



ადმინისტრატორის სახელმძღვანელო

ს შეპათიტის სკრინინგის ელექტრონული
ადრიცხვის სისტემა

სარჩევი

1	სურათებისა და დიაგრამების სარჩევი	4
2	დოკუმენტის დანიშნულება.....	6
3	STOP C - მოკლე აღწერილობა და ფუნქციონალი	6
4	STOP C პროგრამული ეკოსისტემა	6
4.1	NCDC.eHealth	7
4.2	MOH.ElimC.....	7
4.3	MOH.eHealth	7
4.4	სხვა დაკავშირებული მოდულები.....	7
5	STOP C მოდულების ზოგადი აღწერა და დანიშნულება.....	8
6	ადმინისტრირების სქემა	9
7	ინტერფეისის ძირითადი კომპონენტები	11
7.1	ინტერფეისის თავსართი	11
7.2	ნავიგაციის პანელი	11
8	ფორმების ადმინისტრირება	12
8.1	კატეგორიები	12
8.2	ფორმები.....	13
8.2.1	ახალი ფორმის დამატება.....	14
8.2.2	ფორმის ელემენტების დამატება (<i>Elements</i>).....	15
8.2.3	Expressions.....	17
8.2.4	Dependency	17
8.2.5	Field - ველის ელემენტი	17
8.2.6	Grid	18
8.2.7	Tree (იერარქიული სია).....	19
8.2.8	Group (ჯგუფი)	19
8.2.9	TabContainer	21
8.3	ლოგიკები.....	22
8.4	ცნობარები.....	22
9	დამატებითი ბმულები.....	25
9.1	შეტყობინებები.....	25
9.2	კალენდარი	25
9.3	კომპონენტი - სკრინინგის ფორმა	25
10	ანალიტიკა	26

11	რეპორტი.....	26
12	APPENDIX A: ვალიდაციების სინტეზი.....	29
12.1	ოპერატორები	29
12.2	ფუნქციები	34

1 სურათებისა და დიაგრამების სარჩევი

სურათი 1. STOP C მოდულები პროგრამული არქიტექტურის ჭრილობში	8
სურათი 2. STOP C ფორმის, მონაცემებისა და პროცესის დიაგრამა.....	9
სურათი 3. სკრინინგის ფორმის ადმინისტრირების ზოგადი სქემა	10
სურათი 4. ადმინისტრატორის პანელის კომპონენტები: (1) ინტერფეისის თავსართი (2) ნავიგაციის პანელი (3) ძირითადი პანელი.....	10
სურათი 5. ინტერფეისის თავსართი: (1) მთავარი მენიუს რეჟიმის ცვლილება (2) პაროლის შეცვლა (3) სისტემიდან გასვლა (4) პორტალის ქართულ ვერსიაზე გადასვლა /პორტალის ინგლისურ ვერსიაზე გადასვლა და თარგმნის რეჟიმი.	11
სურათი 6	11
სურათი 7	12
სურათი 8	12
სურათი 9	12
სურათი 10	12
სურათი 11. კატეგორიის დამატება: (1) კატეგორიის დასახელება (2) ჩანდეს თუ არა კატეგორია ძირითად მენიუში (3) ძირითად მენიუში კატეგორიის რიგითობის განსაზღვრა (4) შენახვა (5) დახურვა (შენახვის გარეშე).....	13
სურათი 12. ფორმების კატეგორიების სია (1) კატეგორიის დამატება (2) კატეგორიის ძებნა (3) კატეგორიის ნახვა (4) რედაქტირება (5) წაშლა (6) კატეგორიის დასახელება (7) მთავარ მენიუში ჩვენების რიგითობის ინდექსი (8) მთავარ მენიუში ხილვადობის მაჩვენებელი.....	13
სურათი 13. ფორმის მზა ვერსია.....	14
სურათი 14. ფორმების სია: (1) ახალი ფორმის დამატება (2) ფორმის ძებნა (3) მზა ფორმის ნახვა (4) ფორმის კონფიგურაციის ნახვა (5) რედაქტირება (6) ფორმის დაკოპირება (დუბლირება) (7) ფაილების ატვირთვა ფორმაზე (8) ფორმის წაშლა (9) უნიკალური ნომერი (10) ფორმის რიგითობა სიაში	14
სურათი 15. ფორმის ბაზისური ატრიბუტები.....	15
სურათი 16. ელემენტის დამატების ზოგადი ინტერფეისი.....	16
სურათი 17. ველის (Field) ელემენტის ატრიბუტები.....	18
სურათი 18. ცხრილის (Grid, მატრიცა) ელემენტის ატრიბუტები.....	19
სურათი 19. ჯგუფის ელემენტის (Group) ატრიბუტები.....	20
სურათი 20. იერარქიული სიის (Tree, ხე) ელემენტის ატრიბუტები.....	20
სურათი 21. ელემენტთა გამაერთიანებელი ტაბ კონტეინერის ატრიბუტები.....	21
სურათი 22. ფორმის ელემენტების სია: (1) ელემენტის დამატება (2) რედაქტირება (3) წაშლა (4) კოპირებულის ჩასმა (5) კოპირება (6) ელემენტის დამატება (Grid, Tree და Group ტიპებისთვის)	21
სურათი 23. ფორმის შაბლონის შენახვა	22
სურათი 24. ცნობარების ადმინისტრირება: (1) ცნობარის დამატება (2) ცნობარის ძებნა (3) ცნობარის მონაცემები (4) ცნობარის სტრუქტურის რედაქტირება (5) ცნობარის წაშლა	22
სურათი 25. ცნობარში მონაცემების შეტანა: (1) ახალი ჩანაწერის დამატება (2) ჩანაწერის წაშლა (3) მონაცემების იმპორტი (4) დახურვა.....	23
სურათი 26. ახალი ცნობარის დამატება (1) ცნობარის შენახვა (2) დახურვა (3) ცნობარში ველის დამატება (4) ველის სახელი (5) ველის შენახვა (6) დახურვა	23
სურათი 28. შეტყობინების გაგზავნის ფუნქციონალი	24

სურათი 27. ცნობარში მონაცემების შეტანა: ახალი ჩანაწერის დამატებისთვის საჭირო ველები	24
სურათი 29. მართვის ძირითადი ინსტრუმენტები: (1) მონაცემის დათვალიერება (2) მონაცემის რედაქტირება (3) მონაცემის ბეჭდვა (4) მონაცემის წაშლა (5) მონაცემის დამატება.....	25
სურათი 30. HCV სკრინინგის კვლევის შედეგის რეპორტი.....	26
სურათი 31. ნიმუშების და კონფირმაციული კვლევის შედეგის რეპორტი.....	26
სურათი 33. სამაგალითო სტატიკური ცხრილის გრაფიკული გამოსახულება და მართვის ინსტრუმენტები: (1) მონაცემების ჩამოტვირთვა. (2) აქსისის ღერძის მონაცემების ფილტრი (3) ჩარტის ტიპის ცვლილება (4) ცხრილის გრაფიკული გამოსახულება (5) მონაცემების ცხრილის სახით ვიზუალიზაცია.....	27
სურათი 32. სტატიკური ცხრილის ტიპის ცვლილება(მათ შორის 3 განზომილებაში).....	28

2 დოკუმენტის დანიშნულება

ადმინისტრატორის სახელმძღვანელო წარმოადგენს C ჰეპატიტის სკრინინგის ელექტრონული მოდულის - „STOP C“ სისტემის ¹ დანერგვისა და გამართვის პროექტის ფუნდამენტურ ნაწილს. დოკუმენტი აღწერს სისტემასთან ადმინისტრატორის როლით მუშაობის პრინციპებს - ფუნქციონალს და მომხმარებლისთვის აუცილებელ დეტალურ ინსტრუქციას. მომხმარებლის სახელმძღვანელო დოკუმენტი განკუთვნილია სისტემის ადმინისტრატორებისა და დეველოპერებისთვის და შეიცავს ყველა საჭირო და აუცილებელ ინფორმაციას სისტემის მართვის, ადმინისტრირებისა და მხარდაჭერისთვის.

მიმდინარე დოკუმენტი არ მოიცავს სისტემის დეტალური ტექნიკური არქიტექტურის აღწერილობას. ამ მიზნისთვის მნიშვნელოვან დოკუმენტს წარმოადგენს „STOP C“ სისტემის ტექნიკური არქიტექტურის დოკუმენტი ², რომელიც ასევე არის სისტემური ტექნიკური დოკუმენტაციის აუცილებელი ნაწილი.

3 STOP C - მოკლე აღწერილობა და ფუნქციონალი

C ჰეპატიტის სკრინინგის ელექტრონული მოდული - „STOP C“ წარმოადგენს C ჰეპატიტის მართვის სახელმწიფო პროგრამის ფარგლებში გათვალისწინებული გარკვეული სამედიცინო სერვისების ელექტრონული აღრიცხვისა და მონაცემთა მართვის სისტემას, რომელიც უზრუნველყოფს შემდეგი ძირითადი ფუნქციების რეალიზაციას:

- სკრინინგის კომპონენტით გათვალისწინებული C ჰეპატიტზე სკრინინგის შედეგების აღრიცხვა
- დიაგნოსტიკის კომპონენტით გათვალისწინებული კონფირმაციული კვლევისთვის საჭირო სისხლის ნიმუშის აღებასთან და ტრანსპორტირებასთან დაკავშირებული მონაცემების ელექტრონული რეგისტრაცია
- ლუგარის ლაბორატორიის მიერ საკვლევი ნიმუშზე ჩატარებული კონფირმაციული კვლევის შედეგების რეგისტრაცია
- ლუგარის ლაბორატორიის მიერ საკვლევი ნიმუშზე ჩატარებული კონფირმაციული კვლევის შედეგების მკურნალობის ბაზაში სინქრონიზაცია

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

4 STOP C პროგრამული ეკოსისტემა

გარდა STOP C ანალიტიკური 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 სისტემური არქიტექტურის დოკუმენტის ლინკი

4.1 NCDC.eHealth

ეს მოდულების მოიცავს დაავადებათა კონტროლისა და საზოგადოებრივი ჯანდაცვის ეროვნული ცენტრის³ (NCDC) პროცესების მართვის სხვადასხვა პროგრამულ ინსტრუმენტებს, რომლის ნაწილიცაა STOP C მონაცემების ანალიტიკის მოდულიც. ანალიტიკური მოდულის გარდა ამ კომპონენტის შემადგენლობაშიც შედის STOP C-ს პროგრამების მართვის შემდეგი მოდულები:

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

4.2 MOH.ElimC

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

- MOH.ElimC.Treatment: მკურნალობა
- MOH.ElimC.SSA: SSA კომისია
- MOH.ElimC.Drugs: მედიკამენტების მართვა
- MOH.ElimC.Monitoring: მონიტორინგი

4.3 MOH.eHealth

საქართველოს შრომის, ჯანმრთელობისა და სოციალური დაცვის ელექტრონული ჯანდაცვის მოდულები წარმოადგენს კომპლექტურ 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:

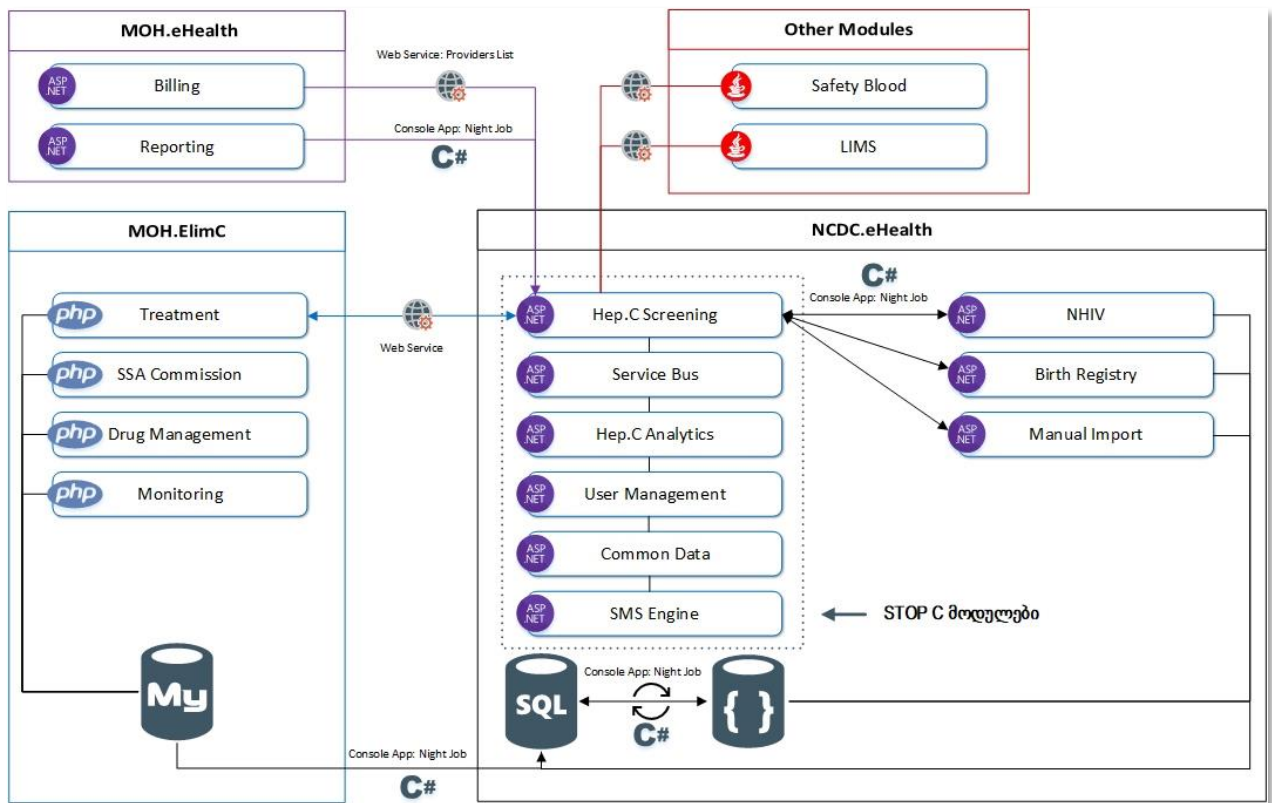
- MOH.eHealth.Billing: ფინანსური მოდული
- MOH.eHealth.Reporting: რეპორტირების მოდული

