

საქართველოს ტექნიკური უნივერსიტეტი

ხელნაწერის უფლებით

დავით სირბილაძე

მომხმარებელთა მართვის განვითარებული სისტემების
პროექტირება
ბაზისური მომხმარებელთა მართვის სქემების ბაზაზე

სადოქტორო პროგრამა "ინფორმატიკა"

შიფრი 0401

დოქტორის აკადემიური ხარისხის მოსაპოვებლად

წარდგენილი დისერტაციის

ავტორეფერატი

თბილისი

2019 წელი

სამუშაო შესრულებულია საქართველოს ტექნიკურ უნივერსიტეტში
ინფორმატიკისა და მართვის სისტემების ფაკულტეტის " _____
_____ " დეპარტამენტში

სამეცნიერო ხელმძღვანელი: **პროფ. თენგიზ ბახტაძე**

რეცენზენტები: პროფ. რომან სამხარაძე

ასოც. პროფ. გელა ჭიკაძე

დაცვა შედგება 2019 წლის ___ ივლისს, _____ საათზე

საქართველოს ტექნიკური უნივერსიტეტის - ინფორმატიკისა და მართვის

სისტემების ფაკულტეტის სადისერტაციო საბჭოს სხდომაზე,

კორპუსი _____, აუდიტორია _____

მისამართი: 0175, თბილისი, კოსტავას 77.

დისერტაციის გაცნობა შეიძლება სტუ-ის ბიბლიოთეკაში,

ხოლო ავტორეფერატისა - ფაკულტეტის ვებ-გვერდზე

სადისერტაციო საბჭოს მდივანი პროფ. თინათინ კაიშაური

ნაშრომის ზოგადი დახასიათება

თემის აქტუალურობა. თანამედროვე კომპიუტერული ტექნოლოგიები ფართოდ შემოიჭრა ადამიანის საქმიანობის მრავალ სფეროში, მათ შორის თანამშრომელთა სააღრიცხვო სისტემებშიც. ყველა კომპანიასა და ორგანიზაციას სჭირდება სისტემა, რომელშიც თანამდებობრივი პერსონალი გაივლის ავტორიზაციას და მოხდება მათი მოვლენური ლოგი. კომპანიები არიან ძიებაში, რათა დაწერონ საუკეთესო პროდუქტი თავიანთი შიდა მიზნების მიღწევისთვის, რისთვისაც არ ზოგავენ რესურსებს. ავტორი გვაჩვენებს ახლებურ მიდგომას, რომლის მეშვეობით მარტივი მომხმარებლური ბაზა გარდაიქმნება მულტიფუნქციურ ცენტრალიზებულ სისტემად, რომლის მეშვეობითაც ხდება სტრუქტურული სრული მენეჯმენტი იერარქიებისა და დონეების მიხედვით, რაც მოახდინა Microsoft Active Directory მომხმარებლური ბაზისა და PHP პროგრამული ენის შერწყმით და გაერთიანებით LDAP ბმულის მეშვეობით.

ნაშრომში გამოკვეთილია აქტუალური პრობლემა, რომელიც მრავალ ორგანიზაციას გააჩნია. თანამშრომლების სწორად აღრიცხვა არის ძალიან მნიშვნელოვანი ფენომენი, რომელიც გათვალისწინებულ უნდა იქნას და სათანადოდ დამუშავებული და გაანალიზებული.

კვლევის მიზანი. კვლევის მიზანია ორგანიზაციული მომხმარებლების ავტორიზების და აღრიცხვის პროცესის სრულყოფა და გამარტივება, მათი დამუშავებისა და ანალიზის გაუმჯობესება.

კვლევის ობიექტი. კვლევის ობიექტია ორგანიზაციული სტრუქტურები, რომლებიც საჭიროებენ მომხმარებელთა ავტორიზების ფუნქციონალის პროცესის და მათი პარამეტრების იერარქიულად დალაგების და ანალიზის გაუმჯობესებას და გამარტივებას. ესაა დიდი და პატარა ზომის კომპანიები და ფირმები.

მეცნიერული სიახლე. ნაშრომის მეცნიერული სიახლე მდგომარეობს მის უნიკალურობაში და ინოვაციური მოდელის შემუშავებასა და დანერგვაში. ესაა სამომხმარებლო ბაზის გამოყენებისადმი ახალი მიდგომა, რომელიც მოიცავს პროგრამირების ენებისა და პროდუქტიული ინსტრუმენტების ინტეგრირებას, რაც იწვევს ბიზნეს პროცესის სრულყოფას და გამარტივებას საადრიცხო სისტემაში და სამომხმარებლო ინფორმაციის დამუშავებისას. ასეთი მიდგომა გამოიყენება პროგრამულ სისტემებში და მისი მეშვეობით ეფექტურად იმართება ძირითადი პროგრამული სქემები.

PHP ენის მეშვეობით Microsoft Active Directory (AD) სამომხმარებლო ბაზა პროგრამულად უკავშირდება პროდუქტების სერვისების სისტემებს და აქტიურ მომხმარებელთა ინფორმაციას ამუშავებს და იყენებს ანალიზისთვის და მომხმარებლების უფლებების განაწილებისთვის, აგრეთვე პრივილეგიების მინიჭებისა და დახარისხებისთვის, რაც გვამძლევს მძლავრ პროგრამულ მექანიზმს, რომელიც ინერგება ორგანიზაციაში და ინტეგრირდება შიდა რესურსებით ცენტრალიზებულ ბაზასთან და ყველა გამოყენებით მოდულთან.

კვლევის მეთოდები. კვლების მეთოდებია: ორგანიზაციის ავტორიზაციის ფუნქციის ანალიზის მეთოდი, ბაზური ინტეგრირების მეთოდი, პროგრამული ბაზის დანერგვისა და საინფორმაციო ლოკალური ქსელის იერარქიულად დალაგების მეთოდი.

პრაქტიკული ღირებულება. სადისერტაციო ნაშრომის პრაქტიკული ღირებულება შემდეგში მდგომარეობს: შემუშავებული მიდგომა კომპანიებისთვის აუმჯობესებს პროგრამულ მხარეს მომხმარებელთა საადრიცხო ბაზის ინტეგრირების ხარჯზე, რომელიც სინქრონიზებას ამყარებს ყველა არსებულ სერვისთან, მოდულთან და იმართება ცენტრალიზებულად ვებ აპლიკაციის მეშვეობით, რომელზეც წვდომა და მართვა ბევრად იოლი და სწრაფია, ვიდრე სხვა ადმინისტრირების პროგრამის მეშვეობით.

ნაშრომის აპრობაცია. სადისერტაციო თემის მნიშვნელოვანი ნაწილი მოხსენებულია საერთაშორისო სამეცნიერო კონფერენციაზე, აგრეთვე უკვე წარმატებით დანერგილია რამდენიმე დიდ ორგანიზაციაში.

პუბლიკაციები. დისერტაციის თემის მიმართულებით გამოქვეყნებულია ოთხი ნაშრომი სეს-ის მიერ რეკომენდებულ სამეცნიერო ჟურნალებში, მათ შორის ერთი არის უცხო ენაზე.

დისერტაციის სტრუქტურა და მოცულობა. სადისერტაციო ნაშრომი შედგება შესავლის, ორი თავის, დასკვნებისა და გამოყენებული ლიტერატურის სიისგან. საერთო მოცულობა შეადგენს 110 გვერდს.

