



Administrator's Manual

Hepatitis C Screening Module

2018

Table of Contents

1	PICTURES AND DIAGRAMS	4
2	PURPOSE OF THE DOCUMENT	5
3	SHORT DESCRIPTION OF STOP C SCREENING MODULE AND ITS FUNCTIONALITY	5
4	STOP C PROGRAM ECOSYSTEM	5
4.1	NCDC.eHealth	5
4.2	MOH.ElimC.....	6
4.3	MOH.eHealth	6
4.4	Other Related Modules	6
5	GENERAL DESCRIPTION AND PURPOSE OF STOP C MODULES	7
6	CONFIGURATION SCHEME	8
7	MAIN COMPONENTS OF INTERFACE	10
7.1	Header	10
7.2	Navigation Tool	10
8	CONFIGURATION OF FORMS	11
8.1	Categories.....	11
8.2	Forms	12
8.2.1	Add new forms	13
8.2.2	Add form elements.....	14
8.2.3	Expressions.....	15
8.2.4	Dependency	16
8.2.5	Grid.....	17
8.2.6	Tree	18
8.2.7	Group.....	18
8.2.8	Tab Container.....	20
8.3	Logics	21
8.4	Dictionary.....	21
9	ADDITIONAL LINKS.....	23
9.1	Notification	23
9.2	Calendar	23
9.3	Component - Screening.....	24
10	ANALYTICS	24



11 REPORT25

12 APPENDIX A: VALIDATION EXPRESSIONS.....27

12.1 Operations..... 28

12.2 Functions 29

1 Pictures and Diagrams

Picture 1. STOP C Modules in program architecture perspective	7
Picture 2. Diagram for Stop C Form, Data and Process	8
Picture 3. General scheme of configuration screening form.....	9
Picture 4. Main components of configuration: (1) Interface Header (2) Navigation Panel (3) Main Panel	9
Picture 5. Interface Header: (1) Changing main menu (2) Changing Password (3) Log out (4) Translation Mode.....	10
Picture 6.	10
Picture 7.	10
Picture 8.	10
Picture 9.	11
Picture 10.	11
Picture 12. List of Form Categories (1) Add (2) Search (3) View (4) Edit (5) Delete (6) Category Name (7) Order Index (8) Visible	11
Picture 11. Add Category: (1) Name (2) Visible (3) Order Index (4) Save (5) Close (without saving) ...	12
Picture 14. Form List: (1) Add new form (2) Search (3) Preview(4) View (5) Edit (6) Copy (7) Files (8) Delete (9) Number (10) Order Index.....	12
Picture 13. Ready Version of forms	13
Picture 15. Basic Attributes of the form	14
Picture 16. Interface of Adding elements	15
Picture 17. Field Attributes	17
Picture 18. Grid Attributes.....	18
Picture 19. Group Attributes.....	19
Picture 20. Tree Attributes.....	19
Picture 21. Tab Container Attributes	20
Picture 22. List of Form Elements (1) Add (2) Edit (3) Delete (4) Paste (5) Copy (6) Add Element (Grid, Tree and Group).....	20
Picture 23. Saving the Form template	21
Picture 24. Dictionary Configuration (1) Add (2) Search (3) Data (4) Edit (5) Delete	21
Picture 26. Add New Dictionary (1) Save (2) Close (3) Add Fields (4) Name (5) Save (6) Close	22
Picture 25. Add Data into the Dictionary (1) New (2) Clear (3) Import (4) Cancel	22
Picture 27. Notification Tool	23
Picture 28. Add Data into the Dictionary: 3 necessary fields for adding new data	23
Picture 29. Main Configuration Tools (1) View (2) Edit (3) Print (4) Delete (5) Add.....	24
Picture 30. Report of HCV Screening test results	25
Picture 31. Report of Sample and confirmation test results	25
Picture 32. Graphic visualization of Static table and its configuration tools: (1) Export. (2) Y Value (3) Chart type (4) Table (5) Table	26
Picture 33. Chart Type (including 3D)	27

2 Purpose of the Document

Administrator's manual is fundamental for developing and launching STOP-C Screening module. Document describes general principles for admins how to work with the system tools and give detailed instruction to work with the system configuration. Document is created for system admins and developers and combines all necessary information for system management, support and further development.

Following document does not involve any detailed information on technical structure of system components. This kind of information is provided in separate Stop C Screening Architecture Document, which is important part of system technical documentation.

3 Short Description of STOP C Screening module and its functionality

Stop C Screening module records and manages data in medical services within the Hepatitis C state program. Module ensures several functions:

- Recording Hepatitis C screening results within the screening component
- Electronic registration of data regarding sample taking and transportation for confirmation tests within the diagnostic component
 - Registration of confirmation tests results performed by Lugar laboratory
 - Synchronization of confirmation tests results performed by Lugar laboratory with Treatment database

Stop C Screening module is allocated on following web address : <http://stop-c.moh.gov.ge>

4 STOP C Program Ecosystem

Besides the STOP C Analytics, STOP C Elimination ecosystem unites different types of systems and modules, which are placed in different physical and program platforms. These modules are divided into 4 main groups:

- NCDC.eHealth
- MOH.ElimC
- MOH.eHealth
- Other modules

4.1 NCDC.eHealth

These modules involve different program tools for National Center of Disease Control and Public Health (NCDC) to manage business processes. STOP C Analytics is part of these processes. It also combines following modules for managing hepatitis C programs:

- NCDC.eHealth.Screening: Hepatitis C Screening Module
- **NCDC.eHealth.Analytics: Stop C Analytics**
- NCDC.eHealth.UM: User Management Module
- NCDC.eHealth.CM: Common Data
- NCDC.eHealth.SB: Integration Module (RPC)

- NCDC.eHealth.Tools: Tool Libraries
- NCDC.eHealth.SMS: SMS Module

4.2 MOH.ElimC

These are part of Social Service Agency modules, which are applied for managing Hepatitis C Elimination Program and consists of following components:

- MOH.ElimC.Treatment: Treatment
- MOH.ElimC.SSA: SSA Commission
- MOH.ElimC.Drugs: Drug Management
- MOH.ElimC.Monitoring: Monitoring

4.3 MOH.eHealth

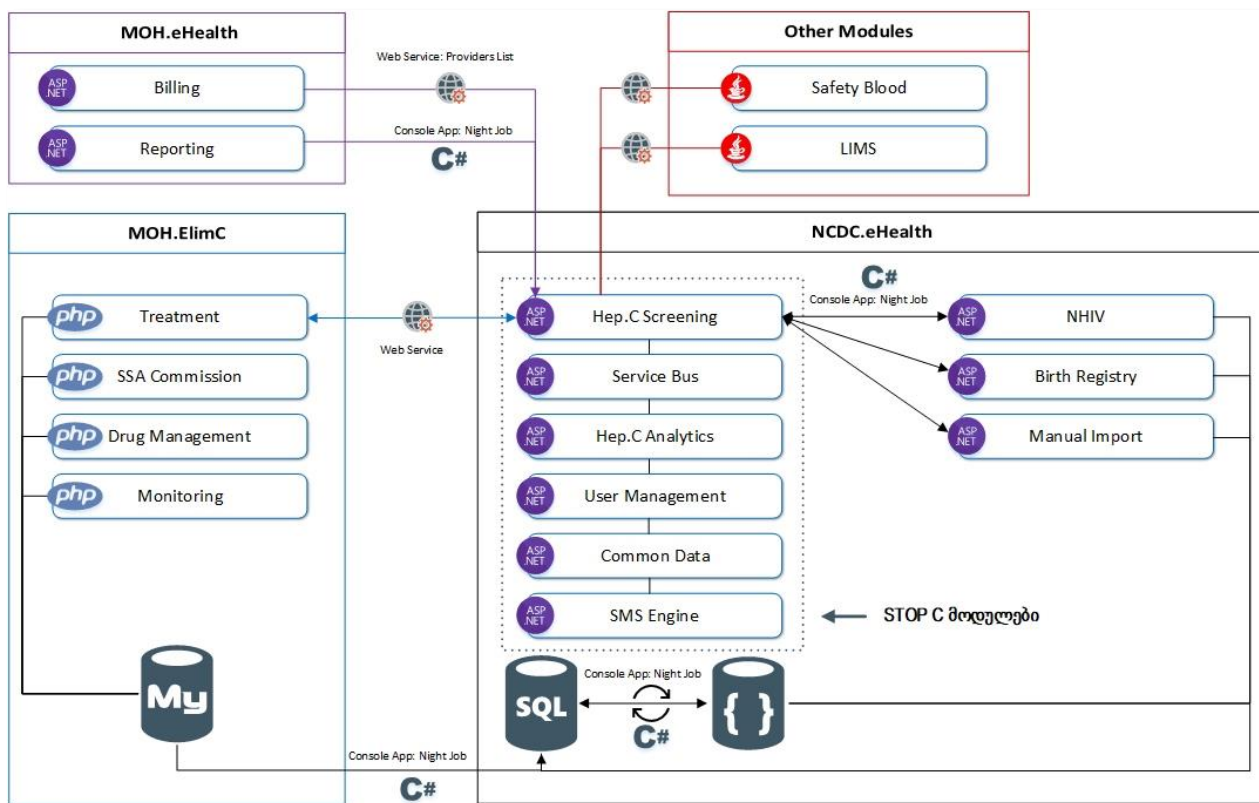
E health of Ministry of Labour, Health and Social Affairs of Georgia (MoLHSA) is complex, independent ecosystem, which is considered to automate different state processes. E health involves more than 30 modules, but data source and target for Stop C analytics, are following modules:

- MOH.eHealth.Billing: Billing Module
- MOH.eHealth.Reporting: Reporting Module

4.4 Other Related Modules

Two other external modules are also united in the ecosystem, which are involved in Hepatitis C business processes:

