

# ტექსტური ინფორმაციის შექმნისა და საგამომცემლო მარკეტინგის ერთიანი სისტემის ბიზნესპროცესების მოდელირება

დავით გოგშელიძე, გია სურგულაძე, ეკატერინე თურქია  
საქართველოს ტექნიკური უნივერსიტეტი

## რეზიუმე

განხილულია მხატვრული ლიტერატურისა და სამეცნიერო ნაშრომების შექმნის ხელშემწყობ სისტემასთან ერთად, მისი ყიდვა-გაყიდვისა და წაკითხვის სისტემების პროგრამული პროდუქტის აგების კონცეფცია. უნდა შეიქმნას ერთი, მულტი-პლატფორმული პროგრამა, რომელიც გააერთიანებს ისეთ მოთხოვნებს, როგორცაა, ერთი მხრივ მწერლისთვის წიგნის ან მეცნიერისთვის სამეცნიერო ნაშრომის წერის პროცესის ხელშემწყობა, ასევე მისი მკითხველამდე მიწოდების, გაყიდვის და სხვა მარკეტინგული საშუალებების უზრუნველყოფა, ხოლო მეორე მხრივ მკითხველის ხელშემწყობა წიგნის ან ნაშრომის არჩევის, მისი შეძენისა და წაკითხვისათვის.

**საკვანძო სიტყვები:** წიგნი. წერის პროცესი. ტექსტური ინფორმაცია. საგამომცემლო მარკეტინგი. პროგრამული პროდუქტი.

## 1. შესავალი

მწერლისა და მკითხველის მოთხოვნების დაკმაყოფილება ერთი პროგრამული პროდუქტის საშუალებით, სადაც ასევე იქნება მათი დამაკავშირებელი ისეთი გზების არსებობა, როგორცაა, წიგნის ყიდვა-გაყიდვა, მარკეტინგულ პროცესში კრიტიკოსების, ლიტერატურათმცოდნეების, მეცნიერების და სხვა კომპეტენტური პირების მონაწილეობა, გვადლევს იმის საშუალებას, რომ დავნერგოთ ინოვაციური მეთოდები, ზემოთ აღწერილი ბიზნეს პროცესების მოდელირებისას. თავის მხრივ ინოვაციურია წიგნებისა და სამეცნიერო ნაშრომების შექმნის ხელშემწყობი სისტემა. მსგავსი პროდუქტის შესაქმნელად საჭიროა მწერლების მეთოდოლოგიების, მათი წერისადმი მიდგომის შესაწავლა, კლასიფიკაცია და პროგრამულ პროდუქტად ისეთი გარდაქმნა, როგორც შეიძლება მოერგოს მათ ინდივიდუალურობას. აღსანიშნავია ფაქტი, რომ არ უნდა დაიკარგოს გამომცემლობის როლი, თუმცა ნიჭიერ მწერალთა და მეცნიერთა წარმატება არა მხოლოდ გამომცემლობებზე, უმეტესწილად, სფეროთი დაინტერესებულ მომხმარებელზე იქნება დამოკიდებული.

სასურველია პასუხის გაცემა რამდენიმე შეკითხვაზე. კერძოდ, „იქნებოდა თუ არა კარგი ყოფილიყო ერთი სისტემა, რომელიც გააერთიანებდა მწერლებს და მკითხველებს, ლექტორებსა და სტუდენტებს?“, „იქნებოდა თუ არა კარგი იმავე პროგრამაში გაყიდულიყო წიგნი ან ნაშრომიც რომელშიც იგი შეიქმნა?“, „იქნებოდა თუ არა უკეთესი გვექონოდა ერთი პროგრამა - მუშაობის, გართობის ან ინფორმაციის ძებნისათვის, ერთი ანგარიშით და ეს ანგარიში გააერთიანებდა მოწყობილობებს, რომლებიც დღითიდღე ვითარდება და რომელთა შეცვლაც ძალიან ხშირად გვიწევს?“, „იქნებოდა თუ არა უკეთესი დაგვეწერა წიგნი ან სამეცნიერო ნაშრომი უფრო მარტივად (მოხერხებულად), სპეციალურად წიგნის წერისათვის განკუთვნილი პროგრამით, რომელიც თავიდან აგვარიდებდა რთულ პროგრამებს და მოგვეცემა საშუალებას გაგვეგრძელებინა წერა კომპიუტერიდან

მობილურში, მობილურიდან პლანშეტში და ა.შ.“, „იქნებოდა თუ არა უკეთესი გვესწავლა იმავე პროგრამაში, რომელშიც მხატვრულ ლიტერატურას ვკითხულობთ?“.

რა თქმა უნდა, ყველა კითხვაზე პასუხი ერთი და ძალიან მარტივია „დიახ!“ ეს იქნებოდა პროგრამა, რომელიც განკუთვნილია კონკრეტული სფეროსთვის თუმცა აერთიანებს ამ სფეროში მოღვაწეთა და მომხმარებელთა ინტერესებს.