ნაშრომის შინაარსი

შესავალში დასაბუთებულია პრობლემის აქტუალობა და ნაჩვენებია მეცნიერული კვლევების გააქტიურების აუცილებლობა აღნიშნული პრობლემის გადასაჭრელად. მოცემულია მეცნიერული სიახლე, მიზნები, კვლევის ობიექტი, კვლევის მეთოდები და პრაქტიკული ღირებულება.

პირველ თავში მოყვანილია თემასთან დაკავშირებული ლიტერატურის მიმოხილვა და ძირითადი ბიბლიოთეკები, რომლებზეც დაფუძნებულია ავტორის პროგრამული უზრუნველყოფა. დღესდღეობით ჩვენი ყოველდღიური ცხოვრება გარდამტეხი ცვლილებების პირასაა. სულ უფრო და უფრო აქტუალური ხდება მანქანური სწავლების სფერო, რაც გავლენას ახდენს ჩვენს ყოველდღიურ ცხოვრებაზე. თანამედროვე ტექნოლოგიები უკვე იძლევა ხელოვნური ინტელექტის შექმნის საშუალებას. პროგრამირების ენების მეშვეობით არაერთი მანქანა გარდაიქმნა მოაზროვნე მოწყობილობად. ახლა უკვე შესაძლებელია საკუთარი რობოტის ყოლა, რომელიც დაგვეხმარება სხვადასხვა ამოცანის მაქსიმალური ეფექტურობით შესრულებაში, რაც მათი მრავალფეროვნების ზრდასთან ერთად დღითიდღე ამატებს მათ სახეობებს.

მოცემულ დისერტაციაში განხილულია მომხმარებელთა მართვის გამარტივებული სისტემის დაპროექტების რამდენიმე ძირითადი მეთოდი და დეტალურადაა აღწერილი მათი დადებითი და უარყოფითი მხარეები.

ძირითადი სამუშაო შევასრულე PHP ძრავზე, რომლითაც დავუკავშირდი Microsoft Active Directory მომხმარებელთა და ობიექტთა ქსელურ ბაზას და ანალიზისა დაკვირვების შედეგად ავაწყე სრულყოფილი პროექტი ვებ აპლიკაციის სახით, რომელიც ინტეგრაციისა და სინქრონის მეშვეობით ამუშავებს უფლებების მიხედვით შემოსულ აქტიურ თანამშრომლებს.

თემა აქტუალურია, რადგან თანამედროვე კომპანიებს სჭირდებათ პროგრამული უზრუნველყოფა, რომელიც მარტივად და საიმედოდ იმუშავებს და ხელს შეუწყობს მათ მზარდ განვითარებასა და პროგრესს როგორც ტექნოლოგიურად, აგრეთვე ტექნიკურად. გამოყენების ერთ-ერთი ყველაზე ფართო მაგალითი არის ინტრანეტი, რომელიც გაუადვილებს კომპანიის სამმართველოს ადმინისტრირების პროცესს და დაუზოგავს დროს მათ ქსელურ და სისტემურ ადმინისტრატორებს, რომელთა ფუნქციის ნაწილობრივ შესრულებას შეძლებს ჩვენს მიერ შემოთავაზებული პროგრამა. დღევანდელი თანამედროვე სამყარო ტექნოლოგიური განვითარების პიკზე დგას. პროგრამებისა და აპლიკაციების აწყობა უკვე მულტიმომხმარებლური მიდგომითაა საჭირო. ყველაფერი ეყრდნობა მსოფლიო ქსელსა და ინტერნეტს. შესაბამისად, აპლიკაცია, რომელსაც უშვებენ კომპანიები, უნდა იყოს უნივერსალური, უნდა იყოს ყველასთვის ხელმისაწვდომი და გასაგები. ხალხის ყოველი კატეგორიისთვის ცალკე აპლიკაციის დაწერა არის დიდი დროისა და რესურსის კარგვა, რისთვისაც ინერგება სისტემები, რომლების მიხედვითაც ხდება მომხმარებლების მახასიათებლების ამოცნობა და მათი გარჩევა ერთმანეთისგან.

ნაშრომში განხილულია მიდგომები მომხმარებლებისა და მათი მახასიათებლების ამოსაკითხად სწრაფად, მარტივად და საიმედოდ. თემაში შემოთავაზებულია მათი სუსტი და ძლიერი მხარეების გამიჯვნა

ერთმანეთისგან და სწორი მიმართულება საკუთარ აპლიკაციაში დანერგვისთვის.

მაგალითად, მომხმარებელს, რომელსაც უყვარს ლურჯი ფერი და მომხმარებელს, რომელსაც უყვარს მწვანე ფერი, არ უნდა მივცეთ აპლიკაცია, რომელიც არის წითელი ფერის. ასევე გამორიცხულია მაგათთვის ცალ-ცალკე ფერის აპლიკაციის შექმნა. ყოველ მომხმარებელს უნდა ჰქონდეს საშუალება, რომ თვითონ აირჩიოს მისთვის სასურველი ფერი, რა ქმედებაც უნდა დაიმახსოვროს აპლიკაციამ. მომხმარებელი ერთხელ რომ აირჩევს პარამეტრს, მეორედ შესვლისას იმ არჩეული პარამეტრით უნდა ჩაიტვირთოს პროგრამა. მომხმარებლების ერთმანეთისგან გამორჩევისთვის არსებობს რამდენიმე მიდგომა. ერთია საკუთარი მომხმარებლების ბაზის შექმნა და მასთან მუშაობა, მეორე კი არის რამე არსებული მომხმარებლური სისტემის ინტეგრირება საკუთარ აპლიკაციაში. ორივე მიდგომას აქვს თავისი პლიუსებიც და მინუსებიც.

თუ საკუთარი ბაზის აწყობა გვინდა, მაშინ ბევრი მუშაობა მოგვიწევს და ყველაფერი უნდა გავხადოთ დამოკიდებული მასზე. ყველა ნაწილი იქნება მიბმული მასზე და იქნება მუდმივი ჩაწერა-წაკითხვა და თუ რამეს არასწორად გავაკეთებთ, მაშინ ჩამოვკიდება მთელი აპლიკაცია ან როგორც მინიმუმ რამე ნაწილი იმუშავებს არასწორად. შესარჩევია შესაფერისი და სათანადო ბაზა, სადაც გასათვალისწინებელია აპლიკაციის მოთხოვნები, მომხმარებლების მოცულობა, რაოდენობა, ლოკაცია, სეგმენტი და ა.შ. თუ ავირჩევთ MySQL ბაზას - იგი სწრაფია, მაგრამ თუ არ გავმართავთ, არ არის სტაბილური, შეიძლება გაჭედოს, ვერ დაამუშავოს მონაცემები და შეჩერდეს მთელი პროგრამა მაგის გამო. თუ ავირჩევთ MSSQL ბაზას, მაგას სჭირდება დიდი რესურსი, მაგრამ საიმედოობაც მეტია. ასევე მინუსია, რომ წერა მეტი სჭირდება. გასათვალისწინებელია აპლიკაციის ძრავიც. მაგალითად თუ გვაქვს PHP სერვერი, მას უჭირს MS ბაზებთან მუშაობა და ამ პრობლემის მოგვარებაზე ცალკე დრო დაიხარჯება. არის როგორც ფასიანი, ასევე უფასო

ვერსიები. უფასო ვერსიები შეზღუდულია შესაძლებლობებში. თუ გაკმაყოფილებთ ლიმიტები, მაშინ შეიძლება დაპროგრამების დაწყება.

