Electronic form definition reference guide

(Release 2023-03-06)

Contents

Introduction 2

Where to start? 2

Definition of a new electronic form 3

Definition of an existing electronic form 3

Design of an electronic form 5

- Data input fields, see details in [On-screen layout 5

Form field design 7

Form design reference 9

- Address (URL) to which the data will be submitted 9
- The name and the name extension 9
- The description of a component (additional data) 9
- Electronic form configuration restrictions (restricted edit) 10
- Preview and explore the electronic form 13
- On-screen layout 15

The component's configuration reference 17

- addresses 17
- atc 17
- dates 19
- dictionaries 20
- documents 21
- droplist 22
- headings 23
- intervals 24
- legacy 25
- links 26
- literals and strings 28
- logical 28
- numbers 29
- persons 30

registers 31
resources 32
schedulers 33
things 34
String validation patterns 35

Introduction

This guide helps OpenRIMS Supervisors design electronic forms.

The OpenRIMS "Administration-Configuration-Data Configurator" feature allows the designing of electronic form data input controls and their on-screen layout.

An electronic form collects high-cohered data and files while business and NMRA electronic collaborations.

If it will be necessary to collect many blocks of low coupling data are possible to define a sequency of electronic forms.

Where to start?

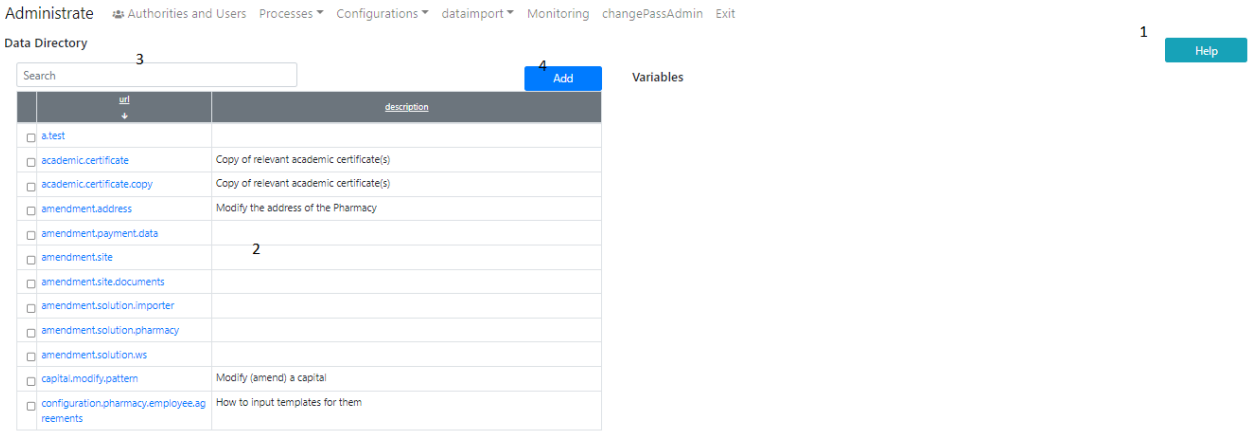


Figure 1 Initial screen of the Administrate-Configurations-Data Configurator feature

1. Get this document
2. List of the existing electronic forms sorted alphabetically by Url
3. Search for the existing electronic forms
4. Add a new electronic form

Definition of a new electronic form

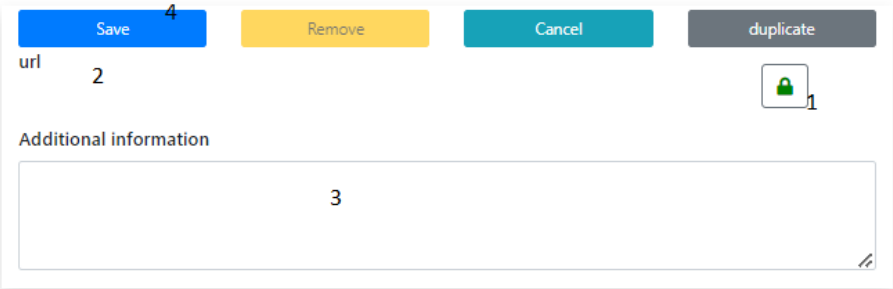


Figure 2 Electronic form definition describes the form

1. Click here to unlock the field “Url”
2. Address (URL) to which the collected data and files will be submitted. To assign a URL, please use these recommendations:[Address (URL) to which the data will be submitted]
3. Help text or description
4. Save the electronic form definition

The “additional information” is a helpful text.

Definition of an existing electronic form

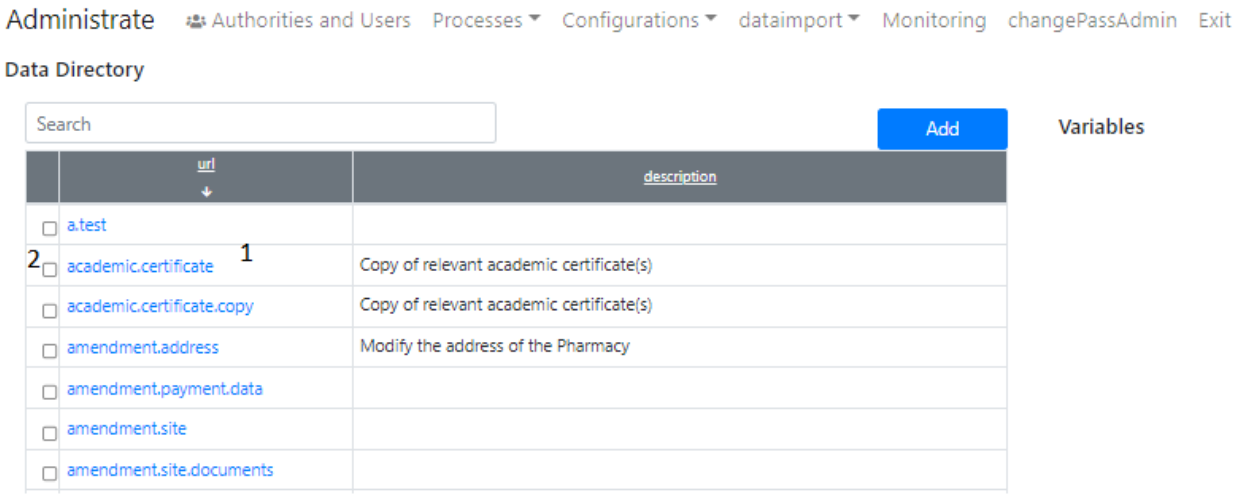


Figure 3 Access existing form definition and design

1. Open the definition
2. Open the design

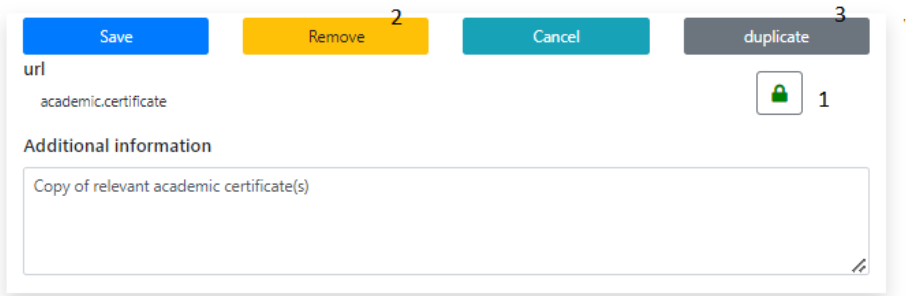


Figure 4 For existing electronic form "Remove" and "Duplicate" features are active

1. Despite it is possible to change the Url of the existing form it is an extremally dangerous operation. In most cases it will be better:
   1.1. Duplicate the form
   1.2. Remove the original
   1.3. Rename the duplicate
2. Remove the feature that makes the form definition inactive:
   2.1. This electronic form will be not available for design
   2.2. The form will be available for data input/display
3. The duplicate feature allows the creation exact copy of the electronic form design. Url will be renamed to original_url.copy, e.g., academic.certificate.copy

Design of an electronic form

Variables

The edition of this data configuration is restricted due to external references

Search

1 Add 2 Preview 3 explore

row	col	Order	Variables	ext	Class	Additional information
0	0	0	register		registers	initial register number
0	0	1	retail_pharmacy		heading	
0	0	2	prefLabel		literals	Enter name of pharmacy in English. Should include the word "Pharmacy" both in the Nepali and English name.
0	0	3	name_nepali		strings	
0	0	4	tolename		strings	
0	0	5	wadano		numbers	1-32
0	0	6	address		addresses	
0	1	5	docs_tenancy		documents	
0	1	9	photo_resize		heading	resize photo
0	1	10	site_photos		documents	
100	100	0	classifiers		things	
100	100	1	pvt_data		things	PVT name, address and registration data
100	100	2	person_auth		things	Person with Authority
100	100	3	pharmacists		things	Pharmacists
100	100	4	report_payment		things	Payment data
100	100	5	cover		things	

4

5

100

Figure 5 Design of an electronic form

1. Add a new field to the form layout see details in [Form field design]
2. Preview the form, see details in [Preview and explore the electronic form]
3. Export design details to MS Excel for exploring, see details in [Preview and explore the electronic form]

Data input fields, see details in [On-screen layout

The basic configuration layout of an electronic form is in two columns. Depending on the room on the screen a form may be represented in two or one column. In the case of the one-column representation, the second column will be placed under the first one.