## 2. ძირითადი ნაწილი

პროგრამას უნდა ჰქონდეს წიგნს წერის, მისი კითხვის, ყიდვისა და გაყიდვის ფუნქციები. შეეძლოს სამეცნიერო ნაშრომების ყიდვა, გაყიდვა. მწერლებისა და გამომცემლობების დაკავშირება, სიახლეების გადმოცემა. პლატფორმებს შორის კავშირი უნდა იყოს მოსახერხებელი და დიზაინი კი განსხვავებული.

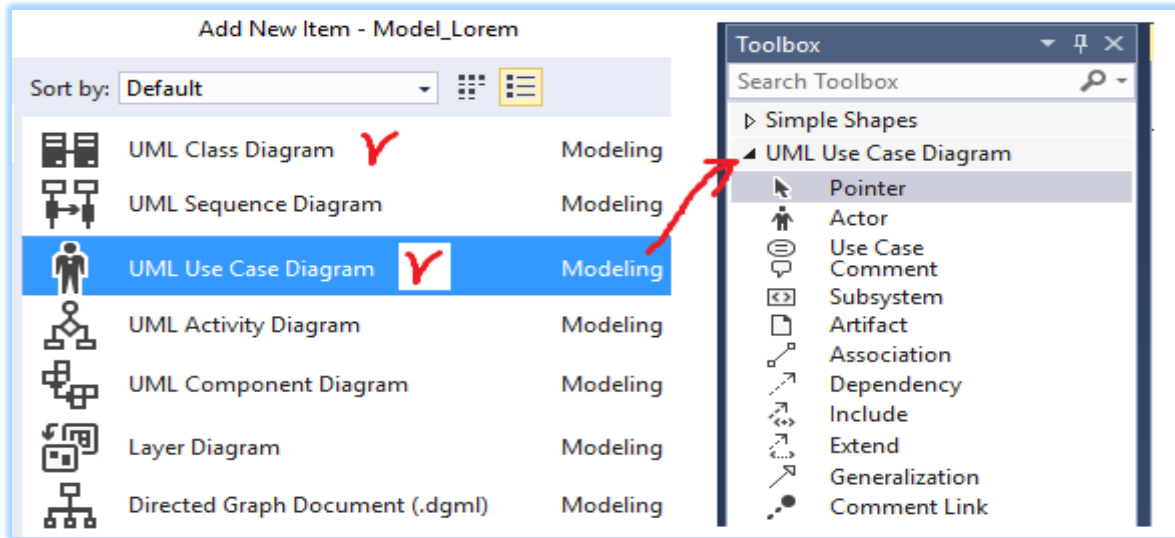
პროგრამა შეიძლება წარმოვადგინოთ სამი ძირითადი ნაწილით:

- 1) წიგნისა და სამეცნიერო ნაშრომის შექმნის ხელშემწყობი სისტემა;
- 2) წიგნებისა და ნაშრომების ელექტრონული მაღაზია;
- 3) წიგნის კითხვის სისტემა.

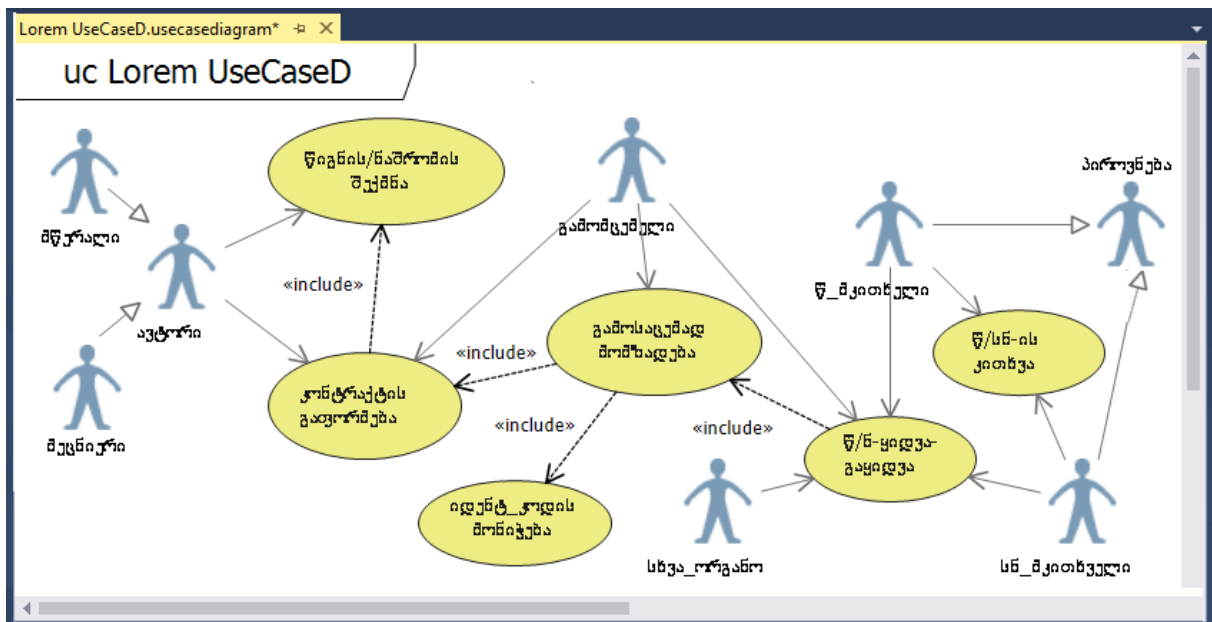
პროგრამაში შეიძლება გამოიყოს მეოთხე სისტემაც, ეს სისტემა კი მომხმარებელთა დაკავშირებას ეხება. უნდა არსებობდეს ისეთი როლის მქონე მომხმარებლები, როგორცაა მეცნიერი, მწერალი, მკითხველი, გამომცემელი, კრიტიკოსი, ლიტერატურათმცოდნე და სხვა. ასევე უნდა არსებობდეს განსხვავებული კავშირები მათ შორის. მაგალითად, კავშირის ტიპები მწერალსა და გამომცემელს, კრიტიკოსსა ან ლიტერატურათმცოდნეს შორის. უნდა იყოს კავშირები მომხმარებლებთან. შესაძლებელი უნდა იყოს ერთმანეთთან დამეგობრება, ჯგუფების შექმნა და მათი მართვა. რომელიმე მწერლის, კრიტიკოსის, გამომცემლობის ან ლიტერატურათმცოდნე მომხმარებლის ან ჯგუფის სიახლეების გამოწერა და ცვლილებების თვალყურისდევნება. ეს ფუნქციონალი ძალიან ჰგავს სოციალური ქსელის ბიზნესმოდელს, თუმცა ვფიქრობთ, რომ მსგავსი ფუნქციების გარეშე საპროექტო პროგრამული პროდუქტი წარუმატებელი იქნება.