ასევე არსებობს შერეული სისტემები, რომლებშიც ინტეგრირებულია როგორც გარე სამომხმარებლო ბაზები, ასევე შიდა სერვისები, რომლებსაც იყენებენ ინფორმაციის გადამუშავებისთვის და დახარისხებისთვის. თუ არის შესაძლებლობა, მაშინ ეგ ყოველმხრივ უფრო მოქნილია და გამოსადეგი. წვდომები ნაწილდება. ზოგი იმართება გარე სისტემის მიერ, ზოგი კი - იწერება პერსონალურ ბაზებში. ამ ორის ერთად შერწყმით კი გვევლინება ერთი დიდი სრულყოფილი პროგრამა მძლავრი სქემის სახით.

ერთ-ერთი ყველაზე ფართოდ მოხმარებადი სისტემა არის ინტრანეტი, რომელსაც იყენებს ფაქტიურად ყველა კომპანია. იგი ძირითადად გამოიყენება და მოიხმარება შიდა კომუნიკაციისთვის და ფაილების გაცვლისთვის. შესაბამისად, სისტემა უნდა მუშაობდეს ლოკალურად და არ უნდა იყოს მისი ქსელს გარეთ გასვლა და ინფორმაციის გაჟონვა. უსაფრთხოება პრიორიტეტი არ არის ასეთ სისტემებში, მაგრამ როცა კომპანია დიდია, შესაძლებელია რომ ზოგიერთი თანამშრომელი აღმოჩნდეს ბოროტმოქმედი და პროგრამის სისუსტეები გამოიყენოს ორგანიზაციის საწინააღმდეგოდ. მიმსგავსებული პუბლიკაციები ინტერნეტ სივრცეში ფაქტიურად არ არსებობს და არ აქვს არავის ამ დონემდე სრულყოფილად მიყვანილი სისტემა.

არის რაღაც ფრაგმენტები, პატარა ბიბლიოთეკები, რომლებიც დამუშავებენ ჩემი თემის დასრულებასა და დახვეწაში, მის ამუშავებაში და რეალიზაციაში.

მეორე თავში განხილულია თემის პროგრამული უზრუნველყოფის ასპექტების ანალიზი და დანერგვის დეტალური აღწერა, კვლევის მსვლელობის განვლილი პუნქტები, შედეგები და მათი განსჯა. ერთ-ერთი მთავარი პროგრამა, რაზე დაფუძნებითაც მაქვს გაკეთებული ნაშრომი არის PHP AdeLdap, რაც არის ძალიან მნიშვნელოვანი იარაღი პროგრამისტებისთვის. მისი მეშვეობით შესაძლებელია PHP ძრავიდან

დაკავშირება Active Directory ბაზებთან და მომხმარებლებისა და ობიექტების შესახებ ინფორმაციის წამოღება და ჩაწერა.

აგრეთვე გამოყენებული მაქვს Microsoft Active Directory ქსელური ადმინისტრირების ბაზა, რომელსაც აქვს მრავალ პროგრამასა და პროდუქტთან ინტეგრირების საშუალება მარტივი მიდგომებით. ვინაიდან Microsoft-ს აქვს მოწინავე ოპერაციული სისტემა, სწორედ ამოტომ ავირჩიე მასზე დაფუძნებული სისტემის აწყობა და დანერგვა, რაზეც მაქვს დაწერილი ჩემი სადისერტაციო ნაშრომი. პროგრამის ვიზუალიზაციისთვის ვიყენებ Bootstrap ძრავს, რომელიც არის მსუბუქი, თანამედროვე და ყველა პლატფორმაზე მორგებული, როგორც მობილურებზე, ასევე პლანშეტებზე, ლეპტოპებზე და პერსონალურ კომპიუტერებზე.

დღესდღეობით ყველა კომპანია და ორგანიზაცია ფიქრობს განვითარებასა და პროგრესზე როგორც თავისი კომპანიის, აგრეთვე მისი შიდა ინფრასტრუქტურის. ყველას სჭირდება შიდა საკომუნიკაციო და ფაილების გაცვლის საშუალება, რისთვისაც ზოგი ყიდულობს პროგრამებს, ზოგი - ლიცენზიებს, ზოგი კი საკუთარი რესურსით წერს მათ. საკუთარი პროგრამის ქონა რა თქმა უნდა კარგია, მაგრამ მას სჭირდება ბევრი მოვლა და სანამ შედეგამდე მივალთ, ძალიან ბევრი პუნქტია გასავლელი. თუმცა ყველა მოდულის დაწერა აუცილებელი არ არის. როდესაც ვმუშაობთ Microsoft პროდუქტთან, მას აქვს ძალიან მძლავრი ცენტრალიზებული ბაზა Active Directory, რომლის გამოყენებაც შეგვიძლია სისტემებში ავტორიზაციისა და აუთენტიფიკაციისთვის.

მე დავწერე აპლიკაცია, რომელიც PHP-ის მეშვეობით უკავშირდება AD ბაზას და ასინქრონებს მომხმარებლებთან ნებისმიერ სისტემას. ამ მიდგომის მეშვეობით შესაძლებელია ნებისმიერი პროექტის ავტორიზაციის გატარება Windows მომხმარებლის სახელისა და პაროლით ისევე, როგორც მისი მონაცემების მართვა და დაჯგუფება სხვა მომხმარებლებთან, რომლებიც არიან მსგავს პოზიციაზე, დეპარტამენტში, ფილიალსა ან ორგანიზაციულ სტრუქტურაში. მიდგომა, რომელიც გამოვიყენე, ბევრ

დაწესებულებას გაუმარტივებს შიდა მუშაობის პროცესს. მე უკვე დანერგილი მაქვს იგი საქართველოს რამდენიმე დიდ ორგანიზაციაში, რომლებიც წარმატებით ფუნქციონირებენ ბაზარზე. მისი მეშვეობით უამრავი რამის გაკეთებაა შესაძლებელი, რასაც დეტალურად განვიხილავ მოცემულ სადისერტაციო ნაშრომში. უპირველესია მოხმარებლების დახარისხება დეპარტამენტებისა და ფილიალების მიხედვით. ყველა პოზიციას თავისი წვდომა და უფლებები უნდა ჰქონდეს. ამის საკუთარი სახსრებით გათვალისწინება და დაწერა დიდ რესურსს საჭიროებს.

ძირითადად კომპანიები Windows პლატფორმებზე მუშაობენ და შესაბამისად სერვისიც უნდა ამოვირჩიოთ მასთან თავსებადი. საუკეთესო არჩევანია, როცა მოხმარებლები ცალკე კი არ შეიქმნება, არამედ გაივლიან ავტორიზაციას ავტომატურად - კომპიუტერის სახელისა და პაროლის მეშვეობით. ამის საშუალებას უკვე იძლევა თანამედროვე პროგრამები და სისტემები.

ყველა მომხმარებელი მოიაზრება ერთი დიდი დომენის ქვეშ, რომლებსაც მართავს ადმინისტრატორი. სისტემური ადმინისტრატორები მომხმარებლების მენეჯმენტისთვის იყენებენ Active Directory-ს (აბრევიატურა - AD), სადაც შესაძლებელია თანამშრომლებზე ბევრი დამატებითი ინფორმაციის შევსება, როგორცაა მათი მისამართი, ტელეფონის ნომერი, ფოსტა, სახელი/გვარი და ა.შ. ეს ყველაფერი ინტეგრირებადია სპეციალური ბიბლიოთეკისა და დაპროგრამების სკრიპტული ენის მეშვეობით. სინქრონიზაციას ვუკეთებთ ბაზას და გვაქვს ყველა ის მომხმარებელი, რომელიც რეგისტრირებულია და ვარჩევთ მარტივად თითოეულ მათგანს და შესაბამისად ვურთავთ უფლებებს. იმისთვის, რომ შევძლოთ ქსელში მყოფი ყველა კომპიუტერის მართვა და დახარისხება, ანუ ადმინისტრირება, ისინი უნდა იყვნენ ერთი დომენის ქვეშ, რომელსაც დავუკავშირდებით AD-თი.

