



სისტემის არქიტექტურა

С ჰეპატიტის სკრინინგის ელექტრონული
აღრიცხვის სისტემა

დოკუმენტის შინაარსი

1	შემოკლებები და აკრონიმები	4
2	დიაგრამებისა და სურათების სარჩევი.....	5
3	ცხრილების სარჩევი.....	5
4	დოკუმენტის დანიშნულება	6
5	STOP C - მოკლე აღწერილობა და ფუნქციონალი	6
6	STOP C პროგრამული ეკოსისტემა.....	6
6.1	NCDC.eHealth	7
6.2	MOH.ElimC.....	7
6.3	MOH.eHealth	8
6.4	სხვა დაკავშირებული მოდულები.....	8
6.5	ეკოსისტემის სამომავლო განვითარება	8
7	STOP C მოდულების ზოგადი აღწერა და დანიშნულება	9
7.1	პორტალი	10
7.1.1	ადმინისტრირების პროცესის მოკლე აღწერა	10
7.2	მომხმარებელთა მართვის მოდული.....	12
7.2.1	მოდულის ზოგადი აღწერა.....	12
7.2.2	მოდულის ფუნქციონალის აღწერა	13
7.3	ანალიტიკური მოდული	13
7.4	საკომუნიკაციო მოდული	13
7.5	საერთო მონაცემების მოდული.....	14
8	პლატფორმა და გამოყენებული ტექნოლოგიები.....	15
9	სისტემის არქიტექტურა	17
9.1	ცენტრალური მოდული (პორტალი).....	17
9.1.1	კლასები.....	17
9.1.2	ფიზიკური არქიტექტურა	24
9.2	მომხმარებელთა მართვის მოდული.....	25
9.2.1	ზოგადი არქიტექტურა	25
9.2.2	ფიზიკური არქიტექტურა	28
9.2.3	ინფორმაციული ნაკადები	28
9.2.4	უსაფრთხოება	30
9.3	საერთო მონაცემების მოდული.....	30
9.3.1	საერთო მონაცემების მოდულის კლასები	30

9.3.2 ინფორმაციული ნაკადები	32
10 მონაცემთა ბაზის სტრუქტურა და აღწერილობა	33
10.1 მომხმარებელთა მართვის მოდულის მონაცემთა ბაზა	33
10.1.1 მონაცემთა ბაზის სტრუქტურა	34
10.1.2 მონაცემთა ბაზის ცხრილები და ველები	35
10.1.3 ცხრილებს შორის კავშირები	38
10.2 ცენტრალური მოდულის (პორტალის) ბაზა	41
10.2.1 Microsoft SQL: მონაცემთა ბაზის სტრუქტურა	42
10.2.2 Microsoft SQL: მონაცემთა ბაზის ცხრილები და ველები	43
11 აუცილებელი ბიბლიოთეკები	47
11.1 NPOI	47
11.2 ASP Chart Control	47
11.3 DevExpress	47
11.4 NHibernate	47
11.5 Bootstrap	47
12 მოთხოვნები ფიზიკური არქიტექტურისადმი	48
12.1 მინიმალური არქიტექტურა	48
12.2 კომპონენტების არსებული მისამართები	48
12.3 რეკომენდაციები რეზერვირების სქემასთან დაკავშირებით	49
13 ინსტალაცია და კონფიგურაცია	49
14 APPENDIX A: გალიდაციების სინტაქსი	50
14.1 ოპერატორები	50
14.2 ფუნქციები	51

1 შემოკლებები და აკრონიმები

STOP C / მოდული	C ჰეპატიტის სკრინინგის ელექტრონული მოდული
Analytics	C ჰეპატიტის მონაცემების ანალიტიკური სისტემა
Portal	მთავარი პორტალი / Portal
UMM	მომხმარებელთა მართვის მოდული (User Management Module)
CDM	Common Data Module (საერთო მონაცემების მოდული)
SB	Service Bus (საკომუნიკაციო მოდული)
MOH / MoLHSA	საქართველოს შრომის, ჯანმრთელობისა და სოციალური დაცვის სამინისტრო (Ministry of Labour, Healthcare and Social Affairs)
ElimC	H ჰეპატიტის ელიმინაციის პროგრამა
SSA	Social Service Agency / სოციალური მომსახურების სააგენტო
NCDC	დაავადებათა კონტროლის ეროვნული ცენტრი / National Center of Disease Control
CRA	საჯარო რეესტრი / Civil Registry Agency
SDA	სერვისების განვითარების სააგენტო / Service Development Agency

2 დიაგრამებისა და სურათების სარჩევი

სურათი 1. STOP C პროგრამული ეკოსისტემის დიაგრამა, რეალიზებული ტექნოლოგიების მითითებით.....	7
სურათი 2. STOP C პროგრამული ეკოსისტემის სამომავლო განვითარების ვარიანტი	9
სურათი 3. STOP C ფორმების სამაგალითო გვერდი (პირადი მონაცემების ფორმა).....	10
სურათი 4. STOP C ფორმის, მონაცემებისა და პროცესის დიაგრამა.....	11
სურათი 5. სკრინინგის ფორმის ადმინისტრირების ზოგადი სქემა	12
სურათი 6. საკომუნიკაციო მოდულის მუშაობის პრინციპის დიაგრამა.....	14
სურათი 7. STOP C კომპონენტების არქიტექტურა.....	17
სურათი 8. პორტალის Helper-კლასების დიაგრამა (UML).....	18
სურათი 9. Converter კლასები	19
სურათი 10. Entity კლასები.....	20
სურათი 11. Model კლასები	22
სურათი 12. მომხმარებლების მართვის მოდულის დონეები.....	25
სურათი 13. მომხმარებელთა მართვის მოდულის ბიზნეს-ლოგიკა.....	27
სურათი 14. UMM: ინფორმაციული ნაკადები.....	28
სურათი 15. STOP C კომპონენტების არქიტექტურა მონაცემთა ბაზების ჭრილში	33
სურათი 16. ატრიბუტების სქემა	34
სურათი 17. მომხმარებლები, უფლებები და ატრიბუტების მნიშვნელობები	35
სურათი 18. ატრიბუტების სქემები	40
სურათი 19. მომხმარებლები და ატრიბუტების მნიშვნელობები.....	41
სურათი 20. პორტალის მონაცემთა ბაზის ცხრილების სტრუქტურა (ნაწილი 1)	42

3 ცხრილების სარჩევი

ცხრილი 1. Helper კლასები	18
ცხრილი 2. Converter კლასები: Entity to Model და Model to Entity	19
ცხრილი 3. Entity კლასები	20
ცხრილი 4. Model კლასების აღწერა.....	23
ცხრილი 5. შეტანილი/მიღებული ინფორმაცია.....	29
ცხრილი 6. გაცემული ინფორმაციის ნაკადები	29
ცხრილი 7. შეტანილი/მიღებული ინფორმაცია.....	32
ცხრილი 8. გაცემული ინფორმაციის ნაკადები	32
ცხრილი 9. მომხმარებლების ადმინისტრირების მოდულის მონაცემთა ბაზის ცხრილების აღწერილობა და ველები	35
ცხრილი 10. მონაცემთა ბაზის ცხრილებს შორის კავშირები და ტიპები	38
ცხრილი 11. პორტალის მონაცემთა ბაზის ცხრილების აღწერილობა და ველები	43
ცხრილი 12. STOP C სისტემის მოდულების მისამართები.....	48

4 დოკუმენტის დანიშნულება

C ჰეპატიტის სკრინინგის ელექტრონული მოდულის¹ (STOP C სისტემა) ტექნიკური არქიტექტურის დოკუმენტი წარმოადგენს STOP C სისტემის დანერგვისა და გამართვის პროექტის ფუნდამენტურ ნაწილს. დოკუმენტი აღწერს სისტემის კომპონენტების ტექნიკურ სტრუქტურას, მათი ურთიერთქმედების სპეციფიკას და გამოყენებულ ტექნოლოგიებს. არქიტექტურის დოკუმენტი განკუთვნილია სისტემის ადმინისტრატორებისა და დეველოპერებისთვის და შეიცავს ყველა საჭირო და აუცილებელ ინფორმაციას სისტემის სამომავლო მხარდაჭერისა და განვითარებისთვის.

მიმდინარე დოკუმენტი არ მოიცავს სისტემური ადმინისტრატორის ინტერფეისთან მუშაობის დეტალურ ინსტრუქციას. ამ მიზნისთვის მნიშვნელოვან დოკუმენტს წარმოადგენს STOP C სისტემის სახელმძღვანელოები, რომელიც ასევე არის სისტემური ტექნიკური დოკუმენტაციის აუცილებელი ნაწილი:

- C ჰეპატიტის სკრინინგის ელექტრონული აღრიცხვის სისტემის ადმინისტრატორის სახელმძღვანელო
- სახელმძღვანელო C ჰეპატიტის მართვის სახელმწიფო პროგრამის ფარგლებში კონფირმაციული კვლევისთვის საჭირო სისხლის ნიმუშის აღებასთან დაკავშირებული მონაცემების ელექტრონული რეგისტრაციისთვის²
- სახელმძღვანელო C ჰეპატიტის სკრინინგთან დაკავშირებული მონაცემების ელექტრონული რეგისტრაციისთვის³

5 STOP C - მოკლე აღწერილობა და ფუნქციონალი

C ჰეპატიტის სკრინინგის ელექტრონული მოდულის - „STOP C“ (შემდგომში სისტემა) დანიშნულებაა C ჰეპატიტის მართვის სახელმწიფო პროგრამის ფარგლებში სკრინინგის კომპონენტით გათვალისწინებული C ჰეპატიტზე სკრინინგის შედეგების აღრიცხვა და დიაგნოსტიკის კომპონენტით გათვალისწინებული კონფირმაციული კვლევისთვის საჭირო სისხლის ნიმუშის აღებასთან და ტრანსპორტირებასთან დაკავშირებული მონაცემების ელექტრონული რეგისტრაცია.

C ჰეპატიტის სკრინინგის ელექტრონული მოდული - „STOP C“ განთავსებულია შემდეგ ელექტრონულ მისამართზე: <http://stop-c.moh.gov.ge>.

6 STOP C პროგრამული ეკოსისტემა

გარდა STOP C მოდულისა, C ჰეპატიტის ელიმინაციის პროგრამული ეკოსისტემა შედგება სხვადასხვა ტიპის სისტემებისა და მოდულებისგან, რომლებიც განთავსებულია სხვადასხვა ფიზიკურ და პროგრამულ პლატფორმებზე. ეს მოდულების დაყოფილია 4 ძირითად ჯგუფად:

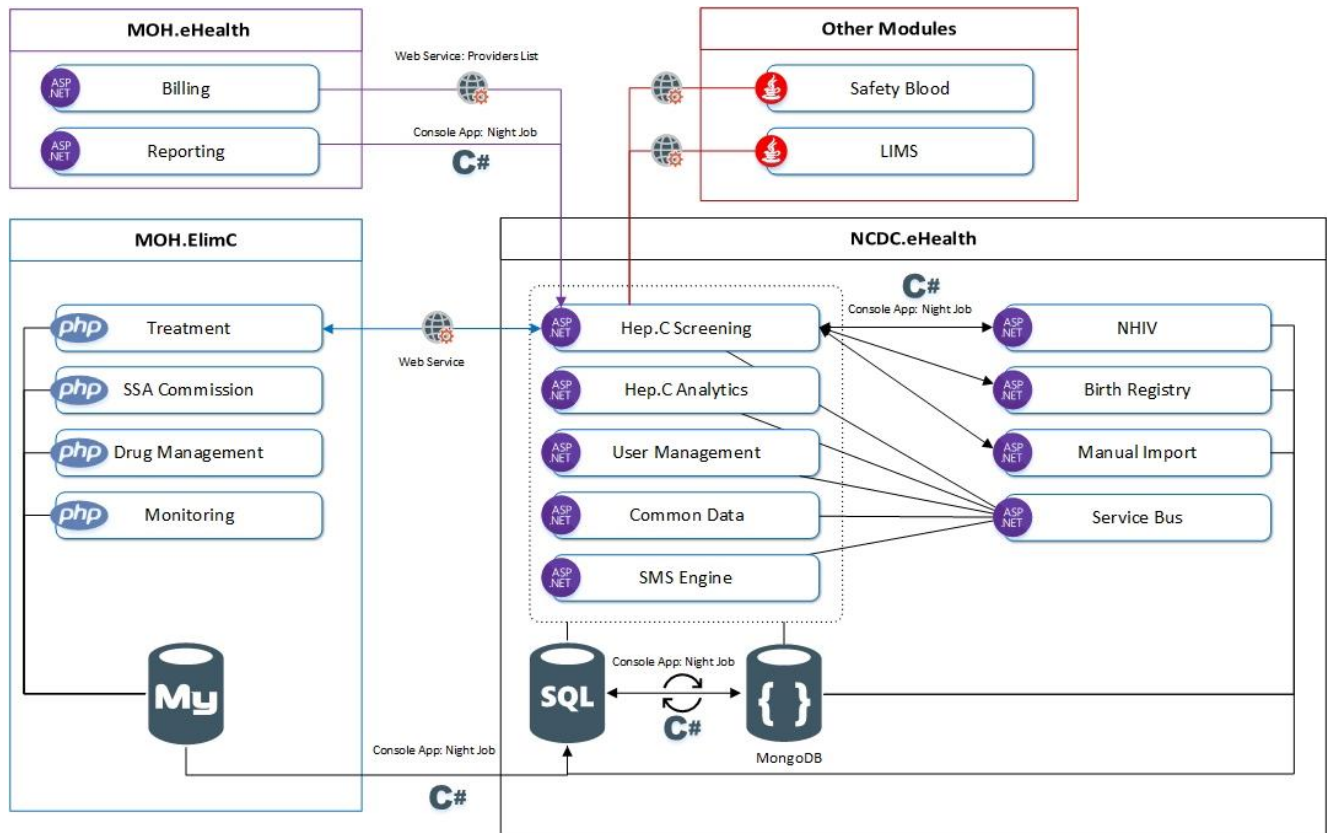
- NCDC.eHealth
- MOH.ElimC
- MOH.eHealth
- სხვა მოდულები

¹ <http://stop-c.moh.gov.ge>

² <http://stop-c.moh.gov.ge/Files/doc1.docx>

³ <http://stop-c.moh.gov.ge/Files/doc2.docx>

სურათი 1 წარმოადგენს STOP C ეკოსისტემის შემადგენელი პროგრამული კომპონენტებისა და მათი კავშირების დიაგრამას, ტექნიკური მახასიათებლების გათვალისწინებით.



სურათი 1. STOP C პროგრამული ეკოსისტემის დიაგრამა, რეალიზებული ტექნოლოგიების მითითებით

6.1 NCD.eHealth

ეს მოდულების მოიცავს დაგადებათა კონტროლისა და საზოგადოებრივი ჯანდაცვის ეროვნული ცენტრის⁴ (NCDC) პროცესების მართვის სხვადასხვა პროგრამულ ინსტრუმენტებს, რომელთა შემადგენლობაშიც შედის STOP C-ს პროგრამების მართვის შემდეგი მოდულები:

- NCD.eHealth.Screening: C ჰეპატიტის სკრინინგი
- NCD.eHealth.Analytics: C ჰეპატიტის ანალიტიკა
- NCD.eHealth.UM: მომხმარებელთა მართვის მოდული
- NCD.eHealth.CM: საერთო მონაცემების მოდულის
- NCD.eHealth.SB: საინტეგრაციო მოდული (RPC)
- NCD.eHealth.Tools: ინსტრუმენტების ბიბლიოთეკა
- NCD.eHealth.SMS: SMS მოდული

6.2 MOH.ElimC

ეს მოდულები წარმოადგენს სოციალური მომსახურების სააგენტოში არსებულ მოდულებს, რომლის საშუალებითაც ხდება C ჰეპატიტის ელიმინაციის პროგრამის განხორციელება და შედგება შემდეგი მოდულებისგან:

- MOH.ElimC.Treatment: მკურნალობა
- MOH.ElimC.SSA: SSA კომისია

⁴ <http://www.ncdc.ge>

- MOH.ElimC.Drugs: მედიკამენტების მართვა
- MOH.ElimC.Monitoring: მონიტორინგი

6.3 MOH.eHealth

საქართველოს შრომის, ჯანმრთელობისა და სოციალური დაცვის ელექტრონული ჯანდაცვის მოდულები წარმოადგენს კომპლექსურ დამოუკიდებელ ეკო-სისტემას, რომლის დანიშნულებაცაა სამინისტროს სხვადასხვა მიმართულების პროცესის ავტომატიზაცია. მოდულების რაოდენობა 30-ზე მეტია, თუმცა STOP C სისტემისთვის ინფორმაციის წყაროსა და სამიზნეს წარმოადგენს შემდეგი მოდულები:

- MOH.eHealth.Billing: ფინანსური მოდული
- MOH.eHealth.Reporting: რეპორტირების მოდული

6.4 სხვა დაკავშირებული მოდულები

ეკოსისტემაში ასევე შედის ორი გარე მოდული, რომლებიც C ჰეპატიტის ბიზნეს პროცესში მონაწილეობენ:

- SafetyBlood: უსაფრთხო სისხლის ელექტრონული მოდული
- LMIS: ლაბორატორიის ინფორმაციული მართვის სისტემა

6.5 ეკოსისტემის სამომავლო განვითარება

აღსანიშნავია, რომ არსებული ინფრასტრუქტურულ-პროგრამული ეკოსისტემა წარმოადგენს დეცენტრალიზებული, ჰეტეროგენული ეკოსისტემის მაგალითს, რომელსაც დადებითთან ერთად გააჩნია უარყოფითი მხარეებიც, რომელთაგან აღსანიშნავია ორი:

1. განმეორებადი (redundant) აპლიკაციები და სერვისები
 - a. ზოგიერთი კრიტიკული მონაცემი და სერვისი მოთავსებულია ფიზიკურად და ხანდახან ტექნოლოგიურად სხვადასხვა გარემოში, რაც ხელს უშლის მონაცემების უნიფიცირებას და ზრდით მათი მომსახურების რესურსებსა და დროს
2. აპლიკაციების დაკავშირების საჭიროება
 - a. მოშორებული აპლიკაციების შემთხვევაში საჭირო ხდება მათი დაკავშირება და რადგან ზოგიერთ აპლიკაციას ერთმანეთისგან სრულიად განსხვავებული ტექნოლოგიური საფუძვლები აქვს, ერთადერთი საშუალებაა ვებ-სერვისები. შედეგად, დიდი მონაცემების მიმოცვლა ხდება ქსელით.

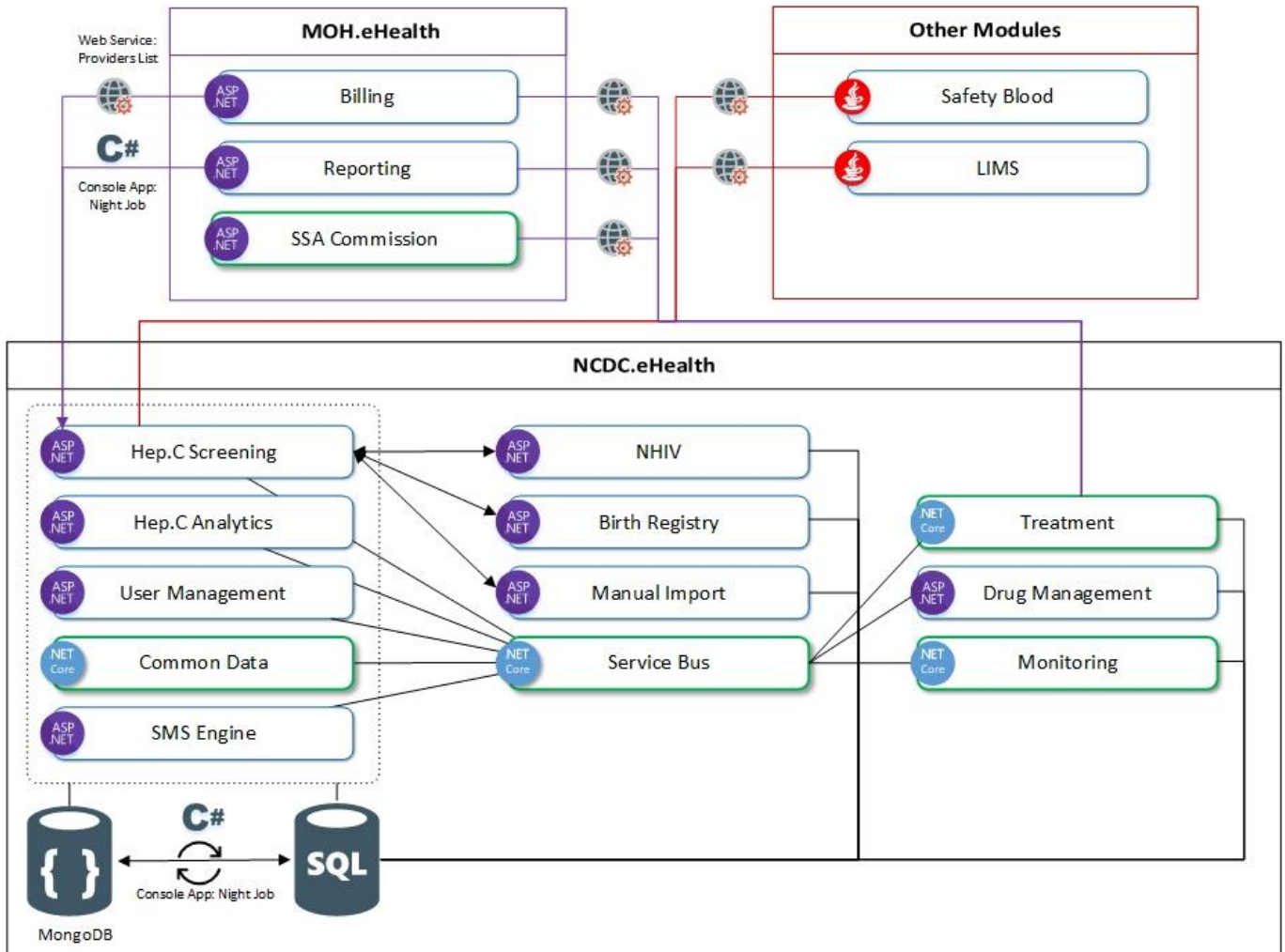
ამ და რამდენიმე სხვა ფაქტორის გათვალისწინებით, შესაძლოა მოხდეს ეკოსისტემის მოდიფიკაცია, სისტემის ცენტრალიზაციისა და ჰომოგენიზაციის მიმართულებით. **სურათი 4-ზე** წარმოდგენილია ეკოსისტემის სამომავლო განვითარების ერთ-ერთი ვარიანტი. არსებული ეკოსისტემისგან განსხვავებით.