4.4 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

³ <http://www.ncdc.ge>



სურათი 1. STOP C მოდულები პროგრამული არქიტექტურის ჭრილში

5 STOP C მოდულების ზოგადი აღწერა და დანიშნულება

STOP C მოდული შედგება 3 ძირითადი და 2 დამხმარე ქვე-მოდულისგან:

- ძირითადი მოდულები
 - მომხმარებელთა მართვის მოდული (User Management Module, UMM)
 - სისტემის მომხმარებელთა და წვდომების უფლებების მართვის ინსტრუმენტი
 - ცენტრალური მოდული / პორტალი (Portal)⁴
 - STOP C სისტემის ძირითადი ფუნქციონალი, ნავიგაციის ინსტრუმენტები
 - ანალიტიკური მოდული (Analytics)⁵
 - სკრინინგისა და მასთან დაკავშირებული სისტემებისა და მოდულების მონაცემების დინამიური რეპორტირება და ანალიტიკა
- დამხმარე მოდულები
 - საერთო მონაცემების მოდული (Common Data Module, CDM)
 - მოდულებისთვის საერთო მონაცემების შენახვა და დამუშავება
 - მრავალენოვანი მხარდაჭერა და ლოგირება
 - საკომუნიკაციო მოდული (Service Bus, SB)

⁴ მიმდინარე დოკუმენტი აღწერს ცენტრალურ პორტალთან მუშაობის პრინციპებს, ადმინისტრატორის წვდომის მომხმარებლისთვის

⁵ ანალიტიკური მოდულის ადმინისტრირების სახელმძღვანელო და სისტემური არქიტექტურა წარმოდგენილია დამატების დოკუმენტაციაში

- [C ჰეპატიტის მონაცემების ანალიტიკის სისტემის ადმინისტრატორის სახელმძღვანელო](#)

- [C ჰეპატიტის მონაცემების ანალიტიკის სისტემის არქიტექტურა](#)

- მოდულებს შორის მონაცემების მიმოცვლის უზრუნველყოფა
 - SMS მოდული
 - ამჟამად არ გამოიყენება, იგეგმება სამომავლო იმპლემენტაცია

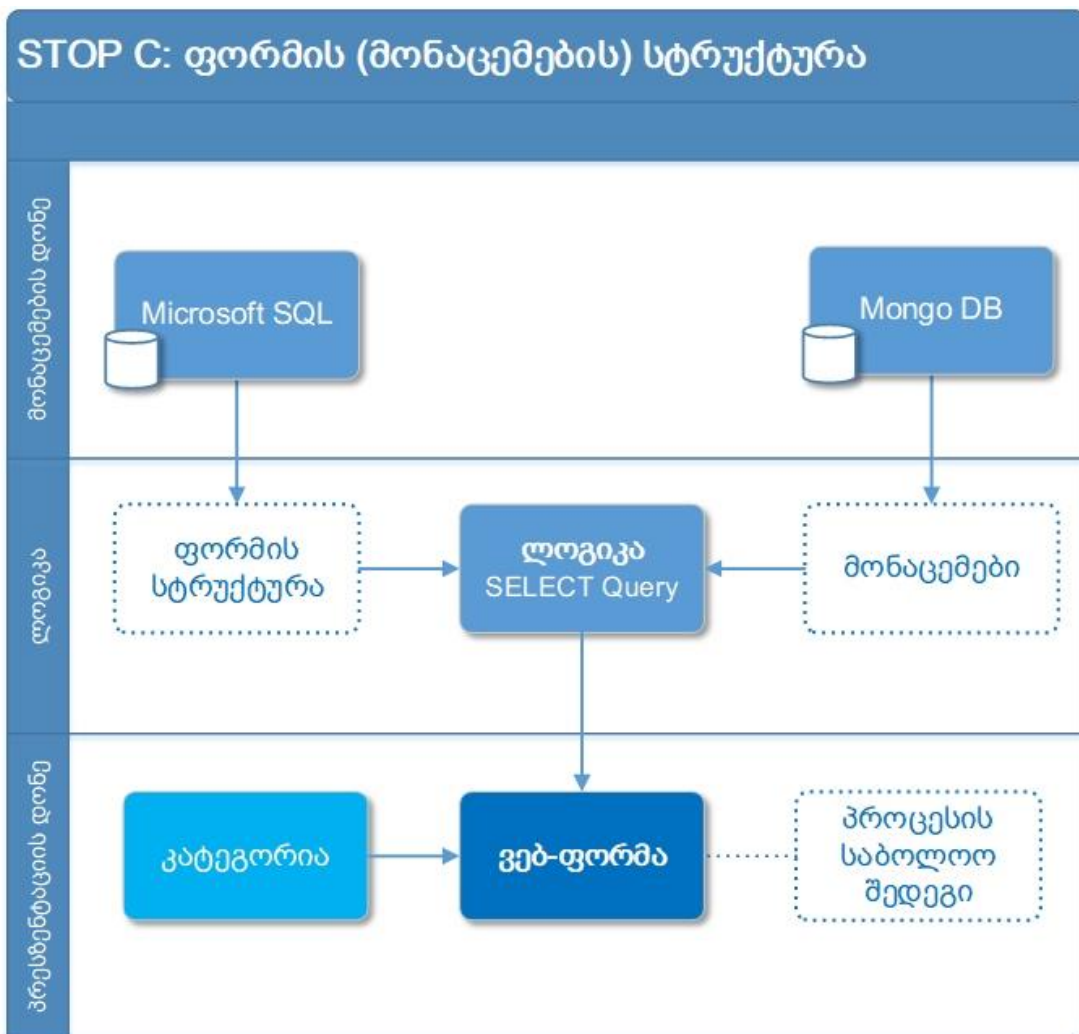
სურათი 1-ზე წარმოდგენილია სისტემის ფუნქციის განხორციელების ტექნიკური გადაჭრის დიაგრამას C ჰეპატიტის პროგრამულ ეკოსისტემაში.

6 ადმინისტრირების სქემა

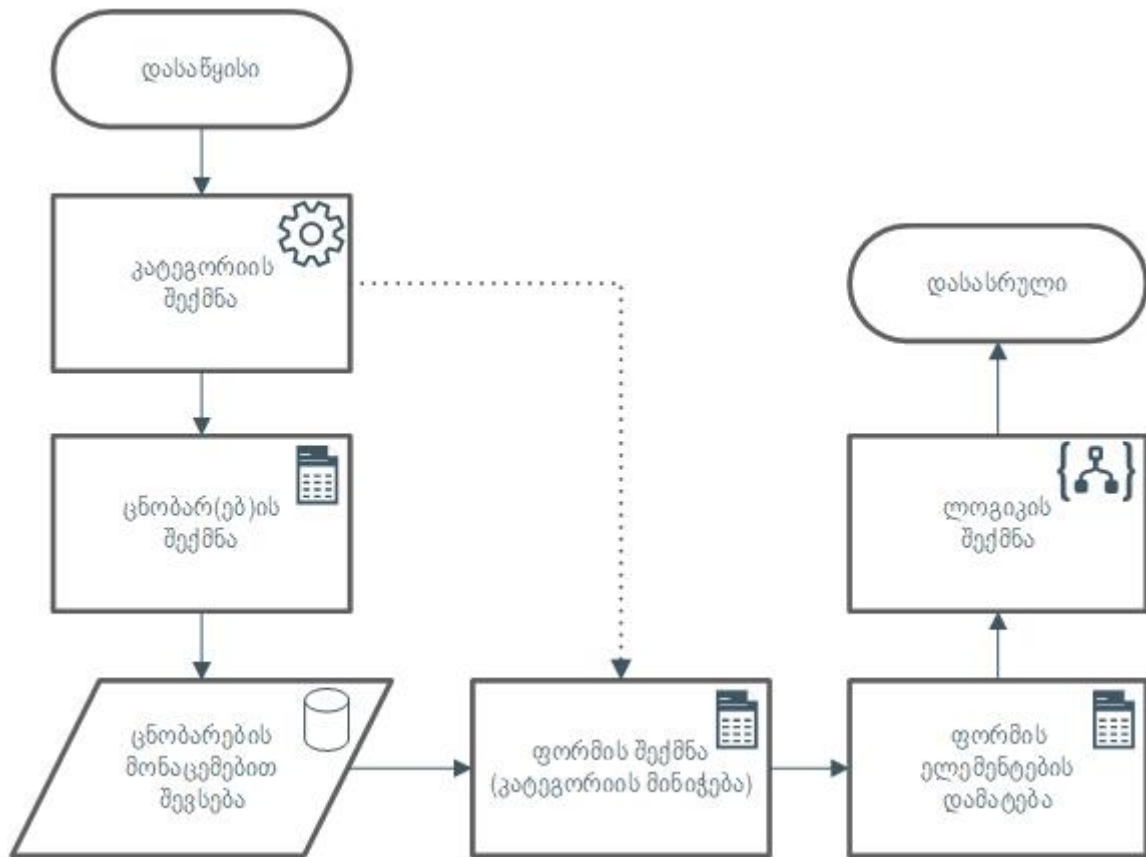
სისტემის ადმინისტრირების ძირითადი ელემენტია ვებ-ფორმები, რომლებიც შექმნა, მოდიფიცაცია და წაშლა შესაძლებელია ადმინისტრატორის ინტერფეისით.

ფორმებთან მუშაობის ზოგადი სქემა შემდეგია:

1. საწყის ეტაპზე საჭიოა ფორმის კატეგორიის შექმნა, რომელიც უნდა განესაზღვროს ყველა მომავალში შექმნილ ფორმას ფორმას
2. ცნობარების შექმნა ფორმებისთვის
 - a. ცნობარების მონაცემებით შევსება
3. ახალი ფორმის შექმნა
 - a. ფორმის კატეგორიის განსაზღვრა და არჩევა კატეგორიების არსებული სიიდან



სურათი 2. STOP C ფორმის, მონაცემებისა და პროცესის დიაგრამა

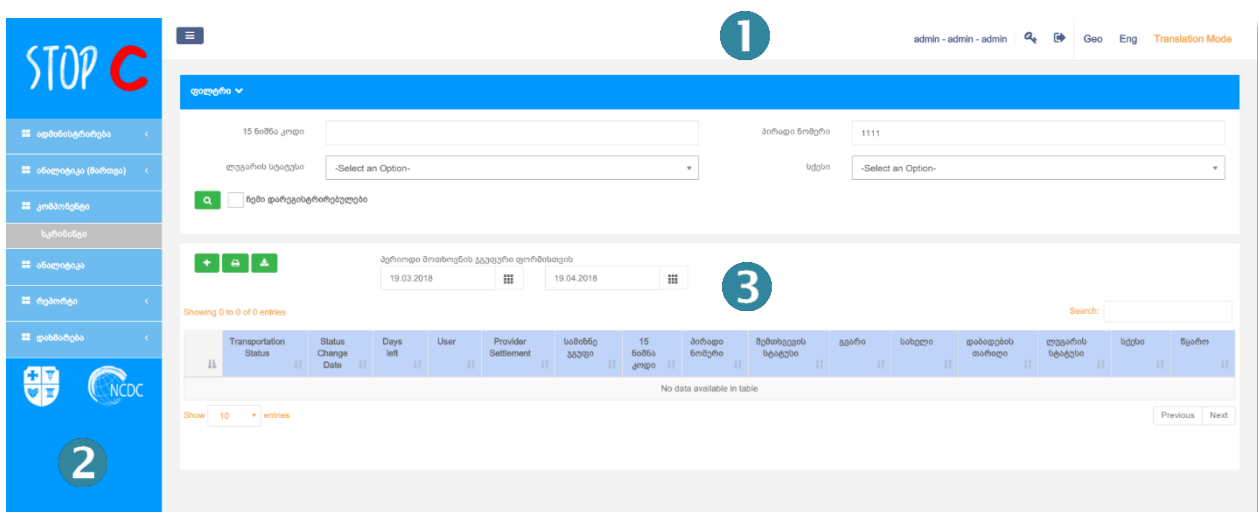


სურათი 3. სკრინინგის ფორმის ადმინისტრირების ზოგადი სქემა

b. ფორმის ელემენტების დამატება

4. ფორმაში მონაცემების შევსების ლოგიკის განსაზღვრა

სურათი 3 წარმოადგენს ამ პროცესის სქემატურ დიაგრამას, ხოლო სურათი 2-ზე მოცემულია იგივე პროცესის კომპონენტების ვიზუალიზაციის სხვადასხვა ჭრილში (მონაცემების, ლოგიკისა და პრეზენტაციის დონეებზე).