- SafeBlood: Safe Blood Module
- LMIS: Laboratory Management Information System



Picture 1. STOP C Modules in program architecture perspective

4.5 General Description and Purpose of STOP C Modules

STOP C Screening module unites 3 main modules and 2 additional sub modules

- Main Modules
 - User Management Module, UMM)
 - Tool for managing user roles and permits
 - Central Module/Portal ¹
 - Main module of STOP C Screening, configuration and navigation tools
 - Analytics ²
 - Dynamic reports and analytics of Stop C Screening database and its related modules
- Additional Modules
 - Common Data Module, CDM
 - Storage and process of common data
 - Multilingual and logging functions
 - Communication Modules : Service Bus, SB
 - Ensuring data exchange among different modules
 - SMS Module

¹ Following document describes working principles with main portal for administrators

² Administrator's manual for STOP C Analytics and system architecture is provided in separate document

- [Administrator's manual for STOP C Analytics](#)

- [System architecture for STOP C Analytics](#)

- Not applied now, but is planned to be implemented in future

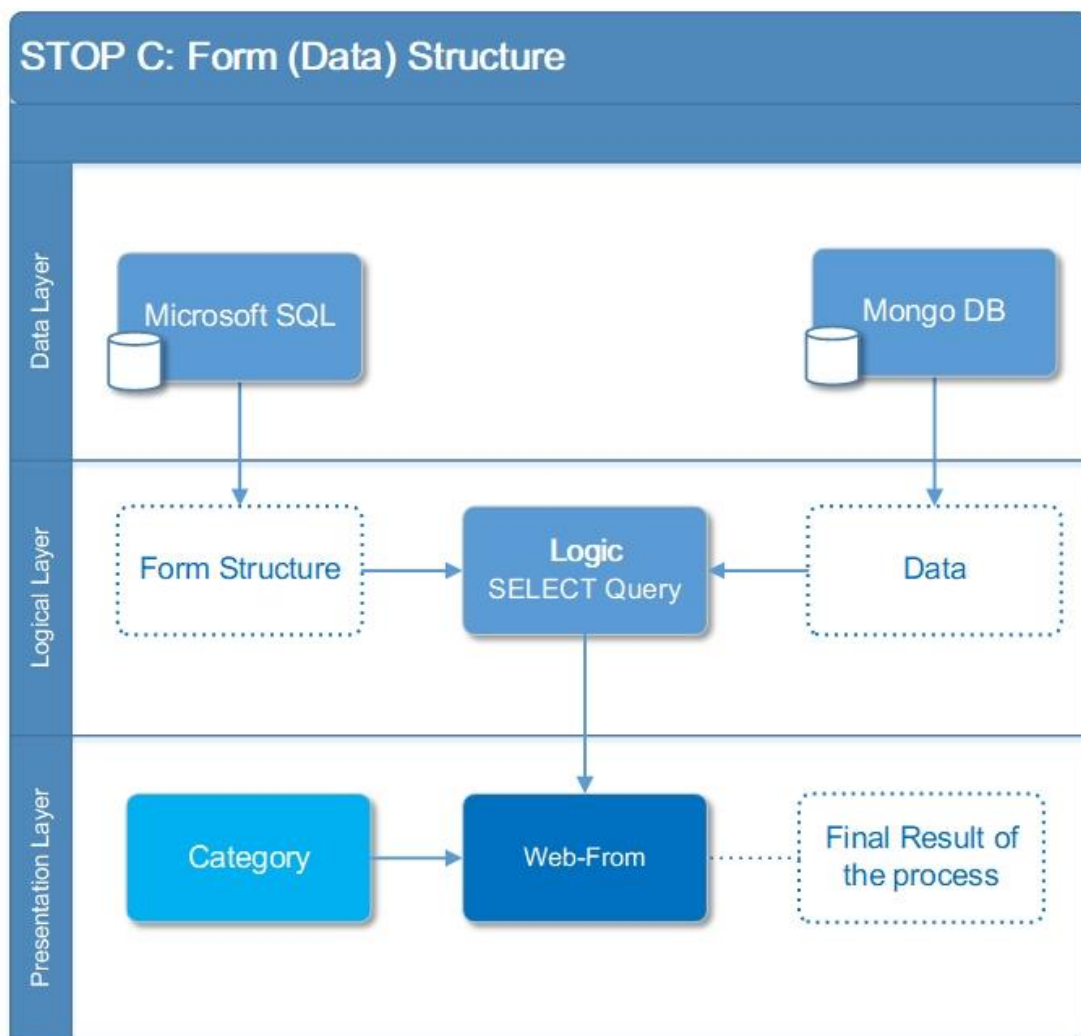
Picture 1 shows diagram of system technical approach in Hepatitis C program ecosystem

4.6 Configuration Scheme

Main elements of system configuration are Web forms, which can be created, modified and deleted from admin's interface.

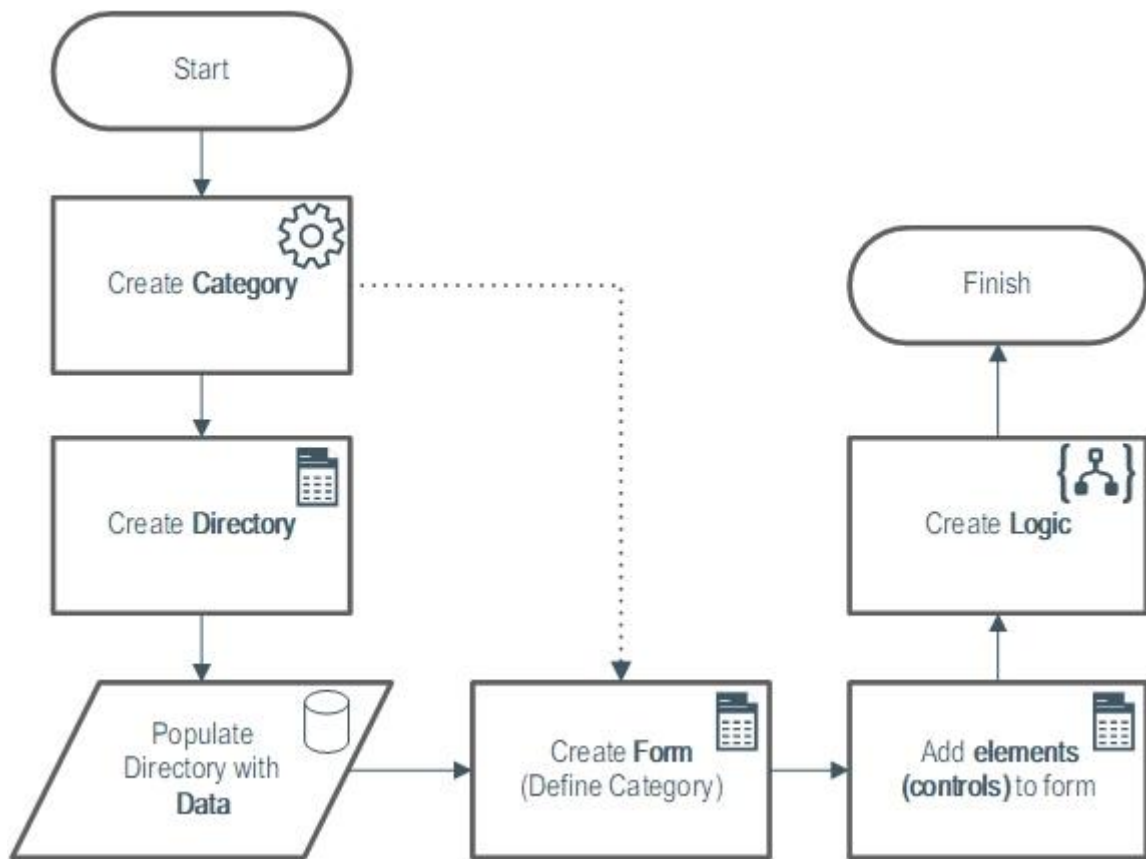
Scheme for working with web forms consists of the following steps:

1. Category should be created before creating any forms and should be defined for any form in advance.
2. Creation of form dictionaries
 - a. Filling dictionaries with data



Picture 2. Diagram for Stop C Form, Data and Process

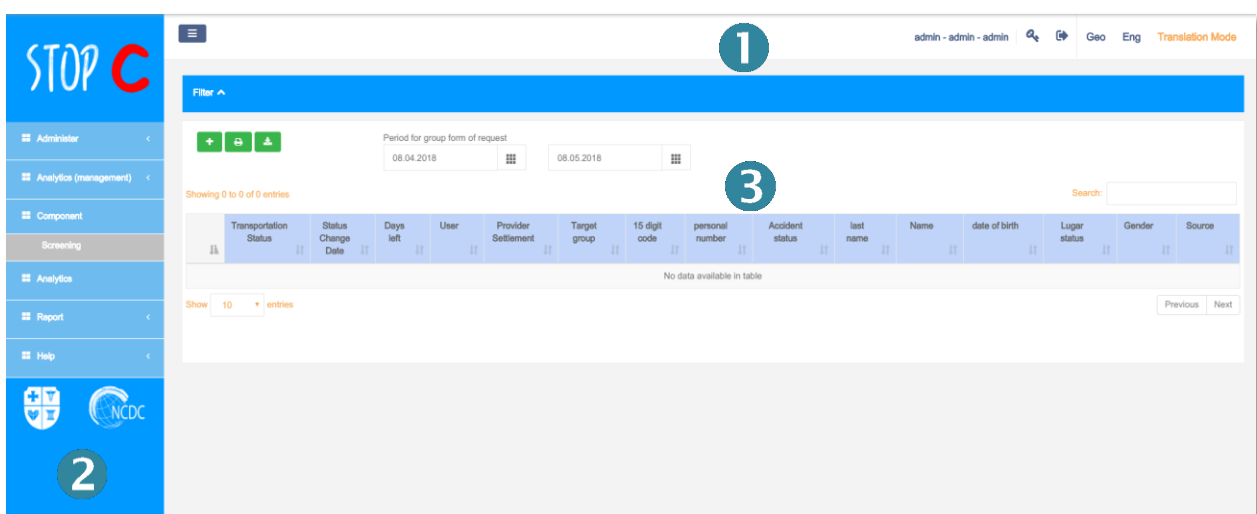
3. Creation of new forms
 - a. Define form category and select from existing list
 - b. Add form elements



Picture 3. General scheme of configuration screening form

4. Define logics of filling data in forms

Picture 3 Shows schematic diagram of the process, but **Picture 2** Displays components of the same process in different aspects (at Data, logics and Presentation level).