AD მომხმარებლების ბაზის ინტეგრირება შესაძლებელია მრავალ პროგრამაში. იგი გამოიყენება როგორც Outlook ფოსტაში, ასევე

კომუნიკატორში - Microsoft Lync, ასევე Microsoft-ის სხვა გარემოებშიც, როგორცაა მაგალითად SSRS. თვითონ Microsoft Lync-იც შესაძლებელია დაინტეგრირდეს და გაეშვას Skype-დან სპეციალური კონფიგურაციის საფუძველზე, რომელსაც ფაქტიურად ყველა ის ხედვა აქვს, რაც Lync-ს.

Microsoft Lync არის საკომუნიკაციო პროგრამა, რომელში ავტორიზების მერე ჩნდება საშუალება ყველა ოდესმე შემოსული კოლეგის მოძებნის, რომლებიც არ არიან გაუქმებული. მას აქვს ბევრი ფუნქცია - როგორც ჯგუფური ზარების წამოწყების შესაძლებლობა, ასევე ვიდეო კონფერენციები, ფაილების გადაგზავნა, მიმოწერის ისტორიის ნახვა, რომელიც ინახება და სინქრონდება მთავარი სერვერიდან. თუმცა AD-ზე წვდომა ყველას არ უნდა მიეცეს. მისი სტრუქტურის ნახვის და შეცვლის უფლება უნდა ჰქონდეს მხოლოდ სისტემურ ადმინისტრატორს და პროგრამისტს, რადგან იგი მოიცავს დიდ და კონფიდენციალურ ინფორმაციას. ამ ბოლო სისტემას - Microsoft SQL Server Reporting Services-ს სჭირდება ერთი ლინკის გაწერა დასაკავშირებლად, სადაც არის AD-ს სერვერის მისამართი, საიდანაც წამოიღებს შევსებულ ინფორმაციას და გამოიყენებს მას რეპორტების დახარისხებისთვის და შევსებისთვის.

AD მართავს აბსოლუტურად ყველა კომპონენტს - როგორც მომხარებლის სურათს, ასევე მის გრაფიკს, სამუშაო დროებს, წასვლა-მოსვლებს, ტექნიკის ფლობებს, სტატუსს და ა.შ. მისი მეშვეობით მარტივად ხდება ინფორმაციის გაცვლა სხვადასხვა აპლიკაციებს შორის, რადგან ყველა დაფუძნებული იქნება ერთ ძირითად ბაზაზე - AD-ზე. მაგალითად, მომხმარებელი თუ აღნიშნავს Windows Calendar-ში, რომ ოთხშაბათს ისვენებს, ეგ სინქრონდება მის საფოსტო ყუთთან და მის კოლეგებს უჩვენებს, რომ აღნიშნული პიროვნება ოთხშაბათს არის ოფისს გარეთ. მსგავსი პრინციპების დანერგვა შესაძლებელია მრავალ პროგრამას შორის. იმისთვის, რომ შევძლოთ AD-ს მართვა, საჭიროა მისი ადმინისტრირების პროგრამის დაინსტალირება კომპიუტერში.

AD-ს აქვს ხის იერარქია. ყველაფერი იშლება მშობლიდან შვილისკენ. თითოეულ განშტოებას ჰყავს ერთი მშობელი და შეიძლება ჰყავდეს რამდენიმე შვილი. თითოეულ განშტოებაზე დაჭერით გამოდის მისი შიგთავსი ობიექტები, რომლებიც შეიძლება იყოს როგორც ჯგუფი, ასევე ელემენტი. ჯგუფები არის რამდენიმე ელემენტის ნაკრები, რომელიც რაღაცა წინასწარ განსაზღვრული ლოგიკით აერთიანებთ მათ. "მშობელი" AD-ს ენაზე არის MemberOf, რომელიც მიუთითებს ელემენტის ორგანიზაციული სტრუქტურის ერთეულზე. შეიძლება ერთდროულად იყო რამდენიმე ჯგუფის წევრი - Domain Users და Database Developer. ყველა ელემენტი აუცილებელია იყოს რამე ჯგუფის წევრი. ნებისმიერი ელემენტისთვის შესაძლებელია როგორც მისი მშობლის ნახვა, ასევე მისი შვილის ნახვა. მშობლიდან მერე გადავალთ მის მშობელზე და ა.შ. სანამ არ მივალთ მთავარ განშტოებამდე. ანუ არის ორი ტიპის ობიექტი - საქაღალდე ან User (მომხმარებელი). AD-სთან ინტეგრირებით და მასზე პროგრამული მანიპულაციით შევიძლება უნივერსალური აპლიკაციის შექმნის მიდგომა - ინტრანეტი. ინტრანეტი კი აქვს რამდენიმე განვითარებულ კომპანიას კომპანიას, მაგრამ ისინი ძირითადად ყიდულობენ მზა პროდუქტებს, რომლებიც საკუთარ ძრავზე მუშაობენ - ეს კი პრობლემის გადაჭრის იოლი გზაა, მაგრამ ძალიან ძვირი.

ზოგიერთ კომპანიას აქვს ასევე საკუთარი ინტრანეტიც, მაგრამ მასზე მუშაობას საკუთარი რესურსით ძალიან დიდი დრო უნდა. AD ბიბლიოთეკის მეშვეობით, პატარა სკრიპტის დაწერით და ძლიერი პროგრამისტით ეს პრობლემა მარტივად წყდება. ინტეგრირდება Windows-ის სამომხმარებლო ბაზა და ავტორიზაციის სისტემაც მარტივდება და ასევე სრული ინფორმაციის წყაროც ჩვენს ხელთაა. ერთხელ დაწერის შემდეგ კი მისი გამოყენება შეიძლება მრავალ მოდულში. მომხმარებლების უფლებები და ინფორმაცია ხდება წვდომადი პროგრამიდან და მისი დამუშავება და გამოყენება შეიძლება ნებისმიერ ადგილას. ავტორიზაციაზე იდება წვდომები და ყველა მომხმარებელს თავისი ჯგუფის მიხედვით ენიჭება

ხედვები ველებზე და მათი ჩართვის შესაძლებლობა. მაგალითად, ადმინებს ექნებათ კონფიგურაციის გაკეთების საშუალება, საიდანაც ასინქრონებენ სხვების წვდომებს. მომხმარებლების სიისთვის გამოვიყენე სტანდარტული DataTable ბიბლიოთეკა, რომელიც დავაკონფიგურირე ისე, რომ მოსახერხებელი იყოს მისი გამოყენება. თითოეული უფლების ჩართვა-გათიშვით კონტროლირდება მომხმარებლების უფლებები პროგრამის ნაწილებზე. მაგალითად, თუ ჩართული ექნებათ კლიენტების ძებნის ფუნქცია, მაშინ დაინახავენ მას, ხოლო თუ გავუთიშავთ შესაბამის უფლებას, მაშინ წვდომა აღარ ექნებათ მასზე.