1. ერთი ტიპის ფუნქციური აპლიკაციები განთავსებულია ერთ ინფრასტრუქტურულ გარემოში და იყენებენ გაზიარებულ (shared) რესურსებს (ქსელური ინფრასტრუქტურა, მონაცემთა საერთო ბაზები, სერვერული ინფრასტრუქტურა და ა.შ.)
2. ეკოსისტემის ძირითადი კომპონენტები ეფუძნება ერთი და იმავე ან მსგავს ტექნოლოგიებს (Microsoft .NET Framework ⁵, Microsoft .NET Core ⁶).

⁵ <https://www.microsoft.com/net/download/dotnet-framework-runtime>

⁶ <https://docs.microsoft.com/en-us/dotnet/core>

- a. ეს საგრძნობლად ამცირებს ეკოსისტემის როგორც ინფორმაციისა და მონაცემების მიმოცვლისა და დამუშავების დროს, ასევე ადმინისტრირების საოპერაციო დროსა და რესურსებს, რაც უზრუნველყოფს ეკოსისტემის (როგორც ერთი მთლიანი სერვისის) გაზრდილ წარმადობას.



სურათი 2. STOP C პროგრამული ეკოსისტემის სამომავლო განვითარების ვარიანტი

7 STOP C მოდულების ზოგადი აღწერა და დანიშნულება

STOP C მოდული შედგება 2 ძირითადი და 2 დამხმარე ქვე-მოდულისგან:

- ძირითადი მოდულები
 - ცენტრალური მოდული / პორტალი (Portal)
 - STOP C სისტემის ძირითადი ფუნქციონალი, ნავიგაციის ინსტრუმენტები
 - ანალიტიკური მოდული (Analytics)⁷
 - სკრინინგისა და მასთან დაკავშირებული სისტემებისა და მოდულების მონაცემების დინამიური რეპორტირება და ანალიტიკა
 - მომხმარებელთა მართვის მოდული (User Management Module, UMM)
 - სისტემის მომხმარებელთა და წვდომების უფლებების მართვის ინსტრუმენტი

⁷ ანალიტიკური სისტემის არქიტექტურის აღწერა არ წარმოადგენს მიმდინარე დოკუმენტის ნაწილს და წარმოდგენილია ცალკე დოკუმენტში: [STOP C ანალიტიკური სისტემის არქიტექტურა](#)

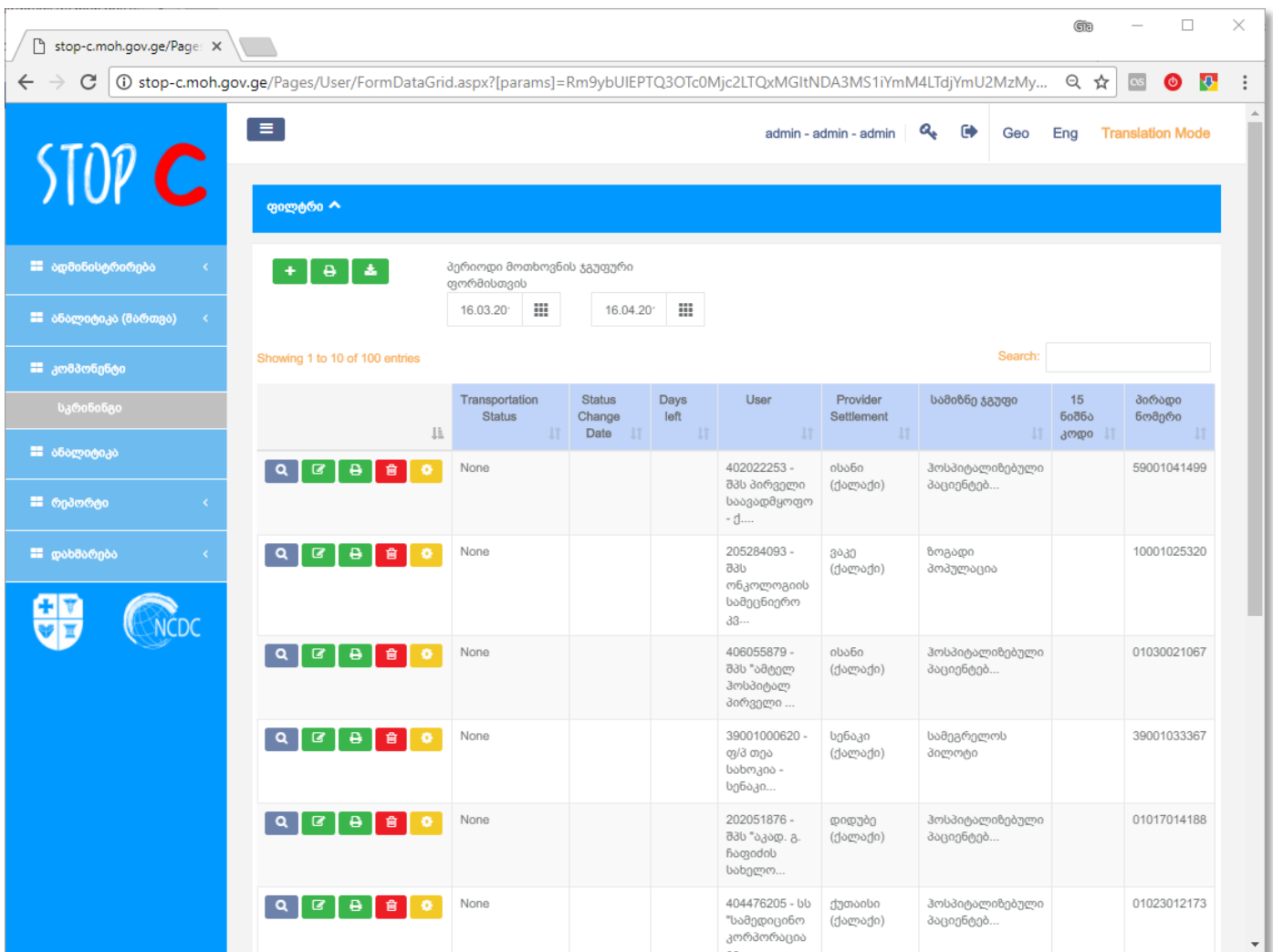
- დამხმარე მოდულები
 - საერთო მონაცემების მოდული (Common Data Module, CDM)
 - მოდულებისთვის საერთო მონაცემების შენახვა და დამუშავება
 - მრავალენოვანი მხარდაჭერა და ლოგირება
 - საკომუნიკაციო მოდული (Service Bus, SB)
 - მოდულებს შორის მონაცემების მიმოცვლის უზრუნველყოფა
 - SMS მოდული
 - ამჟამად არ გამოიყენება და მიმდინარე დოკუმენტში არ არის აღწერილი

7.1 პორტალი

სისტემის ცენტრალურ კომპონენტს წარმოადგენს პორტალი (Portal, HMIS.StopC.Web). პორტალის უმთავრესი ფუნქციაა ვებ-ფორმების დინამიური შექმნის შესაძლებლობა, ფორმების მონაცემების სტრუქტურის შექმნა და მართვა, მონაცემებით ფორმების შევსება და შედეგად C ჰეპატიტის პროგრამის პროცესების ელექტრონული ადმინისტრირება.

7.1.1 ადმინისტრირების პროცესის მოკლე აღწერა

STOP C ადმინისტრირება შესაძლებელია სპეციალური დაშვების მომხმარებლის ავტორიზაციით სისტემაში (admin).



The screenshot shows the STOP C web portal interface. The top navigation bar includes the STOP C logo and a menu with options like 'ადმინისტრირება', 'ანალიტიკა (მართვა)', 'კომპონენტი', 'სკრინინგი', 'ანალიტიკა', 'რეპორტი', and 'დახმარება'. The main content area displays a data grid with the following columns: Transportation Status, Status Change Date, Days left, User, Provider Settlement, სამიზნე ჯგუფი, 15 ნიშნა კოდი, and პირადი ნომერი. The grid shows several entries with details about users and their status.

Transportation Status	Status Change Date	Days left	User	Provider Settlement	სამიზნე ჯგუფი	15 ნიშნა კოდი	პირადი ნომერი
None			402022253 - შპს პირველი საავადმყოფო - ქ...	ისანი (ქალაქი)	პოსპიტალიზებული პაციენტებ...		59001041499
None			205284093 - შპს ონკოლოგიის სამედიცინო კვ...	ვაკე (ქალაქი)	ზოგადი პოპულაცია		10001025320
None			406055879 - შპს "ამტელ პოსპიტალიზებული პირველი ...	ისანი (ქალაქი)	პოსპიტალიზებული პაციენტებ...		01030021067
None			39001000620 - ფ/მ თეა სახოკია - სენაკი...	სენაკი (ქალაქი)	სამედიცინო პაციენტები		39001033367
None			202051876 - შპს "აკად. გ. ჩაფიძის სახელო...	დიდუბე (ქალაქი)	პოსპიტალიზებული პაციენტებ...		01017014188
None			404476205 - სს "სამედიცინო კორპორაცია...	ქუთაისი (ქალაქი)	პოსპიტალიზებული პაციენტებ...		01023012173

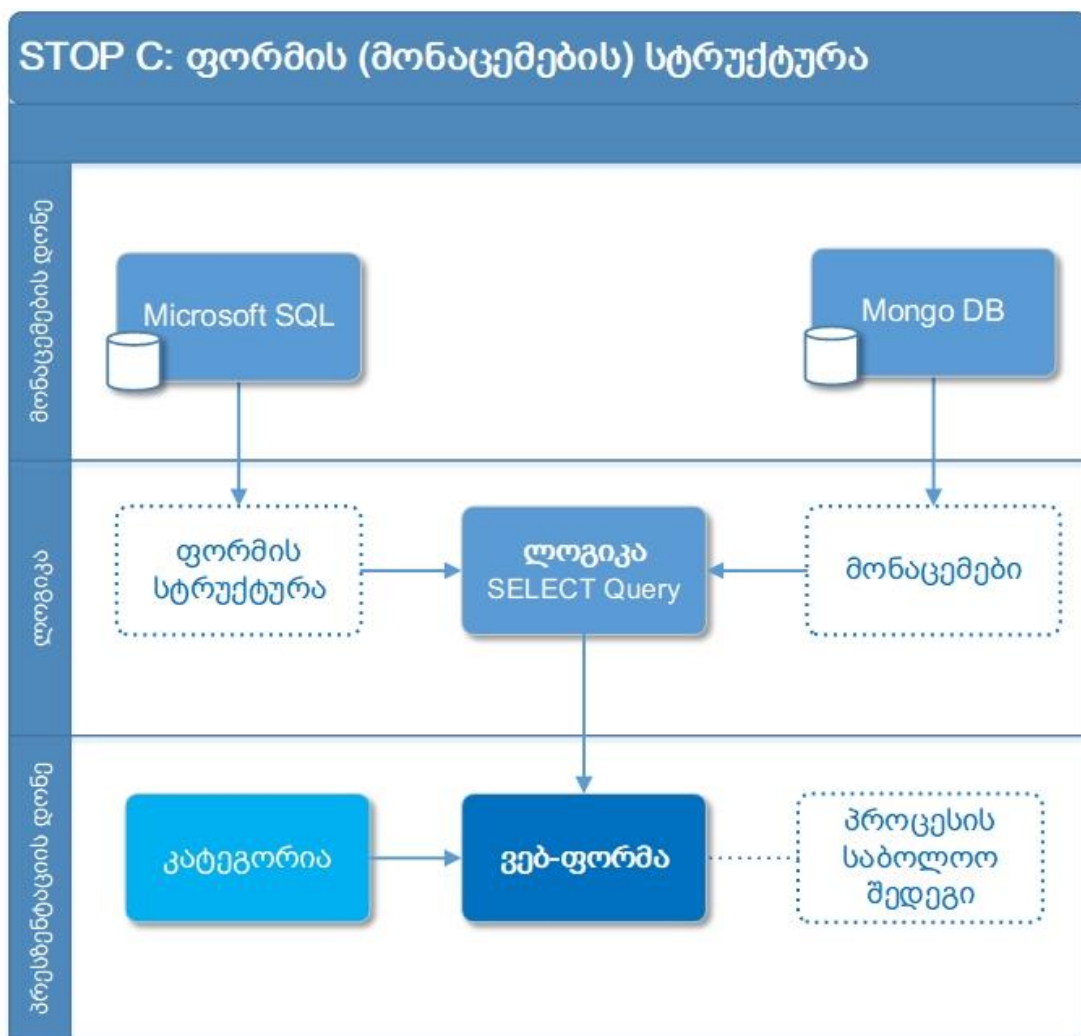
სურათი 3. STOP C ფორმების სამაგალითო გვერდი (პირადი მონაცემების ფორმა)

ავტორიზაციის შემდეგ შესაძლებელია STOP C პორტალის ყველა კომპონენტში შინაარსის, ფორმისა და დიზაინის რედაქტირება. დეტალური ინფორმაციის შეგიძლიათ იხილოთ STOP C ადმინისტრატორის სისტემის ადმინისტრატორის სახელმძღვანელოში⁸. თუმცა, აღსანიშნავია, რომ ცენტრალური ფუნქციონალი (ფორმების დიზაინი, შინაარსი, მონაცემების ანალიტიკა და ვიზუალიზაცია) გულისხმობს ფორმებისა და მათი მონაცემების სპეციფიკურ არქიტექტურას. **სურათი 3** აჩვენებს ამ ფუნქციონალის სამაგალითო ხედს ადმინისტრატორის პანელიდან (პირადი მონაცემების ფორმის მაგალითზე), ხოლო **სურათი 4** წარმოადგენს ამ ინსტრუმენტის უკან მონაცემებისა და პროცესების არქიტექტურას.

სისტემა უზრუნველყოფს ადმინისტრატორის მიერ სასურველი ფორმებისა და პროცესების აგების მექანიზმს. გარდა ფორმების მართვის ინტერფეისისა, სისტემაში რეალიზებულია ფორმების ველების კავშირებისა და ვალიდაციების გამართივებული სინტაქსი (Appendix A: ვალიდაციების სინტაქსი).

სისტემის ადმინისტრირების ძირითადი ელემენტია ვებ-ფორმები, რომლებიც შექმნა, მოდიფიკაცია და წაშლა შესაძლებელია ადმინისტრატორის ინტერფეისით.

ფორმებთან მუშაობის ზოგადი სქემა შემდეგია:

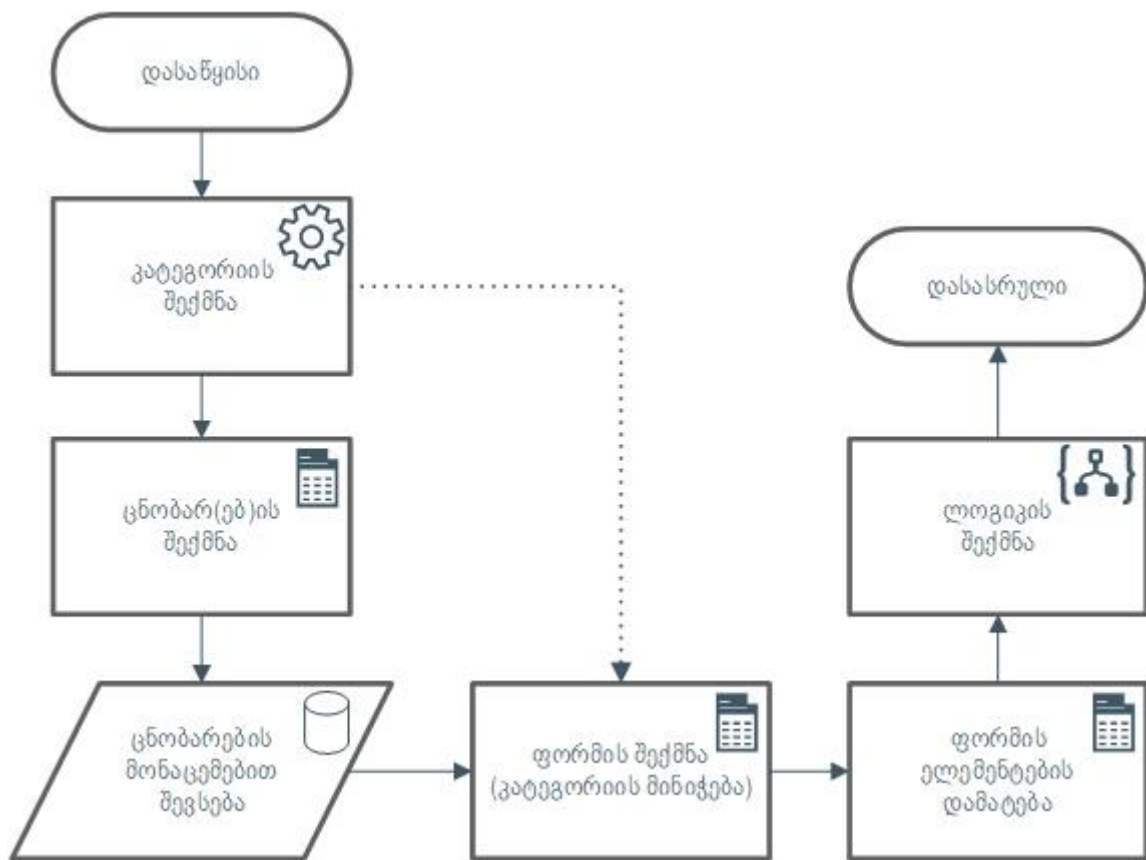


სურათი 4. STOP C ფორმის, მონაცემებისა და პროცესის დიაგრამა

⁸ [STOP C სისტემის ადმინისტრატორის სახელმძღვანელოს ბმული](#)

1. საწყის ეტაპზე საჭიროა ფორმის კატეგორიის შექმნა, რომელიც უნდა განესაზღვროს ყველა მომავალში შექმნილ ფორმას
2. ცნობარების შექმნა ფორმებისთვის
 - a. ცნობარების მონაცემებით შევსება
3. ახალი ფორმის შექმნა
 - a. ფორმის კატეგორიის განსაზღვრა და არჩევა კატეგორიების არსებული სიიდან
 - b. ფორმის ელემენტების დამატება
4. ფორმაში მონაცემების შევსების ლოგიკის განსაზღვრა

სურათი 5 წარმოადგენს ამ პროცესის სტრუქტურულ დიაგრამას, ხოლო **სურათი 4**-ზე მოცემულია იგივე პროცესის კომპონენტების ვიზუალიზაციის სხვადასხვა ჭრილში (მონაცემების, ლოგიკისა და პრეზენტაციის დონეებზე).



სურათი 5. სკრინინგის ფორმის ადმინისტრირების ზოგადი სქემა

დაუმუშავებელი მონაცემების საფუძველზე ანგარიშებისა და შესაბამისი დიაგრამებისა და ცხრილების აგების დეტალური ინსტრუქცია იხილეთ “STOP C: ადმინისტრატორის მომხმარებლის სახელმძღვანელოში“, თავი 11 - რეპორტი. ასევე, “STOP C: ანალიტიკის მომხმარებლის სახელმძღვანელოში“.

7.2 მომხმარებელთა მართვის მოდული

7.2.1 მოდულის ზოგადი აღწერა

მომხმარებელთა მართვის მოდული STOP C სისტემის ნაწილს, რომლის დანიშნულებაც ცენტრალური მოდულის მომხმარებლებისა და წვდომების ადმინისტრირების პროცესის მართვა. კერძოდ, ის განსაზღვრავს მომხმარებლებს, მათ როლებს და დაშვების დონეებს, რითაც

უზრუნველყოფს მათ ვალიდაციის მექანიზმის გამოყენებით, რათა არ მოხდეს ცენტრალურ მოდულში არავტორიზებული და არასანქცირებული დაშვება. ამ მოდულში განთავსებულია მომხმარებელთა საერთო ნუსხა, მინიჭებული როლები, დაშვებები და შეზღუდვები. მომხმარებელთა მართვის მოდული აღჭურვილია ერთჯერადი (გამარტივებული) ავტორიზაციის მექანიზმით. იგი თავის თავში აერთიანებს ინტერფეისებს, რომელთა საშუალებითაც ავტორიზებული მომხმარებელი განმეორებითი ავტორიზაციის გარეშე ინარჩუნებს შესაძლებლობას იმუშაოს სხვა, მისთვის განკუთვნილ მოდულებთან შესაბამისი დაშვებებით.

7.2.2 მოდულის ფუნქციონალის აღწერა

მომხმარებელთა მართვის მოდულის ზოგადი ფუნქციონალი მოიცავს შემდეგ პროცესებს:

- მოდულში მომხმარებლების ერთიანი ნუსხის ფორმირება, მართვა და კონტროლი. აღნიშნული ნუსხა წარმოადგენს იმ პიროვნებებისა და/ან ორგანიზაციების სიას, რომლებსაც ადმინისტრატორის დაშვებით აქვთ წვდომა ცენტრალურ მოდულთან
- საჭიროების შემთხვევაში, ცენტრალური მოდულის მიმდინარე ან სამომავლო სპეციფიკიდან გამომდინარე როლების/მომხმარებლების ჯგუფების ფორმირება და შესაბამისი მომხმარებლების განწევრიანება აღნიშნულ ჯგუფებში. როლების ფორმირების ამოცანა მოიცავს შექმნილი როლისთვის იმ რესურსების მკაცრად განსაზღვრას, რომლიც იქნება აღნიშნული როლით მოსარგებლე მომხმარებლის განკარგულებაში როლის შესაბამისი საქმიანობის წარმართვისათვის.
- ჯგუფებზე რესურსების მიხედვით უფლებების განწერა. მომხმარებლისთვის საჭირო დამატებითი ატრიბუტების შექმნა და შესაბამისი მნიშვნელობების მინიჭება.
- STOP C-ს ფარგლებში რეალიზებული მოდულ(ებ)ისთვის ნებისმიერი ინფორმაციული სახის შეტყობინებების რეგისტრაცია.