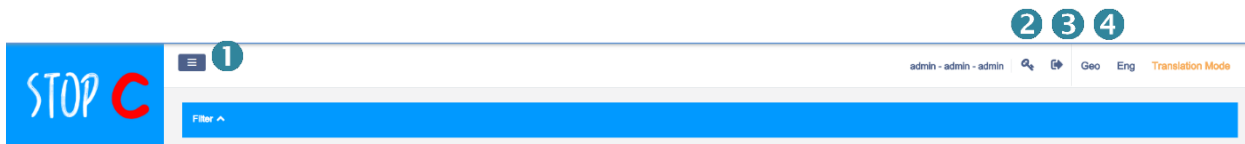
Picture 4. Main components of configuration: (1) Interface Header (2) Navigation Panel (3) Main Panel

5 Main components of Interface

Main components of configuration are displayed on main page of interface (**Picture 4**).

5.1 Header

Interface Header unites links for changing several fundamental parameters (**Picture 5**). Changing languages is available from Header as well.



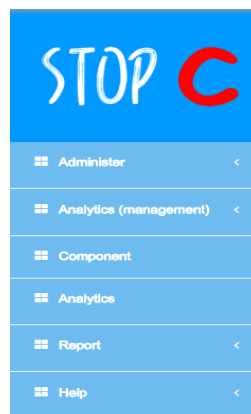
Picture 5. Interface Header: (1) Changing main menu (2) Changing Password (3) Log out (4) Translation Mode

5.2 Navigation Tool

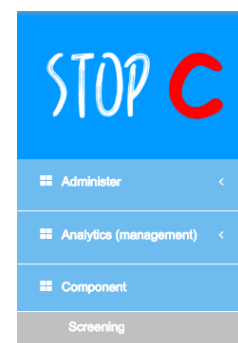
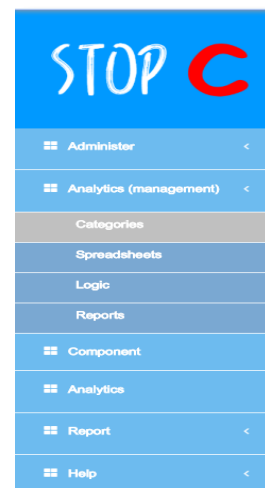
Navigation tool is group of basic links, which ensures navigation within configuration elements

Navigation tool combines following links:

1. Configuration Components (**Picture 6**):
 - a. Categories
 - b. Forms
 - c. Logics
 - d. Dictionaries
 - e. User Groups
 - f. Notification
 - g. Calendar
 - h. Users
2. Analytics/management (**Picture 7**)^{3 4}
 - a. Categories
 - b. Tables
 - c. Logics
 - d. Reports
3. Component – User's working link (**Picture 8**).
 - a. Screening
4. Analytics - Static templates and tables, created in Analytics configuration components, can be visualized in this category



Picture 7

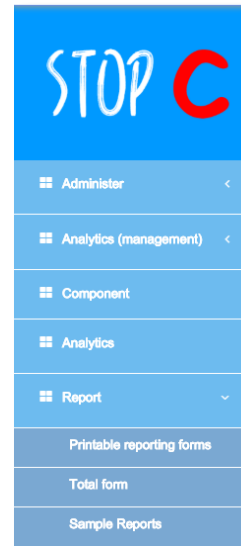
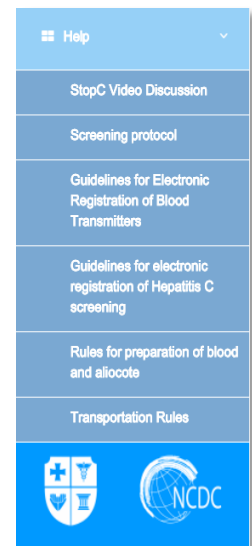


Picture 8

³ Data collected in Stop C Screening database, is processed in Stop C Analytics: <http://birthregistry.moh.gov.ge/Hmis/Hmis.Dashboards.Web>.

⁴ Administrator's Manual for Stop C Analytics is provided in Separate document.

5. Report (**Picture 10**).
 - a. Printing reporting forms
 - b. Cumulative Form
 - c. Sample Reports
6. Help - Different guidelines, protocols and video tutorials related to system business processes are allocated in this category (**Picture 9**).

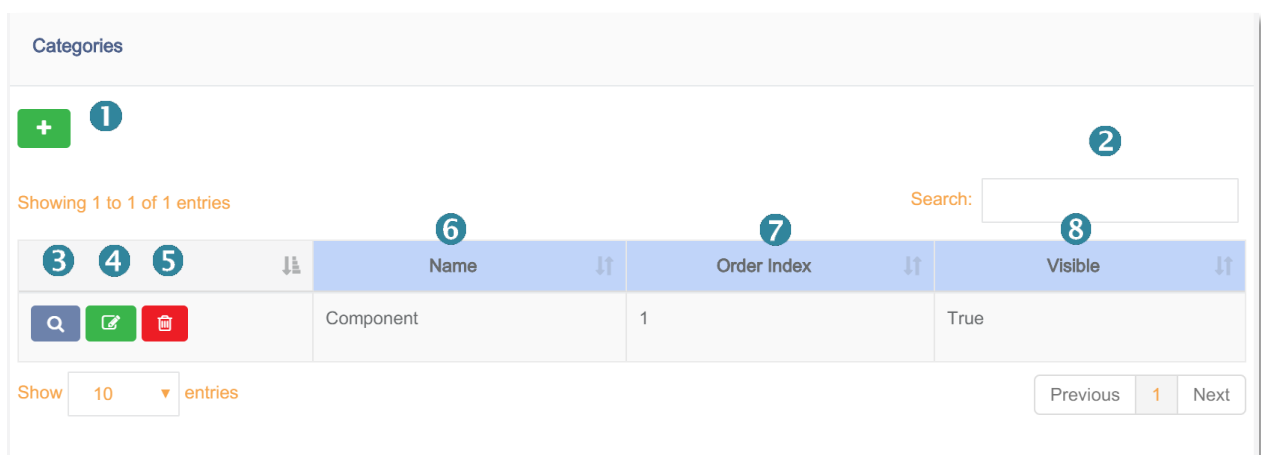

Picture 10

Picture 9

6 Configuration of forms

Forms are main element of the system, which determines dynamic administration process of the portal.

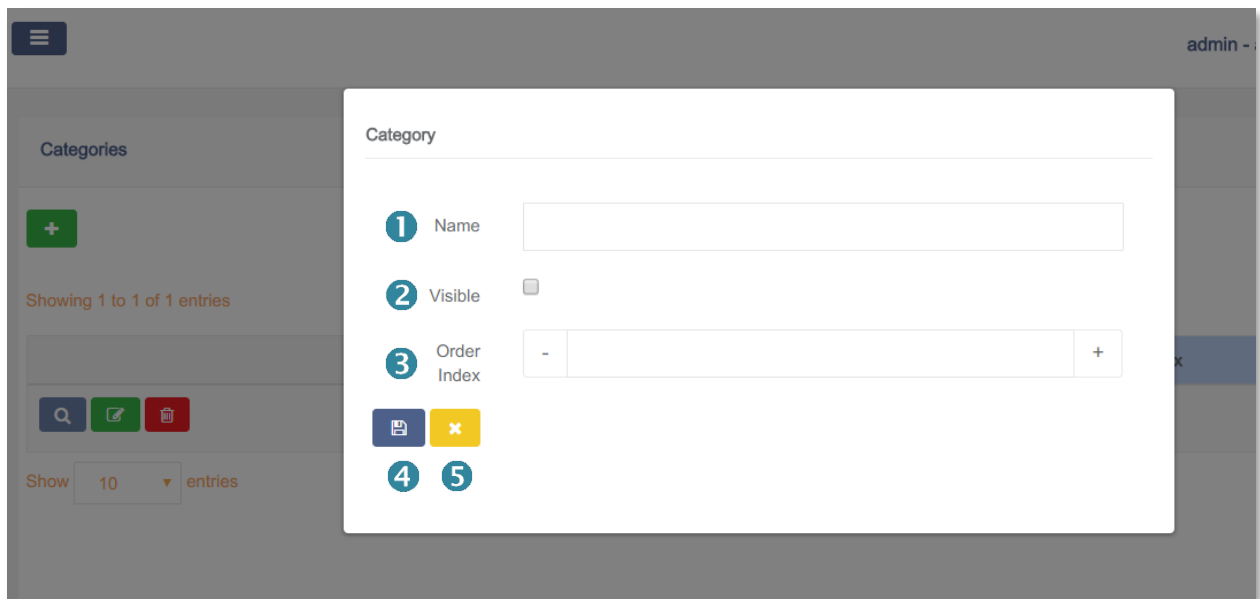
6.1 Categories

Category should be created before creating any forms, where different forms will be united. Categories are grouped in configuration. List of created categories is available in category component (**Picture 12**).



Picture 11. List of Form Categories (1) Add (2) Search (3) View (4) Edit (5) Delete (6) Category Name (7) Order index (8) Visible

Pressing the Add button will open window to add new category (**Picture 11**).



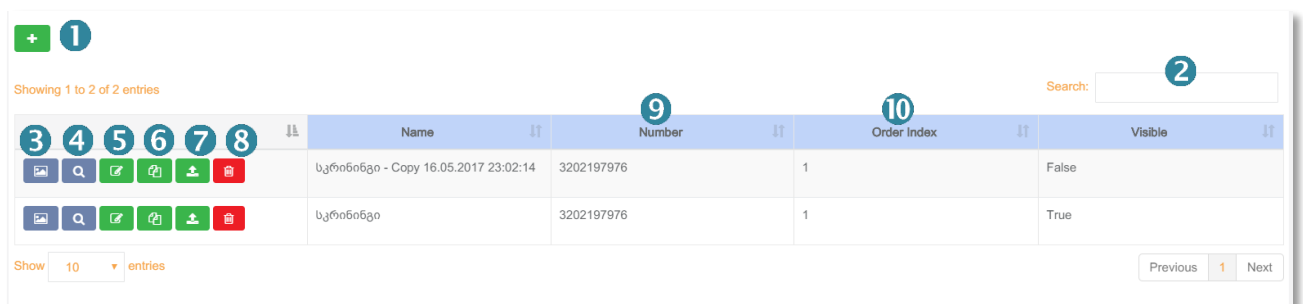
Picture 12. Add Category: (1) Name (2) Visible (3) Order Index (4) Save (5) Close (without saving)

After saving category, it will appear in main menu, even though no forms is added and grouped in it.