უფლებების ჩაშლა ასევე ხდება პარამეტრების მიხედვით. ზოგს შეგვიძლია მხოლოდ ნახვის რეჟიმი ჩავუერთოთ, ზოგს რედაქტირების, ზოგსაც მოდერირების თითოეული გვერდის ნაწილის ჭრილში. რეპორტინგისთვის როდესაც ვიყენებთ Microsoft-ის პროდუქტს, მანდაც შესაძლებელი ხდება მარტივად ინტეგრაცია AD-ს სამომხმარებლო ბაზასთან და მისი აუთენტიფიკაციის გამოყენება.

დომენური სახელისა და პაროლის შემდეგ იძლევა საშუალება სერვერის რეპორტების ნახვისა, რაც მოიცავს დიდ უსაფრთხოებას და საიმედოობას, ისევე როგორც დიდი დროის დაზოგვას.

რეპორტინგ სერვერის საქალაქლებზეც მოქმედებს იგივე უფლებები. წვდომების ჩართვა შეიძლება როგორც კონკრეტულ პირზე, ასევე ჯგუფზე. ჯგუფური უფლების გაწერა უფრო მიზანშეწონილია, რადგან მაგალითად თუ გვინდა, რომ რუსთავის სერვის ცენტრის მენეჯერს ჰქონდეს რაღაც რეპორტის გენერირების უფლება და 1 წელში იცვლის პოზიციას ან მიდის სამსახურიდან, მაშინ მისი წვდომა აირევა და იქნება თავიდან გასაწერი, ხოლო თუ უფლებას ჩავუერთავთ მისი პოზიციის ჯგუფს, მაშინ მის შემცვლელს უკვე ექნება იგივე უფლებები, რაც ჰქონდა ძველ მენეჯერს, ყოველგვარი ხელახალი კონფიგურაციის გარეშე. რეპორტინგ სერვერზე უფლებები შეიძლება ჩაერთოთ როგორც კონკრეტულ რეპორტზე, ასევე საქალაქლებზე. საქალაქლებზე წვდომის ჩართვით შეგვიძლია მის ყველა

რეპორტზე მივცეთ დაშვება, ან შიგნითაც მანუალურად ვაკონტროლოთ და ზოგი ვაჩვენოთ, ზოგი კი - დავუმალოთ.

რეპორტინგ სერვერს შეუძლია დანახვა AD-ს თითოეული მოდულის, რომელიც ზევით აღვწერეთ. AD ხომ ჩვეულებრივი ბაზაა, რომელზეც შესაძლებელია სკრიპტი დაიწეროს და წამოვიღოთ ნებისმიერი ვარიაციის ინფორმაცია, რასაც შემდგომ SQL-ში გავარჩევთ ერთმანეთისგან და დავადებთ შესაბამის პირობას და ლოგიკას იერარქიის მიხედვით. რეპორტის აწყობისთვის საჭიროა Visual Studio და რეპორტინგის სერვისი, რომელიც ინტეგრირდება მასში. რეპორტები ეწყობა ვიზუალურად, ხოლო მათი სკრიპტისთვის გამოყოფილია სპეციალური სივრცე. ყველა კატეგორიისთვის იწერება ცალკე სკრიპტი და ბოლოს რეპორტში ჯამდება და ამოდის საერთო დიდი ლოგიკის მქონე რეპორტი. ყველა სკრიპტს ერქმევა უნიკალური დასახელება. მათი დაწერა შეიძლება როგორც ლოკალურად, ასევე გლობალურად, რომ პაკეტში მყოფმა სხვა რეპორტებმაც მიმართონ.

Visual Studio-ს ყოველი ახალი ვერსიის გამოსვლით უმჯობესდება რეპორტინგ სერვერის ინტერფეისი, ფუნქციონალი, ვიზუალი, ემატება ავტომატური რეპორტების გენერირების საშუალებები და იხვეწება გრაფიკები, ემატება სხვადასხვა სისტემებთან ინტეგრირების მხარდაჭერები. წამოღებული ინფორმაციის გარჩევისთვის და რეპორტში გამოყენებისთვის არის ჩანართი დაპარამეტრებისთვის, საიდანაც SQL-ს გადაეცემა შევსებული ცვლადის მნიშვნელობა. მნიშვნელობა შეგვიძლია წამოვიღოთ როგორც ავტომატურად სერვერიდან, ასევე გადავცეთ სტატიკურად ხელით.

AD-დან შეგვიძლია როგორც სრული მონაცემები წამოვიღოთ, ასევე გაფილტრული. პროგრამას ჩაშენებული აქვს შემოსული მომხმარებლის ამომცნობი ფუნქცია, რომლის გამოძახებისთვის საჭიროა მანუალური პარამეტრის გამოძახება და მინიჭება. ასევე შესაძლებელია შემოსული მომხმარებლის დადარება SQL-ის ბაზიდან წამოსულ მონაცემებთან და

გამიჯვნა ერთმანეთისგან. ამ მიდგომით შევძლებთ დინამიური რეპორტების შექმნას. თუ გვინდა რეპორტი, რომელიც სხვადასხვა ფილიალისთვის და პოზიციისთვის უნდა აგენერირებდეს სხვადასხვა ლოგიკით ერთი და იგივე მონაცემებს, მაგის გასაკეთებლად არაა ამ ყველა ვარიაციის ცალ-ცალკე რეპორტის შექმნა.

ინტრანეტი კომპანიას სჭირდება იმისთვის, რომ მისმა თანამშრომლებმა შეძლონ ერთმანეთთან კომუნიკაციის დამყარება და ფაილების გაცვლა გარე სერვისებისგან დამოუკიდებლად. ინტრანეტებისთვის შესაძლებელია მრავალი ინტერფეისის მიცემა და ორგანიზება, გაფორმება ორგანიზაციის საქმიანობიდან გამომდინარე და მასზე დაყრდნობით მრავალი ფუნქციონალის მიმატება. იგი არის ჩვეულებრივი პროექტი, რომელიც განვითარებას საჭიროებს. იმ შემთხვევაში, თუ დაწესებულების პოლიტიკა არ იძლევა AD-ზე დაშვების უფლებას, შესაძლებელია ინტეგრირება და ავტორიზაციის სხვა სისტემებთანაც, ან პირადი სისტემის შექმნა, რომელიც აგრეთვე შესაძლებელია გახდეს ზოგადი მოდული სხვადასხვა სისტემებისთვის, სადაც იქნება პირადი ბაზა და შიფრაცია.

ინტეგრაციის დროს უნდა შედგეს Job-ი, რომელიც ავტომატურად გაეშვება დროის რაღაც პერიოდში, მაგალითად ყოველ ღამე და წამოიღებს თანამშრომლებს და ჩაწერს ბუფერში. სტანდარტულად AD-დან ავტორიზებისთვის ვიყენებ adLDAP ბიბლიოთეკას, რომელსაც აქვს ღია ძრავი და დაწერილია PHP ენაზე. იგი გვამძლევეს საშუალებას, რომ კომპანია პროგრამულად ძალიან მაღალ დონეზე განვავითაროთ. შესაძლებელია ერთი პროგრამის დაწერა, რომელიც მართავს ყველა სტრუქტურას. ამ სისტემით შესაძლებელი იქნება როგორც მომხმარებლების რეგისტრაცია, ასევე რედაქტირება და გაუქმება, სინქრონი ყველა სისტემასთან, გამოყენება, დამუშავება, ანალიზი. adLDAP-ის მეშვეობით შესაძლებელია როგორც მომხმარებლებისთვის ავტორიზების გაკეთება, აგრეთვე სესიის დახურვა, რაც შესაძლებელია, რომ ჩავიმახსოვროთ და შესაბამისად გვექნება

თანამშრომელთა აღრიცხვა - შემოსვლა/გასვლები, რაც გვადლევს ღრმა ანალიზის გაკეთების საშუალებას იმავე, ერთი უნივერსალური აპლიკაციიდან მარტივად.