7.3 ანალიტიკური მოდული

ანალიტიკური მოდულის საშუალებით ხდება სკრინინგის სისტემის მონაცემებისა და სხვადასხვა დაკავშირებული სისტემების მონაცემების გაერთიანებული დინამიური რეპორტირება და ანალიზი სხვადასხვა ჭრილში. დინამიურობა გულისხმობს ფუნქციონალს, რომლის შესაძლებლობითაც მომხმარებელს (ადმინისტრატორს) შეუძლია როგორც არსებული რეპორტებისა და ვიზუალური გამოსახულებების ცვლილება, ასევე ახალი რეპორტებისა და დიაგრამების შექმნა არსებული მონაცემების საფუძველზე. ამისთვის საჭირო არ არის დეველოპერის ჩართულობა. დეტალური ინსტრუქცია აღწერილია ანალიტიკის ადმინისტრატორის მომხმარებლის სახელმძღვანელოში ⁹, ხოლო ანალიტიკის სისტემის არქიტექტურა აღწერილია შესაბამის ცალკე დოკუმენტში ¹⁰.

7.4 საკომუნიკაციო მოდული

STOP C სისტემის მოდულებს შორის მონაცემების მიმოცვლა რეალიზებულია შიდა სერვისული გამტარის (*internal service bus*) მეშვეობით. ეს კომპონენტი მოქმედებს, როგორც Remote Method Call, HTTP-ის გავლით. სისტემის ერთ-ერთი კომპონენტის საკომუნიკაციო მოდული (HMIS.messaging.web), რომლის მეშვეობითაც ყველა მოდული აგზავნის შეტყობინებებს და მოთხოვნებს სხვა მოდულებთან, და არა პირდაპირ. ყოველი გაგზავნილი შეტყობინება უნდა შეიცავდეს შემდეგ მონაცემებს:

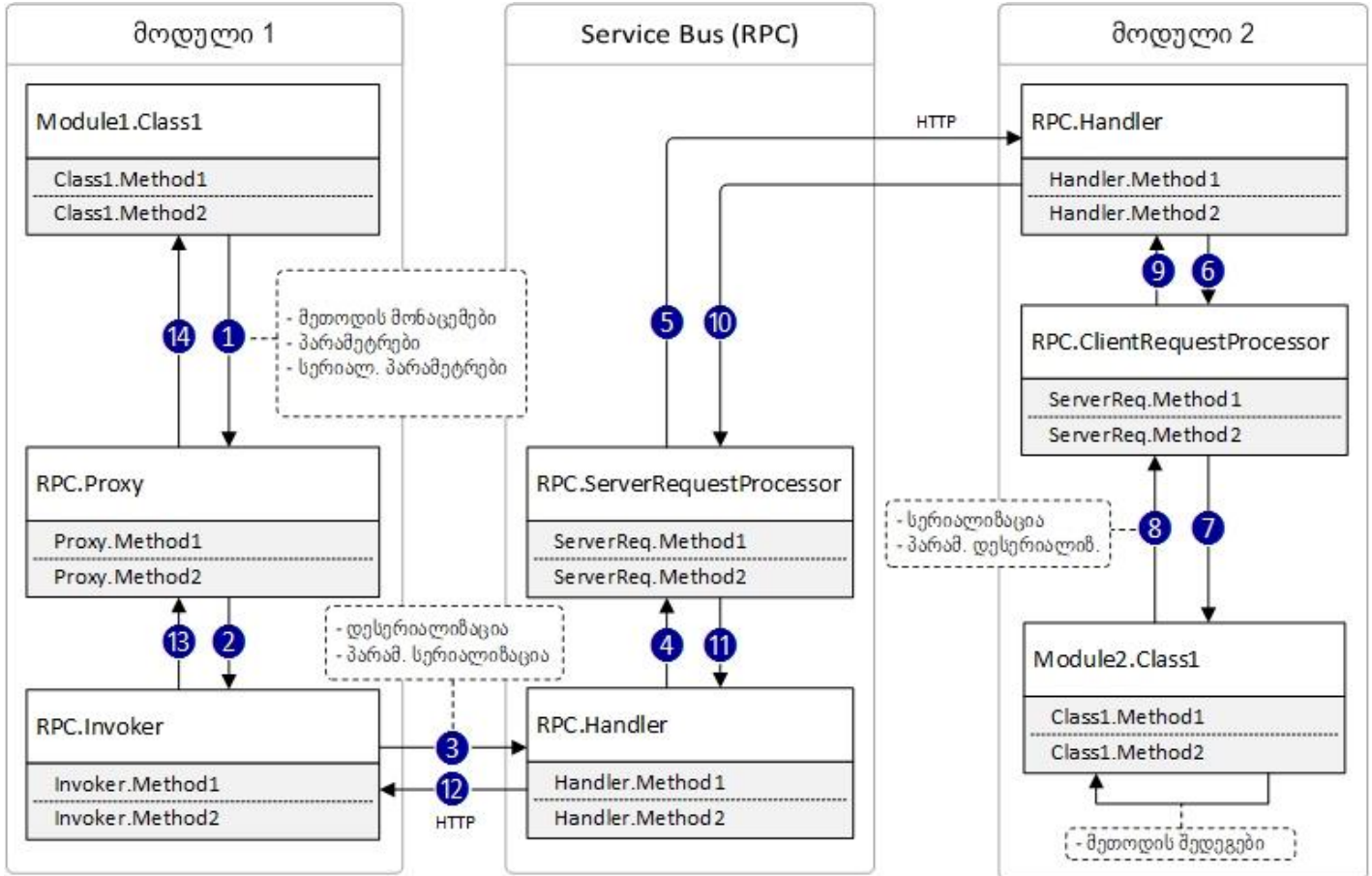
- იმ მეთოდის მონაცემები, რომელიც უნდა გაეშვას კონკრეტულ მოდულში

⁹ ანალიტიკის ადმინისტრატორის სახელმძღვანელოს ბმული

¹⁰ STOP C მონაცემების ანალიტიკის სისტემის არქიტექტურა

○ ფორმატი: მოდულის_სახელი.კლასის_სახელი.მეთოდის_სახელი

- მეთოდის პარამეტრები
- მეთოდის სერიალიზებული პარამეტრები, (*BinaryFormatter-ის მეშვეობით*)



სურათი 6. საკომუნიკაციო მოდულის მუშაობის პრინციპის დიაგრამა

საკომუნიკაციო მოდული კითხულობს მოდულის სახელს მოთხოვნაში და გადასცემს მოთხოვნას შესაბამის მოდულს. როდესაც სამიზნე მოდული იღებს მოთხოვნას საკომუნიკაციო მოდულისგან, ის ცდილობს მოძებნოს ის კლასი და ის მეთოდი იმ კლასში, ომელიც მითითებულია შეტყობინებაში და ცდილობს მის გაშვებას. შემდეგ, გაშვების შედეგები ბრუნდება საკომუნიკაციო მოდულში, რომელიც გადასცემს მას მომთხოვნ მოდულს. სისტემაში მონაცემების მიმოცვლის პროცესის სქემატური დეტალებისთვის იხ. **სურათი 6**.

ამ მიზნისთვის, ანუ მოდულებს შორის მონაცემების მიმოცვლისთვის გამოიყენება ე.წ. პროექსი-კლასები სწორედ მათი საშუალებით ხდება მოთხოვნების და შედეგების მოდულებს შორის გაცვლა. შეტყობინებები, რომლებიც მიმოიცვლება, შეიცავენ ორობით მონაცემებს, სერიალიზებულს *BinaryFormatter-ის* მეშვეობით და არანაირი სხვა სტანდარტი (მაგ.: SOAP) არ გამოიყენება.

7.5 საერთო მონაცემების მოდული

საერთო მონაცემების მოდული (CDM, HMIS.commondata.web) წარმოადგენს საყოველთაოდ მოხმარებადი ელექტრონული სერვისების ერთობლიობას, რომელიც შეიცავს ისეთი ტიპის სერვისებს და მონაცემებს, რომელიც საერთოა და გამოიყენება STOP C სისტემის სხვა მოდულების

მიერ. ასეთი მონაცემები შეიძლება იყოს სატელეფონო ინდექსები, დასახლებული პუნქტის კოდები, მრავალენოვანი მხარდაჭერის ინტერფეისებზე ნათარგმნი ტექსტები და სხვა.

მოდულს გააჩნია ინტერფეისი, რომლის მეშვეობითაც შესაძლებელია სიტყვების სხვადასხვა ენაზე თარგმანების ბაზაში დამატება/ცვლილება. საჭიროების შემთხვევაში CDM-ში ასევე შეიძლება მოხდეს STOP C-ის ყველა მოდულის ვებ გვერდებზე მიმართვების ლოგირება (რომელ გვერდს რომელმა მომხმარებელმა მიმართა, როდის და რამდენი ხანი დაჭირდა ვებ სერვერს შედგის დასაბრუნებლად). არის შესაბამისი ინტერფეისიც, რომელშიც მომხმარებელს შეუძლია მოცემული ინფორმაციის ვიზუალურად დათვალიერება.

ინტერფეისები განლაგებულია შემდეგ მისამართებზე:

- თარგმნის ინტერფეისი:
<http://birthregistry.moh.gov.ge/HMIS/HMIS.CommonData.web/Pages/TranslationsList.aspx>
- ლოგების ინტერფეისი:
<http://birthregistry.moh.gov.ge/HMIS/HMIS.CommonData.web/Pages/ActionLogViewer.aspx>

8 პლატფორმა და გამოყენებული ტექნოლოგიები

სისტემის ყველა მოდულის არქიტექტურის ტიპები და ტექნოლოგიები ძირითადად საერთოა და დაფუძნებულია შემდეგ პლატფორმაზე და ტექნოლოგიებზე:

- Microsoft .NET Framework 4.5.1 ¹¹
 - STOP C-ისთვის Microsoft .NET წარმოადგენს პროგრამულ გარემოს, რომელიც შეიცავს კლასების დიდ ბიბლიოთეკას (Framework Class Library - FCL) და უზრუნველყოფს ინტეროპერაბერულობას. .NET Framework-ისთვის დაწერილი კოდი ეშვება პროგრამული გარემოში (განსხვავებით აპარატურული გარემოსგან) - Common Language Runtime-ში (CLR), რომელიც წარმოადგენს აპლიკაციის ვირტუალურ მანქანას და უზრუნველყოფს ისეთ სერვისებს, როგორებიცაა უსაფრთხოება, მესხიერების მართვა და გამონაკლისების დამუშავება. FCL და CLR ერთად ქმნიან Microsoft .NET Framework-ს. STOP C-ში რეალიზებულია ერთ-ერთი ბოლო, სტაბილური ვერსია 4.5.1
- Microsoft ASP.NET WebForms ¹²
 - Microsoft ASP.NET უზრუნველყოფს STOP C დინამიურ არქიტექტურას, რაც იძლევა შესაძლებლობას მარტივად, დამატებითი დეველოპმენტის გარეშე შეიქმნას სასურველი ვებ-გვერდები (ფორმები).
- CITI.EVO.TwoWayModelBinding – MVC-ის ¹³ ანალოგი ASP.NET WebForms-თვის
 - ვებ-აპლიკაციების გარემო, რომელიც დაფუძნებულია Model-view-controller მეთოდოლოგიაზე.

¹¹ <https://www.microsoft.com/en-us/download/details.aspx?id=40779>

¹² <https://www.asp.net/web-forms>

¹³ [https://msdn.microsoft.com/en-us/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx)

- NHibernate ¹⁴
 - NHibernate წარმოადგენს ე.წ. ობიექტურ-რელაციურ მაკავშირებელს (Object-Relational Mapper - ORM), რაც გულისხმობს STOP C-ის მონაცემების „თარგმნას“ .NET გარემოდან თითქმის ნებისმიერი რელაციური ბაზისთვის გასაგებ ფორმატში (Oracle DB, IBM DB2, mysql, postgre SQL და ა.შ.). შედეგად, STOP C არ არის დამოკიდებული მონაცემების კონკრეტულ ბაზაზე და შესაძლებელი ნებისმიერი რელაციური ბაზის გამოყენება.
- Microsoft SQL ¹⁵
 - Microsoft SQL წარმოადგენს STOP C-ის ე.წ. მეტა-მონაცემების საცავს, რაც გულისხმობს ინფორმაციას მონაცემების სტრუქტურის შესახებ და არა თვითონ მონაცემებს.
- MongoDB ¹⁶
 - MongoDB ასრულებს უშუალოდ მონაცემების საცავის როლს. ის წარმოადგენს არა რელაციურს, არამედ დოკუმენტზე ორიენტირებულ ბაზას (document-oriented), რაც STOP C-ის შემთხვევაში იძლევა შესაძლებლობას შეიქმნას დინამიური ვებ-გვერდები და შესაბამისი მონაცემებიც შეიქმნას და დამუშავდეს დინამიურად, დამატებითი დეველოპმენტის გარეშე, ადმინისტრატორის მიერ. ბაზა ეფუძნება NoSQL ¹⁷ ბაზების ერთ-ერთ ყველაზე გავრცელებულ პრინციპს - key-value ¹⁸ წყვილის პარადიგმას.

¹⁴ <http://nhibernate.info>

¹⁵ <https://www.microsoft.com/en-us/sql-server/sql-server-2017>

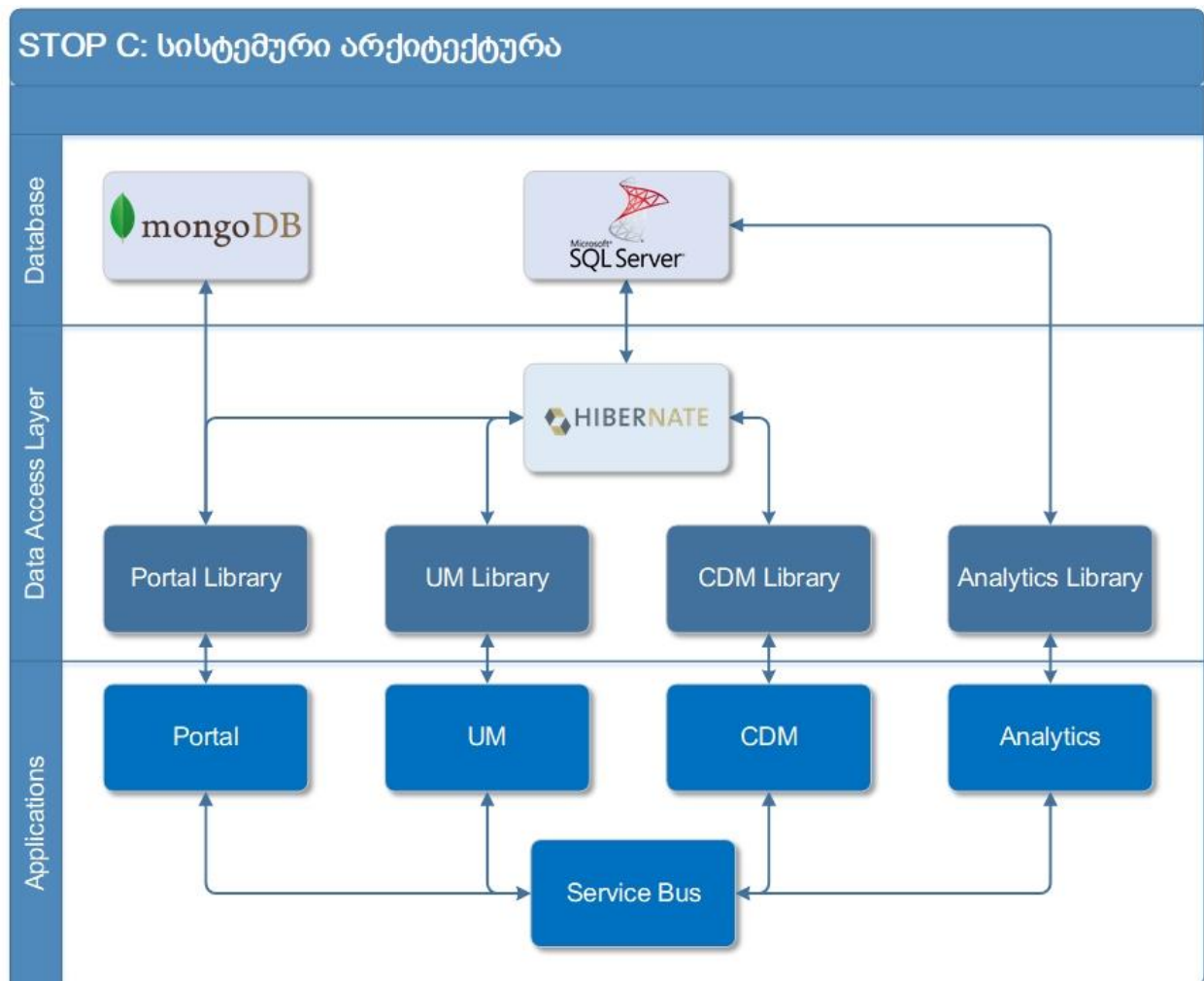
¹⁶ <https://www.mongodb.com>

¹⁷ <http://nosql-database.org>

¹⁸ https://www.wikiwand.com/en/Key-value_database

9 სისტემის არქიტექტურა

სურათი წარმოადგენს STOP C სკრინინგის ელექტრონული აღრიცხვის სისტემის კომპონენტების ლოგიკურ დიაგრამას აპლიკაციებისა და მათთან დაკავშირებული მონაცემთა ბაზების ჭრილში.



სურათი 7. STOP C კომპონენტების არქიტექტურა

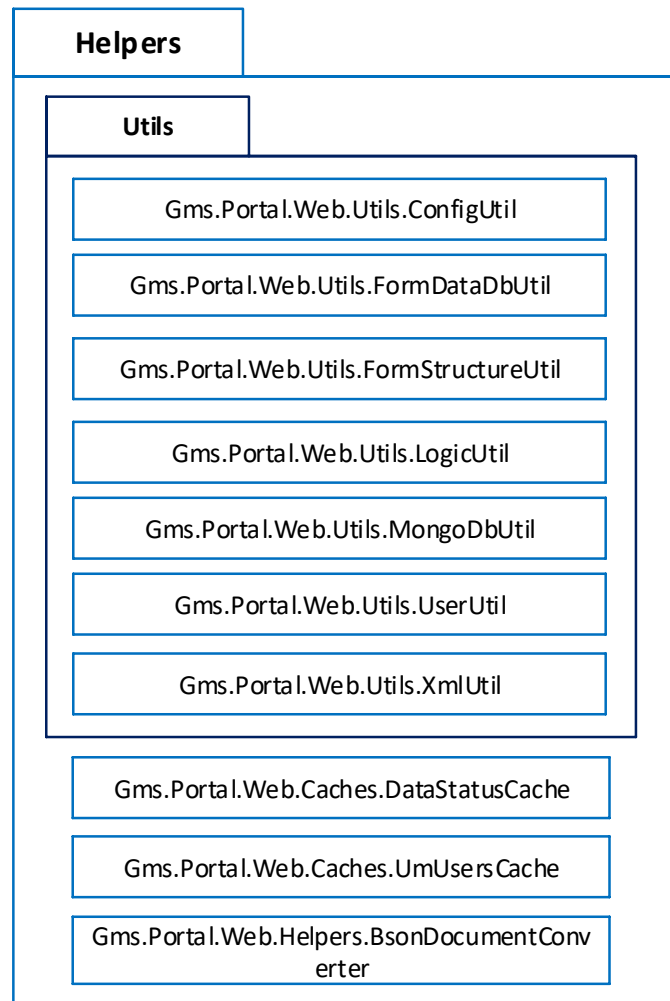
9.1 ცენტრალური მოდული (პორტალი)

9.1.1 კლასები

პორტალის კლასები როლებისა და ფუნქციების მიხედვით დაჯგუფებულია რამდენიმე კატეგორიად:

- Helper კლასები (სურათი 8, ცხრილი 1): დამხმარე კლასები
- Converter კლასები (სურათი 9, ცხრილი 2):
 - Entity to model converter კლასები: მოცემული კლასები აკონვერტირებს მონაცემთა ბაზის ცხრილებთან შესაბამის კლასებს (ე.წ DB Entity-ებს) Model კლასებში
 - Model to entity converter კლასები: მოცემული კლასები აკონვერტირებს Model კლასებს მონაცემთა ბაზის ცხრილებთან შესაბამის კლასებში (ე.წ DB Entity-ებს)
- Entity კლასები (სურათი 10, ცხრილი 3): შეიცავს სხვადასხვა მონაცემების, სტრუქტურისა და ა.შ. მატარებელი კლასების ჯგუფს

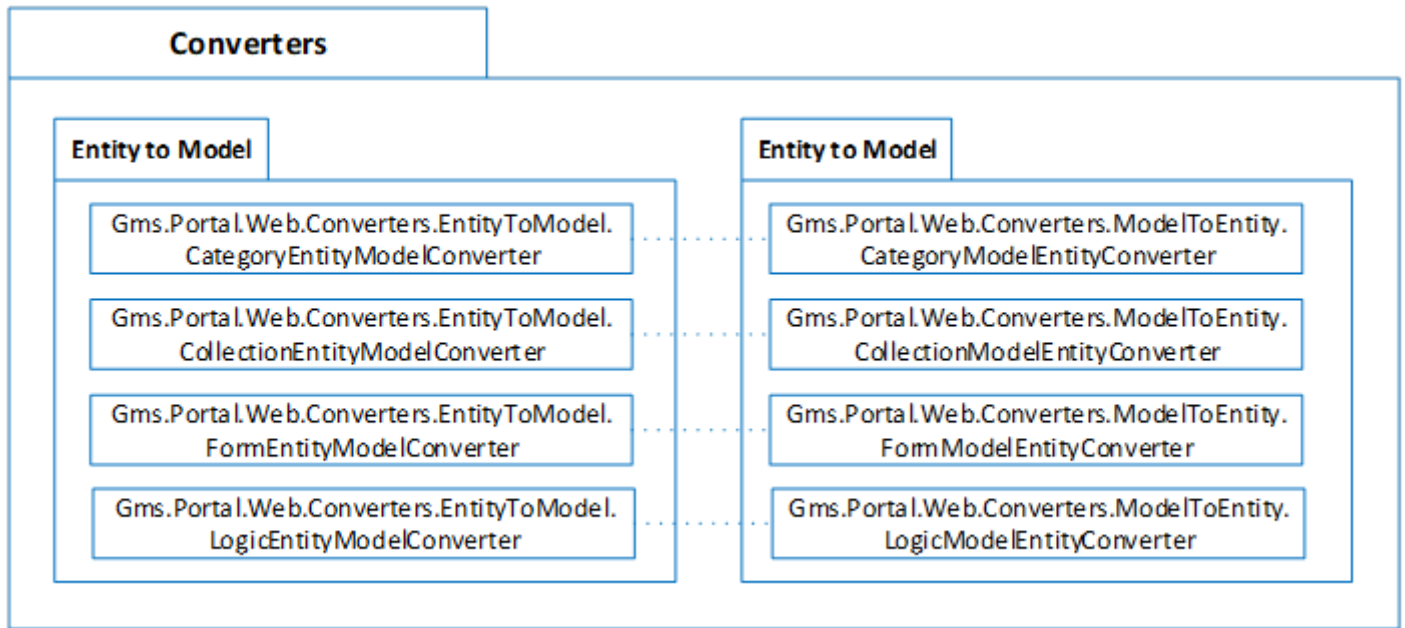
- Model კლასები (სურათი 11, ცხრილი 4): მომხმარებლის ინტერფეისთან შესაბამისი ველების მქონე კლასები. ყველა ამ კლასის სტრუქტურა ემთხვევა შესაბამის კონტროლზე არსებულ ელემენტებს.



