შავი ზღვის საერთაშორისო უნივერსიტეტი

კომპიუტერული ტექნოლოგიების და საინჟინრო საქმის

ფაკულტეტი

ადმინისტრაციული და აკადემიური შესრულების (პერფორმანსის) მენეჯმენტის სისტემა (CMS) განვითარებული უნივერსიტეტისთვის კონტენტ მენეჯმენტის სისტემური ტექნოლოგიების გამოყენებით

მუსა მაჰამატ ბუკარ

სადოქტორო დისერტაციის ავტორეფერატი
კომპიუტერულ მეცნიერებაში

თბილისი / 2013

ხელმძღვანელი : პროფ. დოქტ. ნიაზი არი

ექსპერტები :

პროფ. დოქტ. ირაკლი როდონაია . . . . . .

დოქტ. გიორგი ღლონტი . . . . . .

ოპონენტები:

პროფ. დოქტ. მედეა თევდორაძე . . . . . .

პროფ. დოქტ. ალექსანდრე ედიბერიძე . . . . . .

ნაშრომის ზოგადო მახასიათებლები

ნაშრომის თემის აქტუალობა

კონტენტ მენეჯმენტის სისტემა (CMS) წარმოადგენს კომპიუტერულ აპლიკაციას, რომელიც გამოიყენება სხვადასხვა სახის ციფრული მედიის, კომპიუტერული ტექსტებისა და ელექტრონული დოკუმენტების შექმნის, რედაქტირების, მართვის, კვლევა-ძიებისა და გამოცემის მოზნებისთვის. CMS ხშირად გამოიყენება სპეციფიკური სამრეწველო დოკუმენტაციის, ახალი ამბების საინფორმაციო სტატიების, სახელმძღვანელოების, ტექნიკური სახელმძღვანელოების, სავაჭრო გზამკვლევებისა და მარკეტინგული ბროშურების შენახვის, მართვის, ვერსიების შექმნისა და გამოცემისა და აგრეთვე. სისტემის ავტომატიზებისთვის. შინაარსის მართვა შესაძლოა, მოიცავდეს კომპიუტერულ ფაილებს, გამომსახველობით (იმიჯურ) მედიას, აუდიო ფაილებს, ვიდეო ფაილებს, ელექტრონულ დოკუმენტებსა და Web კონტექსტს. ეს კონცეფტები წარმოადგენენ ინტეგრირებულ და ურთუერთდამოკიდებულ პლასტებს.

ამსფეროში ცნობილია სხვადასხვა ნომენკლატურა: Web კონტენტ მენეჯმენტი, ციფრული აქტივის მენეჯმენტი, ციფრული ჩანაწერების მენეჯმენტი, ელექტრონული კონტენტ მენეჯმენტი, და ა.შ. ამ სისტემის ყველაზე ქვედა დონეს წარმოადგენს კონტენტ მენეჯმენტი და საგამომცემლო საქმეიმ სამუშაო პროცესში, რაც მოითხოვება. ისეთი ინფორმაცია, როგორიცაა საფოსტო მიმოწერა (i-mails), გადახდა-ჩაბარების აქტები, ჩანაწერების წარმოება, სამომხმარებლო ინფორმაცია, ფინანსური ანგარიშები, ვიდეოთეკა და სხვ. შეადგენენ ბიზნეს კონტექსტებს, რომლებიც გამოიყენება ბიზნესის ტრანსაქციების საწარმოებლად.ეს შინაარსი დღითი დღე იზრდება.ამ ელექტრონული შინაარსის მზარდი გაფართოების შესაბამისად, კომპანიები საჭიროებენ კონტენტ მენეჯმენტისა და პროცესუალური მართვის გადაწტვეტილებების ავტომატიზებას პროდუქტიულობის ზრდისა და მთლიანი ოპერაციების გაუმჯობესების მიზნით. კონტენტ მენეჯმენტის გადაწყვეტილებები მართავენ შინაარსის სასიცოცხლო ციკლის ყველა ფაზას, რაც

მოიცავს ინფორმაციის შექმნის, მოწონების, გამოცემის, კვლევა-მოძიების, მოძველებისა და არქივირების სტადიებს. კონტენტ მენეჯმენტის გადაწყვეტილებები გულისხმობენ აგრეთვე, Web კონტენტ მენეჯმენტს, დოკუმენტების მენჯმენტს, ციფრული აქტივის მენეჯმენტს, კოლბორაიციულ (თანამშრომლობით) მენეჯმენტს, ჩანაწერების მენეჯმენტს, საიმიჯო, ბიზნეს პროცესების მენეკმენტს და საწარმოო ანგარიშების მენეჯმენტსაც. აქ განვიხილავთ კონტენტ მენეჯმენტის რამდენიმე ამგვარი გადაწყვეტის ნიმუშს. ყველა ეს სუბიექტი რეალიზდება სხვადასხვა საინფორმაციო ტექნოლოგიური საშუალების (ე.წ. სოფტის (Software) გამოყენებით. მომავალში მოვახდენთ ამ ინფორმაციული ინსტრუმენტების წარმოდგენას, შედარებასა და ილუსტრირებას პრაქტიკული გამოყენების თვალსაზრისით.