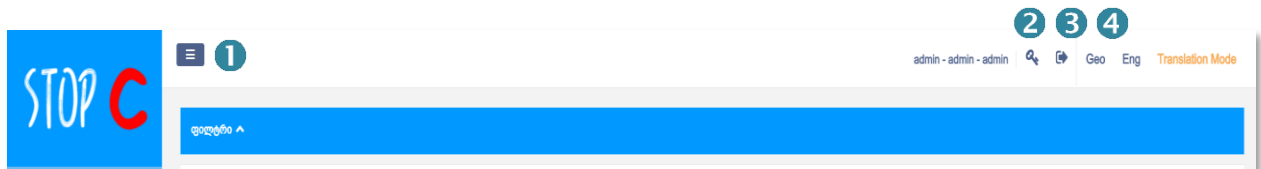
სურათი 4. ადმინისტრატორის პანელის კომპონენტები: (1) ინტერფეისის თავსართი (2) ნავიგაციის პანელი (3) ძირითადი პანელი

7 ინტერფეისის ძირითადი კომპონენტები

ინტერფეისის მთავარ გვერდზე განლაგებულია სისტემის მართვის ძირითადი კომპონენტები (სურათი 4).

7.1 ინტერფეისის თავსართი

ინტერფეისის თავსართი წარმოადგენს პორტალის რამდენიმე ფუნდამენტური პარამეტრის ცვლილებისთვის საჭირო ბმულების ნაკრებს (სურათი 5). თავსართიდან შესაძლებელია ასევე პორტალის ენის არჩევა ქართულ და ინგლისურ ენებს შორის.



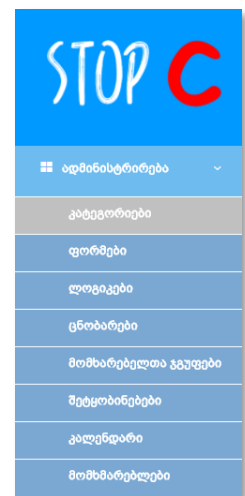
სურათი 5. ინტერფეისის თავსართი: (1) მთავარი მენიუს რეჟიმის ცვლილება (2) პაროლის შეცვლა (3) სისტემიდან გასვლა (4) პორტალის ქართულ ვერსიაზე გადასვლა /პორტალის ინგლისურ ვერსიაზე გადასვლა და თარგმნის რეჟიმი.

7.2 ნავიგაციის პანელი

ნავიგაციის პანელი წარმოადგენს ბმულების ბაზისურ ნაკრებს, რომელთა საშუალებითაც შესაძლებელია სისტემის ადმინისტრირების უმთავრესი კატეგორიის ფუნქციების მართვის გვერდებზე გადასვლა.

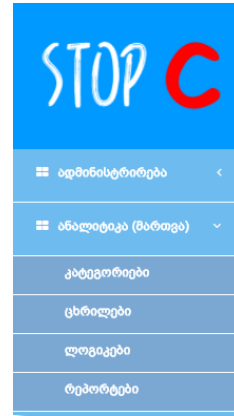
ნავიგაციის პანელი მოიცავს შემდეგ ბმულებს:

1. ადმინისტრირების კომპონენტები (სურათი 6):
 - a. კატეგორიები
 - b. ფორმები
 - c. ლოგიკები
 - d. ცნობარები
 - e. მომხმარებელთა ჯგუფები
 - f. შეტყობინებები
 - g. კალენდარი
 - h. მომხმარებლები



სურათი 6

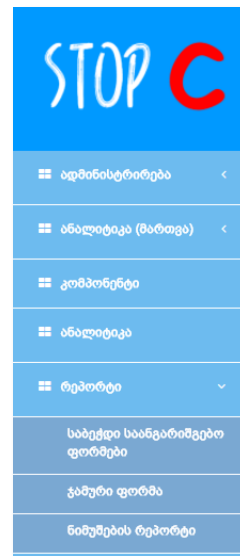
2. ანალიტიკა/მართვა (სურათი 7)^{6 7}
 - a. კატეგორიები
 - b. ცხრილები
 - c. ლოგიკები
 - d. რეპორტები
3. კომპონენტი - მომხმარებლის სამუშაო ბმული (სურათი 8).
4. ანალიტიკა - ანალიტიკის მართვის ბმულიდან შექმნილი სტატიკური შაბლონების გრაფიკული და ცხრილების სახით ვიზუალიზაცია
5. რეპორტი (სურათი 10).
 - a. საბეჭდი საანგარიშგებო ფორმები
 - b. ჯამური ფორმა
 - c. ნიმუშების რეპორტი
6. დახმარება - კატეგორიაში განთავსებულია სხვადასხვა ტიპის მომხმარებლისთვის სისტემით სარგებლობის სახელმძღვანელოები და C ჰეპატიტის სახელმწიფო პროგრამის კანონმდებლობით გათვალისწინებული გაიდლაინები და პროტოკოლები, რომელიც სისტემის ბიზნეს პროცესთან არის დაკავშირებული (სურათი 9).



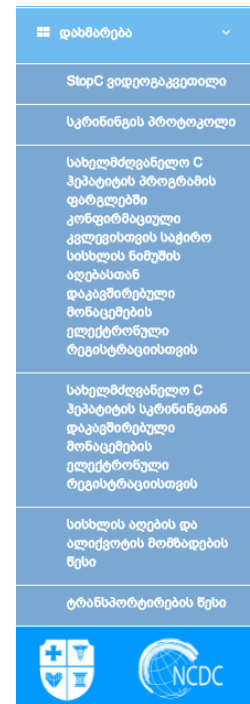
სურათი 7



სურათი 8



სურათი 10



სურათი 9

8 ფორმების ადმინისტრირება

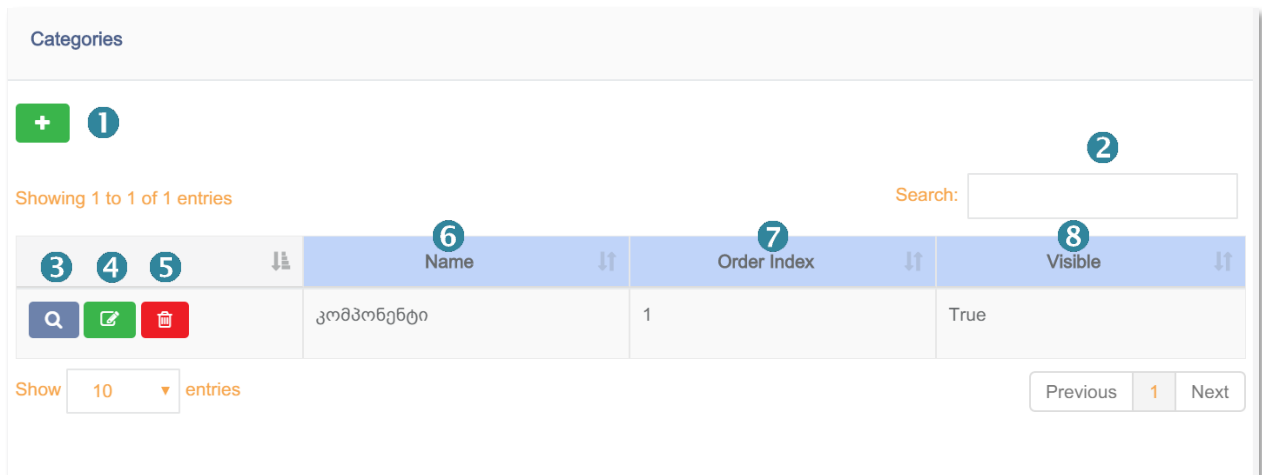
ფორმები წარმოადგენს სისტემის ძირითად ელემენტს, რომელთა საშუალებითაც ხდება პორტალის ფუნქციონირებისთვის საჭირო web-ელემენტების დინამიური ადმინისტრირება.

8.1 კატეგორიები

პორტალის ფორმის შექმნამდე აუცილებელია შეიქმნას ფორმის დამატებული კატეგორია, რომელშიც გაერთიანდება სხვადასხვა ფორმა. კატეგორიები ჯგუფდება ადმინისტრირების მენიუში. კატეგორიების განყოფილებაში გადასვლისას ჩანს კატეგორიების სია (სურათი 12).

⁶ აღსანიშნავია, რომ STOP C –ს სისტემაში რეგისტრირებული მონაცემების სტატისტიკური და ანალიტიკური დამუშავება ძირითადად წარმოებს ანალიტიკურ მოდულში შემდეგ ელექტრონულ მისამართზე: <http://birthregistry.moh.gov.ge/Hmis/Hmis.Dashboards.Web>.

⁷ ანალიტიკური მოდულის წარმოების წესი ადმინისტრატორის ტიპის მომხმარებლისთვის აღწერილია ცალკე სახელმძღვანელოში.



Categories

+ 1

Showing 1 to 1 of 1 entries

Search:

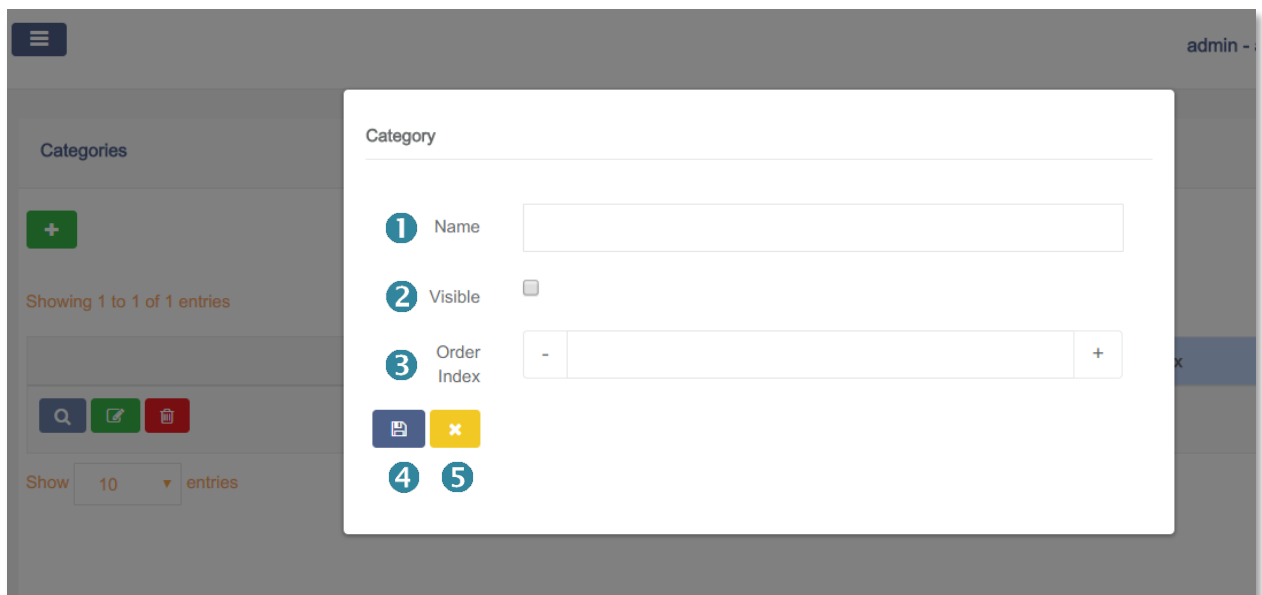
3	4	5	6 Name	7 Order Index	8 Visible
			კომპონენტი	1	True

Show 10 entries

Previous 1 Next

სურათი 12. ფორმების კატეგორიების სია (1) კატეგორიის დამატება (2) კატეგორიის ძეგნა (3) კატეგორიის ნახვა (4) რედაქტირება (5) წაშლა (6) კატეგორიის დასახელება (7) მთავარ მენიუში ჩვენების რიგითობის ინდექსი (8) მთავარ მენიუში ხილვადობის მაჩვენებელი

დამატების ღილაკზე დაჭერით გამოჩნდება კატეგორიის დამატების ფანჯარა (სურათი 11).