6.2 Forms

List of created categories is available in category component (**Picture 14**).

From the list, created forms can be displayed for users, e.g. **Picture 13** Shows ready form of Screening module.



3	4	5	6	7	8	9	10	
სკრინინგ - Copy 16.05.2017 23:02:14	3202197976	1	False					
სკრინინგ	3202197976	1	True					

Picture 13.Form List: (1) Add new form (2) Search (3) Preview(4) View (5) Edit (6) Copy (7) Files (8) Delete (9) Number (10) Order Index

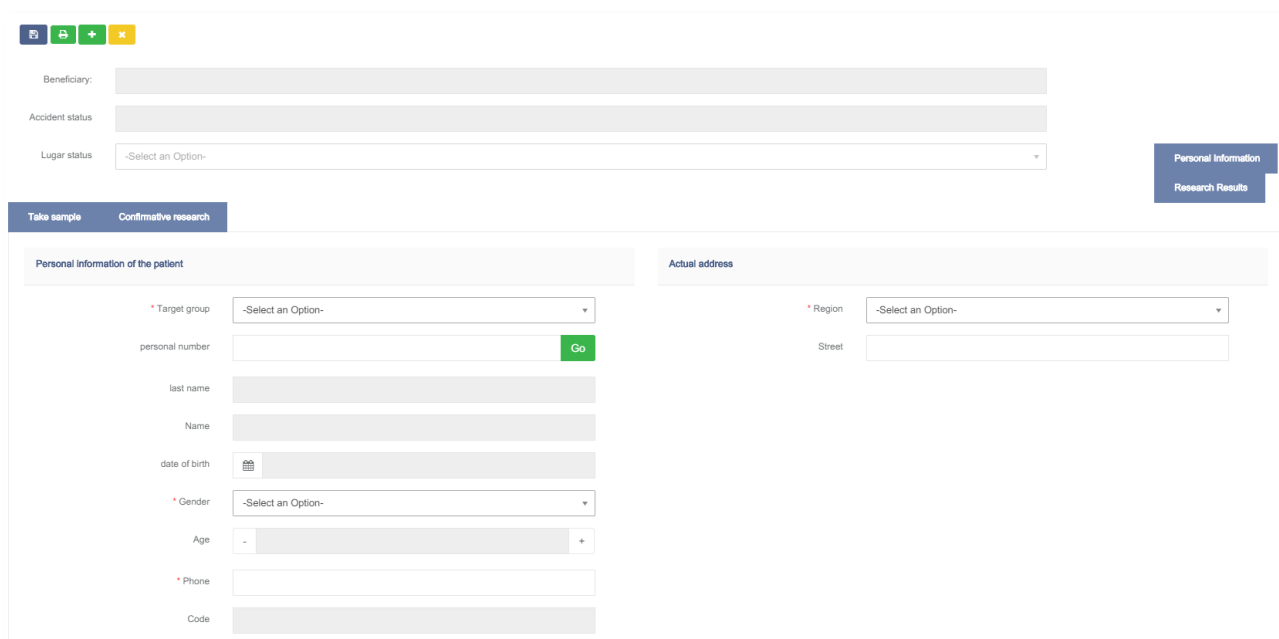
6.2.1 Add new forms

When creating the category, following features should be appointed: **(Picture 15):**

1. Name - Form Name
2. Number – Unique number, is generated automatically
3. Index – order index in the list
4. Visible – visibility in main menu (Yes/No)

Visible Expression: conditions, to manage form visibility. Expressions are provided in **Chapter 0** (Appendix A: Validation Expressions).

5. Filling Validation Expression: Expression for filling the forms
6. Filling Validation Error Message: Error message if Filled Value Expression is not realized
7. ID Prefix Expression: to define ID number prefix
8. In Category: dropdown menu of existing category lists, where the forms should be grouped



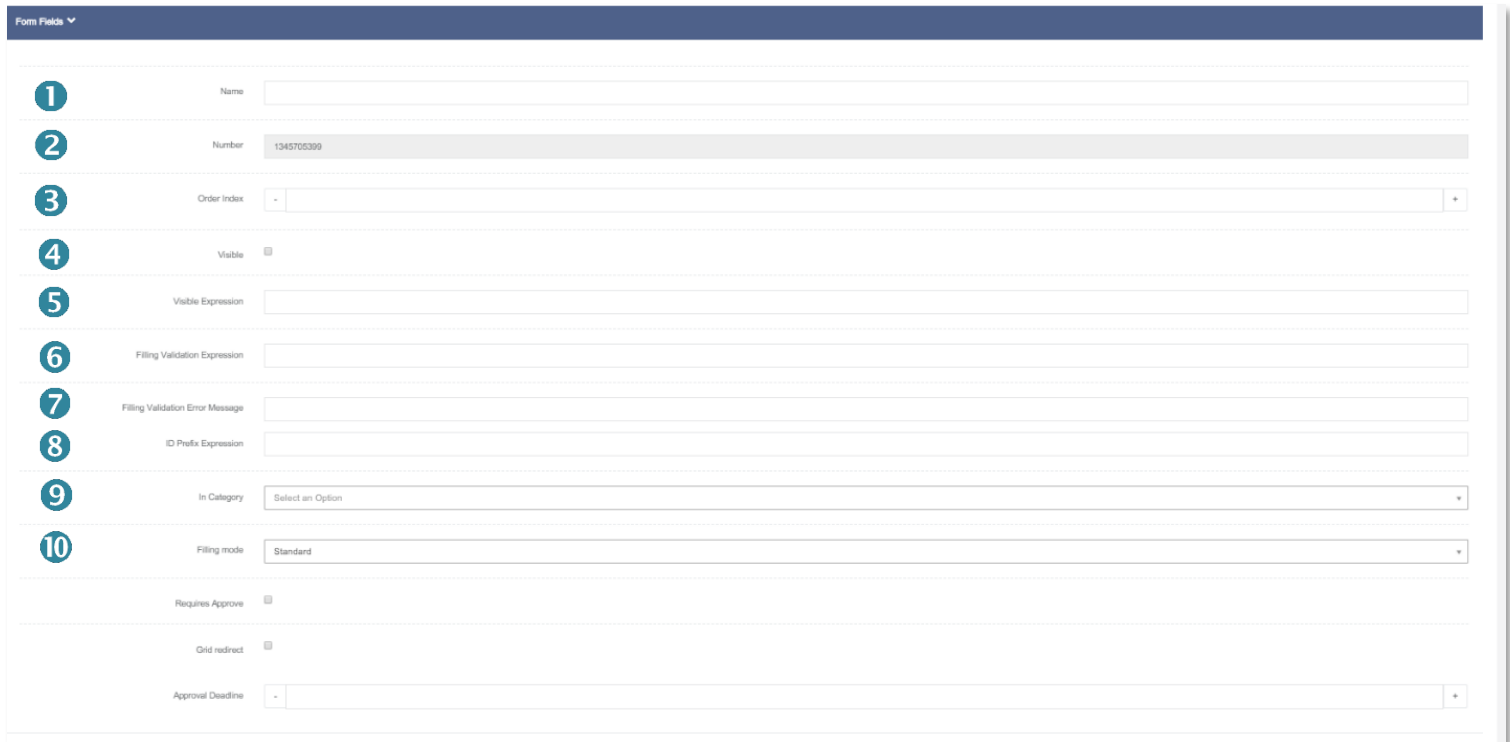
Picture 14. Ready Version of forms

9. Filling Mode

1. Standard: Standard Mode
2. Single Records: Filling the form will be available only once. Final version of the form is not list/table of records, but only filling form.

10. Requires Approve: if this form requires approval (YES/NO)

11. Approved Deadline: Approval deadline



The screenshot shows a 'Form Fields' configuration window with 10 numbered fields:

- 1** Name: Text input field.
- 2** Number: Text input field with value '1345705399'.
- 3** Order Index: Text input field with a dropdown arrow.
- 4** Visible: Checkbox.
- 5** Visible Expression: Text input field.
- 6** Filling Validation Expression: Text input field.
- 7** Filling Validation Error Message: Text input field.
- 8** ID Prefix Expression: Text input field.
- 9** In Category: Dropdown menu with 'Select an Option'.
- 10** Filling mode: Dropdown menu with 'Standard'.

Below these fields are three checkboxes: 'Requires Approve', 'Grid redirect', and 'Approval Deadline' (with a dropdown arrow).

Picture 15. Basic Attributes of the form

6.2.2 Add form elements

In this category, adding elements into the form is available. There are 5 types of elements:

- 1. Field:**
- 2. Grid:**
- 3. Tree:**
- 4. Group:** Groups other elements, except of Tab Container
- 5. TabContainer:**
Element type should be defined initially, which changes dynamically element attributes (**Picture 16** but some attributes are common for all types of elements:
 - 1. Element Type** (See above)
 - 2. Name:** Name of the element. This name will be used in expressions
 - 3. Description** – Description of the element
 - 4. Alias:** Alternative name of the element, while element name contains prohibited symbols (egg **"/"**)
 - 5. Index:** Elements position in the list
 - 6. Visible Expression** -
 - 7. Properties:** Additional attribute of the element, which is a dropdown list and defines which attribute can be modifiable for the element
 - Visible
 - Privacy
 - Unique
 - ReadOnly
 - Mandatory
 - Inversion
 - FilterByUser
 - NotPrintable
 - FirstTimeFill

- DisplayOnFilter
- RequiresApproval
- 8. Dependent Field**
 - Form or element, in which current element value is depend on
- 9. Disable Expression -**
- 10. Visible Expression -**
- 11. Display on Grid**
 - None
 - Always
 - Conditional

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 ▼

Picture 16. Interface of Adding elements

6.2.3 Expressions

Expressions - Implicates all conditions, which should be performed for this element and identical to validation expressions.