კონტენტ მენეჯმენტი (CM) ერთ-ერთი ინსტრუმენტია, რომელსაც საწარმოო საჭიროებს ცოდნის მართვის პროექტის იმპლიმენტაციის საკუთარი მიზნების შესასრულებლად.ესაა მეთოდებისა და ტექნიკების სისტემა კომპანიაში შინაარსობრივი კონტექსტის შეკრების, მართვისა და გამოცემის წარმოებისთვის.ამ პერსპექტივაში CM აღმოცენდა არა კომპიუტერების გვერდით, არამედ წერითი ჩარევისა და პირველი ბიბლიოთეკების დაფუძნებიდან. რამაც განსაზღვრა საგნის აქტუალობა, ეს CMS კონფიგურაცია და ინფორმაციული ტექნოლოგიებია, როგორც პასუხი დოკუმენტებისა და ინფორმაციის საჩვენებელ მრავალფეროვნებაზე, რომელიც მოვიდა ინტერნეტ ტექნოლოგიებთან და მსოფლიო გლობალურ საინფორმაციო ქსელთან ერთად. ამ გამოკვლევაში კონტენტ მენეჯმენტიგანისაზღვრება, როგორც მეთოდებისა და ტექნიკების სისტემა, რომელიც მიმართულიაინფორმაციული ტექნოლოგიების გამოყენებით შინაარსობრივი ერთეულების შეგროვების, მართვისა და გამოცემის პროცესების ავტომატიზებაზე. CM ემყარება შინაარსისაგან დამოუკიდებელი ლოგიკურობის პრინციპებსა და ფორმატს. კონტენტ მენეჯმენტის სისტემა უზრუნველყოფს ინფორმაციის შექმნისა და განაწილების მართვას.იგი ახდენს ცოდნის დაშვების ორგანიზებასა და ინფორმაციის მნიშვნელოვნების მონიტორინგს, აგრეთვე, ინფორმაციის მიმღების (acceptor) გადაწყვეტილებებსა და მოცემული მონაცემების გადაცემის.

თეზისების საგნობრივი ბმულები (კავშირები)

თეზისების თემა განეკუთვნება ე.წ. case-study-ს, კომპიუტერული პროგრამის კონკრეტული შემთხვევის მთლიანად გამოყენებითი ასპექტის კვლევას; სინამდვილეში,მოცემული საგანი თვითგანმარტებითი ხასიათისაა.მოცემული კვლევის ტექნოლოგიური ინსტრუმენტი (ე.წ. სოფტი) განვითარდა ობიექტზე ორიენტირებული პროგრამირების (oop) მეთოდისა და კონტენტმენეჯმენტის სისტემის ტექნოლოგიების გამოყენების შედეგად. მოცემული წყაროს კოდი ჩაიწერა შაბლონური ჩაწერის ენაში.

ნაშრომის მიზნები და ამოცანები

აღნიშნული განვითარებული პილოტური პროექტის მთავარი ამოცანა კონტენტ მენეჯმენტის სისტემის (CMS) მრავალმხრივ გამოკვლევასა და (CMS) ტექნოლოგიიის დანერგვაში მდგომარეობს, როგორც წინამორბედი ადმინისტრირების მზარდი, ტოტალური, სრულად ავტომატიზებული სისტემისადა აკადემიური შესრულების (პერფორმანსი) მენეჯმენტის სისტემის მართვის საქმეში, მეტიც, ამ გამოკვლევის ერთ-ერთი მიზანია **CMS-ის ბაზაზე ინტერაქტიური, დინამიური . სისტემის კონსტრუირება,** თუგანვითარება, რომლის დახმარებითაცსავსებითშესაძლებელია უნივერსიტეტის მართვის სისტემის ადმინისტრირებისა და აკადემიურასპექტებთან დაკავშირებული რიგი პრობლემების გაანალიზება და ამ პრობლემების დაძლევის გარკვეული გზების შემუშავება, პრაქტიკული გატარება და სრულყოფა. გარდა ამისა, ჩვენ გვსურს ახალი კონტენტ მენეჯმენტის სისტემის შექმნა, რომელიც სათანადოდ უპასუხებდა არსებული CMS სისტემის წინაშე მდგარ გამოწვევებს ფუნქციონალურობისა და უნივერსიტეტის შინაარსობრივი ასპექტის მართვის გზების გაუმჯობესების თვალსაზრისით და მას აიყვანდა თვისებრივად ახალ დონეზე.

მთელი ამ კვლევის მანძილზე ჩვენ მივისწრაფით, ისეთი კონტენტ მენეჯმენტის სისტემის პრობლემის გადაწყვეტის გამონახვას, როგორიცააფალსიფიცირებული CMS პროდუქტი გარდა ჩვენი გამოკვლევისა,

რომელიც ჩვენ პირველ თავში წარმოვადგინეთ, აგრეთვე, ვამატებთ რამდენიმე ახალ ფუნქციას, რომლებიც არ არის არსებულ CMS პროდუქტში, მაგალითად, განწყობის მიმართულების მენეჯმენტის სისტემას.

სისტემა ფოკუსირებულია გძელვადიან, სტრატეგიულ საუნივერსიტეტო პრობლემებზე, რომლებიც თავისი ბუნებით, დინამიური და მუდმივია, როგორიცაა სტუდენტისა და ფაკულტეტის მზარდი პროპორცია, სწავლების მწირი ხარისხი, და დაბალი კვლევითი პროდუქტიულობა, აკადემიური კადრების შესრულების (პერფორმანსის) გაუმჯობესების დეფიციტი.აკადემიური ინსტიტუციები შეიძლება გაანალიზდეს სხვადასხვა პერსპექტივიდან, როგორიცაა ერთობის,ინსტიტუციური, კორპორატიული, ორგანიზაციული და თვით პოლიტიკური სისტემის კუთხიაც კი. ყველა ეს მხარე თანაბრად მარტივად არ თანაარსებობს უნივერსიტეტების საზღვრებში, სადაც თითოეულ უნივერსიტეტს უნიკალური,საკუთარი შერეული აკადემიური მიზნები, ადამიანური რესურსები, ინფრასტრუქტურა, კაპიტალბრუნვა და ცვალებადი გარემო გააჩნია. ეს წარმოქმნის უამრავ პრობლემას მის შინაარობრივ ასპექტშიდა საჭიროებს მათი გადაჭრის ეფექტური მექანიზმების რეალურ სისტემას.რასაკვირველია, იგივე ელემენტები შეიძლება ვიპოვოთ ჰოსპიტალურ, ბიზნეს და ინდუსტრიულ ორგანიზაციებშიც, გარდა ამისა, უნივერსიტეტები წარმოადგენენ მულტიფუფუნქციურ ორგანიზაციებს, რომლებთავ აქვთ სწავლებისა და კვლევის ვალდებულება, თუმცა ისინი აგრეთვე, ახდენენ საზოგადოებრივი სერვისის მიწოდებასაც (სამუშაო ერთობის საკეთილდღეოდ).ამ ტიპის ორგანიზაციები შეუძლებელია ფუნქციონირებდნენ, კერძო ბიზნესის მსგავსად, რომელიც საკუთრივ,კერძო მოგებაზე ორიენტირებითოპერირებს.გარდა ამისა, შეუძლებელია ცალკეული, თითოეული ფუნქციითშეტანილი წვრილისმარტივად გარჩევა, რაც ართულებს უნივერსიტეტის ეფექტურობის შეფასების. გაზომვას, რამდენადაც არც შემოსავლისა და არც ხარჯების ზუსტი აღრიცხვა მონეტარულადარ არის შესაძლებელი.

