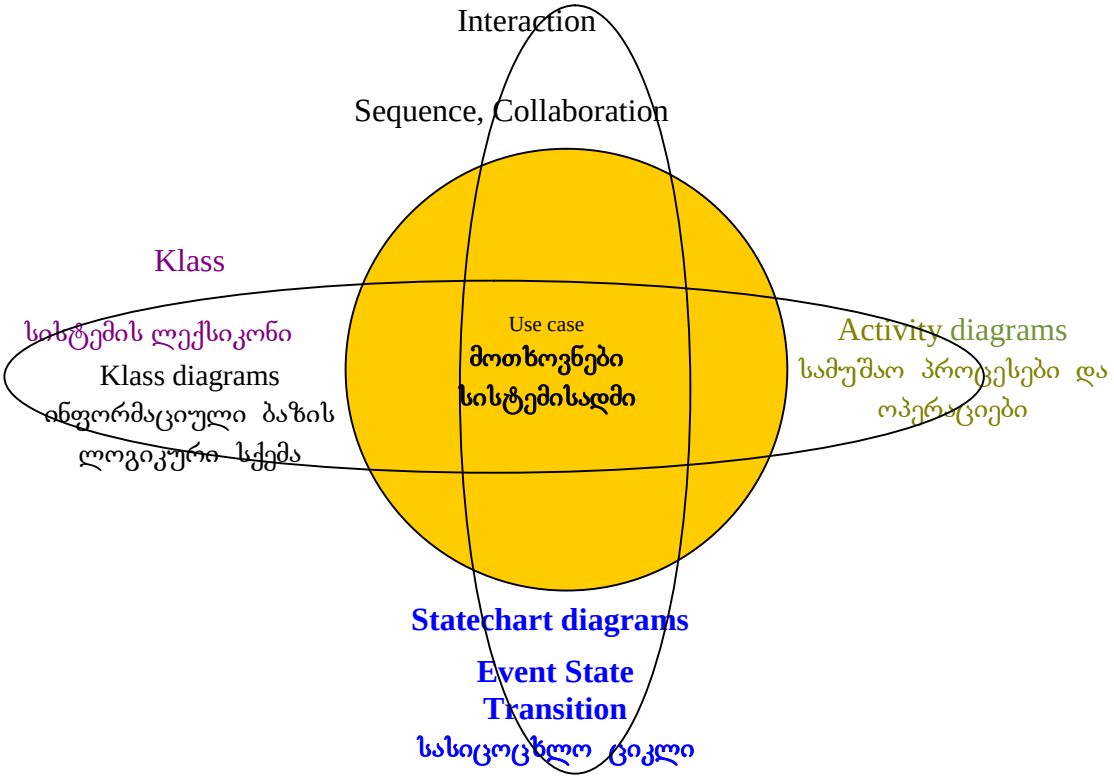
თ. სუხიაშვილი

sistemebis obieqt-orientirebuli daproeqteba

საკურსო პროექტის მეთოდური სახელმძღვანელო



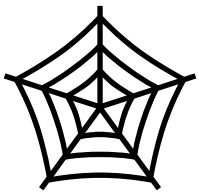
“ტექნიკური უნივერსიტეტი”

saqarTvelos teqnikuri universiteti

თეიმურაზ სუხიაშვილი

sistemebis obieqt-orientirebuli daproeqteba

მეთოდური სახელმძღვანელო საკურსო პროექტის შესასრულებლად
(UML, MsVisio, Rational Rose, Enterprise Architect)



დამტკიცებულია სტუ-ს სამეცნიერო-ტექნიკური საბჭოს მიერ

თბილისი 2016

უაკ 681.3.06.

მოცემულია მართვის კომპიუტერული სისტემის შექმნისათვის სამართავი ობიექტის შესწავლისა და ანალიზის მეთოდიკა ობიექტ-ორიენტირებული მიდგომით, რაციონალური უნიფიცირებული პროცესის(RUP) ფარგლებში სხვადასხვა CASE საშუალებების გამოყენებით(MsVisio, Rational Rose, Enterprise architect).

მოყვანილია სისტემისადმი მოთხოვნების და მისი რეალიზების საშუალებების მოდელირება, რომელიც მოიცავს საპრობლემო სფეროს სტრუქტურული და ქცევითი მოდელირების საკითხებს. განკუთვნილია ინფორმატიკის სპეციალობის სტუდენტების, მაგისტრანტებისა და სპეციალისტებისათვის.

saqarTvelos teqnikuri universiteti

marTvis avtomatizebuli sistemebis kaTedra

sakurso proeqti

დისციპლინაში: “**sistemebis obieqt-orientirebuli daproeqteba**”

ჯგ. №

სტუდენტი:

თემა:

გაცემის თარიღი:

ჩაბარების თარიღი:

ხელმძღვანელი:

proeqtis sawyisi monacemebi

1. საპრობლემო სფერო (კვლევის ობიექტი):

2. მართვის სფერო (ფუნქციური ამოცანა):

პროფესორის ხელმოწერა:

საკურსო პროექტის თემატიკა

საკურსო პროექტის თემატიკა ორიენტირებულია სისტემის დამუშავებისათვის მოცემულ საგნობრივ სფეროში კონკრეტული გამოყენებითი ამოცანების გადასაწყვეტად.

საკურსო პროექტის ძირითადი დავალება - პროგრამული უზრუნველყოფის მოდელების აგებაა ინსტრუმენტალური საშუალება (Ms Visio, Rational Rose, Enterprise Architect) მეშვეობით. მოდელის შექმნის პროცესი შედგება რამოდენიმე ეტაპისაგან.

1. პროექტის ლექსიკონის შექმნა.
2. პრეცედენტთა მოდელის შექმნა.
3. პრეცედენტთა ანალიზი.
4. სისტემის დაპროექტება.

მოდელის შექმნის პროცესი უნდა განხორციელდეს ისე, როგორც ეს აღწერილია მოცემულ პრაქტიკუმში. მოდელის სტრუქტურა უნდა შეესაბამებოდეს სტრუქტურას, რომელიც გათვალისწინებულია Rational Unifed Proces-ის ტექნოლოგიით.

მესამე ეტაპის შესრულების შემდეგ მოდელი უნდა აკმაყოფილებდეს შემდეგ მოთხოვნებს:

- პროექტის ლექსიკონს უნდა ქონდეს ცხრილის სახე და ინახებოდეს ცალკე ფაილად;
- პრეცედენტების დიაგრამებზე ყოველი მოქმედი პირი და პრეცედენტი უნდა აღწერილი იქნას თანმხლები დოკუმენტით. მოქმედი პირის აღწერა მოკლედ უნდა ახასიათებდეს მოცემული პირის როლს. პრეცედენტების აღწერა უნდა მოიცავდეს განმარტებებს, წინაპირობებს, მოვლენათა ნაკადს(ძირითადს და ალტერნატიულს) და შემდგომ პირობებს. აღწერები წარმოადგენენ ან მიერთებულ ტექსტურ ფაილებს, ან ტექსტს, რომელიც

შეტანილია დიაგრამის შესაბამისი ელემენტის სპეციფიკაციაში ველში Documentation;

- ურთიერთქმედების დიაგრამებს, რომლებიც შეესაბამებიან პრეცედენტების მოვლენათა ნაკადებს, უნდა შეიცავდნენ აუცილებელ განმარტებებს.

სისტემის პროექტირებისას საჭიროა:

- შევქმნათ სისტემის კლასების იერარქია;
- განვალაგოთ კლასები პაკეტების მიხედვით;
- დაუკავშიროთ ობიექტები კლასებს, შეტყობინებები ურთიერთქმედების დიაგრამებზე – ოპერაციებს;
- აღვწეროთ მოკლედ ყოველი კლასი(კლასების მოვალეობები), ატრიბუტების აღწერით ცხრილის სახით(დასახელება, აღწერა, ტიპი) და ცხრილით ოპერაციების აღწერით(დასახელება, აღწერა, სიგნატურა);
- მიუთითოდ სტერეოტიპები კლასებისათვის;
- ავაგოთ კლასების დიაგრამა, რომლებზეც ჩანს კლასებს შორის კავშირები;
- ავაგოთ მდგომარეობათა დიაგრამები ცალკეულ კლასთა ობიექტების ქცევის აღწერისათვის;
- დამუშავდეს(თუ საჭიროა) მონაცემთა ბაზის სქემა და გამოვსახოთ იგი დიაგრამაზე „არსი-კავშირი“.

საჭიროა აგრეთვე განლაგების დიაგრამის დამუშავება, რომელზედაც ნაჩვენები იქნება კომპონენტების განთავსება.

საკურსო პროექტის სტრუქტურა

საკურსო პროექტი უნდა შედგებოდეს ოთხი თავისა და დასკვნისაგან. პირველი თავი „ამოცანის დასმა“ უნდა შეიცავდეს დავალების ფორმულირებას.

მეორე თავი “მოთხოვნების ანალიზი“ უნდა შეიცავდეს ლექსიკონს, პრეცედენტების დიაგრამას, მოქმედი პირების და პრეცედენტების აღწერას.

მესამე თავი “სისტემის ანალიზი“ უნდა შეიცავდეს ურთიერთქმედების დიაგრამებს ობიექტებს შორის(მიმდევრობის და კოოპერაციის). რომლებიც შეესაბამებიან პრეცედენტების მოვლენათა ნაკადს. აუცილებლობის შემთხვევაში შესაძლებელია ჩაირთოს მოღვაწეობის დიაგრამები შესაბამისი განმარტებების თანხლებით, რომლებშიც ახსნილი იქნება თუ რომელ მოვლენათა ნაკადს შეესაბამებიან.

მეოთხე თავი „დაპროექტება“ უნდა შეიცავდეს სისტემის კლასების იერარქიას და პაკეტების აღწერას. სისტემის ყოველი კლასისათვის მოყვანილი უნდა იქნას აღწერა, რომელიც მოიცავს კლასის მოვალეობას, ატრიბუტების აღწერას ცხრილის სახით(დასახელება, აღწერა, ტიპი) და ოპერაციების აღწერას ასევე ცხრილით (დასახელება, აღწერა, სიგნატურა). მოყვანილი უნდა იქნას კლასების დიაგრამა, სადაც მითითებული იქნება კლასებს შორის მიმართებები, მდგომარეობის დიაგრამები, ცალკეული კლასების ქცევის აღწერისათვის. გარდა ამისა, უნდა წარმოვადგინოთ განლაგების დიაგრამა, კვანძებზე კომპონენტების განთავსებით. თუ ვარიანტი ითვალისწინებს მონაცემთა ბაზის სქემის აღწერას, იგი წარმოდგენილი უნდა იქნას. ბოლოს, მოყვანილი უნდა იქნას დასკვნა და შეფასებული იქნას სამუშაოს შედეგები.

სისტემაში მოდელების წარმოდგენის თვალსაჩინოებისათვის განვიხილავთ პრაქტიკულ ამოცანას, რომელიც დაკავშირებულია უმაღლეს

სასწავლებლებში სტუდენტების დამატებით ფასიან კურსებზე რეგისტრაციის სისტემის დამუშავებასთან.

1. ამოცანის დასმა - უმაღლეს სასწავლებლებში სტუდენტების დამატებით ფასიან კურსებზე რეგისტრაციის სისტემის დამუშავება

ამჟამად სტუდენტების რეგისტრაციის პროცესი, რომლებსაც სურთ მოისმინონ ესა თუ ის დამატებითი კურსები, და კურსების განაწილება პროფესორებს შორის რთული და ხანგრძლივია.

მას შემდეგ რაც პროფესორები მიიღებენ გადაწყვეტილებას, თუ რომელი კურსების წაკითხვას აპირებენ მომდევნო სემესტრში, დეკანატის წარმომადგენელს შეყავს მიღებული ინფორმაცია მონაცემთა ბაზაში, რომელიც შეიცავს მთელ ინფორმაციას კურსების შესახებ(კურსების კატალოგი) და ამობეჭდავს ანგარიშს, რომელიც შეიცავს ცნობებს პედაგოგიური დატვირთვების განაწილების შესახებ. შესაბამისი კურსების კატალოგი ასევე გამოქვეყნდება და გადაეცემა სტუდენტებს.

ყოველი სტუდენტი ავსებს სპეციალურ ფორმას, აღნიშნავს ამორჩეულ კურსებს და გადასცემს ფორმას დეკანატს. სტუდენტს შეუძლია სემესტრის განმავლობაში დაესწროს ოთხი კურსის მეცადინეობებს. მონაცემები ფორმიდან, აღებული სტუდენტებისაგან, შეიტანება სისტემაში. როგორც კი მთელი ინფორმაცია იქნება შეტანილი, სრულდება სასწავლო გეგმის ფორმირების პროცედურა. უმეტეს შემთხვევაში ამორჩევის პირველი ვარიანტი, წარმოდგენილი სტუდენტის მიერ, ხდება საბოლაო. მაგრამ თუ წარმოიშვება წინააღმდეგობა ან შეუსაბამობა, სტუდენტს იბარებენ დეკანატში, სადაც მისი მოთხოვნები და სურვილები ზუსტდება და განიხილება ხელახლა. შეთანხმების დასრულების შემდეგ სტუდენტებს ეგზავნებათ მეცადინეობების ცხრილის “მყარი” კოპიები. მთელ პროცესზე ჩვეულებრივ მიდის ერთი-ორი კვირა.

მას შემდეგ რაც სტუდენტთა რეგისტრაციის პროცესი დამთავრებულია, დეკანატის წარმომადგენელი აგზავნის ინფორმაციას საანგარიშო სისტემაში, იმისათვის, რომ სტუდენტს შეეძლოს შეიტანოს გადასახადი სემესტრზე. რეგისტრაციის დამთავრების შემდეგ პროფესორებს გადაეცემათ სტუდენტების სიები ყოველი კურსის მიხედვით.

კურსებზე რეგისტრაციის პროცესის დაჩქარებისა და ეფექტურობის გაზრდისათვის დაისვა ამოცანა დამუშავდეს სტუდენტთა რეგისტრაციის კლიენტ-სერვერული სისტემა. სისტემის საშუალებით სტუდენტები უნდა დარეგისტრირდნენ კურსებზე და ნახონ მოსწრების ტაბელი პერსონალური კომპიუტერიდან, დაკავშირებული უნივერსიტეტის ლოკალურ ქსელთან. პროფესორებს უნდა შეეძლოთ ონლაინ სისტემასთან მიმართვა, რათა მიუთითონ კურსები, რომლებსაც ისინი წაიკითხავენ, და შეიტანენ შეფასებებს შესაბამის კურსზე. ყოველი სემესტრის დასაწყისში სტუდენტებს შეუძლიათ მოითხოვონ კურსების კატალოგი, რომელიც შეიცავს კურსების ცხრილს და წარედგინება სწავლებისათვის მოცემულ სემესტრში. ინფორმაცია ყოველი კურსის შესახებ უნდა შეიცავდეს პროფესორის სახელს, კათედრის დასახელებას და მოთხოვნებს მომზადების წინასწარი დონის შესახებ.

სისტემა საშუალებას უნდა აძლევდეს სტუდენტებს აირჩიონ ოთხი კურსი მომდევნო სემესტრისათვის. დამატებით ყოველ სტუდენტს უნდა შეეძლოს მიუთითოს ორი ალტერნატიული კურსი იმ შემთხვევისათვის, თუ რომელიმე არჩეული კურსი უკვე აღმოჩნდება შევსებული ან გამორიცხული. ყოველ კურსზე შესაძლებელია ჩაეწეროს არა უმეტეს 10 და არა ნაკლები 3 სტუდენტისა. ყოველ სემესტრში არის დროის მონაკვეთი, როდესაც სტუდენტებს შეუძლიათ შეცვალონ თავიანთი გეგმები. ამ დროს სტუდენტებს უნდა ქონდეთ სისტემასთან მიმართვის საშუალება, რათა დაამატონ ან ამოიღონ არჩეული კურსები. მას შემდეგ რაც რეგისტრაციის პროცესი რომელიმე სტუდენტის დამთავრდება, რეგისტრაციის სისტემა

აგზავნის ინფორმაციას საანგარიშსწორებო სისტემაში, რათა სტუდენტს შეეძლოს შეიტანოს ანგარიში სემესტრზე. თუ კურსი აღმოჩნდება შევსებული რეგისტრაციის პროცესში, სტუდენტი უნდა ინფორმირებული იყოს ამის შესახებ მანამდე, ვიდრე მისი პირადი სასწავლო გეგმა საბოლაოდ იქნება ფორმირებული.

სემესტრის ბოლოს სტუდენტებს უნდა ქონდეთ სისტემასთან მიმართვის საშუალება თავიანთი მოსწრების ტაბელების ნახვისათვის. რამდენადაც ეს ინფორმაცია კონფიდენციალურია, სისტემა უნდა უზრუნველყოფდეს მის დაცვას არასანქცირებული მიმართვისაგან.

პროფესორებს უნდა ქონდეთ ონლაინ სისტემასთან მიმართვის საშუალება, რათა მიუთითონ კურსები, რომლებსაც ისინი წაიკითხავენ, და გადასინჯონ სტუდენტთა ცხრილი, რომლებიც ჩაეწერნენ მათ კურსებზე. ამას გარდა, პროფესორებს უნდა შეეძლოთ ნახონ შეფასებები კურსზე.

2. პროგრამული უზრუნველყოფისადმი მოთხოვნების სპეციფიკაცია

მოთხოვნები სისტემისადმი დოკუმენტირდება რიგი დოკუმენტებით, რომელთაგან ერთ-ერთი მნიშვნელოვანი დოკუმენტია საპრობლემო სფეროს ლექსიკონი. მისი მეშვეობით დგინდება საერთო ტერმინოლოგია ყველა მოდელისათვის.

2.1. პროექტის ლექსიკონის შედგენა

ლექსიკონი განკუთვნილია საპრობლემო სფეროს ტერმინოლოგიის აღწერისათვის. იგი შესაძლებელია გამოყენებულ იქნას როგორც სისტემის არაფორმალური *მონაცემთა ლექსიკონი.*

მაგალითისათვის, ქვევით მოყვანილია ტერმინები და მათი მნიშვნელობა.

ტერმინი	მნიშვნელობა
კურსი	სასწავლო კურსი, რომელიც შეთავაზებულია უნივერსიტეტის მიერ
შეთავაზებული კურსი	კონკრეტულ სემესტრში მოცემული კურსის წაკითხვის შეთავაზება. მოიცავს კვირის კონკრეტულ დღეებსა და დროს.
კურსების კატალოგი	ყველა კურსების მთლიანი კატალოგი, რომელსაც სთავაზობს უნივერსიტეტი
პროფესორი	უნივერსიტეტის პედაგოგი
სტუდენტი	პიროვნება, რომელიც გადის სწავლებას უნივერსიტეტში
კურსის სია	ყველა სტუდენტები, რომლებიც ჩაეწერნენ შეთავაზებულ კურსზე
სასწავლო გრაფიკი	კურსები, რომლებიც აირჩია სტუდენტმა მიმდინარე სემესტრში

2.2. პრეცედენტების მოდელის შექმნა ბიზნეს – პროცესებისათვის

ბიზნეს-პროცესების მოდელირება ითვალისწინებს პრეცედენტების მოდელის აგებას – ეს მოდელია, რომელიც აღწერს ორგანიზაციის ბიზნეს პროცესებს როლებისა და მათი მოთხოვნების ტერმინებში.

მოქმედი პირების ცხრილი დგება შემდეგ კითხვებზე პასუხის გაცემის შედეგათ:

- ვინ იღებს სარგებელს ორგანიზაციის არსებობიდან.
- ვინ ეხმარება ორგანიზაციას განახორციელოს თავისი მოღვაწეობა.
- ვის გადასცემს ორგანიზაცია ინფორმაციას და ვისგან ღებულობს.

მოქმედი პირების სწორი გამოვლენისათვის საჭიროა პირველ რიგში განისაზღვროს განსახილველი ორგანიზაცია ან მოღვაწეობის სფერო. მოცემულ შემთხვევაში ასეთი ორგანიზაციის როლს ასრულებს დეკანატი, რომელიც პაცუხიმგებელია სტუდენტების კურსებზე რეგისტრაციაზე, შესაბამისად, ბიზნეს-პროცესების მოქმედი პირები არიან:

სტუდენტი – ეწერება კურსებზე;

პროფესორი – ირჩევს კურსებს სასწავლებლად.

პრეცედენტები ბიზნეს-პროცესების თვალთახედვით (Busines Use Case) განისაზღვრება როგორც მიმდევრობითი მოქმედებების თანმიმდევრობა რომელიღაც ბიზნეს-პროცესის ფარგლებში, რომლებსაც მოაქვთ მნიშვნელოვანი შედეგი კონკრეტული მოქმედი პირისათვის.

მოცემული მეთოდიკით ყურადღება მახვილდება ელემენტარულ ბიზნეს-პროცესებზე. ელემენტარული ბიზნეს-პროცესი შესაძლებელია განვსაზღვროთ როგორც ამოცანა, რომელიც სრულდება ერთი ადამიანით ერთ ადგილზე ერთდაიმავე დროს გარკვეული მოვლენის პასუხად, რომელსაც მოაქვს კონკრეტული შედეგი და გადაყავს მონაცემები გარკვეულ მდგრად მდგომარეობაში. ასეთი ამოცანის გადაწყვეტა მოიცავს 5 დან 10 ბიჯამდე და შესაძლებელია დაიკავოს რამოდენიმე წუთიდან

რამოდენიმე დღემდე, მაგრამ განიხილება როგორც ერთი სეანსი მოქმედი პირისა შემსრულებლებთან.

ყოველი Busines Use Case ასახავს მიზანს ან მოთხოვნას გარკვეული მოქმედი პირის. გამომდინარე მოქმედი პირების (სტუდენტები და პროფესორები) მოთხოვნებიდან, შესაძლებელია გამოვყოთ შემდეგი პრეცედენტები: დავრეგისტრირდეთ კურსებზე და ამოვირჩიოთ კურსები სწავლებისათვის.

Busines Use Case-ს აღწერა წარმოადგენს სპეციფიკაციას, რომელიც შედგება შემდეგი პუნქტებისაგან:

- დასახელება;
- მოკლე დახასიათება;
- მიზნები და შედეგები (მოქმედი პირის თვალთახედვით);
- სცენარების აღწერა (ძირითადის და ალტერნატიულის);
- სპეციალური მოთხოვნები (შეზღუდვები დროში ან სხვა რესურსებში);
- გაფართოება (კერძო შემთხვევები);
- კავშირები სხვა Busines Use Case –თან;
- მოღვაწეობის დიაგრამები (სცენარების თვალსაჩინო აღწერისათვის – აუცილებლობის შემთხვევაში).

მაგალითისათვის მოვიყვანოთ გამოყენებითი შემთხვევის “კურსებზე დარეგისტრირება” სპეციფიკაცია.

დასახელება:

კურსებზე დარეგისტრირება.

მოკლე აღწერა:

მოცემული გამოყენებითი შემთხვევა საშუალებას აძლევს სტუდენტს დარეგისტრირდეს შემოთავაზებულ კურსებზე მიმდინარე სემესტრში.

ძირითადი სცენარი:

1. სტუდენტი მიდის დეკანატის მოსამსახურესთან და გადასცემს მას შევსებულ ფორმას კურსებზე რეგისტრაციისათვის.
2. დეკანატის მოსამსახურე ადასტურებს ფორმის შევსების სისწორეს.
3. დეკანატის მოსამსახურე ადასტურებს, რომ სტუდენტმა შეასრულა წინასწარი მოთხოვნები ყოველი არჩეული კურსისათვის (გარკვეული კურსების გავლა), ასევე თავისუფალი ადგილების არსებობას.
4. დეკანატის მოსამსახურეს შეყავს სტუდენტი, მის მიერ არჩეულ, ყოველი კურსის სიაში.
5. დეკანატის მოსამსახურე ავსებს სტუდენტის გრაფიკს კურსებზე მიმდინარე სემესტრში და გადასცემს მას სტუდენტს.

ალტერნატიული სცენარი:

2ა. არასწორად არის შევსებული რეგისტრაციის ფორმა.

დეკანატის მოსამსახურე უბრუნებს სტუდენტს ფორმას შეცდომების გასწორებისათვის.

3ა. არ არის შესრულებული წინასწარი მოთხოვნები ან კურსი შევსებულია.

თუ დეკანატის მოსამსახურე აღმოაჩენს, რომ სტუდენტს არ შეუსრულებია აუცილებელი წინასწარი მოთხოვნები ან მის მიერ არჩეული კურსი შევსებულია (უკვე ჩაწერილია 10 სტუდენტი), მაშინ იგი სთავაზობს სტუდენტს შეიცვალოს თავისი არჩევანი ან წაიღოს ფორმა და დაუბრუნდეს მას მოგვიანებით.

2.3. დამატებითი სპეციფიკაციების აღწერა

დამატებითი სპეციფიკაციების დანიშნულებაა განისაზღვროს მოთხოვნები სისტემისადმი, რომლებსაც არ მოიცავს პრეცედენტების მოდელი. ერთად ისინი ქმნიან სისტემისადმი მოთხოვნების სრულ ნაკრებს.

დამატებითი სპეციფიკაციები განსაზღვრავენ სისტემისადმი არაფუნქციონალურ მოთხოვნებს ისეთის, როგორიც არის გამოყენების

მოხერხებულობა, საიმედობა, წარმადობა, ასევე რიგი ფუნქციონალური მოთხოვნებისა, რომლებიც საერთოა რამოდენიმე პრეცედენტებისათვის: უსაფრთხოება, საპროექტო შეზღუდვები.

ფუნქციონალური შესაძლებლობები

- სისტემა უნდა უზრუნველყოფდეს მუშაობის მრავალმომხმარებლის რეჟიმს.
- გამოყენების მოხერხებულობა.
- მომხმარებლის ინტერფეისი უნდა იყოს Windows-შეთავსებადი.
- სისტემის მომხმარებლის ინტერფეისი უნდა იყოს მარტივი და არ მოითხოვდეს მომხმარებლისაგან, რომელსაც აქვს კომპიუტერული განათლება, დამატებით შესწავლას.
- სისტემის ყოველი ფუნქცია უნდა უზრუნველყოფილი იყოს ჩართული ონლაინ დახმარებით, რომელიც უნდა შეიცავდეს ინსტრუქციას სისტემასთან მუშაობისათვის.

საიმედობა

- სისტემა უნდა იყოს სამუშაო მდგომარეობაში 24 სთ კვირაში 7 დღე, უქმად დგომის დრო არ უნდა აღემატებოდეს 10. საიმედო მუშაობის საშუალო დრო არ უნდა აღემატებოდეს 300 სთ-ს.

წარმადობა

- სისტემა უნდა აკავებდეს დაახლოებით 2000 მომხმარებელს, რომლებიც ერთდროულად იმუშავებენ მონაცემთა ცენტრალურ ბაზასთან, და დაახლოებით 500 მომხმარებელს, რომლებიც ერთდროულად იმუშავებენ ლოკალურ სერვერებთან.
- სისტემა უნდა უზრუნველყოფდეს კურსების კატალოგის მონაცემთა ბაზასთან მიმართავას დაყოვნების დროით არა უმეტეს 10 წმ.
- სისტემას უნდა შეეძლოს ყველა ტრანზაქციების 80 –ის დასრულება 2 წთ-ის განმავლობაში.

უშიშროება

- სისტემა არ უნდა აძლევდეს საშუალებას სტუდენტებს შეცვალონ ნებისმიერი სასწავლო გრაფიკები, საკუთარის გარდა, ასევე პროფესორებმა მოახდინონ კონკრეტული კურსების მოდიფიცირება, რომლებიც სხვა პროფესორების მიერ არის არჩეული.
- მხოლოდ პროფესორებს აქვთ უფლება დაუსვან შეფასებები სტუდენტებს.
- მხოლოდ რეგისტრატორს შეუძლია შეცვალოს ნებისმიერი ინფორმაცია სტუდენტების შესახებ.

2.4. პრეცედენტების საწყისი ვერსიის მოდელის შექმნა

სისტემისადმი ფუნქციონალური მოთხოვნები მოდელირდება და დოკუმენტირდება პრეცედენტების (Use Case) მეშვეობით, რომლებიც ითვალისწინებენ შემდეგს:

- პრეცედენტი აფიქსირებს შეთანხმებას პროექტის მონაწილეებისა სისტემის ქცევასთან;
- პრეცედენტი აღწერს სისტემის ქცევას სხვადასხვა პირობებისას, როდესაც სისტემა პასუხობს ერთერთი მონაწილის დაკვეთას, რომელიც იწოდება ძირითად მოქმედ პირად.
- ძირითადი მოქმედი პირი ახდენს სისტემასთან ურთიერთქმედების ინიცირებას, იმისათვის რომ მიღწეულ იქნას გარკვეული მიზანი. სისტემა პასუხობს, იცავს რა ყველა მონაწილის ინტერესებს.

პრეცედენტების აღწერისას არსებობს სიზუსტის ოთხი დონე:

- მოქმედი პირები და მიზნები (დგინდება ყველა მოქმედი პირები და მათი მიზნები, რომელსაც უზრუნველყოფს სისტემა);

- პრეცედენტის მოკლე დახასიათება (ერთ სტრიქონში) ან მოვლენათა ძირითადი ნაკადი (მოსალოდნელი შეცდომების ანალიზის გარეშე);
- უარყოფის (შეფერხების) პირობები (მოვლენათა ძირითად ნაკადში მოსალადნელი შეცდომების აღძვრის ადგილების ანალიზი);
- შეფერხებების დამუშავება (მოვლენათა ალტერნატიული ნაკადების აღწერა).