The numbering of rows and columns starts from zero.

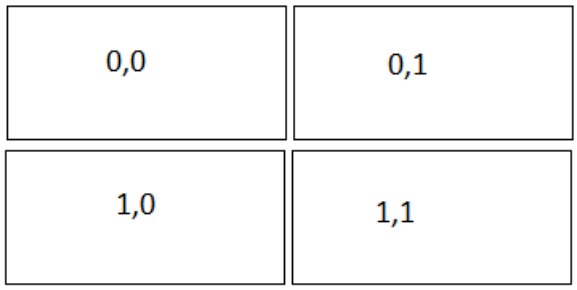


Figure 21. The two columns' layout

The height of components may be very different. Thus, placing one component in a cell (row, col) is almost unusable

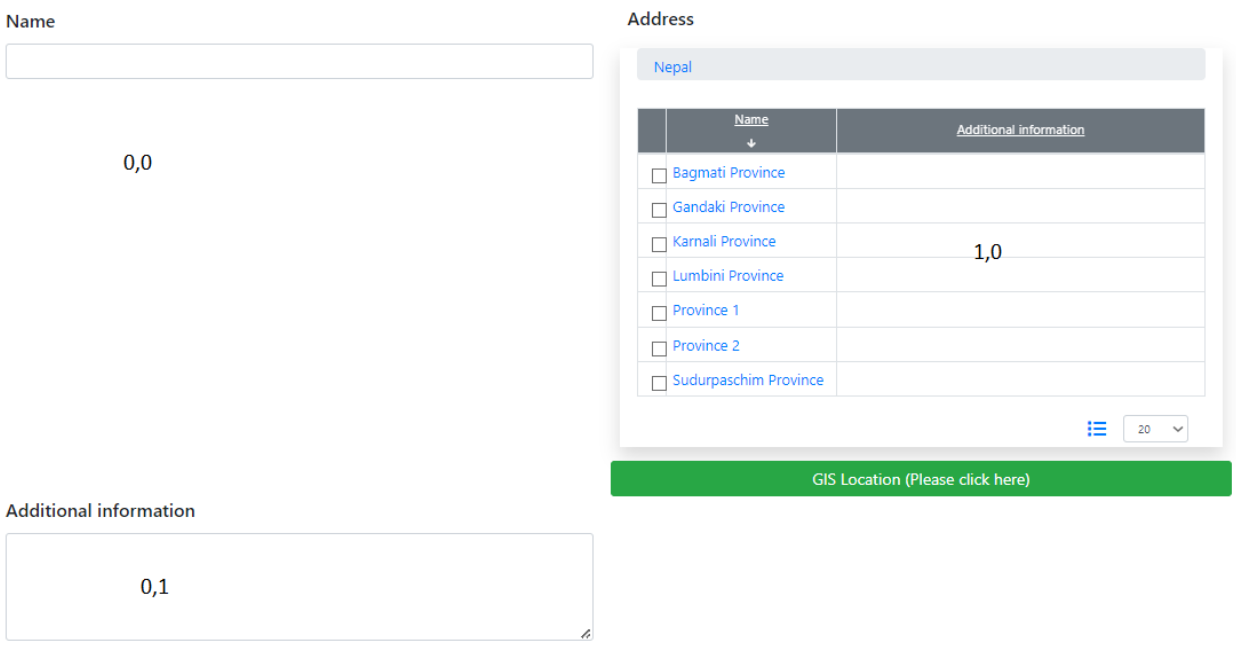


Figure 22. The only cell is not enough

To avoid this, OpenRIMS provides the third coordinate – an order inside the cell. The order starts from zero and allows the place of many components in a cell.

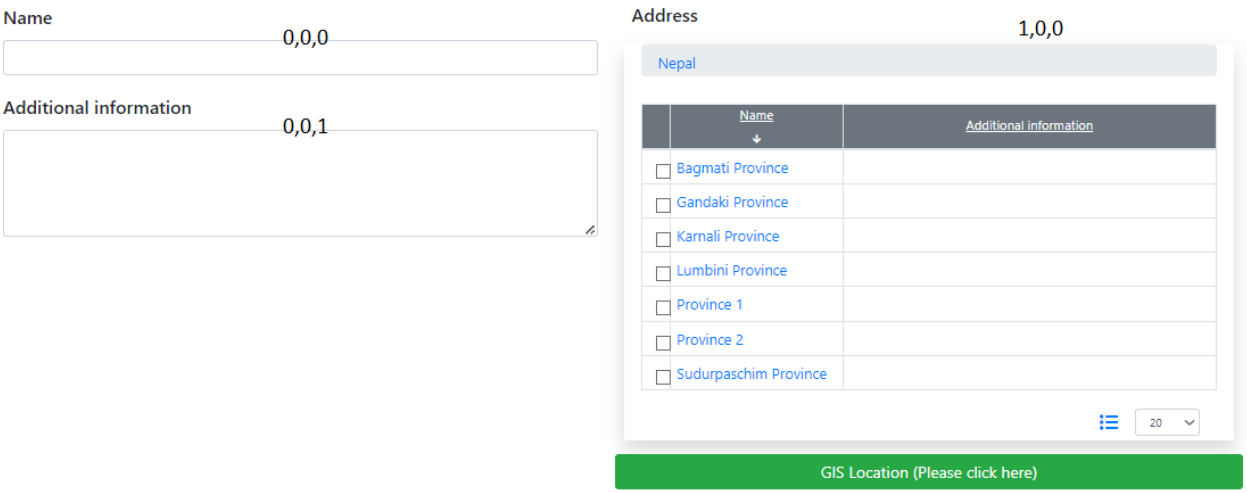


Figure 23. Cell and order inside a cell are acceptable

OpenRIMS layout editor allows the define and re-arrange components easily using the following rules:

- Columns number possible only 0 and 1
- Values of rows and orders allowed up to 89. Row number 100 is reserved for “things”

When the user tries to place a component in a busy position inside a cell, the rest of the components in this cell will be shifted down

4. The component's configuration reference]:
   4.1. Row, col, Order – the layout, see details in [On-screen layout]
   4.2. Variables and ext – the label of a data input field, see details in [The name and the name extension]
   4.3. Class – the type of the data input component, see details under [The component's configuration reference]
   4.4. Additional information – help text
5. A sequency on other electronic forms. Parameter “row” is 100, Class is “things”

Form field design

A form field is a data input component that allows an electronic form user:

- Input text, numbers, dates
- Select data from a set of pre-defined choices
- Download/upload a file
- Open a link to another website

The types of possible input data components are described under [The component's configuration reference] For a particular type it will be convenient to search using the Table of Contents at the top of this document

