što je učinjeno prethodnog dana i određuju što je prioritet narednog dana. Ovakva strategija se stoga najčešće primjenjuje u nadogradnjama postojećih sustava, a rezultati su joj vidljivi u novoj verziji aplikacije.

Moodle razvojni tim također primjenjuje ovu strategiju pri ugradnji novih dodataka ili modula sukladnih zahtjevima korisnika. Različite implementacijske ideje, projekti korisnika se obično postavljaju na GitHub.com servis što omogućava širokoj zajednici korisnika vidjeti kod, raspolagati njime, te nuditi različite prijedloge i procjene.

Prije same ugradnje u jezgru sustava takva implementacija prolazi kroz ruke Scrum tima.

Na slici dolje je prikaz tijeka rada u periodu jednog sprinta.(Moodle, Development processes)

[8]



Slika 3:7 Workflow

Preuzeto sa : <https://docs.moodle.org/dev/Process>

4. Model sustava za upravljanje razredom

Zahtjevi korisnika

1. Dodavanje kolegija
2. Za svaki kolegij omogućeno je praćenje evidencije o ocjenama, evidencije o teorijskim i praktičnim tjednim testovima, evidencije o izostancima
3. Dodavanje studenata putem "import" funkcije (xml datoteka)
4. Automatsko stvaranje studijskih grupa iz xml datoteke
5. Pridruživanje studijskih grupa i/ili studenata kolegiju
6. Dodavanje stavki za praćenje kolegiju
7. Definiranje vrijednosti stavki na temelju aritmetičkih i relacijskih operatora primijenjenih na druge stavke za praćenje
8. Prijave kolokvija: "import" studenata koji su prijavili kolokvij iz xml datoteke
9. Prijave ispita: "import" studenata koji su prijavili ispit iz xml datoteke
10. Generiranje izvještaja:
 - popis studenata koji mogu dobiti potpis (npr imaju najviše 4 izostanka)
 - popis studenata koji ne mogu dobiti potpis (npr imaju više od 4 izostanka)
 - popis studenata koji su prijavili kolokvij, a ispunjavaju kriterij za izlazak na kolokvij (npr 6 točno riješenih praktičnih tjednih testova)
 - popis studenata koji su prijavili ispit, a ispunjavaju kriterij za izlazak na ispit (broj izostanaka)
 - popis studenata koji su prijavili ispit, a ispunjavaju kriterij za izlazak na usmeni ispit (npr broj izostanaka i položen praktični dio ispita putem kolokvija, min 120 bodova)

- popis studenata koji su prijavili ispit, a ne ispunjavaju kriterij za izlazak na usmeni ispit (npr broj izostanaka i položen praktični dio ispita putem kolokvija, min 120 bodova)
- raspored za polaganje praktičnog ispita po učionicama generiran slučajnim odabirom (npr 2 učionice po 18 studenata - svakih 45 minuta)
- raspored za polaganje usmenog ispita po učionicama generiran slučajnim odabirom (npr 1 učionica po 4 studenta - svakih 30 minuta)
- izvještaj sa svim (ili samo odabranim) elementima praćenja (kolokviji, ispiti, izostanci)
- Svi popisi se mogu sortirati po odabranom stupcu, pretraživati i filtrirati

Dijagram slučajeva korištenja



Slika 4:1 Prikaz dijagrama slučajeva korištenja

Napomena: Dijagramom zbog opsega nije obuhvaćeno definiranje aritmetičkih, logičkih i relacijskih operatora.

Nakon proučavanja mogućnosti raznih popularnih sustava za upravljanje nastavom možemo utvrditi da niti jedan ne nudi rješenje za definirane zahtjeve korisnika, stoga pristupamo izgradnji vlastitog sustava koji se naknadno može ugraditi u neki od pregledanih sustava kao dodatni modul. Za izvedbu ovog projekta nije korišten niti jedan popularni besplatni okvir zbog relativne jednostavnosti aplikacije s jedne strane i vremena uloženog u učenje korištenja okvira i njegovih ograničenja, te otežanog utvrđivanja događanja u pozadini sustava što otežava otkrivanje pogreške. Ipak ovaj projekt sadrži dodane vanjske slojeve dvaju sustava kako bi tvorilo samostalnu funkcionalnu cjelinu:

- **PHPMYRESERVATION**

Web bazirani sustav prvotno namijenjen rezervaciji perlice rublja po tjednima, danima i terminima. Zaštićen je GNU licencom, odnosno otvorenog je koda. Njegov autor je mladi Norvežanin Ole Jon Bjørkum. Detaljnije o sustavu se može pogledati na sljedećoj stranici <https://code.google.com/p/phpmyreservation>.

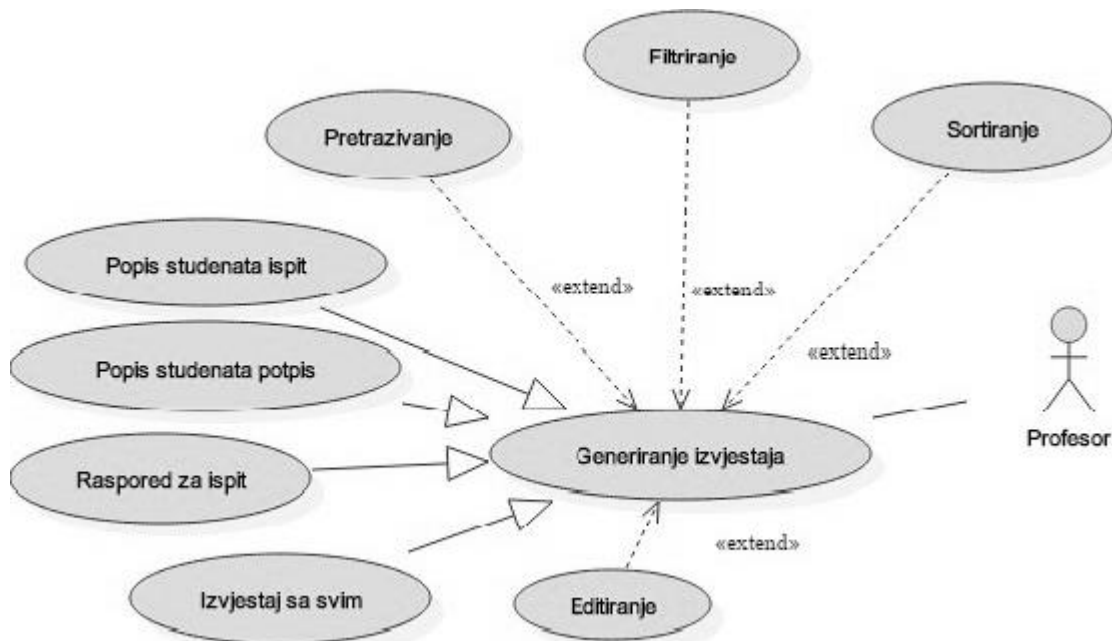
Za potrebe projekta je izmijenjen dio koda javascript i php funkcija. izmijenjene su tablice baze podataka, nepotrebni dijelovi koda su izostavljeni, promijenjen je način rezerviranja tako da možemo iz kalendara odabrati datum rezervacije, a dani u tjednu su izmijenjeni u brojeve dvorana. Iz ovog projekta je preuzet i sustav logiranja/kreiranja korisnika. Ugrađeni kalendar je potpuno otvorenog koda, kako je navedeno na stranici <http://www.triconsole.com>, a njegov autor je kako je vidljivo u dokumentaciji stanoviti „TJ“.

- **JQGRID**

jqGrid je JavaScript kontrola koja preko Ajax tehnika nudi rješenja za prikazivanje i manipuliranje tabličnim podacima na webu. Budući da je grid rješenje za učitavanje podataka na strani klijenta, dinamički kroz Ajax povratne pozive, može se integrirati s bilo kojom tehnologijom na strani poslužitelja, uključujući PHP.