პროგრამული აპლიკაციის მიზანია გაამარტივოს მწერლისა და მკითხველის საქმიანობა. დაეხმაროს ავტორებს საკუთარი ნაშრომის პოპულარიზაციაში, საგამომცემლო დაწესებულებების მოძებნაში. როდესაც ზემოთ აღწერილი ფუნქციონალი ერთ სისტემაში ერთიანდება, შესაძლებელი ხდება მათი მომხმარებლების ინტერესების გაერთიანების მიზნით, ისეთი პროგრამული პროდუქტის შექმნა, რომელიც აუცილებლად ესაჭიროება ახალი საუკუნის საგამომცემლო სფეროს. 1-ელ ნახაზზე მოცემულია Ms Visual Studio .NET 2015 პლატფორმაზე Modeling Projects გარემოში ჩვენი საკვლევი ობიექტის როლებისა და ძირითადი ფუნქციების UML-ის UseCase დიაგრამა (ნახ.2).

განვიხილოთ მე-2 ნახაზზე გამოსახული დიაგრამის არსი. დავიწყეთ წიგნის და სამეცნიერო ნაშრომის შექმნიდან. ავტორის (მწერლის, მეცნიერის) ინტერესებში შედის წიგნის წერა, მისი გაყიდვა, გამომცემლობების ძებნა და მათთვის წიგნის ჩვენება, შეთავაზებების გაკეთება. ამავდროულად ნაშრომების და წიგნის გაყიდვის ინტერესები აქვს მეცნიერსაც, რომელიც შეიძლება დაკავშირებული იყოს სახელმწიფო ან კერძო სტრუქტურის საგანმანათლებლო ან სამეცნიერო ორგანიზაციასთან. გასათვალისწინებელია გრეთვე არსებული იურიდიული კანონმდებლობა ამ სფეროში [1,8].



ნახ.1. Visual Studio.NET 2015-ის UML ინსტრუმენტი (Modeling Projects)



ნახ.2. UseCase დიაგრამის ფრაგმენტი: როლები და ფუნქციები

გაყიდვა და მწერლების მოძიებაა გამომცემელთა და სხვა ზემოთ ნახსენები ორგანიზაციების მიზანი. ყიდვისა და გაყიდვის სისტემა, რა თქმა უნდა, ყოველთვის ერთია, ხოლო წიგნი და სამეცნიერო ნაშრომი შეიძლება ერთი საერთო კატეგორიის ქვეშაც გავატაროთ. რაც შეეხება მყიდველებს, მეცნიერებით ან ლიტერატურით დაინტერესებული კლიენტების ინტერესებში შედის ის, რომ მათ იმავე პროგრამაში წაიკითხოთ წიგნი სადაც ყიდულობენ. მოცემულ სფეროში, მსოფლიოს უმსხვილესი კომპანიები, Amazon და Google აერთიანებს წიგნის შექმნისა და მისი კითხვის ფუნქციებს, თუმცა არაა გაერთიანებული მისი შექმნისა და გაყიდვის სისტემები. საერთო ჯამში შეიძლება ითქვას ჩვენი პროგრამის ერთ-ერთი ყველაზე მნიშვნელოვანი უპირატესობა არის ის, რომ იგი აერთიანებს შექმნას, გაყიდვა-ყიდვას და მოხმარებას. ეს ყველაფერი კი გვაძლევს საშუალებას მწერილსა და მკითხველის კავშირი იყოს ინოვაციური.