მეორე მხრივ, მენეჯმენტი შეიძლება განვიხილოთ, როგორც განსხვავებული მეთოდოლოგიის გამოყენების პროცესი კარგად სტრუქტურირებული ორგანიზაციის საზღვრებში, თანმხლები ეფექტური ინტეგრაციის პრინციპის მოქმედებით, რაც ორგანიზაციას ცვალებად გარემოსთან წარმატებით ადაპტაციის საშუალებას

აძლევს.სრულყოფის მოთხოვნა აყენებს უმაღლესი განათლების მართვის საფუძველმდებარე პრინციპების იდენტიფიკაციისა და მენეჯმენტის რესურსებზე კონცენტრირების აუცილებლობას, არა მხოლოდ რუტინული აქტივობების ჰარმონიზაციის მიზნით, არამედ ორგანიზაციული ელემენტების იმ სახით კომბინირების მოთხოვნის შესაბამისადაც, რომელიც პროდუქტიულ და ეფექტურ შესრულებასა და უნივერსიტეტის ხარისხის ნიშანს განაპირობებს, რაც აგრეთვე, თავის მხრივ, საჭიროებს ინსტიტუციის შინააარსის მართვისა და განკარგვის დინამიური სისტემის დანერგვას.

ქვემოთ მოცემულია ნაშრომის ზოგიერთი ამოცანის ჩამონათვალი:

1. კონტენტ მენეჯმენტის ახალი დინამიური სისტემის (CMS) განვითარება, რომელსაც ექნება უნივერსიტეტის ადმინისტრირებისა და აკადემიური შესრულების (პერფორმანსის) მართვის უნარი.
2. ახალი ინსტრუმენტის შემუშავება უნივერსიტეტის აკადემიური კადრების შესრულების (პერფორმანსის) გამოკვლევის მიზნით.
3. ღია წყაროს კონტენტ მენეჯმენტის (CMS) მრავალმხრივობის გამოკვლევა.
4. თანამედროვე უნივერსიტეტის მოდელის დანერგვა, როგორც case-study და სათანადოშედეგის მიღება.
5. ალგორითმის შედგენა ობიექტზე ორიენტირებული პროგრამირების (OOP) მეთოდის გამოყენებით.

ნაშრომის მეთოდოლოგია

ნაშრომში გამოიყენება შემდეგი მეთოდები: კონტენტ მენეჯმენტის სისტემის (CMS) მეთოდი, Rubyობიექტზე ორიენტირებული პროგრამირების მეთოდი (OOP), რელსების Rubyმეთოდი.სისტემის მთავარი ალგორითმი ჩაწერილია Ruby-ში, რამდენადაც Rubyწარმოადგენს მთლიან ობიექტზე ორიენტირებული პროგრამირების ენას. MySQL გამოიყენება მონაცემთა ბაზების მართვის სისტემისთვის.

მეცნიერული სიახლე, ნოვაცია

მეცნიერული სიახლე შემდეგში მდგომარეობს:

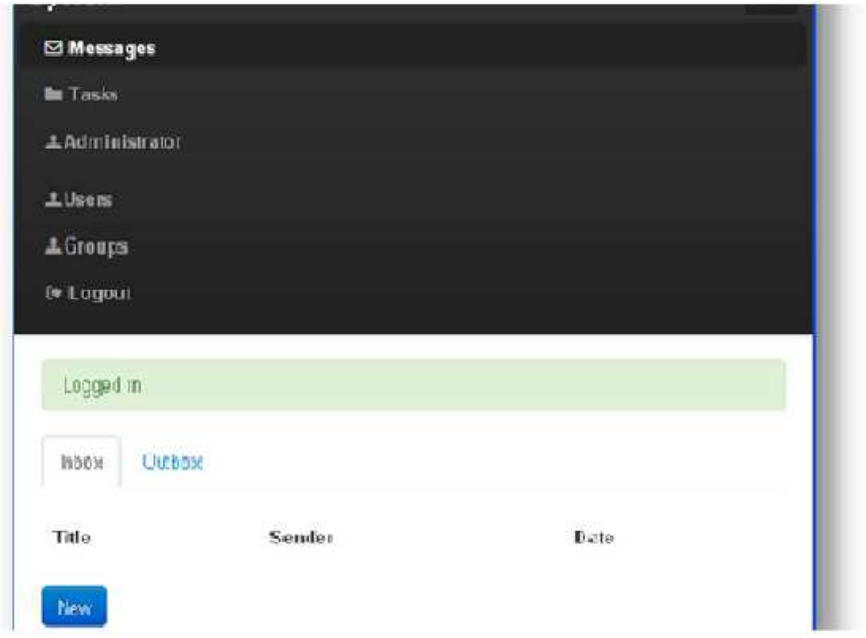
1. ახალი და ორიგინალური დინამიური სისტემა CMS ბაზაზე განვითარდა საგანგებოდ ადმიანისტრირებისა და აკადემიური შესრულების (პერფორმანსის) მართვის სისტემისთვისგანვითარებულ უნივერსიტეტში და შესაძლებელი ხდებამისი დანერგვა ნებისმიერ თანამედროვე ტიპის უნივერსიტეტში.
2. მოცემული სისტემის ალგორითმი პირველად დაფუძნდა კლასიკული ვარიაციული მეთოდების გამოყენებით. მსგავსი მეთოდიარსებობდა, თუმცა მოცემული მეთოდისაგან მნიშცნელოვნად განსხვავდებოდა დინამიურობისა და უნივერსიტეტის ფართო სტრუქტურის შესაბამისობის მხრივ, ამგვარი ცვლილებები. შეიძლება განხორციელდეს სტრუქტურიდან სისტემის წყაროს კოდის შეცვლის გარეშე.
3. ალგორითმი ჩაიწერა სისტემის მთავარი პროგრამისთვისOOP მეთოდის გამოყენებით.
4. სისტემას აქვს აკადემიური კადრების შესრულების (პერფორმანსის) შემოწმების უნარი განწყობოს მიმართულების მენეჯმენტის სისტემის მეშვეობით და აგრეთვე, სხვა მოქმედი სათანადო ტექნიკების საშუალებით.
5. სისტემა დანერგილია IAAU-ში და მიღებული შედეგი ახდენს მძრავრ მოტივირებას აღნიშნულისისტემის იმპლიმენტაციისათვის ჩვენი უნივერსიტეტის ავტომატიზების მიზნით.