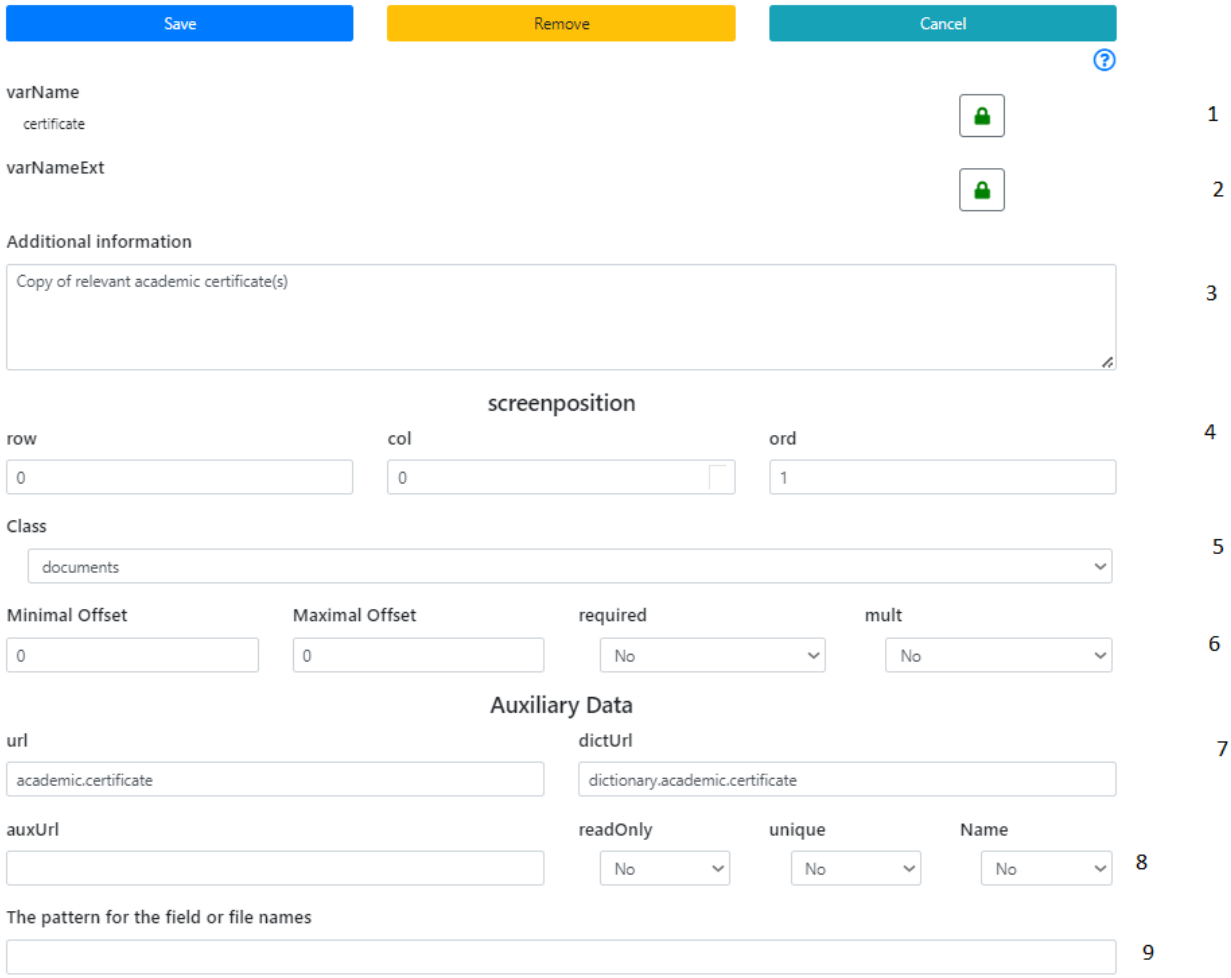


Figure 6 From field design electronic form

1. Click to edit the name of the component [The name and the name extension]
2. The experimental feature that allows change validation rules depends on the user's choice. The only known implementation is the "intervals" component
3. The help [The description of a component (additional data)]
4. Select a type of component, e.g., text field, or file uploader. See under [The component's configuration reference]. It will be better to use the Table of Contents at the top of the document.
5. Validation parameters. Refer to a description of the particular component.
6. Screen position [On-screen layout]
7. Refer to the description for a particular component
8. Refer to the description for a particular component, typically:
   8.1. readOnly – this component should be placed in the form of a read-only mode
   8.2. unique – experimental feature, works only for strings and literals
   8.3. Name – copy some data from this component to literal "prefLabel" and, possibly to other literals.
9. Validation rules data - sizes, regular expressions (https://regex101.com/) for texts, etc. Refer to the description for a particular component.

Form design reference

Address (URL) to which the data will be submitted

As common for the Internet, OpenRIMS stores the data collected by one page at a unique internal address or URL (Universal Resource Locator). For example, all personal academic certificates will be stored on the internal address “academic.certificates”, all general medical product data will be stored on “medicinal.product.definition”, etc. The FHIR implementation of the HL7 standard provides a clue on how to assign URLs for electronic form pages.

The name and the name extension

Attention! The name extension is the experimental feature that is in use only for the “intervals” component.

A variable name in the data configuration should be alphanumerical with possible Latin characters, dots, and underscores. The last underscore or dot should be followed by a number or Latin character. The name should be started with a Latin character. For example:

- min.value.1 – right
- минимальное значение – wrong
- c1 – right
- 1c – wrong
- Etc.

In an on-screen form, the software tries to convert a variable to a field label using the internationalization feature, even if only one language is defined. In case, if for a variable the message is not defined, the variable will display “as is”.

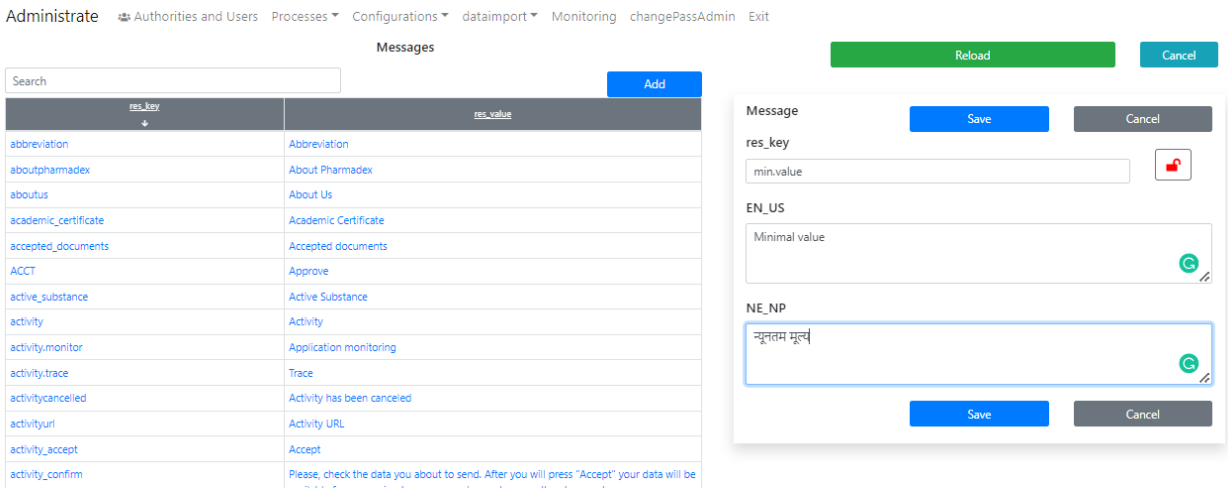


Figure 7 Each variable should be represented in human-readable form using the "messages" feature

The description of a component (additional data)

The description or “additional data” of a component is a text that should help a user to understand how to fill out the component. Examples are:

- “Please, select a main business of your company” – good

- “The key to recognizing the type of a company” - bad

Electronic form configuration restrictions (restricted edit)

For the production installation, the changing electronic form configuration is dangerous.

The following changes will badly affect the existing data:

1. Re-naming a component
2. Removing a component
3. Changing the type of a component

By default, the changes listed above are disabled if at least one of the conditions below occurs:

- The database already contains data created using the data configuration
- The data configuration is used by other data configuration
- The data configuration is used by activity configuration
- The data configuration is used by resource configuration

In the “sandbox” installations it is possible to bypass this restriction. To do this you must have access to edit the “application.properties” file in the server installation folder[1]

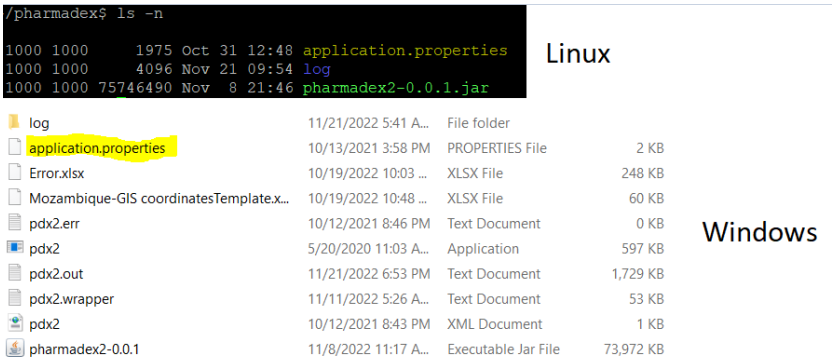


Figure 8 application.properties

Use any text editor to add to this file the configuration parameter variables.properties.edit=true. It allows bypassing the validation above. To stop bypassing the rules, remove this line from the application.properties or set it to “false”.

variables.properties.edit=false

[1] Ask IT personal to do it for the Sandbox installation

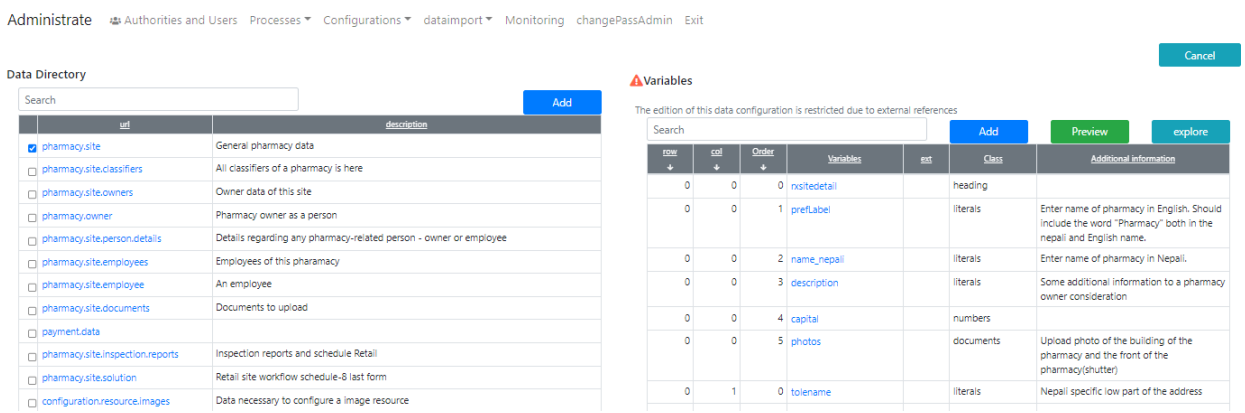
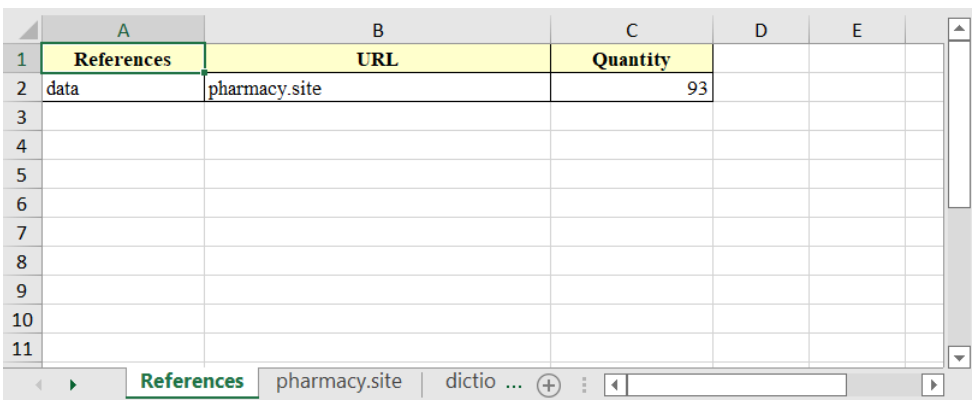