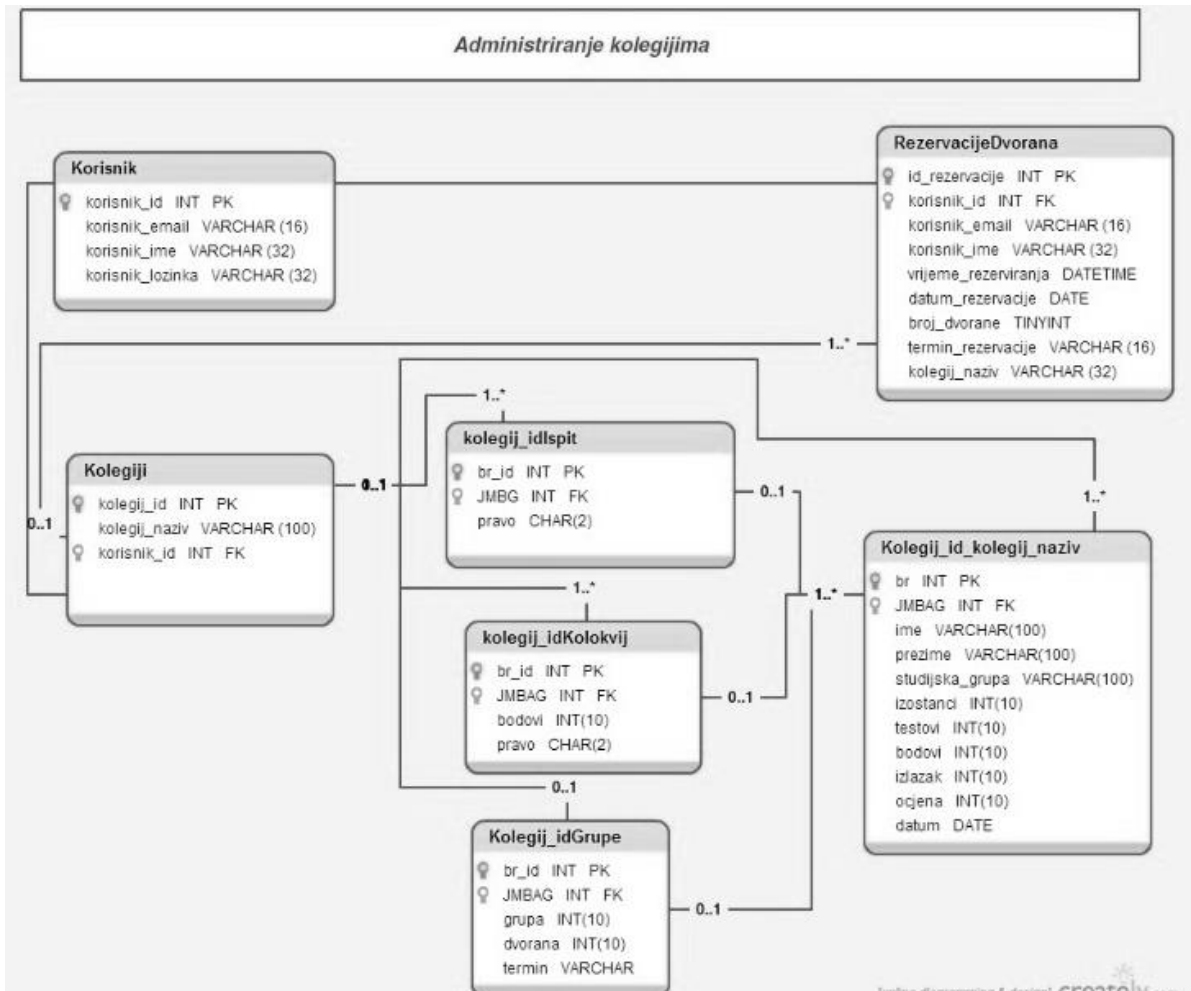
Njegov autor je Bugarin Tony Tomov iz tvrtke Trirand Inc, koji je pronašao rješenje za jednostavnu prezentaciju informacija iz baze podataka. jqGrid omogućava jednostavno editiranje, filtriranje, pretraživanje, sortiranje, grupiranje, dodavanje, brisanje tabličnih podataka. Osnovni paket kontrola je besplatan i otvorenog koda. Razvila se široka mreža korisnika koje za vlastite potrebe proširuju postojeća rješenja. Više podataka se može naći na stranici <http://www.trirand.com/blog>.

jqGrid rješava značajan dio priloženih korisničkih zahtjeva, odnosno slučajeva korištenja, odnosno omogućava pretraživanje, filtriranje, editiranje, sortiranje, prikaz određene liste putem SQL upita.



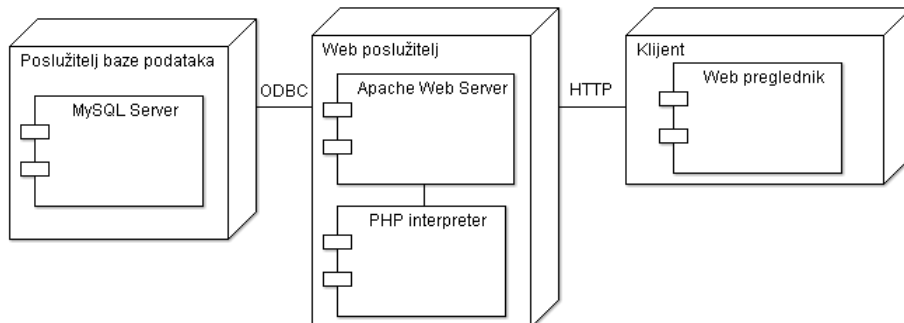
Slika 4:2 Prikaz razriješenih slučajeva korištenja instalacijom jqGrid kontrole

4.1. Model baze podataka



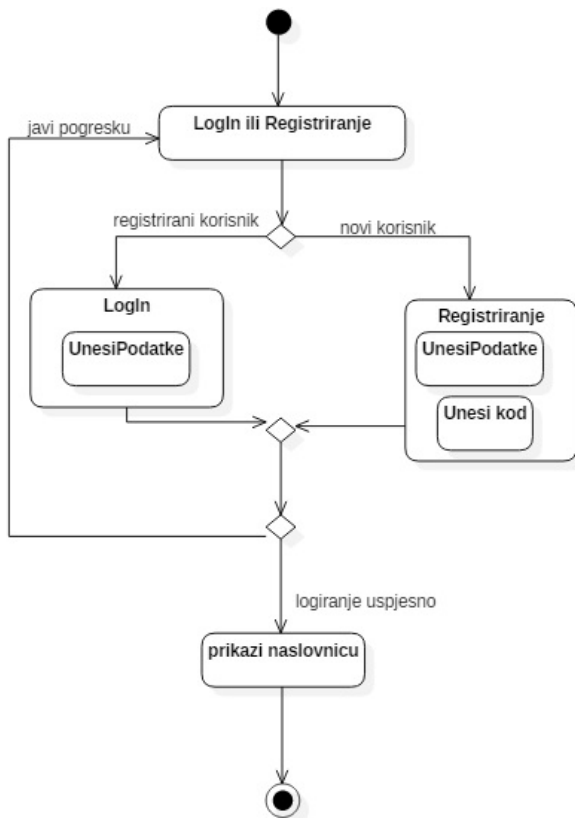
Slika 4:3 Model baze podataka

4.2. Zahtjev za okruženjem potrebnim za realizaciju projekta



Slika 4:4 Ilustracija zahtjeva za okruženjem

Iz sustava phpMyreservation je preuzet sistem logiranja i kreiranja korisničkog računa.



Slika 4:5 Dijagram aktivnosti logiranje/registriranje

The image displays two overlapping web forms. The background form is titled "Log in" and contains fields for "Email:" and "Password:", a "Remember me" checkbox, a "Log in" button, and links for "New user" and "Forgot password". The foreground form is titled "Start > New user" and contains fields for "Name:", "Email:", "Password:", "Confirm password:", and "Secret code:". The "Secret code" field has a "what's this?" link. To the right of the registration fields is an "Information:" section with a bulleted list and a grey box containing text about the secret code. A "Create user" button is located at the bottom of the registration form.

Log in

Email:

Password:

Remember me

[New user](#) | [Forgot password](#)

Start > New user

Name:

Email:

Password:

Confirm password:

Secret code: [what's this?](#)

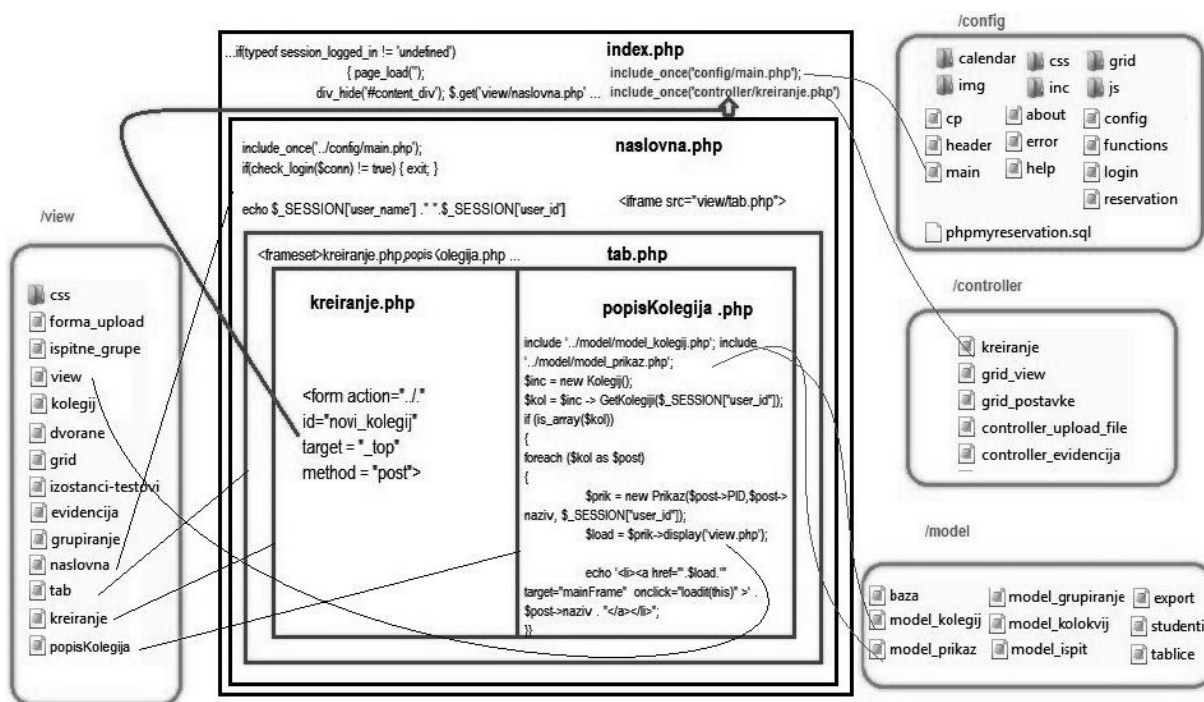
Information:

- With just a click you can make your reservation
- Your usage is stored automatically
- Your password is encrypted and can't be read

Secret code is used to only allow certain people to create a new user. Contact the webmaster by email at your@email.address to get the secret code.