- **Visible Expression** - To define field visibility
- **Field Value Expression** - To define Default value for the field
- **Validation Expression** - condition, which should be performed, otherwise error message will be displayed
- **Error Message** – text for error message in case of validation expression is not performed
- **Edit Visible Expression** - Condition to manage or ban edit of the element
- **Disable Expression** - If this condition is done, following element gets disable

6.2.4 Dependency

Dependency - Consists of forms and elements, on which element value or actions is depend:

- **Dependent Field**
 - Forms and elements, on which element value or actions is depend
- **Dependent expression**
 - Condition, which should be performed to make element visible in a form
- **Dependent fill expression**
 - Condition, which should be performed not only to make element visible in a form, but to be filled according to predefined conditions

If only Dependent element is selected, without selecting expressions, element will be visible in a form only if dependent element is not empty.

Field – Field element

Field is a form element, which can be several types:

- TextBox - Short Text
- TextArea – Long Text
- Lookup
- Label
- Date
- Time
- Number
- CheckBox
- ComboBox
- CheckBoxList
- RadioButton
- FileUpload

Besides the common attributes, field element has following specific attributes (**Picture 17**):

- Caption Size
 - Element Name size
- Control Size
 - Control combines element name and element value
- Mask
 - Mask for (eg.: 99999999999 allows only 11 digit number on a field)
- Tag

○ N/A

Element

Element Type	Field ▼	Mask	
Name		Validation Expression	
Description		Error Message	
Alias		Tag	
Order Index	- 10 +	Sort Fields	
Control Type	Select an Option ▼	Dependent field	Select an Option ▼
Caption size	- +	Dependent expression	
Control size	- +	Dependent Fill Expression	
Field Value Expression		Disable Expression	
Visible Expression		Visible Expression	
Properties	Visible ✕	Display On Grid	Select an Option ▼




Picture 17. Field Attributes

6.2.5 Grid

Grid should be filled from existing fields and field elements and they are displayed on a grid as columns, and rows are their values (**Picture 20**).

Element

Element Type

Grid

Name

Description

Alias

Order Index

- 10 +

Delete Visible Expression

Visible Expression

Edit Visible Expression

Properties

Visible

Validation Expression

Error Message

Dependent field

Select an Option

Include In Filter

☐

Disable Expression

Visible Expression

Display On Grid

Select an Option

Picture 18. Grid Attributes

6.2.6 Tree

Tree Is a grid element, which should be filled from existing field elements, but different from grid, field elements in tree is placed hierarchically.

Besides standard attributes, tree elements has “Tree max level” attributes, which value describes highest level in hierarchy. (**Picture 20**). E.g. value “3” means that “Parent Element” only can have 2 level “Child elements”.

6.2.7 Group

Group unites other elements: Fields, Grids, Trees and Groups itself (**Picture 19**). Only one type of element can be organized in a group, e.g. field and tree cannot be united at the same group, but different types of field elements are possible to be together.

Besides standard attributes, Group has following specific attributes:

- Group Size
 - Maximum number of elements in the group
- Background Color
 - Background color of the group
- Title Color
 - Group title color

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

Picture 20. Tree Attributes

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

Picture 19. Group Attributes

6.2.8 Tab Container

Tab is a biggest grouping element and is allocated on the top of interface. Different from the Group, unlimited different types of elements can be grouped in tab. **(Picture 21).**

Element

Element Type

TabContainer

Dependent field

Select an Option

Name

Disable Expression

Description

Visible Expression

Alias

Display On Grid

Select an Option

Order Index

- 10 +

Visible Expression

Properties

Visible

Picture 21. Tab Container Attributes

+

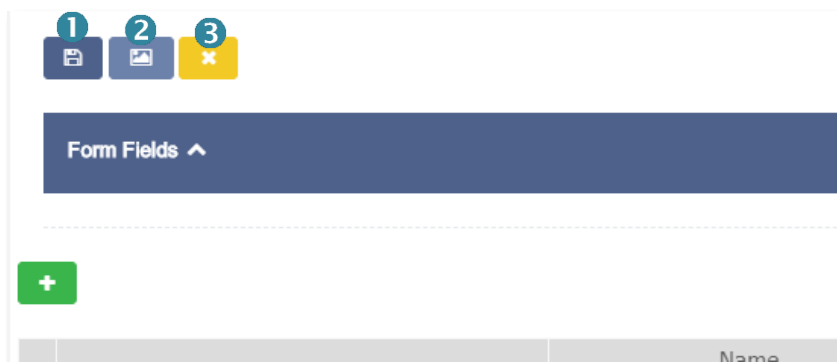
1

	2	3	4	5	6	Name	Element Type	Control Type
						Beneficiary:	Field	TextBox
						Accident status	Field	TextBox
						Lugar status	Field	ComboBox
						Screening form	TabContainer	TabContainer
						Personal Information	TabPage	TabPage
						Personal information of the patient	Group	Group
						Target group	Field	ComboBox
						Target group new	Field	ComboBox
						15 digit code	Field	TextBox
						Note	Field	Label
						personal number	Field	PersonInfo
						last name	Field	TextBox
						Name	Field	TextBox

Picture 22. List of Form Elements (1) Add (2) Edit (3) Delete (4) Paste (5) Copy (6) Add Element (Grid, Tree and Group)

Form elements are placed hierarchically and element can be added directly from the list (**Picture 22**). From this form configuration tab, form can be saved as a template for future use (**Picture 23**).

1. Save
2. Preview
3. Close



Picture 23. Saving the Form template

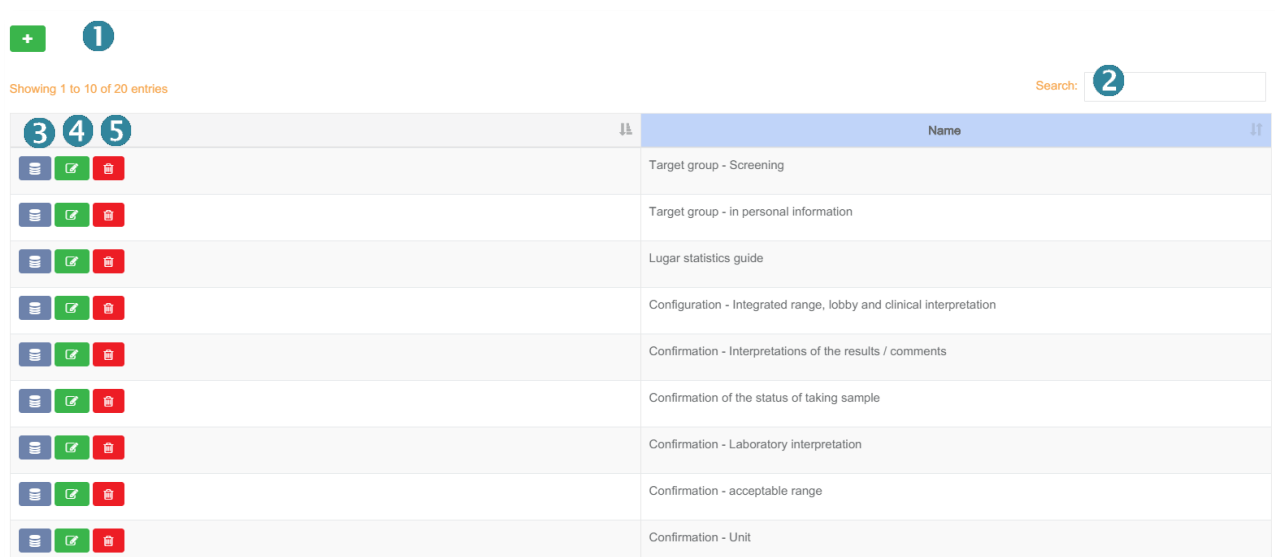
6.3 Logics

These components are used for creating and administering reporting logics.

6.4 Dictionary

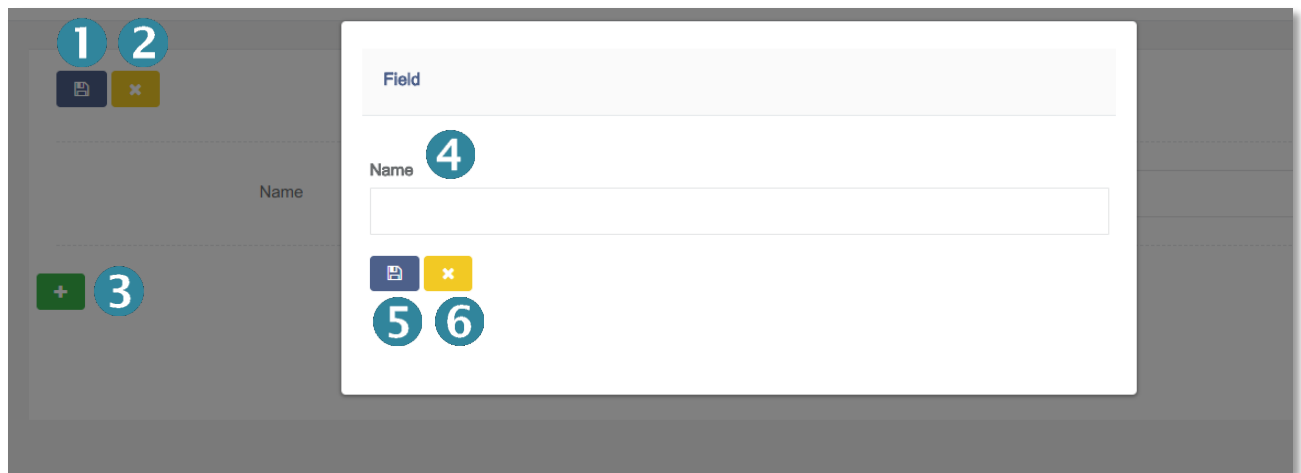
Dictionaries are used for creating and administering any kind of data lists. Main interface of dictionary is list of existing dictionaries (**Picture 24**).