სურათი 8. პორტალის Helper-კლასების დიაგრამა (UML)

ცხრილი 1. Helper კლასები

სკლასის სრული დასახელება	ბიბლიოთეკა	კლასის აღწერა
Gms.Portal.Web.Caches.DataStatusCache	App_Code	მონაცემთა სტატუსები ქეში
Gms.Portal.Web.Caches.UmUsersCache	App_Code	მომხმარებლების ქეში
Gms.Portal.Web.Helpers.BsonDocumentConverter	App_Code	დამხმარე კლასი რომელიც აკონვერტირებს BsonDocument-ს FormDataUnit-ში ან Dictionary-ში და პირიქით

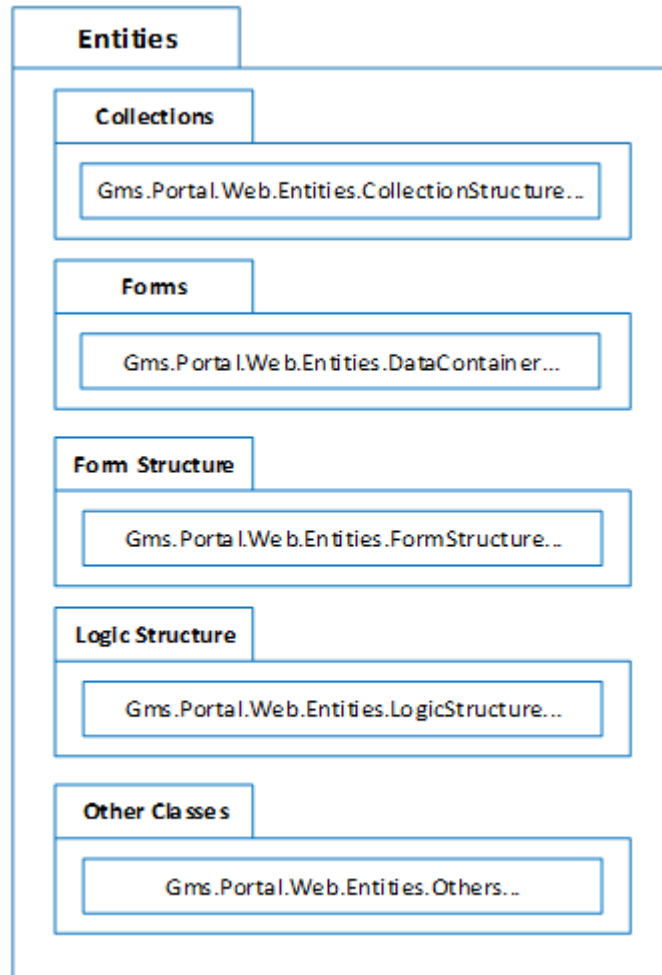


სურათი 9. Converter კლასები

ცხრილი 2. Converter კლასები: Entity to Model და Model to Entity

სკლასის სრული დასახელება	ბიბლიოთეკა	კლასის აღწერა
Gms.Portal.Web.Converters.EntityToModel		Namespace შეიცავს კონვერტერ კლასებს რომლებიც მონაცემთა ბაზის სტრუქტურულ ერთეულებს (DB entity) აკონვერტირებს ინტერფეისის მოდელ კლასებში
Gms.Portal.Web.Converters.EntityToModel.CategoryEntityModelConverter	App_Code	კატეგორიების კონვერტერი
Gms.Portal.Web.Converters.EntityToModel.CollectionEntityModelConverter	App_Code	კოლექციების (ცნობარების სტრუქტურის) კონვერტერი
Gms.Portal.Web.Converters.EntityToModel.FormEntityModelConverter	App_Code	ფორმის (ფორმის სტრუქტურის) კონვერტერი
Gms.Portal.Web.Converters.EntityToModel.LogicEntityModelConverter	App_Code	ლოგიკების კონვერტერი
Gms.Portal.Web.Converters.ModelToEntity		Namespace შეიცავს კონვერტერ კლასებს რომლებიც ინტერფეისის მოდელ კლასებს აკონვერტირებს მონაცემთა ბაზის სტრუქტურულ ერთეულებში (DB entity)
Gms.Portal.Web.Converters.ModelToEntity.CategoryModelEntityConverter	App_Code	კატეგორიების კონვერტერი
Gms.Portal.Web.Converters.ModelToEntity.CollectionModelEntityConverter	App_Code	კოლექციების (ცნობარების სტრუქტურის) კონვერტერი

სკლასის სრული დასახელება	ბიბლიოთეკა	კლასის აღწერა
Gms.Portal.Web.Converters.ModelToEntity. FormModelEntityConverter	App_Code	ფორმის (ფორმის სტრუქტურის) კონვერტერი
Gms.Portal.Web.Converters.ModelToEntity. LogicModelEntityConverter	App_Code	ლოგიკების კონვერტერი



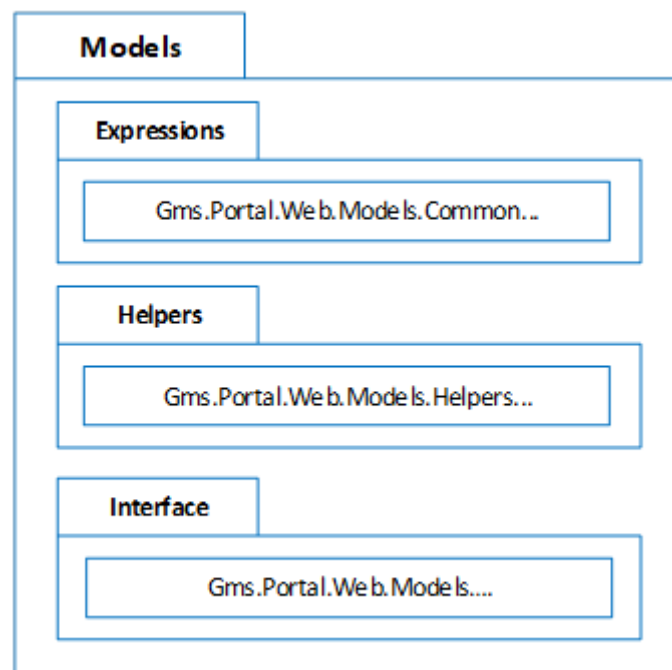
სურათი 10. Entity კლასები

ცხრილი 3. Entity კლასები

კლასის სრული დასახელება	ბიბლიოთეკა	კლასის აღწერა
Gms.Portal.Web.Entities.CollectionStructure	App_Code	შეიცავს კოლექციების (ცნობარების) სტრუქტურის მატარებელ კლასებს (რომლებიც შემდეგ ინახება XML-ის სახით GM_Collections.XmlData სვეტში)
Gms.Portal.Web.Entities.CollectionStructure.C ollectionEntity	App_Code	კოლექციის (ცნობარის) მონაცემები
Gms.Portal.Web.Entities.CollectionStructure.F ieldEntity	App_Code	კოლექციის (ცნობარის) ველების მონაცემები

კლასის სრული დასახელება	ბიბლიოთეკა	კლასის აღწერა
Gms.Portal.Web.Entities.DataContainer.Form DataBase	App_Code	ფორმის მონაცემების შემცველი საბაზისო კლასი
Gms.Portal.Web.Entities.DataContainer.Form DataUnit	App_Code	ფორმის მონაცემების შემცველი კლასი
Gms.Portal.Web.Entities.DataContainer.Form DataBaseList	App_Code	ფორმის მონაცემების სიის შემცველი კლასი
Gms.Portal.Web.Entities.DataContainer.Form DataLazyList	App_Code	ფორმის მონაცემების სიის შემცველი კლასი (საჭიროებისამებრ ჩატვირთვის ფუნქციონალით)
Gms.Portal.Web.Entities.DataContainer.Form DataListRef	App_Code	Reference ტიპის კლასი, რომელიც შეიცავს ბმულს ფორმის მონაცემების სიაზე
Gms.Portal.Web.Entities.FormStructure	App_Code	შეიცავს ფორმების სტრუქტურის მატარებელ კლასებს (რომელიც შემდეგ ინახება XML-ის სახით GM_Forms.XmlData სვეტში)
Gms.Portal.Web.Entities.FormStructure.ContentEntity	App_Code	კონტროლის საბაზისო კლასი (რომელსაც შეიძლება ყავდეს შვილი კონტროლები)
Gms.Portal.Web.Entities.FormStructure.ControlEntity	App_Code	კონტროლის საბაზისო კლასი
Gms.Portal.Web.Entities.FormStructure.FieldEntity	App_Code	ველის დეტალური ინფორმაციის მატარებელი კლასი
Gms.Portal.Web.Entities.FormStructure.FormEntity	App_Code	ფორმის ინფორმაციის მატარებელი კლასი
Gms.Portal.Web.Entities.FormStructure.GridEntity	App_Code	ცხრილი კლასი
Gms.Portal.Web.Entities.FormStructure.GroupEntity	App_Code	კონტროლების ჯგუფის კლასი
Gms.Portal.Web.Entities.FormStructure.TabContainerEntity	App_Code	ტაბების კონტეინერის კლასი
Gms.Portal.Web.Entities.FormStructure.TabPageEntity	App_Code	ტაბ გვერდის კლასი
Gms.Portal.Web.Entities.LogicStructure	App_Code	შეიცავს ლოგიკების სტრუქტურის მატარებელ კლასებს (რომლებიც შემდეგ ინახება XML-ის სახით GM_Logics.RawData სვეტში)
Gms.Portal.Web.Entities.LogicStructure.ExpressionEntity	App_Code	ლოგიკის გამოსახულების კლასი
Gms.Portal.Web.Entities.LogicStructure.Logic ContainerEntity	App_Code	გამოსახულებების კონტეინერის საბაზისო კლასი

კლასის სრული დასახელება	ბიბლიოთეკა	კლასის აღწერა
Gms.Portal.Web.Entities.LogicStructure.LogicEntity	App_Code	ძირითადი მატარებელი კლასი, რომელიც შეიცავს ტექსტურ ბრძანებას ან ლოგიკას
Gms.Portal.Web.Entities.LogicStructure.LogicExpressionsEntity	App_Code	გამოსახულებების მატარებელი კლასი
Gms.Portal.Web.Entities.LogicStructure.LogicQueryEntity	App_Code	ტექსტური ბრძანების შემცველი კლასი
Gms.Portal.Web.Entities.LogicStructure.Name dExpressionEntity	App_Code	სახელის მქონე გამოსახულების კლასი
Gms.Portal.Web.Entities.Others.ControlTreeEntity	App_Code	კონტროლის შესახებ მონაცემების მატარებელი დამხმარე კლასი
Gms.Portal.Web.Entities.Others.ElementTreeNodeEntity	App_Code	ელემენტის შესახებ მონაცემების მატარებელი დამხმარე კლასი
Gms.Portal.Web.Entities.Others.TreeNodeEntity	App_Code	ხის შესახებ მონაცემების მატარებელი დამხმარე კლასი



სურათი 11. Model კლასები

ცხრილი 4. Model კლასების აღწერა

კლასის სრული დასახელება	ბიბლიოთეკა	კლასის აღწერა
Gms.Portal.Web.Models		ეს Namespace შეიცავს მომხმარებლის ინტერფეისის მოდელებს (შედგად, ყველა კლასის სტრუქტურა ემთხვევა შესაბამის კონტროლზე არსებულ ელემენტებს)
Gms.Portal.Web.Models.Common.Expression Model	App_Code	ლოგიკის გამოსახულებების კლასი
Gms.Portal.Web.Models.Common.Expression sListModel	App_Code	ლოგიკის გამოსახულებების სიის კლასი
Gms.Portal.Web.Models.Common.Expression sLogicModel	App_Code	ლოგიკის გამოსახულებების დამატებული კლასი (Filter, Group By, Order By, Select გამოსახულებები ერთად)
Gms.Portal.Web.Models.Common.NamedExpr essionModel	App_Code	სახელის მქონე გამოსახულებები კლასი (მაგ. (A + B) AS Amount)
Gms.Portal.Web.Models.Common.NamedExpr essionsListModel	App_Code	სახელის მქონე გამოსახულებების ნუსხის კლასი
Gms.Portal.Web.Models.Common.NamedMod el	App_Code	სისტემის მარტივი ელემენტების (მაგ. კატეგორიები) რედაქტირების კლასი, რომელსაც შეიძლება ჰქონდეს მაქსიმუმ 2 ველი (სახელი და მშობლის ID)
Gms.Portal.Web.Models.Helpers.CategoriesF ormsModel	App_Code	კატეგორიების და ფორმების მენიუში ერთობლივად გამოტანისთვის გათვალისწინებული მოდელი (შეიცავს მომდევნო კლასების სიას)
Gms.Portal.Web.Models.Helpers.CategoryFor mModel	App_Code	კატეგორიების და ფორმების მენიუში ერთობლივად გამოტანისთვის გათვალისწინებული მოდელი
Gms.Portal.Web.Models.Helpers.ElementNod esModel	App_Code	ფორმის ელემენტების ხისებურად დალაგების და გამოტანისთვის გათვალისწინებული მოდელი (შეიცავს ElementTreeNodeEntity კლასის ნუსხას)
Gms.Portal.Web.Models.Helpers.FieldUnitsMo del	App_Code	შეიცავს კოლექციების (ცნობარების) ველების ნუსხას (ცხრილში გამოსატანად)
Gms.Portal.Web.Models.CategoriesModel	App_Code	შეიცავს კატეგორიების ნუსხას
Gms.Portal.Web.Models.CategoryModel	App_Code	შეიცავს კატეგორიის შესახებ ინფორმაციას
Gms.Portal.Web.Models.CollectionDataModel	App_Code	შეიცავს ცნობარის ერთი სტრიქონი/ჩანაწერის მონაცემების
Gms.Portal.Web.Models.CollectionDatasMode l	App_Code	შეიცავს ერთი ცნობარის მონაცემებს (CollectionDataModel-ების ნუსხას)
Gms.Portal.Web.Models.CollectionModel	App_Code	შეიცავს ცნობარის სტრუქტურის მონაცემებს
Gms.Portal.Web.Models.CollectionsModel	App_Code	შეიცავს ცნობარების ნუსხას
Gms.Portal.Web.Models.ElementModel	App_Code	შეიცავს ფორმის ელემენტის მონაცემებს

კლასის სრული დასახელება	ბიბლიოთეკა	კლასის აღწერა
Gms.Portal.Web.Models.FormDataGridModel	App_Code	შეიცავს ფორმის მონაცემების ცხრილს
Gms.Portal.Web.Models.FormDataModel	App_Code	შეიცავს ფორმის ერთი ჩანაწერის მონაცემებს
Gms.Portal.Web.Models.FormModel	App_Code	შეიცავს ფორმის სტრუქტურას
Gms.Portal.Web.Models.FormsModel	App_Code	შეიცავს ფორმების ნუსხას
Gms.Portal.Web.Models.LogicModel	App_Code	შეიცავს ლოგიკის მონაცემებს
Gms.Portal.Web.Models.LogicsModel	App_Code	შეიცავს ლოგიკების ნუსხას
Gms.Portal.Web.Models.RecordStatusModel	App_Code	შეიცავს ფორმის ჩანაწერის სტატუსის მონაცემებს
Gms.Portal.Web.Models.TableDataModel	App_Code	ამ ეტაპზე არ გამოიყენება, მაგრამ შეიცავს სამომავლოდ გამოსაყენებელ მეთოდებს.

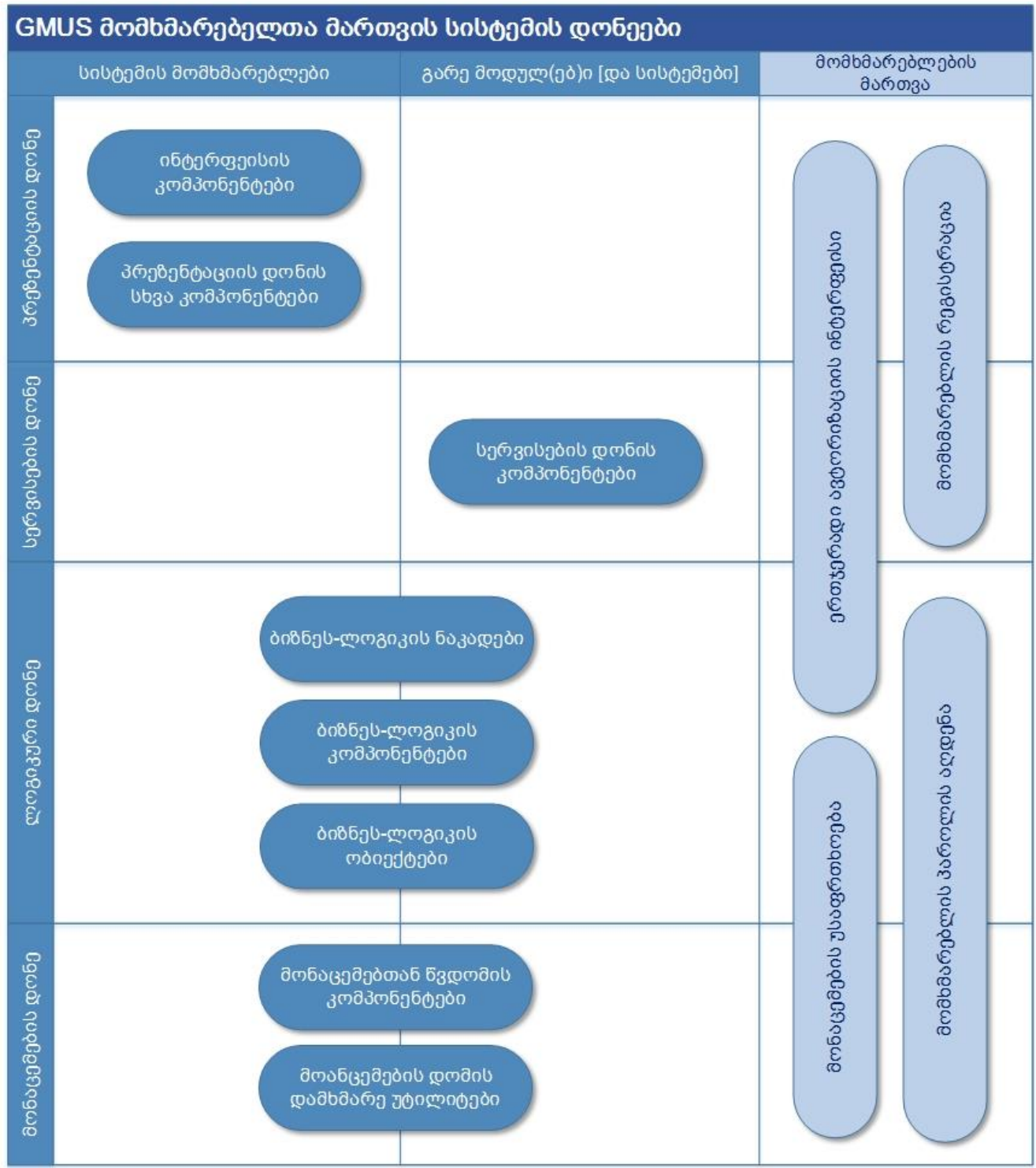
9.1.2 ფიზიკური არქიტექტურა

პორტალის ფიზიკური არქიტექტურა წარმოდგენილია აპლიკაციისა (Web) და მონაცემთა ბაზის (DB) ორი სერვერით. მათი ვირტუალური და ფიზიკური მისამართები მოცემულია ქვემოთ:

- Web სერვერი: <http://STOP-C.moh.gov.ge> 172.17.216.53
- MogoDB სერვერი: 172.17.7.132
- Microsoft SQL სერვერი: 172.17.7.132

9.2 მომხმარებელთა მართვის მოდული

მომხმარებელთა მართვის მოდული რეალიზებულია ვებ-ტექნოლოგიების გამოყენებით. დეტალური ინფორმაცია იხილეთ [ქვეთავში 5](#).