Slika 4:6 Izgled formi logiranje i registriranje

4.3. Prilagodba sučelja phpMyreservation sučelju sustava za administriranje nastavom



Slika 4:7 Ilustracija prilagodbe sučelja

Početa stranica poziva potrebne funkcije i postavke konfiguracije iz okvirnog sustava phpMyreservation.

Dodali smo poveznicu na klasu Controllera za kreiranje kolegija. Postavljena je javascript hash() funkcija za automatsko učitavanje stranice „naslovna“, ukoliko nije zadano nešto drugo. ...

```
function hash()
{
    var hash = window.location.hash.slice(1);
    if(hash == '')
    {
        if(typeof session_logged_in != 'undefined')
        {
            page_load('');
        }
    }
}
```

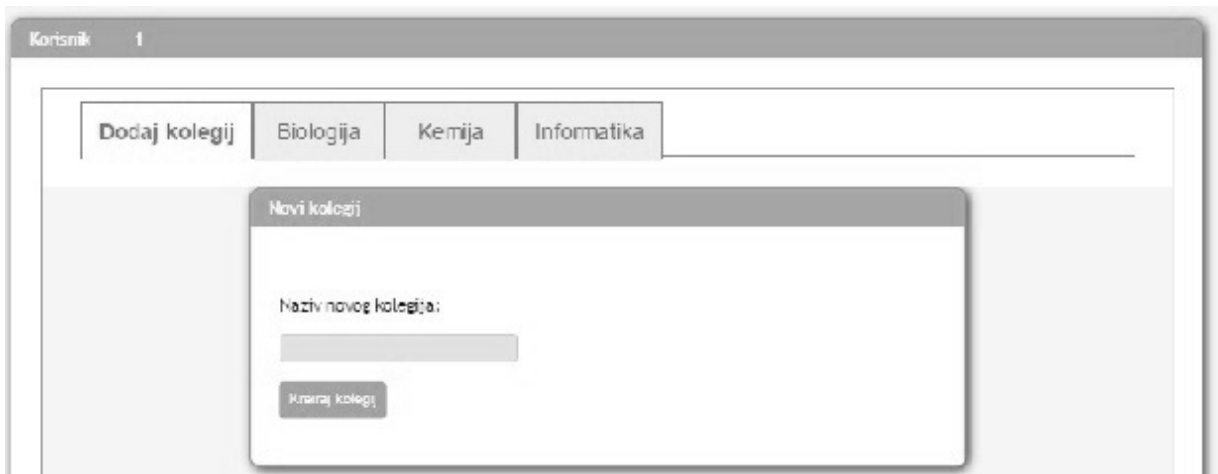
```

        div_hide('#content_div');
        $.get('view/naslovna.php', function(data) {
    $('#content_div').html(data);          div_fadein('#content_div');
page_loaded('about'); }); }...

```

Kôd 5.3.1 – Podešavanje prikaza sučelja u datoteci `cnfig/js/main.js`

Naslovna stranica predstavlja točku prelaska u sferu dodanog sustava za administriranje nastavom. Naslovnica preko okvira poziva stranicu „tab“ kojoj je prosljedila podatke o ID-u i nazivu korisnika, te provjerava status konekcije, odnosno ne otvara novo sučelje ako postavke logiranja nisu unesene (ispravne) (*if(check_login(\$conn) != true) { exit; }*). Stranica „tab“ sadrži okvire za glavnu zadanu karticu povezanu na formu za kreiranje kolegija i kartice koji prikazuje listu kreiranih kolegija logiranog korisnika. Potvrda forme za kreiranje kolegija time vodi na ponovno učitavanje stranice, koja ovog puta prikaže proširenu listu kolegija.



Slika 4:8 Lista kreiranih kolegija

4.4. Kreiranje kolegija

Prilikom kreiranja kolegija u bazi podataka se automatski stvaraju 4 jedinstvene tablice povezane s kreiranim kolegijem. Nakon što je novi kolegij unesen u tablicu koja sadrži Id autora, naziv i Id kolegija, iz iste tablice se nanovo dobavlja uprabo kreirani Id kolegija, koji služi kao prefiks u nazivu tablica pri kreiranju tablica povezanih uz kolegij. Glavna tablica

sadrži podatke o studentu, izostancima, testovima, osvojenim bodovima, pravu na potpis, ocjeni, broju pristupanja ispitu i datumu unosa ocjene. Ostale pomoćne tablice služe za selekciju studenata pri prijavi kolokvija, ispita i prilikom grupiranja studenata u ispitne grupe.

Nakon brisanja kolegija, sve dodatne tablice vezane uz kolegij se također brišu iz baze. Nastavnik ne može kreirati dva kolegija istog naziva, dok različiti nastavnici mogu imati jednake nazive kolegija.

Lista kolegija raspolaže podatkom proslijeđenim od sustava `$_SESSION["user_id"]` na temelju čega iz baze podataka odabere sve kolegije aktivnog korisnika logiranog na sustav, prosljeđujući dalje svakoj kartici kolegija, odnosno sadržaju povezanu na karticu kolegija sljedeće podatke: ID kolegija, naziv kolegija i ID korisnika.

```
...
include          '../model/model_kolegij.php';           include
'../model/model_prikaz.php';
$inc = new Kolegij();
$kol = $inc -> GetKolegiji($_SESSION["user_id"]);
if (is_array($kol))
{
    foreach ($kol as $post)
    {
        $prik      =      new      Prikaz($post->PID,$post->naziv,
$_SESSION["user_id"]);
        $load = $prik->display('view.php');
        echo      '<li><a      href="'. $load.'"      target="mainFrame"
onclick="loadit(this)" >'. $post->naziv. " </a></li>";
    }
} ...
```

Kôd 5.4.1 – Kreiranje popisa kolegija sa pripadajućim kartisama i sadržajem

Otvaranjem kartice pojedinog kolegija pojavljuje se novi niz kartica koje predstavljaju pojedine stavke kolegija, koje bi odgovarale značajkama omogućenih aktivnosti u spomenutim sustavima za upravljanje nastavom. Kartice su primile potrebne podatke nužne za komunikaciju s bazom podataka, kako je već navedeno. Definirane stavke koje bi zadovoljile navedene zahtjeve korisnika obuhvaćaju unos studenata, vođenje evidencije, upravljanje

prijavom i unosom bodova kolokvija, upravljanje prijavom ispita, grupiranjem studenata u ispitne grupe s dodatkom dodavanja događaja u rezervirane dvorane korisnika koje namjenjuje tekućem kolegiju.

Svaka od ovih stavki posredstvom controllera komunicira s klasom modela pojedine stavke, koje nadalje pozivaju funkcije klase modela baze. Gotovo sve spomenute stavke indirektno prosljeđuju različite upite prema bazi podataka, izlisti se prikazuju na sučelju uz pomoć elegantne jqGrid kontrole (osim stavke o evidenciji koju ta kontrola ne podržava).

4.5. Dodavanje stavki kolegiju

Iz gore navedenih razloga zahtjev korisnika za dodavanjem nove stavke nedefiniranih svojstava može se realizirati dodavanjem stavke u izvorni kod koji se odnosi na upravljanje stavkama.

Dovoljno je učiniti sljedeće:

- Otvoriti dokument `Administriranje/view/kolegij.php` i dodati naziv stavke te naziv parametra za novi okvir i odrediti postavke okvira u koji će biti smještena nova stavka.

```
...
<div id="container">
  <ul class="tabNav">
    <li class="current"><a href="#">Unos studenata</a></li>
    <li><a href="#"> Evidencija </a></li>
    <li><a href="#">Prijava kolokvija</a></li>
    <li><a href="#">Prijava ispita</a></li>
    <li><a href="#">Grupiranje</a></li>
    <li><a href="#">Nova stavka</a></li>
  </ul>
  ...

  <div class="tabContainer">
    <div class="tab current">
      <iframe src="<?php echo $studenti; ?>"width="910" height="800"
name = "studenti" scrolling="no"frameborder="0">
      <p>Your browser does not support iframes.</p>
    </iframe>
  </div>
  ...
```

```

    <div class="tab">
    <iframe src="<?php echo $evidencija; ?>"width="900"
height="1500" frameborder="0"scrolling="no"></iframe> </div>
    <div class="tab">
    <iframe src="<?php echo $kolokvij; ?>"width="910"
height="800" frameborder="0"scrolling="no"></iframe></div>
    <div class="tab">
    <iframe src="<?php echo $ispit; ?>"width="920" height="1000"
frameborder="0"scrolling="no"></iframe></div>
    <div class="tab">
    <iframe src="<?php echo $grupe;?>"width="915" height="1700"
frameborder="0"style="float:center"scrolling="no"></iframe>
</div>
<div class="tab">
    <iframe src="<?php echo $nova_stavka; ?>"width="920"
height="1000" frameborder="0"scrolling="no"></iframe></div>
</div> </div>

```

Kôd 5.5.1 – Dodavanje stavke u HTML preložak stavki

- Otvoriti dokument `Administriranje/view/view.php` i dodati vrijednost prethodno odabranom parametru.