ნაშრომის პრაქტიკული შედეგები

ნაშრომის პრაქტიკული მნიშვნელობა მდგომარეობს სისტემის განვითარებული ალგორითმის მრავალი მიმართულებით წარმატებით გამოყენების შესაძლებლობაში ისეთ საგანმანათლებლო ორგანიზაციებში, როგორიცაა უნივერსიტეტი, კოლეჯი, ინსტიტუტიდა სხვ. მოცემული ნაშრომი იმის ჩვენებაა, თუ

CMS ტექნოლოგიის მეთოდი რა დიდი ამოცანების მართვის უნივერსალური შესაძლებლობების მქონე მძლავრი ინსტრუმენტია და როგორი მნიშვნელოვანი წვრილი შეაქვს უნივერსიტეტის მენეჯმენტის სისტემაში.

დინამიური ადმინისტრირებისა და აკადემიური შესრულების (პერფორმანსის) მენეჯმენტის სისტემა წარმოადგემს ინფორმაციული ტექნოლოგიური საშუალების ე.წ. სოფტის (Software) გამოყენებას, რომელსაც ჩვენ განვიხილავთ ამ სადისერტაციო გამოკვლევაში. დმინისტრირების მენეჯმენტის სისტემა უზრუნველყოფილია ყველა საჭირო მახასიათებლით ფართო რიგის აკადემიური, ადმინისტრაციული, უსაფრთხოების, მომსახურების საკითხების შესასრულებლად და ინახავს ყველა ინფორმაციას ცალკეული მონაცემთა ბაზიდან.მას გააჩნია მარტივი მოხმარებისა და მეგობრული დიზაინის ინტერფეისი. სისტემა ემყარება კონტენტ მენეჯმენტის სისტემის (CMS) ტექნოლოგიას პრეზენტაციების წარმოდგენისთვის.მისიიმპლიმენტაცია მოქმნილი და სკალირებადია CMS ღია წყაროსა და პოპულარული ტექნოლოგიების გამოყენების საშუალებით საწარმოო ძალისხმევის განვითარებისა და ინტეგრაციისთვის.სისტემის მნიშვნელოვანი მახასიათებელია აპლიკაციების მუშაობის გავრცობა-გაფართოება მთელი უნივერსიტეტის შიდა მოხმარების ქსელში განაწილების ავტომატიზებისთვის. ქვემოთ მოყვანილი დიაგრამა წარმოადგენს სისტემის მთავარ ინტერფეისს.



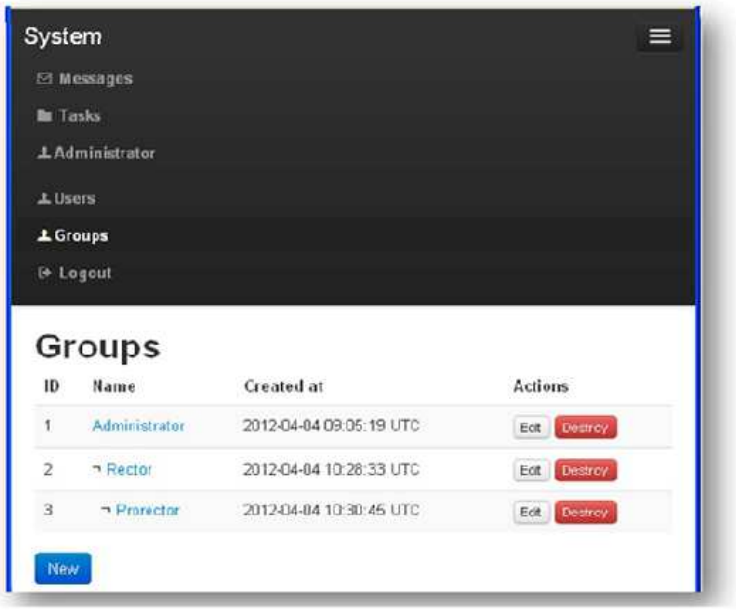
დიაგრამა 4.2: სისტემის მთავარი

(სისტემა, გზავნილები (მესიჯები), ამოცანები, ადმინისტრატორი, მომხმარებლები, ჯგუფები, გამოსვლა, შესვლა ლოგზე, ყუთის შიგნით, ყუთის გარეთ, სათაური, გამგზავნელი, თარიღი, ახალი)