მოთხოვნების სპეციფიკაცია Rational Unified Process – ის ტექნოლოგიით არ ითვალისწინებს ორგანიზაციის ბიზნეს-პროცესების აუცილებელ მოდელირებას, რომლისთვისაც იქმნება პროგრამული უზრუნველყოფა, მაგრამ ბიზნეს მოდელის არსებობა საგრძნობლად ამარტივებს პრეცედენტის სისტემური მოდელის აგებას. ბიზნეს – მოდელიდან პრეცედენტის მოდელის საწყის ვერსიაზე გადასვლისას სრულდება შემდეგი წესები:

- ყოველი შემსრულებლისათვის ბიზნეს – ანალიზის მოდელში, რომელიც პერსპექტივაში გახდება ახალი სისტემის მომხმარებელი, პრეცედენტის მოდელში იქმნება მოქმედი პირი ასეთივე დასახელებით. მოქმედი პირების შემადგენლობაში ჩაირთვება ასევე გარე სისტემები, რომლებიც ბიზნეს – პროცესებში ინფორმაციის წყაროს პასიურ როლს თამაშობენ;
- პრეცედენტები მოცემული მოქმედი პირისათვის იქმნებიან შესაბამისი შემსრულებლის მოთხოვნათა ანალიზის საფუძველზე (უმარტივეს შემთხვევაში შემსრულებლის ყოველი ოპერაციისათვის იქმნება პრეცედენტი, რომელიც მოახდენს მოცემული ოპერაციის რეალიზებას სისტემაში).

მოდელის ასეთი საწყისი ვერსია აღწერს სისტემის მინიმალურ ვარიანტს, რომლის მომხმარებლები არიან მხოლოდ ბიზნეს – პროცესების შემსრულებლები. თუ მომავალში სისტემის განვითარებისას მისი უშუალო

მომხმარებლები იქნებიან ბიზნეს – პროცესების მოქმედი პირები, პრეცედენტების მოდელი დაიწყებს მოდიფიცირებას.

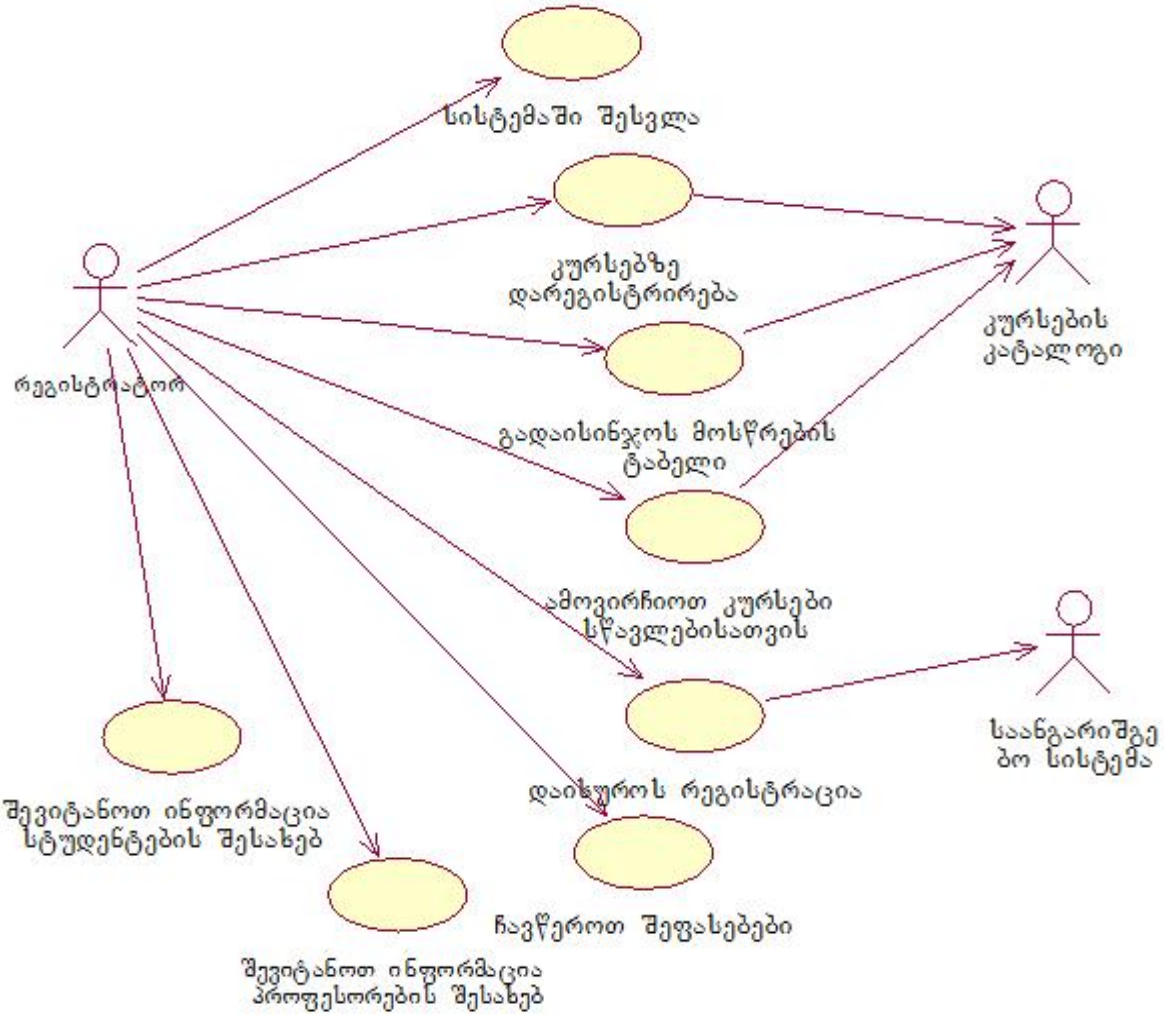
მოყვანილი წესების გამოყენებას რეგისტრაციის სისტემისათვის მივყავართ ახალი მოქმედი პირების გამოჩენასთან სისტემის საწყისი ვერსიისათვის:

- **რეგისტრატორი** – ახდენს სასწავლო გეგმის და კურსების კატალოგის ფორმირებას, წერს სტუდენტებს კურსებზე, მიყავს ყველა მონაცემები კურსების, პროფესორების, მოსწრებისა და სტუდენტების შესახებ.
- **საანგარიშსწორებო სისტემა** – ღებულობს ინფორმაციას მოცემული სისტემიდან კურსებზე დასწრების ანგარიშსწორების შესახებ;
- **კურსების კატალოგი** – მონაცემთა ბაზა, რომელიც შეიცავს ინფორმაციას კურსების შესახებ.

მოქმედი პირების მოთხოვნებიდან გამომდინარე, გამოიყოფა შემდეგი პრეცედენტები:

- სისტემაში შესვლა;
- კურსებზე დარეგისტრირება;
- მოსწრების ტაბელის ნახვა;
- ავირჩიოთ კურსები სწავლისათვის;
- შეფასებების ნახვა;
- შევიტანოთ ინფორმაცია პროფესორების შესახებ;
- შევიტანოთ ინფორმაცია სტუდენტების შესახებ;
- დავხუროთ რეგისტრაცია.

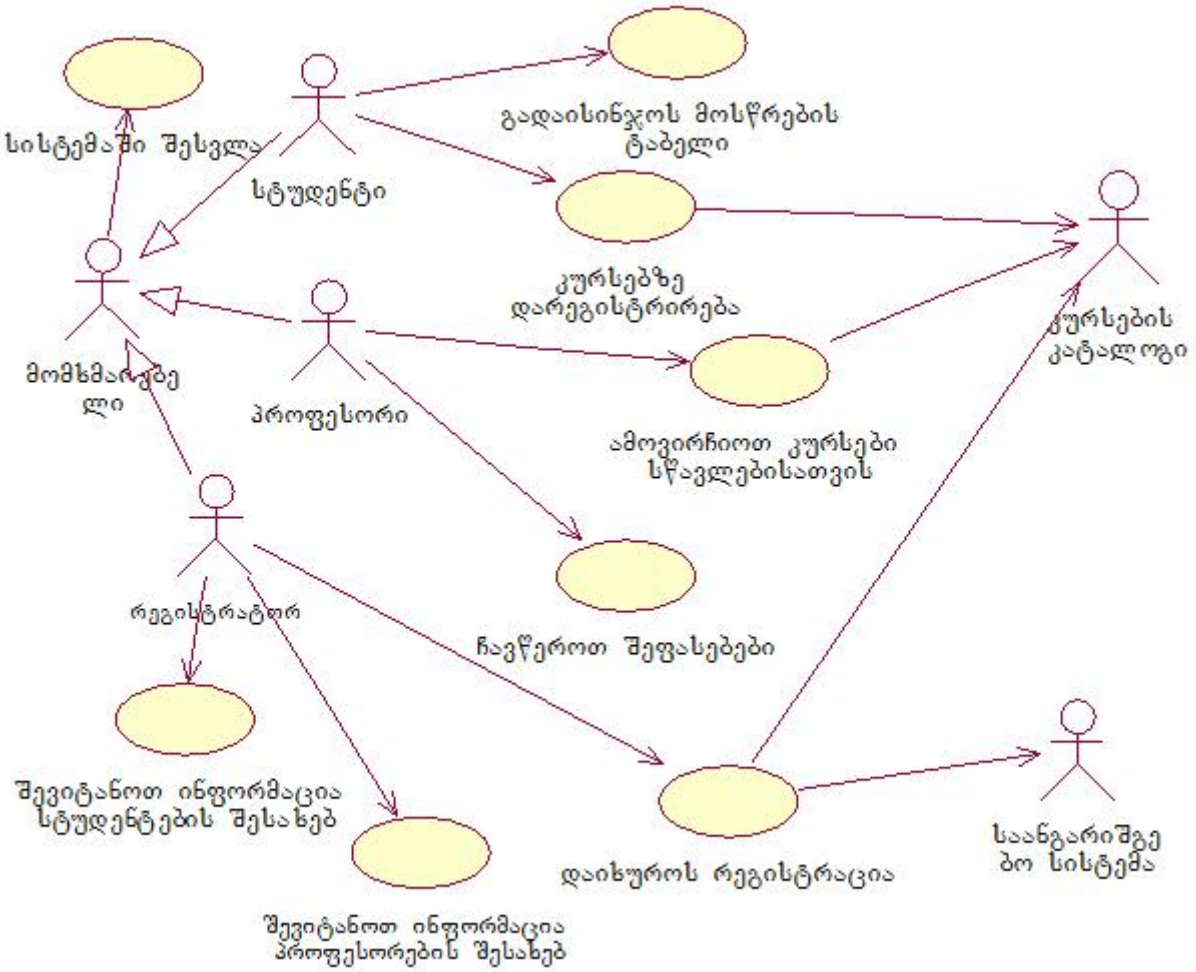
პრეცედენტების დიაგრამის საწყისი ვერსია მოყვანილია ნახ.2.1. – ზე.



ნახ.2.1.

2.5. პრეცედენტების მოდელის მოდიფიკაცია

ამოცანის დასმის თანახმად სისტემის მომხმარებელთა შემადგენლობაში უნდა იყვნენ სტუდენტები და პროფესორები. რამდენადაც სისტემაში შესვლა რეგისტრატორისათვის, სტუდენტებისა და პროფესორებისათვის ერთიდაიგივეა, მათი ქცევა შესაძლებელია განვაზოგადოთ და შემოვიტანოთ ახალი მოქმედი პირი “მომხმარებელი” (სუპერ ტიპი) საერთო პრეცედენტით “სისტემაში შესვლა”, რომლის ქვეტიპებია რეგისტრატორი, სტუდენტი და პროფესორი. პრეცედენტთა დიაგრამის მოდიფიცირებული ვერსია მოყვანილია ნახ.2.2.-ზე.



ნახ.2.2.

მოქმედი პირები:

- სტუდენტი – ეწერება კურსებზე და ათვალიერებს მოსწრების ტაბელს.
- პროფესორი - ირჩევს კურსებს საწავლებისათვის და სვამს შეფასებებს.
- რეგისტრატორი – ახდენს სასწავლო გეგმის და კურსების კატალოგის ფორმირებას, მიყავს ყველა მონაცემები კურსებზე, პროფესორებისა და სტუდენტების შესახებ.
- საანგარიშსწორებო სისტემა – ღებულობს მოცემული სისტემისაგან ინფორმაციას კურსების გადახდის შესახებ.

- კურსების კატალოგი – მონაცემთა ბაზა, რომელიც შეიცავს ინფორმაციას კურსების შესახებ.

გამოყენებითი შემთხვევა "სისტემაში შესვლა"

მოკლე აღწერა:

პრეცედენტების მოცემული ვარიანტი აღწერს მომხმარებლის შესვლას კურსების რეგისტრაციის სისტემაში.

მოვლენათა ძირითადი ნაკადი:

პრეცედენტის მოცემული ვარიანტი იწყებს შესრულებას, როდესაც მომხმარებელს სურს კურსების რეგისტრაციის სისტემაში შესვლა.

1. სისტემა ჩაეკითხება მომხმარებლის სახელს და პაროლს.
2. მომხმარებელს შეყავს სახელი და პაროლი.
3. სისტემა ადასტურებს სახელს და პაროლს, რომლის შემდეგ იხსნება სისტემასთან მიმართვის შესაძლებლობა.

ალტერნატიული ნაკადები:

არასწორი დასახელება/პაროლი:

თუ ძირითადი ნაკადის შესრულებისას აღმოჩნდება, რომ მომხმარებელმა შეიტანა არასწორად სახელი/პაროლი, სისტემას გამოყავს შეტყობინება შეცდომის შესახებ. მომხმარებელს შეუძლია დაბრუნდეს ძირითადი ნაკადის დასაწყისთან ან უარი თქვას სისტემაში შესვლაზე, ამასთან გამოყენებითი შემთხვევის შესრულება მთავრდება.

გამოყენებითი შემთხვევა "კურსებზე დარეგისტრირება"

მოკლე აღწერა:

გამოყენების მოცემული ვარიანტი საშუალებას აძლევს სტუდენტს დარეგისტრირდეს შემოთავაზებულ კურსებზე მოცემულ სემესტრში. სტუდენტს შეუძლია შეცვალოს თავისი არჩევანი (განაახლოს ან გამორიცხოს კურსები), თუ ცვლილება სრულდება დადგენილ დროს

სემესტრის დასაწყისში. კურსების კატალოგის სისტემა წარადგენს მიმდინარე სემესტრის ყველა შეთავაზებული კურსების ცხრილს.

მოვლენათა ძირითადი ნაკადი:

გამოყენების მოცემული ვარიანტი იწყებს შესრულებას, როდესაც სტუდენტს სურს დარეგისტრირდეს კონკრეტულ კურსებზე ან შეცვალოს თავისი კურსების გრაფიკი.

1. სისტემა გამოკითხავს საჭირო მოქმედებას (შექმნას გრაფიკი, განაახლოს გრაფიკი, მოსპოს გრაფიკი).
2. როდესაც სტუდენტი უთითებს მოქმედებას, სრულდება ერთ ერთი დაქვემდებარებული ნაკადი (შეიქმნას, მოისპოს, ან მიიღოს გრაფიკი).

შეიქმნას გრაფიკი:

1. სისტემა ასრულებს შესათავაზებელი კურსების ძებნას კურსების კატალოგში და გამოყავს მათი ცხრილი.
2. სისტემას გამოყავს ცარიელი გრაფიკი შევსებისათვის.
3. სტუდენტი არჩევს ცხრილიდან ოთხ ძირითად და ორ ალტერნატიულ კურსს გრაფიკში ჩასართველად.

ყოველი არჩეული კურსისათვის სრულდება დაქვემდებარებული ნაკადი “დაემატოს კურსი გრაფიკში”.

სისტემა ინახავს სტუდენტის გრაფიკს.

გრაფიკის განახლება:

1. სისტემას გამოყავს სტუდენტის მიმდინარე გრაფიკი.
2. სისტემა ასრულებს კურსების კატალოგში დასაშვები შესათავაზებელი კურსების ძებნას და გამოყავს მათი ცხრილი.
3. სტუდენტს შეუძლია განაახლოს თავისი კურსების არჩევანი, მოსპოს ან დაამატოს შეთავაზებული კურსები.
4. ყოველი არჩეული კურსისათვის სრულდება დაქვემდებარებული ნაკადი “დაემატოს კურსი გრაფიკში”.
5. სისტემა ინახავს სტუდენტის გრაფიკს.

გრაფიკის მოსპობა:

1.სისტემას გამოყავს სტუდენტის მიმდინარე გრაფიკი.

2. სისტემა ეკითხება სტუდენტს დაადასტუროს გრაფიკის მოსპობა.
3. სტუდენტი ადასტურებს მოსპობას.
4. სისტემა სპობს გრაფიკს. თუ გრაფიკი შეიცავს შეთავაზებულ კურსებს, რომლებზედაც ჩაეწერა სტუდენტი, ის უნდა პოისპოს ამ კურსების ცხრილიდან.

დაუმატოთ კურსი გრაფიკს:

ყოველი არჩეული კურსისათვის სისტემა ამოწმებს სტუდენტის მიერ წინასწარი მოთხოვნების შესრულების ფაქტს (გარკვეული კურსების გავლა) და შეთავაზებულ კურსებზე მიღების არსებობას. შემდეგ სისტემა ამატებს სტუდენტს არჩეული კურსის ცხრილში. კურსი აღინიშნება გრაფიკში როგორც "დარეგისტრირებული".

ალტერნატიული ნაკადები:

შევინახოთ გრაფიკი:

სტუდენტს შეუძლია შეინახოს გრაფიკი ნებისმიერ მომენტში, ისე რომ არ დააფიქსიროს ამორჩეული კურსები. ამ შემთხვევაში გრაფიკი ინახება სისტემაში, მაგრამ სისტემა არ ამატებს სტუდენტს ამორჩეული კურსების ცხრილში. კურსები აღინიშნებიან გრაფიკში "ამორჩეულები".

არ შესრულებულა წინასწარი მოთხოვნები ან კურსი შევსებულია:

თუ დაქვემდებარებული ნაკადის "დაუმატოთ კურსი გრაფიკში" შესრულებისას სისტემა აღმოაჩენს, რომ სტუდენტს არ შეუსრულებია აუცილებელი წინასწარი მოთხოვნები ან ამორჩეული მის მიერ კურსი შევსებულია, მაშინ გამოიცემა შეტყობინება შეცდომის შესახებ. სტუდენტს შეუძლია ან ამოირჩიოს სხვა კურსი და განაგრძოს პრეცედენტის შესრულება, ან უარყოს ოპერაცია, რომლის შემდეგაც ძირითადი ნაკადი დაიწყება ახლიდან.

გრაფიკი არ იქნა ნანახი:

თუ დაქვემდებარებული ნაკადის “გრაფიკის განახლება” ან “მოვსპოთ გრაფიკი” შესრულებისას სისტემას არ შეუძლია იპოვოს სტუდენტის გრაფიკი, მაშინ გამოიცემა შეტყობინება შეცდომის შესახებ. მას შემდეგ რაც სტუდენტი დაადასტურებს ამ შეტყობინებას, ძირითადი ნაკადი დაიწყება თავიდან.

კურსების კატალოგის სისტემა მიუწვდომელია:

თუ აღმოჩნდება, რომ კურსების კატალოგის სისტემასთან კავშირის დამყარება შეუძლებელია, მაშინ გამოიცემა შეტყობინება შეცდომის შესახებ. მას შემდეგ რაც სტუდენტი დაადასტურებს ამ შეტყობინებას, გამოყენებითი შემთხვევა დამთავრდება.

კურსებზე რეგისტრაცია დამთავრდა:

თუ გამოყენებითი შემთხვევის შესრულების დასაწყისში აღმოჩნდება, რომ რეგისტრაცია მიმდინარე სემესტრზე დამთავრდა, გამოიცემა შეტყობინება და გამოყენებითი შემთხვევა დამთავრდება.

წინაპირობები:

მოცემული გამოყენებითი შემთხვევის დაწყების წინ სტუდენტი უნდა შევიდეს სისტემაში.

გამოყენებითი შემთხვევა “დავხუროთ რეგისტრაცია”

მოკლე აღწერა:

მოცემული გამოყენებითი შემთხვევა საშუალებას აძლევს რეგისტრატორს დახუროს რეგისტრაციის პროცესი. შეთავაზებული კურსები, რომლებზედაც არ ჩაწერილანსტუდენტების საკმარისი რაოდენობა (სამზე ნაკლები), გამოირიცხება. ანგარიშსწორების სისტემაში გადაიცემა ინფორმაცია ყოველ სტუდენტზე თითოეული შეთავაზებული კურსის მიხედვით, იმისათვის, რომ სტუდენტებს შეეძლოს შეიტანონ გადასახადი კურსებზე.

მოვლენათა ძირითადი ნაკადი:

მოცემული გამოყენებითი შემთხვევა იწყებს შესრულებას, როცა რეგისტრატორი მოითხოვს რეგისტრაციის შეწყვეტას.

სისტემა დაადასტურებს რეგისტრაციის პროცესის დასრულებას.

ყოველი შეთავაზებული კურსისათვის სისტემა ამოწმებს, მიყავს იგი რომელიმე პროფესორს თუ არა და ჩაეწერა თუ არა მასზე არა ნაკლებ სამი სტუდენტისა. თუ ეს პირობები სრულდება, სისტემა საბოლაოდ აფიქსირებს კურსს ყოველ გრაფიკში, რომელიც მოიცავს მოცემულ კურსს. სისტემა ხურავს ყველა კურსებს, ანგარიშობს გადასახადს სწავლაზე ყოველი სტუდენტისათვის მიმდინარე სემესტრში და აგზავნის ინფორმაციას სააგარიშსწორებო სისტემაში. მოცემული სისტემა უგზავნის სტუდენტებს ანგარიშს მათ საბოლაო გრაფიკის კოპიოსთან ერთად.

ალტერნატიული ნაკადები:

რეგისტრაცია არ დასრულებულა:

თუ რეგისტრაციის პროცესის დასრულების შემოწმებისას აღმოჩნდება, რომ რეგისტრაცია კიდევ სრულდება, გამოიცემა შეტყობინება და გამოყენებითი შემთხვევა სრულდება.

კურსზე ჩაეწერა სამ სტუდენტზე ნაკლები:

თუ ძირითადი ნაკადის შესრულების დროს აღმოჩნდება, რომ რომელიმე კურსზე ჩაეწერა სამ სტუდენტზე ნაკლები, მაშინ ეს კურსი უქმდება და სრულდება დაქვემდებარებული ნაკადი “კურსის გაუქმება”.

კურსი არავის არ მიყავს:

თუ ძირითადი ნაკადის შესრულების დროს აღმოჩნდება, რომ რომელიღაც კურსი არც ერთ პროფესორს არ მიყავს, მაშინ ეს კურსი უქმდება და სრულდება დაქვემდებარებული ნაკადი “კურსის გაუქმება”.

კურსის გაუქმება:

სისტემა აუქმებს შეთავაზებულ კურსს. ყოველი სტუდენტისათვის, რომელიც ჩაეწერა გაუქმებულ კურსზე, სისტემა ახდენს მისი გრაფიკის მოდიფიცირებას. პირველი მოპოვებული ალტერნატიული კურსი ჩაისმება

გაუქმებული კურსის მაგივრათ. თუ ალტერნატიული კურსები არ არის, ჩასმა არ ხდება და მართვა გადაეცემა მოვლენათა ძირითად ნაკადში შემდეგი შეთავაზებული კურსის დამუშავებისათვის.

მიმდინარე სემესტრის ყველა გრაფიკების დამუშავების შემდეგ სისტემა ელექტრონული ფოსტით ატყობინებს სტუდენტებს მათ გრაფიკებში ცვლილებების შესახებ.

საანგარიშსწორებო სისტემა მიუწდომელია:

თუ საანგარიშსწორებო სისტემასთან კავშირის დამყარება შეუძლებელია, სისტემა ცდილობს ხელახლა დაუკავშირდეს მას გარკვეული დადგენილი დროის შემდეგ. დაკავშირების ცდები განმეორდებიან მანამდე, სანამ კავშირი არ დამყარდება.

წინაპირობები:

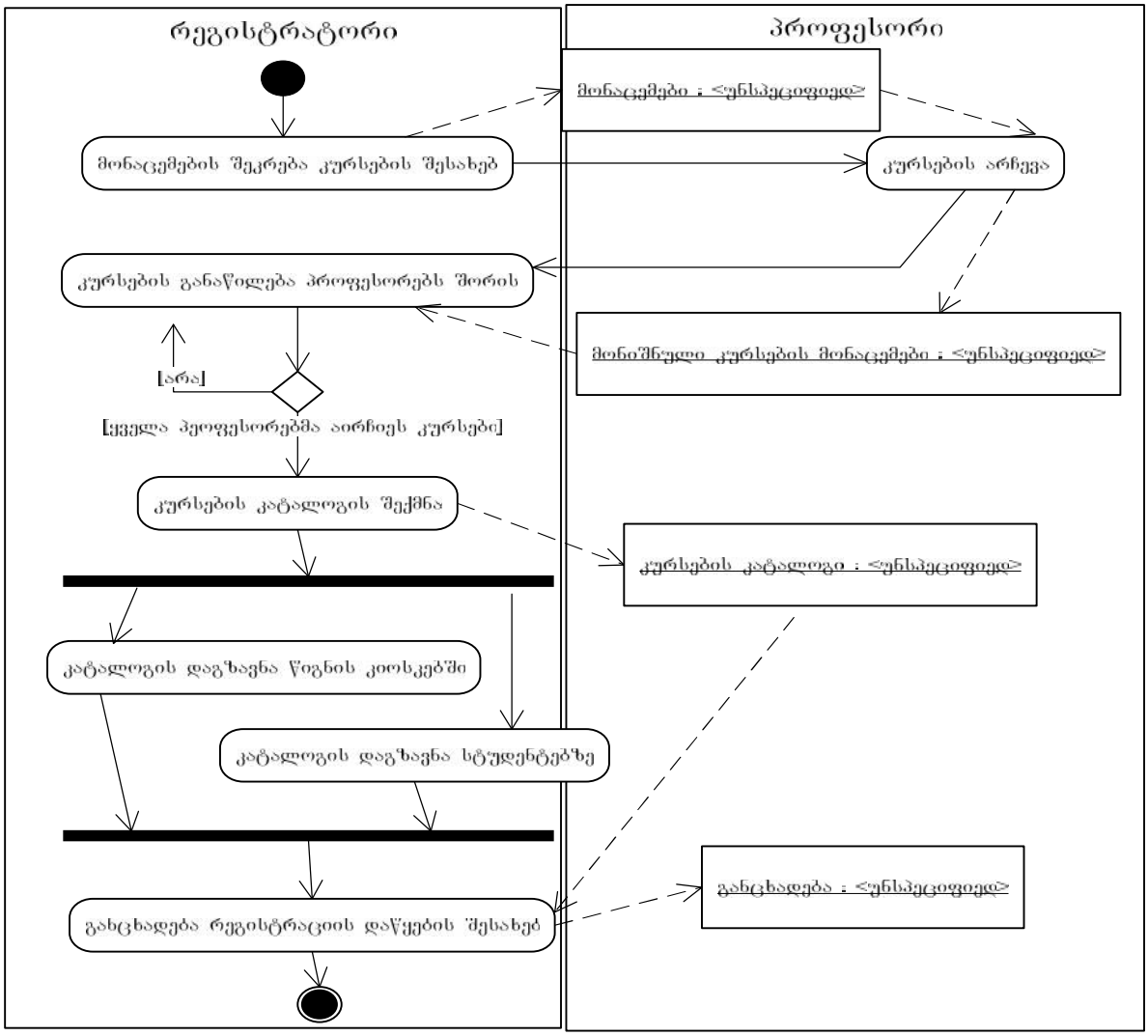
მოცემული გამოყენებითი შემთხვევის შესრულების დაწყების წინ რეგისტრატორი უნდა შევიდეს სისტემაში.

მოთხოვნების ანალიზისას მნიშვნელოვანია აგრეთვე მოღვაწეობის დიაგრამები ობიექტების ნაკადებით და ბილიკებით, რომლებიც აღწერენ ურთიერკავშირს ერთ ან სხვადასხვა Busines Use Case-ს სცენარებს შორის.

მდგომარეობის დიაგრამებს, რომლითაც აღიწერება ცალკეული ბიზნეს-ობიექტების ქცევა.

კურსების კატალოგის და მათი სტუდენტებზე განაწილებისათვის შესაძლებელია ავაგოთ მოღვაწეობის(აქტიურობის) შემდეგი

დიაგრამა(ნახ.2.3.).



ნახ.2.3.

ასეთივე დიაგრამები სასურველია აიგოს სხვა პრეცედენტებისათვის, რაც საშუალებას მოგვცემს სცენარები გამოვსახოთ მოქმედებების თანმიმდევრობის სახით.

3. მოთხოვნების ანალიზი პროგრამული უზრუნველყოფისადმი

3.1.არქიტექტურული ანალიზი

ობირქტ-ორიენტირებული ანალიზის მიზანია პროგრამული უზრუნველყოფისადმი ფუნქციონალური მოთხოვნების ტრანსფორმაცია წინასწარ სისტემურ პროექტში და სისტემის არქიტექტურისათვის სტაბილური საფუძვლის შექმნა. პროექტირების პროცესში სისტემური პროექტი "იტვირთება" რეალიზაციის გარემოში ყველა არაფუნქციონალური მოთხოვნების გათვალისწინებით.