...

```

public function set ($id, $naz,$korisnik){
    $prikaz = New Prikaz($id, $naz,$korisnik);
    $studenti                =                $prikaz->
>display('../controller/grid_view.php')."&tablica=kolegij";
    $evidencija                =                $prikaz->
>display('../controller/controller_evidencija.php');
    $kolokvij                =                $prikaz->
>display('../controller/grid_view.php')."&tablica=kolokvij";
    $ispit                    =                $prikaz->
>display('../controller/grid_view.php')."&tablica=ispit";
    $grupe                    =                $prikaz->
>display('../controller/grid_view.php')."&tablica=grupe";
    $ispit = $prikaz->display('http://www.w3schools.com');
include '../view/kolegij.php';

```

```

}
}
$kolegij= New View();
$kolegij->set($_GET['id'],$_GET['ime'],$_GET['korisnik']);

```

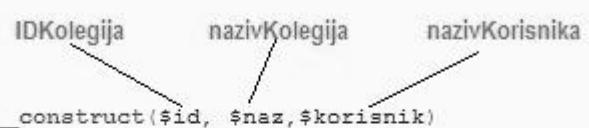
Kôd 5.5.2 – Dodavanje podataka putanji

U gornjem primjeru smo dodali stavku koja se referira na neku vanjsku poveznicu. Nije definirana „tablica“ koja u navedenim slučajevima predstavlja prikaz određene jgGrid tablice. Funkcija „display“ pripada klasi modela „Prikaz“ koja automatski našoj putanji dodaje osnovne podatke o trenutnom kolegiju ako želimo dodati stavku unutar aplikacije koja će koristiti te podatke.

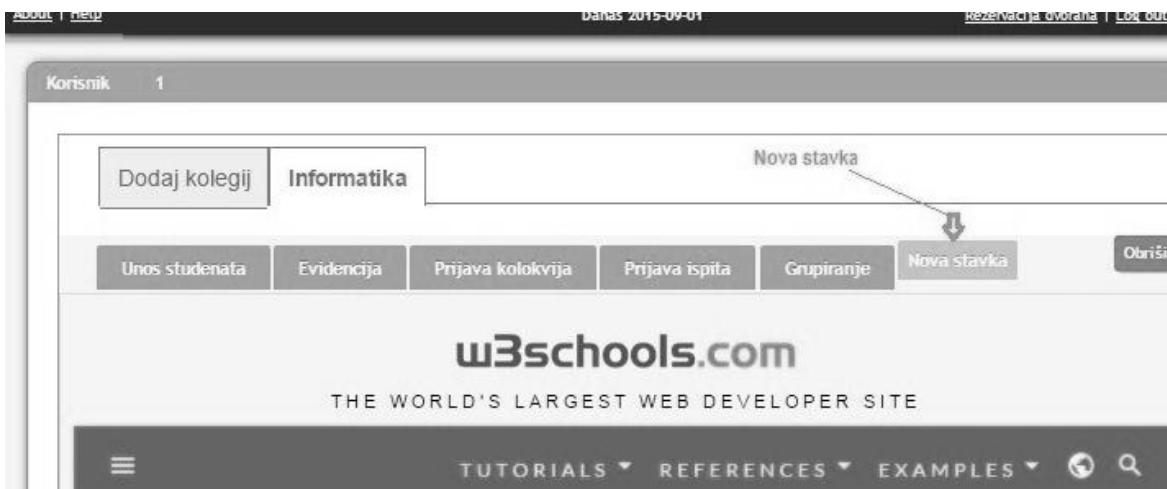
```

class Prikaz
{
public $naziv;
public function __construct($id, $naz, $korisnik)
{
    $this->naziv = "?id=".$id."&ime=".$naz."&korisnik=".$korisnik;
}
public function display($path)
{
    $path .= $this->naziv ;
    return $path;
}
}

```

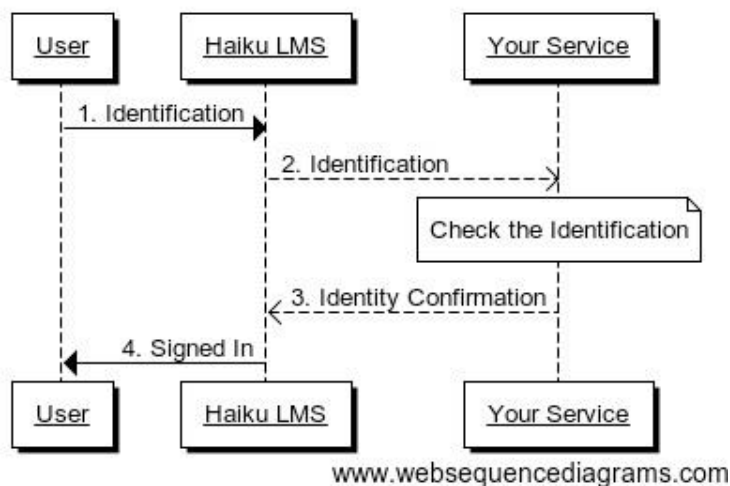


Slika 4:9 Funkcionalnost klase Prikaz



Slika 4:10 Prikaz nove stavke

Stavke kolegija možemo promatrati kao odvojeni dio ovog projekta koji se može ugraditi kao dodatak u navedene LMS sustave koji identificiraju aktivnog, logiranog korisnika i njegovu ulogu, raspoložu popisom kolegija kojima on eventualno upravlja, te prikazuju klijentu sadržaj kolegija i dozvoljene funkcionalnosti ovisno o nezaobilaznom podatcima o ID i nazivu kolegija i ulozi posjetitelja sadržaja. Ponovimo da su Id kolegija, naziv kolegija i Id korisnika temeljni podatci potrebni za pristup kreiranim stavkama kolegija.

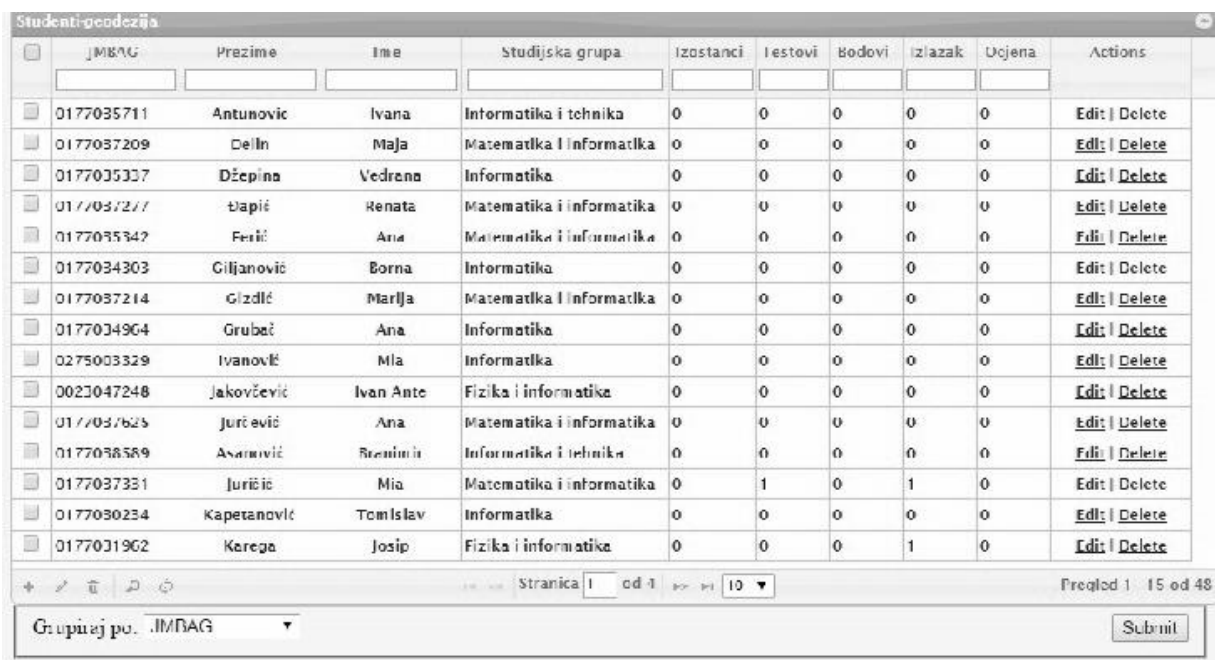


Slika 4:11 Identifikacija korisnika od strane LMS sustava

(Preuzeto sa www.myhaikuclass.com)

4.6. Pridruživanje studenata kolegiju

Kao što je u zahtjevima korisnika definirano studente možemo dodavati učitavanjem Xml datoteke. Na sličan način dodajemo studente određene studijske grupe, uz prethodan odabir studijske grupe iz padajućeg izbornika. Studente je moguće dodavati i pojedinačno, te unesene podatke ispravljati ili brisati. Matični broj studenta je identifikacijski strani ključ svim pripadajućim tablicama kolegija u bazi podataka, što onemogućava višestruko dodavanje istog studenta u isti kolegij, ujedno to je i jedino obavezno ispunjeno polje. Ostale podatke vezane uz polje (kao i samo polje) je moguće editirati. U ovoj listi je moguće mijenjati broj izostanaka i testova za one studente koji su naknadno na neki način odradili dug prema kolegiju. Listu studenata je moguće pretraživati raznim naprednim metodama zahvaljujući funkcionalnosti ugrađene jqGrid kontrole.