For creating dictionary, Name of the dictionary and Fields should be filled which should be on a dictionary (**Picture 26**).



Picture 24. Dictionary Configuration (1) Add (2) Search (3) Data (4) Edit (5) Delete

For example, in case of address dictionary, following fields should be filled:

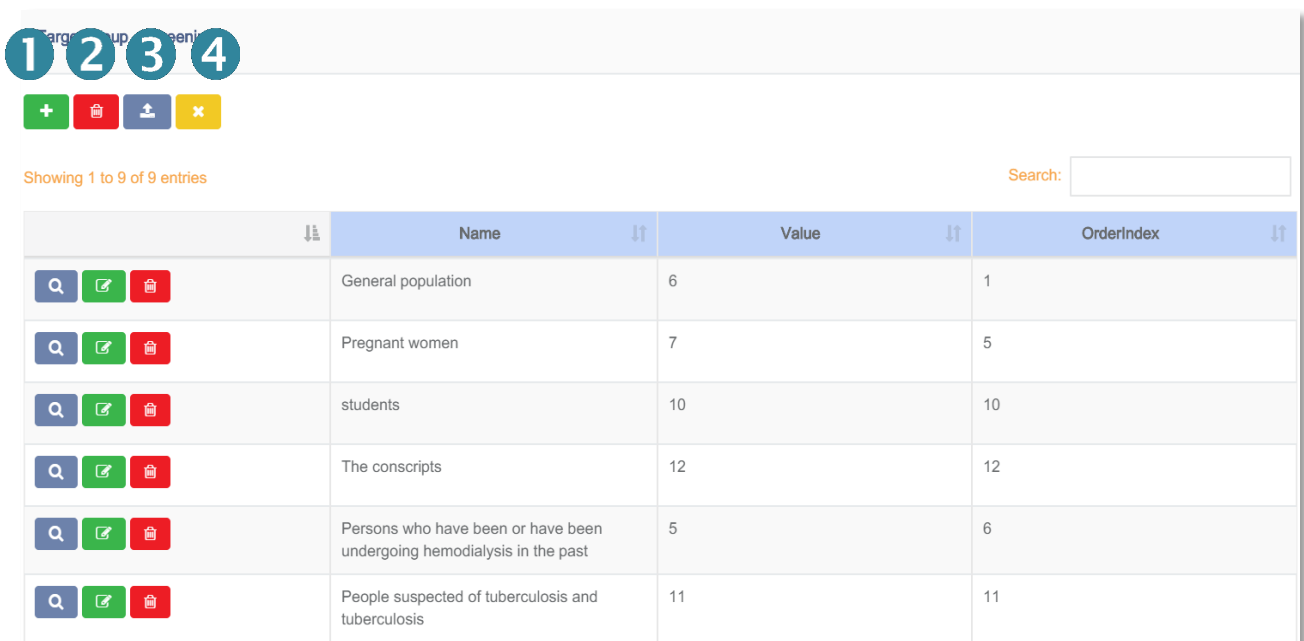


Picture 25. Add New Dictionary (1) Save (2) Close (3) Add Fields (4) Name (5) Save (6) Close

- Name
- Code
- Region
- Post Code

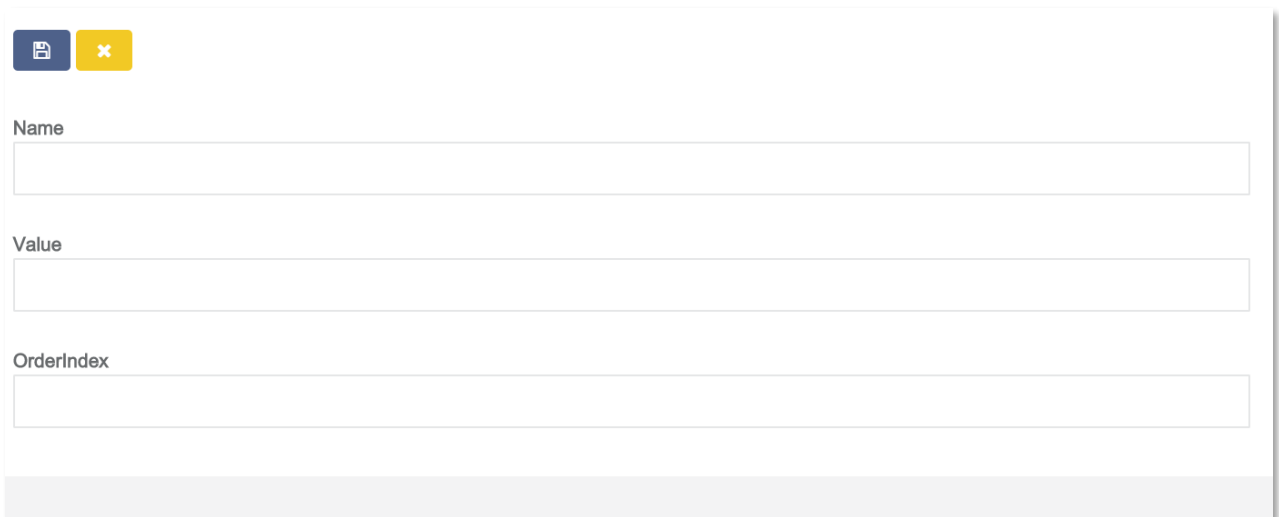
After structuring the dictionary, it should be filled with information (**Picture 25**, **Picture 28**). While filling the information, only fields will be shown which are defined by dictionary structure.

Dictionary can also be filled by importing (CSV, XLS) files. In this case, imported file should be structured with the same logic and ordered as current dictionary.

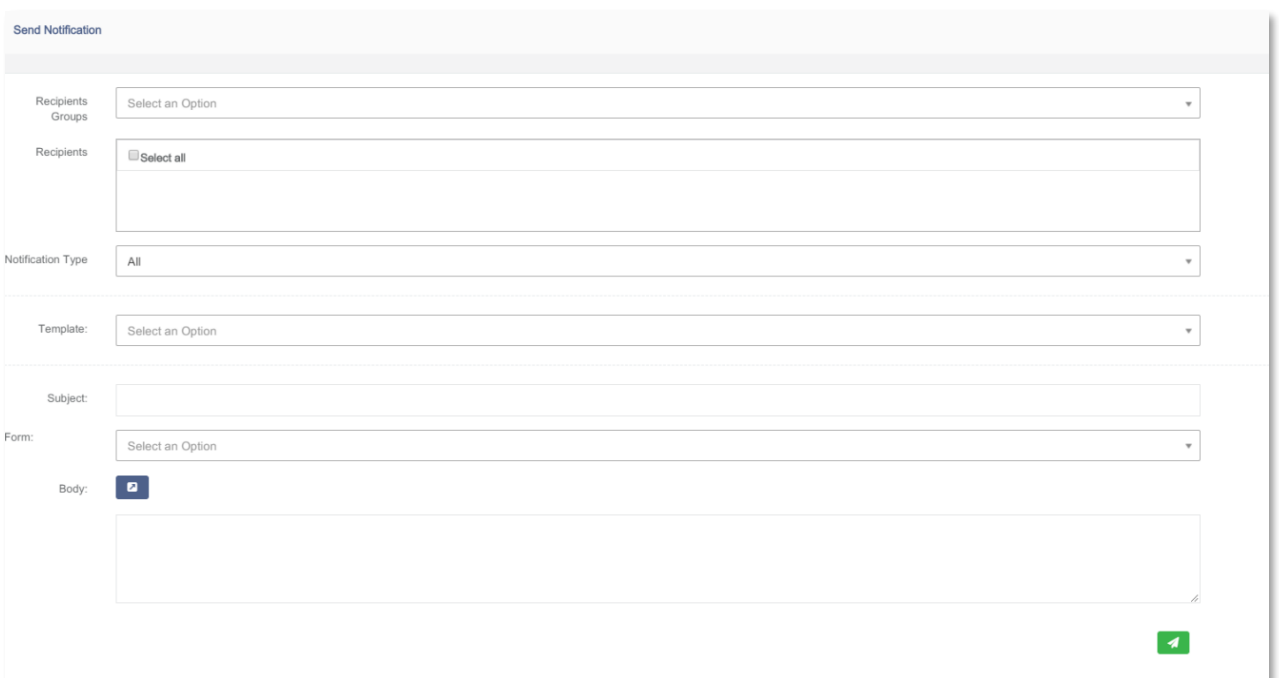


	Name	Value	OrderIndex
	General population	6	1
	Pregnant women	7	5
	students	10	10
	The conscripts	12	12
	Persons who have been or have been undergoing hemodialysis in the past	5	6
	People suspected of tuberculosis and tuberculosis	11	11

Picture 26. Add Data into the Dictionary (1) New (2) Clear (3) Import (4) Cancel



Picture 27. Add Data into the Dictionary: 3900 Necessary fields for adding new data



Picture 28. Notification Tool

7 Additional Links

7.1 Notification

The system is equipped with a function to send notifications from interface, which is not used currently (**Picture 27**).

7.2 Calendar

Calendar allows to create events, their stages, groups and can be shown in desirable form. Calendar is not used currently.