AD-ში ძიებისთვის არის შემდეგი სკრიპტი:

```
$ad = new Adldap($configuration);  
$results = $ad->search()->all();
```

შეგვიძლია რომელიმე კონკრეტული თანამშრომლის ინფორმაციის ძიება:

```
$results = $ad->search()->where('cn', '=', 'John Doe')->get();
```

ან სახელის რომელიმე კონკრეტული ნაწილის მიხედვით ძიება:

```
$results = $ad->search()->where('cn', 'starts_with', 'John')->get();  
$results = $ad->search()->where('cn', 'ends_with', 'Doe')->get();  
$results = $ad->search()->where('cn', 'contains', 'John Doe')->get();
```

ასევე რამდენიმე მომხმარებლის ერთად ძიება:

```
$results = $ad->search()  
->where('cn', '=', 'John Doe')  
->orWhere('cn' '=', 'Suzy Doe')  
->get();
```

შეგვიძლია წამოვიღოთ ის მომხმარებლები, რომლებსაც შევსებული აქვთ რამე კონკრეტული ველი:

```
$results = $ad->search()->select(array('cn', 'displayname'))->all();
```

ასევე შეგვიძლია სკრიპტშივე დავალაგოთ მონაცემები და ისე წამოვიღოთ:

```
$results = $ad->search()->where('cn', '=', 'John*')->sortBy('cn', 'desc')->get();
```

შეგვიძლია პირიქით, წამოვიღოთ მომხმარებლები გარდა რომელიმე თანამშრომლისა:

```
$escapedValue = $ad->getLdapConnection()->escape('John Doe');
```

შეგვიძლია გავიგოთ მომხმარებლების ჯამური რაოდენობა:

```
$results->count();
```

ანუ ბრუნდება სტანდარტული მასივის ობიექტი, რომელზეც შესაძლებელია ნებისმიერი ოპერაციის ჩატარება და იტერაცია.

შეგვიძლია გავფილტროთ დეტალურად მომხმარებლები, რომლებსაც არ აქვთ რამე კონკრეტული სახელი:

```
$results = $ad->search()  
    ->where('objectClass', '=', $ad->getUserIdKey())  
->where('cn', '!', 'John')  
->get();
```

ან რამდენიმეს ერთად გამოკლებით:

```
$results = $ad->search()  
    ->where('objectClass', '=', $ad->getUserIdKey())  
    ->orWhere('cn', '!', 'John')  
    ->orWhere('cn', '!', 'Suzy')  
->get();
```

ან რამე ჯგუფში მყოფი ელემენტები:

```
$results = $ad->search()  
    ->where('objectClass', '=', 'computer')  
->get();
```

შეგვიძლია გავფილტროთ ოპერაციული სისტემის მიხედვითაც:

```
$results = $ad->search()  
    ->where('objectClass', '=', 'computer')  
    ->where('operatingSystem', 'starts_with', 'Windows 7')  
->get();
```

შეგვიძლია წამოვიღოთ რომელიმე საქალაქის ელემენტები:

```
$folderName = 'Accounting';  
  
$results = $this->adldap->search()  
->where('OU', '=', $folderName)  
->first();
```

ყველა ფოლდერის წამოღება შეგვიძლია კოდით:

```
$folders = $ad->folder()->all();
```

ფოლდერის შექმნა ხდება შემდეგნაირად:

```
$attributes = [  
    'ou_name' => 'Employees',  
    'container' => [  
        'Users'  
    ]  
];
```

ხოლო წამლისთვის არის სკრიპტი:

```
$distinguishedName = 'OU=Accounting,OU=User
Accounts,DC=corp,DC=acme,DC=com';

$deleted = $ad->folder()->delete($distinguishedName);
```

შეგვიძლია ყველა საქალაქის წაშლაც:

```
$folder = $ad->folder()->find('Accounting');

if(is_array($folder) && array_key_exists('dn', $folder)) {
    $ad->folder()->delete($folder['dn']);
}
```

ჯგუფების დეტალური ინფორმაცია მოგვდის შემდეგი კოდით:

```
// Get the users information
$user = $ad->user()->find('John Doe');

// Get their primary group ID
$primaryGroupId = $user['primarygroupid'];

// Get their object SID
$objectSid = $user['objectsid'];

/*
 * Get the primary groups DN by
 * passing in the users primary group ID
 * and SID
 */
$groupDn = $ad->group()->getPrimaryGroup($primaryGroupId, $objectSid);

echo $groupDn; // Returns 'CN=Domain Users,CN=Users,DC=corp,DC=acme,DC=org'
```

მომხმარებლებისთვის საძიებოდ არის შემდეგნაირად:

```
$users = $ad->user()->all();
```

ან სახელის მიხედვით:

```
$username = 'jdoe';
$user = $ad->user()->find($username);
```

მომხმარებლის ინფორმაციის წამოღება:

```
$username = 'jdoe';
$dn = $ad->user()->dn($username);
```

მომხმარებლის წაშლა:

```
$username = 'jdoe';  
$deleted = $ad->user()->delete($username);
```

თუმცა მომხმარებლის წაშლა არ არის მიზანშეწონილი. უმჯობესია წაშლილთა ჯგუფის შექმნა და ყველა წასაშლელი მომხმარებლის გადატანა იქ და აღრიცხვა როგორც წაშლილების შესაბამისი უფლებებით.

adLDAP ბიბლიოთეკით შეგვიძლია მომხმარებლის პაროლის ცვლილება:

```
try  
{  
    $newPassword = 'newpassword123';  
    $oldPassword = 'oldpassword123';  
  
    $changed = $ad->user()->changePassword('jdoe', $newPassword,  
$oldPassword);  
}  
catch(Adldap\Exceptions\WrongPasswordException $e)  
{  
    return "Uh oh, you've entered the wrong old password!";  
}  
catch(Adldap\Exceptions>PasswordPolicyException $e)  
{  
    return "Looks like your new password doesn't meet our requirements. Try  
again."  
}
```

შეგვიძლია გაგება, თუ როდის გასდის მის პაროლს ვადა:

```
$results = $ad->user()->passwordExpiry('jdoe'); // Returns array|bool  
  
$results['expires']; // Returns true / false if the users password expires  
$results['has_expired']; // Returns true / false if the users password  
**has** expired  
$results['expiry_timestamp']; // Returns the users password expiry date in  
unix time  
$results['expiry_formatted']; // Returns the users password expiry date in  
a formatted string ('YYYY-MM-DD HH:MM:SS')
```

შეგვიძლია გავიგოთ ბოლოს თუ როდის იყო შემოსული:

```
$time = $ad->user()->getLastLogon('jdoe'); // Returns in Unix time  
$date = date('Y-m-d h:i:s', $time);
```