<input type="checkbox"/>	JMBAG	Prezime	Ime	Studijska grupa	Izostanci	Testovi	Bodovi	Izlazak	Ocjena	Actions
<input type="checkbox"/>	0177035711	Antunovic	Ivana	Informatika i tehnika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177037209	Delin	Maja	Matematika i informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177035337	Džepina	Vedrena	Informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177037277	Đapić	Renata	Matematika i informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177035342	Ferić	Ana	Matematika i informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177034303	Giljanović	Borna	Informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177037214	Glzdić	Marlja	Matematika i informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177034964	Grubaž	Ana	Informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0275003329	Ivanović	Mia	Informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0023047248	Jakovčević	Ivan Ante	Fizika i informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177037625	Jurčević	Ana	Matematika i informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177038569	Asanović	Brenjina	Informatika i tehnika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177037331	Jurić	Mia	Matematika i informatika	0	1	0	1	0	Edit Delete
<input type="checkbox"/>	0177030234	Kapetanović	Tomislav	Informatika	0	0	0	0	0	Edit Delete
<input type="checkbox"/>	0177031962	Karega	Josip	Fizika i informatika	0	0	0	1	0	Edit Delete

Slika 4:12 primjer tablice generirane pomoću jqGrid kontrole

4.7. Evidencija izostanaka i testova

Evidencija testova i izostanaka sadrži listu studenata uz dva dodatna stupca: „izostanci“ i „testovi“ koji sadrže radio button za čekiranje. Na dnu liste se nalazi dugme za potvrdu unosa. Svaki čekirani unos potvrdom inkrementira pripadajući broj u glavnoj tablici kolegija.

4.8. Prijava kolokvija

Na sličan način na koji se studenti mogu pridružiti kolegiju, pridružuju se i listi studenata koji su prijavili kolokvij. Pojedinačno se studenti mogu dodati isključivo unosom matičnog broja studenta. Sustav provjerava postoji li student sa odgovarajućim matičnim brojem u listi studenata koji su upisali kolegij. Ako takav student (broj) ne postoji, lista će ostati nepromijenjena. Odjava se vrši klikom na retke studenta koji žele odjaviti i koša. Nakon unosa studenata u padajućem izborniku definiramo minimalni broj riješenih testova i maksimalni broj izostanaka koji odobravaju studentu polaganje kolokvija.

Nakon potvrde lista studenata sa pravom pristupa je izmijenjena. Postavljena početna vrijednost „Ne“ je izmijenjena u „Da“ onim studentima koji su stekli pravo pristupa. Ako želimo promijeniti uvjete pristupa, najjednostavnije je isprazniti tablicu (select all, koš za smeće) i unijeti ponovo studente i nove uvjete, iako uvjete možemo mijenjati i za svaki pojedinačni slučaj (klikom na odgovarajuće polje stupca „Pravo“).

U tablicu je moguće direktno unijeti broj bodova stečen na kolokviju. Bodovi uneseni u kolokvij se ne prikazuju u glavnoj tablici. Nakon kolokvija, sadržaj pripadajuće tablice se prazni, a osvojeni bodovi bivaju upisani u glavnu tablicu kolegija. Prazna tablica omogućava organiziranje novog kolokvija, pod eventualno drugačijim uvjetima pristupa, a osvojeni bodovi se pribrajaju bodovima u glavnoj tablici, prilikom novog pražnjenja tablice.

4.9. Prijava ispita

Prijava ispita se odvija na sličan način kao i prijava kolokvija. Pravo pristupa ispitu i pravo na potpis definira broj izostanaka (odabran iz padajućeg izbornika). Odjava ispita se vrši odabirom studenta i ikone koša. U ovoj listi nije moguće unijeti bodove, konačnu ocjenu ni datum unosa ocjene, ona služi za definiranje konačne liste studenata koji su prijavili ispit.

Prilikom definiranja prava pristupa, studentima koji su stekli pravo se inkrementira broj izlazaka u glavnoj tablici, stoga je praktično odjaviti ispit prije definiranja prava (iako se uz male izmjene ta funkcionalnost može prilagoditi želji korisnika). Ostale složene funkcionalnosti vezane u ispit nalaze se u stavci Grupiranje, kojoj će biti prosljeđena lista

studentata koji su prijavili i ostvarili pravo na polaganje ispita, prilikom definiranja ostalih stavki grupiranja.

4.10. Grupiranje studenata u ispitne grupe

Grupiranje u ispitne grupe se vrši na dva načina, ovisno o tome radi li se o pismenom ili usmenom ispitu. Studenti se ne unose iz Xml datoteke, nego se preuzimaju podatci iz tablice „Ispit“. Ukupan broj studenata u tablici grupiranja je umanjen za broj onih koji nemaju pravo potpisa i ako je riječ o pismenom ispitu, za broj onih koji su već stekli dovoljno bodova za pristup usmenom ispitu, a ako je riječ o usmenom ispitu, za broj onih koji nemaju dovoljno bodova za pristup. Ako postavimo min broj bodova (u pismenom ispitu) na broj veći od onog kojeg je moguće ostvariti, u listi će se pojaviti svi studenti sa pravom pristupa, potom možemo pojedinačno izostaviti one koji su zadovoljni osvojenim bodovima koji im omogućavaju pristup usmenom ispitu. Prilikom grupiranja studenata za usmeni ispit, konačnom listom će biti obuhvaćeni svi studenti sa pravom pristupa i dovoljnim brojem bodova iz pismenog ispita (makar prilikom grupiranja za pismeni iz gore navedenih razloga budu izostavljeni). U tablicama grupiranja je moguće unositi konačne bodove, ocjene i datum unosa. Tablica je postavljena na grupiranje po ostvarenim bodovima, ali se proizvoljno može odabrati način grupiranja podataka. Valja napomenuti da stupac grupe odgovara ispitnim, a ne studijskim grupama i da grupiranje u padajućem izborniku odgovara načinu na koji je moguće organizirati prikaz, dok se razvrstavanje studenata u ispitne grupe odvija putem formi.

Grupiranje za pismeni ispit		Grupiranje za usmeni ispit	
Broj studenata po grupi:	<input type="text"/>	Broj studenata po grupi:	<input type="text"/>
<u>Min broj bodova (kolokvij):</u>	<input type="text"/>	<u>Min broj bodova za pristup:</u>	<input type="text"/>
<u>Broj potrebnih dvorana:</u>	<input type="text"/>	<u>Broj dvorane (jedne):</u>	<input type="text"/>
Trajanje ispita u min:	<input type="text"/>	Trajanje ispita u min:	<input type="text"/>
Vrijeme početka ispita (npr 8:45)	<input type="text"/>	Vrijeme početka ispita (npr 8:45)	<input type="text"/>
<input type="button" value="Submit"/>		<input type="button" value="Submit"/>	

Podcrtani unosi imaju različito značenje i ulogu, u ovisnosti o tipu ispita

Slika 4:13 Forma za kreiranje ispitnih grupa

- Broj studenata po grupi odgovara maksimalnom broju studenata po ispitnoj grupi.