Categories

+ 1

Showing 1 to 1 of 1 entries

Search:

Show 10 entries

admin -

Category

1 Name

2 Visible ☐

3 Order Index - +

4 5

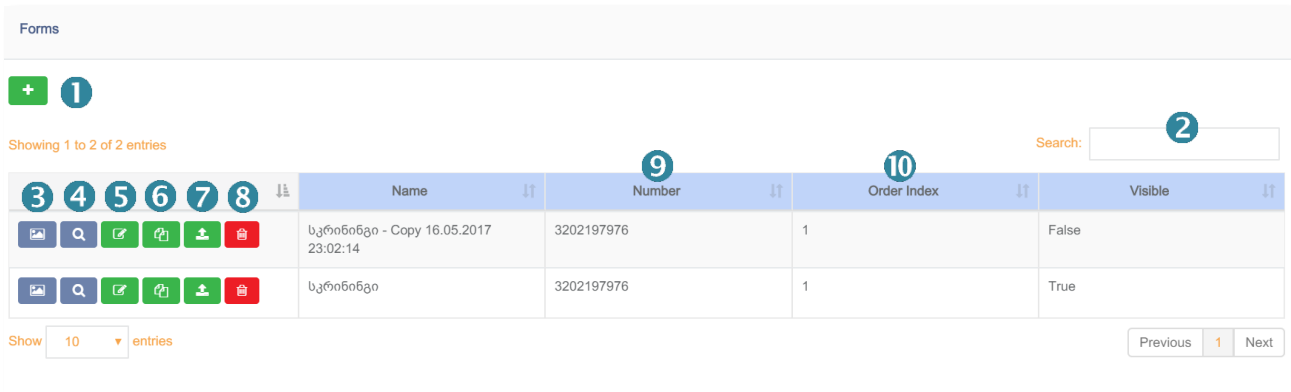
სურათი 11. კატეგორიის დამატება: (1) კატეგორიის დასახელება (2) ჩანდეს თუ არა კატეგორია ძირითად მენიუში (3) ძირითად მენიუში კატეგორიის რიგითობის განსაზღვრა (4) შენახვა (5) დახურვა (შენახვის გარეშე)

შენახვის შემდეგ, შექმნილი კატეგორია ავტომატურად აისახება მთავარ მენიუში, იმ შემთხვევაშიც კი, თუ ჯერ მასში დამატებული და გაერთიანებული არ არის ფორმები.

8.2 ფორმები

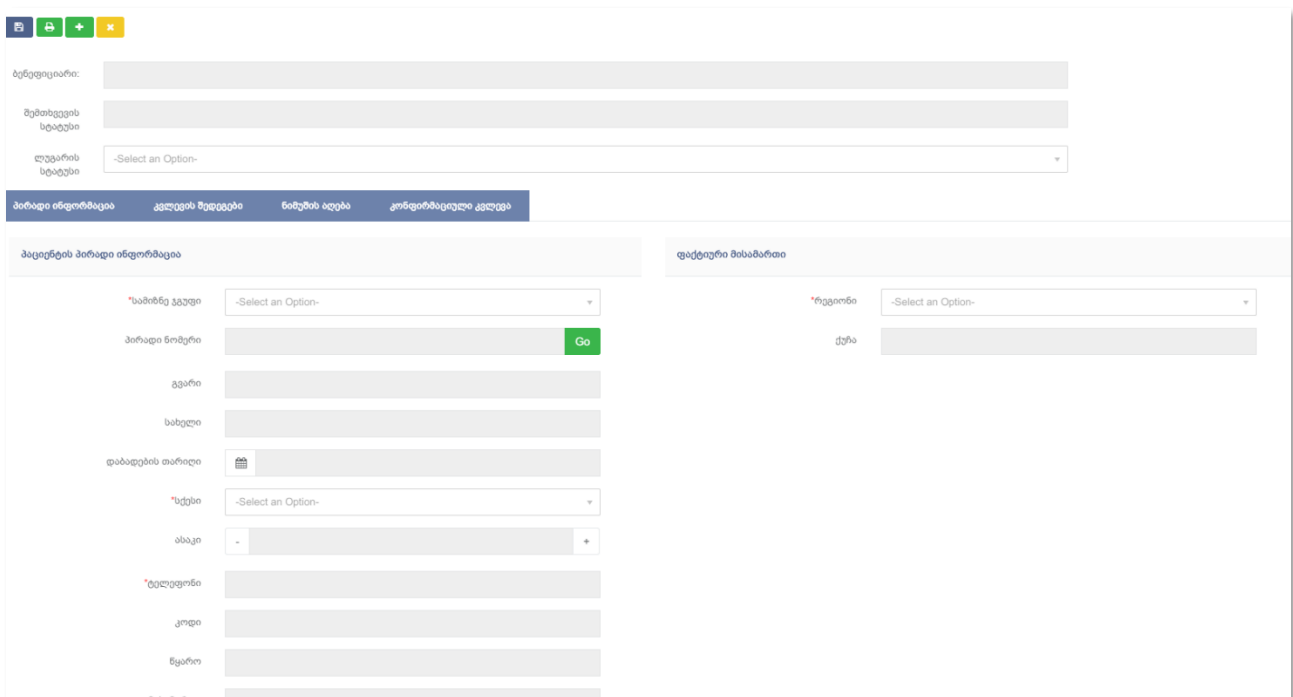
ფორმების გვერდზე გადასვლისას ინტერფეისზე ჩანს პორტალზე შექმნილი არსებული ფორმების სია (სურათი 14).

სიიდან შესაძლებელი მზა ფორმების ნახვა დასრულებული სახით, როგორც ჩანს პორტალზე მომხმარებლისთვის. მაგალითად, **სურათი 13** წარმოადგენს სკრინინგის ფორმის მზა ვერსიას.



Name	Number	Order Index	Visible
სკრინინგი - Copy 16.05.2017 23:02:14	3202197976	1	False
სკრინინგი	3202197976	1	True

სურათი 14. ფორმების სია: (1) ახალი ფორმის დამატება (2) ფორმის ძებნა (3) მზა ფორმის ნახვა (4) ფორმის კონფიგურაციის ნახვა (5) რედაქტირება (6) ფორმის დაკოპირება (დუბლირება) (7) ფაილების ატვირთვა ფორმაზე (8) ფორმის წაშლა (9) უნიკალური ნომერი (10) ფორმის რიგითობა სიაში



სურათი 13. ფორმის მზა ვერსია

8.2.1 ახალი ფორმის დამატება

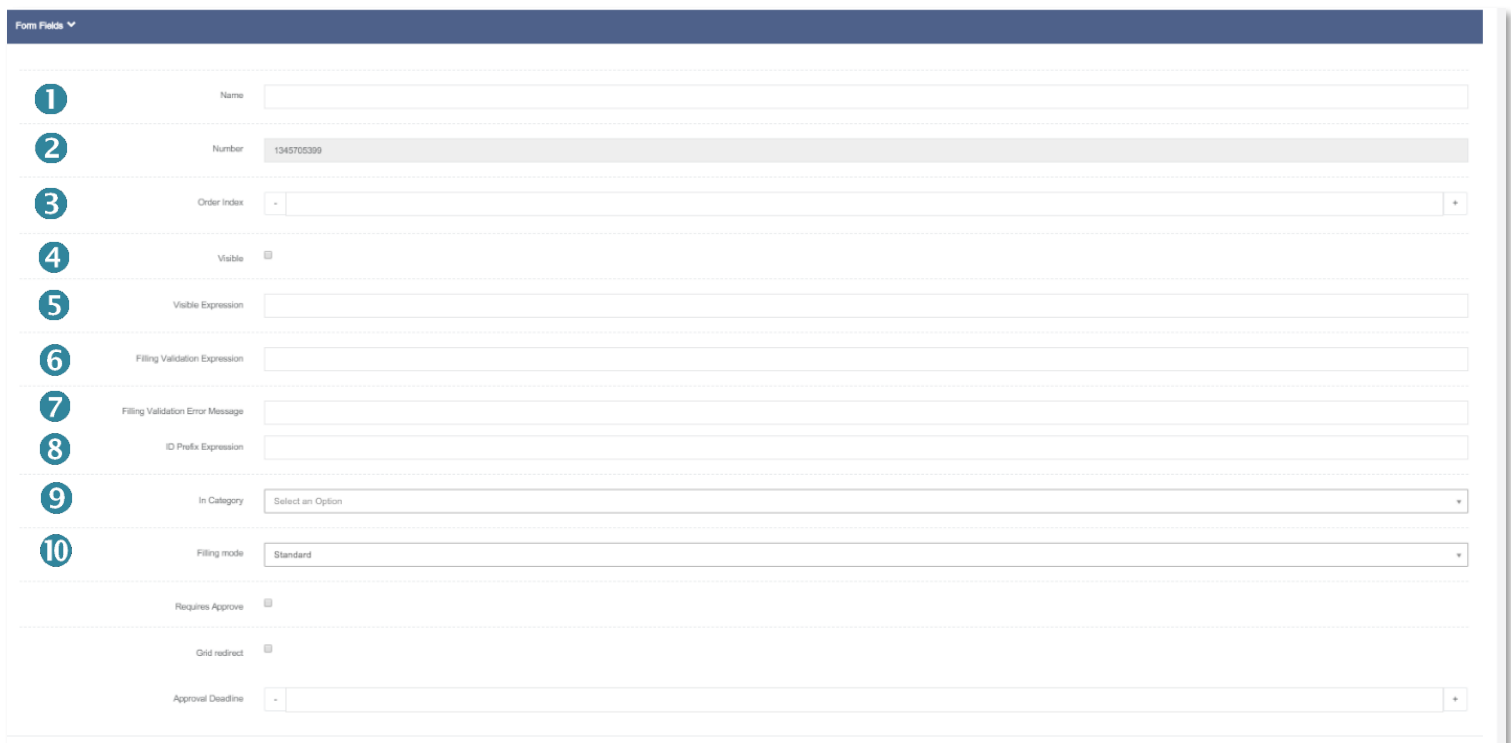
ფორმის კონფიგურაციის ეს ტაბი მოიცავს ფორმის ბაზისურ ატრიბუტებს (სურათი 15):

1. დასახელება: ფორმის სახელი
2. ნომერი: უნიკალური ნომერი (ენიჭება ავტომატურად)
3. ინდექსი: ფორმის რიგითობა კატეგორიის სიაში
4. ხილვადობა: ფორმის ხილვადობა მთავარი მენიუს (კი/არა)

Visible Expression: ამ ველში იყვება ის პირობა, რომლის შესრულების შემთხვევაშიც უნდა ჩანდეს ფორმა. პირობის სინტაქსი წარმოდგენილია **თავი 12**-ში (Appendix A: ვალიდაციების სინტაქსი).

5. Filling Validation Expression: ამ ველში იყვება ის პირობა, რომლის მიხედვითაც უნდა შეივსოს ფორმა.
6. Filling Validation Error Message: ეს ველი წარმოადგენს შეცდომის შეტყობინების ტექსტს იმ შემთხვევაში, თუ ფორმის შევსების პირობა არ შესრულდა
7. ID Prefix Expression: ID ნომრის პრეფიქსის განმსაზღვრელი

8. კატეგორიაში გაერთიანება: კატეგორია, რომელშიც უნდა გაერთიანდეს ფორმა (dropdown მენიუ, მხოლოდ იმ კატეგორიების სიით, რომელიც უკვე არსებობს პორტალზე)
9. შევსების ტიპი
 1. Standard: სტანდარტული ფორმა
 2. Single Records: ამ ოპციის არჩევის შემთხვევაში ფორმის შევსება შესაძლებელი იქნება მხოლოდ ერთხელ. ანუ, ფორმის საბოლოო ვერსია არ წარმოადგენს ჩანაწერების სიას/ცხრილს, არამედ მხოლოდ შესავსებ ფორმას (მაგალითად, მეცნიერი პროფილი). თუმცა, ადმინისტრატორის ინტერფეისიდან ეს ფორმა მაინც ცხრილი სახით გამოჩნდება
10. Requires Approve: განსაზღვრავს იმას, უნდა დადასტუროს ორგანიზაციამ ამ ფორმის შევსება, თუ არა (კი/არა)
11. Approved Deadline: დადასტურების ვადა



სურათი 15. ფორმის ბაზისური ატრიბუტები

8.2.2 ფორმის ელემენტების დამატება (Elements)

ამ ტაბიდან შესაძლებელია ფორმის ელემენტების ფორმაზე დამატება. ელემენტები შეიძლება იყოს ხუთი ტიპის:

1. **Field:** ფორმის ველი
2. **Grid:** ველების ცხრილი
3. **Tree:** ველების იერარქიული ცხრილი
4. **Group:** ნებისმიერი ელემენტის (TabContainer-ის გარდა) დამატებული. ჯგუფში შეიძლება დაჯგუფდეს Group-ის ტიპის ელემენტებიც
5. **TabContainer:** „ტაბი“ ფორმაზე

ფორმის ელემენტის დამატებისას თავიდანვე უნდა შეირჩეს ელემენტის ტიპი, რის შედეგადაც დინამიურად (ავტომატურად) იცვლება ელემენტების ატრიბუტების ინტერფეისი, ელემენტის ტიპის შესაბამისად (სურათი 16). თუმცა, ზოგიერთი ატრიბუტი საერთოა ყველა ტიპისთვის:

1. **ელემენტის ტიპი** (იხ. ზემოთ)
2. **დასახელება:** ელემენტის სრული დასახელება. ამ დასახელებით ელემენტი შეიძლება გამოყენებულ იქნას ვალიდაციის სინტაქსში

3. **Description** - ელემენტის აღწერა
4. **Alias:** ელემენტის ალტერნატიული დასახელება, იმ შემთხვევისთვის, რომდესაც ელემენტის დასახელება შეიცავს „აკრძალულ“ სიმბოლოს (მაგ.: „/“)
5. **ინდექსი:** ელემენტის რიგითობა ფორმაზე
6. **Visible Expression** -
7. **Properties:** ელემენტის დამატებითი ატრიბუტები, რომელთა ამორჩევა შესაძლებელია dropdown სიიდან და განსაზღვრავს რომელი ატრიბუტი შეიძლება იყოს ელემენტისთვის რედაქტირებადი:
 - ხილვადობა
 - კონფიდენციალურობა
 - Unique (უნიკალურობა)
 - ReadOnly (რედაქტირების აკრძალვა)
 - აუცილებელი
 - ინვერსია
 - FilterByUser (მომხმარებლის მიერ ფილტრაცია)
 - NotPrintable (ბეჭდვის აკრძალვა)
 - FirstTimeFill (მხოლოდ პირველად შევსება)
 - DisplayOnFilter (ფილტრში გამოჩენა)
 - RequiresApproval (ორგანიზაციის მიერ დასტურის საჭიროება)
8. **Dependent Field**
 - ფორმა ან ელემენტი, რომელზეც დამოკიდებულია მიმდინარე ელემენტის მნიშვნელობა
9. **Disable Expression** -
10. **Visible Expression** -
11. **ჩანდეს გარე ცხრილზე** (ჩანდეს თუ არა სვეტის სახით გარე ცხრილში)
 - სტატუსის გარეშე
 - Always (ყოველთვის ჩანდეს)
 - Conditional (ჩანდეს, მხოლოდ მაშინ, როცა ველში მნიშვნელობა შევსებულია)

Element

1

Element Type

Select an Option ▼

2

Name

3

Description

4

Alias

5

Order Index

- 10 +

6

Visible Expression

7

Properties

Visible ✕

8

Dependent field

Select an Option ▼

9

Disable Expression

10

Visible Expression

Display On Grid

Select an Option ▼

📄

✕

სურათი 16. ელემენტის დამატების ზოგადი ინერფეისი

8.2.3 Expressions

Expressions მოიცავს იმ პირობებს, რომლებიც გათვალისწინებულ უნდა იქნას ამ ელემენტისთვის და ჩანაწერის სინტაქსი ვალიდაციების სინტაქსის იდენტურია.

- **Visible Expression** – ამ ველში იწერება ის პირობა, რომლის შესრულების შემთხვევაში ველი გახდება ხილვადი.
- **Field Value Expression** - ეს ველი ივსება Default (ნაგულისხმევი) მნიშვნელობის გამოსათვლელად.
- **Validation Expression** - ამ ველში იწერება ის პირობა, რომლის შესრულების შემთხვევაში პორტალმა მომხმარებელს უნდა გამოუტანოს შეცდომა
- **Error Message** - ტექსტი, რომელიც უნდა დაინახოს მომხმარებელმა ზედა პირობის შესრულების შემთხვევაში
- **Edit Visible Expression** - ამ ველში იწერება ის პირობა, რომლის შესრულების შემთხვევაში შესაძლებელი ხდება ელემენტის რედაქტირება ან რედაქტირების შეზღუდვა.
- **Disable Expression** - ამ ველში განერილი პირობის შესრულების შემთხვევაში აღნიშნული ელემენტი პასიურდება.

8.2.4 Dependency

Dependency შეიცავს იმ ფორმებს და ელემენტებს, რომელზეც დამოკიდებულია მიმდინარე ელემენტის მნიშვნელობა ან ქმედება:

- **Dependent Field**
 - ფორმა ან ელემენტი, რომელზეც დამოკიდებულია მიმდინარე ელემენტის მნიშვნელობა
- **Dependent expression**
 - ის პირობა, რომლის შესრულების შედეგადაც უნდა გამოჩნდეს ფორმაზე მიმდინარე ელემენტი
- **Dependent fill expression**
 - ის გამოსახულება, რომლის შესრულების შედეგადაც არა მხოლოდ უნდა გამოჩნდეს ფორმაზე მიმდინარე ელემენტი, არამედ შეივსოს გამოსახულების შესაბამისად

თუ არჩეულია მხოლოდ Dependent ელემენტი, expression-ების მითითების გარეშე, მაშინ მიმდინარე ელემენტი ფორმაში გამოჩნდება მხოლოდ მაშინ, თუ Dependent ელემენტი ცარიელი არ არის.

8.2.5 Field - ველის ელემენტი

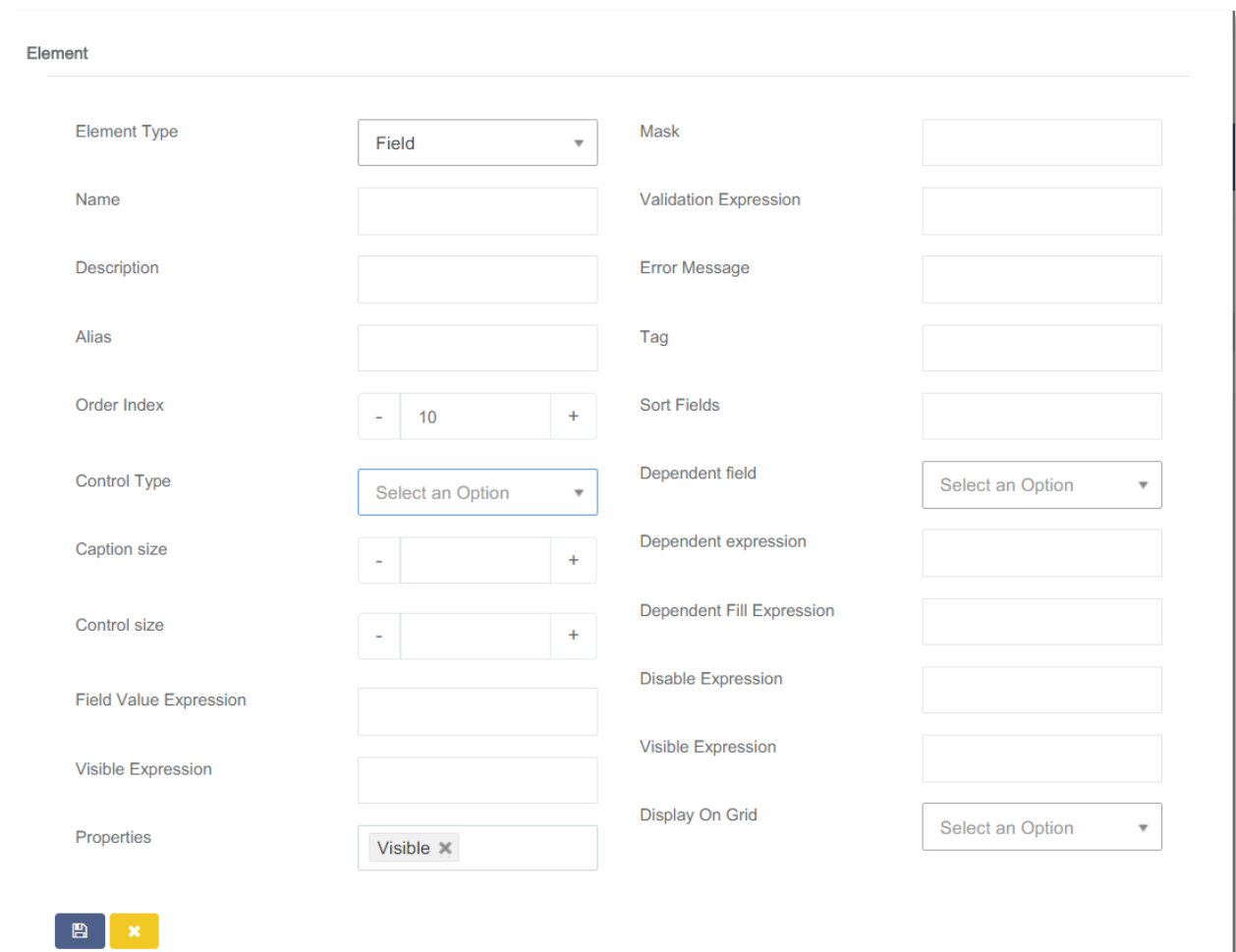
Field წარმოადგენს ფორმის ველის ელემენტს, რომელიც თავის მხრივ შეიძლება იყოს რამოდენიმე ტიპის (კონტროლის ტიპი):

- | | |
|------------|-----------------------|
| • TextBox | ტექსტური ველი (მოკლე) |
| • TextArea | ტექსტური ველი |
| • Lookup | სია |
| • Label | იარლიყი |
| • Date | თარიღი |
| • Time | დრო |
| • ნომერი | ნომერი |
| • CheckBox | მოსანიშნი |
| • ComboBox | ამოსარჩევი სია |

- CheckBoxList მოსანიშნების სია
- RadioButton მოსანიშნი (შეზღუდული სიით)
- FileUpload ფაილის ატვირთვა

ელემენტებისთვის საერთო ატრიბუტების გარდა ველის ელემენტს გააჩნია დამატებით შემდეგი ატრიბუტები (სურათი 17):

- დასახელების ზომა
 - ელემენტის დასახელების ზომა
- კონტროლის ზომა
 - კონტროლი აერთიანებს ელემენტის დასახელებასა და ელემენტის ველს.
- მასკა
 - შაბლონი მნიშვნელობისთვის (მაგ.: 9999999999 ადნიშნავს, რომ ველში უნდა ჩაიწეროს 11-ნიშნა რიცხვი)
- ტეგი
 - არ გამოიყენება



The screenshot shows a web-based configuration interface for an 'Element'. The 'Element Type' is set to 'Field'. The form contains various input fields and dropdown menus for configuring the element's appearance and behavior. At the bottom left, there are two buttons: a blue one with a document icon and a yellow one with an 'x' icon.

სურათი 17. ველის (Field) ელემენტის ატრიბუტები

8.2.6 Grid

Grid (მატრიცა) წარმოადგენს ცხრილის ელემენტს, რომელიც უნდა შეიცავს უკვე არსებული Field ელემენტებით, რომლებიც ცხრილში წარმოადგენენ სვეტებს, ხოლო სტრიქონები - მათ მნიშვნელობებს (სურათი 20).

Element

Element Type

Grid

Validation Expression

Name

Error Message

Description

Dependent field

Select an Option

Alias

Include In Filter

☐

Order Index

- 10 +

Disable Expression

Delete Visible Expression

Visible Expression

Edit Visible Expression

Display On Grid

Select an Option

Properties

Visible

სურათი 18. ცხრილის (Grid, მატრიცა) ელემენტის ატრიბუტები

8.2.7 Tree (იერარქიული სია)

Tree (იერარქიული სია) ასევე წარმოადგენს ცხრილის ელემენტს, რომელიც უნდა შეიცავს უკვე არსებული Field ელემენტებით, მაგრამ Grid-ისგან განსხვავებით ველების ელემენტების მასში დალაგებულია იერარქიულად.

სტანდარტული ატრიბუტების გარდა Tree ელემენტს გააჩნია “Tree max leve” ატრიბუტი, რომლის რიცხვითი მნიშვნელობა ასახავს იერარქიის მაქსიმალურად შესაძლო დონეს (სურათი 20). მაგალითად მნიშვნელობა „3“ ნიშნავს, რომ „დედა ელემენტს“ შეიძლება ქონდეს მხოლოდ 2 დონის „შვილი ელემენტი“.

8.2.8 Group (ჯგუფი)

Group (ჯგუფი) წარმოადგენს სხვა ელემენტების დამატებული ელემენტს - Field, Grid, Tree და თვითონ Group ელემენტები შეიძლება დაჯგუფდეს ამ ელემენტის ფარგლებში (სურათი 19). Group-ში შეიძლება გაერთიანდეს მხოლოდ ერთი ტიპის ელემენტები, ანუ მაგალითად Filed და Tree ერთ ჯგუფში ვერ გაერთიანდება. თუმცა, სხვადასხვა ტიპის Field ელემენტების (კონტროლის) გაერთიანება ერთ ჯგუფში შესაძლებელია.