სურათი 12. მომხმარებლების მართვის მოდულის დონეები

9.2.1 ზოგადი არქიტექტურა

მომხმარებელთა მართვის მოდულში ზოგადი არქიტექტურა წარმოდგენილია შემდეგი დონეების (შრეების) ერთობლიობით:

- პრეგნენტაციის დონე
- ბიზნეს ლოგიკის დონე
- სერვისების დონე
- მონაცემების დონე

თითოეულ დონეს აქვს შესაბამისი დოკი და ფუნქცია მომხმარებელთა მართვის მოდულის ერთიან არქიტექტურაში. ამავე დროს, ეს დონეები მჭიდროდ არის ერთმანეთთან დაკავშირებული და აქტიურად ცვლის საჭირო ინფორმაციულ ნაკადებს (იხ. **სურათი 12**).

9.2.1.1 პრეგნენტაციის დონე

პრეგნენტაციის დონე წარმოადგენს იმ ინტერფეისებისა და სამუშაო ფორმების ერთობლიობას, რომელთა საშუალებითაც მოდულზე მუშაობენ ადმინისტრატორის უფლების მქონე მომხმარებლები. მოცემული დონე არის ერთგვარი ფასადი, რომლის გავლითაც ხდება ბიზნეს დონეზე რეალიზებული სცენარების შესრულება და სისტემით ოპერირება.

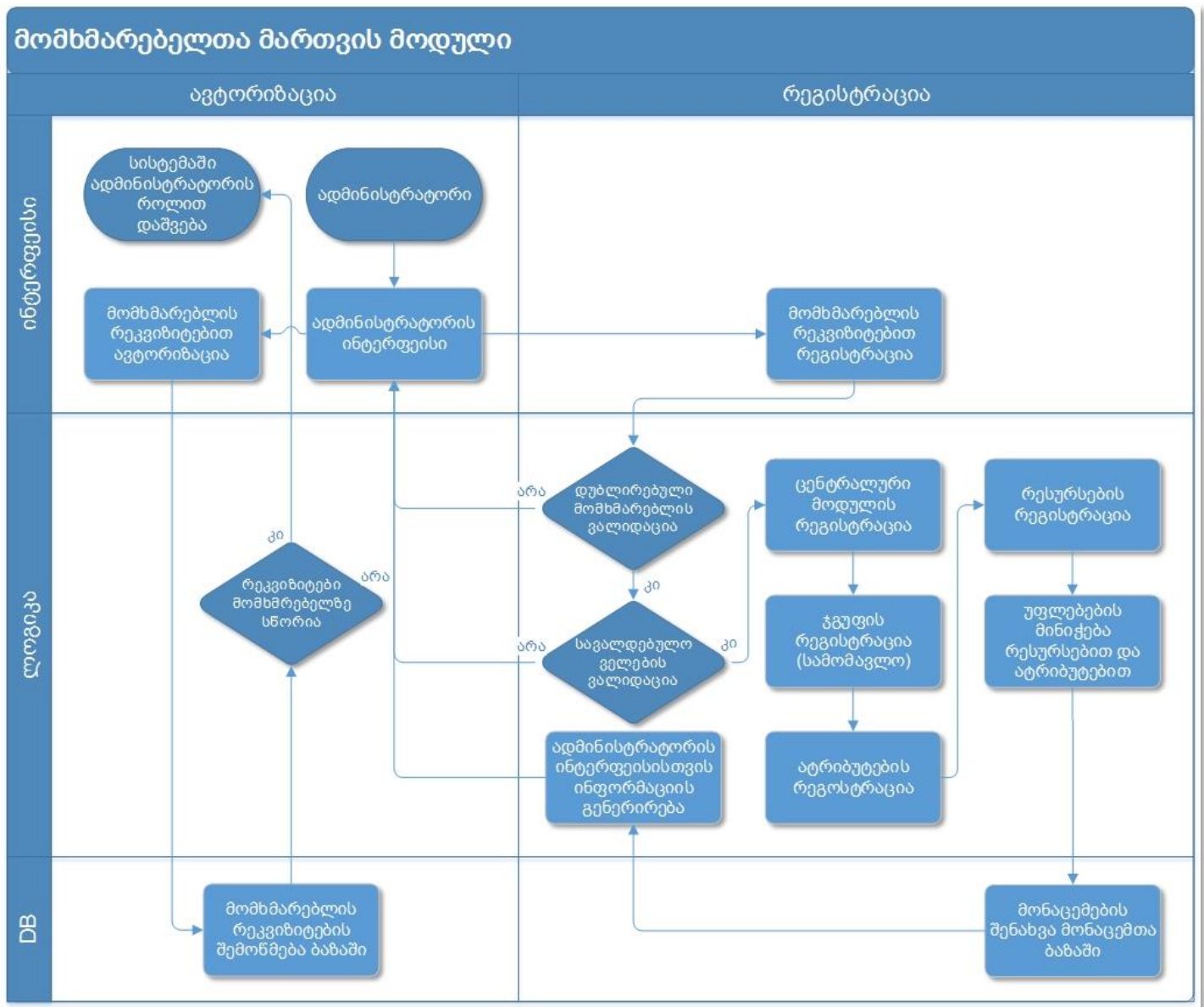
9.2.1.2 ბიზნეს-ლოგიკის დონე

ბიზნეს ლოგიკის დონე წარმოადგენს მნიშვნელოვან კომპონენტს მოდულის ზოგად არქიტექტურაში. მასში გაწერილია ყველა ის სცენარი, ბიზნეს პროცესი, ვალიდაცია და კონტროლის მექანიზმი, რომელთა ერთობლიობაც განსაზღვრავს მოდულის სპეციფიკას და დანიშნულებას. აღნიშნული დონე კავშირშია ზოგადი არქიტექტურის შესაბამის სხვა დონეებთან. სწორედ ბიზნეს ლოგიკის დონის გავლით ხდება საპრეგნენტაციო და სერვის დონეების დაკავშირება მონაცემთა დონესთან.

მომხმარებელთა მართვის მოდულის ფარგლებში რეალიზებულია რიგი ვალიდაციებისა, რომლებიც აუცილებელია სისტემისთვის დასმული ამოცანების გადაწყვეტისთვის, მომხმარებლების შესახებ ინფორმაციის სწორი შეგროვებისა და განაწილებისათვის. ინტერფეისის სპეციფიკიდან გამომდინარე, ვალიდაციები უზრუნველყოფს მომხმარებელთა მართვის მოდულის და მასზე დამოკიდებული ცენტრალური მოდულის, ასევე სამომავლოდ სხვა მოდულების გამართულ ფუნქციონირებას. ქვემოთ მოცემულია იმ ვალიდაციების ნუსხა შესაბამისი განმარტებებით, რომლებიც კრიტიკულად მნიშვნელოვანია მოდულის სწორი ფუნქციონირებისათვის.

- **აუცილებელი ველების ვალიდაცია** - აუცილებელია მომხმარებლისთვის მიეთითოს მინიმუმ მომხმარებლის სახელი და პაროლი.
- **ჯგუფის შექმნის წინაპირობა** - მომხმარებელი აუცილებლად უნდა იყოს გაწევრიანებული სასურველ მოდულში (ამ ეტაპზე მხოლოდ პორტალი) სასურველ ჯგუფში (*ადმინისტრატორი ან სტანდარტული მომხმარებელი*).
- **დუბლირების ვალიდაცია** - ვალიდაცია დუბლირებაზე მომხმარებლის სახელის მიხედვით.
- **სხვადასხვა დამატებითი ვალიდაციები** - ასევე გვაქვს ინტერფეისის სპეციფიკიდან გამომდინარე ვალიდაციები, რომლებიც უზრუნველყოფენ მოდულის გამართულ ფუნქციონირებას (მაგალითად: მოდულში კონკრეტული სერვისების გამოძახებისას არსებობს ისეთი ველები რომლებიც აუცილებლად უნდა მიეთითოს. ასევე ვალიდაციები ველის ფორმატიდან გამომდინარე, რათა არ მოხდეს თარიღობრივი მნიშვნელობის ნაცვლად არასწორი ფორმატის მქონე მნიშვნელობის გადაცემა რომელიმე სამომავლო სერვისისთვის).

სურათი 13-ზე მოცემულია გალიდაციებისა და მოდულის ლოგიკურ დონეებს შორის ნაკადების გაცვლის სცენარები ამოცანის მიხედვით:



სურათი 13. მომხმარებელთა მართვის მოდულის ბიზნეს-ლოგიკა

9.2.1.3 სერვისების დონე

სერვისების დონე მოდულის ერთ-ერთი მნიშვნელოვანი კომპონენტია, თუმცა ამ ეტაპზე მისი გამოყენება შეზღუდულია, რადგან სისტემის ფარგლებში წარმოდგენილია მხოლოდ ორი მოდული და მათ შორის სერვისებით მონაცემთა გაცვლის საჭიროება მცირეა. ამ დონის ძირითადი დანიშნულებაა მოდულის გარესამყაროსთან კავშირისა და ინტერაქტივის ორგანიზება (ბუნებრივია, შესაბამისი დაშვების არსებობის შემთხვევაში). ყველა, გარე თუ STOP C-ის ფარგლებში არსებული მოდულები UMM-თან ინფორმაციის გაცვლას ახორციელებენ სწორედ სერვის დონის გავლით.

9.2.1.4 მონაცემთა დონე (DB)

მონაცემთა დონის ძირითადი დანიშნულებაა მოდულის სხვა დონეების გავლით მიღებული ინფორმაციის დაგროვება და ოპტიმალური ორგანიზება, რაც მისი სწრაფი მოძიების, ინდექსირების,

გენერირებისა და გაცემის საშუალებას აძლევს მთელ მოდულს. აღნიშნული დონის წარმადობა ერთ-ერთი ყველაზე კრიტიკულია და მნიშვნელოვნად განსაზღვრავს მოდულის სწრაფმუდებებს. მისი საშუალებით არ არის რეალიზებული არც ერთი ბიზნეს პროცესი, რათა არ მოხდეს გარკვეული სცენარების დუბლირება ბიზნეს დონესთან.

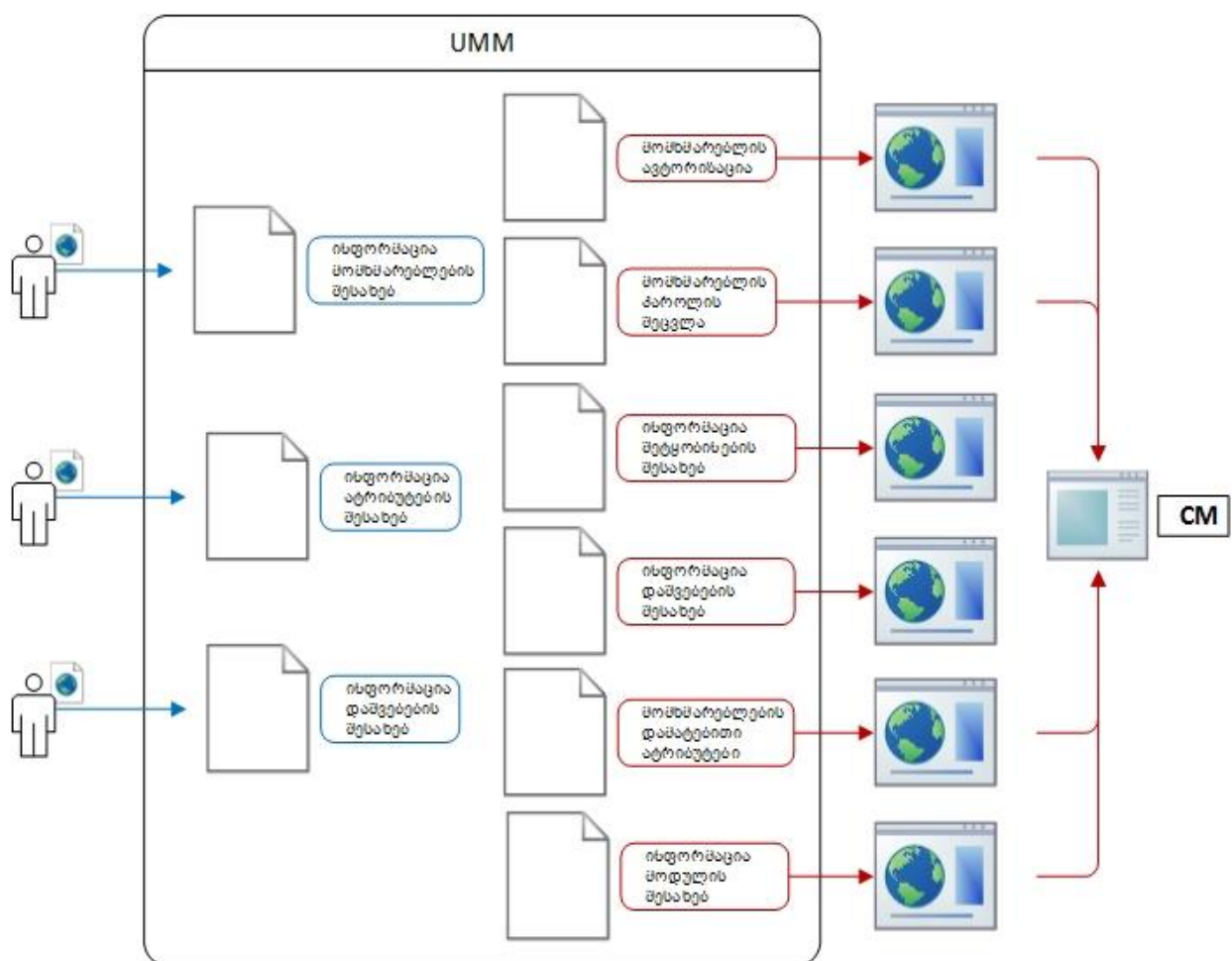
9.2.2 ფიზიკური არქიტექტურა

მომხმარებელთა მართვის მოდულის ფიზიკური არქიტექტურა წარმოდგენილია აპლიკაციის (Web) და მონაცემთა ბაზის (DB) სერვერებით. მათი ვირტუალური და ფიზიკური მისამართები მოცემულია ქვემოთ:

- Web სერვერი: <http://STOP-C.moh.gov.ge> 172.17.216.53
- DB სერვერი: 172.17.7.130

9.2.3 ინფორმაციული ნაკადები

მომხმარებელთა მართვის მოდული სხვადასხვა მეთოდებით და არხებით იღებს და გასცემს სხვადასხვა ინფორმაციას. შესაბამისად, აღნიშნული ინფორმაციული ნაკადები იყოფა ორ ძირითად ჯგუფად. კერძოდ, „შეტანილი/მიღებული ინფორმაცია“ და „გაცემული ინფორმაცია“. ასევე, განსხვავებულია ამ ინფორმაციული ნაკადების მიღების ან გაცემის, ასევე მონაცემთა ბაზაში მათი განთავსების მეთოდები. ქვემოთ (სურათი 14) იხილეთ ინფორმაციული ნაკადების გრაფიკული



სურათი 14. UMM: ინფორმაციული ნაკადები

სქემა, სადაც მწვანე ფერში მოცემულია შეტანილი/მიღებული და წითელ ფერში - გაცემული ინფორმაციული ნაკადები.

ცხრილი 5 და ცხრილი 6 წარმოადგენს „შეტანილი/მიღებული ინფორმაციის“ და „გაცემული ინფორმაციის“ დეტალებს.

ცხრილი 5. შეტანილი/მიღებული ინფორმაცია

ინფორმაციული ნაკადი	წყარო	მიღების ფორმა	პერიოდულობა	ინახება მონაცემთა ბაზაში	მისამართი	მეთოდი
ინფორმაცია მომხმარებლის შესახებ	მოდულის მომხმარებელი	ინტერფეისი	მოთხოვნის შესაბამისად	30	19	N/A
ინფორმაცია შეცვლილი პაროლის შესახებ	მომხმარებელთა მართვის მოდული (UMM)	Web-სერვისი	მოთხოვნის შესაბამისად ან პაროლის ვადის გასვლის შემთხვევაში			20
ინფორმაცია ატრიბუტების სქემების, ატრიბუტების სქემის კვანძების და ატრიბუტების მნიშვნელობების შესახებ ინფორმაცია	მომხმარებელთა მართვის მოდული (UMM)	ინტერფეისი	მოთხოვნის შესაბამისად			N/A
ინფორმაცია რესურსების და ამ რესურსებზე უფლებების განსაზღვრის შესახებ	მომხმარებელთა მართვის მოდული (UMM)	ინტერფეისი	მოთხოვნის შესაბამისად			N/A

ცხრილი 6. გაცემული ინფორმაციის ნაკადები

ინფორმაციული ნაკადი	მიმღები	მიღების ფორმა	პერიოდულობა	მისამართი	მეთოდი
ინფორმაცია ყველა მომხმარებლის შესახებ ავტორიზირებული მომხმარებლის მიხედვით	Portal	Web-სერვისი	მოთხოვნის შესაბამისად	21	22
ინფორმაცია მიმდინარე მომხმარებლის შესახებ					23
ინფორმაცია ავტორიზირებული მომხმარებლის ტოკენის აქტივობა/პასივობის შესახებ					24

¹⁹ <http://birthregistry.moh.gov.ge/HMIS/HMIS.StopC.web>

²⁰ PasswordChangeResultEnum ChangePassword(Guid token, String newPassword, String oldPassword)

²¹ <http://birthregistry.moh.gov.ge/HMIS/HMIS.UserManagement.web/Services/UserManagementWcf.svc>

²² List<UserContract> GetAllUsers(Guid token)

²³ public UserContract GetCurrentUser(Guid token)

²⁴ bool IsTokenActual(Guid token)

ავტორიზაცია					25
ინფორმაცია ჯგუფში გაწევრიანებული მომხმარებლების შესახებ					26
ინფორმაცია მომხმარებლის ჯგუფების შესახებ					27
ინფორმაცია ჯგუფის ატრიბუტების (დამატებითი ინფორმაცია) შესახებ					28
ინფორმაცია მომხმარებლის ატრიბუტების (დამატებითი ინფორმაცია) შესახებ					29

9.2.4 უსაფრთხოება

მომხმარებლების მართვის მოდულზე არასანქცირებული წვდომა და ინფორმაციის დაცულობა პროგრამულად კონტროლდება ამავე მოდულის საშუალებით. აღნიშნული ასევე უზრუნველყოფს შესაბამისი რეკვიზიტების გამოყენებით (მომხმარებელი და პაროლი) ერთჯერადი ავტორიზაციის შესაძლებლობას, რაც გულისხმობს ერთი რეკვიზიტის საშუალებით ცენტრალურ და, სამომავლოდ სხვა მოდულზე წვდომის მიღების შესაძლებლობას, UMM-ში გაწერილი დაშვებებითა და ვადით, თითოეული მოდულისთვის ინდივიდუალურად. მომხმარებლების მართვის მოდულის კომპეტენციაში შედის ასევე მისი რესურსებით სარგებლობაზე მომხმარებლებისა და შესაბამისი დაშვებების განსაზღვრა და კონტროლი.