- Minimalni broj bodova (osvojen na kolokvijima) koji studenta oslobađa pismenog ispita, odnosno minimalan broj bodova koji omogućava pristup usmenom ispitu.
- Broj potrebnih dvorana odgovara broju dvorana koje stoje na raspolaganju za pismeni ispit, odnosno broj dvorane (jedne) u kojoj će biti održan usmeni ispit
- Trajanje ispita se unosi u minutama (npr. 48, 167, 60...)
- Vrijeme početka ispita prvih grupa po dvoranama : sat (bez nule), dvotočka, minute.

```

public function construct($inJMBAG=null, $inGrupa=null, $inDvorana=null, $inTermin=null){
# Transform minutes like "1:45" into hours like "105".
public function hoursToMinutes($hours)
}{
# Transform minutes like "105" into hours like "1:45".
public function minutesToHours($minutes)
}{
public function grupiranjePismeni($ispit, $naz, $gp){
    $conn= New Db();
    $poc=$this->hoursToMinutes($_POST['vrij']);

    $bod=$_POST['bod'];
    $conn->query("TRUNCATE TABLE ".$gp);
    $rs = $conn->query("SELECT * FROM ".$ispit."
INNER JOIN ".$naz." ON ( ".$ispit.".JMBAG) = ( ".$naz.".JMBAG ) )
WHERE ('Bodovi'< '$bod' and ".$ispit.".Pravo='Da')ORDER BY RAND()") or die ('Cannot process SQL count totals query');

    if($rs->num_rows > 0)
    {
        $num_rows = $rs->num_rows;
        $group = ceil($num_rows/$_POST['bs']);
        $b=0;
        while($row = $rs->fetch_assoc())
        {
            $j = $row['JMBAG'];
            $ab=fmod($b,$group) + 1 ;
            $cd = fmod($ab,$_POST['dvor'])+1;
            $pocet = $poc + $_POST['min'] * (ceil($ab/$_POST['dvor']) -1);
            $kraj=$this->minutesToHours($pocet + $_POST['min']);
            $termin=$this->minutesToHours($pocet) . " - " . $kraj;
            // echo $row['JMBAG'] . ". grupa - " . $ab . ". dvorana - " . $cd . ". termin - " . $this->minutesToHours($pocet) . " - " . $kraj . "<br>";
            $b=$b+1;
            $conn->query("INSERT INTO ".$gp." (JMBAG, Grupa, Dvorana, Termin) VALUES ('$j', '$ab', '$cd', '$termin')");
        }
    }
}

```

Slika 4:14 Grupiranje za pismeni ispit

Klasa modela IspitneGrupe sadrži dvije pomoćne funkcije :

hoursToMinutes(\$hours) koja uneseno vrijeme početka ispita određene grupe pretvara u minute za potrebe funkcije grupiranja

minutesToHours(\$minutes) koja u svrhu izlista vremenskog rasporeda grupa pretvara unesene minute u sate.

Funkcija grupiranja za pismeni ispit:

```

public function grupiranjePismeni($ispit, $naz, $gp){
    $conn= New Db();

```

```

    $poc=$this->hoursToMinutes($_POST['vrij']);
    $bod=$_POST['bod'];
    $conn->query("TRUNCATE TABLE ".$gp);
    $rs = $conn->query("SELECT * FROM ".$ispit."
    INNER JOIN ".$naz." ON ( (`".$ispit."`.JMBAG) = (
`".$naz."`.JMBAG ) ) WHERE (`Bodovi` < '$bod' and
`".$ispit."`.Pravo ='Da')ORDER BY RAND()") or die ('Cannot
process SQL count totals query');
    if($rs->num_rows > 0)
    {
        $num_rows = $rs->num_rows;
        $group = ceil($num_rows/$_POST['bs']);
        $b=0;
        while($row = $rs->fetch_assoc())
        {
            $j = $row['JMBAG'];
            $ab=fmod($b,$group) + 1 ;
            $cd = fmod($ab,$_POST['dvor'])+1;
            $pocet = $poc + $_POST['min'] *
(ceil($ab/$_POST['dvor']) -1);
            $kraj=$this->minutesToHours($pocet +
$_POST['min']);
            $termin=$this->minutesToHours($pocet)." -
".$kraj;
            // echo $row['JMBAG']."..grupa-
".$ab.."..dvorana-".$cd.."termin-".$this-
>minutesToHours($pocet). "---".$kraj."<br>";
            $b=$b+1;
            $conn->query("INSERT INTO ".$gp." (JMBAG,
Grupa, Dvorana, Termin) VALUES ('$j','$ab','$cd','$termin)");

```

Kôd 5.10.1 – Funkcija grupiranja za pismeni ispit

Tumačenje koda:

Funkcija prima parametre: `$ispit` (odgovara nazivu tablice ispit u bazi podataka), `$naz` (odgovara nazivu glavne tablice kolegija u bazi podataka sa podacima o studentima), `$gp` (odgovara nazivu pomoćne tablice grupe vezane uz kolegij u bazi podataka). Vrijednost

varijable \$poc odgovara unesenoj vrijednosti vremena početka ispita, pretvorenoj pomoću pomoćne funkcije u minute. Vrijednost \$bod odgovara unesenoj vrijednosti minimalnog boda koji studenta oslobađa od pismenog ispita.

Prije početka grupiranja, tablica sa sadržajem prethodnog grupiranja se prazni:

```
$conn->query("TRUNCATE TABLE ".$gp). Funkcija HoursToMinutes() pretvara uneseno vrijeme početka ispita u minute na način da broj ispred dvotočke pomnoži sa 60 i pridoda mu broj iza dvotočke. SQL naredbom se iz tablice s popisom svih studenata pridruženih kolegiju odaberu oni studenti koji su prisutni i u tablici Ispit, među kojima se zatim nasumično izdvoje oni koji imaju pravo pristupa ispitu i koji imaju nedovoljan broj već osvojenih bodova stečenih na kolokviju ili prethodnom pismenom ispitu. (Prilikom selektiranja studenata za usmeni ispit biraju se oni koji imaju dovoljan broj stečenih bodova za pristup usmenom ispitu(vidi Kôd 5.10.2)).
```

```
$rs = $conn->query("SELECT * FROM ".$ispit."  
INNER JOIN ".$naz." ON ( `".$ispit.`.JMBAG) = ( `".$naz.`.JMBAG ) )  
WHERE (`Bodovi` < '$bod' and `".$ispit.`.Pravo ='Da')ORDER BY RAND()") or  
die ('Cannot process SQL count totals query');
```

Broj prebrojenih studenata koji su zadovoljili spomenute uvijete (\$num_rows = \$rs->num_rows;) se dijeli za zadanim brojem studenata po grupi. Uzimamo zaokruženu veću vrijednost, odnosno prirodni broj koji je veći ili jednak dobivenom rezultatu (\$group = ceil(\$num_rows/\$_POST['bs']);). Time smo dobije broj grupa koje je potrebno razvrstati po terminima i dvoranama. (Kada studente grupiramo za usmeni ispit oni se raspoređuju u istu dvoranu, a zadani broj predstavlja broj dvorane). Svakom studentu while(\$row = \$rs->fetch_assoc())... (iz liste kreirane nasumičnim rasporedom „ORDER BY RAND()”) dodjeljujemo vrijednost brojača \$b. Vrijednost \$j odgovara matičnom broju studenta iz liste . Vrijednost \$ab predstavlja broj grupe u koju je student razvrstan i odgovara vrijednosti brojača po modulu ukupnog broja grupa, uvećanoj za 1, kako bismo izbjegli vrijednost 0 (\$ab=fmod(\$b,\$group) + 1;). Vrijednost \$cd odgovara broju dvorane koja je dodijeljena određenoj grupi, odnosno broj tekuće grupe po modulu ukupnog broja dvorana uvećan za 1 : \$cd = fmod(\$ab,\$_POST['dvor'])+1; .

Određivanje početka i kraja ispitnog termina određene grupe se izvodi na slijedeći način:

```
$pocet = $poc + $_POST['min'] * (ceil($ab/$_POST['dvor']) -1);
```

Početak je suma unesenog početka (pretvorenog u minute) i vremena trajanja pomnoženog sa zaokruženom vrijednošću najbližeg većeg ili jednakog količniku broja tekuće grupe i ukupnog broja dvorana, umanjenog za 1. Na taj način će sve grupe čiji je broj manji ili jednak ukupnom broj dvorana imati vrijeme početka jednako unesenom vremenu početka jer će za njih vrijednost „ $\text{ceil}(\$ab/\$_POST['dvor']) - 1$ ” biti jednaka nuli. Dok će primjerice vrijednost početka ispita sedme grupe u ukupno tri dvorane odgovarati unesenom vremenu početka uvećanom za vrijeme trajanja pomnoženim s dva. Vrijeme kraja je dobiveni početak uvećan za trzanje.

Prvi sudent iz prve ispitne grupe ima pridruženi zadani početak ispita npr. 9 : 00, kraj ispita je uvećan za minute trajanja ispita. Vrijednost vremena kraja ispita određene grupe se dobije pomoću funkcije minutesToHours() koja vraća vrijeme kraja (sati su cjelobrojna vrijednost dobivena dijeljenjem sa 60 i minute koje su zapravo ostatak pri dijeljenju, odnosno minute po modulu 60). Prilikom grupiranja za usmeni ispit sljedeći student će pripadati sljedećoj grupi, a vrijeme početka njegovog ispita odgovara vremenu završetku ispita prethodnog studenta jer se svi ispiti održavaju u jednoj istoj dvorani, kojoj je unesen broj koji služi jedino u svrhu obavijesti. Svaki student će na usmenom ispitu imati ispitnu grupu koja je višekratnik ukupnog broja dvorana $\text{ceil}(\$ab/ 1) - 1 = \$ab - 1$. Broj grupe umanjen za jedan pomnožen sa minutama trajanja ispita označava minute početka, a početak uvećan za trajanje označava kraj njegovog ispita

```
$pocet =$poc + $_POST['min'] * ($ab -1); (vidi Kôd 5.10.2)
```