ანუ ActiveDirectory-ს მეშვეობით შესაძლებელია მარტივად აღიწეროს ქსელში მყოფი ნებისმიერი ელემენტი ადმინისტრირების გარეშე, კოდიდან და კერძო რეპორტების წარმოება მასზე. თუმცა მხოლოდ ეს არ არის AD-ს სრული შესაძლებლობები. შუალედური ინტერპრეტატორის მეშვეობით შესაძლებელია მისი ინტეგრირება სხვა სისტემებშიც, როგორცაა თუნდაც Oracle-ს ბაზების ავტორიზაციის სისტემა, რის შემდეგაც ახალი ჩანართები დაემატება ჩვენს სტრუქტურას. თვითონ სკრიპტის სინტაქსი კი იგივე რჩება, რაც იყო Microsoft-ის პროდუქტებთან. ერთადერთი, იცვლება დაკავშირების მისამართის ტექსტი, რომლებიც მაგალითად გამოიყურება შემდეგნაირად:

```
dsacl s "CN=orcl,CN=OracleContext,OU=Example,O=Com" /G "anonymous logon":GR  
  
dsacl s "CN=orcl,CN=OracleContext,OU=Example,O=Com" /G example\scott:GR  
  
dsacl s "CN=orcl,CN=OracleContext,OU=Example,O=Com" /R "anonymous logon"  
  
dsacl s "CN=orcl,CN=OracleContext,OU=Example,O=com" /R example\scott
```

სადაც პირველ შემთხვევაში ვცდილობთ დაკავშირებას როგორც ანონიმური მომხმარებელი, ხოლო სერვისად ვირჩევთ Generic კითხვის რეჟიმს. მეორე მაგალითში Generic მეთოდი ებმევა კონკრეტულ მომხმარებელს. მესამე ხაზზე განხილულია ანონიმური ავტორიზაცია Generic მეთოდის გარეშე, ხოლო მეოთხე მაგალითი გვიჩვენებს კავშირს კონკრეტული მომხმარებლით Generic ბმის გარეშე, რის შემდგომაც მომხმარებლების ჩანართის სიაში ემატება ახალი მომხმარებლები სამართავად და Oracle ბაზებთან დასაკავშირებლად.

ჩემს მიერ აწყობილი აპლიკაციები მუშაობენ PHP Laravel ძრავზე, რომელიც არის MVC (Model, View, Controller) სტრუქტურით, ხოლო ზოგიერთი ფუნქციის დანერგვისთვის საჭირო იყო NodeJS ფუნქციონალის დამატებაც, რომლებიც ერთმანეთთან იდეალურად მუშაობენ.

განვიხილოთ ჩემი რამდენიმე სოფტის ზოგადი ტექნიკური დოკუმენტაციების ფრანგემტები, რომლებიც მთლიანად აწყობილია და

მუშაობენ AD-სთან სინქრონით. ერთ-ერთი დადებითი მხარე, რაც შეგვიძლია გავაკეთოთ PHP AD-თი, არის მომხმარებლების სესიის გაუქმება, ანუ ნებისმიერ კლიენტს შეგვიძლია კოდიდან წავუშალოთ წვდომა რესურსებზე. აგრეთვე ვახდენ ყველა მომხმარებლის გასვლის სიმულაციას დამის 12 საათზე, რადგან შევძლო მათი აქტიურობის შედარების ანალიზის აგება. ამ მიდგომაზე შემოიძლია უამრავი გამოყენების მაგალითის მოყვანა. ერთ-ერთი მათგანი არის მძღოლების დაჯავშნის ფუნქციონალი, რომელიც ხდება AD-დან და მისი კოდის უკანა მხარე დაწერილია Laravel სრული ინტეგრაციებით. შემდეგი საჭირო და მნიშვნელოვანი პარამეტრი, რისი გაკეთებას შეიძლება ადმინისტრატორის გარეშე, არის მომხმარებლის პაროლის დარესეტება.

მომხმარებლების ავტორიზაციის კოდი შედგება შემდეგი პუნქტებისგან:

- სახელის ამოკითხვა;
- ავტორიზაცია;
- სახელის შედარება;
- შეტყობინების გამოტანა;
- გასვლა;
- იძულებითი გასვლა;
- ლოგი.

თუ ავტორიზაცია ვერ მოხერხდა, ან მომხმარებელი ვერ მოიძებნა, გამომაქვს შესაბამისი შეტყობინება, მაგალითად - "ასეთი მომხმარებელი ვერ მოიძებნა". ავტორიზაციისას ვქმნი სესია, ხოლო გასვლისას - ვხურავ მას. ამავდროულად ვახდენ ავტორიზაციების აღრიცხვას, რასაც შემდგომ ვაანალიზებ.

როგორც უკვე ავღნიშნე, PHP LDAP-ს აქვს ფაქტიურად ყველა ის შესაძლებლობა, რაც Microsoft AD-ში შესვლისას. ერთ-ერთი ძალიან გამოსადეგი ფუნქცია არის ახალი მომხმარებლის დამატების შესაძლებლობა. ინტრანეტის ერთ-ერთი ყველაზე მნიშვნელოვანი ნაწილი არის ჩათი. ჩათსაც ვუკეთებ ავტორიზაციას იგივე AD-დან PHP-ის

მეშვეობით. მომხმარებლებს თვითონაც აქვთ ჯგუფური ჩათების შექმნის შესაძლებლობა. ჯგუფი იქმნება კონკრეტული მომხმარებლის სახელზე, რომელიც ხდება ამ ჯგუფის მფლობელი. მხოლოდ მას აქვს ამ ჯგუფის დახურვის საშუალება, ისევე როგორც მისი ძირითადი მმართველობის მოვალეობის შესრულება.

მომხმარებლების ინფორმაციის წამოღებისთვის მაქვს ორი ფუნქცია, რომლებიც უკავშირდებიან ბაზას, რომელშიც უკვე წამოღებულია AD-ს ყველა აქტიური მომხმარებელი, რომელიც სინქრონიზდება დღეში ერთხელ, ვიღებ ყოველი მათგანის სურათს, სახელს, გვარს, თანამდებობას, დეპარტამენტს, ფილიალს, შიდა ნომერს, სამსახურის ტელეფონს, პირად ტელეფონს, სამსახურის ფოსტას და დაბადების თარიღს, ასევე ვაერთიანებ რამდენიმე მნიშვნელოვან ცხრილთან საჭიროებისამებრ, მაგალითად - მიმდინარე სტატუსთან. თითოეულ მომხმარებელს ეძლევა საშუალება, რომ მიიღოს მონაწილეობა გამოკითხვებში, სადაც ფიქსირდება ერთხელ თავისი ნომრით და მეორედ აღარ ეძლევა იგივე გამოკითხვაში ხმის მიცემის საშუალება, რადგან ვიმახსოვრებთ მათ ყველა ქმედებებს.

იდენტიფიცირებისთვის ვიყენებ დამატებით სტატუსს, რომელსაც ხმის მიცემამდე ვამოწმებ და თუ აღმოჩნდა, რომ მომხმარებელს უკვე აქვს მიცემული ხმა, მაშინ გამომაქვს მისთვის შესაბამისი შეტყობინება. მომხმარებლებზე მაქვს მინიჭებული უფლებები. თუ მომხმარებელს აქვს გამოკითხვის შექმნის უფლება, მაშინ მას შეეძლება დაამატოს ახალი გამოკითხვა. ასევე არის შესაძლებლობა გამოკითხვის რედაქტირების, თუ მომხმარებელს აქვს ამის შესაბამისი უფლება. მომხმარებლებს აქვთ როგორც გამოკითხვის ტექსტის, აგრეთვე შეკითხვების რედაქტირება, თუ აქვთ შესაბამისი უფლება. მომხმარებლის შექმნის დილაკზე დაჭერისას იგი ამოწმებს მისი შექმნის შესაძლებლობას და გამოაქვს შესაბამისი სტატუსის შეტყობინება. რომელზე წვდომაც აქვთ მხოლოდ შესაბამისი უფლების მქონე მომხმარებლებს, რადგან იგი შეიცავს სრულ დეტალურ ინფორმაციას მომხმარებლებზე მათი აქტიურობის შესახებ კონკრეტულ პერიოდსა და

დროზე. მომხმარებლების რეგისტრაციასა და რედაქტირებაზე ადევს შეზღუდვები ველებზე, რათა თავი ავირიდო არავალიდური და არასწორი ინფორმაციის შეყვანისგან. ჩათში შეტყობინების გაგზავნისას თავდაპირველად არის სტატუსი "უნახავი", ხოლო ადრესატის მიერ შეტყობინების ნახვისას სტატუსი გარდაიქმნება "ნანახია" მნიშვნელობაზე. ჩათს ვამუშავებ როგორც ცალკე პროექტს - მისთვის იქმნება არხი, რომელზეც ხდება სხვადასხვა ოპერაციები მისამართების მიმოცვლის გზით.