9.3 საერთო მონაცემების მოდული

მოდულის არქიტექტურა ისეთივეა, როგორიც მომხმარებელთა მართვის მოდულის შემთხვევაში და წარმოდგენილია შემდეგი დონეების (შრეების) ერთობლიობით:

- პრეზენტაციის დონე
- ბიზნეს ლოგიკის დონე
- სერვისების დონე
- მონაცემების დონე

დამატებითი დეტალებისთვის იხ. **თავი 9.2.1.**

9.3.1 საერთო მონაცემების მოდულის კლასები

- **Caches:** დროებითი მონაცემების შემნახველი კლასები
 - TranslationCache.cs
- **Configs:** აპლიკაციის კონფიგურაციის კლასები
 - **InsuranseService:** სერვისის კლასი
 - ServiceUrlElement.cs
 - ServiceUrlElementCollection.cs
 - ServiceUrls.Section.cs

²⁵ Guid? Login(String loginName, String password, bool encryptedPassword)

²⁶ List<UserContract> GetGroupUsers(Guid token, Guid groupId)

²⁷ List<GroupContract> GetUserGroups(Guid token, Guid userID, Guid projectId)

²⁸ List<GroupAttributeContract> GetGroupAttributes(Guid token, Guid groupId)

²⁹ List<UserAttributeContract> GetUserAttributes(Guid token, Guid userID, Guid projectId)

- **Logger:** ლოგირების კონფიგურაციის კლასები
 - LoggerCompitabilitiesSection.cs
 - LoggerCompitabilityElement.cs
 - LoggerCompitabilityElementCollection.cs
- **Entities:** აპლიკაციის ობიექტებისთვის საჭირო კლასები
 - InsuranceEntity.cs
 - PersonInfoSsaEntity.cs
 - TranslationEntity.cs
- **Extensions:** მონაცემთა ბაზის მონაცემების კონვერტაციის კლასი აპლიკაციის შიდა კლასებზე და პირიქით + მოვიერთი კლასებისთვის დამატებითი custom მეთოდების მიბმა
 - ContractsExtensions.cs
- **Handlers:** გვერდებისთვის მოგადი მონაცემების (მაგ.: ფაილი, სურათი) დამუშავების კლასი
 - PersonPhotoHandler.cs
- **Helpers:** სხვადასხვა დამხმარე კლასები
 - **GovTalk:** სახელმწიფო სერვისების განვითარების სააგენტოს სამოქალაქო რეესტრის ვებ-სერვისებთან (ყოფილი CRA.gov.ge) კავშირისთვის საჭირო კლასები
 - EncryptKey.cs
 - GovTalkApiClient.cs
 - GovTalkCallApi.cs
 - GovTalkHelpers.cs
 - KeyExchange.cs
 - Result.Status.cs
 - XmlCrypting.cs
 - XmlSigning.cs
 - Cryptography.cs
 - DynamicEntity.cs
 - LogHelper.cs
 - SyncPersonDeathInfoloader.cs
 - SyncPersonInfoloader.cs
 - TrustAllCertificatePolicy.cs
- **Services:** ვებ-სერვისებისთვის საჭირო კლასები
 - **Managers:** ვებ-სერვისების სამართავი კლასები
 - CommonDataChangeTimeManager.cs
 - CommonDataManager.cs
 - Common.ServiceWrapper.cs
 - MohServicesManager.cs
 - CommonDataWcf.cs
 - CommonDataWeb.cs
 - CommonDataWcf.cs
- **Util:** დამატებითი უტილიტების (ინსტრუმენტების) კლასები
 - EmailUtil.cs
 - MohServicesUtil.cs
 - RemoteAccessUtil.cs
 - UserManagementUtil.cs

9.3.2 ინფორმაციული ნაკადები

Common Data მოდული არის საერთო ინფორმაციისა თუ სერვისების საცავი STOP C-ის ფარგლებში არსებული მოდულებისთვის. ამ მოდულის ფარგლებში არსებული ინფორმაციული ნაკადები. ამ ნაკადების გაცვლის ფორმები და მათი შესაბამისი წყაროები წარმოდგენილია შემდეგ ორ კატეგორიაში:

- STOP C-ის ფარგლებში არსებული მოდულების ინტერფეისის კომპონენტების დასახელებების სხვადასხვა ენაზე;
- მომხმარებლების მიერ STOP C-ის ფარგლებში არსებული მოდულებში განხორციელებული ოპერაციები (Log-ები);

მოდული ასევე სხვადასხვა მეთოდებით იღებს ან გაცემს ინფორმაციას. შესაბამისად, აღნიშნული ნაკადები იყოფა ორ ძირითად ჯგუფად: მიღებული და გაცემული ნაკადები. ასევე განსხვავებულია ამ ინფორმაციული ნაკადების მიღების/გაცემის, ასევე მონაცემთა ბაზაში მათი განთავსების მეთოდები. ქვემოთ მოცემულია ინფორმაციული ნაკადების ნუსხა მათი მიმართულების, მიღების/ გაცემის მეთოდისა და პერიოდულობის მიხედვით:

ცხრილი 7. შეტანილი/მიღებული ინფორმაცია

ინფორმაციული ნაკადი	წყარო	მიღების ფორმა	პერიოდულობა	ინახება მონაცემთა ბაზაში	მისამართი
STOP C-ის ფარგლებში არსებული მოდულების ინტერფეისების კომპონენტების დასახელებები საჭირო ენაზე	ინტერფეისი	ინტერფეისი	ერთჯერადი თითოეული სიტყვისთვის	კი	30
SetTranslatedText: ამატებს კონკრეტული კონტროლის სასურველ ენაზე თარგმანს	STOP C მოდულები	Web-სერვისი	ერთჯერადი თითოეული კონტროლისთვის		

ცხრილი 8. გაცემული ინფორმაციის ნაკადები

ინფორმაციული ნაკადი	წყარო	მიღების ფორმა	პერიოდულობა	მისამართი
GetLanguages: აბრუნებს ენების ჩამონათვალს, რომლებზეც ნათარგმნია STOP C მოდულები	STOP C მოდულები	Web-სერვისი	მოთხოვნის შესაბამისად	31
GetTranslatedText: აბრუნებს ინტერფეისის კონტროლის თარგმანს სასურველ ენაზე	STOP C მოდულები	Web-სერვისი	მოთხოვნის შესაბამისად	

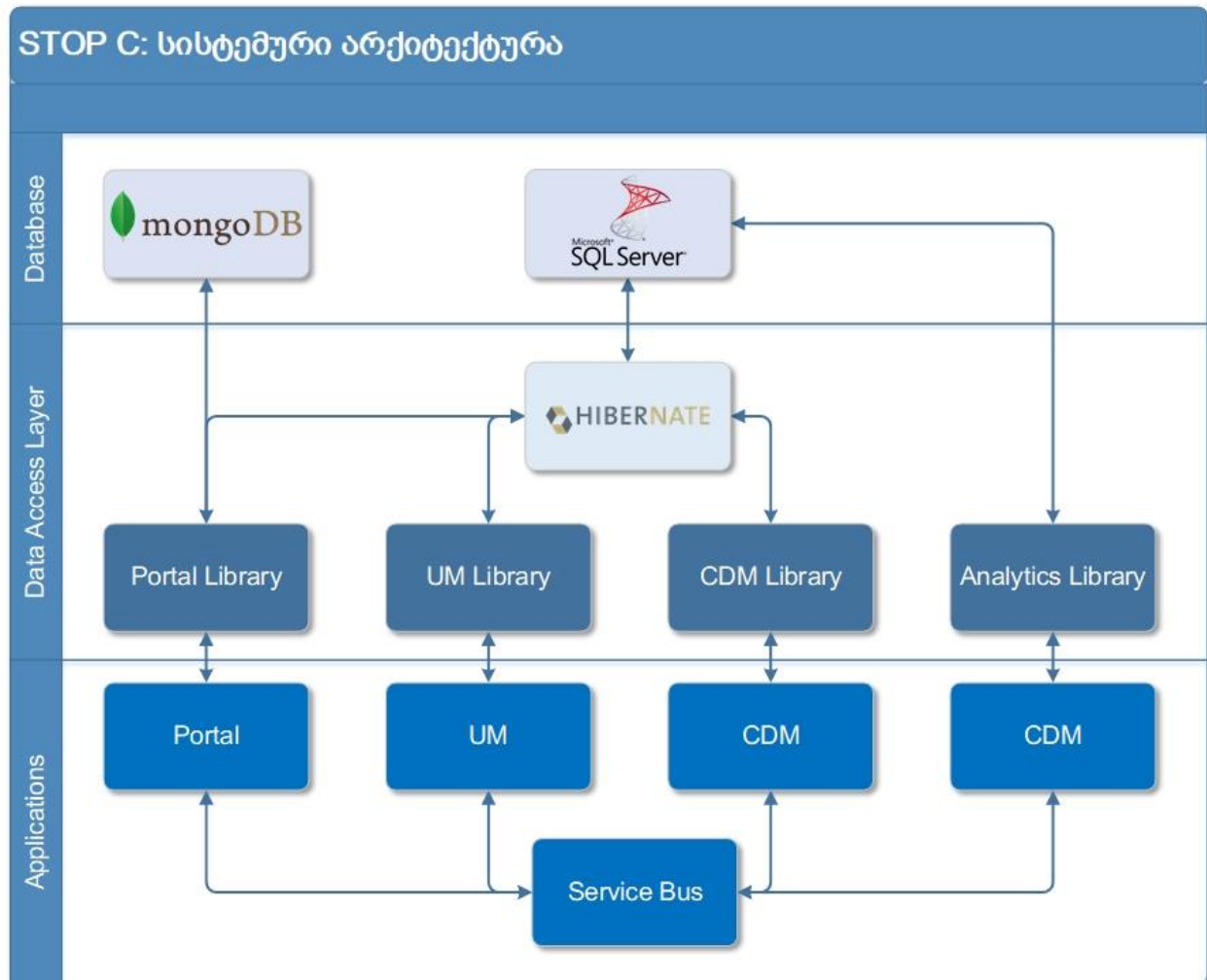
³⁰ <http://birthregistry.moh.gov.ge/HMIS/HMIS.CommonData/Services/CommonDataWcf.svc>

³¹ <http://birthregistry.moh.gov.ge/HMIS/HMIS.CommonData/Services/CommonDataWcf.svc>

10 მონაცემთა ბაზის სტრუქტურა და აღწერილობა

მონაცემების შენახვის, დამუშავებისა და გადაცემის ამოცანები STOP C ბაზაში გადანაწილებულია მონაცემთა ბაზის ორ სხვადასხვა სისტემაში. ეს გადანაწილება დინამიურს ხდის სისტემას და მონაცემების შენახვასთან და დამუშავებასთან დაკავშირებული ამოცანების გადასაჭრელად იყენებს ორი სისტემიდან იმას, რომელიც უფრო მეტად მორგებული და ეფექტურია იმ კონკრეტული ამოცანისთვის.

- Microsoft SQL³² : იხ. [თავი 8.2.1](#) და [8.2.2](#)
- MongoDB³³ : იხ. თავი 8.3



სურათი 15. STOP C კომპონენტების არქიტექტურა მონაცემთა ბაზების ჭრილობაში

10.1 მომხმარებელთა მართვის მოდულის მონაცემთა ბაზა

მომხმარებელთა მართვის მოდულის მონაცემთა ბაზის არქიტექტურა წარმოდგენილია ერთი მოდულისთვის ცენტრალური მონაცემთა ბაზით, რომელიც განთავსებულია მონაცემთა ბაზის სერვერზე (მისამართი და ტექნიკური დეტალები იხ. [ქვეთავი 10.2](#)). ლოგირების მექანიზმი გარკვეულწილად რეალიზებულია ისეთი მონაცემებისთვის, რომელთა ისტორიული ანალიზიც არის

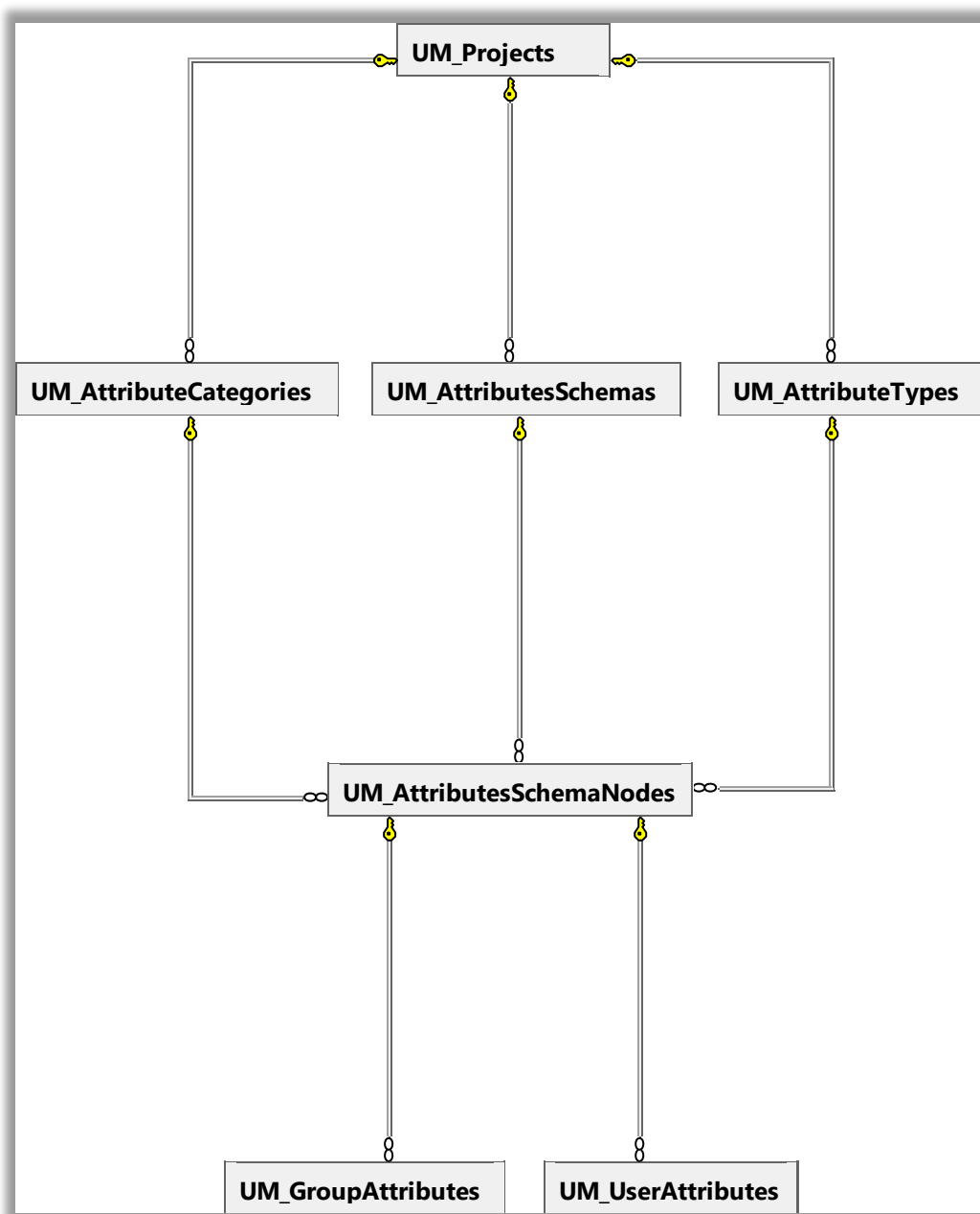
³² <https://www.microsoft.com/en-us/sql-server/sql-server-2017>

³³ <https://www.mongodb.com>

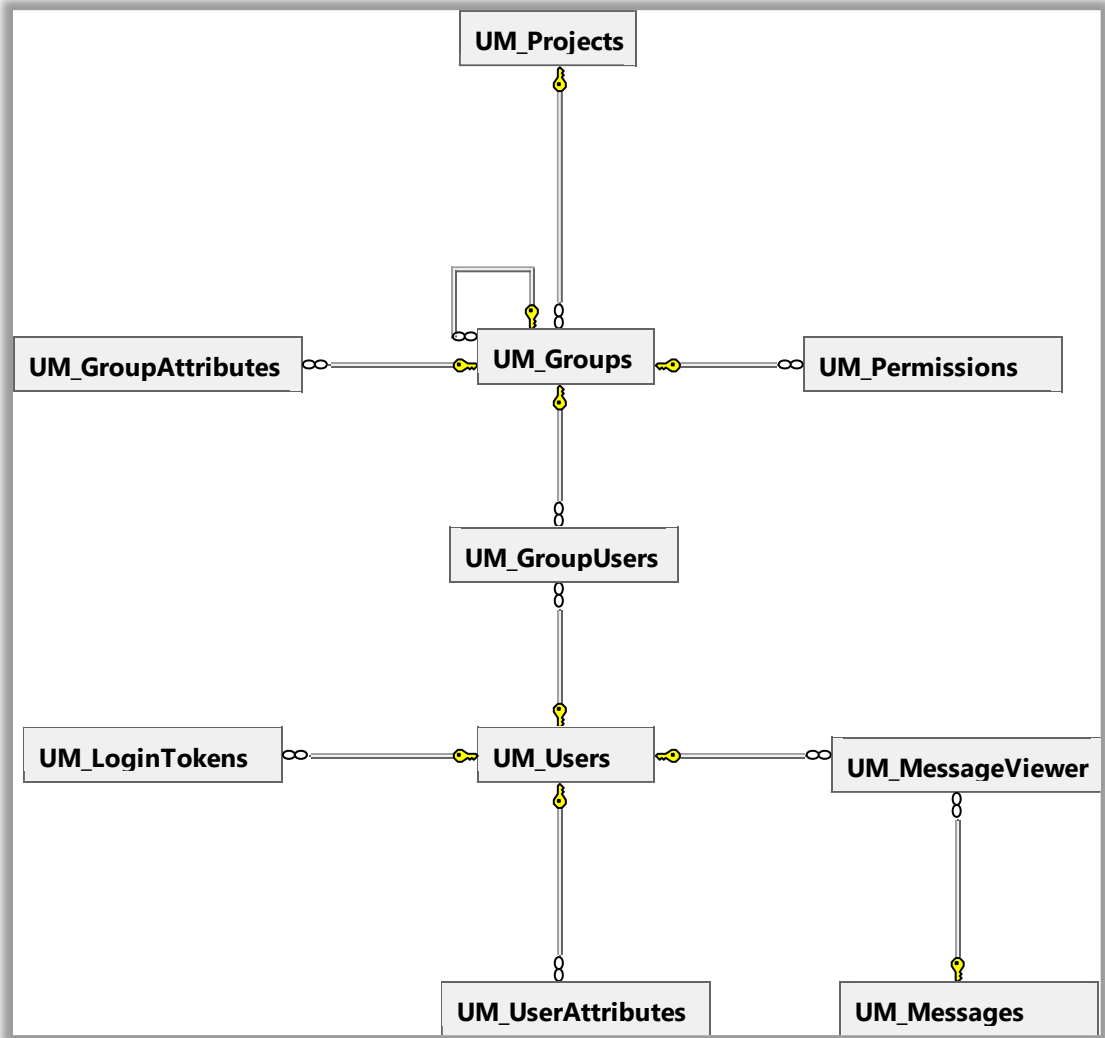
ან შესაძლოა გახდეს მნიშვნელოვანი (ამ მიზნით მნიშვნელოვან ცხრილებში არსებობს შემდეგი ველების ერთობლიობა: DateCreated, DateChanged, DateDeleted).

არსებობს ასევე ლოგირების სისტემა მოდულის წარმადობის მონიტორინგისთვის. მისი ინტერფეისი საშუალებას იძლევა, რეალურ დროში მოხდეს მომხმარებლის მიერ კონკრეტულ ინტერფეისულ ოპერაციაზე დახარჯული დროის აღრიცხვა და მონიტორინგი.

10.1.1 მონაცემთა ბაზის სტრუქტურა



სურათი 16. ატრიბუტების სქემა



სურათი 17. მომხმარებლები, უფლებები და ატრიბუტების მნიშვნელობები

10.1.2 მონაცემთა ბაზის ცხრილები და ველები

ცხრილი 9. მომხმარებლების ადმინისტრირების მოდულის მონაცემთა ბაზის ცხრილების აღწერილობა და ველები

ცხრილის დასახელება	აღწერა	ველები
UM_AttributesSchemaNodes	ატრიბუტების კატეგორიებისა და ატრიბუტების ტიპების მიხედვით იქმნება ატრიბუტების სქემის კვანძები მაგალითად, „დაწესებულების იდენტიფიკატორი“, რომელიც თავის მხრივ ერთიანდება ატრიბუტების სქემაში	ID
		Name
		AttributeCategoryID
		AttributeTypeID
		AttributesSchemaID
		DateCreated
		DateChanged
		DateDeleted
UM_AttributesSchemas	მოდულის დონეზე განსაზღვრული ატრიბუტების სქემა, რომელიც ერთგვარად ასრულებს ატრიბუტების	Hashcode
		ID
		ProjectID
		Name

ცხრილის დასახელება	აღწერა	ველები
	სქემების კვანძების მაჯგუფებლის როლს	DateCreated
		DateChanged
		DateDeleted
		HashCode
UM_GroupAttributes	ჯგუფებისთვის განკუთვნილი დამატებითი ინფორმაციის მნიშვნელობა, რომელიც იქმნება ატრიბუტების სქემების, ატრიბუტების კატეგორიების, ატრიბუტების ტიპების და ატრიბუტების სქემების კვანძების მიხედვით	ID
		GroupID
		AttributesSchemaNodeID
		Value
		DateCreated
		DateChanged
		DateDeleted
		HashCode
UM_Groups	მოდულის დონეზე განსაზღვრული ჯგუფები	ID
		ParentID
		ProjectID
		Name
		DateCreated
		DateChanged
		DateDeleted
		HashCode
UM_GroupUsers	მომხმარებლებისა და ჯგუფების კავშირის ცხრილი	ID
		UserID
		GroupID
		AccessLevel
		DateCreated
		DateChanged
		DateDeleted
		HashCode
UM_LoginTokens	მომხმარებლების სხვადასხვა მოდულში ავტორიზაციის შედეგად შექმნილი ტოკენების ცხრილი	ID
		LoginToken
		UserID
		ExpireDate
		LastAccessDate
		DateCreated
		DateChanged
		DateDeleted
		HashCode
		DeleteReason
UM_Messages		ID
		Subject

ცხრილის დასახელება	აღწერა	ველები
	მოდულის, ჯგუფის და მომხმარებლის დონეზე არსებული შეტყობინებების ცხრილი	Text
		ObjectID
		Type
		DateCreated
		DateChanged
		DateDeleted
UM_MessageViewer	წაკითხული/წაუკითხავი შეტყობინებების ცხრილი	ID
		MessageID
		UserID
		DateCreated
		DateChanged
		DateDeleted
UM_Permissions	მოდულში არსებული ჯგუფის და რესურსების მიხედვით View, Add, Edit, Delete უფლებების ცხრილი	ID
		GroupID
		ResourceID
		RuleValue
		DateCreated
		DateChanged
		DateDeleted
		Hashcode
UM_Projects	მოდულების ცხრილი	ID
		Name
		DateCreated
		DateChanged
		DateDeleted
		IsActive
		Hashcode
UM_Resources	რესურსების ცხრილი, სადაც შეიძლება შევინახოთ ისევე როგორც ვირტუალური, ასევე ამა თუ იმ მოდულში არსებული გვერდების და ამ გვერდებზე არსებული კომპონენტების მისამართები. მაგალითად, module/pages/default.aspx	ID
		ParentID
		ProjectID
		Name
		Description
		Type
		Value
		DateCreated
		DateChanged
		DateDeleted
		Hashcode
UM_UserAttributes	მომხმარებლებისთვის განკუთვნილი დამატებითი ინფორმაციის	ID
		UserID

ცხრილის დასახელება	აღწერა	ველები
	მნიშვნელობა, რომელიც იქმნება ატრიბუტების სქემების, ატრიბუტების კატეგორიების, ატრიბუტების ტიპების და ატრიბუტების საქმების კვანძების მიხედვით	AttributesSchemaNodeID
		Value
UM_Users	მომხმარებლების ცხრილი	DateCreated
		DateChanged
		DateDeleted
		HashCode
		ID
		LoginName
		Password
		FirstName
		LastName
		Email
		Address
		OrganizationName
		Department
		Division
		Position
		Telephone
		IsSuperAdmin
		IsActive
		UserCategoryID
		PasswordExpirationDate
		DateCreated
		DateChanged
		DateDeleted
		HashCode