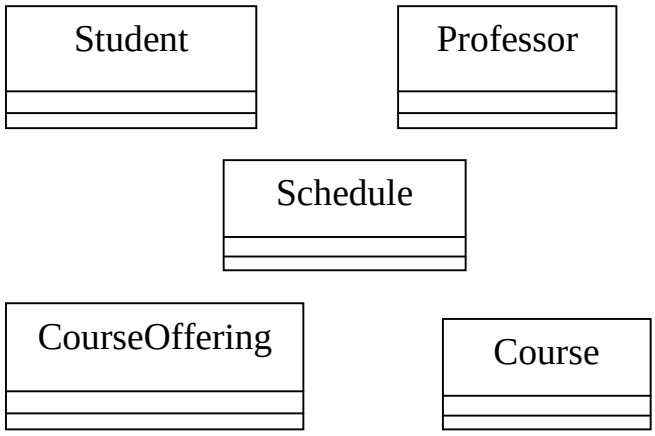
ობიექტ-ორიენტირებული ანალიზი მოიცავს ორი სახის მოღვაწეობას – არქიტექტორულ ანალიზს და პრეცედენტების ანალიზს. არქიტექტორული ანალიზისას თავიდან დგინდება – მოდელირების შესახებ შეთანხმებები, მაგალითად:

- პრეცედენტების დასახელება უნდა იყოს მოკლე ზმნების ფრაზები;
- კლასების დასახელება უნდა იყოს არსებითი სახელი, საპრობლემო სფეროს შესაბამისი;
- კლასების დასახელება უნდა იწერებოდეს მთავრული ასოებით;
- ატრიბუტებისა და ოპერაციების დასახელება უნდა იწერებოდეს დაბალი ასოებით;
- ყველა კლასები და დიაგრამები, რომლებიც აღწერენ წინასწარ სისტემურ პროექტს, თავსდება პაკეტში დასახელებით Analysis Model;

არქიტექტორული ანალიზის შემდეგ ეტაპზე უნდა მოვახდინოთ **ძირითადი აბსტრაქციების იდენტიფიკაცია,** რომლის დანიშნულებაა კლასების ნაკრების(ანალიზის კლასები) წინასწარი განსაზღვრა საპრობლემო სფეროს აღწერისა და სისტემისადმი მოთხოვნების სპეციფიკაციის საფუძველზე(კერძოდ, ლექსიკონის). ძირითადი (არაფორმალური) საშუალება არსების იდენტიფიკაციისა – ეს აბსტრაქციების პოვნაა, რომლებიც აღწერენ ფიზიკურ ან მატერიალურ

ობიექტებს, პროცესებს და მოვლენებს, ადამიანთა როლებს, ორგანიზაციებს და სხვა მცნებებს. ერთადერთ ფორმალურ საშუალებათ არსთა იდენტიფიკაციისა არის საგნობრივი სფეროს ტექსტური აღწერილობების ანალიზი, აღწერიდან არსებითი სახელების გამოყოფა და მათი არჩევა როგორც “კანდიდატებისა” აბსტრაქციების როლზე. ყოველ არსს უნდა გააჩნდეს დასახელება, გამოსახული არსებითი სახელით მხოლობით რიცხვში. თუ მივყვებით ამ რეკომენდაციებს, განვსაზღვრავთ რეგისტრაციის სისტემისათვის ანალიზის ხუთ კლასს (ნახ.3.1.):

- სტუდენტი;
- პროფესორი;
- სასწავლო გრაფიკი;
- კურსი;
- შეთავაზებული კურსი.



ნახ..3.1. რეგისტრაციის სიტემის ანალიზის კლასები

შემდეგ უნდა მოვახდინოთ თითოეული პრეცედენტების ანალიზი.

3.2.პრეცედენტების ანალიზი

პრეცედენტების ანალიზი მოიცავს:

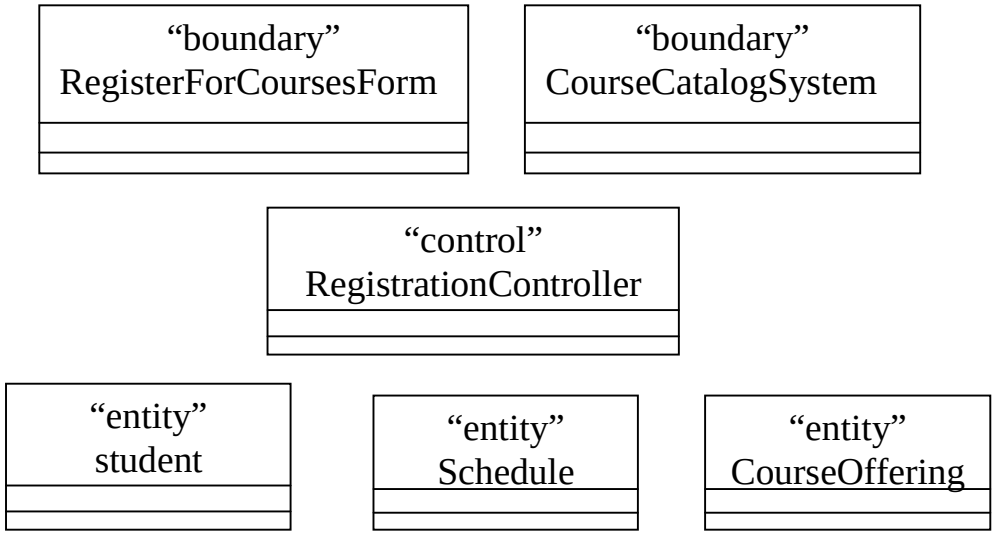
- კლასების იდენტიფიკაციას, რომლებიც მონაწილეობენ პრეცედენტების მოვლენათა ნაკადების რეალიზაციაში;
- ქცევის განაწილება კლასებს შორის (კლასების მოვალეობების განსაზღვრა, რომელიც რეალიზდება პრეცედენტების მიერ,);
- კლასების ატრიბუტებისა და ასსოციაციის განსაზღვრა;
- ანალიზის კლასების უნიფიცირება.

3.2.1. კლასების იდენტიფიკაცია

პრეცედენტის მოვლენათა ნაკადში სამი ტიპის კლასებს გამოავლენენ:

1. **მოსაზღვრე კლასები** (Boundary) - შუამავლები გარე ობიექტებსა და სისტემას შორის ურთიერთქმედებისას. როგორც წესი, ყოველი წყვილისათვის "მოქმედი პირი – პრეცედენტი" განისაზღვრება ერთი მოსაზღვრე კლასი. მოსაზღვრე კლასების ტიპები: მომხმარებლის ინტერფეისი (ინფორმაციის გაცვლა მომხმარებელთან ინტერფეისის დეტალების გარეშე – ღილაკები, ცხრილები, ფანჯრები), სისტემური ინტერფეისი და აპარატული ინტერფეისი (გამოყენებული პროტოკოლები მათი რეალიზაციის გარეშე).
2. **კლასები – არსება** (Entity) – დასამუშავებელი სისტემის ძირითადი აბსტრაქციები (მცნებები). კლასები – არსის გამოვლენის წყაროებია – აბსტრაქციები, რომლებიც იქმნება არქიტექტურული ანალიზისას, პრეცედენტის მოვლენათა ნაკადის აღწერა.
3. **მმართველი კლასები** (Control) – უზრუნველყოფენ ობიექტების ქცევის კოორდინაციას სისტემაში. შეიძლება არ იყოს ზოგიერთ პრეცედენტში, შეზღუდულია მარტივი მანიპულაციებით შენახულ მონაცემებთან. როგორც წესი, ყოველი პრეცედენტისათვის განისაზღვრება ერთი მმართველი კლასი. მმართველი კლასების

მაგალითია: ტრანზაქციების მენეჯერი, რესურსების კოორდინატორი, შეცდომების დამმუშავებელი.



ნახ.3.2.

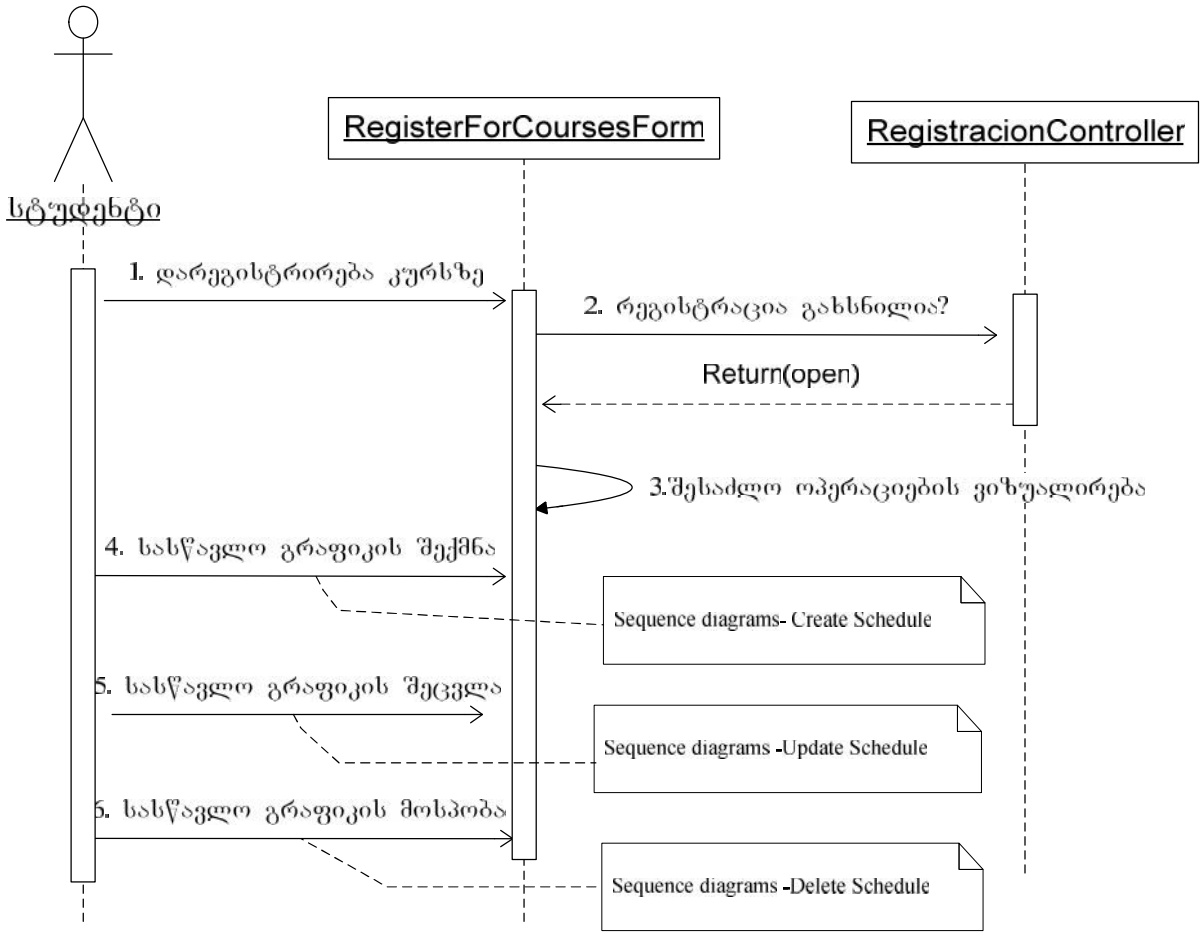
ანალიზის კლასები გამოხატავენ სისტემისადმი ფუნქციონალურ მოთხოვნებს და ახდენენ საგნობრივი სფეროს მოდელირებას. ანალიზის კლასების ერთობლიობა წარმოადგენს სისტემის საწყის კონცეპტუალურ მოდელს. კლასების ნაკრების მაგალითი, რომლებიც მონაწილეობენ პრეცედენტის “კურსზე დარეგისტრირება” რეალიზებაში, მოყვანილია ნახ.3.2.-ზე.

3.2.1. კლასებს შორის მოვალეობების განაწილება

გამოყოფილი სამი ტიპის კლასების დანიშნულებიდან გამომდინარე, შესაძლებელია დავახასიათოთ მოვალეობები მათ შორის:

- მოსაზღვრე კლასები პასუხს აგებენ ურთიერთქმედებაზე სისტემის გარე სამყაროსთან (მოქმედი პირები);
- კლასი არსები პასუხს აგებენ მონაცემთა შენახვასა და მანიპულირებაზე;
- მმართველი კლასები კოორდინაციას უწევენ პრეცედენტების მოვლენათა ნაკადებს.

მოვალეობების უფრო დეტალური განაწილება (კლასების ოპერაციების სახით) სრულდება ურთიერთქმედების დიაგრამებით. პირველ რიგში იგება დიაგრამა, რომელიც აღწერს მონაცემთა ძირითად ნაკადს (ერთი ან რამოდენიმე) და მის დაქვემდებარებულ ნაკადებს.

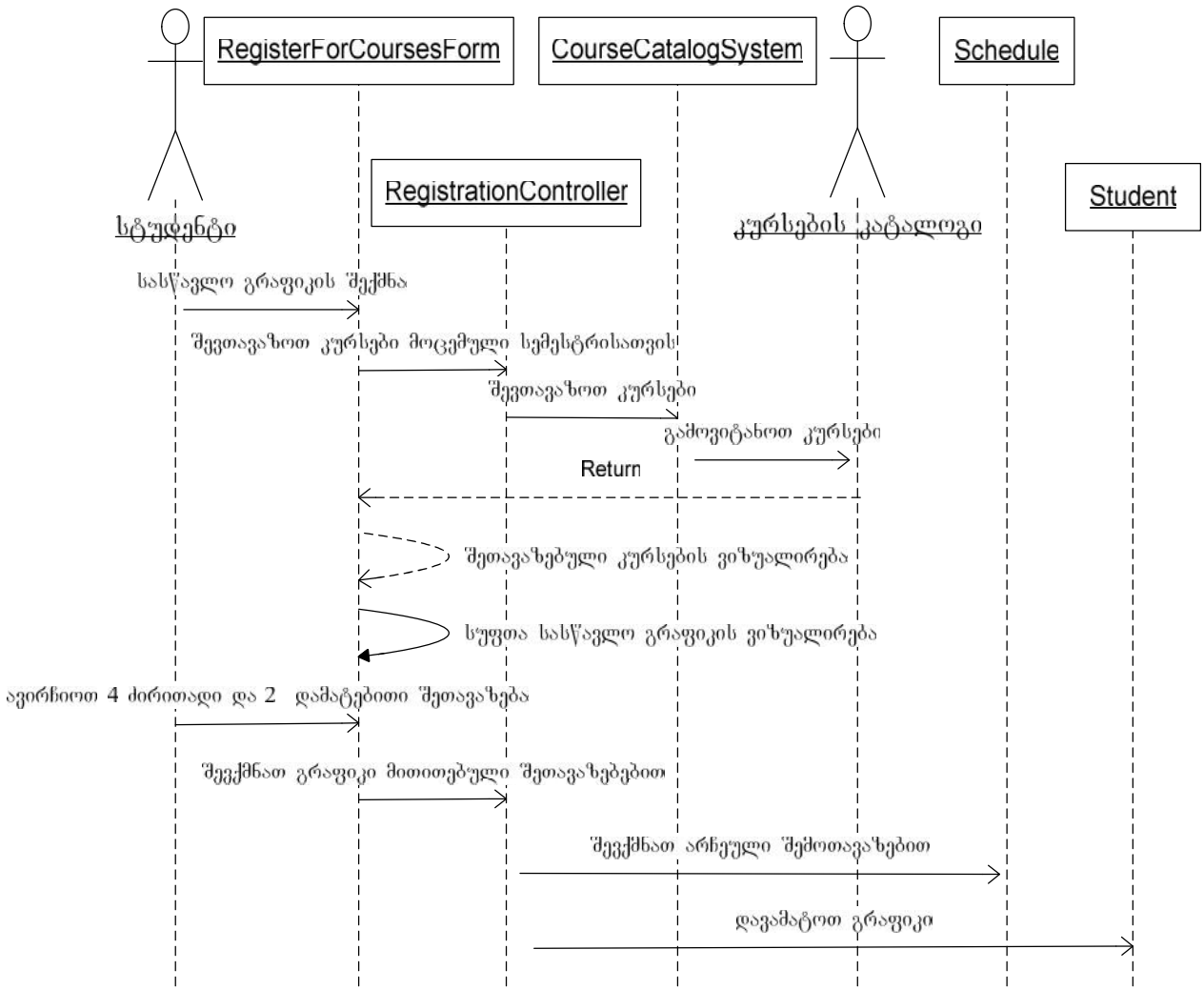


3.3. მიმდევრობის დიაგრამა Register for Courses – ძირითადი ნაკადი შეცდომების დამუშავება

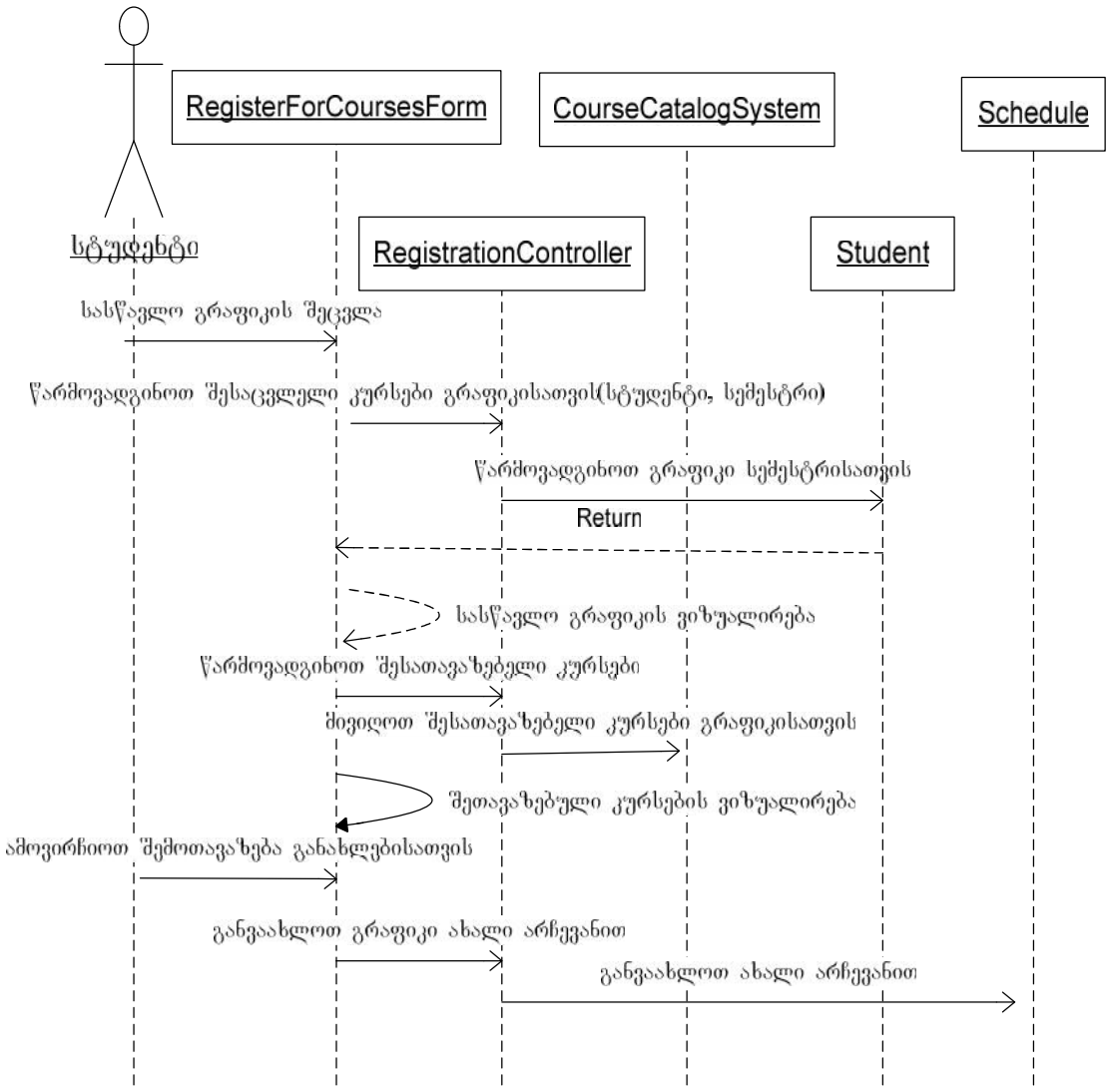
ყოველი ალტერნატიული ნაკადისათვის იგება ცალკე დიაგრამა. მაგალითად:

- შესრულების დროის კონტროლი;
- არასწორად შეტანილი მონაცემების დამუშავება.

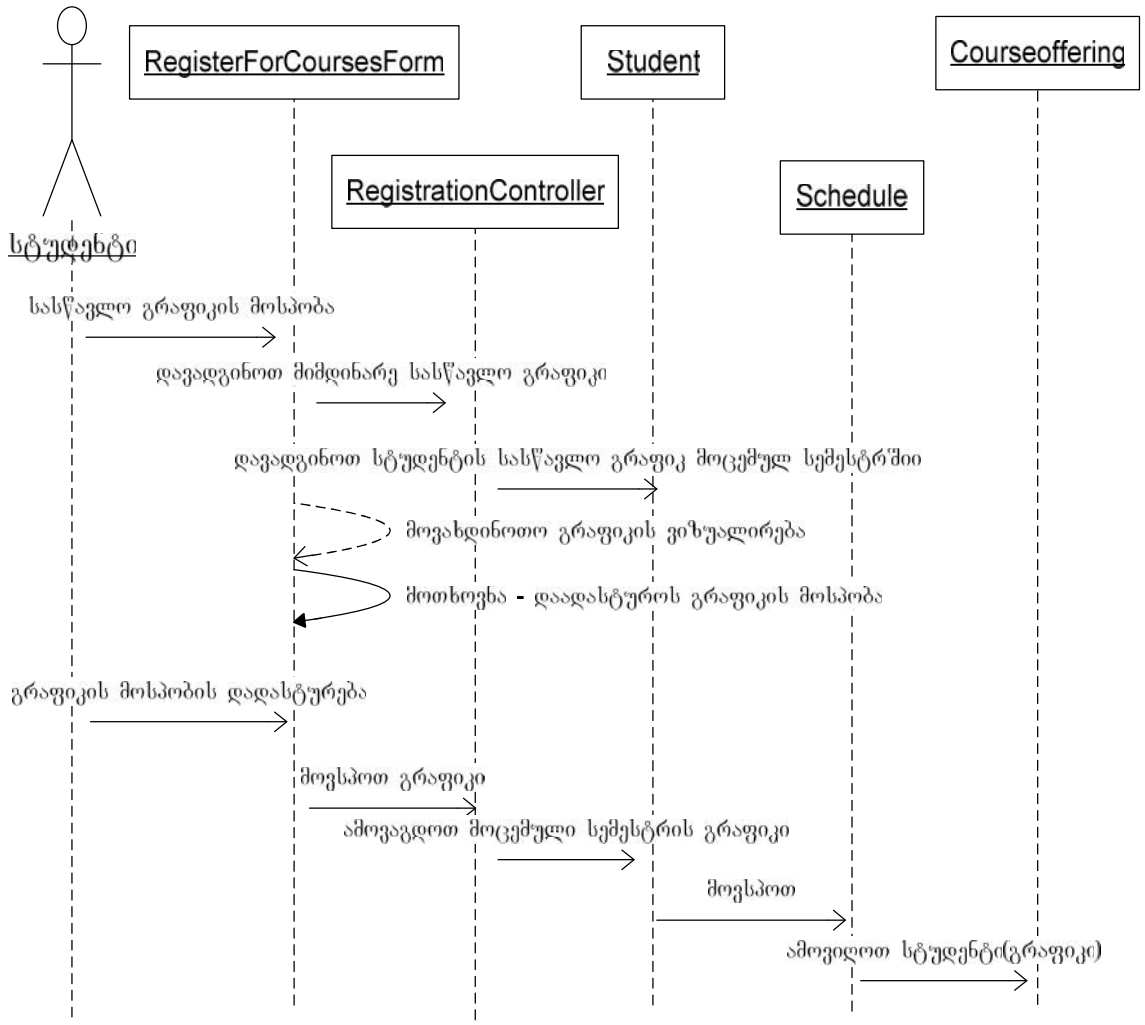
ქვევით მოყვანილია მიმდევრობის დიაგრამები გამოყენებითი შემთხვევისათვის “კურსებზე დარეგისტრირება”.



3.4. მიმდევრობის დიაგრამა Register for Courses – ძირითადი ნაკადი(გრაფიკის შექმნა)

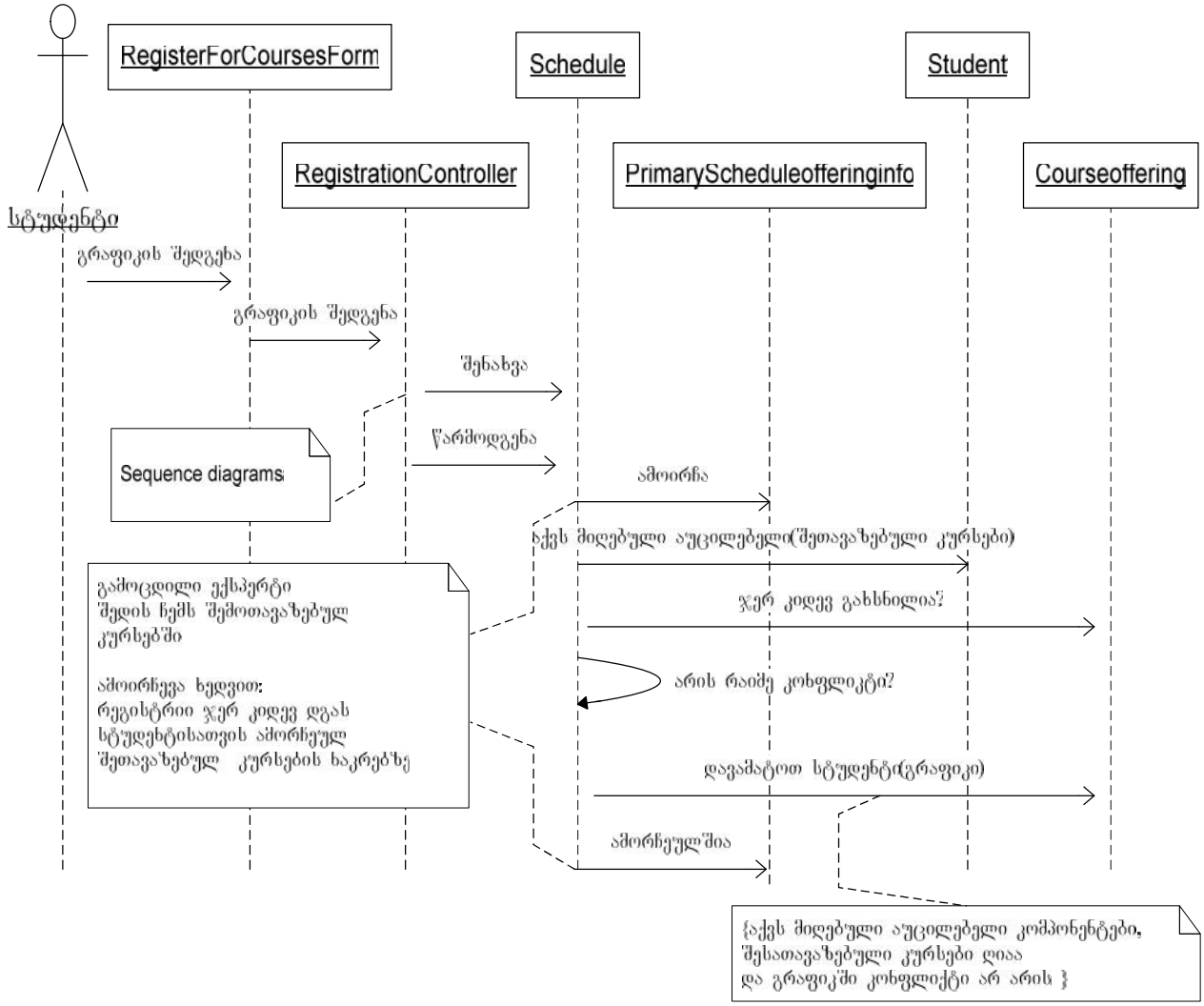


3.5. მიმდევრობის დიაგრამა Register for Courses – ძირითადი ნაკადი(გრაფიკის შეცვლა)



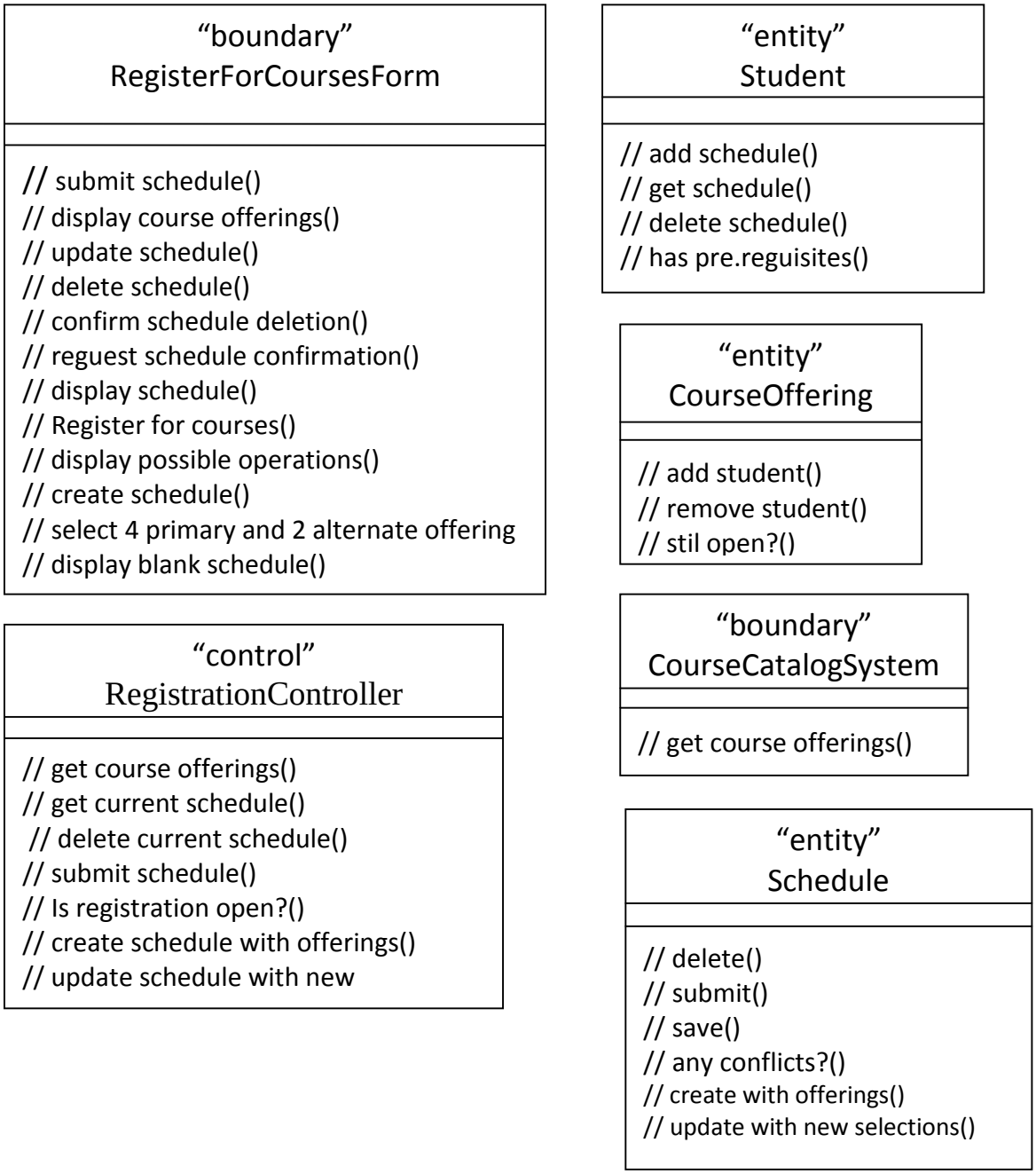
3.6. მიმდევრობის დიაგრამა Register for Courses – ძირითადი ნაკადი(გრაფიკის ამოგდება)

უნდა აღინიშნოს, რომ 3.7. დიაგრამაზე მოყვანილია ახალი კლასის ობიექტი PrimarySheduleOfferinginfo(კლასი ასოციაცია, რომელიც აღწერს კავშირს Shedule და Offeringinfo კლასებს შორის), რომელიც წინასწარ უნდა შეიქმნას.