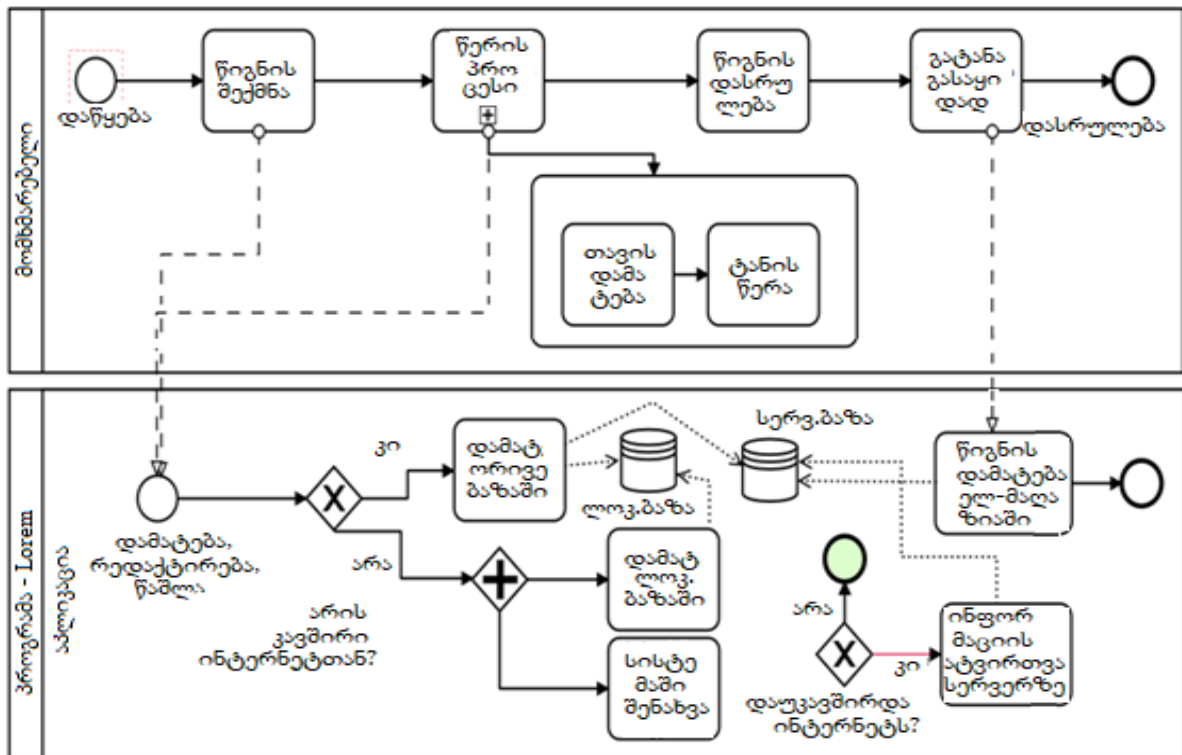
წიგნის წერის სისტემა ერთი შეხედვით ძალიან მარტივია. ყველაზე მნიშვნელოვანი ისაა როდესაც მომხმარებელი წერს და იქმნება მარტივი წერის სისტემის შთაბეჭდილება, მაგრამ ეს სისტემა პროგრამისთვის ძალზე ცოცხალი და აქტიური ფუნქციაა. წერის პროცესში პროგრამის „უკან“ იმალება ის პროცესები, რომელსაც მომხმარებელი ვერ ხედავს.

პირველ რიგში პროგრამის დაწერისას გასათვალისწინებელია .epub ფორმატის მქონე ფაილების მოთხოვნები და სტრუქტურა, რომ წიგნი შეიქმნას ამ ფორმატში. მარტივად ვთქვათ, თუ როგორი უნდა იყოს წერის სისტემა პროგრამაში. ყველა წიგნს ეყოლება ავტორი, ექნება დასახელება, მოკლე აღწერა, Tag სიტყვები, წინასიტყვაობა, შესავალი მომხმარებლის მიერ თავად დამატებული რაიმე ძირითადი კომპონენტი. ყველა წიგნი დაიყოფა თავებად. მონაცემთა ბაზაში შეინახება სტრიქონები ინფორმაციით მომხმარებლის და წიგნის იდენტიფიკატორები, შეიქმნება პარაგრაფის იდენტიფიკატორი. ყოველ პარაგრაფს ექნება საკუთარი ცვლილებების ისტორია. ტექსტური ფაილები არ მოიცავს დიდ მუხსიერებას და შესაბამისად შეგვიძლია გავაკეთოთ ყოველ პარაგრაფზე მუშაობის პროცესში, მისი სარეზერვო ასლი, მაგალითად ყოველ 10 წუთში ხოლო არსებობიდან რამდენიმე დღის შემდეგ ბოლოს ამ სარეზერვოების წაშლა და მაგალითად 5 დღის წინანდელი ერთ-ერთი ასლის დატოვება. როგორც უკვე აღვნიშნეთ, წიგნის მთელი ინფორმაცია, ყოველი სიტყვა იქნება შენახული არა ფაილებში, არამედ მონაცემთა ბაზებში და ბაზებიდან მოხდება „“.epub““““ წიგნების შევსებაც. ეს ყველაფერი საშუალებას მოგვცემს, რომ ყოველი თავისთვის შევქმნათ ერთგვარი ისტორია. ასევე ყოველ პარაგრაფს ექნება სპეციალური ჩანაწერები, იქნება ეს იდეები, შენიშვნები თუ გეგმები. ეს უკანასკნელი ინფორმაცია პარაგრაფის წერისას უნდა იყოს მარტივი სანახავი და შესაქმნელი, რათა გაამარტივოს მწერლისთვის წერის პროცესი.