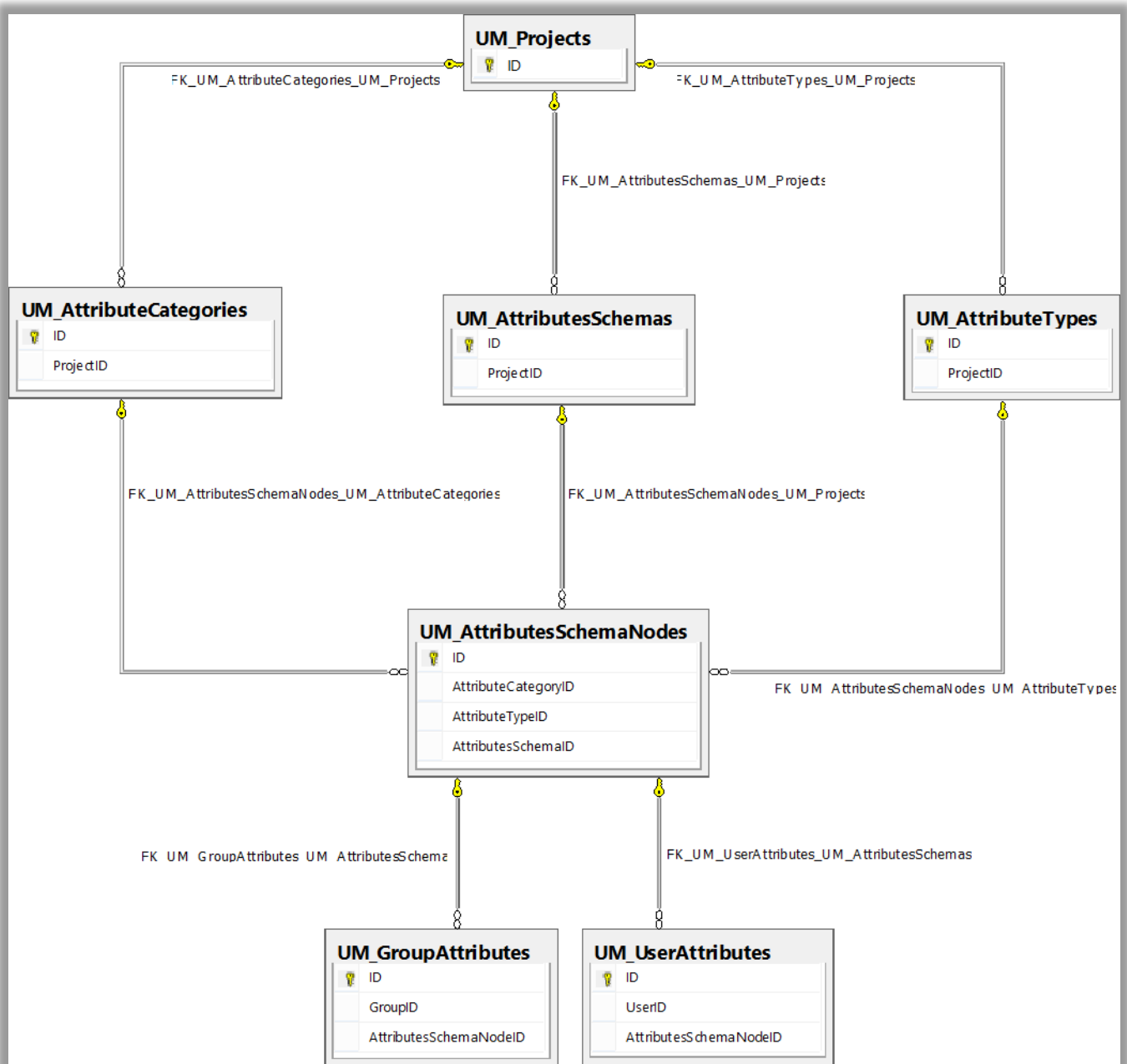
10.1.3 ცხრილებს შორის კავშირები

მონაცემთა ბაზაში არსებული ცხრილები დაკავშირებულია ერთმანეთთან შესაბამისი ველების საშუალებით და კანონზომიერებით. სურათი 6 და სურათი 7-ზე მოცემულია კავშირების დიაგრამები, ხოლო ცხრილი 3 შეიცავს დეტალურ ინფორმაციას ცხრილებს შორის არსებული კავშირებისა და მათი ტიპების შესახებ. ასევე მოცემულია განმარტებები, რომლებიც აღწერს აღნიშნული კავშირების დანიშნულებას.

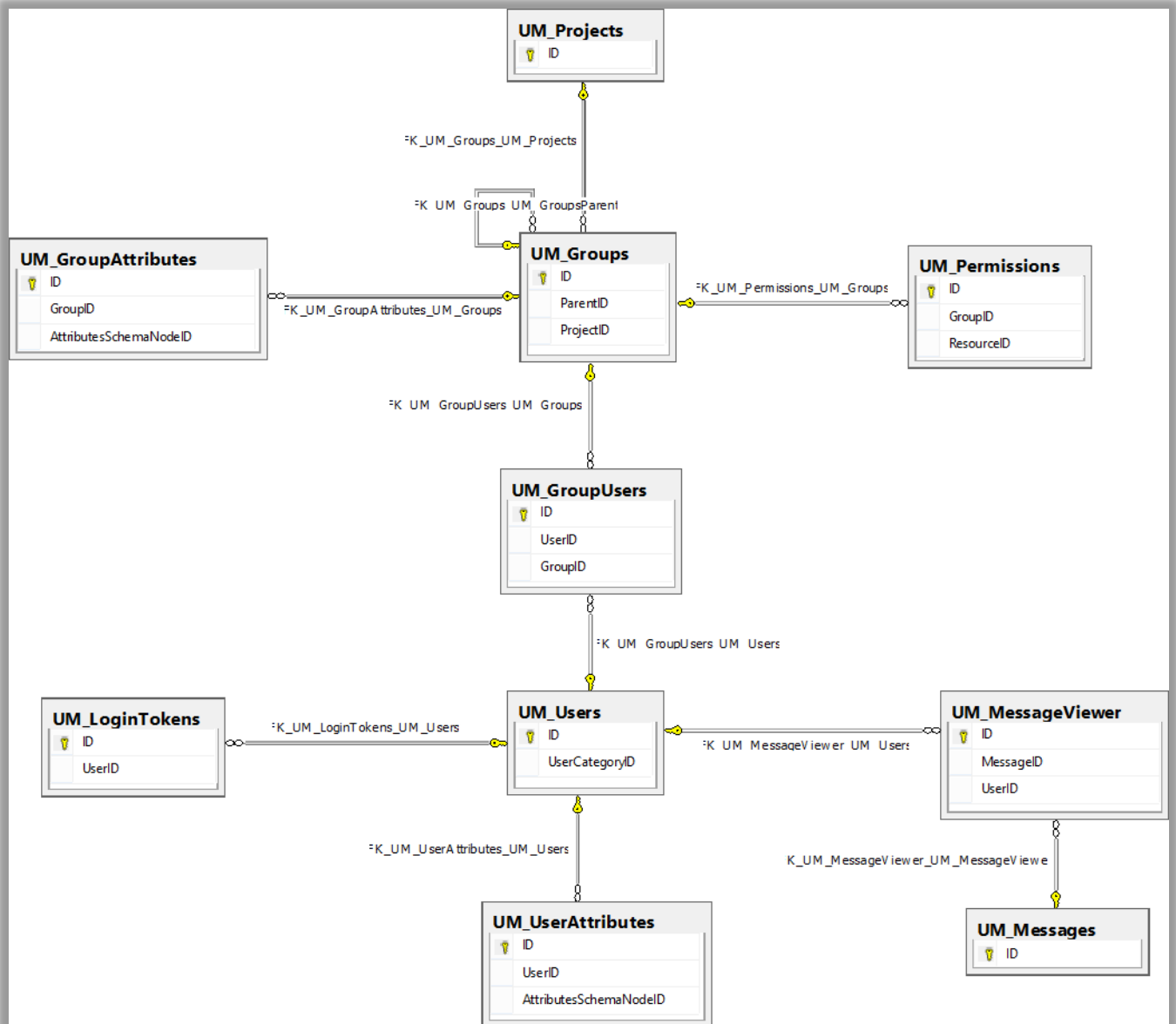
ცხრილი 10. მონაცემთა ბაზის ცხრილებს შორის კავშირები და ტიპები

კავშირის დასახელება	ცხრილი (1)	ცხრილი (2)	კავშირის ტიპი	აღწერა
FK_UM_Groups_UM_GroupsParent	UM_Groups	UM_Groups	One To Many	მშობელი შვილის კავშირი ჯგუფებს შორის

FK_UM_Groups_UM_P rojects	UM_Groups	UM_Projects	One To Many	ჯგუფების ბმა მოდულებთან
FK_UM_GroupUsers_UM_Groups	UM_GroupUsers	UM_Groups	Many To Many	ჯგუფებში განწევრიანებული მომხმარებლების ბმა ჯგუფებთან
FK_UM_GroupUsers_UM_Users	UM_GroupUsers	UM_Users	Many To Many	ჯგუფებში განწევრიანებული მომხმარებლების ბმა მომხმარებლებთან
FK_UM_LoginTokens_UM_Users	UM_LoginTokens	UM_Users	One To Many	ავტორიზირებული მომხმარებლების ტოკენების ბმა მომხმარებლებთან
FK_UM_MessageViewer_UM_MessageViewer	UM_MessageViewer	UM_Messages	One To Many	ნაუკითხავი/ ნაკითხული შეტყობინებების ბმა შეტყობინებებთან გამოიყენება HMIS-ის ფარგლებში არსებულ მოდულებში შეტყობინების კონტროლისათვის
FK_UM_MessageViewer_UM_Users	UM_MessageViewer	UM_Users	One To Many	ნაკითხული/ ნაუკითხავი შეტყობინებების ბმა მომხმარებლებთან HMIS-ის ფარგლებში არსებულ მოდულებში შეტყობინების კონტროლისათვის
FK_UM_Permissions_UM_Groups	UM_Permissions	UM_Groups	One To Many	უფლებების ბმა ჯგუფებთან გამოიყენება HMIS-ის ფარგლებში არსებულ მოდულებში როლების მენეჯმენტისთვის
FK_UM_Permissions_UM_Resources	UM_Permissions	UM_Resources	One To Many	უფლებების ბმა რესურსებთან გამოიყენება HMIS-ის ფარგლებში არსებულ მოდულებში როლების მენეჯმენტისთვის
FK_UM_Resources_UM_Projects	UM_Resources	UM_Projects	One To Many	რესურსების ბმა მოდულებთან გამოიყენება HMIS-ის ფარგლებში არსებულ მოდულებში როლების მენეჯმენტისთვის
FK_UM_UserAttributes_UM_AttributesSchemas	UM_UserAttributes	UM_AttributesSchemasNodes	One To Many	მომხმარებლების ატრიბუტების ბმა ატრიბუტების სქემის კვანძებთან, დამატებითი ინფორმაცია მომხმარებელზე
FK_UM_UserAttributes_UM_Users	UM_UserAttributes	UM_Users	One To Many	მომხმარებლების ატრიბუტების ბმა მომხმარებლებთან დამატებითი ინფორმაცია მომხმარებელზე



სურათი 18. ატრიბუტების სქემები

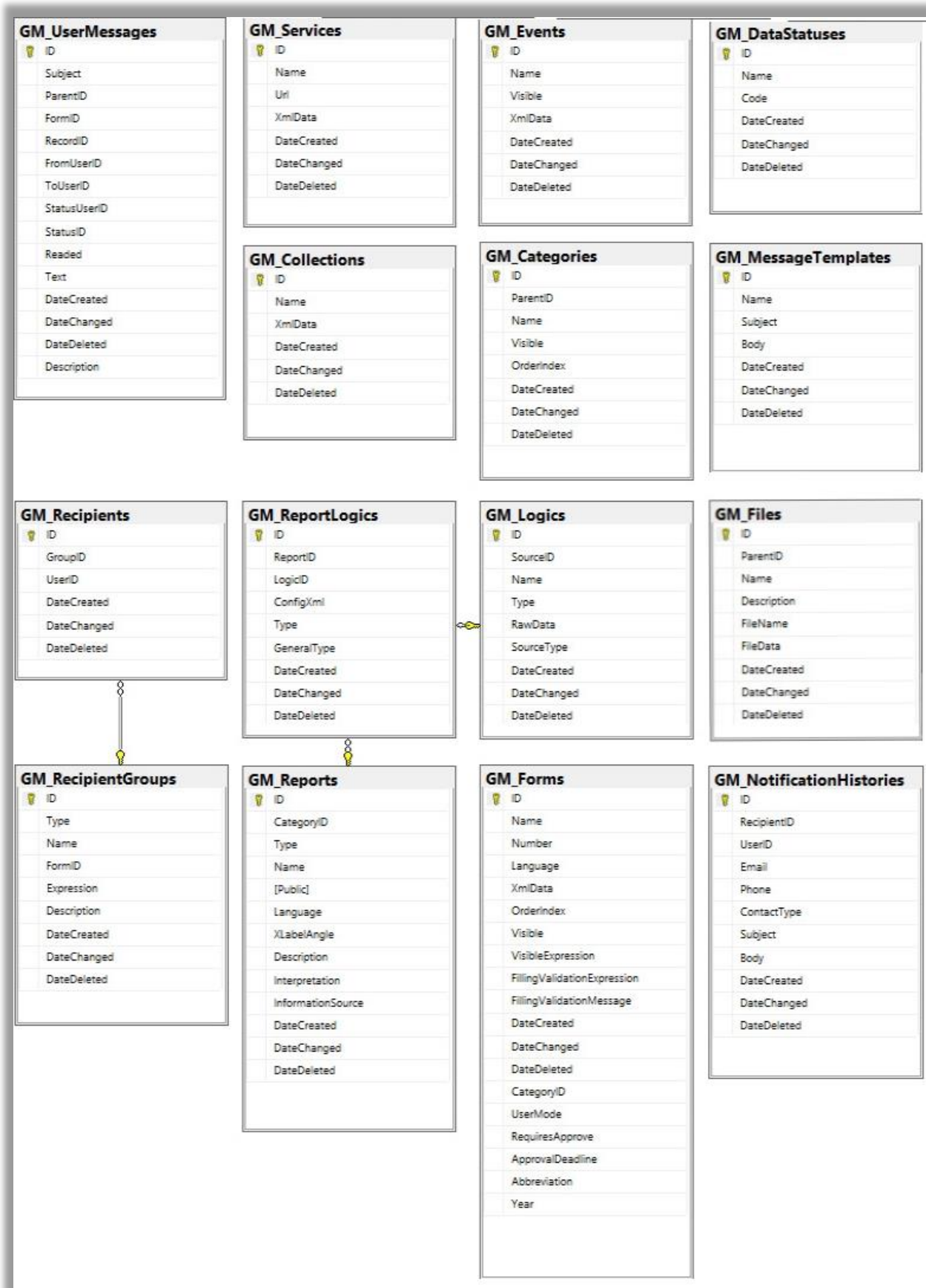


სურათი 19. მომხმარებლები და ატრიბუტების მნიშვნელობები

10.2 ცენტრალური მოდულის (პორტალის) ბაზა

პორტალის მონაცემთა ბაზის არქიტექტურა წარმოდგენილია 2 ტიპის მონაცემთა ბაზით - რელაციური ბაზითა (PostgreSQL) და დოკუმენტ-ორიენტირებული ბაზით (MongoDB). ბაზების მისამართისა და ტექნიკური დეტალებისთვის იხ. [ფაგი 10.2](#).

10.2.1 Microsoft SQL: მონაცემთა ბაზის სტრუქტურა



სურათი 20. პორტალის მონაცემთა ბაზის ცხრილების სტრუქტურა (ნაწილი 1)

10.2.2 Microsoft SQL: მონაცემთა ბაზის ცხრილები და ველები

ცხრილი 11. პორტალის მონაცემთა ბაზის ცხრილების აღწერილობა და ველები

ცხრილი	სვეტი	აღწერა
GM_Services	გარე სერვისების აღწერა	
	ID	იდენტიფიკატორი
	Name	დასახელება
	Url	სერვისის მისამართი
	XmlData	XML-ის სახით შენახული ცნობარის ველების ჩამონათვალი
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_UserMessages	მომხმარებლების შეტყობინებები (ექსპერტის მეცნიერთან მიმონერა)	
	ID	იდენტიფიკატორი
	Subject	შეტყობინების თემა
	ParentID	
	FormID	
	RecordID	
	FromUserID	
	ToUserID	
	StatusUserID	
	StatusID	
	Readed	
	Text	
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
	Description	აღწერა
GM_Recipients	შეტყობინებების სისტემაში შეტყობინების მიმღები მომხმარებლების ცხრილი	
	ID	იდენტიფიკატორი
	GroupID	
	UserID	
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_Files	ფორმისთვის საჭირო ატვირთული ფაილების ცხრილი	
	ID	იდენტიფიკატორი
	ParentID	
	Name	დასახელება
	Description	აღწერა
	FileName	
	FileData	

	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_NotificationHistories	გაგზავნილი შეტყობინებების ცხრილი	
	ID	იდენტიფიკატორი
	RecipientID	
	UserID	
	Email	ელექტრონული ფოსტა
	Phone	ტელეფონი
	ContactType	
	Subject	
	Body	
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_RecipientGroups	შეტყობინების მიმღები მომხმარებელთა ჯგუფების ცხრილი	
	ID	იდენტიფიკატორი
	Type	ჯგუფის ტიპი
	Name	დასახელება
	FormID	
	Expression	
	Description	აღწერა
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_MessageTemplates	შეტყობინების შაბლონების ცხრილი	
	ID	იდენტიფიკატორი
	Name	დასახელება
	Subject	
	Body	
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_Events	კალენდრის ივენტების (Events) ცხრილი	
	ID	იდენტიფიკატორი
	Name	დასახელება
	Visible	
	XmlData	XML-ის სახით შენახული ცნობარის ველების ჩამონათვალი
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_Categories	ცხრილი შეიცავს რეპორტების და ფორმების კატეგორიებს	
	ID	იდენტიფიკატორი
	ParentID	მშობელი კატეგორიის იდენტიფიკატორი

	Name	კატეგორიის სახელი
	Visible	ჩანდეს თუ არა მენიუში
	OrderIndex	რიგითობის ინდექსი
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_Collections	ცხრილი შეიცავს ცნობარების სტრუქტურის აღწერას	
	ID	იდენტიფიკატორი
	Name	დასახელება
	XmlData	XML-ის სახით შენახული ცნობარის ველების ჩამონათვალი
	DateCreated	შექმნის თარიღი
	DateDeleted	წაშლის თარიღი
GM_DataStatuses	ცხრილი შეიცავს ფორმის მონაცემების სტატუსებს (გადაგზავნილი, უარყოფილი, დადასტურებული და ა.შ)	
	ID	იდენტიფიკატორი
	Name	დასახელება
	Code	კოდი
	DateCreated	შექმნის თარიღი
	DateDeleted	წაშლის თარიღი
GM_Forms	ცხრილი შეიცავს ფორმების აღწერას	
	ID	იდენტიფიკატორი
	Name	დასახელება
	Number	უნიკალური ნომერი
	XmlData	ფორმის სტრუქტურის აღწერა XML-ის სახით (ველების ჩამონათვალი ცხრილები, ტაბები, ჯგუფები და სხვა)
	OrderIndex	რიგითობის ნომერი
	Visible	ჩანდეს თუ არა მომხმარებლების მხარეს
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
	CategoryID	კატეგორიის იდენტიფიკატორი
	UserMode	მომხმარებლის მიერ შევსების რეჟიმი
GM_Logics	ცხრილი შეიცავს მონაცემების დამუშავების ლოგიკებს	
	ID	იდენტიფიკატორი
	FormID	ფორმის იდენტიფიკატორი (საიდანაც უნდა მოხდეს მონაცემების ამოღება)
	LogicID	მშობელი ლოგიკის იდენტიფიკატორი (საიდანაც უნდა მოხდეს მონაცემების ამოღება)
	Name	დასახელება

	Type	ტიპი (ტექსტური ბრძანება (Query) თუ დიზაინერით აწყობილი ლოგიკა)
	RawData	XML-ის სახით შენახული ლოგიკა ან ბრძანება
	SourceType	მონაცემთა წყაროს ტიპი (ფორმა ან ლოგიკა)
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_ReportLogics	რეპორტების ლოგიკებთან ბმის ცხრილი	
	ID	იდენტიფიკატორი
	ReportID	რეპორტის იდენტიფიკატორი
	LogicID	ლოგიკის იდენტიფიკატორი
	ConfigXml	კონფიგურაციის XML
	Type	ტიპი
	GeneralType	ზოგადი ტიპი
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი
GM_Reports	ეს ცხრილი შეიცავს რეპორტების აღწერას (დიაგრამა/ცხრილი)	
	ID	იდენტიფიკატორი
	CategoryID	კატეგორიის იდენტიფიკატორი
	Type	ტიპი (ცხრილი/დიაგრამა)
	Name	დასახელება
	Public	არის თუ არა საჯარო (თანხმობის შემთხვევაში უჩანს ყველა მომხმარებელს, წინააღმდეგ შემთხვევაში ხედავს მხოლოდ ადმინისტრატორი)
	Language	ენა
	XLabelAngle	X ღერძზე წარწერების დახრის კუთხე
	Description	მოკლე აღწერა
	Interpretation	ინტერპრეტაცია (ავსებს მომხმარებელი)
	InformationSource	ინფორმაციული წყარო (ავსებს მომხმარებელი)
	DateCreated	შექმნის თარიღი
	DateChanged	ბოლო ცვლილების თარიღი
	DateDeleted	წაშლის თარიღი

11 აუცილებელი ბიბლიოთეკები

სისტემა სრულფასოვანი ფუნქციონირებისთვის მოითხოვს დამატებითი ბიბლიოთეკების აუცილებლობას. სისტემაში გამოიყენება შემდეგი დამატებითი ბიბლიოთეკები.

11.1 NPOI

ეს ბიბლიოთეკა წარმოადგენს .NET ვერსიას POI Java პროექტის³⁴. POI არის ღია (open source) პროექტი³⁵ რომლის საშუალებითაც შესაძლებელია .xls, .doc, .ppt ფაილების წაკითხვა და ჩანწერა. STOP C სისტემაში NPOI გამოიყენება Excel ფაილების იმპორტისა და ექსპორტისთვის, სერვერის მხარეზე Microsoft Office-ის ინსტალაციის გარეშე. ეს გადაწყვეტილება უფრო ეფექტურია, ვიდრე Microsoft Excel ActiveX-ის გამოყენება ფონურ რეჟიმში.