3.7. მიმდევრობის დიაგრამა Register for Courses – ძირითადი ნაკადი(გრაფიკის შედგენა-განხორციელება)

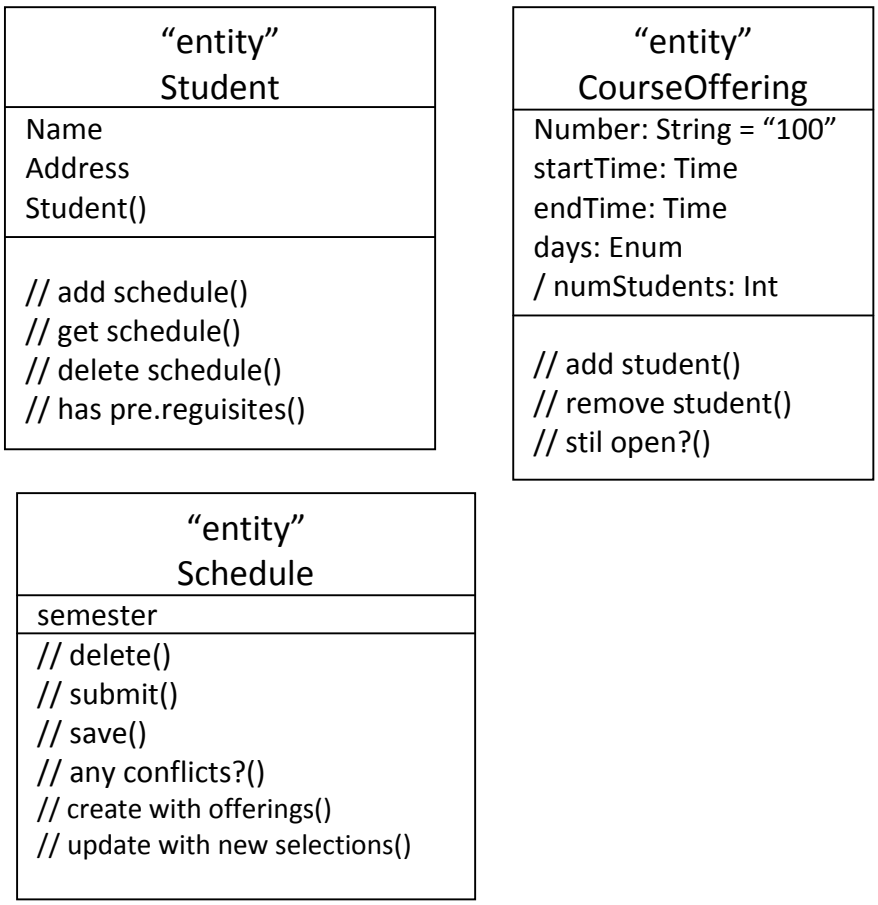
მიმდევრობითი დიაგრამებზე მოყვანილი შეტყობინებები უნდა დაუკავშიროთ ოპერაციებს(აღნიშნულ მოქმედებებს ავტომატურად ასრულებს სისტემა). კლასებში ჩნდება ოპერაციები. შესაბამისად, კლასების დიაგრამა(იხ.ნახ.3.8.), ურთიერთქმედების ოპერაციების აგების შემდეგ მიიღებს შემდეგ სახეს.



ნახ.3.9. კლასების დიაგრამა “კურსებზე დარეგისტრირება”-ში მონაწილე “ანალიზის” ოპერაციებით

3.2.3. კლასების ატრიბუტებისა და ასოციაციის განსაზღვრა

ანალიზის კლასების ატრიბუტები განისაზღვრებიან საგნობრივი სფეროს შესახებ ცოდნისა და სისტემისადმი მოთხოვნების საფუძველზე.



ნახ.3.9. "ანალიზის" კლასები ოპერაციებით და ატრიბუტებით

კლასებს (ასოციაციის) შორის კავშირები განისაზღვრება ორ ეტაპად:

ეტაპი 1. კავშირების საწყისი ნაკრები დგინდება კოოპერაციის დიაგრამების ანალიზის საფუძველზე. თუ ორი ობიექტი ურთიერთქმედებენ (ცვლიან შეტყობინებებს), მათ შორის კოოპერაციის დიაგრამაზე უნდა არსებობდეს კავშირი (ურთიერთქმედების გზა), რომელიც გარდაიქმნება ორმიმართულებიან ასოციაციში შესაბამის კლასებს შორის. თუ შეტყობინებები რომელიმე ობიექტებს შორის გადაიცემიან მხოლოდ ერთი

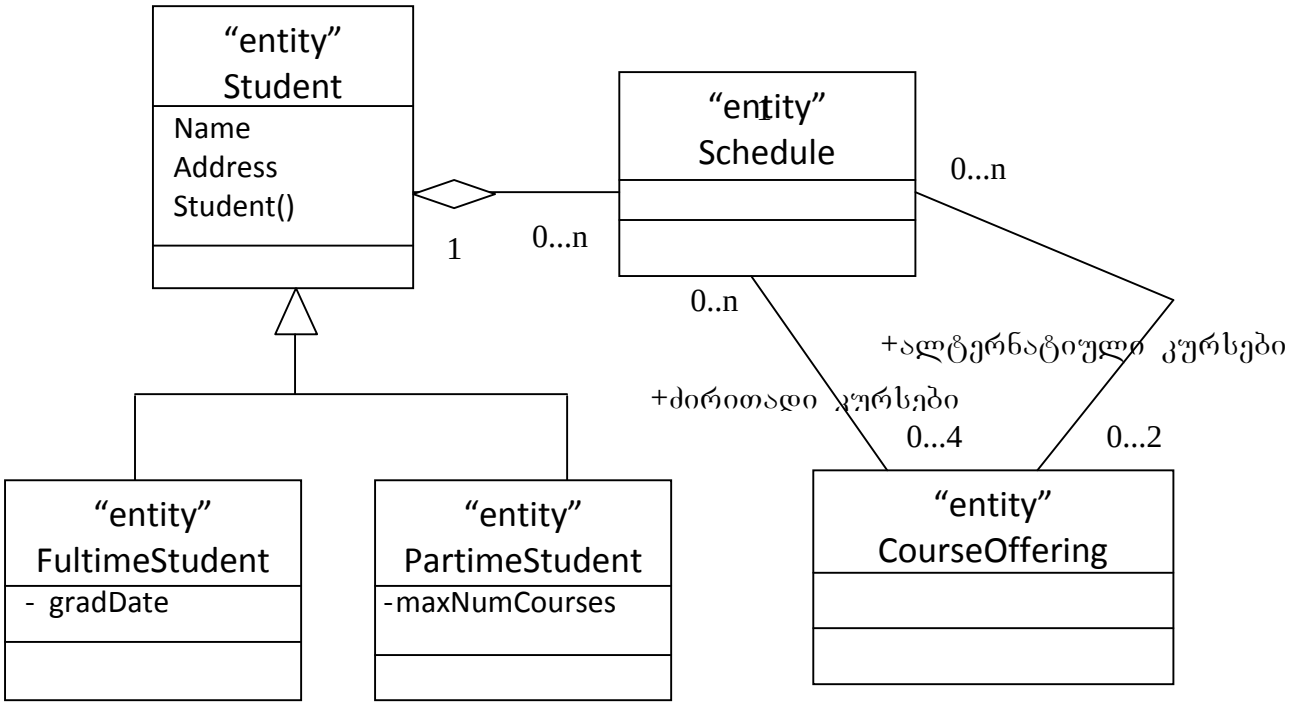
მიმართულებით, მაშინ შესაბამისი ასოციაციისათვის ინიშნება მიმართულების ნავიგაცია.

ეტაპი 2. ანალიზდება და ზუსტდება ასოციაციები კლას-არსებს შორის. დგინდება ასოციაციის სიმძლავრე, შეიძლება გამოყენებულ იქნას მრავლობითი ასოციაციები, აგრეგაციები, განზოგადება და ასოციაცია კლასები.

კავშირების დამატება

დავამატოთ კავშირები კლასებს, რომლებიც მონაწილეობენ პრეცედენტში Register for Courses. კლასებს შორის კავშირების გამოსახვისათვის ავაგოთ სამი ახალი კლასების დიაგრამა კოოპერაციაში Register for Courses პაკეტში.

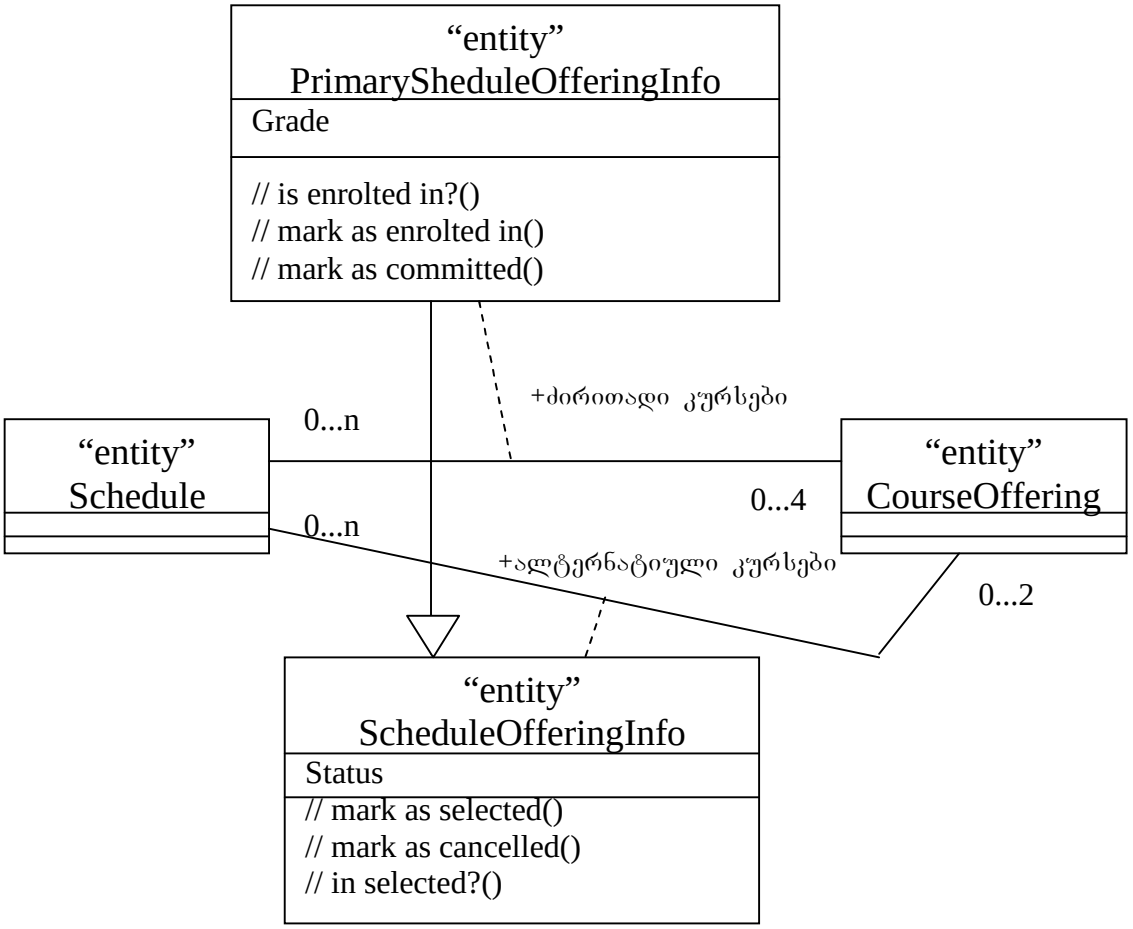
ნახ. 3.10.-ზე ნაჩვენებია მხოლოდ კლასი-არსები. აგრეგაცია კლასებს შორის Student და Schedule გამოსახავს იმ ფაქტს, რომ ყოველი გრაფიკი წარმოადგენს კონკრეტული სტუდენტის საკუთრებას, ეკუთვნის მხოლოდ მას. იგულისხმება, რომ სისტემაში შეინახება არა მარტო მიმდინარე სემესტრის გრაფიკი, არამედ სტუდენტის ყველა გრაფიკები სხვადასხვა სემესტრის. კლასებს შორის Schedule და CourseOffering შემოტანილია ორი ასოციაცია, რამდენადაც კონკრეტული კურსი შესაძლებელია შედიოდეს სტუდენტის გრაფიკში როგორც ძირითადი (არა უმეტეს ოთხი კურსისა) და როგორც ალტერნატიული (არა უმეტეს ორი კურსისა). კლასს Student ემატება ორი ახალი ქვეკლასი – FultimeStudent (დღის განყოფილების სტუდენტი) და artimeStudent (საღამოს განყოფილების სტუდენტი).



ნახ. 3.10. კლასი – არსი დიაგრამა

ნახ.3.11.-ზე მოყვანილია ასსოციაცია – კლასები, რომლებიც წარმოადგენენ კავშირების თვისებებს კლასებს შორის Schedule და CourseOffering. ასოციაცია, რომელიც აკავშირებს გრაფიკს და ალტერნატიულ კურსს, აქვს მხოლოდ ერთი ატრიბუტი – კურსის სტატუსი კონკრეტულ გრაფიკში (status), რომელსაც შეუძლია მიიღოს მნიშვნელობები “ჩართულია გრაფიკში”, “გადაიდო”, “შეტანილია კურსის ცხრილში” და “დაფიქსირებულია გრაფიკში”. თუ კურსი რეგისტრაციის დახურვის პროცესში გადადის ალტერნატიულიდან ძირითადში, მაშინ შესაბამის ასოციაციას ემატება ატრიბუტი “შეფასება”(grade). მაშასადამე, ასოციაცია – კლასი PrimaryScheduleOfferingInfo (ატრიბუტები და ოპერაციები, რომლებიც შედიან ამ კლასში, ეკუთვნიან როგორც ძირითად, ასევე ალტერნატიულ კურსებს) დაემატება თავისი საკუთარი (შეფასება და საბოლაო ჩართვა

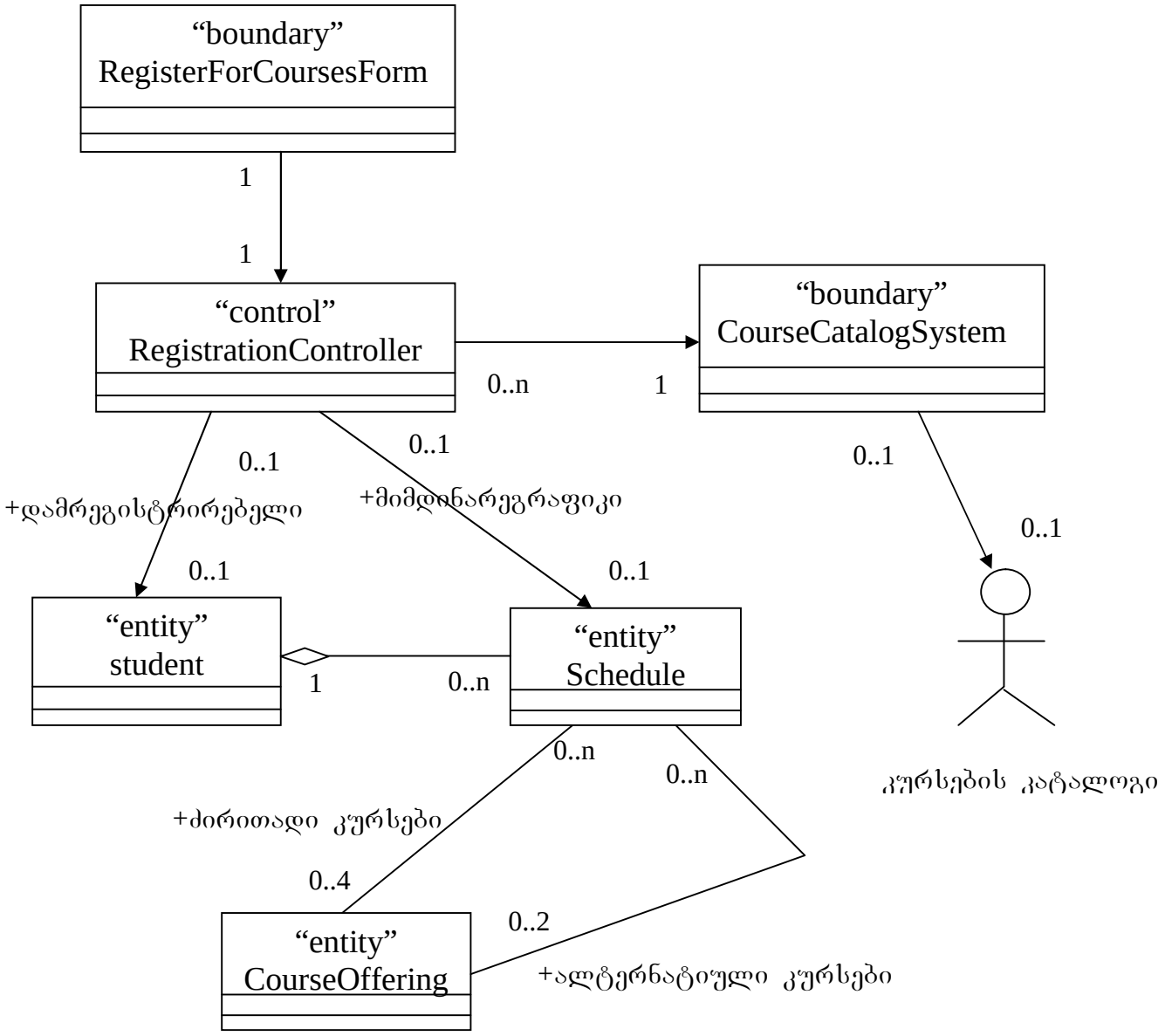
კურსისა გრაფიკში შესაძლებელია ადგილი ქონდეს მხოლოდ ძირითადი კურსებისათვის), რაც ნაჩვენებია განზოგადების კავშირის საშუალებით.



ნახ.3.11. დიაგრამა CourseOfferingInfo (ასოციაცია – კლასების მაგალითები)

ნახ.3.12.-ზე ნაჩვენებია კლასების გაფართოებული დიაგრამა პრეცედენტისათვის “კურსებზე დარეგისტრირება” (ატრიბუტებისა და ოპერაციების გარეშე). ასოციაციები მოსაზღვრე და მმართველ კლასებს შორის, ასევე მმართველ კლასებსა და კლას-არსებს შორის შემოტანილია კოოპერაციის დიაგრამების ანალიზის საფუძველზე და მდგრადი სტრუქტურული (სემანტიკური) კავშირები არსებს შორის გამოხატავენ კავშირებს, რომლებიც დინამიურად აღიძვრებიან შესაბამის ობიექტებს შორის მართვის ნაკადში (დანართის მუშაობის პროცესში). რამდენადაც

ასოციაციისათვის ეს არ არის დამახასიათებელი, მომავალში(პროექტირების პროცესში) ისინი შესაძლებელია გარდაიქმნენ დამოკიდებულებაში.



ნახ.3.12. კლასების მთლიანი დიაგრამა “კურსებზე დარეგისტრირება” – მონაწილე კლასები(ატრიბუტების გარეშე)

4. სისტემის დაპროექტება

ობიექტ – ორიენტირებული დაპროექტების მიზანია წინასწარი სისტემური პროექტის (“ანალიზის” კლასების ნაკრები), რომელიც შეადგენს სისტემის არქიტექტურის სტაბილურ საფუძველს, ადაპტაცია რეალიზაციის სფეროსთან ყველა არაფუნქციონალური მოთხოვნების გათვალისწინებით.
ობიექტ – ორიენტირებული დაპროექტება მოიცავს ორი სახის მოღვაწეობას:

- სისტემის არქიტექტურის დაპროექტება;
- სისტემის ელემენტების დაპროექტება.

4.1. სისტემის არქიტექტურის დაპროექტება

სისტემის არქიტექტურის დაპროექტება მოიცავს:

- “ანალიზის” კლასებს შორის ურთიერთქმედების ანალიზი, ქვესისტემებისა და ინტერფეისების გამოვლენა;
- მართვის ნაკადების სტრუქტურის დაპროექტება;
- სისტემის განაწილებული კონფიგურაციის დაპროექტება;

4.1.1. ქვესისტემებისა და ინტერფეისების გამოვლენა

ქვესისტემების გამოვლენისას არქიტექტორის პირველ მოქმედებას წარმოადგენს “ანალიზის კლასების გარდაქმნა საპროექტო კლასებში (design classes). “ანალიზის” ყოველი კლასისათვის მიიღება ორიდან ერთი გადაწყვეტილება:

- “ანალიზის” კლასი გამოისახება საპროექტო კლასში, თუ იგი მარტივია ან წარმოადგენს ერთადერთ ლოგიკურ აბსტრაქციას;
- “ანალიზის” რთული კლასი შესაძლებელია დაიყოს რამოდენიმე კლასად, გარდაიქმნას პაკეტათ ან ქვესისტემათ.

კლასების გაერთიანება ქვესისტემაში შეიძლება განხორციელდეს სხვადასხვა მოსაზრებით, კერძოთ:

- **ფუნქციონალური კავშირი**: ერთიანდებიან კლასები, რომლებიც მონაწილეობენ ერთ და იმავე პრეცედენტის რეალიზებაში და ურთიერთქმედებენ მხოლოდ ერთმანეთთან;
- **აუცილებლობა**: კლასების ნაკრები, რომლებიც ახდენენ ერთიდაიგივე ფუნქციის რეალიზებას, რომელიც შესაძლებელია გამოვყოთ სისტემიდან ან შევცვალოთ ალტერნატიულზე;
- **შეკავშირება**: ქვესისტემებში აერთიანებენ მეტად დაკავშირებულ კლასებს;
- **განაწილება:** კლასების გაერთიანება, განლაგებულნი ქსელის კონკრეტულ კვანძზე.

შესაძლო ქვესისტემების მაგალითებია:

- კლასების ნაკრები, რომლებიც უზრუნველყოფენ ფუნქციების რთულ კომპლექს (მაგ. უშიშროება და მონაცემების დაცვა);
- მოსაზღვრე კლასები, რომლებიც ახდენენ რთულ სამომხმარებლო ინტერფეისს ან ინტერფეისს გარე სისტემებთან.

ქვესისტემების შექმნისას მოდელში სრულდება შემდეგი გარდაქმნები:

- გასაერთიანებელი კლასები თავსდებიან სპეციალურ პაკეტში ქვესისტემის დასახელებით და სტერეოტიპით „subsistem“.
- კლასების ოპერაციების სპეციფიკაციები გამოიტანება ქვესისტემის ინტერფეისში, სტერეოტიპით „interface“.

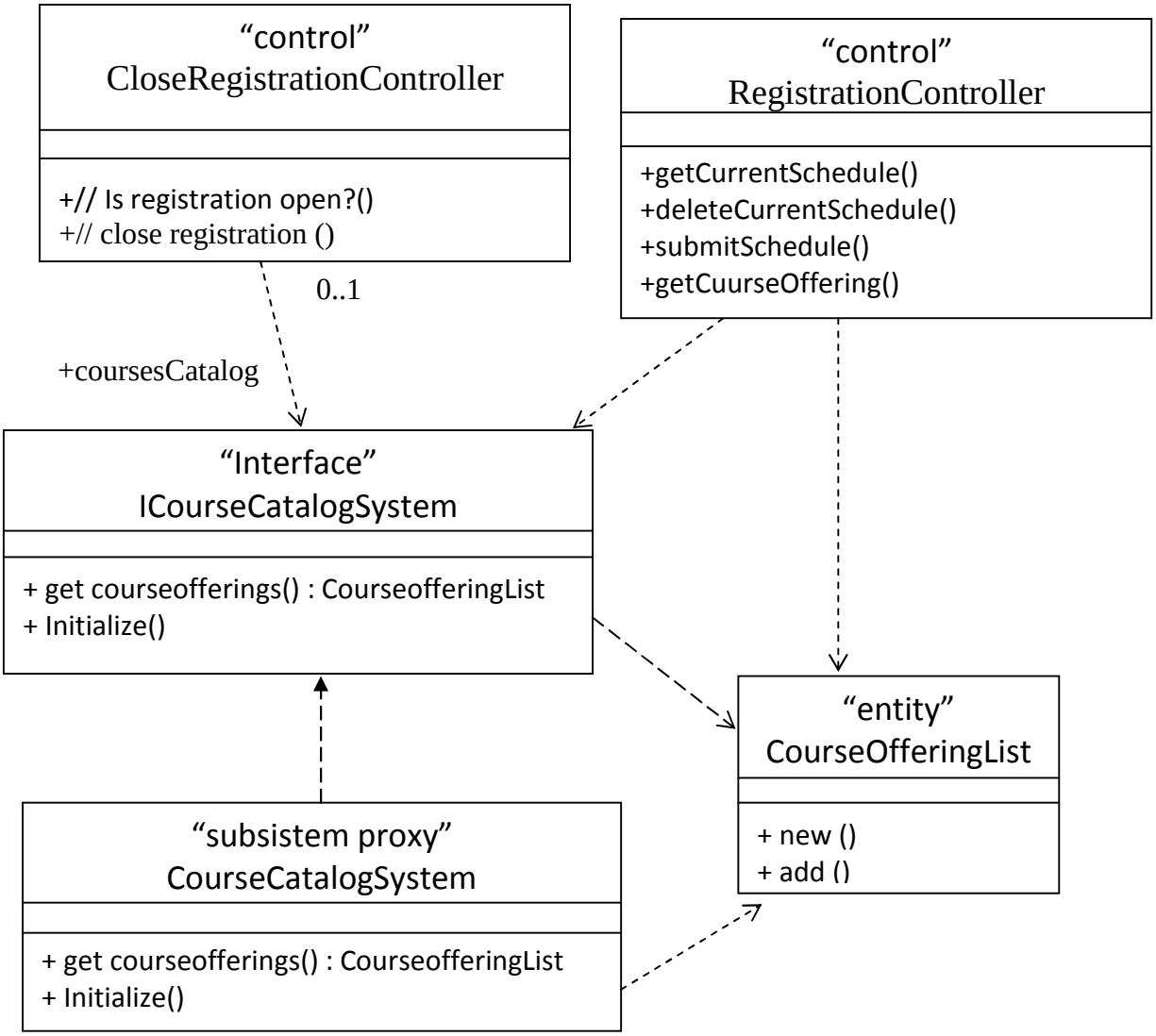
ინტერფეისის აღწერა უნდა შეიცავდეს:

- ინტერფეისის დასახელებას, რომელიც გამოსახავს მის როლს სისტემაში;
- ინტერფეისის ტექსტურ აღწერას, რომლითაც განისაზღვრება მისი მოვალეობები;

ინტერფეისის ოპერაციების გამოყენების ხასიათი და მათი შესრულების თანმიმდევრობა დოკუმენტირებულია ურთიერთქმედების დიაგრამების მეშვეობით, რომლებიც აღწერენ კლასების ურთიერთქმედებას ინტერფეისის ოპერაციების რეალიზაციისას. ეს უკანასკნელნი კი ქვესისტემის კლასების დიაგრამასთან ერთად ერთიანდებიან კოოპერაციაში ინტერფეისის დასახელებით და სტერეოტიპით „interface realization“;

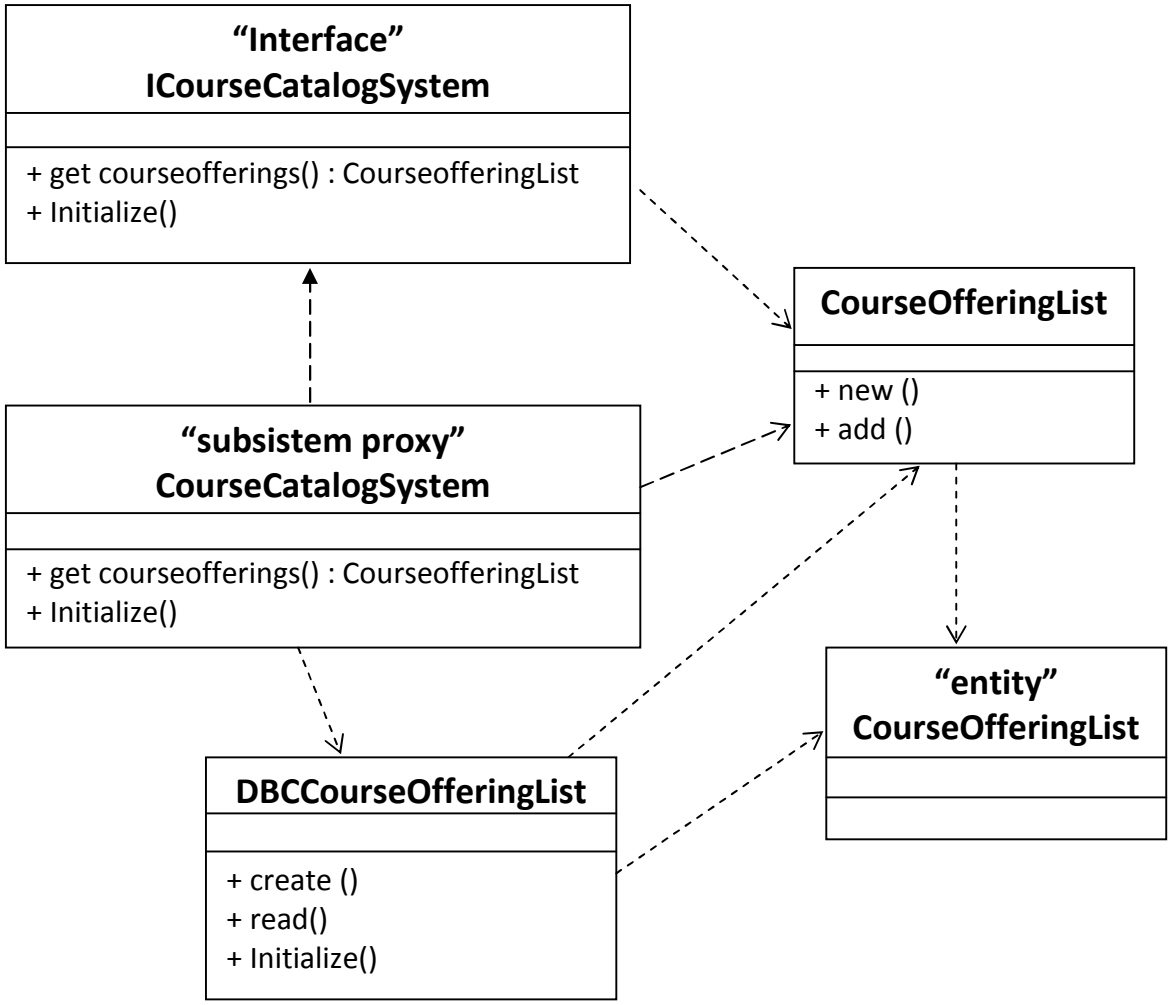
ქვესისტემაში იქმნება კლასი-მოადგილე სტერეოტიპით <<subsistem proxy>>, რომელიც მართავს ინტერფეისის ოპერაციების რეალიზაციას. ქვესისტემის ყველა ინტერფეისები უნდა იყვნენ განსაზღვრულნი არქიტექტურის დაპროექტების პროცესში, რამდენადაც ისინი გამოიყენებიან სინქრონიზაციის წერტილების სახით სისტემის პარალელური დამუშავებისას.