სისტემის ფუნქციონირება

სისტემას საკმაოდ მრავალი ფუნქცია გააჩნია, მხოლოდ სისტემის ადმინისტრატორს აქვს სისტემის ტოტალური მართვის უნარი, რაც სისტემის სრულ კონტროლს გულისხმობს. შისტემას ძალზე ბევრი მომხმარებელი ჰყავს და თითოეულ მათგანს საკუთარი პერსონალურიპაროლი გააჩნია საკუთარ პროფილზე შესასვლელად.მომხმარებელი შეიძლებაშეიქმნას სისტემის ადმინისტრატორის მიერ და ყველა მომხმარებელს საკუთარ პროფილზე კონტროლის უნარი გააჩნია.ეს ჯგუფია ან მომხმარებელი, თითოეულ მომხმარებელს ჰყავს მშობელი ჯგუფის სახით და თითოეულ ჯგუფს გააჩნია თავისი შვილი მომხ,მარებლის სახით, რაც სისტემაში იერარქიულად დაეხმარებათ ინდივიდუალურ მუშაობაში. მრავალი მომხმარებელი შეიძლება შეიქმნას ერთი ჯგუფის საზღვრებში.

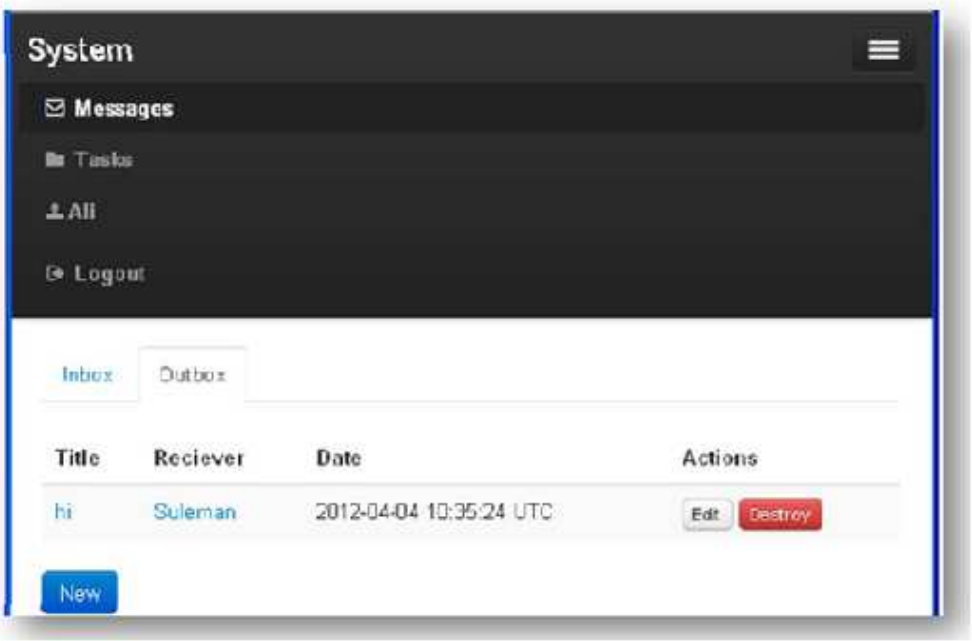
(სისტემა, გზავნილები (მესიჯები), ამოცანები, ადმინისტრატორი, მომხმარებლები, ჯგუფები, გამოსვლა, ჯგუფები, ID, სახელი, შექმნა, მოქმედებები, ადმინისტრატორი, რექტორი, პრორექტორი, რედაქტირება, დაშლა)



დიაგრამა 4.3. სისტემის ჯგუფი

კომუნიკაციის მენეჯმენტის ინსტრუმენტი

ყველა მომხმარებელს შეუძლია გზავნილის გაგზავნა და მიღება სისტემის შიგნით,თუმცა არსებობს გარკვეული შეზღუდვები, რომლის თანახმადაც მომხმარებელს არ ეძლევა ყველასთვის,ვისთანაც სურს,გზავნილის გაგზავნისუფლება,, ყველა მომხმ,არებელს შეუძლია გზავნილის გაგზავნა მხოლოდ იმავე ჯგუფის მომხმარებლებისა, თუ მშობლისთვის.იერარქიის შესაბამისად,სპეციფიკური დეპარტამენტის მომხმარებლებს შეუძლიათ გზავნილის გაგზავნა და მიღება მხოლოდ თავისი დეპარტამენტის შიგნით. შეზღუდვა საკმაოდ ფლექსიბიურობით გამოირჩევა და შესაძლებელია მისი მოხსნა ადმინისტრაციის ბრძანების თანახმად.ქვემოთ მოყვანილი დიაგრამა აჩვენებს, თუ როგორ მუშაობს გზავნილის ინსტრუმენტი