Figure 9 Editions are restricted for pharmacy.site

It is possible to explore which references are in place. The button "explore" will provide both references and structure of the data in the MS Excel format:



References	URL	Quantity
data	pharmacy.site	93

Figure 10 There are 93 data records of this structure in the database. Thus the editions are restricted

To allow restricted editions in a data configuration:

1. Duplicate the data configurations
2. Change the URL of it
3. Make the editions
4. Change references if it will be needed

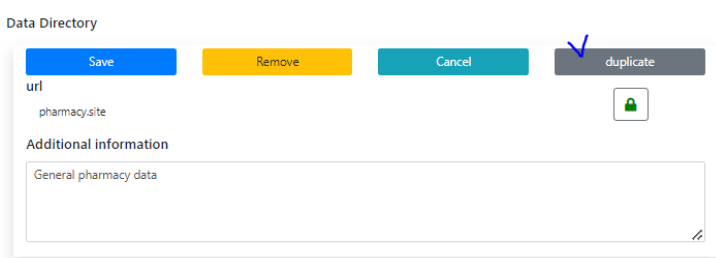


Figure 11 Click on it to duplicate

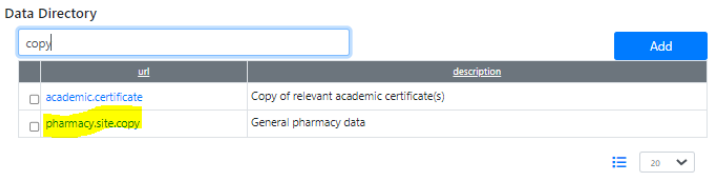


Figure 12 Find a copy just created

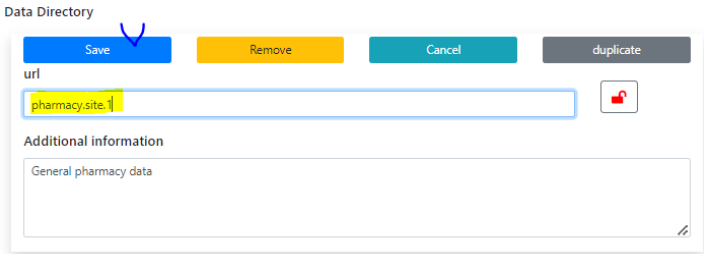


Figure 13 Change the URL and save it

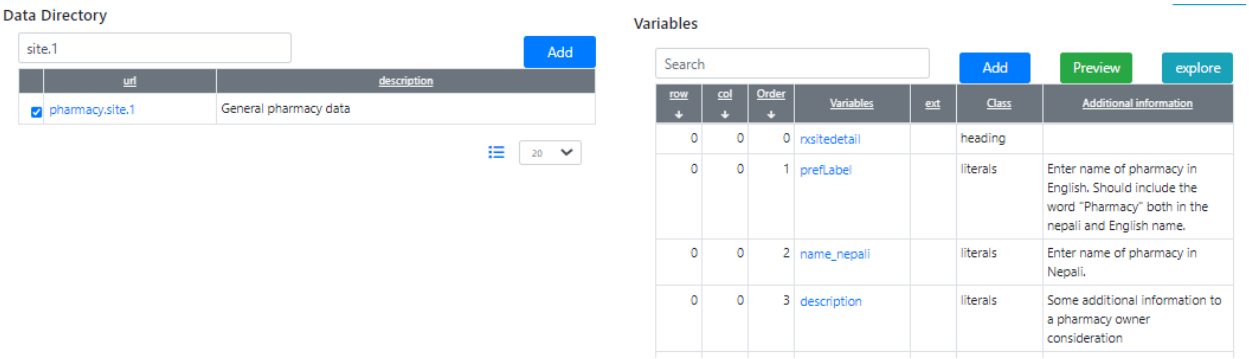


Figure 14 The editions are not restricted to the new data configuration

Preview and explore the electronic form

The data configuration contains data components and, possibly, other data configurations. It is possible to preview the electronic form and explore the whole configuration.

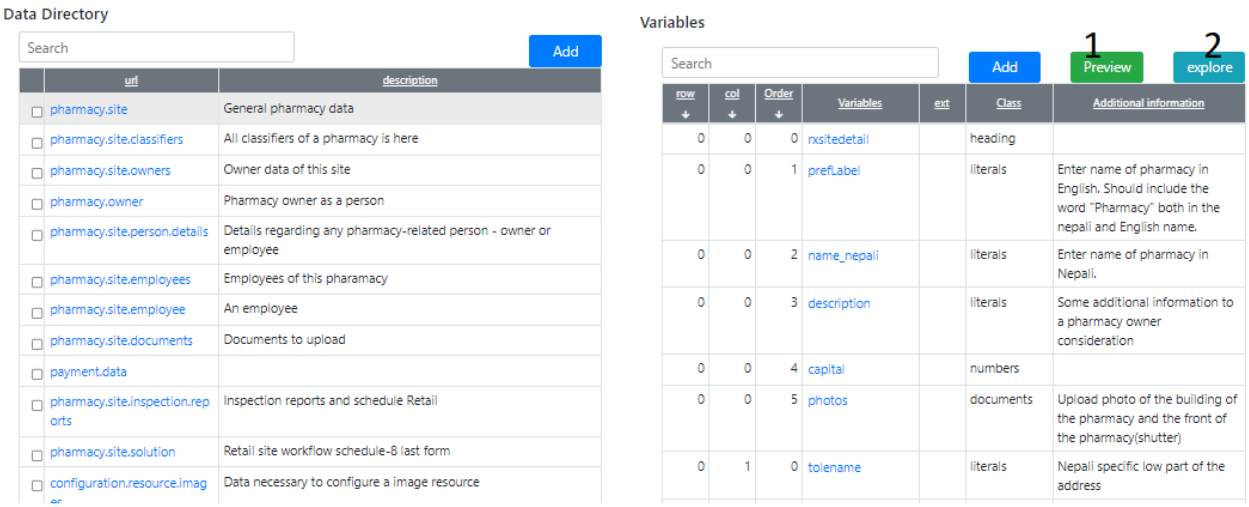


Figure 15 Preview and explore features

The preview allows checking data components' layout and validation rules.

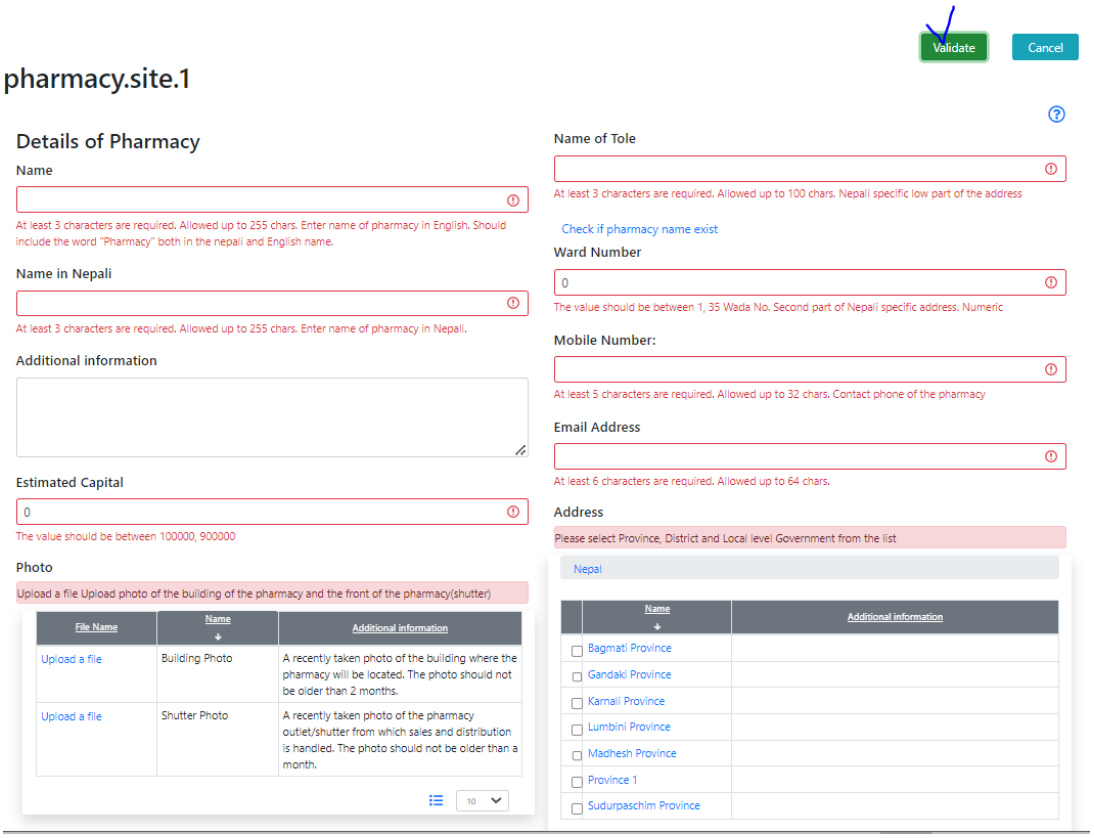


Figure 16 The layout. Press "Validate" to check the validation rules

If the component contains “things” the preview will not check them. The “explore” feature allows checking data structures included and getting a list of references to the data structure.