როგორც მაგალითი, მოვიყვანოთ ქვესისტემა CourseCatalogSystem, რომელიც შექმნილია მოსაზღვრე კლასის CourseCatalogSystem მაგიერ. სტრუქტურა და პაკეტის(ქვესისტემის) დიაგრამები CourseCatalogSystem (კლასების და ურთიერთქმედების დიაგრამა, რომლებიც აღწერენ მოცემულ ქვესისტემას და მის ინტერფეისს) მოყვანილია 4.1.-4.3. ნახაზებზე. კლასები DBCCourseOffering და CourseOfferingList პასუხს აგებენ კურსების კატალოგის მონაცემთა ბაზასთან ურთიერთქმედებაზე.

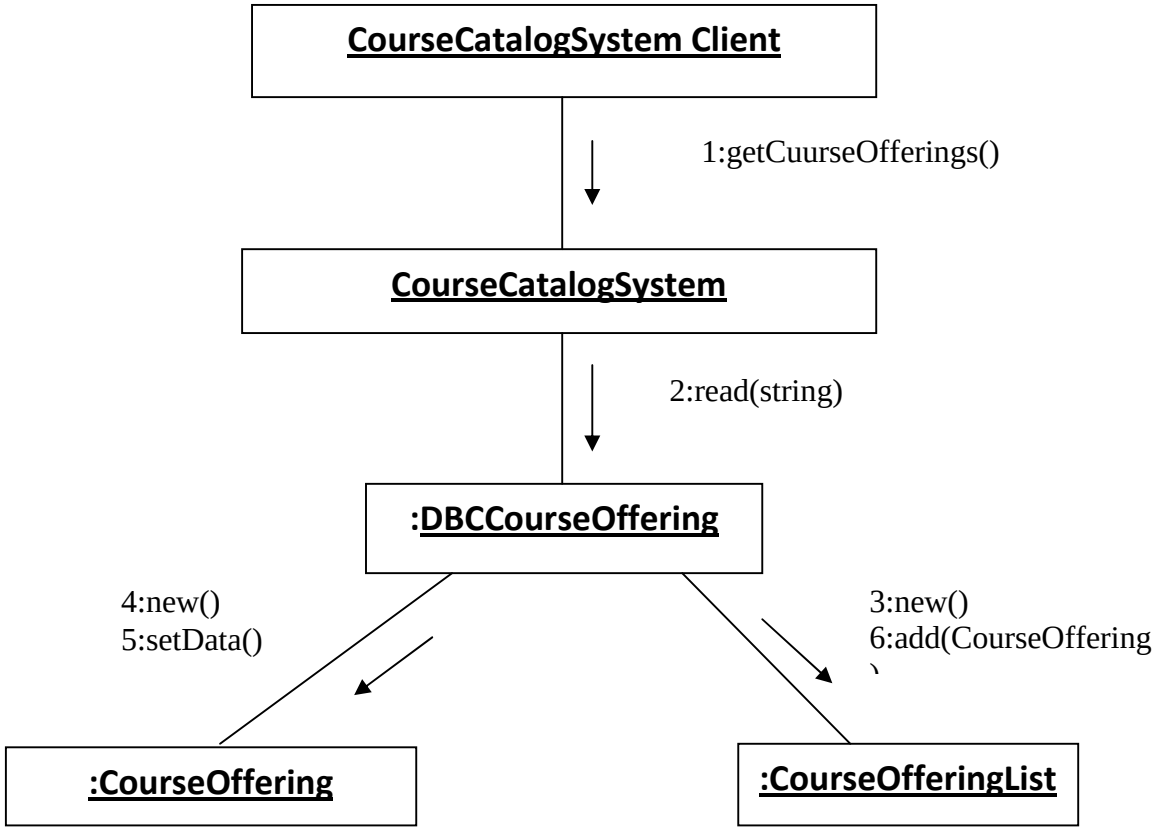


ნახ.4.1.. CourseCatalogSystem ქვესისტემის კონტექსტი არქიტექტორის თვალთახედვით

ობიექტი CourseCatalogSystem Client შესაძლებელია ეკუთვნოდეს CloseRegistrationController ან RegistrationController კლასებს, იმისგან დამოკიდებულებით, რომელ პრეცედენტში მოითხოვება კურსების კატალოგი.



ნახ.4.2. CourseCatalogSystem ქვესისტემის კლასების დიაგრამა დამპროექტებლის თვალთახედვით



ნახ.4.3. კოოპერაციის დიაგრამა, რომელიც აღწერს getCourseOfferings ოპერაციის რეალიზაციას(დიაგრამა ICatalogSystem: getCourseOfferings)

დაპროექტებისას მნიშვნელოვანია პროცესების მოდელირება – სისტემაში პარალელური პროცესების მოდელირება.

4.1.2. მართვის ნაკადების სტრუქტურის დაპროექტება

სტრუქტურის დაპროექტება სრულდება სისტემაში პარალელეური პროცესების (პარარელიზმი) არსებობისას. დაპროექტების მიზანია – სისტემაში არსებული პროცესების გამოვლენა, რეალიზაციის სფეროში მათი ურთიერთქმედების ხასიათის, შექმნის, მოსპობისა და გამოაშკარავებისათვის. პარალელიზმის მოთხოვნა აღიძვრება მაშინ როდესაც:

- საჭიროა დამუშავების განაწილება სხვადასხვა პროცესორებსა და კვანძებს შორის;
- სისტემა იმართება მოვლენათა ნაკადით (event-driven system);
- სისტემაში ერთდროულად მუშაობს მრავალი მომხმარებელი.

კურსებზე რეგისტრაციის სისტემა ფლობს პარალელიზმის თვისებას, რამდენადაც იგი უნდა უზრუნველყოფდეს მრავალი მომხმარებლის (სტუდენტებისა და პროფესორების) ერთდროულ მუშაობას, რომელთაგან თვითეული ბადებს სისტემაში დამოუკიდებელ პროცესს.

პროცესი (process) – ეს რესურსებით უზრუნველყოფილი მართვის ნაკადია, რომელიც სრულდება სხვა პროცესების პარალელურად. მაღალი სირთულის გამო შესაძლებელია გაიყოს ორ ან მეტ ნაკადათ. ნებისმიერი კლასის ობიექტი უნდა არსებობდეს ერთი პროცესის შიგნით მაინც.

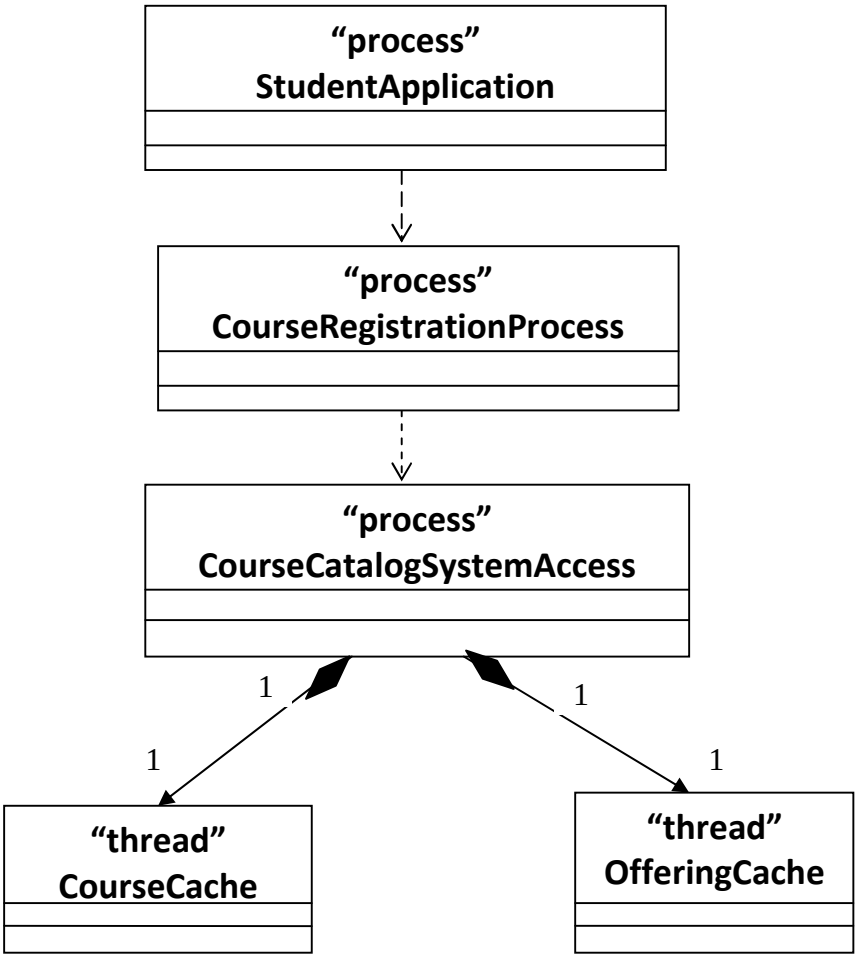
ძაფი (Thread) – ეს შედარებით მცირე მართვის ნაკადია, რომელიც სრულდება სხვა ძაფების პარალელურად ერთ და იმავე პროცესის ფარგლებში.

ნაკადების შექმნის აუცილებლობა კურსების რეგისტრაციის სისტემაში განისაზღვრება შემდეგი მოთხოვნებით:

- თუ კურსი აღმოჩნდება შევსებული იმ დროს, როდესაც სტუდენტი ახდენს სასწავლო გრაფიკის ფორმირებას, რომელიც მოიცავს მოცემულ კურსს, მაშინ იგი უნდა ინფორმირებული იყოს ამის შესახებ (საჭიროა დამოუკიდებელი პროცესი, რომელიც მართავს კონკრეტულ კურსებთან მიმართვას);
- კურსების კატალოგის არსებული მონაცემთა ბაზა ვერ უზრუნველყოფს საჭირო წარმადობას (აუცილებელია შუალედური დამუშავების პროცესი – მონაცემების ამოქაჩვა).

პროცესებისა და ნაკადების რეალიზაცია უზრუნველყოფილია ოპერაციული სისტემის საშუალებით.

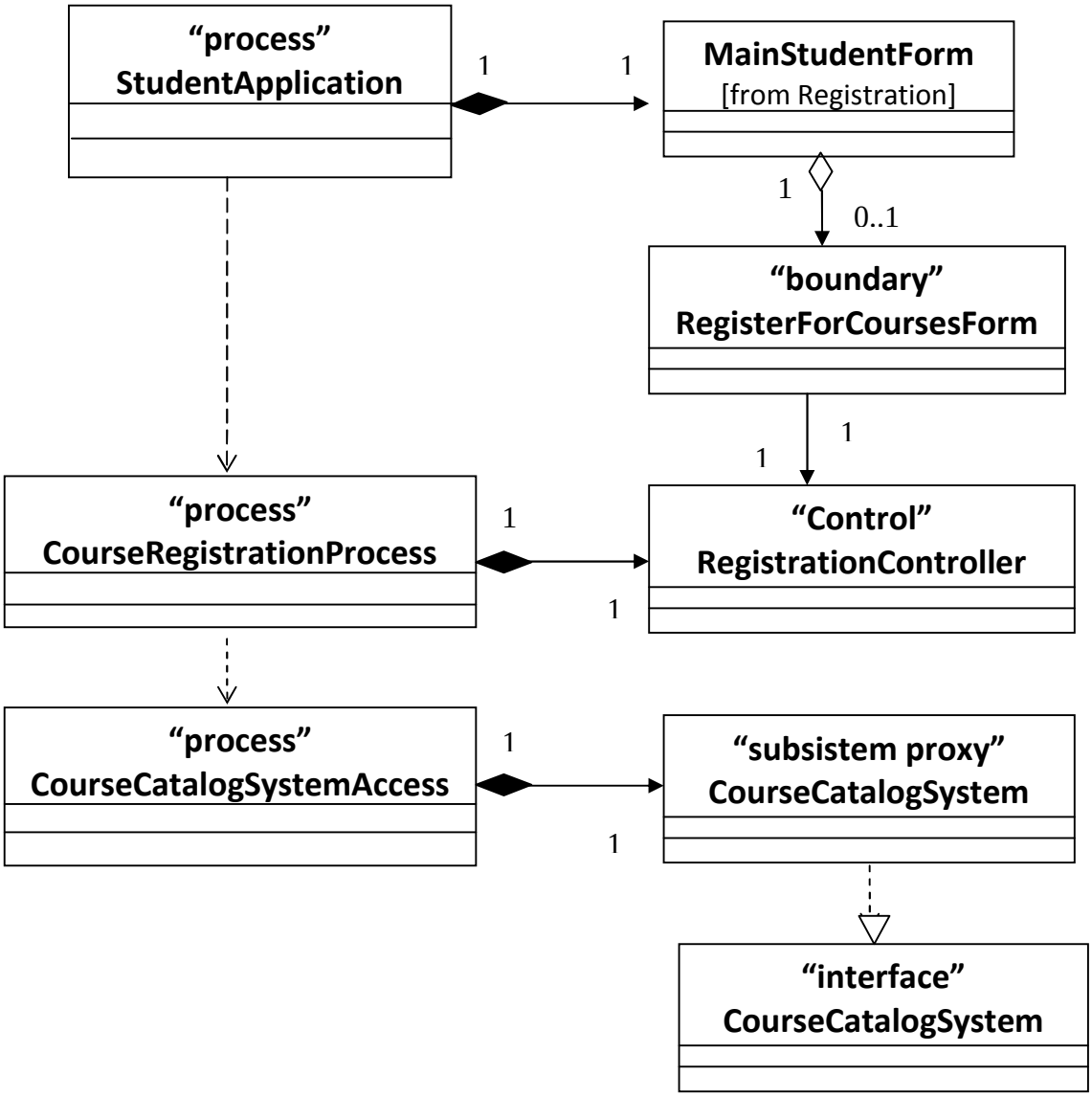
მართვის ნაკადების სტრუქტურების მოდელირებისათვის გამოიყენება აქტიური კლასი – კლასი სტერეოტიპით „process“ და „thread“. აქტიური კლასი ფლობს საკუთარ პროცესს ან ნაკადს და შეუძლია მმართველი ზემოქმედების ინიციირება. კავშირები პროცესებს შორის მოდელირდება როგორც დამოკიდებულება. ნაკადები შესაძლებელია არსებობდეს მხოლოდ პროცესების შიგნით, ამიტომ კავშირები პროცესებსა და ნაკადებს შორის მოდელირდება როგორც კომპოზიცია.



ნახ.4.6. პროცესები და ნაკადები

მაგალითისათვის ნახ.4.6.-4.8.-ზე მოყვანილია კლასების დიაგრამის ფრაგმენტი, რომლებიც აღწერენ სტუდენტის კურსებზე რეგისტრაციის პროცესის სტრუქტურას. აქტიური კლასები ასრულებენ შემდეგ ფუნქციებს:

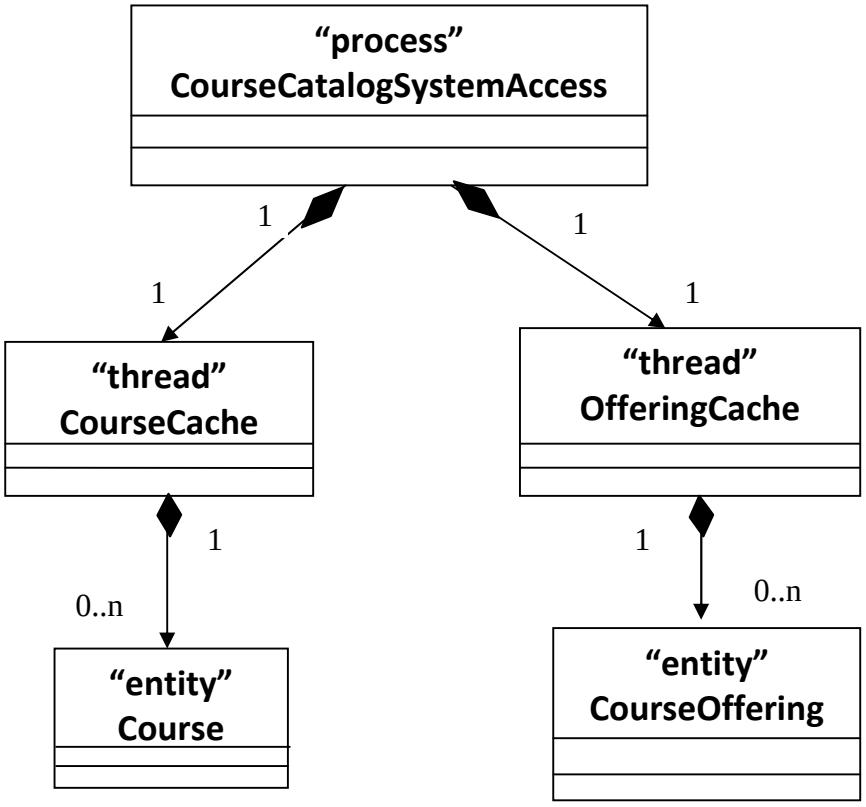
- StudentApplication - პროცესი, რომელიც მართავს სტუდენტი-მომხმარებლის ყველა ფუნქციას სისტემაში. ყოველი სტუდენტისათვის, რომელიც იწყებს რეგისტრაციას კურსებზე, იქმნება ერთი ობიექტი მოცემული კლასის;



ნახ.4.7. კლასები, დაკავშირებულნი პროცესებთან

- CourseRegistrationProcess - პროცესი, რომელიც მართავს უშუალოდ სტუდენტის რეგისტრაციას. ყოველი სტუდენტისათვის, რომელიც იწყებს რეგისტრაციას კურსებზე, ასევე იქმნება ერთი ობიექტი მოცემული კლასის;

- CourseCatelogSystemAccess - მართავს კურსების კატალოგთან მიმართვას, მოცემული კლასის ერთიდაიგივე ობიექტი გამოიყენება ყველა მომხმარებლის მიერ კურსების კატალოგთან მიმართვისას.
- CourseCashe და OfferjngCashe - გამოიყენებიან მონაცემებთან ასინქრონული მიმართვისათვის მონაცემთა ბაზაში სისტემის წარმადობის გაზრდის მიზნით. ისინი წარმოადგენენ მონაცემთა ბაზიდან აღებული კურსების შესახებ კეშ მონაცემებს შუალედური შენახვისათვის.



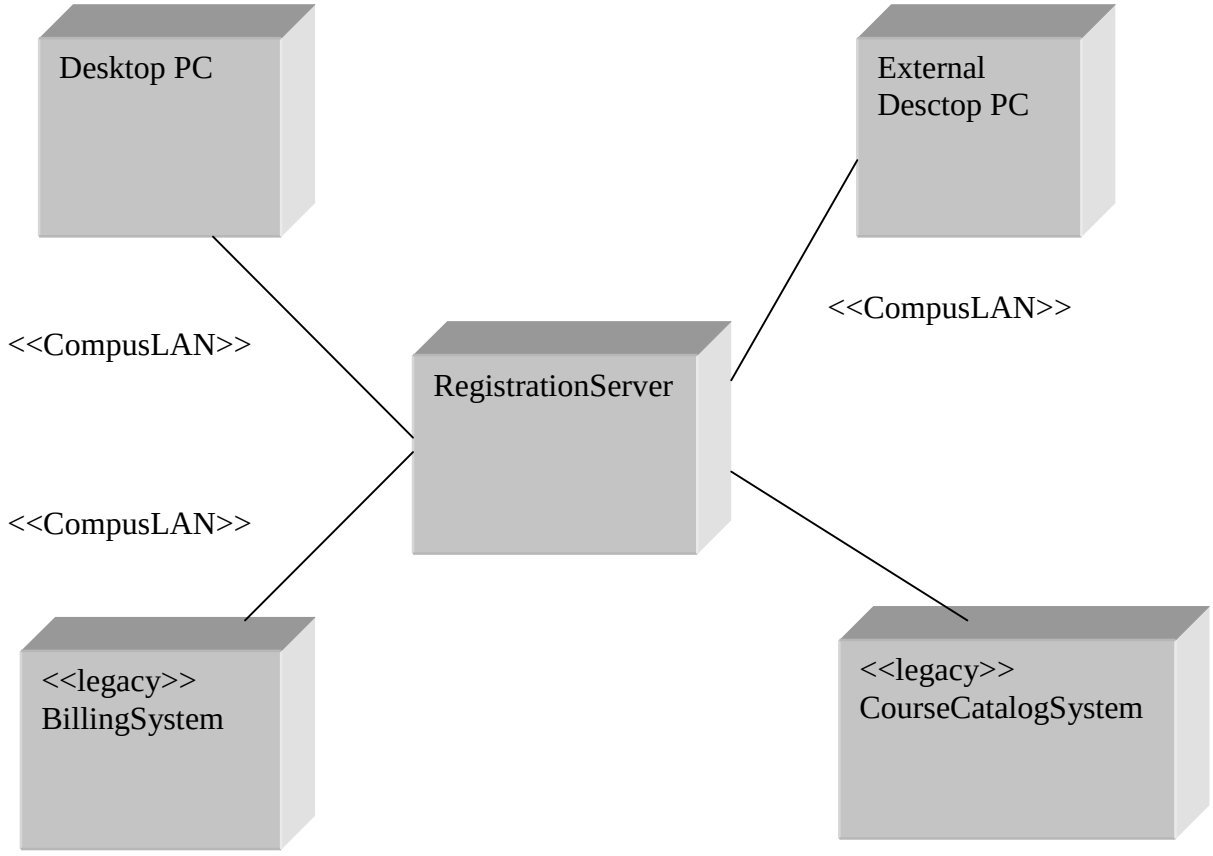
ნახ.4.8. კლასები, დაკავშირებულნი ნაკადებთან

4.1.3. სისტემის განაწილებული კონფიგურაციის მოდელირება

თუ შესაქმნელი სისტემა წარმოადგენს განაწილებულს, საჭიროა მისი კონფიგურაციის დაპროექტება გამოთვლით გარემოში, ე.ი. აღვწეროთ

გამოთვლითი რესურსები, მათ შორის კომუნიკაცია და რესურსების გამოყენება სხვადასხვა სისტემური პროცესების მიერ. სისტემის განაწილებული ქსელური კონფიგურაცია მოდელირდება განლაგების დიაგრამით. მისი ძირითადი ელემენტებია:

- კვანძი (node) – გამოთვლითი რესურსი (პროცესორი ან სხვა მოწყობილობა (დისკური მეხსიერება, სხვადასხვა მოწყობილობების კონტროლერები და ა.შ.). კვანძისათვის შესაძლებელია მიუთითოთ მათზე მიმდინარე პროცესები;

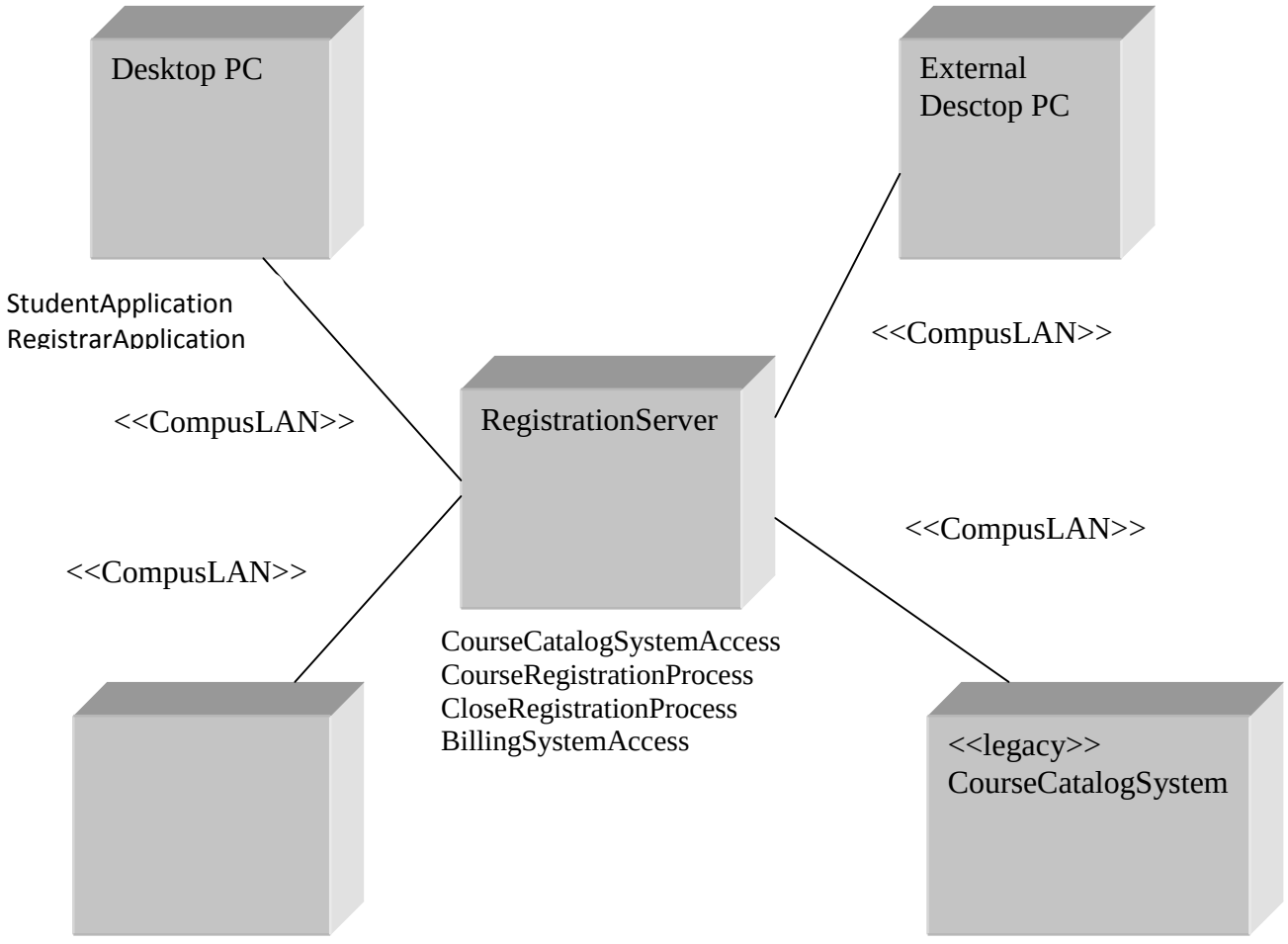


ნახ.4.9. რეგისტრაციის სისტემის ქსელური კონფიგურაცია

- შეერთება (connection) – კვანძების ურთიერთქმედების არხი (ქსელი). ასეთი დიაგრამის მაგალითი რეგისტრაციის სისტემისათვის მოყვანილია ნახ.4.9-ზე.