დიდი რაოდენობის წიგნების ყოველი პროგრამის მონაცემთა ბაზაში ცალკე სტრიქონად შენახვა რთული გადასაწყვეტია. კონკრეტულად საინტერესოა თუ როგორ უნდა განვალაგოთ ინფორმაცია მონაცემთა ბაზაში ისე, რომ პროგრამამ იმუშაოს სწრაფად და ეფექტურად. ამ პრობლემის გადასაწყვეტად, ყველა პროგრამისტისათვის ცნობილ საიტზე stackoverflow.com, დავსვით შეკითხვა: როგორ უნდა განვალაგოთ ბაზაში მილიონობით ან მილიარდობით სტრიქონი ისე, რომ პროგრამამ იმუშაოს სწრაფად [3].

აგრეთვე ვთქვით ის, რომ სტრიქონები დაიყოს ტიპის მიხედვით, ტიპში კი წიგნი იგულისხმებოდა. გვექონდა ორი ვარიანტი, პირველი, რომ ყველა სტრიქონი დაგველაგებინა ერთ ცხრილში და მეორე, რომ ყოველი ტიპისთვის (წიგნისთვის) შეგვექმნა ცალკე ცხრილები პროცედურის მეშვეობით. საუკეთესო პასუხი, რაც მივიღეთ იყო ის, რომ მილიარდობით სტრიქონის ერთ ცხრილში მოთავსება არ წარმოადგენს პრობლემას თანამედროვე სისტემებისათვის და თუ გვინდა ინფორმაციის გამოტანა WHERE შეზღუდვით ისე, რომ არ მოხდეს მილიარდივე ობიექტის შემოწმება აკმაყოფილებს თუ არა ჩვენ მოთხოვნებს, ტიპის ანუ წიგნის სვეტში უნდა შემოვიტანოთ ინდექსი, ეს კი იმას ნიშნავს, რომ თუ გვექნება 10 000 წიგნი და ასი მილიონი სტრიქონი, თუ დაგვჭირდება რომელიმე კონკრეტული წიგნის წამოღება ინტერნეტიდან, ინდექსის მეშვეობით SELECT მოხდება 10 000 ჩანაწერს შორის, ნაცვლად ასი მილიონისა. ჩვენი შეხედულებით, ზემოაღნიშნული ამოცანის გადაჭრა შესაძლებელია ახალი ინფორმაციული ტექნოლოგიების, კერძოდ კი NoSQL ბაზების (მაგალითად, MongoDB) გამოყენებით [4,5].

მე-3 ნახაზზე ნაჩვენებია წიგნის წერის ბიზნესპროცესის მოდელი, აგებული სტანდარტული BPMN ინსტრუმენტის საფუძველზე [2,6,8].



ნახ.3. წიგნის წერის ბიზნესპროცესის BPMN დიაგრამა

დიაგრამით ხაზი გავუსვით ფუნქციას, რომ წიგნის წერის დროს ინფორმაცია უნდა იტვირთებოდეს ან კლიენტის მოწყობილობაზე მყოფ მონაცემთა ბაზაში, ან ამ ბაზასა და სერვერზე ერთად. როგორც უკვე ვთქვით, მოცემული ფუნქცია იმუშავებს მხოლოდ ჭკვიან მოწყობილობებზე, ხოლო ვებ-საიტზე შესაძლებელი იქნება მუშაობა მხოლოდ Online რეჟიმში [7]. განვიხილოთ Offline რეჟიმში მყოფი მოწყობილობის ინტერნეტთან დაკავშირების დროს ინფორმაციის სერვერზე ავტომატური სინქრონიზაციის პროცესის საკითხი. პირველი, რაც ფუნქციის შექმნისას უნდა გავითვალისწინოთ, არის ზოგადი პოლიტიკა, რომელიც გადაწყვეტს თუ რომელი მხარეა „ავტორიტეტული“ სერვერისა და კლიენტის მხარეს ცვლილებების კონფლიქტის შემთხვევაში. მაგალითად რომელიმე წიგნის მეხუთე პარაგრაფის მეორე აბზაცის მესამე სიტყვა მწერალმა შეცვალა მოწყობილობაზე ორი მოწყობილობიდან. პირველი 10:00 საათზე, მან მოახდინა ცვლილება პლანშეტური კომპიუტერიდან, რომელსაც არ აქვს კავშირი ინტერნეტთან. მეორე, 11:00 საათზე კომპიუტერიდან, რომელიც დაკავშირებულია სერვერის აპლიკაციას. პლანშეტური კომპიუტერი ინტერნეტს დაუკავშირდა 12:00 საათზე.