სტანდარტული ატრიბუტების გარდა Groups გააჩნია შემდეგი დამატებითი ატრიბუტები:

- ჯგუფის ზომა
 - მაქსიმალურად დასაშვები ელემენტების რაოდენობა ჯგუფში
- Background Color
 - ჯგუფის ფონური ფერი
- Title Color
 - ჯგუფის სახელწოდების ფერი

Element

Element Type

Tree

Name

Description

Alias

Order Index

- 10 +

Visible Expression

Properties

Visible ✕

Validation Expression

Error Message

Dependent field

Select an Option

Disable Expression

Visible Expression

Display On Grid

Select an Option

სურათი 20. იერარქიული სიის (Tree, ხე) ელემენტის ატრიბუტები

Element

Element Type

Group

Name

Description

Alias

Order Index

- 10 +

Group Size

- +

Background Color

Title Color

Visible Expression

Properties

Visible ✕

Dependent field

Select an Option

Disable Expression

Visible Expression

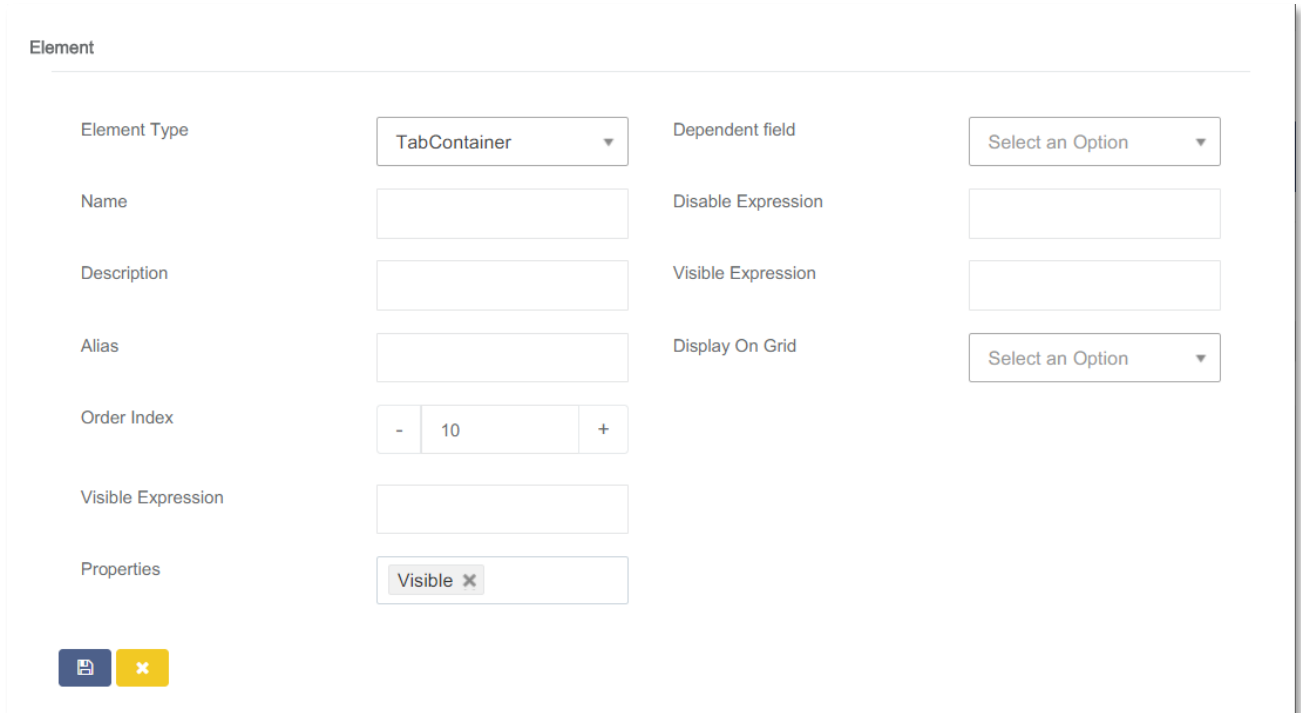
Display On Grid

Select an Option

სურათი 19. ჯგუფის ელემენტის (Group) ატრიბუტები

8.2.9 TabContainer

„ტაბი“ წარმოადგენს ელემენტების ყველაზე მსხვილ დამატგუფებებს და განლაგებულია ინტერფეისი მედა ნაწილში. Group-ისგან განსხვავებით მასში შეიძლება დატეფდეს შეუზღუდავი რაოდენობის სხვადასხვა ტიპის ელემენტი (სურათი 21).



სურათი 21. ელემენტთა გამაერთიანებელი ტაბ კონტეინერის ატრიბუტები

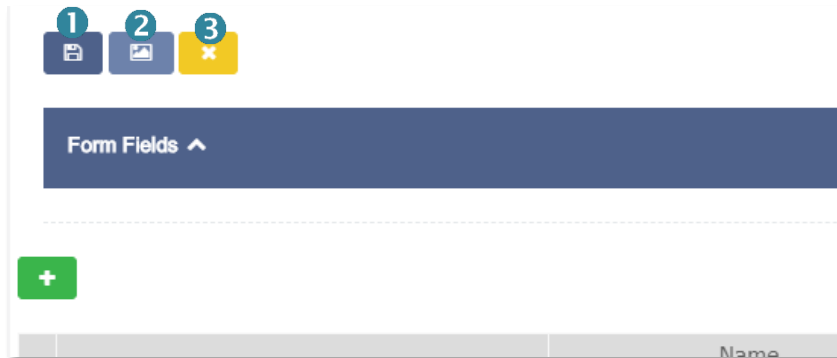
ფორმის ელემენტები სიაში დალაგებულია იერარქიულად და შესაძლებელია პირდაპირ სიიდან ელემენტების დამატება (სურათი 22).

Form Fields ^			
+ 1			
2 3 4 5 6	Name	Element Type	Control Type
    	ბუნეფიცარი:	Field	TextBox
    	შემთხვევის სტატუსი	Field	TextBox
    	ლუგარის სტატუსი	Field	ComboBox
     	სკრინინგის ფორმა	TabContainer	TabContainer
     	პირადი ინფორმაცია	TabPage	TabPage
     	პაციენტის პირადი ინფორმაცია	Group	Group
    	სამიზნე ჯგუფი	Field	ComboBox
    	სამიზნე ჯგუფი new	Field	ComboBox
    	15 წიშა კოდი	Field	TextBox
    	შენიშვნა	Field	Label
    	პირადი ზომერი	Field	PersonInfo
    	გვარი	Field	TextBox
    	სახელი	Field	TextBox

სურათი 22. ფორმის ელემენტების სია: (1) ელემენტის დამატება (2) რედაქტირება (3) წაშლა (4) კოპირების ჩასმა (5) კოპირება (6) ელემენტის დამატება (Grid, Tree და Group ტიპებისთვის)

ფორმის ადმინისტრირების ამ ტაბიდან შესაძლებელია ფორმის შენახვა შაბლონის სახით სამომავლო გამოყენებისთვის (სურათი 23).

1. ფორმის შენახვა
2. ფორმის წინასწარი ნახვა (preview)
3. დახურვა



სურათი 23. ფორმის შაბლონის შენახვა

8.3 ლოგიკები

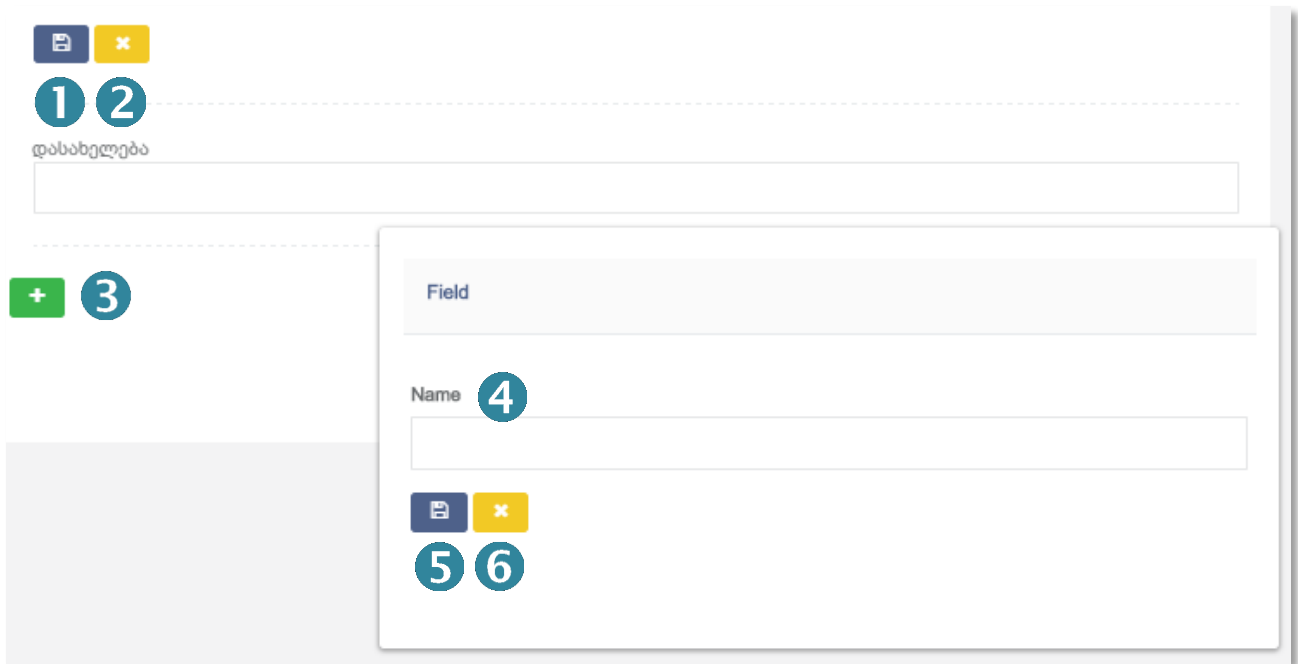
ადმინისტრირების ეს კომპონენტი გამოიყენება რეპორტირების ლოგიკების შექმნისა და ადმინისტრირებისთვის.

8.4 ცნობარები

ცნობარები გამოიყენება სისტემაში ნებისმიერი საცნობარო მონაცემების შექმნისა და ადმინისტრირებისთვის. ცნობარების ძირითადი ინტერფეისი წარმოადგენს ცნობარების სიას (სურათი 24).

Lists	
+ 1	
Showing 1 to 10 of 20 entries Search: 2	
3 4 5	Name
 	სამიზნე ჯგუფი - სკრინინგი
 	სამიზნე ჯგუფი - პირად ინფორმაციაში
 	ლუგარის სტატუსების ცნობარი
 	კონფირმაცია - ინტეგრირებული დიაგნოზი, ლაბორატორიული და კლინიკური ინტერპრეტაცია
 	კონფირმაცია - შედეგების ინტერპრეტაცია /კომენტარები
 	კონფირმაცია ნიმუშის აღების სტატუსი

სურათი 24. ცნობარების ადმინისტრირება: (1) ცნობარის დამატება (2) ცნობარის ძებნა (3) ცნობარის მონაცემები (4) ცნობარის სტრუქტურის რედაქტირება (5) ცნობარის წაშლა



სურათი 26. ახალი ცნობარის დამატება (1) ცნობარის შენახვა (2) დახურვა (3) ცნობარში ველის დამატება (4) ველის სახელი (5) ველის შენახვა (6) დახურვა

ცნობარის დამატებისთვის საჭიროა შეივსოს დასახელება და ის ველები, რომლებსაც უნდა შეიცავდეს ცნობარი (სურათი 26).