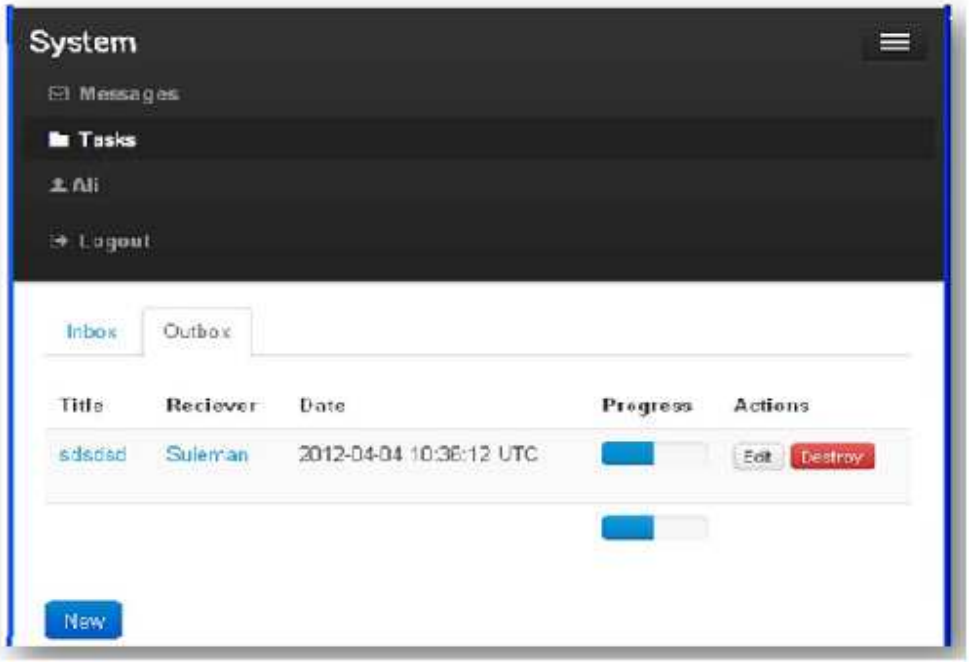


დიაგრამა 4.4. გზავნილის ინსტრუმენტი

ამოცანებისმენეჯმენტის ინსტრუმენტი

ამოცანების მენეჯმენტის ინსტრუმენტი ადმინისტრაციის ზემდგომ წევრს აძლევს უფლებას, მისცეს გარკვეული ამოცანა ადმინისტრაციის წევრს. მაგალითად, ფაკულტეტის დეკანს შეუძლია მისცეს ამოცანა დეპარტამენტის

უფროსს და შემდგომ გაადევნოს თვალი მიცემული დავალების შესრულებას.თუ მიცემული ამოცანა შესრულდა, მაშინ ამოცანის სტატუსი აღინიშნება ალმით ✓სხვა შემთხვევაში, ამოცანის სტატუსი X-ითაღინიშნებაადმინისტრაციის წევრის შესრულების შეფასების თანახმად.ამოცანების მართვის ინდტრუმენტი ძალზე მნიშვნელოვანია, რადგან იგი ეხმარება სისტემას ადმინისტრაციის, ინსტრუქტორის და სხვ. სამუშაოს შესრულების შემოწმებაში და თითოეულ დეპარტამენტს ყოველთვიურ, თუ ყოველწლიურ ანგარიშსაწვდის.ქვემოთ მოყვანილი დიაგრამა აჩვენებს ამოცანის მიცემის ნიმუშს.



დიაგრამა 4.5. ამოცანის მენეჯმენტის ინსტრუმენტი

დოკუმენტების მენეჯმენტის ინსტრუმენტი

დოკუმენტების მენეჯმენტის ინსტრუმენტი ადმინისტრაციის წევრს აძლევს თავისი დოკუმენტების შენახვისა და მათი აღდგენის შესაძლებლობას საჭიროების შემთხვევაში.ყველა შენახული დოკუმენტი გროვდება პირდაპირ სისტემის მონაცემთა ბაზაში და უფრო დაცული ხდება.დოკუმენტი უნდა იყოს doc, pdf, notepad, რაც უბრალოდ ამოტვირთვას საჭიროებს.

ნაშრომის სტრუქტურა და მოცულობა

დისერტაცია შედგება შესავლის, 4 თავის, დასკვნისა და დანართისგან.

თეზისების შინაარსი

თეზისების შინაარსი შედგება 4 თავისგან.პირველი თავი წარმოადგენს ლიტერატურის მიმოხილვას კონტენტ მენეჯმენტის სისტემის (CMS) სფეროში.განხილულია დეტალურად და მიმოხილულია გამოყენების პრობლემების გადაწყვეტა.

თეზისების მეორე თავი ეძღვნება Web პროგრამირების ენას, ჩვენ განვმარტეთშაბლონური ჩაწერის ენების მთელი მრავალმხრივობა და მიმოხილვის მეშვეობით მოვახდინეთ მათ შორის რამდენიმე ყველაზე მნიშვნელოვანი Ruby,Python და Perlშაბლონური ჩაწერის ენის განსხვავება და მოცემული განხილვის შედეგად ჩვენ გვექმნება ერთ-ერთის არჩევის შესაძლებლობა გამოკვლევის საინფორმაციო ტექნოლოგიური ინსტრუმენტის ე.წ. სოფტის (Software) გამოყენებაშიმათი შეტანის მიზნით.მიმოხილვის კვლევა ძალზე მნიშვნელოვანია, რადგან ყველა კონტენტ მენეჯმენტის სისტემის პროდუქცია ჩაწერილია შაბლონური ჩაწერის ენაში (php, ruby, perl, python და სხვ..) და ეს გვეხმარება ალგორითმისა და სისტემის მეთოდის შედგენაში, აგრეთვე, იმ CMSკონტენტ მენეჯმენტის სისტემის გამოვლენაში, რომელსაც ობიექტზე ორიენტირებული პროგრამირების შესრულებამოითხოვს თავისი დინამიზმის გამო.

მესამე თავი ეხება Ruby პროგრამირების ენის მრავალმხრივობას, Ruby ფუნდამენტური თეორიისა და სტრუქტურის პრაქტიკული გამოყენების მხარეს.Ruby პროგრამირების ენა განხილულია დაწვრილებით სინტაქსური და სემანტიკური თვალსაზრისით. სანამ Ruby წარმოადგენდა მოცემული გამოკვლევის პილოტური გამოყენების ნაწილს, ჩვენ შევძელით Ruby-სმრავალი სასარგებლო სინტაქსური ნიუანსის აღმოჩენა სხვა შაბლონური ჩაწერის პროგრამირებაში, როგორიცაა php, perl da python.მუშაობისას პროექტის ეს პროგრამირების ენა გვეხმარება მთავარი ალგორითმის უმარტივესი ხერხით ჩაწერაში, რამდენადაც Ruby სრულადობიექტზე ორიენტირებული პროგრამირების ენაა.