11.2 ASP Chart Control

Chart controls^{36,37} საშუალებას იძლევა ASP.NET გვერდების და Windows Forms აპლიკაციების შექმნას მარტივი, ინტუიტიური და ვიზუალურად ადვილად აღსაქმელი გრაფიკებით, კომპლექსური სტატისტიკური და ფინანსური ანალიზისთვის. ამ ბიბლიოთეკის საშუალების შესაძლებელია 35 სხვადასხვა ტიპის გრაფიკისა და დიაგრამის შექმნა.

11.3 DevExpress

DevExpress³⁸ წარმოადგენს პროგრამირებისთვის საჭირო ინსტრუმენტებისა და ბიბლიოთეკების ნაკრებს, რომლებიც დამატებითი კომპონენტების საშუალებით (plugin) უფრო სწრაფს და ეფექტურს აპლიკაციის შექმნას. STOP C სისტემაში გამოიყენება DevExpress ASP.NET სამომხმარებლო ინტერფეისის კონტროლები.

11.4 NHibernate

NHibernate³⁹ წარმოადგენს ე.წ. ობიექტურ-რელაციურ მაკავშირებელს (Object-Relational Mapper - ORM), რაც გულისხმობს STOP C-ის მონაცემების „თარგმნას“ .NET გარემოდან თითქმის ნებისმიერი რელაციური ბაზისთვის გასაგებ ფორმატში (Oracle DB, IBM DB2, mysql, postgres SQL და ა.შ.). შედეგად, STOP C არ არის დამოკიდებული მონაცემების კონკრეტულ ბაზაზე და შესაძლებელი ნებისმიერი რელაციური ბაზის გამოყენება.

11.5 Bootstrap

Bootstrap⁴⁰ წარმოადგენს მომხმარებლის ინტერფეისის კომპონენტები შექმნისთვის საჭირო ინსტრუმენტს და არის ე.წ. front-end⁴¹ გარემო ვებ-გვერდებისა და ვებ-აპლიკაციების დიზაინისთვის. Bootstrap შეიცავს HTML და CSS-ზე დაფუძნებულ დიზაინის შაბლონებს, რომლებითაც ხდება STOP C-ის სამომხმარებლო ინტერფეისის შექმნა (ფორმები, ლილაკები, ნავიგაცია და სხვა ინტერფეისის კომპონენტები; შესაძლებელია JavaScript დამატებების შექმნა და გამოყენებაც).

³⁴ <http://poi.apache.org>

³⁵ <https://npoi.codeplex.com>

³⁶ <https://code.msdn.microsoft.com/mschart>

³⁷ <https://msdn.microsoft.com/en-us/library/dd456632.aspx>

³⁸ <https://www.devexpress.com>

³⁹ <http://nhibernate.info>

⁴⁰ <http://getbootstrap.com/css>

⁴¹ https://www.wikiwand.com/en/Front-end_web_development

12 მოთხოვნები ფიზიკური არქიტექტურისადმი

12.1 მინიმალური არქიტექტურა

აპარატურული მოთხოვნები:

- CPU: 4x – 2000Mhz
- RAM: 16GB
- HDD: 200GB

პროგრამული გარემო:

- Microsoft Windows Server 2008 R2⁴²
- Microsoft Information Services (IIS)⁴³
- Microsoft Visual Studio.Net 2010/2012/2013⁴⁴
- ASP.NET, DevExpress UI კონტროლები, Microsoft Chart კონტროლები
- NHibernate-თან თავსებადი რელაციური ბაზა (postgresql)

12.2 კომპონენტების არსებული მისამართები

ცხრილი 12. STOP C სისტემის მოდულების მისამართები

STOP C სისტემის კომპონენტი	მისამართი	IP
პორტალი	http://Birthregistry.moh.gov.ge/HMIS/HMIS.StopC.web	172.17.216.53
პორტალის მონაცემთა ბაზა - MongoDB	N/A	172.17.7.132
პორტალის მონაცემთა ბაზა - Microsoft SQL	N/A	172.17.7.132
მომხმარებელთა მართვის მოდული	http://birthregistry.moh.gov.ge/HMIS/HMIS.usermanagement.web	172.17.216.231
მომხმარებელთა მართვის მოდულის მონაცემთა ბაზა	N/A	172.17.7.130
საერთო მონაცემების მოდული	http://birthregistry.moh.gov.ge/HMIS/HMIS.commondata.web	172.17.216.231
საერთო მონაცემების მოდულის მონაცემთა ბაზა	N/A	172.17.7.130
საკომუნიკაციო მოდული	http://birthregistry.moh.gov.ge/HMIS/HMIS.messaging.web	172.17.216.231
SMS მოდული	N/A	
SMS მოდულის მონაცემების ბაზა	N/A	

⁴² [https://technet.microsoft.com/en-us/library/dd349801\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/dd349801(v=ws.10).aspx)

⁴³ <https://www.iis.net>

⁴⁴ <https://www.visualstudio.com>

12.3 რეკომენდაციები რეზერვირების სქემასთან დაკავშირებით

STOP C სისტემის სავარაუდო საშუალო კრიტიკულობის ხარისხიდან გამომდინარე ამ ეტაპისთვის საჭიროებას არ წარმოადგენს მაღალი ხელმისაწვდომობის (High Availability) ტექნოლოგიების დამატებით გამოყენება. თუმცა, სისტემისა და მონაცემების ხარვეზებისგან დაცვის მიზნით რეკომენდებულია სტანდარტული რეზერვირების სქემის გამოყენება შემდეგი პერიოდულობით:

- სრული რეზერვირება (Full backup)
 - სისტემის კომპონენტებისა და მონაცემების სრული რეზერვირება
 - პერიოდულობა: კვირაში ერთხელ
- ინკრემენტული რეზერვირება (Incremental)
 - მხოლოდ იმ მონაცემების რეზერვირება, რომლებიც შეიცვალა უკანასკნელი სრული რეზერვირების შემდეგ
 - პერიოდულობა: საათში ერთხელ

13 ინსტალაცია და კონფიგურაცია

STOP C სისტემის ინსტალაციისთვის ან სარეზერვო ასლის აღდგენისთვის უნდა გამზადდეს პლატფორმა, ზემოთ ჩამოთვლილი სისტემის პროგრამული უზრუნველყოფის კომპონენტების მიხედვით, შემდეგი თანმიმდევრობით:

1. ოპერაციული სისტემის ინსტალაცია და სტანდარტული კონფიგურაცია (Microsoft Windows Server 2008/2012 R2)
2. Microsoft Internet Information Server-ის ინსტალაცია და კონფიგურაცია
3. Microsoft .NET Framework-ის ინსტალაცია და კონფიგურაცია
4. DevExpress ინსტრუმენტებისა და ბიბლიოთეკების კონფიგურირება
5. Microsoft Chart Controls ინსტალაცია და კონფიგურირება
6. მონაცემთა ბაზის სერვერის ინსტალაცია და კონფიგურაცია
 - a. მონაცემთა ბაზისა და ცხრილების შექმნა (ან აღდგენა სარეზერვო ასლიდან) – Microsoft SQL
 - b. მონაცემთა ბაზის შექმნა (ან აღდგენა სარეზერვო ასლიდან) – MongoDB
7. iis.config.cmd ფაილის გაშვება.

14 Appendix A: ვალიდაციების სინტაქსი

ფორმების ველების კომპონენტებისა და ატრიბუტების ვალიდაციებისა და კავშირებისთვის სისტემაში გამოიყენება სპეციალიზებული გამარტივებული სინტაქსი.

14.1 ოპერატორები

ოპერატორი	მნიშვნელობა	კომენტარი
!	ლოგიკური "არ"	მაგალითად არსებობს ფუნქცია IsEmpty(value) რომელიც აბრუნებს true/false (კი/არა) იმისდამხედველად არის თუ არა პარამეტრი (value) ცარიელი, მაგრამ ჩვენ თუ გვინტერესებს შეტრიალებული ვარიანტი (ანუ Is not empty, შეიცავს თუ არა რაიმე მნიშვნელობას value) მაშინ ვწერთ "IsEmpty(value)"
+	მიმატება	
++	ინკრემენტი	მნიშვნელობა იზრდება 1-ით
-	გამოკლება	
--	დეკრემენტი	მნიშვნელობა მცირდება 1-ით
==	ტოლობა/უდრის	მაგალითად a==b, მოწმდება a-ს მნიშვნელობა უდრის თუ არა b-ს
!= <>	უტოლობა	მაგალითად a!=b ან a<>b, მოწმდება a-ს მნიშვნელობა არ უდრის b-ს
<=	ნაკლებია ან უდრის	
>=	მეტია ან უდრის	
<	ნაკლებია	
>	მეტია	
&&	ლოგიკური "და"	მაგალითად a==b && b==c ნიშნავს a უდრის b-ს და b უდრის c-ს
	ლოგიკური "ან"	მაგალითად a==b b==c ნიშნავს a უდრის b-ს ან b უდრის c-ს
^	ხარისხში აყვანა	a^b - a ხარისხად b
&	ტექსტის კონკატენაცია	მაგალითად თუ გვაქვს სახელის და გვარის ველი (firstName და lastName) და გვინდა ამ ორი ველის მნიშვნელობის გაერთიანება ისე რომ შუაში იყოს ტირე მაშინ ვწერთ firstName & ' - ' & lastName
%	პროცენტის დათვლა	a%b იგივეა რაც [a / 100 * b]
*	გამრავლება	
/	გაყოფა	

14.2 ფუნქციები

ფუნქცია	აღწერა
<code>sqrt(value)</code>	ფესვის ამოღება
<code>sin(value)</code>	სინუსი
<code>cos(value)</code>	კოსინუსი
<code>tan(value)</code>	ტანგენსი
<code>asin(value)</code>	არკსინუსი
<code>acos(value)</code>	არკკოსინუსი
<code>atan(value)</code>	არკტანგენსი
<code>abs(value)</code>	აბრუნებს აბსოლუტურ მნიშვნელობას
<code>ceil(value)</code>	ამრგვალებს ათწილადს უახლოესი მაღალ მთელ რიცხვამდე
<code>exp(value)</code>	აბრუნებს e-ს ⁴⁵ მნიშვნელობას მითითებულ ხარისხში
<code>floor(value)</code>	ამრგვალებს ათწილადს უახლოესი დაბალ მთელ რიცხვამდე
<code>log(value, base)</code>	ლოგარიტმის ამოღება, მითითებული ფუძით
<code>log10(value)</code>	ლოგარიტმის ამოღება 10-ის ფუძით
<code>pow(value, base)</code>	ახარისხება მითითებული ფუძით
<code>or(value1, value2)</code>	ბინარული "ან" ოპერაცია ⁴⁶
<code>and(value1, value2)</code>	ბინარული "და" ოპერაცია ⁴⁷
<code>xor(value1, value2)</code>	გამომრიცხავი „ან“ (XOR) ⁴⁸
<code>mod(value1, value2)</code>	მოდულიური ექსპონენტა ⁴⁹ - აბრუნებს გაყოფის ნაშთს (მაგ: <code>mod(19, 4)</code> დააბრუნებს 3 რადგან 19 გაყოფილი 4-ზე უდრის 4-ს და ნაშთი არის 3)
<code>substring(value, index)</code>	აბრუნებს ქვე-ტექსტს დაწყებული index რიგით სიმბოლოდან ბოლომდე (მაგ: თუ value-ს მნიშვნელობა არის abcdefg, მაშინ <code>substring(value, 3)</code> დააბრუნებს defg რადგან, ათვლის სისტემა იწყება 0-დან ანუ 3-ე სიმბოლო არის d)
<code>substring(value, index, count)</code>	აბრუნებს ქვე-ტექსტს დაწყებული index რიგით სიმბოლოდან count-ში მითითებული სიმბოლოების რაოდენობით (მაგ: თუ value-ს მნიშვნელობა არის abcdefg, მაშინ <code>substring(value, 3, 2)</code> დააბრუნებს de-ს, რადგან ათვლის სისტემა იწყება 0-დან ანუ 3-ე სიმბოლო არის d)
<code>getdatetime(value)</code>	აბრუნებს მიმდინარე თარიღს და დროს
<code>getDate()</code>	აბრუნებს მიმდინარე თარიღს
<code>getDate(year)</code>	აბრუნებს თარიღს (თარიღის დანარჩენი წევრები (თვე, დღე და სხვა) იქნება მიმდინარე თარიღის შესაბამისი)
<code>getDate(year, month)</code>	აბრუნებს თარიღს (თარიღის დანარჩენი წევრები (დღე, საათი და სხვა) იქნება მიმდინარე თარიღის შესაბამისი)
<code>getDate(year, month, day)</code>	აბრუნებს თარიღს (საჭიროა თარიღის ოპერაციებისთვის მაგალითად <code>addDays(getDate(2017, 02, 21), -15)</code>)
<code>getLength(value)</code>	აბრუნებს სიგრძეს (სიმბოლოების რაოდენობას)

⁴⁵ [https://en.wikipedia.org/wiki/E_\(mathematical_constant\)](https://en.wikipedia.org/wiki/E_(mathematical_constant))

⁴⁶ https://en.wikipedia.org/wiki/Bitwise_operation#OR

⁴⁷ https://en.wikipedia.org/wiki/Bitwise_operation#AND

⁴⁸ https://en.wikipedia.org/wiki/Exclusive_or

⁴⁹ https://en.wikipedia.org/wiki/Modular_exponentiation

isEmpty(value)	არის თუ არა ცარიელი
isDate(value)	არის თუ არა თარიღი
isDay(value)	არის თუ არა დღე (ანუ მეტია 0-ზე და ნაკლებია 32-ზე)
isMonth(value)	არის თუ არა თვე (ანუ მეტია 0-ზე და ნაკლებია 13-ზე)
isYear(value)	არის თუ არა წელი
isNumber(value)	არის თუ არა რიცხვი
isInteger(value)	არის თუ არა მთელი რიცხვი
rgx(value, exp)	ამოწმებს ტექსტს Regex-ის ⁵⁰ საშუალებით
min(a, b)	აბრუნებს მინიმალურს
max(a, b)	აბრუნებს მაქსიმალურს
if(value, a, b)	მაგ.: if(x==1, a, b) - თუ x უდრის ერთს დააბრუნებს a თუ არა და b
comp(value)	კომპარატორი (აბრუნებს -1, 0, 1)
lower(value)	გადაყავს პატარა სიმბოლოებში
upper(value)	გადაყავს მთავრულ სიმბოლოებში
contains(value, text)	შეიცავს თუ არა პირველ პარამეტრში გადაცემული ტექსტი მეორე პარამეტრში მითითებულ ტექსტს (აბრუნებს true/false), მაგალითად contains(value, "hello")- შეიცავს თუ არა value-ში არსებული ტექსტი სიტყვა hello-ს
startsWith(value, text)	იწყება თუ არა value-ში არსებული ტექსტი text-ში
endsWith(value, text)	მთავრდება თუ არა value-ით text
trim(value)	აშორებს space-ებს თავში და ბოლოში
LTrim(value)	აშორებს space-ებს თავში
RTrim(value)	აშორებს space-ებს ბოლოში
get(collection, nameOrIndex)	თუ პარამეტრად გადაეცა მასივი, მაშინ nameOrIndex პარამეტრში უნდა გადაეცეს ელემენტის ნომერი; ხოლო თუ გადაეცა Dictionary მაშინ nameOrIndex უნდა ჩანაწერის key. მაგალითად, ცხრილის ჩანაწერები არის მასივი, თუ გვინდა ამოვიღოთ ცხრილიდან მე-10 ჩანაწერი, გვწერს get(Grid, 10); ხოლო ცხრილის თითოეული ჩანაწერი არის Dictionary, რადგან ცხრილის თითოეული ჩანაწერის ველებს შეესაბამება ამ ცხრილი სვეტი, რომელსაც აქვს სახელი. ამიტომ, თუ გვინდა ამოვიღოთ ცხრილის 10-ე ჩანაწერის Column3 სვეტის მნიშვნელობა, გვწერს get(get(Grid, 10), 'Column3'). ათვლა იწყება ნოლიდან ანუ ცხრილი პირველი ჩანაწერის ინდექსი არის 0. ანუ თუ გვინდა პირველი ჩანაწერის ამოღება get(Grid, 0), მაგრამ იმ შემთხვევაში თუ collection (მაგ. ცხრილი) ცარიელია, ეს ჩანაწერი გამოიწვევს შეცდომას
getDaysInMonth(value) getDaysInMonth(value, year)	აბრუნებს დღეების რაოდენობას თვის მიხედვით (თუ year პარამეტრში წელი არ მიეთითა მაშინ იგულისხმება მიმდინარე წელი) (მაგ: getDaysInMonth(2) დააბრუნებს 28-ს მაგრამ getDaysInMonth(2, 2016) დააბრუნებს 29-ს)
getYears(date)	აბრუნებს წელს თარიღის მიხედვით (2017-02-17 15:32:25)
getMonths(date)	აბრუნებს წელს თარიღის მიხედვით (2017-02-17 15:32:25)

⁵⁰ https://en.wikipedia.org/wiki/Regular_expression

<code>getDays(date)</code>	აბრუნებს დღეს (2017-02-17 15:32:25) (თუ პარამეტრი არ გადაეცა მაშინ აბრუნებს მიმდინარე თვის მიმდინარე დღეს)
<code>getDayOfMonth(date)</code>	კვირის მერამდენე დღეა
<code>getDayOfWeek(date)</code>	აბრუნებს წლის მერამდენე დღეა
<code>getHours(date)</code>	აბრუნებს საათებს (2017-02-17 15:32:25)
<code>getMinutes(date)</code>	აბრუნებს წუთებს (2017-02-17 15:32:25)
<code>getSeconds(date)</code>	აბრუნებს წამებს (2017-02-17 15:32:25)
<code>getTotalDays(date)</code>	აბრუნებს დღეების ჯამურ რაოდენობას 0001-01-01 01:01:01-დან მითითებულ თარიღამდე
<code>getTotalHours(date)</code>	აბრუნებს საათების ჯამურ რაოდენობას 0001-01-01 01:01:01-დან მითითებულ თარიღამდე
<code>getTotalHours(date)</code>	აბრუნებს წუთების ჯამურ რაოდენობას 0001-01-01 01:01:01-დან მითითებულ თარიღამდე
<code>getTotalSeconds(date)</code>	აბრუნებს წამების ჯამურ რაოდენობას 0001-01-01 01:01:01-დან მითითებულ თარიღამდე ⁵¹
<code>addYears(date, count)</code>	უმატებს წლებს პარამეტრად გადაცემულ თარიღს (count რაოდენობა) ⁵²
<code>addMonths(date, count)</code>	უმატებს თვეებს პარამეტრად გადაცემულ თარიღს (count რაოდენობა)
<code>addDays(date, count)</code>	უმატებს დღეებს პარამეტრად გადაცემულ თარიღს (count რაოდენობა)
<code>addHours(date, count)</code>	უმატებს საათებს პარამეტრად გადაცემულ თარიღს (count რაოდენობა)
<code>addMinutes(date, count)</code>	უმატებს წუთებს პარამეტრად გადაცემულ თარიღს (count რაოდენობა)
<code>addSeconds(date, count)</code>	უმატებს წამებს პარამეტრად გადაცემულ თარიღს (count რაოდენობა)
<code>select(collection, selector, condition)</code>	ფუნქცია აბრუნებს condition-ში გაწერილი პირობების მიხედვით იმ მონაცემებს რომელიც გაწერილია selector-ში. მაგალითად ფორმაში არის ცხრილი Products, რომელსაც აქვს სამი სვეტი Name, Amount, Cost და გვინდა ამოვიღოთ ყველა ისეთი პროდუქტის სახელი და ფასი, რომლის რაოდენობა არის 10-ზე მეტი: <pre>select(Products, get(@, 'Name') & ' - ' & get(@, 'Cost'), get(@, 'Amount') > 10)</pre> @ - ნიშნავს მიმდინარე ჩანაწერს, ანუ ფუნქცია select-მა უნდა გაიაროს ცხრილის ყველა ჩანაწერი, შეამოწმოს აკმაყოფილებს თუ არა თითოეული ჩანაწერი condition-ში გაწერილ პირობებს და

⁵¹ ზედა ოთხი ფუნქცია საჭიროა მაშინ, როცა გვინტერესებს ორ თარიღს შორის დღეების, საათების, წუთების ან წამების რაოდენობა. მაგალითად getTotalHours(Date1 - Date2) დააბრუნებს Date1-სა და Date2-ს შორის სხვაობას საათებში

⁵² თარიღის მიმატების ყველა ფუნქციისთვის დასაშვებია უარყოფითი მნიშვნელობები (მაგალითად თუ გვინდა რომ რაღაც თარიღს გამოვაკლოთ 10 დღე მაშინ ვწერთ addDays(date, -10))

	დააბრუნოს selector-ში გაწერილი გამოსახულების მიხედვით შედეგი, ანუ @ ნიშნავს იტერაციულად გადავლისას იმ ჩანაწერს რომელზეც არის იმ მომენტში
sum(collection)	ფუნქცია აბრუნებს collection-ში გადაცემული მონაცემები ჯამს Select-ის მაგალითის ანალოგიურად, თუ Products ცხრილიდან გვინდა ისეთი პროდუქტების რაოდენობის ჯამის დათვლა, რომელთა ფასიც აღემატება 100-ს: <pre>sum(select(Products, get(@, 'Amount'), get(@, 'Cost') > 100))</pre> min - მუშაობს sum ფუნქციის ანალოგიურად იმ განსხვავებით რომ აბრუნებს მინიმალურ მნიშვნელობას max - მუშაობს sum ფუნქციის ანალოგიურად იმ განსხვავებით რომ აბრუნებს მაქსიმალურ მნიშვნელობას