References	URL	Quantity
data	pharmacy.site	93

References | pharmacy.site | dictionary.pharm ...

Figure 17 There are 93 records created using this data configuration

dictionary.pharmacy.site.photoss			
level	Name	Additional information	URL
Bad configuration			

dictionary.pharmacy.site.phot | pharamacy.site ...

Figure 18 Wrong URL for the dictionary

pharamacy.site.classifiers								
Bad configuration								
varname	varnameext	descr	Required	Mult	ReadOnly	Url	DictUrl	Mi
Bad configuration								

pharamacy.site.classifiers | pharmacy.site.owner ...

Figure 19 Wrong URL for a thing

	A	B	C	D	E	F	G	H	I
1	**pharmacy.owner**								
2	**Pharmacy owner as a person**								
3	**varname**	**varnameext**	**descr**	**Required**	**Mult**	**ReadOnly**	**Url**	**DictUrl**	**Mi**
4	persondetails								
5	prefLabel		Name in English	+					
6	namenepali		Name in nepali	+					

... pharmacy.site.owners **pharmacy.owner** ph ...

Figure 20 It's OK

On-screen layout

The basic configuration layout of an electronic form is in two columns. Depending on the room on the screen a form may be represented in two or one column. In the case of the one-column representation, the second column will be placed under the first one.

The numbering of rows and columns starts from zero.

0,0	0,1
1,0	1,1

Figure 21. The two columns' layout

The height of components may be very different. Thus, placing one component in a cell (row, col) is almost unusable

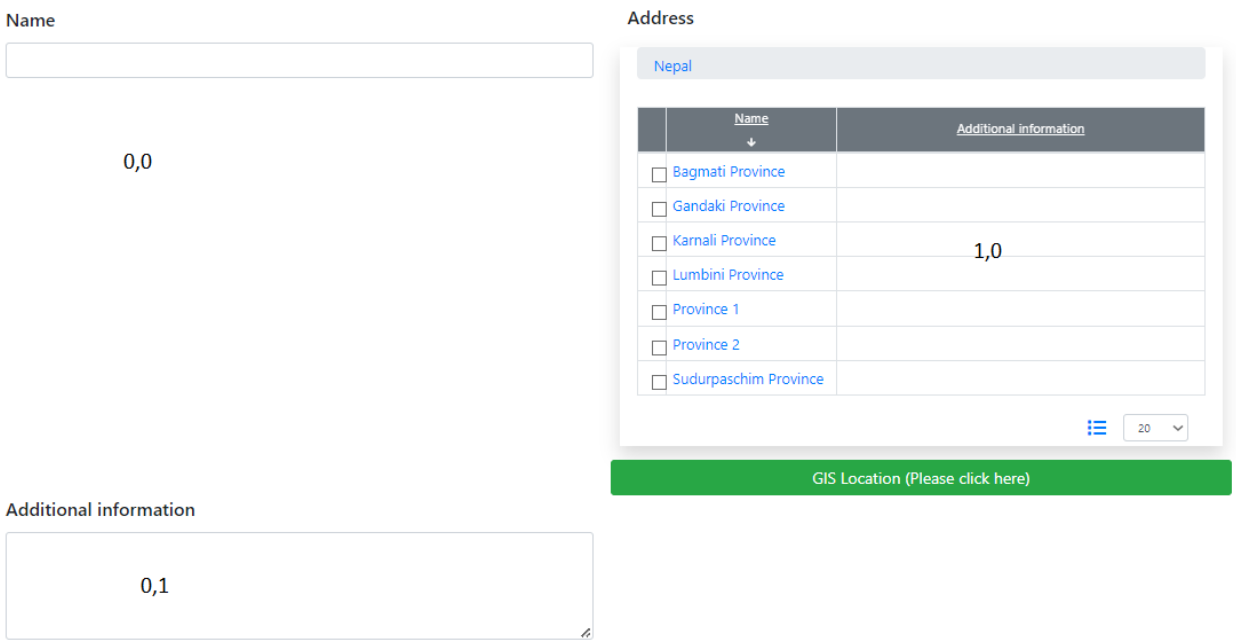


Figure 22. The only cell is not enough

To avoid this, OpenRIMS provides the third coordinate – an order inside the cell. The order starts from zero and allows the place of many components in a cell.

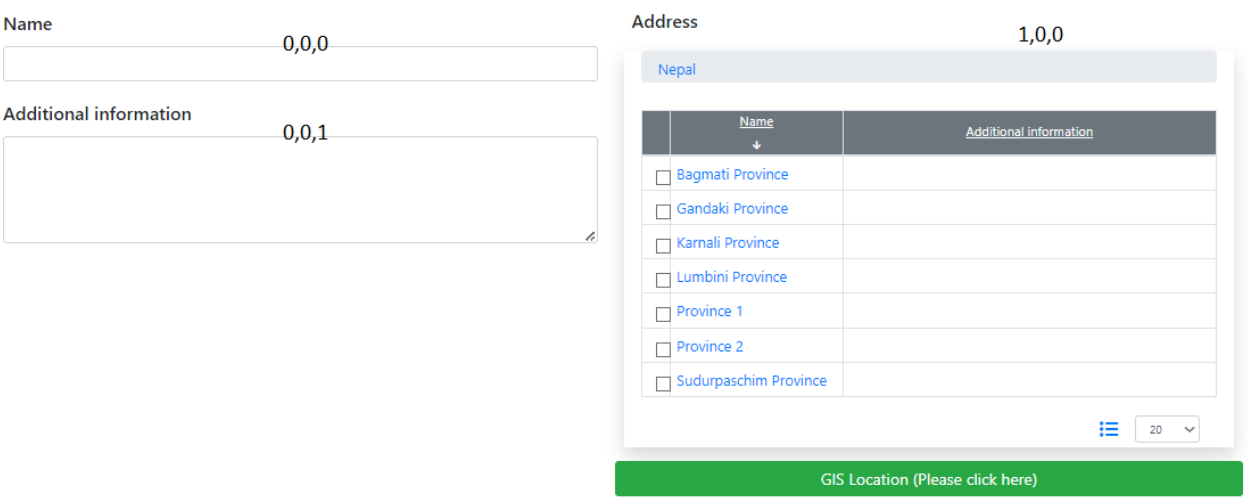


Figure 23. Cell and order inside a cell are acceptable

OpenRIMS layout editor allows the define and re-arrange components easily using the following rules:

- Columns number possible only 0 and 1
- Values of rows and orders allowed up to 89. Row number 100 is reserved for “things”

When the user tries to place a component in a busy position inside a cell, the rest of the components in this cell will be shifted down

The component's configuration reference

addresses

Address dictionary with the possibility to select geographical coordinates

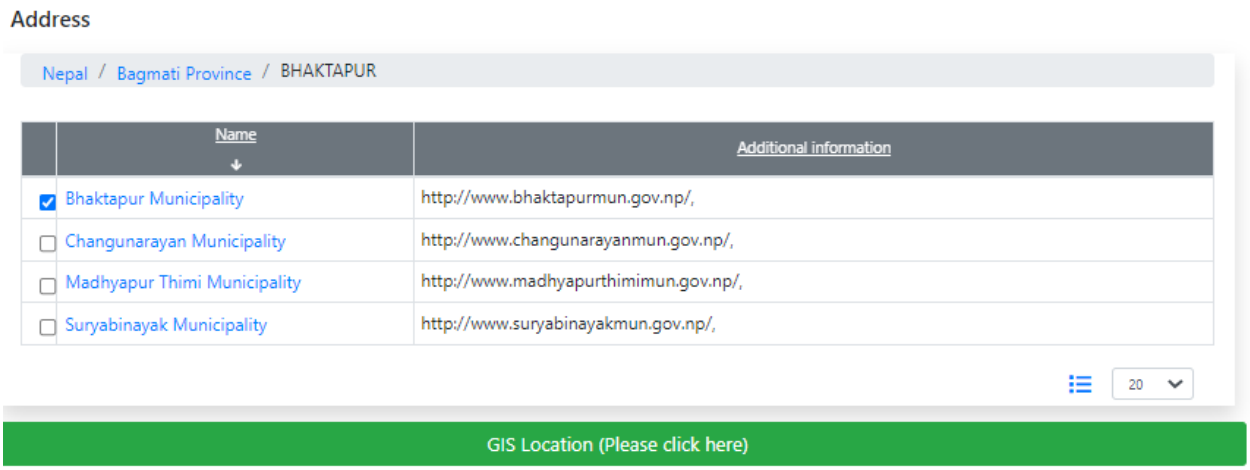


Figure 24. The address controls

Parameter	Usage	Example
Hide from an applicant	Not in use	
Minimal Offset		
Maximal Offset		
required	selection and coordinates are required	
mult		
Auxiliary Data		
url	where to store and find	nepal.address
dictUrl		
auxURL		
readOnly	value is read-only	
unique		
Name		
The pattern for the field or file names		