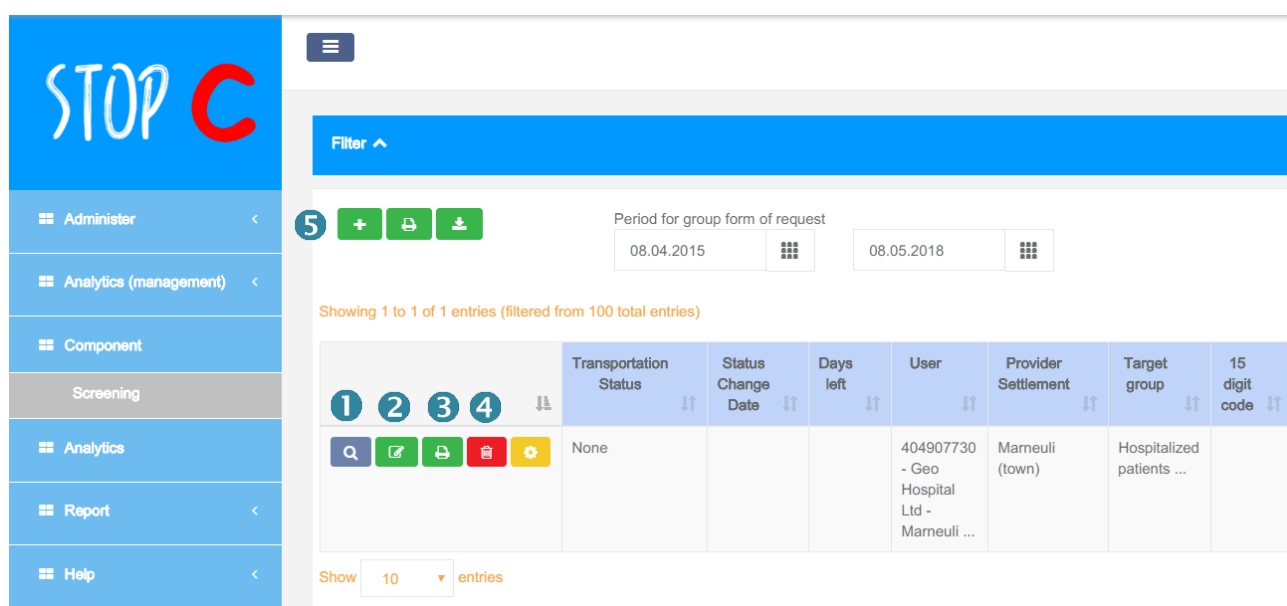
7.3 Component - Screening

Screening form is main working components for system users. User manual for STOP C Screening is provided in separate document.^{5 6}

Administrator, besides administering global resources, can register, view, edit or delete recorded data using relevant tools (**Picture 29**).

8 Analytics

Currently static tables, created in analytics configuration component is visualized in analytics category (**Picture 32**).



Picture 29. Main Configuration Tools (1) View (2) Edit (3) Print (4) Delete (5) Add

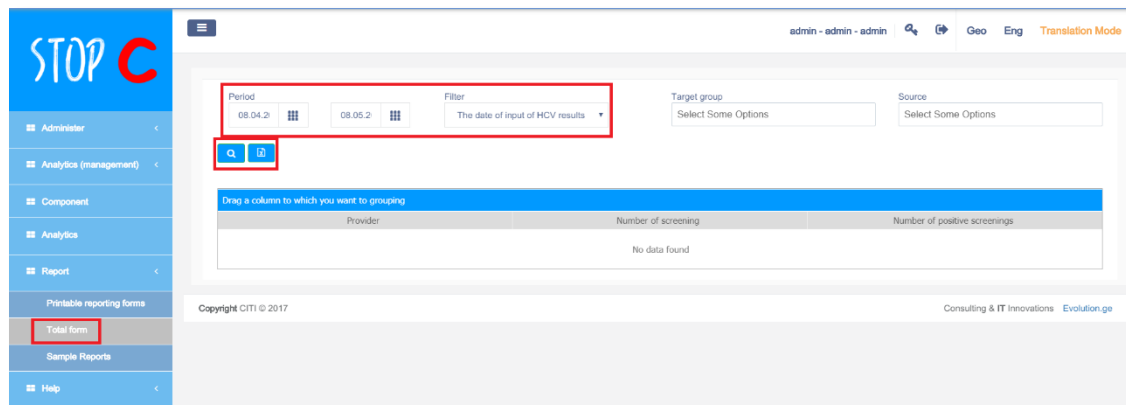
Administrator is able to change graphic type/format and download data in different formats (Excel, CSV, PDF and others, **Picture 33**).

⁵ [Guideline for electronic registration of sample taking and confirmation test electronic registration within Hepatitis C state program](#)

⁶ [Guideline for electronic registration of Hepatitis C screening data](#)

9 Report

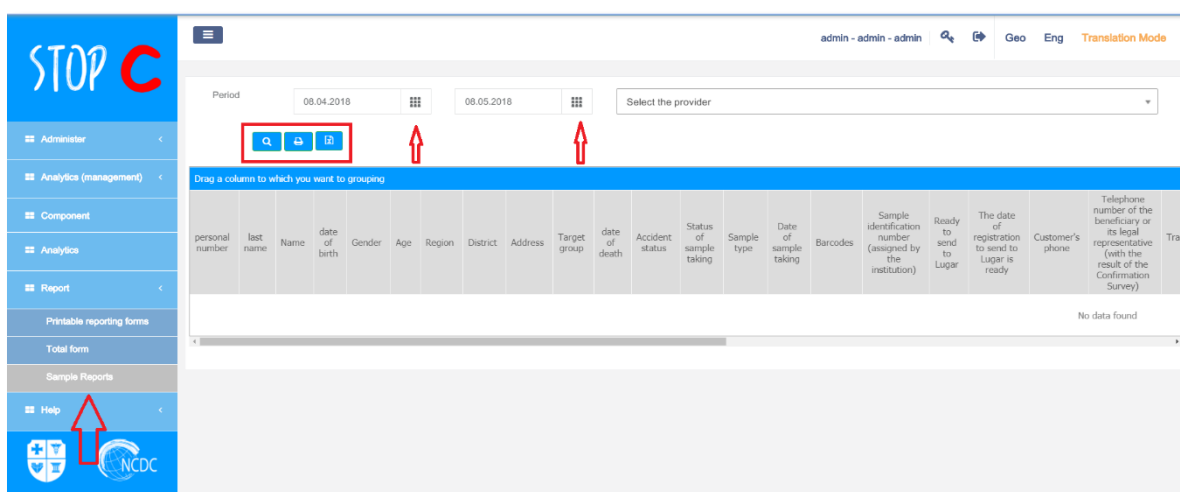
Administrator is able to retrieve detailed report of all registered screening cases, sample taking and confirmation tests in navigation panel using “Printing reporting forms”, “Cumulative forms” and “Sample Reports” (**Picture 30**).



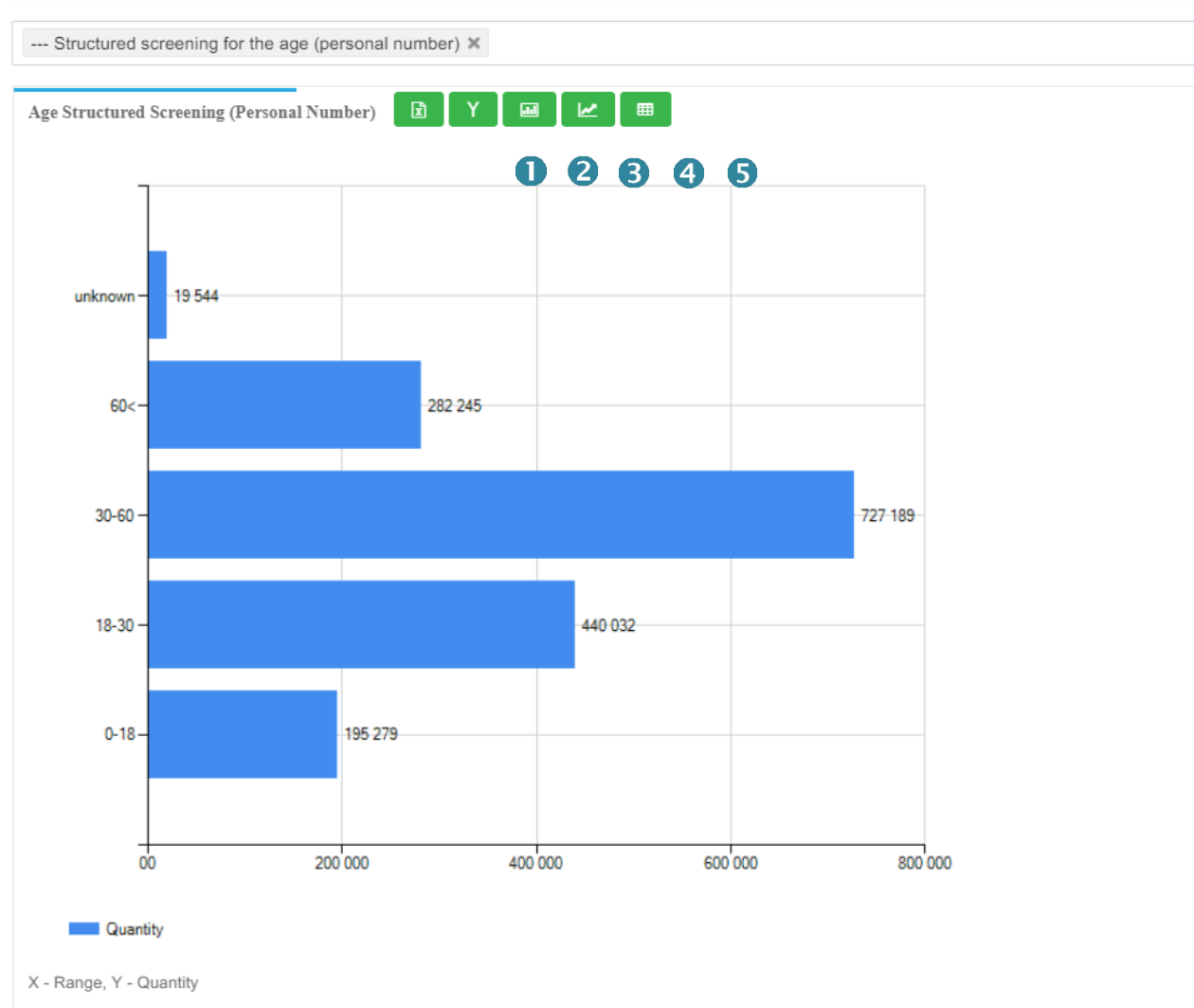
Picture 30. Report of HCV Screening test results

Data can be filtered using period filters for screening and sample taking (**Picture 31**).

Via print and Excel buttons, filtered and searched data can be printed or downloaded in Excel format.