Funkcija grupiranja za usmeni ispit:

```
public function grupiranjeUsmeni($ispit, $naz, $gp){
    $conn= New Db();
    $poc=$this->hoursToMinutes($_POST['vrij']);
    $bod=$_POST['bod'];
    $conn->query("TRUNCATE TABLE ".$gp);
    $rs = $conn->query("SELECT * FROM ".$ispit." INNER JOIN
    ".$naz." ON ( `".$ispit."`.JMBAG) = ( `".$naz."`.JMBAG ) )
    WHERE ( `Bodovi`>= '$bod' and `".$ispit."`.Pravo ='Da')ORDER BY
    RAND()") or die ('Cannot process SQL count totals query');
    if($rs->num_rows > 0)
```

```

{   $num_rows = $rs->num_rows;
    $group = ceil($num_rows/$_POST['bs']);
    $b=0;
    while($row = $rs->fetch_assoc())
    {
        $j = $row['JMBAG'];
        $ab=fmod($b,$group) + 1 ;
        $cd = $_POST['dvor'];
        $pocet = $poc + $_POST['min'] * ($ab -1);
        $kraj=$this->minutesToHours($pocet +
$_POST['min']);
        $termin=$this->minutesToHours($pocet)." -
".$kraj;
        //echo $row['JMBAG'].."..grupa-
".$ab.."..dvorana-".$cd.."termin-".$this-
>minutesToHours($pocet)."---".$kraj."<br>";
        $b=$b+1;
        $conn->query("INSERT INTO ".$gp." (JMBAG,
Grupa, Dvorana, Termin) VALUES ('$j','$ab','$cd','$termin)");
    }
} }

```

Kôd 5.10.2 – Funkcija grupiranja za usmeni ispit

Prikaz tablice baze podataka razvrstanih studenata po grupama i terminima za usmeni ispit

	br	JMBAG	Grupa	Dvorana	Termin
<input type="checkbox"/> Edit Copy Delete	1	0177037209	1	7	9:00 - 9:30
<input type="checkbox"/> Edit Copy Delete	2	0177034303	2	7	9:30 - 10:00
<input type="checkbox"/> Edit Copy Delete	3	0177037401	3	7	10:00 - 10:30
<input type="checkbox"/> Edit Copy Delete	4	0177031962	1	7	9:00 - 9:30
<input type="checkbox"/> Edit Copy Delete	5	0177035342	2	7	9:30 - 10:00
<input type="checkbox"/> Edit Copy Delete	6	0177037331	3	7	10:00 - 10:30
<input type="checkbox"/> Edit Copy Delete	7	0177038087	1	7	9:00 - 9:30
<input type="checkbox"/> Edit Copy Delete	8	0177030234	2	7	9:30 - 10:00

Slika 4:15 Tablica grupiranja za usmeni ispit u bazi podataka

Prikaz tablice liste razvrstanih studenata po grupama, terminima i dvoranama za pismeni ispit

Unos studenata Evidencija Prijava kolokvija Prijava ispita **Grupiranje** **Obriši kolegij**

Export xls

Informatika-grupe-ispit

<input type="checkbox"/>	JMBAG	Prezime	Ime	Grupa	Dvorana	Termin	Bodovi	Izlazak	Ocjena	Datum
1										
<input type="checkbox"/>	0177032629	Roguljić	Luka	1	2	8:00 - 8:56	10	1	0	0000-00-00
<input type="checkbox"/>	0177028010	Kuvačić	Ivan	1	2	8:00 - 8:56	34	1	5	2015-09-20
<input type="checkbox"/>	0177034387	Radan	Ivana	1	2	8:00 - 8:56	20	2	0	0000-00-00
<input type="checkbox"/>	0177037261	Krvavica	Marta	1	2	8:00 - 8:56	0	3	0	0000-00-00
<input type="checkbox"/>	0177037165	Musap	Loredana	1	2	8:00 - 8:56	0	1	0	0000-00-00
										5
2										
<input type="checkbox"/>	0177037459	Čagalj	Antonia	2	1	8:00 - 8:56	0	1	0	0000-00-00
<input type="checkbox"/>	0177038019	Bojanić	Ivan	2	1	8:00 - 8:56	0	3	0	0000-00-00
<input type="checkbox"/>	0177035384	Kelava	Nikolina	2	1	8:00 - 8:56	27	1	0	0000-00-00
<input type="checkbox"/>	0177030047	Šarčević	Mario	2	1	8:00 - 8:56	0	2	0	0000-00-00
<input type="checkbox"/>	0275003329	Ivanović	Mia	2	1	8:00 - 8:56	0	2	0	0000-00-00
										5
3										
<input type="checkbox"/>	0177036098	Šarić	Ivan	3	2	8:56 - 9:52	0	2	0	0000-00-00
<input type="checkbox"/>	0177034964	Grubač	Ana	3	2	8:56 - 9:52	0	1	0	0000-00-00
<input type="checkbox"/>	0177037347	Trtinjak	Matija	3	2	8:56 - 9:52	0	4	0	0000-00-00
<input type="checkbox"/>	0177037401	Lučić Lavčević	Nelija	3	2	8:56 - 9:52	47	1	0	0000-00-00
<input type="checkbox"/>	0177037277	Đapić	Renata	3	2	8:56 - 9:52	0	1	0	0000-00-00
										5
4										
<input type="checkbox"/>	0177034917	Bogovac	Marija	4	1	8:56 - 9:52	0	1	0	0000-00-00
<input type="checkbox"/>	0023057909	Đevra	Tomislav	4	1	8:56 - 9:52	0	1	0	0000-00-00
<input type="checkbox"/>	0177037186	Selak	Marija	4	1	8:56 - 9:52	0	4	0	0000-00-00
<input type="checkbox"/>	0177037103	Plavčić	Stjepan	4	1	8:56 - 9:52	0	4	0	0000-00-00

Stranica 1 od 1 All Pregled 1 - 31 od 31

Grupiraj po: Grupa odaberite ukoliko NE želite sačuvati bodove za slijedeći rok

Slika 4:16 Prikaz grupiranja za pismeni ispit

U prikazanoj tablici je moguće unositi bodove stečene na ispitu, unijeti ocjenu i datum unosa ocjene. U sličnoj tablici za usmeni ispit, bit će prikazani i oni studenti koji nisu prisustvovali pismenom ispitu, ali imaju dovoljan broj bodova za pristup ostvarenih putem kolokvija.

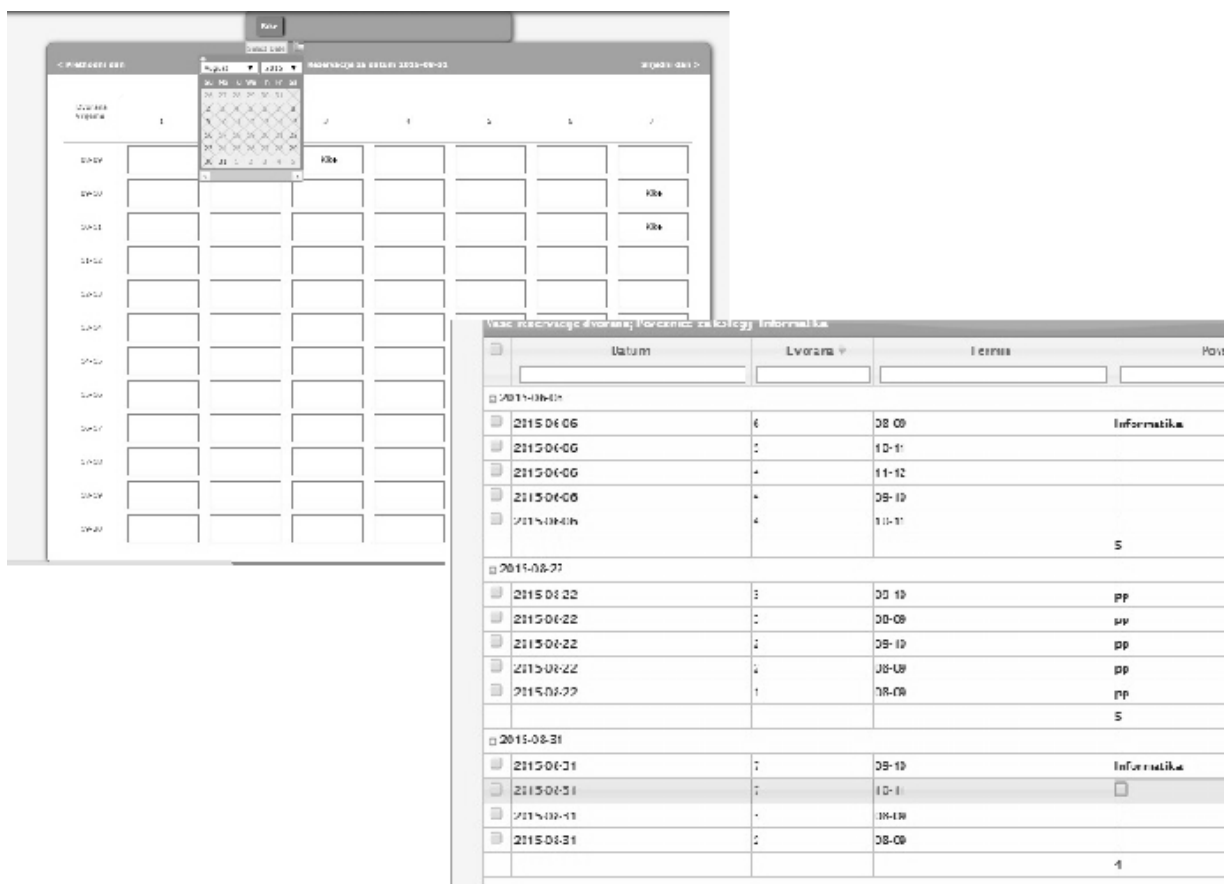
Nakon završetka ispitnog roka, moguće je obrisati stečene bodove koji nisu doveli do zaključivanja konačne ocjene iz kolegija.