atc

ATC codes for medicines

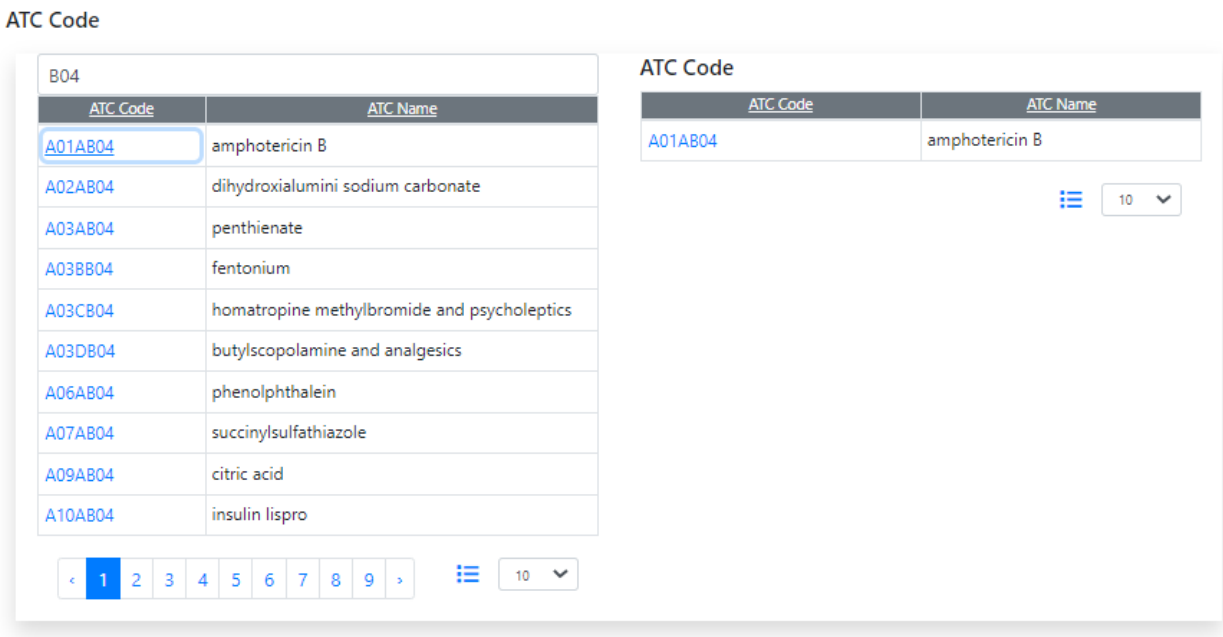


Figure 25. The ATC codes component

Parameter	Usage	Example
Hide from an applicant	Not in use	
Minimal Offset		
Maximal Offset		
required	Require at least one code selected	
mult		
Auxiliary Data		
url	Where to store the selected codes?	medicinalproduct.productclassification
dictUrl	Where to find ATC codes	who.atc.human
auxURL		
readOnly	Display date only	
unique		
Name		
The pattern for the field or file names		

dates

Date input control

Parameter	Usage	Example
Hide from an applicant	Hide dates from an applicant	true
Minimal Offset	months from the current date	-3
Maximal Offset	months after the current date	3
required	value is required	
mult		
Auxiliary Data		
url		
dictUrl		
auxURL		
readOnly	value is read-only	true
unique		
Name		
The pattern for the field or file names		

dictionaries

Allows selection from the pre-defined choices. For example

System of Medicine

Product System Category Hierarchical / Allopathy

	Name	Additional information
☑	GA	The drugs under category "GA" may be sold by any seller on the basis of experience and even without the prescription of doctor and the presence of a pharmacist or a professional shall not be compulsory while selling the drug.
☐	KA	Category "KA" consists of narcotic and poisonous drugs and category. The drugs under this category shall be sold only on the prescription of a doctor and these drugs shall be sold by a pharmacist or professional ownself or only in the presence of any one out of either a pharmacist or a professional.
☐	KHA	Category "KHA" consists Antibiotics, Hormones, etc. The drugs under this category shall be sold only on the prescription of a doctor and these drugs shall be sold by a pharmacist or professional ownself or only in the presence of any one out of either a pharmacist or a professional.

20

Figure 26. The dictionary

Parameter	Usage	Example
Hide from an applicant	The dictionary is available only to NMRA staff.	true
Minimal Offset		
Maximal Offset		
required	at least one choice is required	
mult	It is possible to select more than one value on the last level of a dictionary	false
Auxiliary Data		
url	url of the dictionary	dictionary.guest.deregistration
dictUrl		
auxURL		
readOnly	value is read-only	
unique		
Name	For single-choice dictionaries copy values from the selected prefLabel and description to form`s literal prefLabel and description if ones are existing	
The pattern for the field or file names		

documents

Documents that have been uploaded by a user.

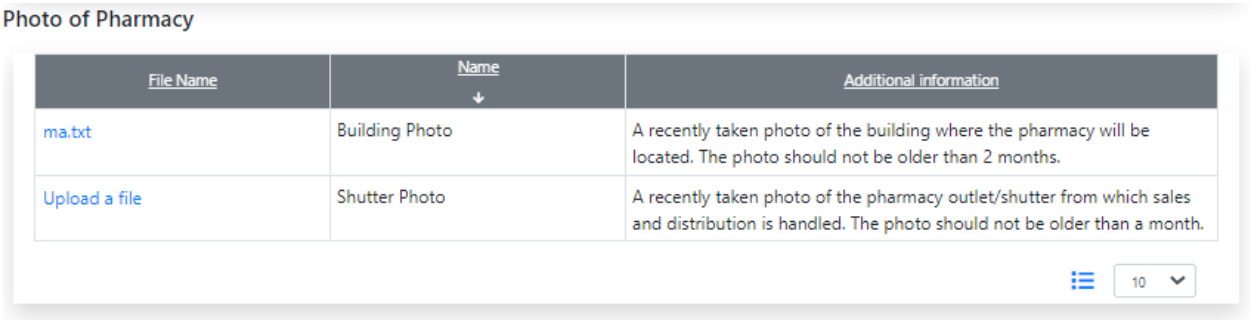


Figure 27. The file uploader component

Parameter	Usage	Example
Hide from an applicant	The dictionary is available only to NMRA staff.	true
Minimal Offset	wide of the image[2] in pixels.Not mandatory	300
Maximal Offset	height of the image in pixels. Not mandatory	300
required	at least one file should be uploaded	
mult		
Auxiliary Data		
url	where to store and find	pharmacy.documents
dictUrl	dictionary with the list of files to upload descriptions	dictionary.pharmacy.site.photos
auxURL		
readOnly	value is read-only	
unique		
Name		
The pattern for the field or file names	file types allowed[3]	.jpg,.png,.tiff

[2] In case if jpg, jpeg or png file will be uploaded. For other types of uploaded files will be ignored
[3] https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file - see attribute “accept”

droplist

The droplist is a simplified representation of a one-level, single-choice dictionary. For example, the dictionary.business.agreements can be represented as a droplist

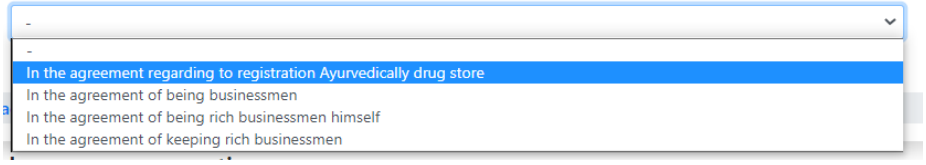


Figure 28 A droplist looks like a general-purpose dropbox

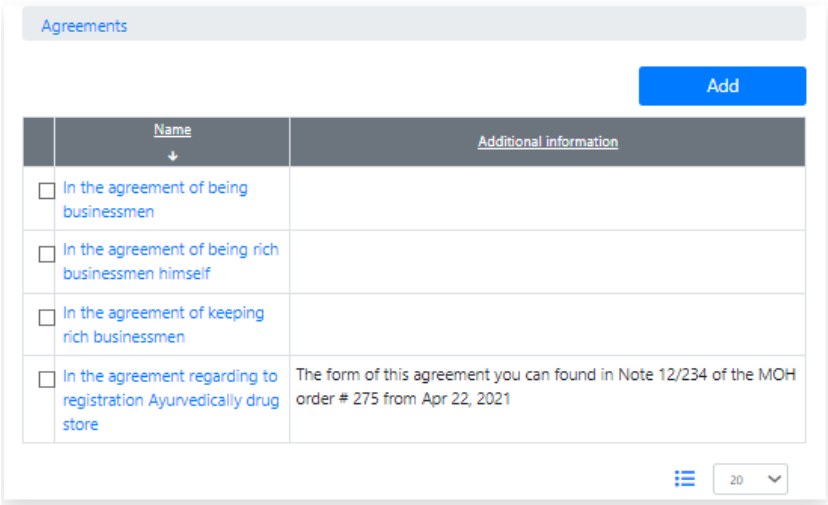
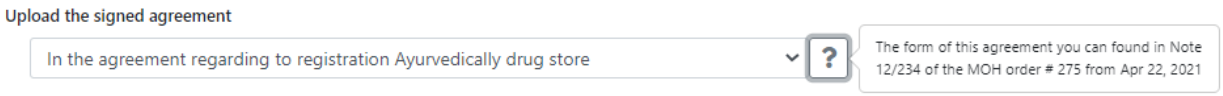


Figure 29 dictionary.business.agreements

Unlike a general-purpose dropbox, the droplist allows additional information defined by the dictionary.