სამიზნე ჯგუფი - სკრინინგი

1 2 3 4

+ - + -

Showing 1 to 9 of 9 entries

Search:

	Name	Value	OrderIndex
  	ზოგადი პოპულაცია	6	1
  	ორსულები	7	5
  	სტუდენტები	10	10
  	წვევამდელები	12	12
  	პირები, რომლებსაც უტარდებათ ან წარსულში უტარებოდათ ჰემოდიალიზი	5	6
  	ტუბერკულოზზე საეჭვო და	11	11

სურათი 25. ცნობარში მონაცემების შეტანა: (1) ახალი ჩანაწერის დამატება (2) ჩანაწერის წაშლა (3) მონაცემების იმპორტი (4) დახურვა

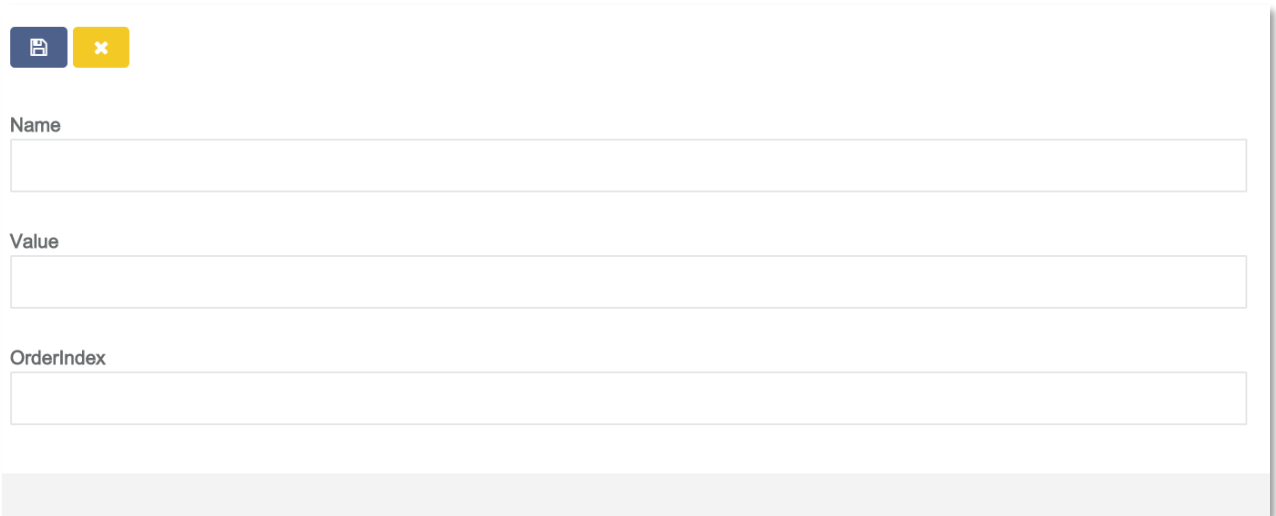
მაგალითად ქალაქების ცნობარის შემთხვევაში ასეთი ველები შეიძლება იყოს:

- ქალაქის დასახელება
- ქალაქის კოდი ცნობარში

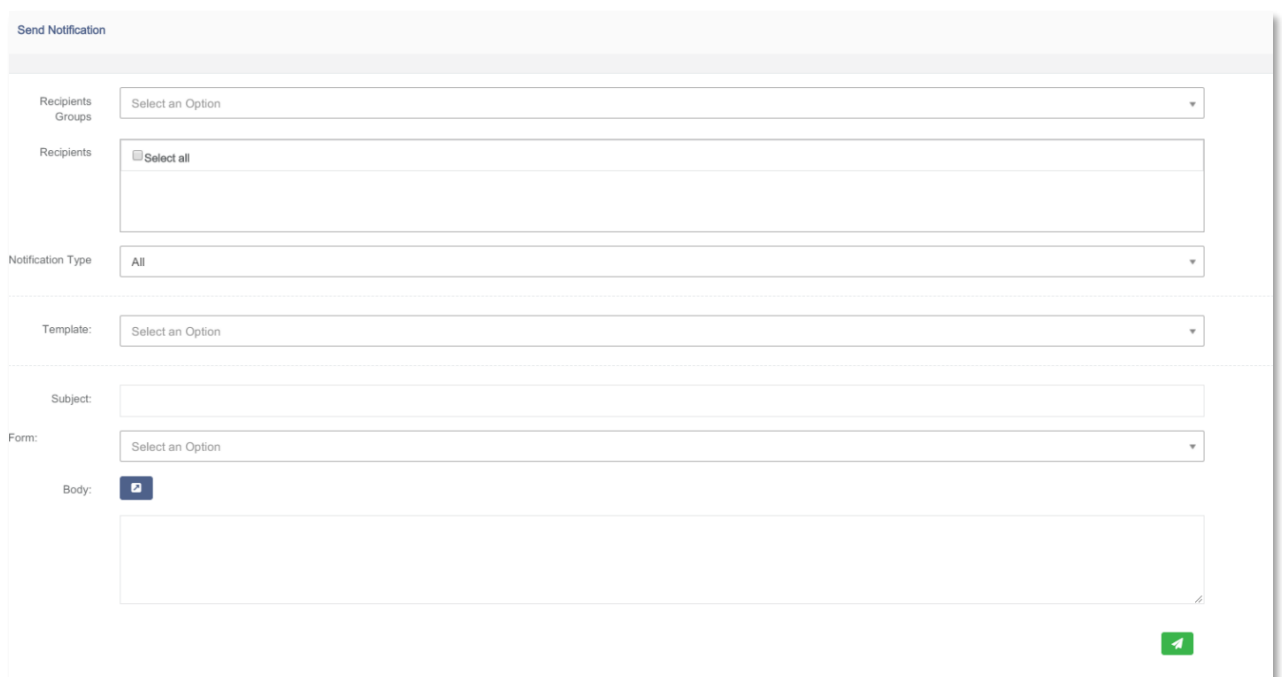
- რეგიონი
- საფოსტო ინდექსი

სტრუქტურის შექმნის შემდეგ ცნობარი უნდა შეივსოს მონაცემებით (სურათი 25, სურათი 28). მონაცემების შეტანისას გამოჩნდება ის ველები, რომლებიც ცნობარის სტრუქტურითაა განსაზღვრული.

ცნობარის მონაცემებით შევსება ასევე შესაძლებელია შესაბამისი ფორმატის ფაილის (მაგ.: .CSV, .XLS) იმპორტის საშუალებითაც. ასეთ შემთხვევაში დასაიმპორტებელ ფაილში ველები იგივე ლოგიკითა და თანმიმდევრობით უნდა იყოს განლაგებული, როგორსაც ითვალისწინებს ცნობარის სტრუქტურა.



სურათი 28. ცნობარში მონაცემების შეტანა: ახალი ჩანაწერის დამატებისთვის საჭირო ველები



სურათი 27. შეტყობინების გაგზავნის ფუნქციონალი

9 დამატებითი ბმულები

9.1 შეტყობინებები

სისტემას აქვს მომხმარებლებთან ინტერფეისის საშუალებით შეტყობინების გაგზავნის შესაძლებლობა, რომელიც მიმდინარე ეტაპზე არ გამოიყენება (სურათი 27).

9.2 კალენდარი

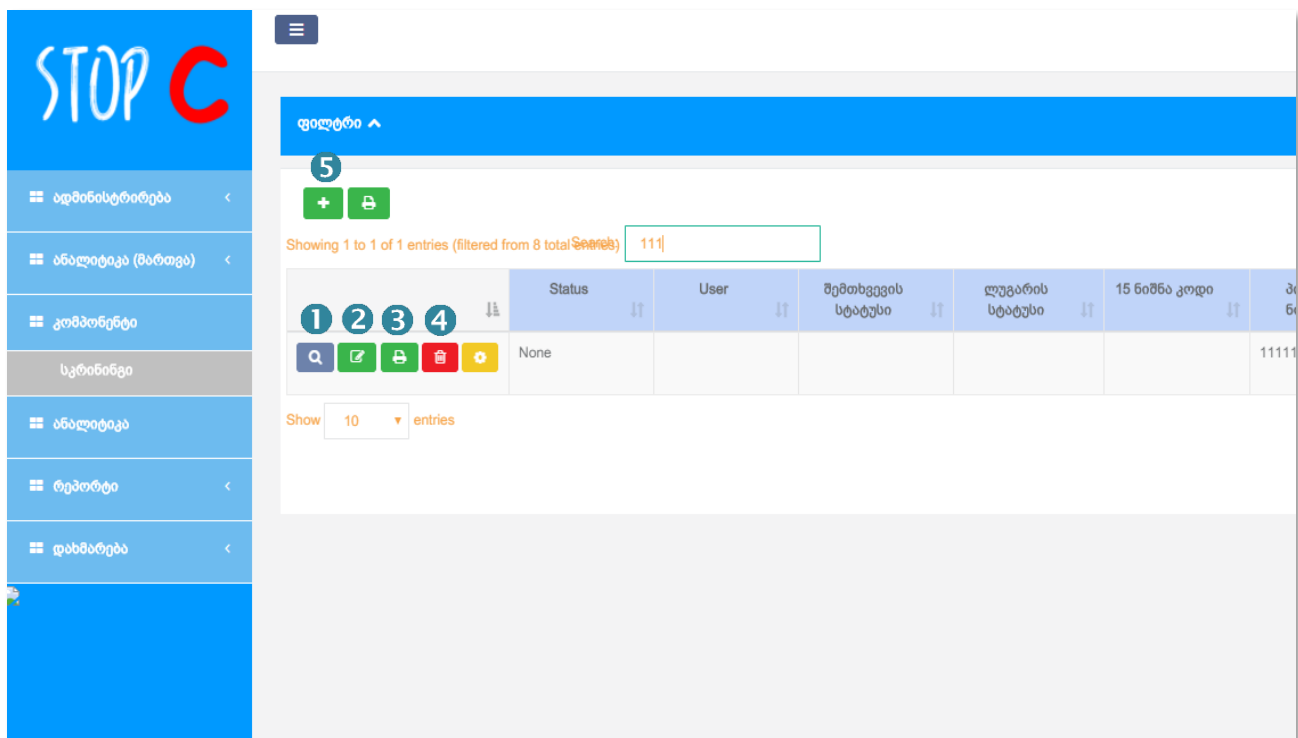
კალენდარი იძლევა იმის შესაძლებლობას, რომ შეიქმნას მოვლენები, ღონისძიებები, მათი ფაზები, ჯგუფები და კალენდარის სახით განთავსდეს სასურველ ფორმაზე.

მიმდინარე ეტაპზე სისტემის მხრიდან კალენდრის ფუნქციონალი არ გამოიყენება.

9.3 კომპონენტი - სკრინინგის ფორმა

სკრინინგის ფორმა წარმოადგენს სისტემაში მომუშავე მომხმარებლების ძირითად სამუშაო კომპონენტს. C ჰეპატიტის სკრინინგის ელექტრონული აღრიცხვის ფორმის (ელექტრონული მოდულის) წარმოების წესი განწერილია ცალკე სახელმძღვანელოებში.^{8 9}

ადმინისტრატორის მომხმარებელს გარდა გლობალური რესურსების ადმინისტრირებისა ეძლევა შესაძლებლობა სისტემაში დაარეგისტრიროს მონაცემი, აგრეთვე დაათვალიეროს, განახორციელოს რედაქტირება და წაშალოს მომხმარებლების მიერ რეგისტრირებული მონაცემები შესაბამისი ფუნქციების გამოყენებით (სურათი 29).



სურათი 29. მართვის ძირითადი ინსტრუმენტები: (1) მონაცემის დათვალიერება (2) მონაცემის რედაქტირება (3) მონაცემის ბეჭდვა (4) მონაცემის წაშლა (5) მონაცემის დამატება

⁸ სახელმძღვანელო C ჰეპატიტის მართვის სახელმწიფო პროგრამის ფარგლებში კონფირმაციული კვლევისთვის საჭირო სისხლის ნიმუშის აღებასთან დაკავშირებული მონაცემების ელექტრონული რეგისტრაციისთვის

⁹ სახელმძღვანელო C ჰეპატიტის სკრინინგთან დაკავშირებული მონაცემების ელექტრონული რეგისტრაციისთვის

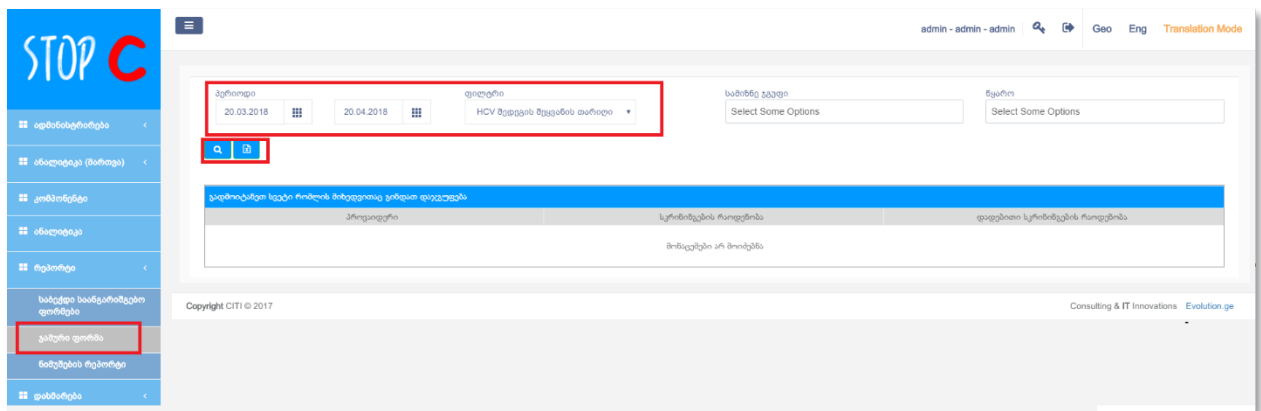
10 ანალიტიკა

მიმდინარე ეტაპზე ანალიტიკურ მოდულში გრაფიკულად ვიზუალდება ანალიტიკის მართვის მოდულიდან უკვე შექმნილი სტატიკური სტატისტიკური ცხრილები (სურათი 32).

ადმინისტრატორის მომხარებელს აქვს შესაძლებლობა შეცვალოს გრაფიკული გამოსახულების ტიპი/ფორმატი და ჩამოტვირთოს მონაცემები სხვადასხვა ფორმატში (Excel, CSV, PDF და სხვა, სურათი 33).