პროცესების განაწილება კვანძების მიხედვით ხდება შემდეგი ფაქტორების გათვალისწინებით:

- გამოყენებული განაწილების ნიმუშები (სამრგოლიანი კლიენტ – სერვერული კონფიგურაცია, „მსხვილი“ და „თხელი“ კლიენტები, და ა. შ);
- გამოძახების დრო;
- კვანძის სიმძლავრე;
- მოწყობილობის და კომუნიკაციის საიმედობა.



ნახ.4.10. რეგისტრაციის სისტემის ქსელური კონფიგურაცია პროცესების კვანძების მიხედვით განაწილებით

4.2. სისტემის ელემენტების დაპროექტება

სისტემის ელემენტების დაპროექტება მოიცავს:

- პრეცედენტების აღწერის დაზუსტებას (ურთიერთქმედებისა და კლასების დიაგრამების მოდიფიცირება კლასებისა და ქვესისტემების დაპროექტებისას ახლად გამოვლენილი ფაქტორების გათვალისწინებით).
- კლასების დაპროექტება.
- მონაცემთა ბაზის დაპროექტება.

4.2.1. კლასების დაპროექტება

კლასების დაპროექტება მოიცავს:

- საპროექტო კლასების დეტალიზირება.
- ოპერაციებისა და ატრიბუტების დაზუსტება.
- კლასების მდგომარეობების მოდელირება.
- კლასებს შორის კავშირების დაზუსტება.

4.2.1.1. საპროექტო კლასების დეტალიზება

ყოველი მოსაზღვრე კლასი გარდაიქმნება კლასების გარკვეულ ნაკრებში თავისი დანიშნულებიდან გამომდინარე. ეს შესაძლებელია იყოს მომხმარებლის ინტერფეისის ელემენტების ნაკრები, დამოკიდებული დამუშავების გარემოს შესაძლებლობებზე, ან კლასების ნაკრები, რომელიც ახდენს სისტემურ ან აპარატულ ინტერფეისს.

კლასი – არსები წარმადობისა და მონაცემთა დაცვის მოსაზრებებიდან გამომდინარე შესაძლებელია დაიყოს რიგ კლასებად. საფუძველს გაყოფისათვის წარმოადგენს კლასებში ატრიბუტების სხვადასხვა სიხშირით გამოყენების ან ხედვის არსებობა. ასეთი ატრიბუტები, როგორც წესი, გამოიყოფიან ცალკე კლასებათ.

მმართველი კლასები, რომლებიც ახდენენ ინფორმაციის უბრალო გადაცემას მოსაზღვრე კლასებიდან არსებზე, შეიძლება გაძევებულ იქნან. შენარჩუნებული იქნება კლასები, რომლებიც ასრულებენ არსებით სამუშაოს მონაცემთა ნაკადების სამართავად (ტრანზაქციების მართვა, განაწილებული დამუშავება და ა.შ.).

მიღებული ამ დაზუსტებების შედეგად კლასები ექვემდებარებიან უშუალო რეალიზაციას სისტემის კოდში.

4.2.1.2. ატრიბუტებისა და ოპერაციების დაზუსტება

კლასების მოვალეობები, რომლებიც განისაზღვრნენ ანალიზის პროცეცში გარდაიქმნებიან ოპერაციებში, რომლებიც რეალიზებულნი უნდა იყვნენ კოდში. ამასთან:

- ყოველ ოპერაციას ენიჭება მოკლე სახელი, რომელიც ახასიათებს მის შედეგს;
- განისაზღვრება ოპერაციების მთელი სიგნატურა (UML ენაში მიღებული ნოტაციის შესაბამისად);
- იქმნება ოპერაციის მოკლე დასახელება, მისი ყველა პარამეტრების შინაარსის გათვალისწინებით;
- იქმნება ოპერაციების ხედვა: public, private ან proteqted;
- განისაზღვრება ოპერაციების მოქმედების არე: ეგზემპლიარი (ობიექტის ოპერაცია) ან კლასიფიკატორი (კლასების ოპერაცია);
- შესაძლებლია აღიწეროს ოპერაციის შესრულების ალგორითმი (მოღვაწეობის დიაგრამის ან ურთიერთქმედების დიაგრამის გამოყენებით).

კლასების ატრიბუტების დაზუსტება (ნახ.4.11) მდგომარეობს შემდეგში:

- ატრიბუტის დასახელების გარდა მიეთითება მისი ტიპი და მნიშვნელობა დუმილით (არააუცილებელი);
- გათვალისწინებულია შეთანხმება ატრიბუტების დასახელების შესახებ, რომელიც მიღებული იყო პროექტში და რეალიზაციის ენაში;
- მიეთითება ატრიბუტების ხედვა: public, private ან proteqted;
- აუცილებლობის შემთხვევაში განისაზღვრება წარმოებული (გამოთვლითი) ატრიბუტები.

“entity” **Student**
-name : string -address : string <<class>> - nextAailID : Int -studentId : int -dateofBirth : Date
+ getTuition() : double + addSchedule(theSchedule : Schedule) + getSchedule(forSemester : Semester) : Schedule + deleteSchedule(forSemester : Semester) + hasPrereguisites(forCourseOffering : CourseOffering) : boolean // passed(the CourseOffering : CourseOffering) : Boolean <<class>> + getNextAailID() : Int + getStudentID() : Int + getName() : string +getAddress() : string

ნახ.4.11. კლასი Student ატრიბუტებითა და ოპერაციებით

4.2.1.3. მდგომარეობების მოდელირება კლასებისათვის

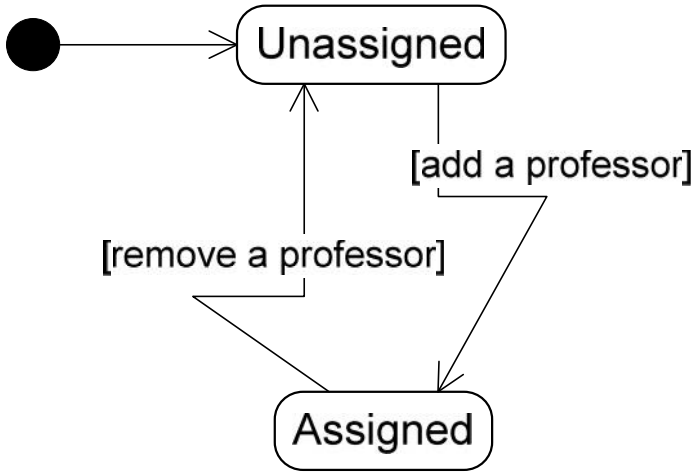
თუ სისტემაში გვაქვს მდგომარეობისაგან დამოკიდებული ობიექტები ქცევის რთული დინამიკით, მაშინ მათთვის შესაძლებელია ავაგოთ მოდელი, რომელიც აღწერს ობიექტის მდგომარეობებს და გადასვლებს მათ შორის. ეს მოდელი წარმოსდგება მდგომარეობის დიაგრამის სახით.

მაგალითისათვის განვიხილოთ კლასი CourseOffering – ის ობიექტის ქცევა. მდგომარეობის დიაგრამა იგება რამოდენიმე ეტაპად.

ეტაპი 1. მდგომარეობების იდენტიფიცირება. მდგომარეობათა გამოვლენის ნიშნებია ობიექტის ატრიბუტების მნიშვნელობების შეცვლა და კავშირების გაწყვეტა სხვა ობიექტებთან. ასე მაგალითად, ობიექტი CourseOffering შეიძლება იმყოფებოდეს მდგომარეობაში Open (კურსზე მიღება გახსნილია) მანამდე, სანამ მასზე დარეგისტრირებულ სტუდენტთა რაოდენობა არ გადააჭარბებს 10-ს, ხოლო როგორც კი გადააჭარბებს 10-ს, ობიექტი გადადის მდგომარეობაში Closed (კურსზე მიღება დახურულია). ამას გარდა, ობიექტი CourseOffering შესაძლებელია იმყოფებოდეს მდგომარეობაში Unassigned (იგი არავის არ მიყავს ანუ არ არსებობს კავშირი Professor – ის რომელიმე ობიექტთან) ან Assigned (ასეთი კავშირი არსებობს).

ეტაპი 2. მოვლენათა იდენტიფიცირება. მოვლენები, როგორც წესი დაკავშირებული არიან გარკვეული ოპერაციების შესრულებასთან. პრეცედენტ „ავირჩიოთ კურსები სწავლებისათვის“ ანალიზისას კლასში CourseOffering მოვალეობების განაწილების შედაგად განისაზღვრა ორი ოპერაცია – addProfessor და removeProfessor, დაკავშირებულნი გარკვეული პროფესორის მიერ კურსების არჩევასთან (ახალი კავშირის შექმნა) და არჩეული კურსზე უარის თქმა (კავშირის გაწყვეტა). ამ ოპერაციებს შეუსაბამებენ ორ მოვლენას – addProfessor და removeprofessor.

ეტაპი 3. მდგომარეობებს შორის გადასვლების იდენტიფიცირება. გადასვლები გამოიძახებიან მოვლენებით. მაშასადამე, Unassigned და Assigned ერთდებიან ორი გადასვლით (ნახ.4.12).

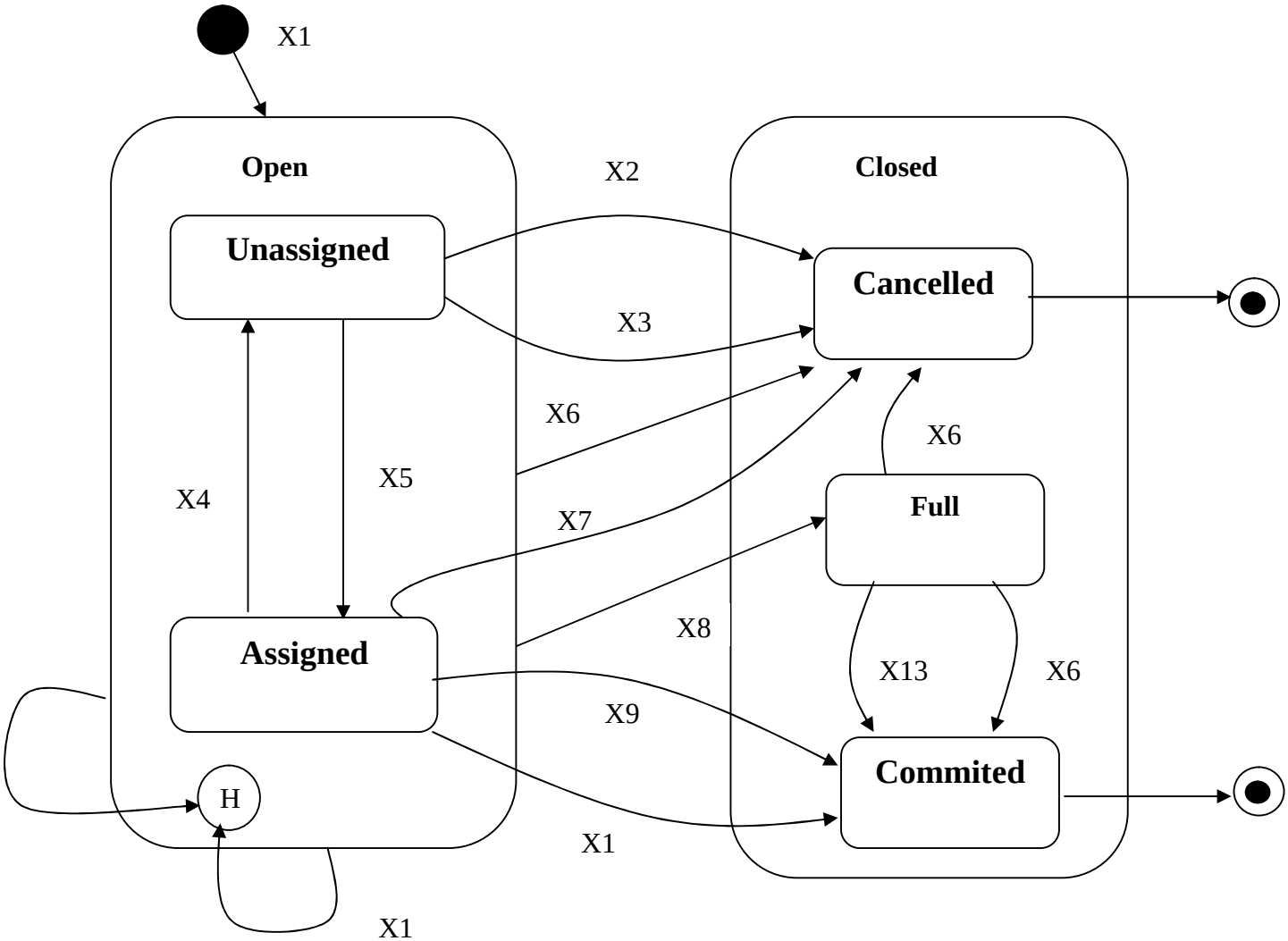


ნახ.4.12. გადასვლები მდგომარეობებს შორის

CourseOfering ობიექტის ქცევის შემდგომი დეტალიზებას მივყავართ მდგომარეობის დიაგრამის აგებამდე (ნახ.4.13). მოცემულ დიაგრამაზე გამოყენებულია მოდელირების ისეთი შესაძლებლობები, როგორიც არის კომპოზიციური (composite state) და ისტორიული მდგომარეობები (history state). მოცემულ შემთხვევაში კომპოზიციური მდგომარეობებია Open და Closed, ხოლო ჩართული მდგომარეობებია – Unassigned, Assigned, Cancelled (კურსი გაუქმდა), Full (კურსი შევსებულია) და Committed (კურსი ჩართულია განრიგში). კომპოზიციურ მდგომარეობები საშუალებას იძლევიან გავამარტივოთ დიაგრამა, შევამციროთ რა გადასვლების რაოდენობა, რამდენადაც ჩართული მდგომარეობები მემკვიდრეობით იძენენ კომპოზიციური მდგომარეობების ყველა თვისებებსა და გადასვლებს.

ისტორიული მდგომარეობა – ეს პსევდომდგომარეობაა, რომელიც აღადგენს წინამდებარე აქტიურ მდგომარეობას კომპოზიტურ მდგომარეობაში. იგი საშუალებას აძლევს კომპოზიტურ მდგომარეობას Open დაიმახსოვროს რომელი ჩართული მდგომარეობა (Unassigned ან

Assigned) იყო მიმდინარე Open – დან გამოსვლის მომენტში, იმისათვის რომ ნებისმიერი გადასვლა Open – ში (add student ან remove student) ბრუნდებოდეს სწორედ ამ ჩართულ მდგომარეობაში, და არა საწყის მდგომარეობაში.



ნახ.4.13. მდგომარეობის დიაგრამა კომპოზიციური მდგომარეობებით

დიაგრამის გადატვირთვის თავიდან ასაცილებლად, ქვემოთ მოყვანილია იმ მოვლენების აღწერა, რომლებიც განაპირობებენ ერთი მდგომარეობიდან გადასვლას მეორეში:

X1 – სტუდენტების რაოდენობა=0;

X2 - რეგისტრაციის დახურვა;

X3 – დახურვა;

X4 - პროფესორის დამატება;

X5 - პროფესორის ამოგდება;

X6 – დამთავრება;

X7 – დახურვა[სტუდენტების რაოდენობა<3];

X8 – სტუდენტების რაოდენობა =10;

X9 – რეგისტრაციის დახურვა[სტუდენტების რაოდენობა>=3];

X10 – დაიხურა {სტუდენტების რაოდენობა>=3};

X11 – სტუდენტის დამატება

/სტუდენტის–რაოდენობა=სტუდენტის–რაოდენობა+1;

X12 – სტუდენტის დამატება

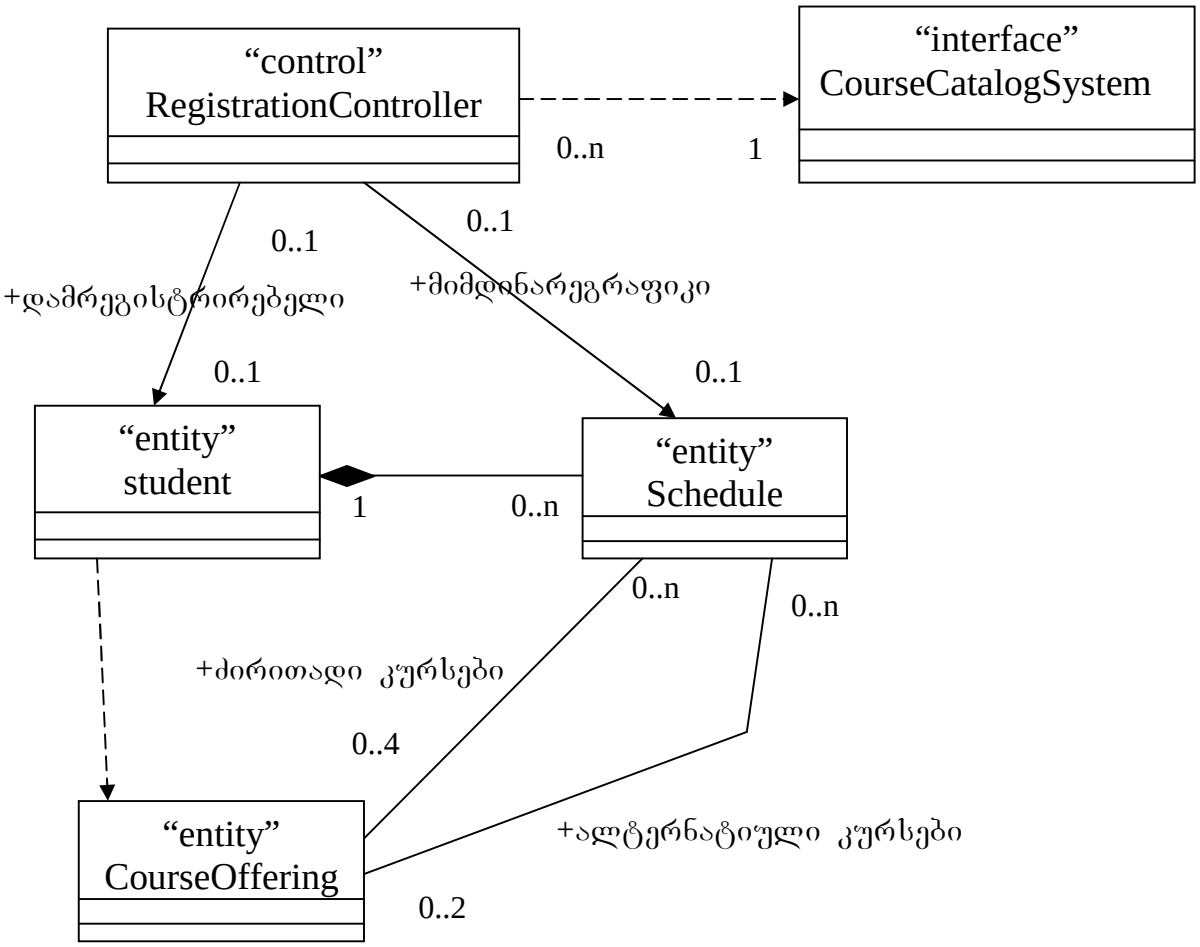
/სტუდენტის–რაოდენობა=სტუდენტის–რაოდენობა+1;

X13 - რეგისტრაციის დახურვა.

4.2.1.4. კლასებს შორის კავშირების დაზუსტება

პროექტირების პროცესში კავშირები კლასებს შორის (ასოციაცია, აგრეგაცია და განზოგადება) მოითხოვს დაზუსტებას:

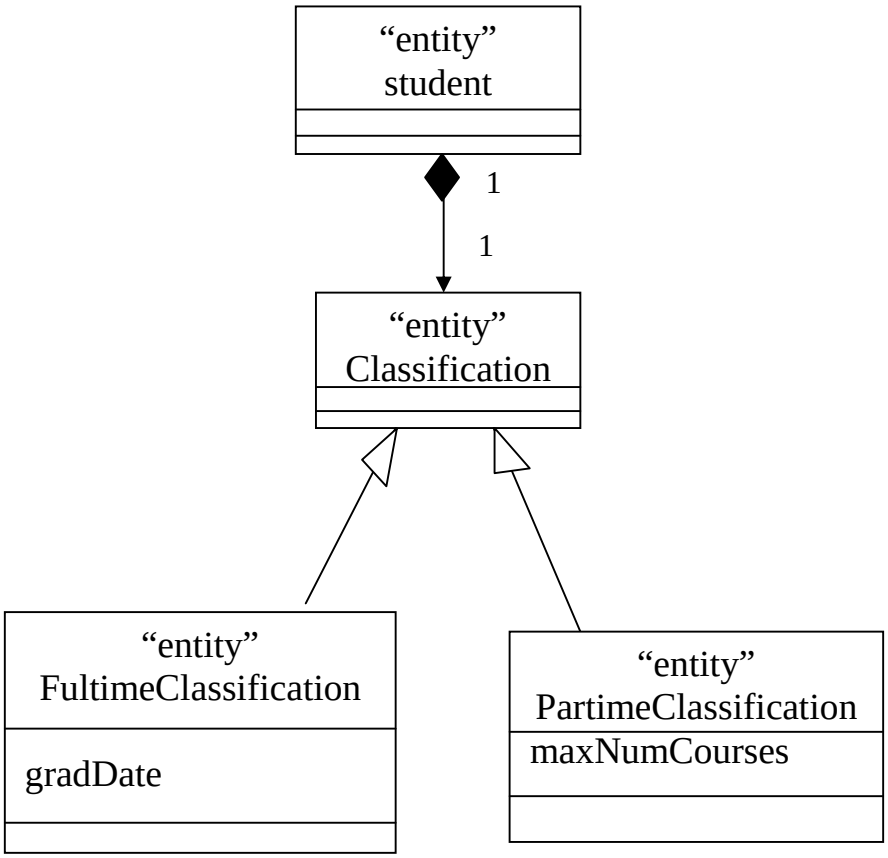
- ასოციაცია მოსაზღვრე და მმართველ კლასებს შორის გამოხატავენ კავშირებს, რომლებიც დინამიურად არიძვრებიან შესაბამის ობიექტებს შორის მართვის ნაკადში. ასეთი კავშირებისათვის საკმარისია უზრუნველვყოთ კლასების ხედვა, ამიტომ ისინი გარდაიქმნებიან დამოკიდებულებაში;
- თუ ზოგიერთი ასოციაციისათვის არ არის აუცილებელი ორმიმართულებიანი კავშირისა, მაშინ შემოიტანება ნავიგაციის მიმართულება;
- აგრეგაციები, რომლებსაც აქვთ კომპოზიციის თვისებები, გარდაიქმნებიან კომპოზიციურ კავშირებში.



ნახ. 4.14. ასოციაციებისა და აგრეგაციების გარდაქმნის მაგალითი

კავშირების გარდაქმნის მაგალითი მოყვანილი რეკომენდაციების შესაბამისად კლასებისათვის გამოყენებითი შემთხვევიდან „კურსებზე დარეგისტრირება“ მოყვანილია ნახ.4.14.-ზე. ასოციაცია მმართველ და მოსაზღვრე კლასებს შორის გარდაიქმნა დამოკიდებულებათ. აგრეგაცია კლასებს შორის Student და Schedule ფლობს კომპოზიციის თვისებას. ნავიგაციის მიმართულება ასოციაციებზე კლასებს შორის Schedule და CourseOffering შემოტანილია შემდეგი მოსაზრებებით: არ არის აუცილებელი გრაფიკების ცხრილის მიღება, რომლებშიც არის რომელიმე კურსი; გრაფიკების რაოდენობა შედარებით ცოტაა კონკრეტული კურსების რაოდენობასთან შედარებით.

შენიშვნა. მნიშვნელობა By Value გულისხმობს, რომ მთელი ნაწილი იქმნებიან და ისპობიან ერთდროულად, რაც შეესაბამება კომპოზიციას. აგრეგაცია (By Reference) გულისხმობს, რომ მთელი ნაწილი იქმნებიან და ისპობიან სხვადასხვა დროს.



ნახ. 4.15. განზოგადების გარდაქმნა

განზოგადების კავშირები შესაძლებელია გარდაიქმნან მეტამორფოზული ქვეტიპების სიტუაციებში. მაგალითად, რეგისტრაციის სისტემის შემთხვევაში სტუდენტს შეუძლია გადავიდეს დღის სწავლებიდან საღამოზე ე.ი. ობიექტ Students შეუძლია შეცვალოს თავისი ქვეტიპი. ასეთი ცვლილებისას მოგვიხდება ობიექტის აღწერის მოდიფიცირება სისტემაში. იმისათვის, რომ თავი ავარიდოთ ასეთ მოდიფიკაციას და ამით გავზარდოთ სისტემის მდგრადობა, მემკვიდრეობის იერარქია რეალიზდება კლასიფიკაციის საფუძველზე.

4.2.2. მონაცემთა ბაზების დაპროექტება

მონაცემთა ბაზების დაპროექტება დამოკიდებულია იმაზე, თუ რა მბმს გამოიყენება მონაცემთა შენახვისათვის – ობიექტური თუ რელაციური. ობიექტური მბ არავითარი დაპროექტება არ არის საჭირო, რამდენადაც კლასი არსები უშუალოდ აისახებიან მბ-ში. რელაციური მბ კლასი არსები ობიექტური მოდელისა უნდა გამოსახულნი იქნან რელაციურ მბ-ში. ცხრილების და მათ შორის კავშირების ერთობლიობა შესაძლებელია წარმოდგენილი იქნან კლასების დიაგრამის სახით, რომელიც წარმოადგენს ფაქტიურად ER – დიაგრამას. წესების ერთობლიობა, რომლებიც მიიღება კლასების მბ-ს ცხრილებში გამოსახვისას, ფაქტიურად ეთანხმებიან მბ-ს ცხრილებში არსებისა და კავშირების გარდაქმნის წესებს. RUP ტექნოლოგიაში ასეთი გამოსახვისათვის გამოიყენება სპეციალური ინსტრუმენტი – Data Modeler. იგი ასრულებს გარდაქმნას კლას-არსებისა კლას-ცხრილებში მისი შემდომი გენერაციით მბ აღწერისა როგორც სტანდარტულ ენაზე SQL (ANSI SQL), ასევე მის დიალექტებზე სხვადასხვა მბმს (Oracle, IBM DB2, Sybase Adaptive Server, MS SQL Server). მბ-ს სქემის აღწერისათვის გამოიყენება UML ენის შემდეგი ელემენტების ნაკრები თავიანთი სტერეოტიპებით:

- ცხრილი წარმოიდგინება კლასის სახით სტერეოტიპით <<Table>>;
- წარმოდგენა გამოისახება კლასის სახით სტერეოტიპით <<View>>;
- ცხრილის სვეტი წარმოიდგინება კლასის ატრიბუტის სახით შესაბამისი მონაცემთა ტიპით;
- ჩვეულებრივი ასოციაცია და აგრეგაცია წარმოიდგინება ასოციაციის სახით სტერეოტიპით <<Non-Identifying>>;
- კომპოზიცია წარმოიდგინება ასოციაციის სახით სტერეოტიპით << Identifying>>;

- მბ-ის სქემა წარმოიდგინება პაკეტის სახით სტერეოტიპით <<Schema>>, რომელიც შეიცავს კლასებს-ცხრილებს;
- შესანახი პროცედურების კონტეინერი წარმოიდგინება კლასის სახით სტერეოტიპით <<SP Container>>;
- მთლიანობის შეზღუდვა, ინდექსები და ტრიგერები წარმოიდგინება კლასი-ცხრილების ოპერაციის სახით სტერეოტიპით <<PK>>, <<FK>>, U<<Unique>>, <<Check>>,<<Index>> და <<Trigger>>;
- ფიზიკური მონაცემთა ბაზა წარმოიდგინება კომპონენტის სახით სტერეოტიპით <<Database>>.

მაგალითისათვის ქვევით მოყვანილია რელაციური ბაზის შექმნის თანმიმდევრობა Rational Rose სისტემაში.