ბოლო თავი შეეხება უნივერსიტეტის სტრუქტურასა და გამოკვლევის გამოყენებით ნაწილს ე.წ. case-study. ბაზისური ინფორმაცია უკავშირდება სტრუქტურასა და უნივერსიტეტების ადმინისტრაციის გადამწყვეტ როლს. გამოკვლევის გამოყენებითი ნაწილის ილუსტრირება მიმდინარეობს ნაბიჯ-ნაბიჯ, გასაგებად მარტივი ფორმით.დანართშიმოცემულია სისტემის წყაროს კოდი.

დასკვნა

მოცემული გამოკვლევის განმავლობაში ზედმიწევნით შემუშავებულ იქნა კონტენტ მენეჯმენტის სისტემის (CMS) ახალი მიდგომის ძირითადი პრინციპები.გამოკვლევაში ნაჩვენებია მთლიანობაში მართვის ისპრობლემები, რომლებიც შესაძლებელია ეფექტურად გადაწყდეს კონტენტ მენეჯმენტის სისტემური ტექნოლოგიების მეშვეობით.

წარმატებული მიმოხილვისთვისსაინტერესოა რეალურად მომქმედი კონტენტ მენეჯმენტის სისტემის (CMS) მახასიათებლები, ტიპები და ატრიბუტები, რომლებიც მომხმარებლებს შესრულებისას მრავალ პრობლემას უქმნიან და მიმოხილვის შედეგად უზრუნველყოფს ისეთ გადაწყვეტას, რომელიც მომხმარებლებს აძლევს უნარს განასხვავონ CMS ბუნება ბაზარზე არსებული CMS პროდუქციისგან.

ობიექტზე ორიენტირებული პროიგრამირების (OOP) მეთოდი მიმართებაშია საინფორმაციო ტექნოლოგიური ინსტრუმენტისე.წ სოფტის .(software) განვითარების პრობლემებთან, რომელიც მართავს სისტემას.Ruby პროგრამირების ენა გამოიყენება სისტემის წყაროს კოდის ჩასაწერად.სისტემა აჩვენებს, რომ Rubyარის სრულად ობიექტზე ორიენტირებული პროგრამირება.რელსებისRuby გამოიყენება სისტემური ინტერფეისის დიზაინისთვის.

ეს გვიჩვენებს, რომ ისეთი პრობლემები, როგორიცაა მართვის სისტემები მყისიერი ინტერნეტ კომერციის, ინტერნეტ სწავლების, ონლაინ-ვაჭრობის, ონლაინ-შეკვეთისთვისა და ა.შ. ეჯახებიან მრავალ პრობლემას მათი კონტექსტის მართვისას და CMS ტექნოლოგიას გააჩნია სრულყოფილად სტრუქტურირებული

ალგორითმი ამ პრობლემების გადასაჭრელად.ადმინისტრირებისა და აკადემიური შესრულების (პერფორმანსის) სისტემის განვითარება ობიექტზე ორიენტირებული პროგრამირების (OOP) მეთოდის კომპიუტერული პროგრამის გამოყენებით, Ruby პროგრამირების ენა და CMS ტექნოლოგიაამ გამოკვლევის შედეგად შეიქმნა. სისტემამ წარმატებით გაიარა ტესტირება ათათურქ ალათუს საერთაშორისო უნივერსიტეტში და ტესტირების შედეგად იგი აღმოჩნდა ადექვატური ადმინისტრირებისა და აკადემიური კადრების შესრულების (პერფორმანსის) სისტემის მართვის შესაძლებლობის შესაბამისი.

მეტიც, მონაცემთა მართვის სისტემა MySql და apache სერვერი გამოიყენებოდა მომხმარებლებისთვის სისტემის ასამუშავებლად, ან სისტემის ექსპლოატაციისთვის ლოკალურად, შიდა მოხმარების მიზნით.

დანართი

სისტემის მთავარი წყაროს კოდის ალგორითმი

კონტროლერის გამოყენება

```
class  ApplicationController<  ActionController::Base
  protect_from_forgery
  helper_method  :current_user
  def  login_required
    if  User.count==0
      @current_user=User.new
    else
      if  current_user
    @unread_messages=@current_user.messages_inbox.find_by_readed(false)
     @unread_tasks=@current_user.tasks_inbox.find_by_readed(false)
     return true
    else
      redirect_to "/sessions/new"
```

```
        return false
      endend
  end
  def  admin_required
    if  current_user  &&  @current_user.id==1

@unread_messages=@current_user.messages_inbox.find_by_readed(false)
       @unread_tasks=@current_user.tasks_inbox.find_by_readed(false)
       return true
    else
        if  !@current_user
          redirect_to "/sessions/new"
        else
          redirect_to "/404.html"
        end
        return  false
    end end

  private
  def  current_user
    @current_user  ||=  User.find(session[:user_id])  if  session[:user_id]

    end
  end
```