11 რეპორტი

ნავიგაციის პანელში რეპორტების კატეგორიაში განთავსებული „საბეჭდი საანგარიშგებო ფორმების“, „ჯამური ფორმის“ და „ნიმუშების რეპორტის“ საშუალებით ადმინისტრატორის მომხმარებელს ეძლევა შესაძლებლობა მიიღოს დეტალური ანგარიში ყველა მომხმარებლის მიერ

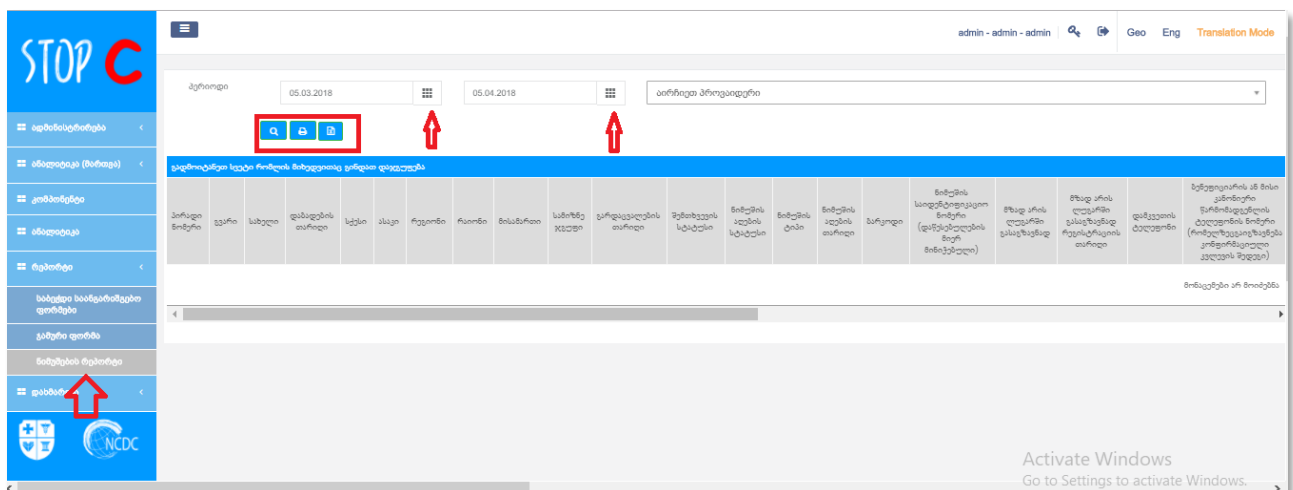


სურათი 30. HCV სკრინინგის კვლევის შედეგის რეპორტი

რეგისტრირებული სკრინინგის, ნიმუშის აღების და კონფირმაციულ კვლევის შემთხვევების შესახებ (სურათი 30).

პერიოდის ფილტრის საშუალებით მონაცემები იფილტრება სკრინინგის ტესტის ჩატარების და ნიმუშის აღების თარიღის მიხედვით (სურათი 31).

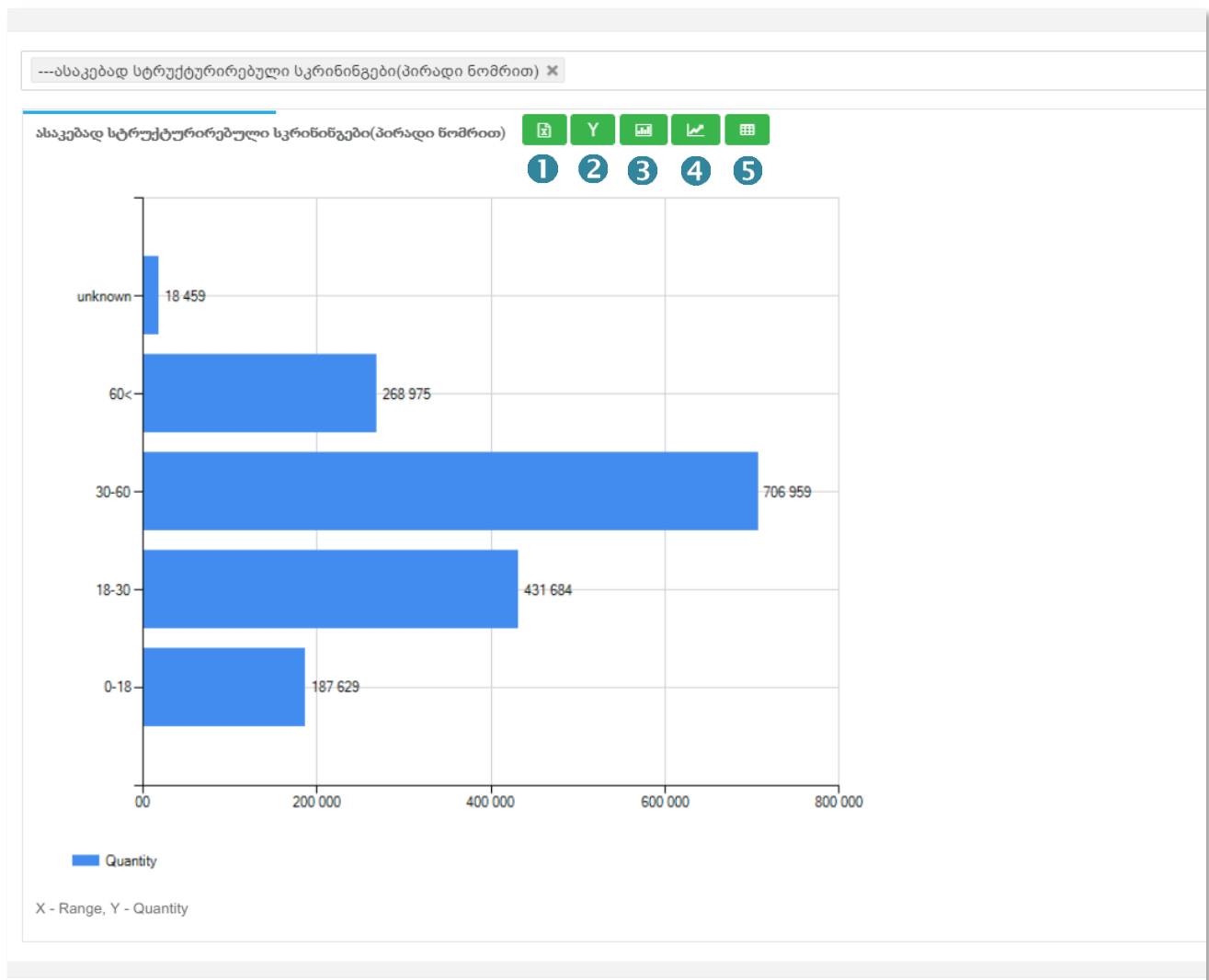
ბეჭდვის და ექსელში ექსპორტის ღილაკების საშუალებით შესაძლებელია პერიოდით გაფილტრული და ლუბით მოძიებული მონაცემების ამობეჭდვა ან ექსელის ფორმატში ექსპორტი.



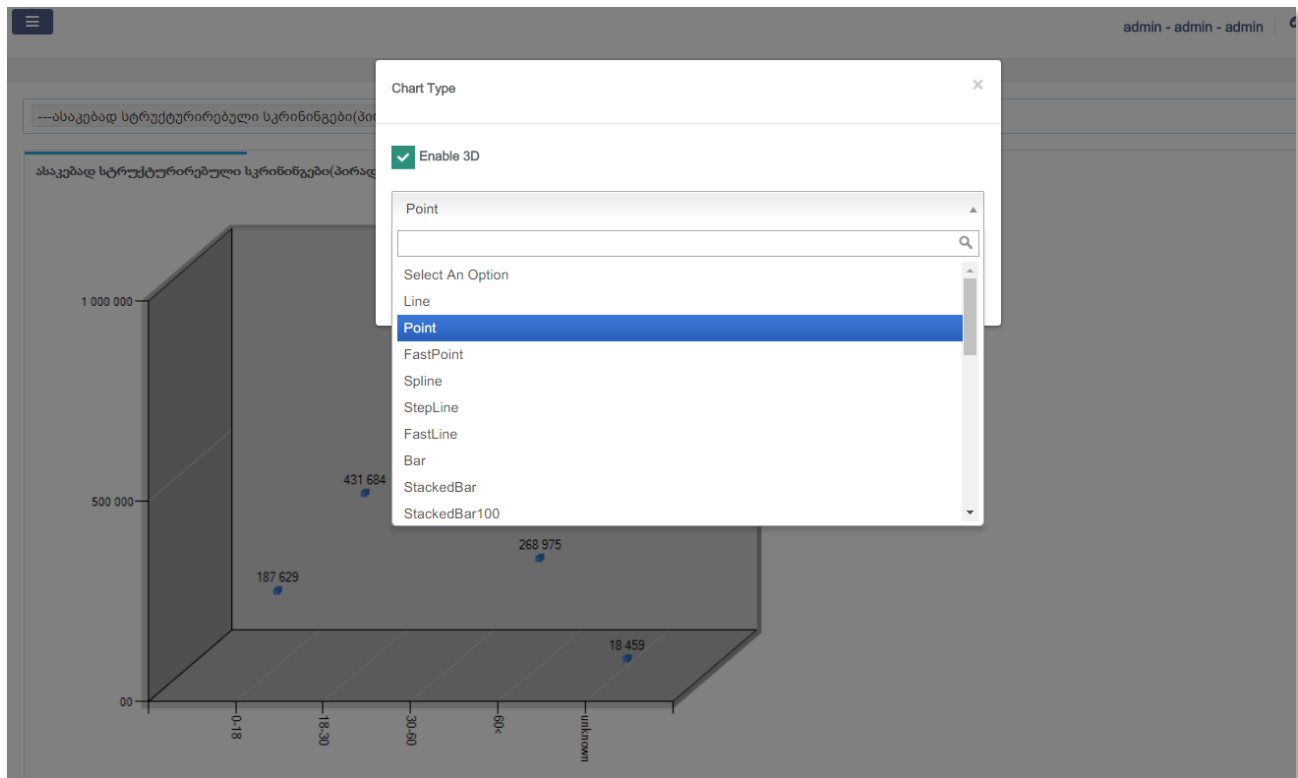
სურათი 31. ნიმუშების და კონფირმაციული კვლევის შედეგის რეპორტი

რეპორტში სისტემიდან ექსპორტირდება:

- პაციენტის დემოგრაფიული მონაცემები
- სკრინინგ ტესტებთან დაკავშირებული ინფორმაცია
- ნიმუშთან და ნიმუშის აღებასთან დაკავშირებული მონაცემები
- კონფირმაციულ კვლევასთან დაკავშირებული მონაცემები



სურათი 32. სამაგალითო სტატიკური ცხრილის გრაფიკული გამოსახულება და მართვის ინსტრუმენტები: (1) მონაცემების ჩამოტვირთვა. (2) აქსისის ღერძის მონაცემების ფილტრი (3) ჩარტის ტიპის ცვლილება (4) ცხრილის გრაფიკული გამოსახულება (5) მონაცემების ცხრილის სახით ვიზუალიზაცია



სურათი 33. სტატიკური ცხრილის ტიპის ცვლილება(მათ შორის 3 განზომილებაში)

12 Appendix A: ვალიდაციების სინტექსი

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

12.1 ოპერატორები

ოპერატორი	მნიშვნელობა	კომენტარი
!	ლოგიკური "არ"	მაგალითად არსებობს ფუნქცია IsEmpty(value) რომელიც აბრუნებს true/false (კი/არა) იმისდამხედვეთ 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	

12.2 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 ¹⁰ მნიშვნელობას მითითებულ ხარისხში
<code>floor(value)</code>	ამრგვალებს ათწილადს უახლოესი დაბალ მთელ რიცხვამდე
<code>log(value, base)</code>	ლოგარითმის ამოღება, მითითებული ფუძით
<code>log10(value)</code>	ლოგარითმის ამოღება 10-ის ფუძით
<code>pow(value, base)</code>	ახარისხება მითითებული ფუძით
<code>or(value1, value2)</code>	ბინარული "ან" ოპერაცია ¹¹
<code>and(value1, value2)</code>	ბინარული "და" ოპერაცია ¹²
<code>xor(value1, value2)</code>	გამომრიცხავი „ან“ (XOR) ¹³
<code>mod(value1, value2)</code>	მოდულიური ექსპონენტა ¹⁴ - 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)

¹⁰ [https://en.wikipedia.org/wiki/E_\(mathematical_constant\)](https://en.wikipedia.org/wiki/E_(mathematical_constant))

¹¹ https://en.wikipedia.org/wiki/Bitwise_operation#OR

¹² https://en.wikipedia.org/wiki/Bitwise_operation#AND

¹³ https://en.wikipedia.org/wiki/Exclusive_or

¹⁴ 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
regex(value, exp)	ამოწმებს ტექსტს Regex-ის ¹⁵ 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, გწერთ 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, გწერთ 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)

¹⁵ 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 ¹⁶
<code>addYears(date, count)</code>	უმატებს წლებს პარამეტრად გადაცემულ თარიღს (count რაოდენობა) ¹⁷
<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 ჩანაწერა, შეამოწმოს აკმაყოფილებს თუ არა თითოეული ჩანაწერი condition-ში გაწერილ პირობებს და

¹⁶ 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

¹⁷ 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