რელაციური ბაზის დაპროექტება

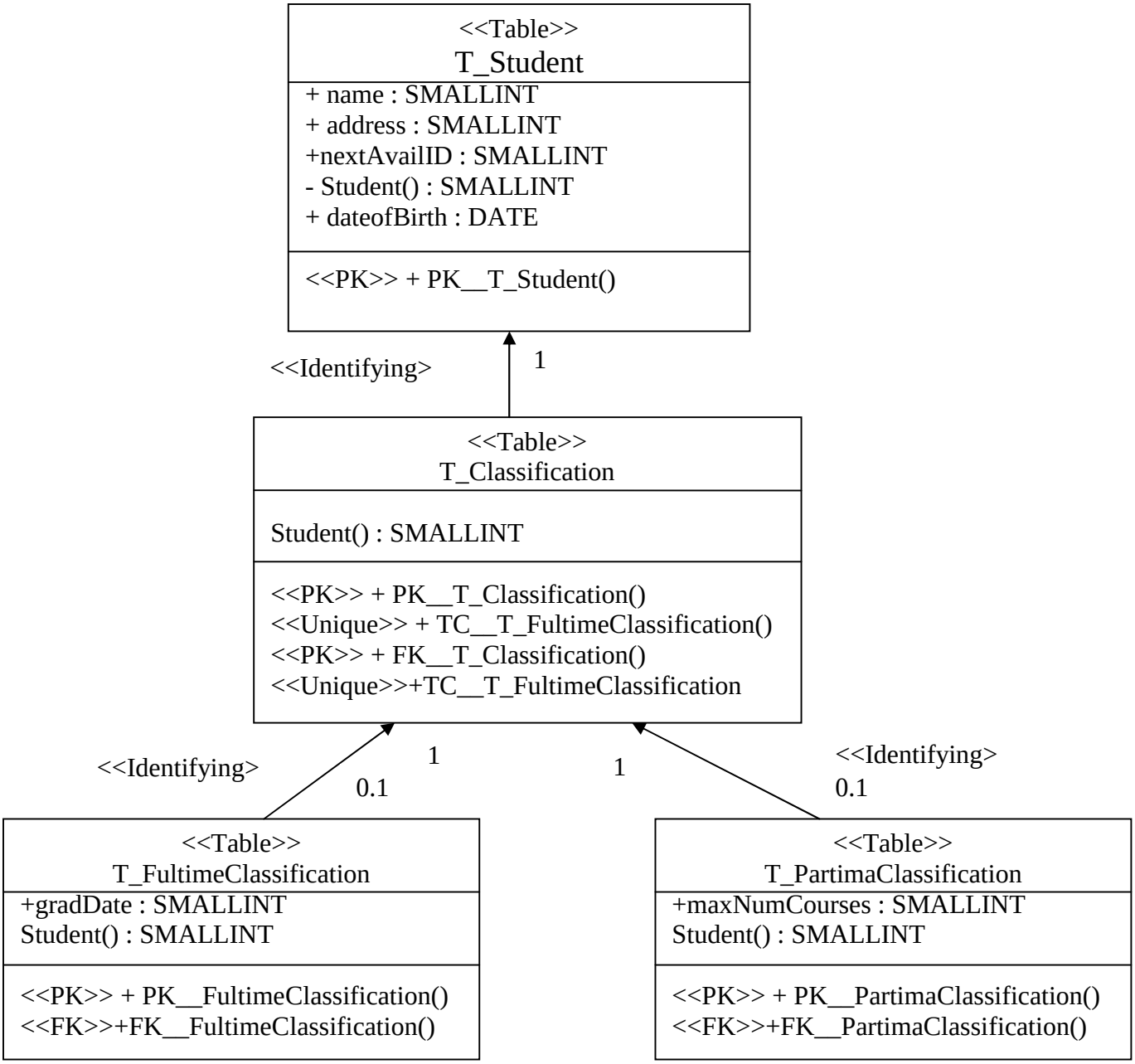
მბ დაპროექტება შედგება შემდეგი ბიჯებისაგან.

ბიჯი 1. მონაცემთა ბაზის – ახალი კომპონენტის შექმნა;

1. დავაწკაპუნოთ მარჯვენა ღილაკით კომპონენტების წარმოდგენაზე.
2. ავირჩიოთ პუნქტი Data Modeler > new > Database გახსნილ მენიუში.
3. გავხსნათ სპეციფიკაციის ფანჯარა ახლად შექმნილი კომპონენტის DB_0 და სიაში Target ავირჩიოთ Oracle *.

ბიჯი 2. მდგრადი კლასების განსაზღვრა:

1. გავხსნათ სპეციფიკაციის ფანჯარა კლასის Student პაკეტში University Artifacts.
2. გადავიდეთ ჩანართზე „Detail“.
3. დავაყენოთ გადამრთველის მნიშვნელობა Persistence Persistent-ზე.
4. ჩავატაროთ ასეთივე მოქმედებები სხვა კლასებზე.



ნახ.4.16. დიაგრამა „არსი-კავშირი“

4. გავხსნათ კლასი Student ბრაუზერში, დავაჭიროთ „+“.
5. დავაწკაპუნოთ მარჯვენა ღილაკით ატრიბუტზე სტუდენტი .
6. ავირჩიოთ პუნქტი Data Modeler > Part of Object Identity გახსნილ მენიუში.

ბიჯი 3. მბ-ს სქემის შექმნა:

1. დავაწკაპუნოთ მარჯვენა ღილაკით პაკეტზე University Artifacts.

2. ავირჩიოთ პუნქტი Data Modeler > Transform to Data Model გახსნილ მენიუში.

3. მიუთითედ DB_0 და დავაწკაპუნოთ ღილაკზე OK გახსნილ ფანჯარაში სიაში Target Database. შედეგად ლოგიკურ წარმოდგენაში გაჩნდება ახალი პაკეტი Schemas.

4. გავხსნათ პაკეტი Schemas და დავაწკაპუნოთ მარჯვენა ღილაკით პაკეტზე <<Schema>> S_0.

5. ავირჩიოთ პუნქტი Data Modeler > New > Data Model Diagram გახსნილ მენიუში.

6. გავხსნათ პაკეტი, შემდეგ ახლად შექმნილი დიაგრამა „არსი-კავშირი“ NewDiagram და გადავიტანოთ მასზე ყველა კლასი-ცხრილები, რომლებიც იმყოფებიან პაკეტში <<Schema>> S_0. მიღებული დიაგრამა ნაჩვენებია ნახ.4.16.-ზე.

მონაცემთა ბაზის დაპროექტების შემდეგ შესაძლებელია Rose სისტემით მონაცემთა ბაზის აღწერის გენერირება SQL ენაზე და დანართის კოდის გენერირება ერთ-ერთ ობიექტ-ორიენტირებულ ენაზე.

მაგალითისათვის ქვევით მოყვანილია მონაცემთა ბაზის აღწერის გენერირება SQL ენაზე და დანართის კოდის გენერირება ერთ-ერთ ობიექტ-ორიენტირებულ ენაზე Rational Rose სისტემაში.

5. კოდის გენერაცია

5.1. მონაცემთა ბაზის აღწერის გენერაცია SQL ენაზე

მბ აღწერის გენერაციისათვის:

7. დავაწკაპუნოთ მარჯვენა ღილაკით პაკეტზე <<Schema>>S_0.
8. ავირჩიოთ პუნქტი Data Modeler > Forward Engineer გახსნილ მენიუში.
9. დავაწკაპუნოთ ღილაკზე Next ოსტატის გახსნილ ფანჯარაში „Forward Engineering Wizard“

10. მონაცემთა აღწერის ენის (DDL) გენერაციის ყველა დროშები დავტოვოთ მონიშნული და დავაწკაპუნოთ ღილაკზე Next.
11. მიუთითოდ დასახელება და ტექსტური ფაილის განლაგება გენერაციის შედეგებით და დავაწკაპუნოთ ღილაკზე Next.
12. გენერაციის დამთავრებისთანავე, გახსენით შექმნილი ტექსტური ფაილი და ნახეთ შედეგები.

გენერაციის შედეგები უნდა გამოიყურებოდნენ შემდეგნაირად:

```
    CREATE TABLE T_ParttimeClassification (
    maxNumCourses SMALLINT NOT NULL,
    studentID SMALLINT NOT NULL,
    CONSTRAINT PK_T_ParttimeClassification29 PRIMARY KEY
(studentID)
    )
    /
    CREATE TABLE T_Classification (
    studentID SMALLINT NOT NULL,
    CONSTRAINT PK_T_Classification23 PRIMARY KEY
(studentID),
    CONSTRAINT TC_T_Classification44 UNIQUE (studentID)
    )
    /
```

```
    CREATE TABLE T_FulltimeClassification (
    gradDate SMALLINT NOT NULL,
    studentID SMALLINT NOT NULL,
    CONSTRAINT PK_T_FulltimeClassification28 PRIMARY KEY
(studentID)
    )
    /
    CREATE TABLE T_Student (
    name SMALLINT NOT NULL,
    address SMALLINT NOT NULL,
    nextAvailID SMALLINT NOT NULL,
    studentID SMALLINT NOT NULL,
    dateofBirth DATE NOT NULL,
    CONSTRAINT PK_T_Student25 PRIMARY KEY (studentID)
    )
    /
    CREATE INDEX TC_T_Classification43 ON T_Classification
(studentID)
    /
    ALTER TABLE T_ParttimeClassification ADD (CONSTRAINT
FK_T_ParttimeClassification29 FOREIGN KEY (studentID) REFER-
ENCES T_Classification (studentID))
    /
    ALTER TABLE T_Classification ADD (CONSTRAINT
FK_T_Classification23 FOREIGN KEY (studentID) REFERENCES
T_Student (studentID))
    /
    ALTER TABLE T_FulltimeClassification ADD (CONSTRAINT
FK_T_FulltimeClassification28 FOREIGN KEY (studentID) REFER-
ENCES T_Classification (studentID))
    /
```

5.2. დანართის კოდის გენერაცია

დანართის კოდის გენერაციის პროცესი შედგება შემდეგი ბიჯებისაგან:

ბიჯი 1. მოდელის გასინჯვა.

ბიჯი 2. კომპონენტების შექმნა.

ბიჯი 3. კლასების შესაბამისობა კომპონენტებთან.

ბიჯი 4. გენერაციის კოდის თვისებების დაყენება.

ბიჯი 5. კლასის, კომპონენტის ან პაკეტის ამორჩევა.

ბიჯი 6. კოდის გენერაცია.

მოდელის გასინჯვა

Rose-ში არსებობს მოდელის შემოწმების საშუალება, რომელიც ენისაგან დამოუკიდებლად გამოიყენება მოდელის კორექტულობის უზრუნველსაყოფად კოდის გენერაციის წინ. რეკომენდირებულია ყოველთვის ასეთი შემოწმების შესრულება, რამდენადაც იგი გვეხმარება გამოვავლინოთ მოდელში უზუსტობანი და შეცდომები, რომლებიც არ გვაძლევენ სათანადოთ კოდის გენერირების საშუალებას. მოდელის შემოწმებისათვის:

1. ავირჩიოთ მენიუში Tools > Check Model.
2. გავაანალიზოთ ყველა შეცდომები, რომლებიც გამოჩნდება ჟურნალის ფანჯარაში.

ყველაზე გავრცელებული შეცდომებია, მაგალითად, შეტყობინება მიმდევრობის ან კოოპერაციის დიაგრამაზე არ არის გამოსახული ოპერაციაზე, ან ამ დიაგრამის ობიექტები, არ არის ასახული კლასზე.

მენიუს პუნქტით Check Model შეიძლება გამოვავლინოთ უზუსტობისა და შეცდომების უმეტესი ნაწილი მოდელში. მენიუს პუნქტი Access Violations საშუალებას იძლევა დავადგინოთ დარღვევები მიმართვის წესებში, რომლებიც არიძვრება მაშინ როდესაც არსებობს კავშირი ორ კლასს

შორის სხვადასხვა პაკეტიდან, მაგრამ კავშირი თვით პაკეტებს შორის არ არის.

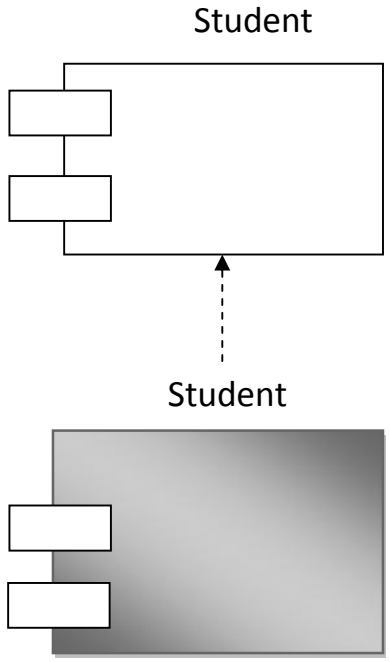
იმისათვის, რომ აღმოვაჩინოთ დარღვევები მიმართვის წესებში:

1. ავირჩიოთ მენიუში Report > Show Access Violations.
2. გავაანალიზოთ ყველა შეცდომები, რომლებიც გამოჩნდება ჟურნალის ფანჯარაში.

კომპონენტების შექმნა და კლასების შესაბამისობა კომპონენტებთან

Rose-ში კომპონენტების დიაგრამები იქმნება კომპონენტების წარმოდგენაში. ცალკეული კომპონენტები შეიძლება შევქმნათ უშუალოდ დიაგრამაზე ან გადავიტანოთ იქ ბრაუზერიდან.

ავირჩიოთ დაპროგრამების ენათ C++ და შევქმნათ კლასისათვის Student ამ ენისათვის შესაბამისი კომპონენტები.



ნახ.6.4. კომპონენტების დიაგრამა

კომპონენტების დიაგრამის შესაქმნელად:

1. დავაწკაპუნოთ ორჯერ კომპონენტების მთავარ დიაგრამაზე Main კომპონენტების წარმოდგენაში.
2. დავაწკაპუნოთ ღილაკზე Paskage Specification ინსტრუმენტების პანელზე.
3. მოვათავსოთ ელემენტი Paskage Specification დიაგრამაზე.
4. გავხსნათ ელემენტის სპეციფიკაცია, შევიტანოთ სახელი Student და მიუთითოდ ფანჯარაში ენა ANSI C++.
5. დავაწკაპუნოთ ღილაკზე Paskage Body ინსტრუმენტების პანელზე.
6. მოვათავსოთ ელემენტი Paskage Body დიაგრამაზე.
7. გავხსნათ ელემენტის სპეციფიკაცია, შევიტანოთ სახელი Student და მიუთითოდ ფანჯარაში ენა ANSI C++.
8. დავაწკაპუნოთ ღილაკზე Dependence ინსტრუმენტების პანელზე.
9. გავატაროთ დამოკიდებულების ხაზი პაკეტის ტანიდან პაკეტის სპეციფიკაციასთან.

კლასის კომპონენტთან შესაბამისობისათვის:

1. ვიპოვოთ კლასი Student ბრაუზერის ლოგიკურ წარმოდგენაში.
2. გადავიტანოთ ეს კლასი პაკეტის კომპონენტ Student სპეციფიკაციაზე ბრაუზერის კომპონენტების წარმოდგენაში. შედეგად კლასი Student შეუსაბამდება სპეციფიკაციას და პაკეტ კომპონენტ Student-ის ტანს.

კოდის გენერაცია

ყოველი ენისათვის Rose-ში გათვალისწინებულია კოდის გენერაციის განსაზღვრული თვისებები. კოდის გენერაციის წინ რეკომენდირებულია გავაანალიზოთ ეს თვისებები და შევიტანოთ აუცილებელი ცვლილებები.

კოდის გენერაციის თვისებების ანალიზისათვის ავირჩევთ Tools > Options, ხოლო შემდეგ შესაბამისი ენის ჩანართი. სიის ფანჯარაში

შეიძლება ავირჩიოთ კლასი, ატრიბუტი, ოპერაცია და მოდელის სხვა ელემენტები. ყოველი ენისათვის ამ სიაში მითითებულია მოდელის თავისი საკუთარი ელემენტები. სხვადასხვა მნიშვნელობების არჩევისას ეკრანზე გამოჩნდება თვისებათა სხვადასხვა ნაკრები. ნებისმიერი ცვლილებები, რომლებიც შეიტანება თვისებათა ნაკრებში ფანჯარაში Tools > Options, იმოქმედებენ მოდელის ყველა ელემენტზე, რომლებისთვისაც გამოიყენება მოცემული ნაკრები.

კოდის გენერაციისას Rose ირჩევს ინფორმაციას მოდელის ლოგიკური და კომპონენტური წარმოდგენებიდან და ახდენს დიდი მოცულობის „ჩონჩხი“ კოდის გენერირებას:

- კლასები. გენერირდება მოდელის ყველა კლასები.
- ატრიბუტები. კოდი მოიცავს ყოველი კლასის ატრიბუტებს, მათ შორის ხედვას, მონაცემის ტიპს და მნიშვნელობას დუმილით.
- ოპერაციის სიგნატურები. კოდი მოიცავს ოპერაციის განმარტებას მთელი პარამეტრებით, მოცემული პარამეტრების ტიპებით და ოპერაციის დასაბრუნებელი მნიშვნელობის ტიპით.
- კავშირები. მოდელის ზოგიერთი კავშირები იწვევენ ატრიბუტების შექმნას კოდის გენერაციისას.
- კომპონენტები. ყოველი კომპონენტი რეალიზდება შესაბამისი ფაილის სახით საწყისი კოდით.

C++ კოდის გენერაციისათვის:

1. გახსენით სისტემის კომპონენტების დიაგრამა.
2. ამოირჩიეთ ყველა ობიექტები კომპონენტების დიაგრამაზე.
3. ამოვირჩიოთ Tools > ANSI C++ > Generate kode მენიუში.
4. ავირჩიოთ კატალოგი კოდის გენერაციისათვის.
5. დავაწკაპუნოთ ღილაკზე OK და შეასრულეთ გენერაცია კოდის გენერაციის ფანჯარაში.

6. ვნახოთ გენერაციის შედეგები (მენიუ Tools > C++ Browse Header და Tools > C++ > Browse Body).

გენერაციის შედეგები უნდა გამოიყურებოდეს შემდეგნაირად:

Data Student.h:

```
#ifndef STUDENT_H_HEADER_INCLUDED_BE29599B
#define STUDENT_H_HEADER_INCLUDED_BE29599B

//##ModelId=35A6336C03DE
class Student
{
public:
//##ModelId=360EBEFA015E
double getTuition();

//##ModelId=374AFB93006B
addSchedule(Schedule theSchedule);

//##ModelId=374AFBDA0117
Schedule getSchedule(Semester forSemester);

//##ModelId=374B00540183
deleteSchedule(Semester forSemester);

//##ModelId=374B02690049
boolean hasPrerequisites(CourseOffering forCourseOffering);

//##ModelId=37812F3903C4
int getNextAvailID();

//##ModelId=379779F60364
int getStudentID();

//##ModelId=37BE859E00CA
string getName();
```

```
//##ModelId=37BF215800D8
string getAddress();

protected:
//##ModelId=37812764010E
boolean passed(CourseOffering theCourseOffering);

private:
//##ModelId=35E9A8EA00BE
string name;

//##ModelId=374D92DE019A
string address;

//##ModelId=37812F2301D8
int nextAvailID;

//##ModelId=378130B900EB
int studentID;

//##ModelId=378BE6A5015D
Date dateofBirth;

};

#endif /* STUDENT_H_HEADER_INCLUDED_BE29599B */
```

Datei Student.cpp:

```
#include "Student.h"

//##ModelId=360EBEFA015E
double Student::getTuition()
{
}

//##ModelId=374AFB93006B
Student::addSchedule(Schedule theSchedule)
{
}
```

```
//##ModelId=374AFBDA0117
Schedule Student::getSchedule(Semester forSemester)
{
}

//##ModelId=374B00540183
Student::deleteSchedule(Semester forSemester)
{
}

//##ModelId=374B02690049
boolean Student::hasPrerequisites(CourseOffering forCourseOffering)
{
}

//##ModelId=37812F3903C4
int Student::getNextAvailID()
{
}

//##ModelId=379779F60364
int Student::getStudentID()
{
}

//##ModelId=37BE859E00CA
string Student::getName()
{
}

//##ModelId=37BF215800D8
string Student::getAddress()
{
}

//##ModelId=37812764010E
boolean Student::passed(CourseOffering theCourseOffering)
{
}
```

6. საკურსო პროექტის თემები

6.1. სახელფასო ანაზღაურების დარიცხვის სისტემა

კომპანიის საინფორმაციო სამსახურის წინაშე დასმულია ამოცანა შექმნას სახელფასო ანაზრაურების ახალი სისტემა. ახალი სისტემა უნდა საშუალებას აძლევდეს მოსამსახურეებს ელექტრონული საშუალებით ჩაწეროს ინფორმაცია სამუშაო დროის აღრიცხვის ბარათიდან და ავტომატურად მოახდინოს გადახდის ქვითრების ფორმირება, რომლებიც ითვალისწინებენ გამომუშავებულ საათებს და გაყიდვების საერთო მოცულობას (იმ მოსამსახურეებისათვის, რომლებიც იღებენ საკომისიო დაჯილდოებას).

ახალი სისტემა საშუალებას უნდა აძლევდეს მოსამსახურეებს შეიტანონ ინფორმაცია სამუშაო დროის აღრიცხვის ბარათიდან და შეკვეთები პროდუქციის მიღებაზე, შეცვალონ თავისი პარამეტრები(ისეთი, როგორიც არის გადახდის ფორმა სამუშაოსათვის) და მოახდინონ სხვადასხვა ანგარიშები. სისტემა უნდა მუშაობდეს მთელი კომპანიის მოსამსახურეთა პერსონალურ კომპიუტერებზე. უსაფრთხოებისა და აუდიტის უზრუნველყოფის მიზნით მოსამსახურეებს შესაძლებლობა უნდა ჰქონდეთ მხოლოდ თავიანთი საკუთარი სამუშაო დროისა და შეკვეთების სააღრიცხვო ბარათებთან მიმართვისა და რედაქტირების.

სისტემაში უნდა ინახებოდეს ინფორმაცია კომპანიის ყველა მოსამსახურეზე, მათ შორის მათზე, რომლებიც სხვადასხვა ქვეყნებში მუშაობენ. სისტემა უინდა უზრუნველყოფდეს ყოველი მოსამსახურის შრომის სწორ და დროულ ანაზღაურებას მათ მირ მითითებული საშუალების შესაბამისად.

ზოგიერთი მოსამსახურეები ღებულობენ საათობრივ ანაზღაურებას, რომელიც დაირიცხება სამუშაო დროის აღრიცხვის ბარათის საფუძველზე,

რომელთაგან თითოეული შეიცავს თარიღს და საათების რაოდენობას, გამომუშავებული კონკრეტული ტარიფის შესაბამისად. თუ რომელიმე მოსამსახურემ დღეში გამოიმუშავა 8 სთ-ზე მეტი, ნამატი დრო ანაზღაურდება კოეფიციენტით 1,5. მომსახურე-სათობრივები ღებულობენ ხელფასს ყოველ პარასკევს. ზოგიერთი მოსამსახურეები ღებულობენ ფიქსირებულ ანაზღაურებას, მაგრამ ისინიც წარადგენენ სამუშაო დროის თავიანთ სააღრიცხვო ბარათებს. ამის შედეგად სისტემას შეუძლია აწარმოოს საათების რაოდენობის აღრიცხვა, გამომუშავებული კონკრეტული ტარიფების შესაბამისად. ასეთი მოსამსახურეები ღებულობენ ანაზღაურებას თვის ბოლო სამუშაო დღეს.

ზოგიერთი მოსამსახურეები, რომლებიც ღებულობენ ფიქსირებულ ანაზღაურებას, ასევე ღებულობენ საკომისიო ჯილდოებს, რომლებიც გულისხმობენ გაყიდვის მოცულობას. ისინი წარმოადგენენ შეკვეთებს მიწოდებაზე, რომლებიც უჩვენებენ გაყიდვის თარიღს და მოცულობას. საკომისიო ჯილდოების ოდენობა განისაზღვრება ინდივიდუალურად ყოველი მოსამსახურესათვის და შეადგენს 10%, 15, 25 ან 35%-ს.

ახალი სისტემის ყველაზე ხშირათ გამოსაყენებელი შესაძლებლობაა სხვადასხვა ანგარიშების მომზადება: დადგინდეს გამომუშავებული საათების რაოდენობა, ჯამური ანაზღაურება, შვებულების დარჩენილი დღეები და ა. შ.

მოსამსახურეებს შეუძლიათ ამოირჩიონ ანაზღაურების საშუალება სამუშაოსათვის და მიიღონ თავიანთი ანაზღაურების ქვითრები ფოსტით, საბანკო ანგარიშზე ან ხელზე ოფისში.

სისტემის ადმინისტრატორი წარმართავს ინფორმაციას მოსამსახურეების შესახებ. მის მოვალებაში შედის მონაცემების შეტანა ახალი მოსამსახურეების შესახებ, მონაცემების გამორიცხვა და ნებისმიერი ინფორმაციის შეცვლა მოსამსახურის შესახებ, როგორც არის, დასახელება,

მისამართი და ანაზღაურების საშუალება, ასევე სხვადასხვა ანგარიშების მომზადება ხელმძღვანელობისათვის.

დანართი შრომითი ანაზღაურების დარიცხვის შესახებ გაიშვება ავტომატურად ყოველ პარასკევს და თვის საბოლაო შრომით დღეს შესაბამისი მოსამსახურეების მოცემულ დღეებში შრომითი ანაზღაურებისათვის. ანაზღაურება უნდა ირიცხებოდეს ავტომატურად, ხელით ჩარევის გარეშე.

6.2. ინტერნეტ-მაღაზია

კომპიუტერების მწარმოებელი სთავაზობს შეიძინოთ თავისი პროდუქცია ინტერნეტის მეშვეობით. კლიენტს შეუძლია ამოირჩიოს კომპიუტერი მწარმოებლის Web-გვერდზე. კომპიუტერები იყოფიან სერვერებათ, სამაგიდო და პორტატიულებათ. შემკვეთს შეუძლია აირჩიოს სტანდარტული კონფიგურაცია ან ააგოს მისთვის საჭირო დიალოგურ რეჯიმში. კონფიგურაციის კომპონენტები(მაგ. ოპერატიული მეხსიერება) წარმოიდგინებიან ცხრილის სახით დასაშვები ალტერნატივებიდან ამოსარჩევათ. სისტემის ყოველი ახალი კონფიგურაციისათვის შესაძლებელია ფასის გამოთვლა.

იმისათვის, რომ გაფორმდეს შეკვეთა, კლიენტმა უნდა შეავსოს ინფორმაცია მიწოდებაზე და გადახდაზე. საგადასახადო საშუალებათ შესაძლებელია საკრედიტო ბარათებისა და ქვითრების გამოყენება. შეკვეთის შეტანის შემდეგ სისტემა უგზავნის კლიენტს ელექტრონული ფოსტით შეტყობინებას, რომელიც შეიცავს დასტურს მიღებული შეკვეთის შესახებ. სანამ კლიენტი ელოდება კომპიუტერის მოწოდებას, მას შეუძლია შეამოწმოს შეკვეთის მდგომარეობა ნებისმიერ დროს დიალოგურ რეჟიმში. შეკვეთის დამუშავების სერვერული ნაწილი შედგება დავალებებისაგან, რომლებიც აუცილებელია კრედიტუნარიანობის და საყიდელზე გადახდის

ფორმის, საწყობიდან შეკვეთილი კონფიგურაციის გამოთხოვის, ანგარიშის ბეჭედისა და კლიენტის კომპიუტერის საწყობში შეტანის მოხსენების შესამოწმებლათ.

დამატებითი მოთხოვნები:

- ასარჩევი სერვერის, სამაგიდო ან პორტატიულ სტანდარტულ კონფიგურაციასთან გაცნობისათვის კლიენტი სარგებლობს ინტერნეტ-მაღაზიის Web-გვერდით;
- კლიენტი ირჩევს კონფიგურაციის დეტალებს, რომლებთანაც მას სურს გაცნობა და შესაძლებელია იყიდოს მზა ან შეადგინოს უფრო მისაღები კონფიგურაცია. ფასი ყოველი კონფიგურაციისათვის შესაძლებელია დათვლილ იქნას მომხმარებლის მოთხოვნის შესაბამისად;
- კლიენტს შეუძლია ამოირჩიოს შეკვეთის ვარიანტი ინტერნეტიდან, მოითხოვოს, რომ გამყიდველი დაუკავშირდეს მას შეკვეთის დეტალების ახსნისათვის, შეუთანხმდეს ფასზე, სანამ შეკვეთა ფაქტიურად იქნება განლაგებული;
- შეკვეთის განლაგებისათვის კლიენტმა უნდა შეავსოს ელექტრონული ფორმა საქონლის მიწოდების და ანგარიშ-ფაქტურის გაგზავნის მისამართებით, ასევე დეტალებით, რომლებიც ეხება გადახდას(საკრედიტო ბარათი ან ქვითარი);
- კლიენტის შეკვეთის სისტემაში შეტანის შემდეგ გამყიდველი აგზავნის საწყობში ელექტრონულ მოთხოვნას, რომელიც შეიცავს შეკვეთილი კონფიგურაციის დეტალურ აღწერას;
- შეთანხმების დეტალები, შეკვეთის ნომრისა და კლიენტის ანგარიშის ჩართვით, ეგზავნება ელექტრონული ფოსტით კლიენტს, ისე, რომ შემკვეთს შეუძლია გასინჯოს შეკვეთის მდგომარეობა ინტერნეტით;

- საწყობი ღებულობს ანგარიშ-ფაქტურას გამყიდველისაგან და გადაუტვირთავს კომპიუტერს კლიენტს.

6.3. უმაღლესი სასწავლებლის ფარგლებში დასაქმების სამსახური

სისტემა განკუთვნილია იმისათვის, რომ დავეხმაროთ სტუდენტს მოეწყოს სამსახურში უმაღლეს სასწავლებელში მისი სწავლის პროცესში. შეაქვს რა განცხადება სისტემაში, სტუდენტი ხდება მისი კლიენტი და იწყება მისი მომსახურება უმაღლეს სასწავლებელში მისი სწავლის განმავლობაში. განცხადება წარმოაგენს ანკეტას. სისტემა ახდენს პროფესიონალურ ტესტირებას, რომელიც ტარდება რეგულარულად(სემესტრში ერთხელ). განსაკუთრებული ყურადღება ეთმობა სტუდენტის სწავლებას, მოსწრების შედეგების მიხედვით დგება ექსპერტული შეფასებები. მოპოვებული ინფორმაციის საფუძველზე დგება რეზიუმე, რომელიც წარმოადგენს სტუდენტის სრულ დახასიათებას. ეს რეზიუმე ეგზავნება ყველა ორგანიზაციას, რომლებსაც გააჩნია ვაკანსია.