Parameter	Usage	Example
Hide from an applicant	Not used	
Additional Information	Not used	
Minimal Offset	Not used	
Maximal Offset	Not used	
required	A value should be selected	True
mult	Not used	
Auxiliary Data		
url	Dictionary URL	
dictUrl	Not used	

Parameter	Usage	Example
auxURL	Not used	retail.site.owned.persons,pharmacy.retail.registered
readOnly	Display only selected value	true
unique	Not used	
Name	Allow copy selected value to prefLabel and additional information, if one, to the description	true
The pattern for the field or file names	Not used	

headings

Displays a label or a link to an external resource.

Parameter	Usage	Example
Hide from an applicant	Not used	
Minimal Offset		
Maximal Offset		
required		
mult		
Auxiliary Data		
url	Link to an external resource. Not mandatory	https://google.com
dictUrl		
auxURL		
readOnly		
unique		
Name		
The pattern for the field or file names		

intervals

An interval between two dates.

Figure 30. The interval component

Parameter	Usage	Example
Hide from an applicant	Not used	
Minimal Offset	The length of the interval in months	
Maximal Offset	The max in the interval, zero means no restrictions	
required	Minimal Offset and Maximal offset constraints should be applied	
mult		
Auxiliary Data		
url		
dictUrl		
auxURL		
readOnly		
unique		
The pattern for the field or file names		

legacy

Allows to select objects uploaded from the legacy data. Applicable for manual legacy data export, usage of the legacy data for new applications, etc.

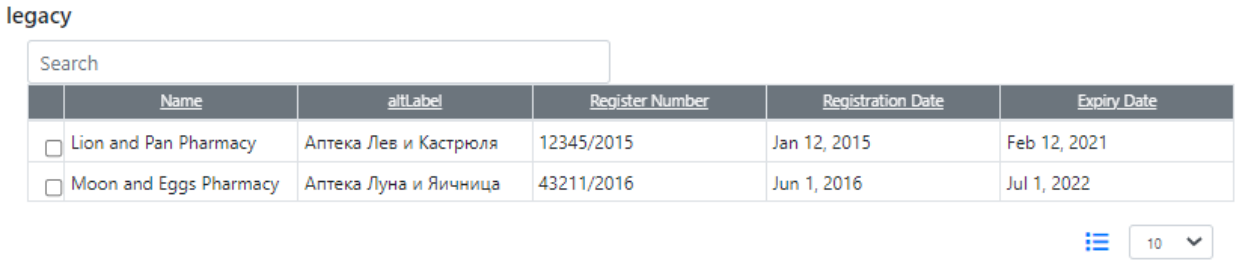


Figure 31. The legacy data component

Parameter	Usage	Example
Hide from an applicant	Not used	
Minimal Offset		
Maximal Offset		
required	Require the selection	
mult		
Auxiliary Data		
url	Where to store the selected codes?	pharmacies.registered.legacy
dictUrl	The kind of legacy data, e.g., pharmacies, medicines, etc.	legacy.pharmacies
auxURL		
readOnly	Display date only	
unique		
Name	See notes below	
The pattern for the field or file names		

Regardless of the configuration field “Name” value, such data will be copied to the form components:

1. “Name” to the literal “prefLabel”, if defined on the form
2. “altLabel” to the literal “altLabel”, if defined on the form
3. Registration date, expiration date, and register number to the register “legacyRegister”, if defined on the form

links

This component allows the usage of data from other applications. For example:

- Manufacturer's data in a medicinal product application
- Medicinal product data in an import permit application
- Facility data in a wholesaler application
- etc.

The only restriction is that the other application should be approved, but not de-registered yet.

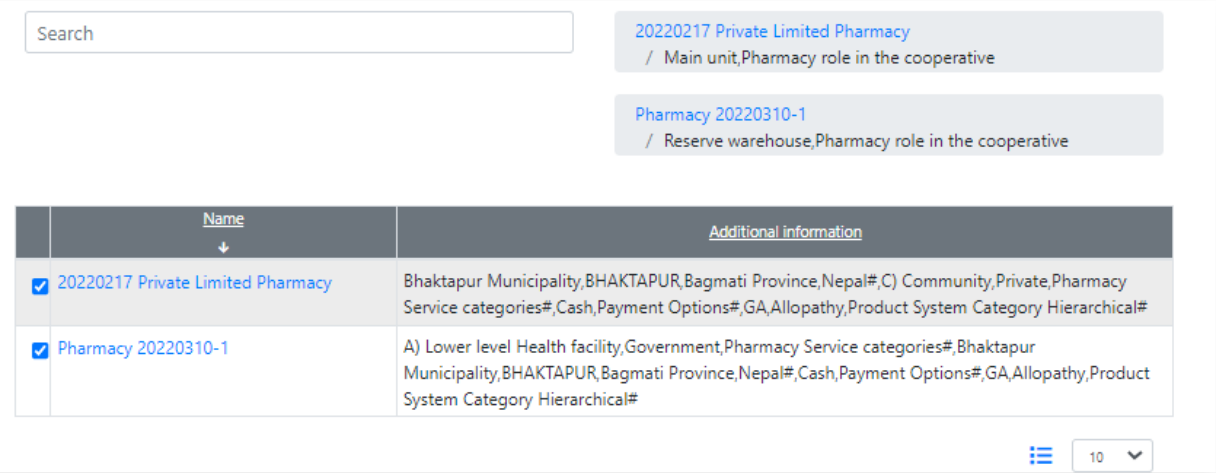


Figure 32 The data of existing facilities are used in the application

Sometimes it will be necessary to identify the data from other applications. For example, a manufacturer may be identified by the role in the manufacturer process, i.e., finished product manufacturer, packager, etc.

The component “links” allows making the identification using a single–level dictionary

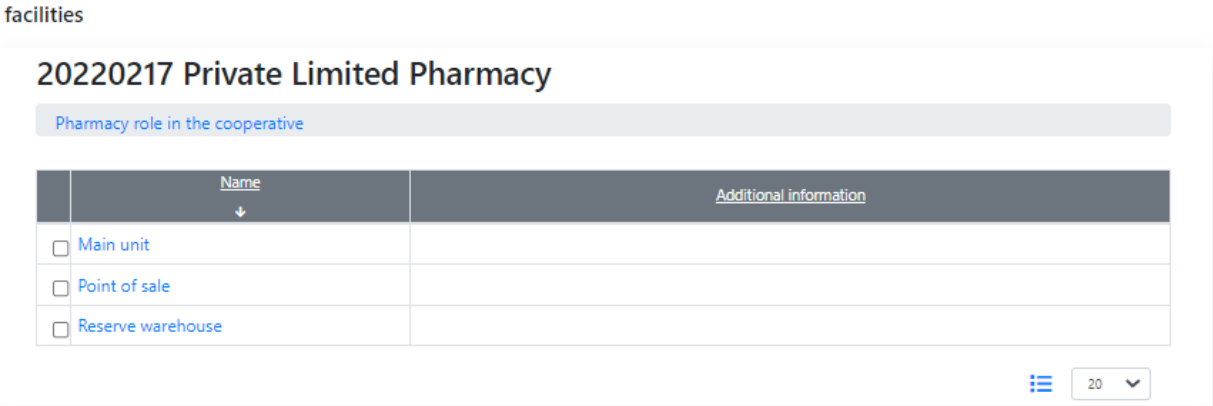


Figure 33 The dictionary allows making the identification

By default, the component “links” allows the usage of data only from applications that belong to the applicant. For example, the import permit application allows only medicinal products registered by the applicant.

It is possible to override this behavior by adding the prefix “all.” to the “url” field in the data configuration:

Parameter	Usage	Example
Hide from an applicant	Not used	
Additional Information	Help message that will be available permanently on the top. The usage of this parameter is necessary for a good usability	Select a pharmacy from the list and, then a pharmacy role in the cooperative
Minimal Offset	Not used	
Maximal Offset	Not used	
required	At least one link should be selected	True
mult	Many links may be selected	True
Auxiliary Data		
url	URL of the links to use in reports.	all.warehouses[4] warehouses
dictUrl	Dictionary to identify the links	dictionary.cooperative.pharmacy.role
auxURL	URL or comma separated a list of URLs from which objects to link will be selected.	retail.site.owned.persons,pharmacy.retail.registered
readOnly	Will display only selected links along with the classifiers, if one	False
unique	Not used	
Name	Copy selected value to prefLabel and description	See Table 1 How selected value will be copied to literals if corresponding literals are defined in the form
The pattern for the field or file names	Not used	

Table 1 How selected value will be copied to literals if corresponding literals are defined in the form

	Other references
prefLabel	prefLabel of the linked object
description	Description of the linked object

[4] By default, selection allowed only from objects belongs to a current user. For example, medicinal product registered by the user. However, if URL starts with “all.” the selection will be from all registered objects.

literals and strings

The text input field. The “strings” keep the same text for all languages, and “literals” allow distinct text for each language.

Parameter	Usage	Example
Hide from an applicant	Hide test fields from an applicant	true
Minimal Offset	min length in characters	3
Maximal Offset	max length in characters	1024
required	value is required	
mult		
Auxiliary Data		
url		
dictUrl		
auxURL		
readOnly	value is read-only	
unique	the value should be unique	
Name		
The pattern for the field or file names	Java Regular expression pattern. The most used patterns and details are here [String validation patterns]	[a-zA-Z]{3}

logical

Yes/No/NA logical field

Parameter	Usage	Example
Hide from an applicant	Not used	
Minimal Offset		
Maximal Offset		
required		
mult		
Auxiliary Data		
url		
dictUrl		
auxURL		
readOnly	value is read-only	
unique		
Name		
The pattern for the field or file names		

numbers

Number input field

Parameter	Usage	Example
Hide from an applicant	Not used	
Minimal Offset	min value	0
Maximal Offset	max value	1000000
required	value is required	
mult		
Auxiliary Data		
url		
dictUrl		
auxURL		
readOnly	value is read-only	
unique		
Name		
The pattern for the field or file names		

persons

Allows adding detailed records to the main data. Examples are pharmacy owners to a pharmacy or warehouses the wholesaler.