Group Controller

```
class GroupsController < ApplicationController
 before_filter :admin_required
  # GET  /groups
  # GET /groups.json
  def index
    @groups = Group.find_all_by_parent_id(0)
    respond_to do |format|
      format.html # index.html.erb
      format.json { render json: @groups }
    end
  end
  # GET  /groups/1
  # GET  /groups/1.json
  def show
    @group = Group.find(params[:id])
    respond_to do |format|
      format.html # show.html.erb
      format.json { render json: @group }
    end
  end
  # GET  /groups/new
  # GET  /groups/new.json
```

```
  def  new
    @group = Group.new
    @groups = Group.all
    respond_to do |format|
      format.html # new.html.erb
      format.json { render json: @group }
    end
  end
  # GET /groups/1/edit
  def edit
    @group = Group.find(params[:id])
    @groups = Group.all
  end
  # POST  /groups
  # POST /groups.json
  def create
    @group  =  Group.new(params[:group])

    respond_to do |format|
      if @group.save
        format.html { redirect_to @group, :flash => {:success
=> 'Group was successfully created.'} }
        format.json  {  render  json:  @group,  status:  :created,  location:
@group }
      else
        format.html  {  render  action:  "new"  }
        format.json { render json:
@group.errors, status: :unprocessable_entity }
```

```
      end
    end
  end

  # PUT  /groups/1
  # PUT  /groups/1.json
  def update
    @group  =  Group.find(params[:id])

    respond_to  do  |format|
      if  @group.update_attributes(params[:group])
        format.html { redirect_to @group, :flash => {:success
=> 'Group was successfully updated.' } }
        format.json { head :no_content
      } else
        format.html  {  render  action:  "edit"  }
        format.json { render json:
@group.errors, status: :unprocessable_entity }
      end
    end
  end
  # DELETE  /groups/1
  # DELETE /groups/1.json
  def destroy
    @group = Group.find(params[:id])
    @group.destroy

    respond_to  do  |format|
```

```
      format.html { redirect_to groups_url
      } format.json { head :no_content }
    end
  end
end
```

Message Controller

```
class MessagesController < ApplicationController
  before_filter :login_required
  # GET  /messages
  # GET  /messages.json
  def index
    @messages = @current_user.messages_inbox
    respond_to do |format|
      format.html # index.html.erb
      format.json { render json: @messages }
    end
  end
  def  outbox
    @messages = @current_user.messages_outbox
    respond_to do |format|
      format.html # index.html.erb
      format.json { render json: @messages }
    end
  end
  # GET  /messages/1
  # GET  /messages/1.json
```

```
  def  show
    @message  =  Message.find(params[:id])
    if (@message.reciever.id != @current_user.id &&
@message.sender.id != @current_user.id )
      redirect_to
    "/404.html" else
      respond_to do |format|
        format.html # show.html.erb
        format.json { render json: @message
      } end
    end
  end
  # GET  /messages/new
  # GET /messages/new.json
  def new
    @message  =  Message.new
    @users = @current_user.group.get_users @current_user
    @users=@users |
    respond_to do |format|
      format.html # new.html.erb
      format.json { render json: @message }
    end
  end
  # GET /messages/1/edit
  def edit
    @message  =  Message.find(params[:id])
    @users  =  @current_user.group.get_users  @current_user
```

```
    if (@message.sender.id != @current_user.id )
      redirect_to "/404.html"
    end
  end
  # POST  /messages
  # POST /messages.json
  def create
    @message  =  Message.new(params[:message])
    @users = @current_user.group.get_users @current_user
    if params[:attachments]
      params[:attachments].each  do  |a|
          attachment=Attachment.new  ({:message  =>  @message,:attached
=>  a})
          attachment.atype="message"
          attachment.save
      end
    end

    respond_to do |format|
      if @message.save
        format.html { redirect_to @message, :flash => {:success =>
'Message was successfully created.'} }
        format.json { render json: @message, status:
:created, location: @message }
      else
        format.html  {  render  action:  "new"  }
        format.json { render json: @message.errors,
status: :unprocessable_entity }
```

```
      end
    end
  end
  # PUT  /messages/1
  # PUT  /messages/1.json
  def update
    @message  =  Message.find(params[:id])
    @users = @current_user.group.get_users @current_user
    if params[:attachments]
      params[:attachments].each  do  |a|
          attachment=Attachment.new  ({:message  =>  @message,:attached
=>  a})
          attachment.atype="message"
          attachment.save
      end
    end
    if  (@message.sender.id  !=@current_user.id  )
      redirect_to  "/404.html"
    else
      respond_to  do  |format|
        if  @message.update_attributes(params[:message])
          format.html { redirect_to @message, :flash => {:success =>
'Message was successfully updated.'} }
          format.json  {  head  :no_content  }
        else
          format.html  {  render  action:  "edit"  }
          format.json { render json: @message.errors,
status: :unprocessable_entity }
```

```
        end
      end
    end
  end
  # DELETE  /messages/1
  # DELETE /messages/1.json def destroy
    @message  =  Message.find(params[:id])
    if ( @message.sender.id !=@current_user.id ) redirect_to "/404.html"
    else @message.destroy
      respond_to  do  |format|
        format.html { redirect_to messages_url } format.json { head
        :no_content }
      end end
  end
end
```

დისერტაციის თემაზე გამოქვეყნებულ ნაშრომთა სია

1- Moussa, Development of object oriented programming by using UML and ObjectiF technologies. 5th International Conference on Electronics and Computer in Kyrgyzstan on 31 October-1 November 2008, p.104-109

2- Moussa, Active University board of trustee using content management system (CMS),6th International Conference on Electronics and Computer in Kazakhstan on 2-3 November 2009, p. 108-112

3- Moussa, Joomla as Content Management System. 7th International Conference on Electronics and Computer in Kyrgyzstan on 1-2 November 2010, p.103-107

4- Mousssa-Mirlan, Time table organizer for university information and management system. 8th Internation conference on Electronic and Computer Kazakhstan on 13-14 December 2011, p. 73-76

5- Moussa, Comparison between difference types of content management system (CMS), Zurich Technique University, Switzerland (Jahresbericht aF&E 2009) Scientific Journal p. 31

6- Moussa, Content Management System(CMS) Evaluation and Analysis,IBSU Journal of Techincal Science and Technologies Vol 1, No1,(2012), p.49-57

7- Moussa, Administration Management system scientific journal of IAAU ,Vol 4 No 2(2009), p.180-184

8- Moussa,Modern High School Automation system Scientific journal of IAAU,Vol 4 No1,(2009), p. 144-153

9- Moussa,Content Management System using Joomla,Scientific journal of IAAU,Vol 4 No2,(2009), p. 142-146