სისტემის ძირითადი დანიშნულებაა სტუდენტების შესახებ საანგარიშო მონაცემების შეტანის და შენახვის ავტომატიზაცია, დახასიათებებისა და რეზიუმების მომზადების ავტომატიზაცია, ფირმაში ვაკანსიების პოვნა. სისტემა საშუალებას იძლევა შევცვალოთ, დავამატოთ, განვახორციელოთ ძებნა და ინფორმაციის ნახვა სტუდენტების შესახებ, უზრუნველყოს მიმართვის შეზღუდვა სისტემისადმი, შეინახოს სტუდენტთა სიები, რომლებმაც დაამთავრეს სწავლა არქივის სახით, დაუკავშიროს ინსტიტუტი ფირმებს, რომლებიც დაინტერესებულნი არიან ახალი თანამშრომლების შეძენით.

ასევა მოცემული სისტემა შესაძლებელია გამოყენებულ იქნას ცალკეული ჯგუფების საგამოცდო უწყისებისა და მთლიანად მონაცემთა ბაზის ბეჭდვისა და სტატისტიკისათვის.

სისტემა შედგება ოთხი ქვესისტემისაგან:

- სტუდენტთა მოსწრების კონტროლი;
- პროფესიონალური და ფსიქოლოგიური ტესტებისაგან;
- მომხმარებელთა უფლებების კატეგორიის განსაზღვრის დაკვეთების დამუშავება;
- ექსპერტული შეფასებები.

1. სტუდენტთა მოსწრების კონტროლი. იგი პასუხს აგებს ცალკეული სტუდენტების, ჯგუფის ან მთლიანად ფაკულტეტის მოსწრების სტატისტიკურ ანგარიშგებაზე, ასევე მათი შენახვისა და შეტანის სისწორეზე.

ქვესისტემისათვის შემავალი მონაცემებია: შეფასებები, გამოცდების ჩაბარების თარიღები, სტუდენტების სახელები, ჯგუფის ნომრები, ფაკულტეტი. გამოსასვლელზე ქვესისტემა გამოსცემს დამუშავებულ მონაცემებს: საშუალო ბალი სტუდენტის, ჯგუფის ან ფაკულტეტის მიხედვით, პროცენტული შეფარდება შეფასებებისა სტუდენტისა ჯგუფში ან ფაკულტეტზე, დასახელება და სტიპენდიანტების რაოდენობა ჯგუფში ან ფაკულტეტზე. ქვესისტემა «სტუდენტების მოსწრების კონტროლი» შესაძლებელია ფუნქციონირებდეს მთელი სისტემისაგან დამოუკიდებლად, რაც იძლევა საშუალებას დავაყენოთ და გამოვიყენოთ იგი დამოუკიდებლად, თუ ეს აუცილებელია.

ქვესისტემა «სტუდენტების მოსწრების კონტროლი» მოიცავს შემდეგ ფუნქციებს:

- ქვესისტემის ინფორმაციულ ობიექტებს შორის ინფორმაციის შეტანა, გამოტანა და რედაქტირება;
- დამუშავებული ინფორმაციის შენახვა;

- სტუდენტის შეფასებების პროცენტული თანაფარდობის გამოთვლა ჯგუფში, ფაკულტეტზე და მათი გამოყვანა ცხრილების, გრაფიკების ან დიაგრამის სახით;
- საშუალო ქულის გაანგარიშება სტუდენტის, ჯგუფის ან ფაკულტეტისათვის;
- მონაცემების ფორმირება სტუდენტის, ჯგუფის ან ფაკულტეტისათვის;
- ძლიერი და სუსტი სტუდენტების გამოვლენა ჯგუფში ან ფაკულტეტზე;
- სტიპენდიანტი სტუდენტების გამოვლენა ჯგუფში ან ფაკულტეტზე;
- მონაცემთა შეტანის სისწორის კონტროლი.

2. პროფესიონალური და ფსიქოლოგიური ტესტირების დანიშნულებაა მოახდინოს სტუდენტების ტესტირება დამუშავებული ტესტების სფუძველზე.

ტესტირება გამოიძახება მონიტორის მიერ, იღებს მისგან საწყის მონაცემებს და მათგან დამოკიდებულებით ახდენს რეგისტრაციას, ტესტების განსაზღვრას(პროფესიული, ფსიქოლოგიური, სპეციალური) გამოთვლას და შედეგების შენახვას.

3. დაკვეთების დამუშავება. მომხმარებელთა კატეგორიის განსაზღვრა.

იგი განკუთვნილია კატეგორიის. უფლებამოსილების განსაზღვრისათვის და დასაქმების სამსახურის მომხმარებელთა დაკვეთების დასამუშავებლად. კერძოდ იგი ახდენს:

- ახალი ფირმების რეგისტრაციას;
- ახალი სტუდენტების რეგისტრაციას;
- დარეგისტრირებული მომხმარებლის მიმართვის უფლების დადგენას;
- დაკვეთების დამუშავებას;
- სარეგისტრაციო მონაცემების მიღებას ფირმებიდან, სტუდენტებისა და მომსახურე პერსონალისაგან;
- რეზიუმეს შედგენა;

• მონაცემების ჩაწერა სტუდენტების მონაცემთა ბაზაში, ფირმების და დარეგისტრირებულ მომხმარებელთა მონაცემთა ბაზაში (მბ);
შესასრულებელი ფუნქციებიდან გამომდინარე სისტემა მუშაობს შემდეგ მონაცემებთან:

- სტუდენტების და ფირმების რეგისტრირებულ მონაცემებთან;
- სტუდენტების პირად მონაცემებთან;
- სტუდენტების შესახებ ინფორმაციასთან (ღებულობს ფირმა);
- ინფორმაცია ფირმების შესახებ (ღებულობს სტუდენტი);
- მომხმარებელთა საიდენტიფიკაციო მონაცემებთან;
- სისტემის შესახებ ინფორმაციასთან;
- დაკვეთით;
- სამოსამსახურო ინფორმაციასთან(მომსახურე პერსონალისათვის);
- ფსიქოლოგიური და პროფესიული ტესტების შედეგებთან;
- ექსპერტულ შეფასებებთან.

4. ექსპერტული შეფასებები

იგი განკუთვნილია სტუდენტის ან მთლიანად ჯგუფის ექსპერტული შეფასების მიღებისათვის. რითაც პედაგოგს შეუძლია მიიღოს ინფორმაცია სტუდენტის შესახებ. სტუდენტს კი შეუძლია მისი მეშვეობით აიღოს ორიენტაცია სწავლაში.

გარდა ამისა მისი მეშვეობით ფირმის წარმომადგენელს საშუალება ეძლევა მიიღოს წარმოდგენა სტუდენტზე ახალი კადრების შერჩევისას.

6.4. ავიაკომპანიის Web-საითი

ავიაკომპანიის კომერციულმა განყოფილებამ გადაწყვიტა გააფართოვოს თავისი ჭებ-საითი, იმისათვის, რომ მომხმარებლებს შეეძლოს:

- გაიგოს მიმდინარე დღეს შესრულებული რეისების შესახებ;
- შეუკვეთოს ინფორმაცია რეისების განრიგის, ბილეთების ღირებულების და ადგილების არსებობის შესახებ;
- იყიდოს ბილეთები;

ავიაკომპანიის მუდმივ კლიენტებს შეუძლიათ ისარგებლონ აგრეთვე შემდეგი ფუნქციებით:

- მიიღოს ინფორმაცია თავისი პირადი ანგარიშის მიმდინარე მდგომარეობის შესახებ(კილომეტრების რაოდენობა, რომელიც მან გაატარა ჰაერში წლის დასაწყისიდან დღემდე, დაფრენილი კილომეტრების რაოდენობა ჯილდოს მისაღებად(უფასო გადაფრენა და ა.შ.);
- იყიდოს ბილეთები, საკრედიტო ბარათის ან დაფრენილი კილომეტრების შესახებ ინფორმაციის საფუძველზე(მუდმივი კლიენტებისათვის) გამოყენებით.

იმისათვის, რომ გარანტირებული იყოს პირადი ინფორმაციის კონფიდენციალურობა და აღმოიფხვრას მუდმივ კლიენტებზე მონაცემების არასანქცირებული გამოყენება, აუცილებელია მოთხოვნილი იქნას, რომ მომხმარებელი დარეგისტრირდეს პირად ანგარიშებზე მიმართვისას, შეიტანოს ანგარიშის ნომერი და ბარათის მფლობელის პირადი საიდენტიფიკაციო ნომერი(PIN). რეგისტრაციის შემდეგ მომხმარებელმა უნდა დაინახოს საწყისი გვერდი, მიღებული მონაცემთა ბაზიდან, რომელშიც ინახება ინფორმაცია მუდმივი კლიენტების გადაფრენების შესახებ. მუდმივ კლიენტებს შეუძლიათ ოპერატიულად განაახლონ მონაცემები თავის შესახებ.

იმისათვის, რომ მოახდინოს ფულის ეკონომია, კომპანიის ხელმძღვანელობამ გადაწყვიტა გამოიყენოს რიგი არსებული სისტემებისა:

- ანგარიშების მართვის სისტემა, რომელშიც ინახება ინფორმაცია მუდმივი კლიენტების შესახებ;
- მარკეტინგული მონაცემთა ბაზა, რომელშიც ინახება ინფორმაცია შესრულებული რეისების, გადახდის კლასის და სხვათა შესახებ. ეს მონაცემები გამოიყენებიან მუდმივი კლიენტების შესახებ თვიურ ამონაწერისათვის;
- ტარიფების მონაცემთა ბაზა;
- ბილეთების არსებობის მონაცემთა ბაზა.

6.5. სასაწყობო აღრიცხვის სისტემა

საწყობის აღრიცხვის სისტემა – პროგრამული სისტემაა, რომელიც მოიცავს ყველა ასპექტს, რომელიც დაკავშირებულია საქონლის მოძრაობასთან საწყობში და საწყობიდან. ანალიზის შედეგების მიხედვით შესაძლებელია გამოვყოთ სისტემის შვიდი ძირითადი ფუნქცია:

შეკვეთების აღრიცხვა – შეკვეთების მიღება კლიენტებისაგან და შეკვეთების მდგომარეობის შესახებ პასუხების გაცემა კლიენტებზე.

ანგარიშების წარმოება – ანგარისების დაგზავნა კლიენტებზე და გადახდებზე თვალის მიდევნება. მიმწოდებლებისაგან ანგარიშების მირება და გადახდებზე თვალის მიდევნება, მიმართული მიმწოდებლებზე.

საწყობიდან გადატვირთვა – საწყობიდან კლიენტებზე გასაგზავნი საქონლის კომპლექტაციაზე სპეციფიკაციების შედგენა,

სასაწყობო აღრიცხვა – შემოსული საქონლის აღრიცხვა და აღრიცხვიდან მოხსნა შეკვეთების გაგზავნისას.

შესყიდვები – მიმწოდებლებზე საქონლის შეკვეთა და მოწოდებაზე თვალის მიდევნება.

მიღება – მიმწოდებლებისაგან საქონლის მიღება საწყობში.

დაგეგმვა – ანგარიშების მომზადება, მათ შორის საქონლის ცალკეულ სახეობებზე მოხმარების ტენდენციის ასახვა და მომხმარებელთა აქტიურობა.

კომპანიის სტრატეგიიდან გამომდინარე, ახალ ბაზრებზე გასვლიდან გამომდინარე, გადაწყვეტილი იყო შექმნილიყო პროდუქციის რამოდენიმე რეგიონალური შედარებით ავტონომიური საწყობი. ყოველ ასეთ საწყობს აქვს პასუხისმგებლობა საქონლის აღრიცხვასა და შეკვეთების შესრულებაზე. თავისი მუშაობის ეფექტურობის გაზრდისათვის საწყობი ვალდებულია თვითონ დააკავოს საქონლის ის ნომენკლატურა, რომელიც ყველაზე უკეთესად შეესაბამება ადგილობრივი ბაზრის მოთხოვნებს. ნომენკლატურა შესაძლებელია იყოს სხვადასხვა ყოველი რეგიონისათვის და უნდა ოპერატიულად იცვლებოდეს კლიენტთა მოთხოვნების შესაბამისად. სათაო კომპანიას სურს, რომ ყველა საწყობებში ქონდეს აღრიცხვის ერთგვაროვანი სისტემა. სისტემის ძირითადი ფუნქციებია:

- საქონლის აღრიცხვა, რომლებიც შემოდის სხვადასხვა მიმწოდებლებისაგან, საწყობში მათი მიღებისას;
- შეკვეთების აღრიცხვა მათი შემოსვლისას ცენტრალური დაშორებული ორგანიზაციიდან; შეკვეთები შესაძლებელია ასევე მიღებულ იქნას ფოსტით. მათი დამუშავება ხდება ადგილზე.
- პერსონალზე მითითებების გენერაცია, კერძოდ, საქონლის შეფუთვაზე;
- ანგარიშების გენერაცია და გადახდებზე თვალის მიდევნება;
- მოწოდებაზე შეკვეთების გენერაცია და მიმწოდებლების გადახდებზე თვალის მიდევნება.

სტანდარტული სასაწყობო ოპერაციების ავტომატიზაციის გარდა სისტემამ უნდა საშუალება მოგვცეს გამოვიყენოთ ფართო შესაძლებლობები სხვადასხვა ფორმის ანგარიშების გენერაციისათვის, მათ

შორის ისეთების, რომლებიც ასახავენ ბაზრის განვითარების ტენდენციას, ყველაზე სანდო და არასანდო მიმწოდებლებისა და კლიენტების, სარეკლამო კომპანიებისათვის მასალების.

6.6. ვიდეოპროდუქციის გაქირავების მაღაზია

ვიდეოპროდუქციის გაქირავების მაღაზია საჭიროებს აღრიცხვის კომპიუტერულ სისტემაში, რადგან მისი ასორტიმენტი შეადგენს დაახლოებით 1000 ვიდეოკასეტას და 500 ვიდეოდისკს. მარაგი უკვე შეკვეთილია მიმწოდებელთან, მაგრამ დირექტორს განზრახული აქვს ისარგებლოს უფრო მეტი მიმწოდებლების მომსახურებით. ყველა ვიდეოკასეტები და დისკები აღჭურვილია შტრიხ-კოდით, ისე, რომ სკანერს, რომელიც ინტეგრირებულია სისტემაში, შეუძლია განახორციელოს ვიდეოფილმების გაქირავებისა და მიღების ოპერაციები. კლიენტების პირადობის ბარათებიც აღჭურვილია შტრიხ-კოდით.

კლიენტებს შეუძლიათ დაარეზერვირონ ვიდეო ისე, რომ ვიდეო ფილმების კომპლექტი აკრეფილი იქნას განსაზღვრული თარიღისათვის, სისტემას უნდა გააჩნდეს საძიებო მექანიზმი კლიენტების დაკვეთებზე საპასუხოდ, იმ კითხვების ჩართვით, რომლებიც ეხება ასორტიმენტში არ არსებულ ფილმებს (მაგრამ რომლებიც მას შეუძლია შეუკვეთოს კლიენტის თხოვნით).

ყოველი ფილმისათვის დადგენილია კონკრეტული პერიოდი გაქირავების (დღეებში) შესაბამისი საფასურით ამ პერიოდში გაქირავებისას.

ვიდეომაღაზია უნდა იყოს მზად მყისიერად გასცეს პასუხი ნებისმიერ დაკვეთას მარაგში არსებული ფილმების, ასევე კასეტების რაოდენობისა ან დისკების შესახებ.

საფასური გაქირავებაზე განსხვავდება ვიდეო მატარებლისგან დამოკიდებულებით: კასეტა ან დისკი.

ვიდეომაღაზიის მუშაკები ცდილობენ დაიმახსოვრონ ყველაზე პოპულარული ფილმების დისკები. ხშირად ფილმების იდენტიფიკაციისას ისინი სარგებლობენ ფილმის კოდით და არა დასახელებით.

დამატებითი მოთხოვნები:

კასეტებსა და დისკებზე, რომლებიც ბრუნდება დადგენილი ვადის დაგვიანებით, ერიცხებათ დამატებითი საფასური.

ფილმები იკვეთება მიმწოდებელთან, რომელსაც შეუძლია ფილმის ან კასეტის მოწოდება ერთი კვირის განმავლობაში. ჩვეულებრივ ხდება ერთი შეკვეთა რამოდენიმე ფილმზე.

დაბრონვა შესაძლებელია იმ ფილმების, რომლებიც შეკვეთილია მიმწოდებელთან ან/და რომლის ყველა კოპიები არის გაქირავებაში, ასევე ფილმები, რომლებიც არ არის მარაგში და არ არიან შეკვეთილი მიმწოდებელთან; ამასთან კლიენტისაგან მოითხოვება საფასური გაქირავების ერთი პერიოდისათვის.

კლიენტს ასევე შეუძლია გააკეთოს რამოდენიმე წინასწარი შეკვეთა, მაგრამ ყოველი დაბრონილი ფილმისათვის უნდა მომზადდეს ცალკე შეკვეთა დაბრონვაზე. ბრონი შესაძლებელია გაუქმდეს იმის გამო, რომ კლიენტმა არ გამოიჩინა არავითარი რეაქცია მთელი კვირის განმავლობაში, რომელიც გავიდა იმ მომენტიდან, როდესაც მას შეატყობინეს ფილმის დაქირავებაში შესაძლო აღების შესახებ. თუ ფილმისათვის აღებული იქნა წინასწარ საფასური იგი იწერება კლიენტის ანგარიშზე.

მონაცემთა ბაზა ინახავს ტრადიციულ ინფორმაციას კლიენტებსა და მიმწოდებლებზე, მისამრთს, ტელეფონის ნომერს და ა.შ. ყოველ შეკვეთაში მიმწოდებელს მიეთითება შესაკვეთი ფილმები, მათი რაოდენობა, კასეტის/დისკის ფორმატი, ასევე მოსალოდნელი მიწოდების თარიღი, საფასური, შესაძლო შემცირება და ა. შ.

როდესაც კასეტა ბრუნდება კლიენტის მიერ ან შემოდის მიმწოდებლისაგან, მაღაზიის მუშაკები პირველ რიგში ემსახურებიან კლიენტებს, რომლებმაც გააკეთეს წინასწარი შეკვეთა. სწორი დამუშავებისათვის ინფორმაცია დაკავშირებული ბრონთან ახლდება ორჯერ: კლიენტთან კონტაქტის დამყარების შემდეგ, როდესაც მას შეატყობინეს, „დაბრონილი ფილმი მოვიდა“, და ფილმის კლიენტზე დასაგირავებლად ჩაბარების შემდეგ. ეს ნაბიჯები უზრუნველყოფენ ბრონირების ოპერაციის სწორ ჩატარებას.

კლიენტს შეუძლია აიღოს რამოდენიმე დისკი და კასეტა, მაგრამ ყოველ ცალკე აღებულ ვიდეომატარებელს შეესაბამება ცალკე ჩანაწერი. ყოველ დასაგირავებლად გაცემულ ფილმზე ფიქსირდება გაცემის თარიღი და დრო, დადგენილი და ფაქტიური დაბრუნების ვადა. მოგვიანებით გაქირავების ჩანაწერი ახლდება, იმისათვის რომ გამოჩნდეს ვიდეოფილმის დაბრუნების ფაქტი და საბოლაო ანგარიშსწორების ფაქტი. გარდა ამისა, ჩანაწერი ინახავს ინფორმაციას გამყიდველის შესახებ, რომელიც პასუხს აგებს ფილმის გაქირავებაზე. დეტალური ინფორმაცია კლიენტზე და გაქირავებაზე ინახება ერთი წლის განმავლობაში, იმისათვის რომ მოგვიანებით ადვილი იყოს კლიენტზე ნდობის დონის დადგენა. გაქირავების ძველი ინფორმაცია ინახება წლის განმავლობაში აუდიტის ჩატარების მიზნით.

ყველა ოპერაციები სრულდება პირადი ანგარიშსწორებით, ფულის ელექტრონული გადარიცხვით ან საკრედიტო ბარათებით. კლიენტებისაგან მოითხოვება აწარმოოს გადახდა კასეტების /დისკების გაქირავებაში გაცემისას.

თუ კასეტა/დისკი დაბრუნებულია დადგენილი ვადის გადამეტებით, საფასური აიღება ან კლიენტის ანგარიშიდან, ან უშუალოდ მიიღება კლიენტისაგან.

თუ კასეტა/დისკი დაკავებულია ორ დღეზე მეტი ხნის განმავლობაში, კლიენტს ეგზავნება შეტყობინება დაკავების შესახებ. ერთ და იმავე კასეტა/დისკზე ორი შეტყობინების გაგზავნის შემდეგ, კლიენტი ღებულობს შენიშვნას იმის შესახებ, რომ იგი არის „დამრღვევი“ და მაღაზიაში მისი შემდგომი მიმართვისას, ხელმძღვანელობა განიხილავს მისგან „დამრღვევი“-ს სტატუსის მოხსნის შესახებ საკითხს.

6.7. სტუდენტური სესხების გაცემისა და მისი დაფარვის კონტროლი

უმაღლეს სასწავლებლებში სტუდენტთა დიდი ნაწილი სწავლების საფასურის გადასახდელად სარგელობს ბანკების მიერ დაწესებული სტუდენტური სესხებით, რომელთა გაცემა ხდება როგორც ფიზიკურ ასევე იურიდიულ პირებზე. ბანკს, სტუდენტს და უმაღლეს სასწავლებელს შორის სწორი და დროული ანგარიშსწორებისათვის მნიშვნელოვანია სტუდენტური სესხების გაცემისა და მათი დაფარვის კონტროლი.

სტუდენტური სესხების გაცემის და დამუშავების პროცედურა მიმდინარეობს საკრედიტო ოფიცრების მიერ. მსესხებელთან პირველად გასაუბრებას აწარმოებს საკრედიტო ექსპერტი და უხსნის სესხის პირობებს, ასევე ვადებს და საჯარიმო სანქციებს, არკვევს შეესაბამება თუ არა პირის მონაცემები ბანკის მოთხოვნებს სტანდარტული სტუდენტური სესხის გაცემას მომხმარებლების მიმართ.

მსესხებელი შეიძლება იყოს 19 - დან 55 წლამდე ასაკის საქართველოს მოქალაქე, ფიზიკური პირი, რომელიც სწავლობს სასწავლებელში ბაკალავრიის ან მაგისტრატურის რომელიმე კურსზე. მას უნდა გააჩნდეს სტაბილური შემოსავლები, რომელიც მიიღება არანაკლებ 6 თვიანი უწყვეტი სამუშაო ანაზღაურების სახით ან ბიზნესიდან მიღებული დივიდენდის სახით. მოითხოვება არანაკლებ 1 წლიანი ბიზნესის

საოპერაციო ისტორია მოცემულ დარგში, პარტნიორების დადებითი რეკომენდაციები, მსესხებლის წილი ბიზნესში უნდა იყოს მინიმუმ 20 %. ხელფასის შემთხვევაში – არანაკლებ 400 ლარი, დივიდენტის შემთხვევაში – არანაკლებ 800 ლარი. ასევე, არ უნდა ხასიათდებოდეს უარყოფითი საკრედიტო ისტორიით. სტუდენტური სესხის მიზნობრიობა – ერთი წლის საფასურის გადახდა, რომელიც გადაირიცხება პირდაპირ სასაწავლებელში. სესხის თანხა – მინიმუმ 400 ლარიდან. სესხის ვადა – 3-84 თვე. წლიური საპროცენტო განაკვეთი: წლიური 28%. დაფარვის გრაფიკი – საშეღავათო პერიოდის განმავლობაში ხდება მხოლოდ დარიცხული პროცენტის ყოველთვიური გადახდა, ხოლო საშეღავათო პერიოდის გასვლის შემდგომ ყოველთვიური ანუიტეტით, მსესხებელს არ მოეთხოვება უზრუნველყოფის და თანამონაწილეობის წარდგენა, საიჯარო სანქციები: ვადაგადაცილებაზე ვადაგადაცილებული თანხის 0,5%, ყოველ ვადაგადაცილებულ დღეზე, სესხის გაცემის საკომისიო: სესხის 2,5%, მინიმუმ 40 ლარი.

თუ მსესხებლის მონაცემები შეესაბამება სესხის მოთხოვნებს, საკრედიტო ოფიცერი გადასცემს საჭირო დოკუმენტების სიას, რომელთა წარმოდგენის მერე სესხის გაცემაზე პასუხი იქმნება 3 სამუშაო დღეში. იგი ასევე ინფორმირებული იქნება დადებითი და უარყოფითი გადაწყვეტილების შესახებ.

საჭირო დოკუმენტების წარმოდგენის შემდეგ ივსება სასესხო განაცხადი, ოფიცერი ახდენს ანალიზს, რომლის ეტაპზე ოფიცერმა უნდა უზრუნველყოს შემოსავლების დამადასტურებელი დოკუმენტა-ციის შემოწმება.

თუ მსესხებლის მონაცემები შეესაბამება სესხის გაცემაზე ბანკის მიერ განსაზღვრულ პირობებს, საკრედიტო ოფიცერი საკრედიტო კომიტეტზე გაიტანს განაცხადს, სადაც რისკების მენეჯერი მოახდენს ყველა დოკუმენტის გადამოწმებას და გამოიტანს გადაწყვეტილებას.

დადებითი გადაწყვეტილების მიღების შემთხვევაში საკრედიტო ოფიცერი ყოველთვიური გადასახადის დასაანგარიშებლად და ოპტიმალური ვადის შესარჩევად ავსებს სტუდენტური სესხის კალკულატორს და შედეგს აცნობებს მსესხებელს.

მსესხებლის თანხმობის შედეგად ხდება სესხის რეგისტრაცია – ხელშეკრულების გაფორმება, თუ კლიენტი არ არის რეგისტრირებული პროგრამაში და არ გააჩნია ანგარიში, კლიენტზე იხსნება ბარათი.

ხელშეკრულების გაფორმების შემდეგ მზადდება შემდეგი დოკუმენტები:

- განაცხადი სესხის აღებაზე;
- სესხის გაცემაზე გადაწყვეტილების ოქმი;
- სტუდენტის პირადობის დამადასტურებელი მოწმობის ასლი;
- სტუდენტების შემოსავლების დამადასტურებელი ცნობა;
- ინსტიტუტის მიერ გაცემული სტუდენტობის დამადასტურებელი ცნობა;
- საბანკო კრედიტის ხელშეკრულება;
- სესხის დაფარვის გრაფიკი;

ამ დოკუმენტების მომზადების შემდეგ, მსესხებლის მიერ ხელშეკრულებაზე ხელმოწერის შემდეგ, კრედიტ- ოფიცერი ახდენს თანხის დასმას ანგარიშზე, ასევე საკრედიტო ოფიცერი ამზადებს თანხის უმაღლეს სასწავლებელში გადარიცხვის განკარგულებას, სადაც მისი რეკვიზიტებია მითითებული და გადასცემს საოპერაციო დეპარტამენტს შესასრულებლად.

სტუდენტური სესხების გაცემისა და მისი დაფარვის პროცესის მართვისათვის უნდა შეიქმნას კომპიუტერული ქსელი განაწილებული სამუშაო ადგილებით, კლიენტ-სერვერ ორგანიზაციით.

გამოყენებული ლიტერატურა

1. Джим Арлоу и Айла Нейштадт, **UML 2** н Унифицированный процесс. 2-е издание, Практичеслий объектно- ориентированный анализ и проектирование, Санкт-Петербург - Москва, 2008.
2. Кватрани Т. Rational Rose и UML. Визуальное моделирование; Пер. с англ.М.:ДМК Пресс, 2001.-176 с:нл.
3. სუხიაშვილი თ. ავტომატიზებული მართვის თეორიული საფუძვლები. დამტკიცებულია სტუ-ს სამეცნიერო-ტექნიკური საბჭოს მიერ. გამომცემლობა "ტექნიკური უნივერსიტეტი", 2005, 210 გვ.
4. სუხიაშვილი თ. სისტემების ობიექტ-ორიენტირებული ანალიზი. დამხმარე სახელმძღვანელო საკურსო პროექტის შესადგენად. დამტკიცებულია სტუ-ს სამეცნიერო-ტექნიკური საბჭოს მიერ. გამომცემლობა "ტექნიკური უნივერსიტეტი", 2008, 70 გვ.
5. სუხიაშვილი თ. არაფუნქციონალური მოთხოვნების მოდელირება ბიზნეს-პროცესების მართვისას სტუ-ს შრომები, 2012, № 1(13), გვ. 100-104.
6. სუხიაშვილი თ. კლასების დაპროექტება ობიექტთა მდგომარეობების გათვალისწინებით. სტუ-ს შრომები, 2013, № 1(14), გვ. 86-90.
7. სუხიაშვილი თ. კლასების მოვალეობების განსაზღვრა პროგრამული სისტემის დაპროექტებისას. სტუ-ს შრომები, 2013, № 3(13), გვ. 57-60.
8. სუხიაშვილი თ. სისტემის განაწილებული კონფიგურაციის მოდელირება პარალელურად შესრულებადი მართვის ნაკადებისას. სტუ-ს შრომები, 2014, № 2(18), გვ. 63-67.
9. სუხიაშვილი თ. მონაცემთა ბაზის ლოგიკური სქემის დამუშავება ბიზნეს-პროცესების კომპიუტერული მართვისათვის. სტუ-ს შრომები, 2016, № 1(21), გვ. 110-116.