მომხმარებლებთან სამუშაოდ Microsoft Active Directory არ არის ერთადერთი საშუალება. გვხვდება სხვა პოპულარული პროგრამული უზრუნველყოფებიც, რომლებიც მეტნაკლებად ჰგავან მას როგორც ფუნქციონალურად, ასევე ინტერფეისულადაც. მათრიცხვს განეკუთვნება:

- Apache Directory;
- Open LDAP;
- Univention Corporation Server (UCS);
- Lepide Auditor for Active Directory;
- JXplorer;
- FreeIPA;
- Samba;
- GoSa;
- eDirectory;
- Zentyal;
- 389 Directory Server;
- Red Hat Directory Servers;
- OpenSSO;
- SME Server;
- Resara Server;
- Sun Java System Directory Server;
- IBM Tivoli Directory Server;
- Windows NT Directory Services;

- Lotus Domino;
- SolarWinds Permissions Analyzer;
- ManageEngine AdManager Plus;
- Specops Command.

თითოეული მათგანის ფუნქციას განეკუთვნება თქვენი უსაფრთხოება უცხო პირების შემოღწევისგან და წვდომების შეზღუდვა მომხმარებლებისთვის როგორც მათი მართვის, ასევე მენეჯმენტისთვის.

დასკვნები

1. დასაბუთებულია, რომ Microsoft Windows Active Directory არის მძლავრი იარაღი მომხმარებლების სამართავად, მათი უფლებების გასაწერად და ამ ყველაფრის გასაზიარებლად სხვადასხვა სისტემებში - ინტრანეტში, საფოსტო ყუთში, კალენდარში, კომუნიკატორსა და Microsoft-ისა თუ პირად პროგრამებში და აგრეთვე საბრაუზერო აპლიკაციებში.
2. ნაჩვენებია, რომ Microsoft Windows Active Directory სისტემის ეფექტურად გამოყენება შეიძლება საბრაუზერო აპლიკაციებში.
3. ის გვაძლევს მონაცემების შევსებისა და დაყოფის საშუალებას. იმახსოვრებს AD დირექტორიის ჩაშენებულ ველებს, ასევე შეუძლია პერსონალური ველების შექმნა და შევსება.
4. რადგან მისი ინტეგრირების მხარდაჭერა თითქმის ყველა თანამედროვე პროგრამირების ენას აქვს, ამიტომ მისი მეშვეობით შესაძლებელია ერთი ბაზის შევსება და მრავალ აპლიკაციაში გამოყენება.

ABSTRACT

Designing simplified systems for user management

with basic user management schemes

Nowadays our daily life is on the edge of change. Information processing currently is becoming more and more actual problem, what interacts with our daily lives. Modern technologies already provide the development of artificial intelligence. With the help of programming languages, many machines got the ability to think. Now it is possible to have your own robot, which will help us to perform various tasks with maximal efficiency, the number and types of which increases every day with a very high speed.

This dissertation discusses several key methods of designing a simplified system of customer management and describes their advantages and disadvantages. Basic work on the PHP engine that I connected with Microsoft Active Directory is a network of users and objects, and as a result of analyzing the process, you can create a complete project as a web application that collects the active employees with their rules through integration and synchronization. The theme is relevant because modern companies need software that is easy and reliable and will enhance their development and progress both technologically and technically. One of the most widely used examples is the intranet that facilitates the management of the company's division and saves time for their network and system administrators whose part of the function is partially implemented by us.

By integrating with AD I have detailed personal information of all active and inactive users such as their name, email, position, department, branch, mobile and further analyzing all these information and use it in my applications. Consumers can be collected either in groups or in groups, such as rooms or floors, as is the structure of the hierarchy structure of the AD hierarchy.

We can also have several companies in a single database and we use them centrally, which can be used to simplify the work process and to improve. Users' information allows us to discern each other and draw a variety of rights and categories for the use of software accessibility, which can be achieved by creating artificial intelligence. When we need a sales report, which should be used by a branch manager, a loan officer, a sales agent, and an analyst, but we should have different visions - to do that, we would use 4 different reports with a similar structure and loaded over as a programmer and a machine. Integrating with AD allows you to find out which user is currently authorized and creating a report that automatically generates the desired result. The information received from the AD can be distinguished with a code, recording and using it in its own database and further analysis and subsequent analysis.

For the software side I have a PHP script language that has all the supporting engine to communicate with the LDAP text addresses to the user base and to synchronize it with different operations. PHP ADELdap is a library that can fetch, update, edit, delete, modify, generate, and fill out the information from Active Directory Server. It is a very powerful program, but still has drawbacks. With full power and access it processes the software code from the centralized base of users, which is a threat to the company. When running incorrect or random code, it may be possible to remove all user base, their information, and all the attachments. The program can be used for any purpose in this field. With good

management, you will come to the conclusion that it can change the functionality of the company. It is written on a process such as the arrival of employees, departures, logging in programs, clicking on buttons, browsing pages, generating reports, and much more. Once again, its base allows us to use it absolutely everywhere - mail, calendar, communicator, reporting server, computer, bases and in your own applications. One change affects all products that the user uses. No need to update everyone separately.

გამოქვეყნებული ნაშრომების სია

1. სტუ-ს სტუდენტთა 86-ე ღია საერთაშორისო სამეცნიერო კონფერენცია, თეზისების კრებული. თემის დასახელება: ხელოვნური ნეირონული ქსელების ანალიზი, გვ. 248. ავტორები: დ. სირბილაძე, თ. ბახტაძე.
2. საქართველოს ტექნიკური უნივერსიტეტის არჩილ ელიაშვილის მართვის სისტემების ინსტიტუტი, შრომათა კრებული N21, 2017. სტატიის დასახელება: ხელოვნური ნეირონული ქსელების (ინტელექტის) ანალიზი, გვ. 168-172. ავტორები: დ. სირბილაძე, თ. ბახტაძე.
3. საქართველოს ტექნიკური უნივერსიტეტის არჩილ ელიაშვილის სახელობის მართვის სისტემების ინსტიტუტი, შრომათა კრებული N22, 2018. სტატიის დასახელება: მომხმარებლების ამომცნობი სისტემის ინტეგრირება აპლიკაციაში, გვ. 180-182. ავტორები: დ. სირბილაძე, თ. ბახტაძე.
4. შრომები - მართვის ავტომატიზირებული სისტემები N3 (27), 2018. სტატიის დასახელება: მონაცემთა გრაფიკული ვიზუალიზაცია Microsoft Power BI-ს მეშვეობით, გვ. 87-94.
5. შრომები - მართვის ავტომატიზირებული სისტემები N3 (27), 2018. სტატიის დასახელება: Machine learning usage in different fields, გვ. 163-166. ავტორი: დ. სირბილაძე.