Figure 34. The "persons" component - detail records

Parameter	Usage	Example
Hide from an applicant	Not used	
Minimal Offset		
Maximal Offset		
required	at least one detailed record is required	
mult		
Auxiliary Data		
url	Where to store detail records	site.owners.persons
dictUrl		
auxURL		
readOnly		
unique		
Name		
The pattern for the field or file names		

registers

Provides assigning numbers, registration dates, and expiration dates like a typical filing system register.

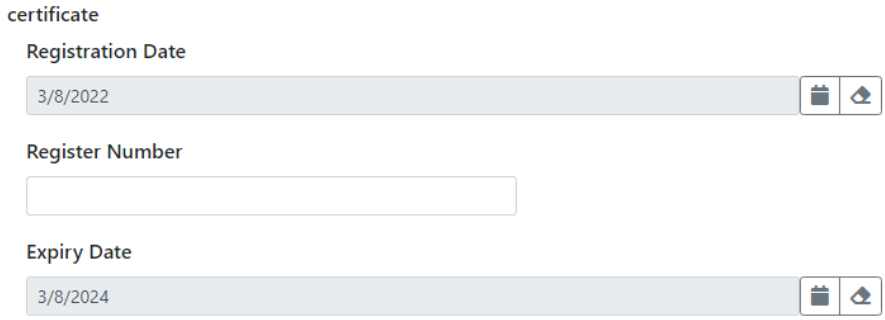


Figure 35. The register component

Parameter	Usage	Example
Minimal Offset	Offset in months from the current date to the minimal possible registration date	-1
Maximal Offset	Offset in months from the current date to the maximal possible expiration date	
required	the dates should fit the Minimal offset – Maximal offset criteria	
mult	If true, the expiration date will be available. Otherwise, the registration date only.	
Auxiliary Data		
url	Where to store the register?	pharmacy.site.renew
dictUrl		
auxURL		
readOnly	Display date only	
unique		
Name		
The pattern for the field or file names	The registration number prefix. For most filing systems – the code of cases	MR/

resources

Download documents, pictures, and templates.

Invoice

File Name	Name	Additional information
owner_image.docx	1) If capital is up to 100,000	
Receipt for Capital of 1,00,001 to 5,00,000.docx	2) If capital is 100,001 to 500,000	
Receipt for Capital above 5,00,001.docx	3) if capital is above 500,000	

10

Figure 36. The file resource component

Parameter	Usage	Example
Minimal Offset		
Maximal Offset		
required		
mult		
Auxiliary Data		
url	Resource configuration URL	res.pharmacy.new.invoices
dictUrl		
auxURL		
readOnly		
unique		
Name		
The pattern for the field or file names		

schedulers

Allows to run another follow-up application after the application will be approved. Examples are “scheduled renewal payment”, “scheduled inspection”, etc.

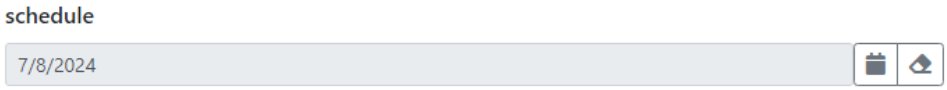


Figure 37. The scheduler component

Parameter	Usage	Example
Minimal Offset	Offset in months from the current date to the minimal possible date to run the follow-up application	
Maximal Offset	Offset in months from the current date to the maximal possible date to run the follow-up application	
required	the scheduled date should fit the Minimal offset – Maximal offset criteria	
mult		
Auxiliary Data		
url	Where to store the scheduler?	pharmacy.site.renew
dictUrl		
auxURL	Which application should be scheduled?	application.pharmacy.renew
readOnly	Display only	
unique		
Name		
The pattern for the field or file names		

things

A sequency of electronic forms that placed at the right of the main form

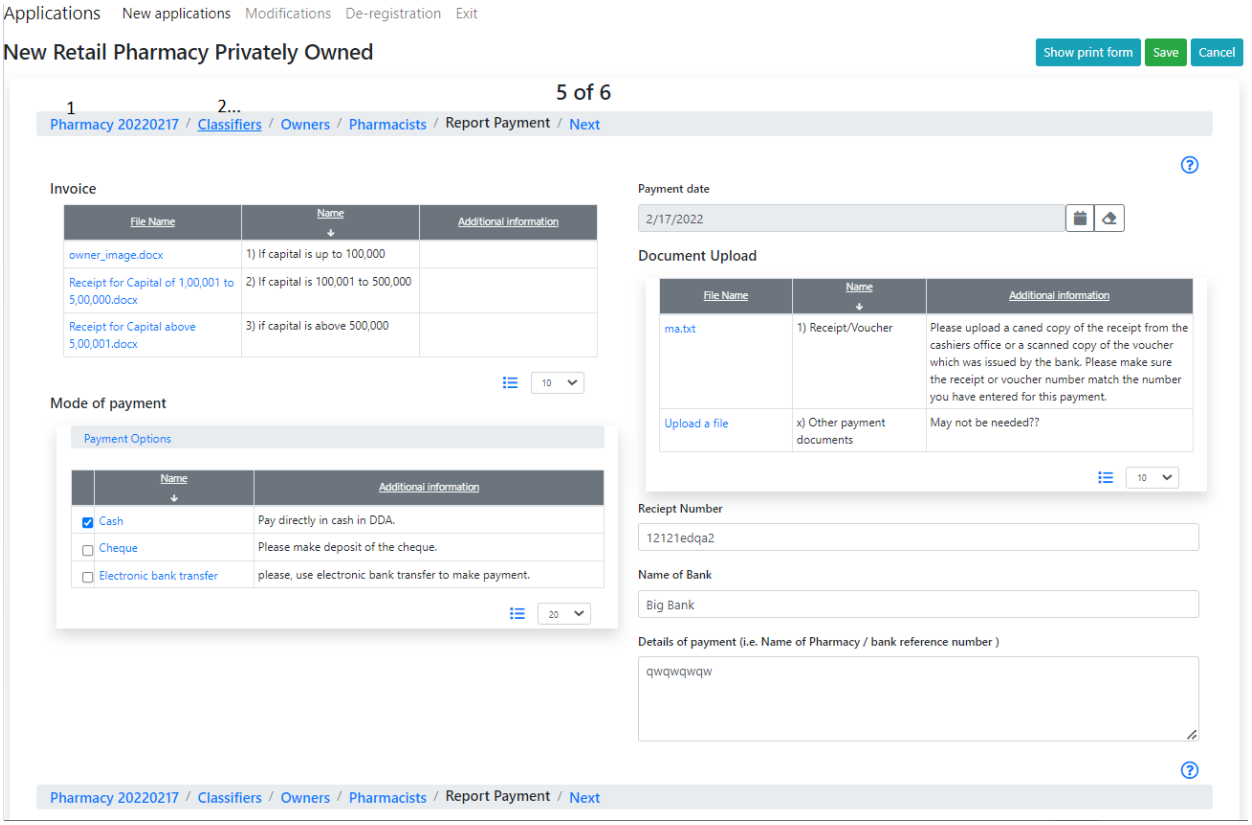


Figure 38. The main form and sub-forms

1. The main form. This form must contain a literal with the name prefLabel.
2. Additional forms that should be included using the “thing” class

Parameter	Usage	Example
Minimal Offset		
Maximal Offset		
required		
mult		
Auxiliary Data		
url	URL of an electronic form configuration. This form should not contain “things”, however, may contain any other component	retail.pharmacy.classifiers
dictUrl		
auxURL	In case when this electronic form configuration contains class [persons] .The URL of a configuration of the detail record electronic form.	site.owner.person
readOnly		
unique		
Name		

Parameter	Usage	Example
The pattern for the field or file names		

String validation patterns

For “literals” and “strings” components it is possible to assign a pattern that the user’s input should match. For example, pattern **pharmacy** means that the input text should contain the word **pharmacy** anywhere.

On https://www.vogella.com/tutorials/JavaRegularExpressions/article.html you can find a detailed description of this approach.

The most common examples are:

Phone number:

^\+\d{1,3}\(\d{1,3}\)\d{3}\-\d{2}\-\d{2}|^\+\d{1,3}\(\d{1,3}\)\d{7}

Will match

+1(111)111-11-11

or

+1(111)1111111

Email address:

^[-a-z0-9~!$%^&*_=+}{\'?]+(\.[-a-z0-9~!$%^&*_=+}{\'?]+)*@([a-z0-9_][-a-z0-9_]*(\.[-a-z0-9_]+)*\.(aero|arpa|biz|com|coop|edu|gov|info|int|mil|museum|name|net|org|pro|travel|mobi|[a-z][a-z])|([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}))(:[0-9]{1,5})?$

Will match

nobody.real.never.man@neverland.com.ua

or, we believe, will match any other email address in this format

The https://regex101.com/ resource will help to build your patterns. Please, do not forget to select Java as a language.