ასეთ შემთხვევაში, სინქრონიზაცია უნდა მოხდეს არა მოწყობილობის მხრიდან სერვერის მიმართულებით, არამედ პირიქით, ინფორმაცია უნდა განახლდეს სერვერზე. მაგრამ თუ პარალელურად პლანშეტიდან მწერალს ჰქონდა რაიმე სხვა ცვლილება ან ახალი დანამატი სხვა წიგნზე ამ შემთხვევაში სინქრონიზაციის მიმართულება იქნება ორმხრივი.

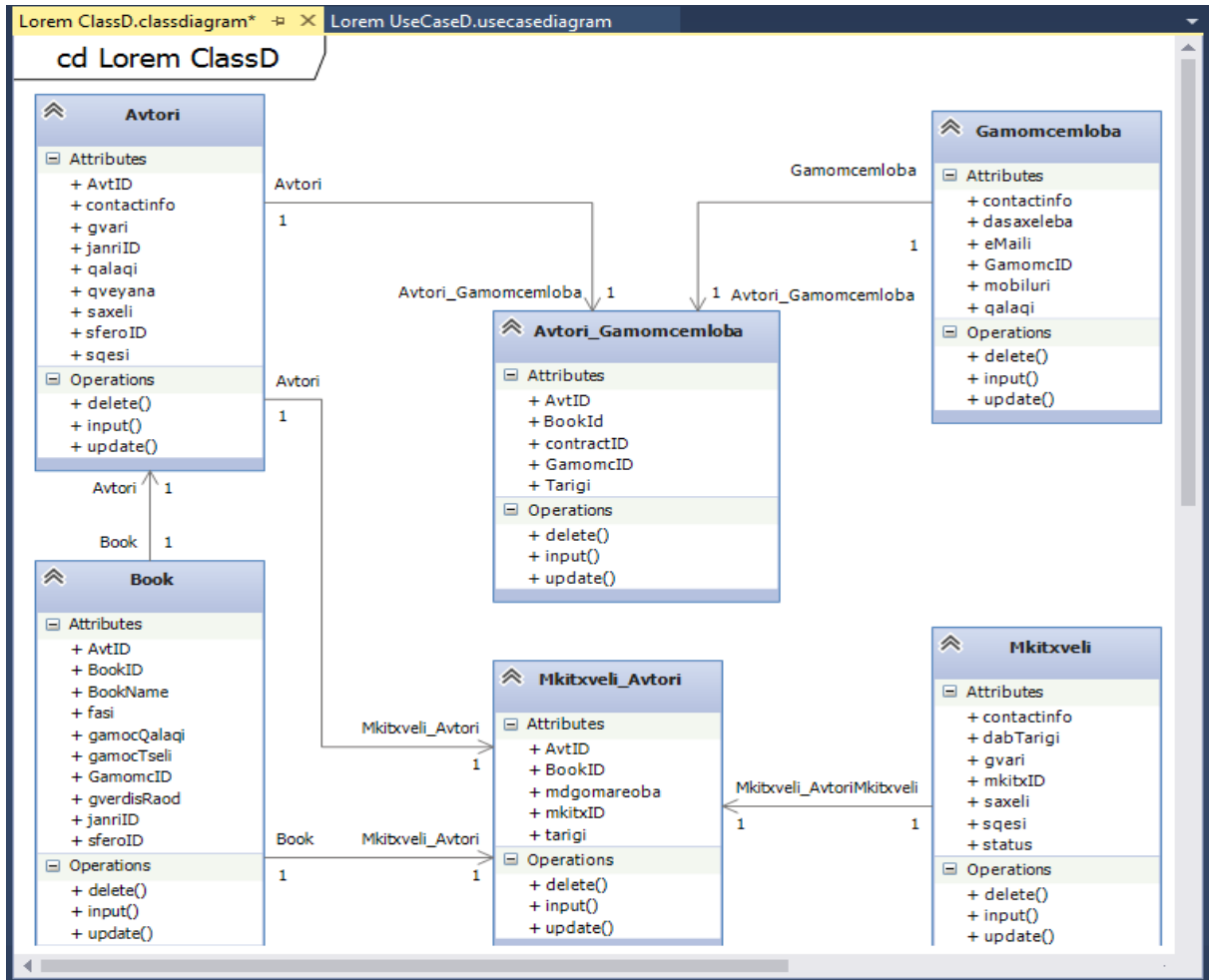
ამ გადაწყვეტაში ერთი გადაუჭრელი პრობლემა რჩება, როდესაც მომხმარებელი Offline მოწყობილობაში განაახლებს პირველ თავს, ხოლო ამის შემდეგ Online