4.11.Rezerviranje dvorana i pridruživanje kolegija rezervacijama

Moguće je odabrati s pomoćnog kalendara datum rezervacije, zatim biramo termin i dvoranu klikom na polje.

Ponovnim klikom na isto polje otkazujemo rezervaciju. Kad kursorom prijeđemo preko imena korisnika možemo vidjeti njegovu e-mail adresu (ukoliko ga želimo kontaktirati i zamoliti da nam eventualno prepusti neki termin).

U zajednoj zatvorenoj tablici (koju otvaramo klikom na desni ugao), pod nazivom “Vaše rezervacije dvorana, poveznice na kolegij-****” možemo pregledati također sve naše rezervacije i klikom unijeti „event“, odnosno automatski se klikom unosi ime kolegija nad kojim trenutno administriramo.



Slika 4:17 Rezerviranje dvorana i pridruživanje kolegija rezervacijama

Za sve navedene stavke kolegija je moguće kreirati izvještaj u vidu xls datoteke.

Ostale funkcionalnosti sustava, MVC model, dijagrami klasa, dijagrami aktivnosti, prikazi sučelja, relacije među klasama, njihovi atributi i operacije, tumačenja kodova, upute za instalaciju, korištenje ili testiranje se nalaze u projektnoj dokumentaciji.

Zaključak

Realizirani primjer bi mogao poslužiti kao poticaj ostalim razvojnim programerima početnicima da se upuste u slične projekte proširujući svijet otvorenog koda.

Brzo stvaranje prototipova karakteristično za razvoj projekata otvorenog koda, može ubrzo uzrokovati neuspjeh, no kad je dozvoljeno eksperimentiranje također vodi brzom pronalasku rješenja jer je moguće sagledati problem na nov način i potražiti odgovore na novim mjestima, što ponovo rezultira novom dobrobiti ubrzanog učenja kroz rad.

Stoga otvoreni kod potiče unutarnji motiv za učenjem i stvaranjem svrhe. Možemo zaključiti kako je primjena strategije otvorenog koda i u raznim odgojnim i obrazovnim ustanovama izuzetno korisna kroz svoje različite paradigme.

Ovaj rad je teoretska podloga razvijenoj aplikaciji za administriranje nastavom sa pripremljenom dokumentacijom po principima objektno-orijentirane analize i dizajna.

Prilikom istraživanja osnovnih funkcionalnosti nekih LMS sustava utvrđeno je da niti jedan istražen sustav ne zadovoljava navedene specifične zahtjeve korisnika. Razmotrena je mogućnost razvoja dodatka koji bi proširio njihove funkcionalnosti, rastao je interes za pozadinska zbivanja u sustavu i način razvoja pojedinih dodataka uz doprinos široke zajednice korisnika.

Razvijena aplikacija se može lako pridodati stavkama kolegija određenog sustava bez izmjena koda u samoj jezgri sustava, uz preuzimanje nekoliko identifikacijskih parametara iz matičnog sustava, međutim takva nova funkcionalnost može uzrokovati nepredviđene nestabilnosti zbog neprilagođenosti jezgre sustava. Eventualno rješenje takve situacije može biti postignuto zakrpama, što zahtijeva bolje poznavanje čitave platforme. Ukoliko dodane, javno objavljene funkcionalnosti naiđu na interes i podršku široke zajednice, mogu biti ugrađene u samu jezgru neke od sljedećih verzija (nadogradnji) od strane razvojnog tima sustava.

Čitavo istraživanje literature, praćenje različitih tutoriala i blogova, forumsko konzultiranje sa zajednicom razvojnih programera oko traženja specifičnog rješenja i eksperimentiranje kodom je otvorilo novi niz zanimljivih poglavlja koja nisu obrađena u ovom radu.

5. Literatura

- [1] Open Source Initiative: Licenses & Standards
(Dostupno na: <http://opensource.org/licenses>)
- [2] ŽITKO B. Predavanje:1 Životni ciklus programske podrške, Uvod u programsko inženjerstvo
(Dostupno na: <https://drive.google.com/file/d/0B1kUXPROf-eWX1lPaUlzOU5DcTQ/view>)
- [3] T. HUNT Poglavlje 13: Moodle, The Architecture of Open Source Applications II, 2012.
(Dostupno na: <http://www.aosabook.org/en/moodle.html>)
- [4] SAID ALLY, Security Vulnerabilities of the Web Based Open Source Information Systems: Adoption Process and Source Codes Screening, Journal of The Open University of Tanzania, Volume 17 September, 2014 ISSN 08566739
(Dostupno na:
http://www.out.ac.tz/files/documents/Huria%20Journal%20Vol%2017,2014_2.pdf)
- [5] ATIF KHAN, RUP & Agile (Scrum), University of Waterloo Canada 2011 (Dostupno na: https://cs.uwaterloo.ca/~a78khan/courses-offered/cs446/2010_05/lecture-slides/08_development_processes.pdf)
- [6] Select Business Solutions, Inc., © Copyright 1988-2015, What is SCRUM Development?
(Dostupno na: <http://www.selectbs.com/process-maturity/what-is-scrum-development>)
- [7] KEN SCHWABER, Chapter 11:SCRUM Development Process, BUSINESS OBJECT DESIGN AND IMPLEMENTATION 1997
(Dostupno na: http://navegapolis.net/files/Scrum_Development_Process.pdf)
- [8] RICHEY, R.C. "REFLECTIONS ON THE 2008 AECT DEFINITIONS OF THE FIELD".
TECHTRENDS : 24–25.
- [9] Moodle, Development processes
(Dostupno na: https://docs.moodle.org/dev/Process#Major_release_cycles)
- [10] Cognitive Design Solutions, Inc., ©2003, E-Learning System Architecture
(dostupno na:
<http://web.archive.org/web/20090414122635/http://www.cognitivedesignsolutions.com/ELearning/Architecture.htm>)

Sve navedene poveznice su aktivne na dan : 12.09.2015

Sažetak

U ovom radu razmotrene su osnovne značajke sustava namijenjenih administraciji i sadržaju nastave, način proširivanja sustava dodatnim modulima ili dodatcima Scrum razvojnom metodologijom. Razmotrene su osnovne funkcionalnosti raznih LMS sustava i specifični administrativni zahtjevi korisnika koji trenutno nisu obuhvaćeni u niti jednom LMS rješenju.

Prikazana je integracija zadanog projekta i dviju modificiranih kontrola otvorenog koda, kako bi gotova aplikacija sačinjavala cjelovit sustav.

Kroz projekt rada razvijen je sustav prema konceptualnom modelu objektno orijentirane analize i dizajna. Gotova aplikacija postavljena je na besplatni servis za udomljavanje stranica gdje ju je moguće testirati.

Tagovi: OOAD, LMS, UML, OOP, open source, PHP, Scrum, MySQL

Summary

This paper reviewed the main features of the system intended for the administration and content of learning process, a way of expanding the system with additional modules or plugins using Scrum development methodology. Consideration of the basic functionality of various LMS systems and specific administrative requirements of users which are not currently covered in any reviewed LMS solution.

Present the integration of a given project with two modified open source controls, so that finished application could appear as a complete system.

Through the project work is developed a system according to the conceptual model of object-oriented analysis and design. The finished application is set to a free web hosting service, where it can be tested.

Tags: OOAD, LMS, UML, OOP, open source, PHP, Scrum, MySQL

Skraćenice

API	<i>Application Programming Interface</i>	aplikacijsko programsko sučelje
CMS	<i>Course Management System</i>	sustav za upravljanje tečajem
MOODLE	<i>ModularObjectOrientedLearningEnvironment</i>	modularna OO okolina za učenje
LMS	<i>Learning Management System</i>	sustav za upravljanje nastavom
LCMS	<i>Learning Content Management System</i>	sustav za upravljanje sadržajem nastave
OSIS	<i>Online Student Information System</i>	online informacijski sustav o studentima
RUP	<i>Rational Unified Process</i>	Rational ujedinjeni proces
RFID	<i>Radio-frequency identification</i>	radio-frekvencijska identifikacija
SCROM	<i>Sharable Content Object Reference</i>	referentni model za djeljive objekte sadržaja
SQL	<i>Structured Query Language</i>	strukturirani upitni jezik
ADDIE	<i>AnalysisDesignDevelopmentImplementationEvaluation</i>	

Privitak

Preuzimanje aplikacije i pripadajuće dokumentacije na

<https://www.dropbox.com/sh/pq6atb3h93t716l/AAARo4ekDGSm9JyHKWsuLrlXa?dl=0>