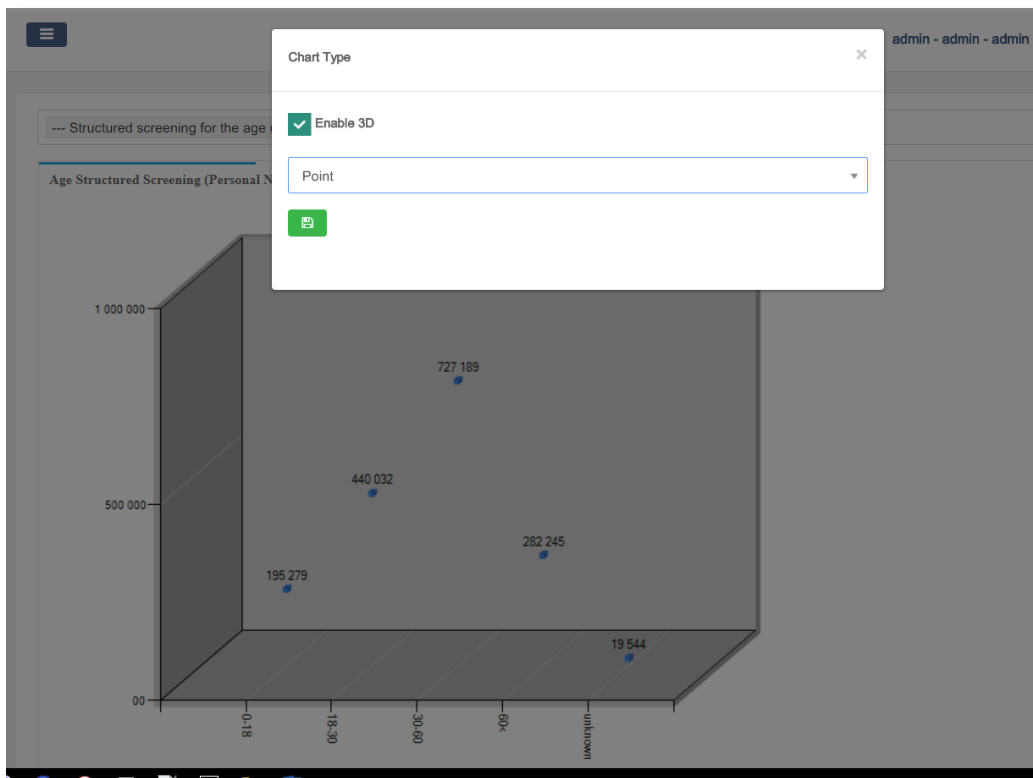
Picture 31. Report of Sample and confirmation test results



Picture 32. Graphic visualization of Static table and its configuration tools: (1) Export. (2) Y Value (3) Chart type (4) Table (5) Table

In reports category from the system can be downloaded:

- Patient's Demographic data
- Data related to screening tests
- Data related to sample and sample taking
- Data related to confirmation tests



Picture 33. Chart Type (including 3D)

10 Appendix A: Validation Expressions

For validation system form components and attributes, system is equipped with specially modified Validation Expressions.

10.1 Operations

Operation	Value	Comment
!	Logical "No"	Function IsEmpty(value) returns true/false depend on parameter value is empty or not. But if we want reverse option Is not empty, has any value, than we write "IsEmpty(value)"
+	Addition	
++	Increment	Typically adds the integer 1
-	Subtraction	
--	decrement	Typically subtracts the integer 1
==	Equality	e.g. a==b, checks value of a, if it is equal to b or not. This returns reference equality (identity test)
!= <>	Not equal.	e.g. a!=b or a<>b, checks if a is not equal to b
<=	Less than or equal to	
>=	Greater than or equal to	
<	Less than	
>	Greater than	
&&	Logical AND	E.g. a==b && b==c means that a is equal to b and b is equal to c
	Logical OR	E.g. a==b b==c means that a is equal to b or b is equal to c
^	Exclusive XOR	a^b - a power of b
&	Concatisation	String Concatisation Operation. If we have firstName and lastName and want to concatenate their values and separate with "-", we use firstName & ' - ' & lastName
%	Percentage	a%b same as [a / 100 * b]
	Reminder	
*	Multiplication	
/	Division	

10.2 Functions

Functions	Description
sqrt(value)	Square Root
sin(value)	Sinus
cos(value)	Cosinus
tan(value)	Tangens
asin(value)	Arcsinus
acos(value)	Arccosinus
atan(value)	Arctangens
abs(value)	Absolute value
ceil(value)	Rounds up the nearest integer
exp(value)	Exponential value
floor(value)	Rounds to an integer towards 0
log(value, base)	Logarithm calculator with Base
log10(value)	The logarithm to the base 10
pow(value, base)	Exponent used to raise the base
or(value1, value2)	Binary "OR" Operation ⁷
and(value1, value2)	Binary "AND" Operation ⁸
xor(value1, value2)	Exclusive „OR“ (XOR) ⁹
mod(value1, value2)	Modulus operation ¹⁰ - returns remainder but also calculates division in the process (E.g: mod(19, 4) returns 3 as 19 divided into 4 is equal to 4 and reminder is 3)
substring(value, index)	Substring method attempts to extract characters from index startIndex to index startIndex + length - 1. To extract a substring that begins with a particular character or character sequence, call a method such as IndexOf or LastIndexOf to get the value of startIndex.
substring(value, index, count)	Returns sub string Index and substring length (E.g. if value is equal to abcdefg, then substring (value, 3, 2) returns d, as count started from 0 and thirds 3 symbol is d)
getdatetime(value)	Returns Current Date and Time
getDate()	Returns Current Date
getDate(year)	Returns just the year (Month, day and others will be current)
getDate(year, month)	Returns Date (others :Day, Hours and so on will be current)
getDate(year, month, day)	Returns Date (is necessary for Datetime operations, e.g. addDays(getDate(2017, 02, 21), -15))
getLength(value)	Returns length

⁷ https://en.wikipedia.org/wiki/Bitwise_operation#OR

⁸ https://en.wikipedia.org/wiki/Bitwise_operation#AND

⁹ https://en.wikipedia.org/wiki/Exclusive_or

¹⁰ https://en.wikipedia.org/wiki/Modular_exponentiation

isEmpty(value)	The ISEMPY function returns TRUE if the value is a blank cell or uninitialized variable. The ISEMPY function returns FALSE if the value is a cell or variable that contains a value
isDate(value)	Determines if a value is a date
isDay(value)	Determines if a value is a day (More than 0 and less than 32)
isMonth(value)	Determines if a value is a Month (More than 0 and less than 13)
isYear(value)	Determines if a value is a Year
isNumber(value)	Determines if a value is a number
isInteger(value)	Determines if a value is a Integer
rgx(value, exp)	Defines a search pattern with Regex ¹¹
min(a, b)	Returns Minimum
max(a, b)	Returns maximum
if(value, a, b)	if(x==1, a, b) – if x=1, returns a, or b
comp(value)	Comparator (Returns -1, 0, 1)
lower(value)	Changes to Lower letter
upper(value)	Changes to Upper letter
contains(value, text)	Function returns the location of one text string inside another. (Returns true/false), e/g contains (value, "hello")- If Text in value Contains the word hell
startsWith(value, text)	Returns a logical value indicating whether a text value substring was found at the beginning of a string
endsWith(value, text)	Returns a logical value indicating whether a text value substring was found at the end of a string
trim(value)	The trim() method removes whitespace from both sides of a string
LTrim(value)	This function returns a string with whitespace stripped from the beginning of string
RTrim(value)	Returns a string with whitespace (or other characters) stripped from the end of string
get(collection, nameOrIndex)	Returns item of array by index or value of dictionary by key
getDaysInMonth(value) getDaysInMonth(value, year)	Returns the number of days in a specific month of a year (if year is not defined in parameters, it counts current year) e.g.: getDaysInMonth(2) returns 28, but getDaysInMonth(2, 2016) returns 29
getYears(date)	Returns the year for the specified date (2017-02-17 15:32:25)
getMonths(date)	Returns the month (from 0 to 11) for the specified date, (2017-02-17 15:32:25)
getDays(date) getDayOfMonth(date)	Returns the name of the weekday (2017-02-17 15:32:25) (If not Returns primitive int value for the day-of-month.
getDayOfWeek(date)	Returns the day of the week
getDayOfYear(date)	Returns primitive int value for the day-of-year.
getHours(date)	Returns Hours (2017-02-17 15:32:25)

¹¹ https://en.wikipedia.org/wiki/Regular_expression

getMinutes(date)	Returns Minutes (2017-02-17 15:32:25)
getSeconds(date)	Returns Seconds (2017-02-17 15:32:25)
getTotalDays(date)	Returns total number of days from 0001-01-01 01:01:01- till indicated date
getTotalHours(date)	Returns total number of Hours from 0001-01-01 01:01:01till indicated date
getTotalHours(date)	Returns total number of Minutes from 0001-01-01 01:01:0101till indicated date
getTotalSeconds(date)	Returns total number of Seconds from 0001-01-01 01:01:0101till indicated date ¹²
addYears(date, count)	To add a given number of years to a date ¹³
addMonths(date, count)	To add a given number of months to a date (count quantity)
addDays(date, count)	To add a given number of days to a date (count quantity)
addHours(date, count)	To add a given number of hours to a date (count quantity)
addMinutes(date, count)	To add a given number of minutes to a date (count quantity)
addSeconds(date, count)	To add a given number of Seconds to a date (count quantity)
select(collection, selector, condition)	Selects rows/items. Finds data collection by condition (if condition is specified) E.g. there is a collection Products with three columns: Name, Amount, Cost. To select all product, which has name and value more than 10: select(Products, get(@, 'Name') & ' - ' & get(@, 'Cost'), get(@, 'Amount') > 10) @ - Means current record, so function “select “should check all records in the collection, checks if each record meets the conditions and returns result to selector.
sum(collection)	Returns sum of numeric values in collection. Like Select, if we want to retrieve Products from the collection, which has price more than 100: sum(select(Products, get(@, 'Amount'), get(@, 'Cost') > 100)) min – works the same ways as sum, but returns minimum value max - works the same ways as sum, but returns maximum value

¹² Following four function is necessary when we need to calculate days, hours, minutes and seconds difference in two dates E.g. getTotalHours(Date1 - Date2) returns difference hours between Date1 and Date2.