მოწყობილობაში მეორე თავს. როდესაც პირველი მოწყობილობა ინტერნეტს დაუკავშირდება პირველი თავის განახლებული ვარიანტი წაიშლება. ამის საწინააღმდეგოდ შეიძლება გამოვყოთ წიგნებში თავების ისტორიები, მაგრამ შეიძლება ეს ცვლილება ერთ თავშიც ხდებოდეს და ამას უკვე ვეღარაფერს ვუშველით, რადგან თავში აბზაცების ისტორიის ცალკე შენახვა უკვე მეტისმეტად დაამძიმებს სისტემას. ამ პრობლემის მოსაგვარებლად შეიძლება სისტემამ ინტერნეტთან დაკავშირების დროს მომხმარებელს ჰკითხოს თუ რომელი ვერსიის დატოვება სურს. მაგრამ ეს ფუნქცია მომხმარებლისათვის უფრო რთულს გახდის სისტემას და ეს უკანასკნელი ძალიან მნიშვნელოვანია. მაგრამ თუ მომხმარებელმა Offline -ში დაწერა რამდენიმე ფურცელი, ამის შემდეგ online მყოფი სხვა მოწყობილობიდან უბრალოდ უკვე არსებულ ტექსტს დაამატა მაგალითად, მძიმე, მაშინ ცვლილების სტატუსი გახდება შეცვლილი და შესაბამისად მეორე შემთხვევა დარჩება სერვერზე და პირველი შემთხვევა დაიკარგება, როდესაც მოწყობილობა გავა გლობალურ ქსელში. ამ პრობლემის მოგვარების ყველაზე რეალური შანსია ავტომატურად ასლების შექმნა და მათი შენახვა სერვერსა თუ მოწყობილობაზე, რომ თუ მომხმარებელი დაკარგავს ინფორმაციას, მან შეძლოს მისი აღდგენა წიგნების პარამეტრების სპეციალური გვერდიდან.

წიგნის წერის პროცესი მოკლედ რომ გადმოვცეთ, როდესაც მომხმარებელი იწყებს წერას, იქ ქმნის წიგნს, ჩანაწერების გაკეთება მას შეუძლია ზოგადად წიგნის ქვეშაც. შემდეგ მას შეუძლია შექმნას შესავალი. ამის შემდეგ იწყება რეალური წიგნის წერის პროცესი და ის ქმნის პირველ თავს. კონკრეტული თავის ქვეშ მას შეუძლია ჩანაწერების შექმნა. ის იწყებს წიგნში თავების წერას. პროგრამა აღიქვამს წერის პროცესს და ამ პროცესში ის ქმნის სარეზერვო ასლებს. ნაწერი ოფიციალურად მისთვის განკუთვნილ მონაცემთა ბაზაში არ მოხვდება მანამ, სანამ მწერალი არ შეინახავს მას სპეციალური ლილავის საშუალებით. თუ პროგრამა გამოირთვება ავარიულად, ანუ მოხდება აპარატის გამორთვა, სისტემის გათიშვა და ა.შ., მწერალს მარტივად შეეძლება სარეზერვოდ შენახული ყოველი 5 წუთის წინ შესრულებული ინფორმაციის აღდგენა. როდესაც მწერალი დააწვება შენახვის ლილავს, სისტემა მოახდენს ტექსტის გაშიფვრას, ის გამოყოფს აბზაცებს და შეინახავს მათ ცალ-ცალკე სტრიქონებში, სადაც მათ ტრიგერის საშუალებით გაშვებული პროცედურა შექმნის სპეციალურ ინდექსებს. თუ წიგნი შეიცავს გრაფიკულ ფაილებს, პროგრამა ყოველ გრაფიკულ ფაილს მიანიჭებს GUID უნიკალურ იდენტიფიკატორს და შეინახავს ამ სურათებს სერვერისა და მოწყობილობის მეხსიერებაში ამ უნიკალური სახელით, ტექსტში, კი მიეთითება HTML კოდი, სადაც იქნება დაწერილი სურათის მიბმა, თავის შესაბამისი სტილებით, ხოლო ამ სურათის სახელად მითითებული იქნება, რა თქმა უნდა, უკვე შექმნილი GUID იდენტიფიკატორი, რომელიც ფაილს ერქმევა. თუ მოწყობილობა არ იქნება კავშირში გლობალურ ქსელთან, სარეზერვო ასლები, ისევე როგორც ყველა სხვა ინფორმაცია, რომელიც მონაცემთა ბაზებში, უნდა აიტვირთოს, შეიქმნება თავად მოწყობილობაში და მასში არსებული სერვისის საშუალებით, ინტერნეტთან კავშირის დამყარებისთანავე აიტვირთება სერვერზე, ხოლო სარეზერვო ასლები ლოკალურ მოწყობილობაზე წაიშლება. როდესაც მწერალი მორჩება წიგნის წერას, მას შეეძლება მიანიჭოს წიგნს ფასი და გაიტანოს ის გასაყიდად, ასევე მას შეეძლება დაუკავშირდეს პროგრამაში დარეგისტრირებულ ოფიციალურ გამომცემლობებს, წარუდგინოს მათ წიგნი

(ნაშრომი), ორმხრივი შეთანხმების საფუძველზე გააფორმონ კონტრაქტები, მიიღონ შემოსავალი მათი გაყიდვებიდან და ა.შ.

მე-4 ნახაზზე მოცემულია დასაპროექტებელი სისტემის კლასთა-ასოციაციის დიაგრამის ფრაგმენტი, რომელშიც რეალიზებულია ავტორის, გამომცემლობის, მკითხველის, წიგნების და სხვა კლასების (დასახელება, ატრიბუტები და ფუნქციები) ერთობლიობა, ასოციაციური კავშირებით.



ნახ.4. UML Class-Association დიაგრამის ფრაგმენტი

### 3. დასკვნა

შემუშავებული კონცეფციის საფუძველზე ასაგები პროგრამული პროდუქტი იქნება მოქნილი და ეფექტური მხარდამჭერი სისტემა თანამედროვე საგამომცემლო სფეროში, რომელიც აერთიანებს წიგნების (ან სამეცნიერო ნაშრომების) შექმნის, გამოცემის, ყიდვა-გაყიდვის ფუნქციებს, აგრეთვე მკითხველებთან ონლაინ კავშირებს მარკეტინგული ფუნქციის თვალსაზრისით. უბრალო ტექსტური რედაქტორები ჩანაცვლებულ იქნება წიგნის წერისათვის განკუთვნილი სპეციალური სისტემით, რომელიც სწორი დაპროექტების, დაპროგრამების, სისტემის ანალიზის და მარკეტინგის შემთხვევაში გაუწევს კონკურენციას მსოფლიოს წამყვან კომპანიებს მოცემულ სფეროში. ნაშრომში შემოთავაზებული სისტემის ფუნქციონალი შეუწყობს ხელს ელექტრონული მაღაზიებისა და წიგნის კითხვის სისტემის გამოყენებასაც.

**ლიტერატურა - References – Литература:**

1. სურგულაძე გ., თურქია ე. (2016). პროგრამული სისტემების მენეჯმენტის საფუძვლები. სტუ, გამომც. „ტექნიკური უნივერსიტეტი“. თბილისი,
2. ჩოგოვაძე გ., ფრანგიშვილი ა., სურგულაძე გ. (2017). მართვის საინფორმაციო სისტემების დაპროგრამების ჰიბრიდული ტექნოლოგიები და მონაცემთა მენეჯმენტი. სტუ, გამომც. „ტექნიკური უნივერსიტეტი“. თბილისი.
3. Learn, Share, Build in Stack Overflow. <https://stackoverflow.com/>
4. სურგულაძე გ., კვიციანი გ. (2017). შესავალი NoSQL მონაცემთა ბაზებში (MongoDB). სტუ, „IT-კონსალტინგის ცენტრი“. თბილისი.
5. სურგულაძე გ., პეტრიაშვილი ლ. (2017). მონაცემთა მენეჯმენტის თანამედროვე ტექნოლოგიები (Oracle, MySQL, MongoDB, Hadoop). სტუ. „IT-კონსალტინგის ცენტრი“, თბ.
6. სურგულაძე გ., ქრისტესიაშვილი ხ., სურგულაძე გ. (2015). საწარმოო რესურსების მენეჯმენტის ბიზნეს-პროცესების მოდელირება და კვლევა. სტუ. „ტექნიკური უნივერსიტეტი“. თბილისი.
7. გოგშელიძე დ. (2017). Lorem: სისტემა წიგნების, სტატიების, სამეცნიერო ნაშრომების და სხვა ტექსტური ინფორმაციის შექმნის, მოძიებისა და გამოყენებისთვის. სტუ. სამაგისტრო ნაშრომი. ხელმძღვ. გ სურგულაძე. თბილისი.
8. თურქია ე. (2010). ბიზნეს-პროექტების მართვის ტექნოლოგიური პროცესების ავტომატიზაცია. სტუ. „ტექნიკური უნივერსიტეტი“, თბილისი.

**MODELING OF BUSINESS PROCESSES OF A UNITED SYSTEM OF CREATION OF TEXT INFORMATION AND PUBLISHING MARKETING**

Gogshelidze David, Surguladze Gia, Turkia Ekaterine  
Georgian Technical University

**Summary**

Our goal is to create a system that not only helps to create fiction and scientific works but also creates a software product for its implementation, creating a single multiplatform program that combines such requirements as on one hand helping the writer create a work or a scientist with his discovery and also with another hand filing the reader the opportunity to acquaint themselves with the purchase or read the given literature or scientific manual.

**МОДЕЛИРОВАНИЕ БИЗНЕС-ПРОЦЕССОВ ЕДИНОЙ СИСТЕМЫ СОЗДАНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ И ИЗДАТЕЛЬСКОГО МАРКЕТИНГА**

Гогшелидзе Д., Сургуладзе Г., Туркия Е.  
Грузинский Технический Университет

**Резюме**

Рассмотрена система содействия созданию художественной литературы и научных трудов, а также концепция построения ее программного продукта. Должна быть создана одна, мультиплатформенная программа, которая объединит такие требования, как, с одной стороны, содействие автору в написании книги или научной работы, доставка ее до пользователя и продажа, а с другой стороны, содействие читателю в поиске, приобретении и чтении книги или научной работы.