

School.AR

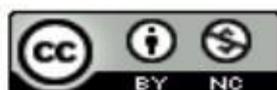
User documentation

Author: Bojan Kraut, Gerhilde Meissl-Egghart

Developed and presented as part of the School:AR project of Verein Offenes Lernen (<http://o-le.org>) funded by netidee (<https://www.netidee.at/>; project number 2052). Follow the project on ar.o-le.org.



Licence: Creative Commons



Content

1. Intro.....	3
2. Web Admin.....	4
2.1. Login.....	4
2.2. Users.....	5
2.3. Countries.....	6
2.4. Postal Codes.....	7
2.5. Catalogue Types.....	8
2.6. Catalogues.....	9
2.7. Publishers.....	10
2.8. Targets.....	11
2.9. Bookmarks.....	15
3. School.AR Mobile App.....	16
3.1. Login.....	16
3.2. Register.....	17
3.3. Main.....	18
3.4. Scanning targets.....	19
3.5. Bookmarks.....	20
3.6. Help.....	21
Appendix A - json-definitions.....	22
Available field types.....	23
Available attributes.....	23
checking correctness.....	24

<u>version</u>	<u>date</u>	<u>content</u>
1.0	Oct 2017	initial version by Gerhilde and Bojan
1.1	May 2018	update description of json-editing feature by Gerhilde

1. Intro

Printed books are still widely used in every day teaching. Their problem, however, is, that they can never include all the content that a teacher would like to have included. This is partly due to the length of the production cycle, partly to the constant growth of online resources.

School.AR enables the teacher to associate certain parts of the book ("targets") with online resources (e.g. links, videos, 3D-models). These targets are pictures or other parts that uniquely identify the page of the book. Typically targets are organised in catalogues and are connected with books-

School.AR is composed of 2 parts:

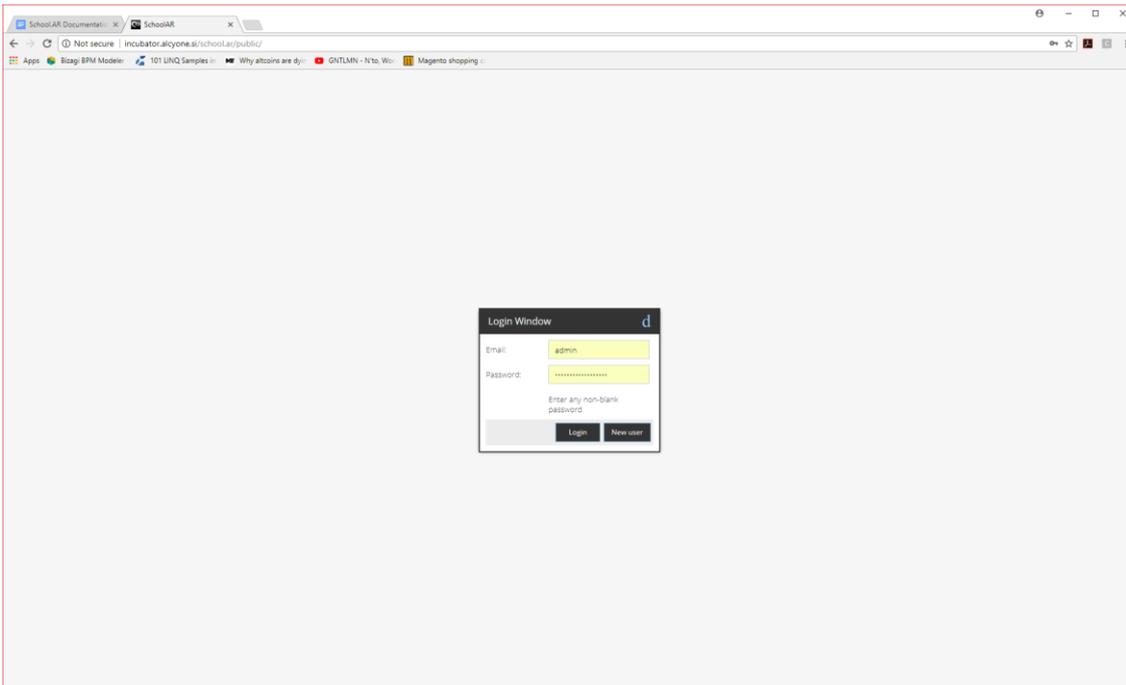
- Web Admin interface: a web interface, where teachers define targets (and links) and connect them to books
- Mobile App: an app, that is used by students who want to use the additional content: The app scans the page of a book (the target) and shows available links.

2. Web Admin

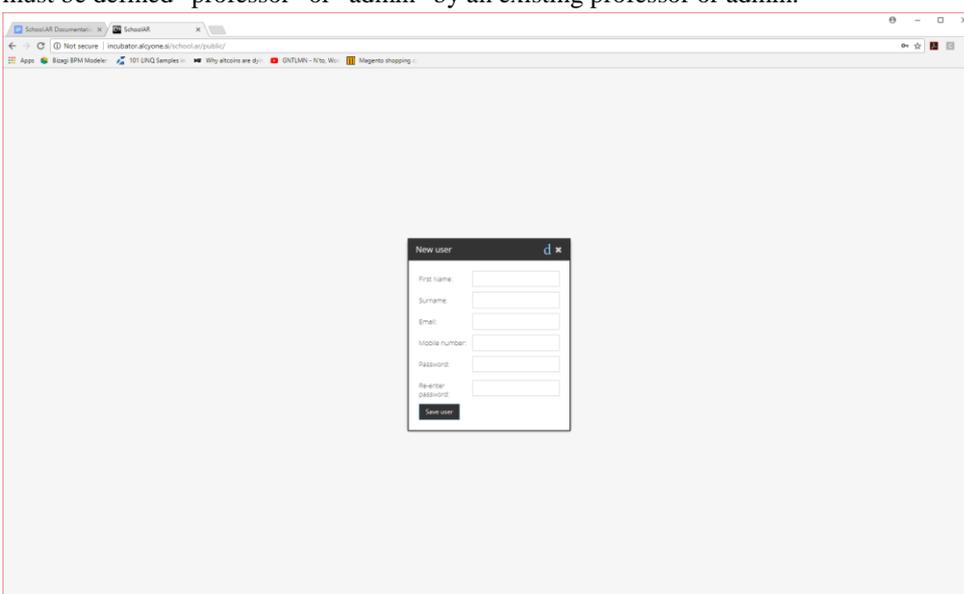
Web admin is an administration tool, where the teacher can insert content for the mobile application over the web interface. The content has to be prepared locally and later uploaded into the system, where it is served to the mobile application.

2.1. Login

This is the first screen where a web user can log in. Only users with roles "professor" and "administrator" can log in. (Normal students can not).

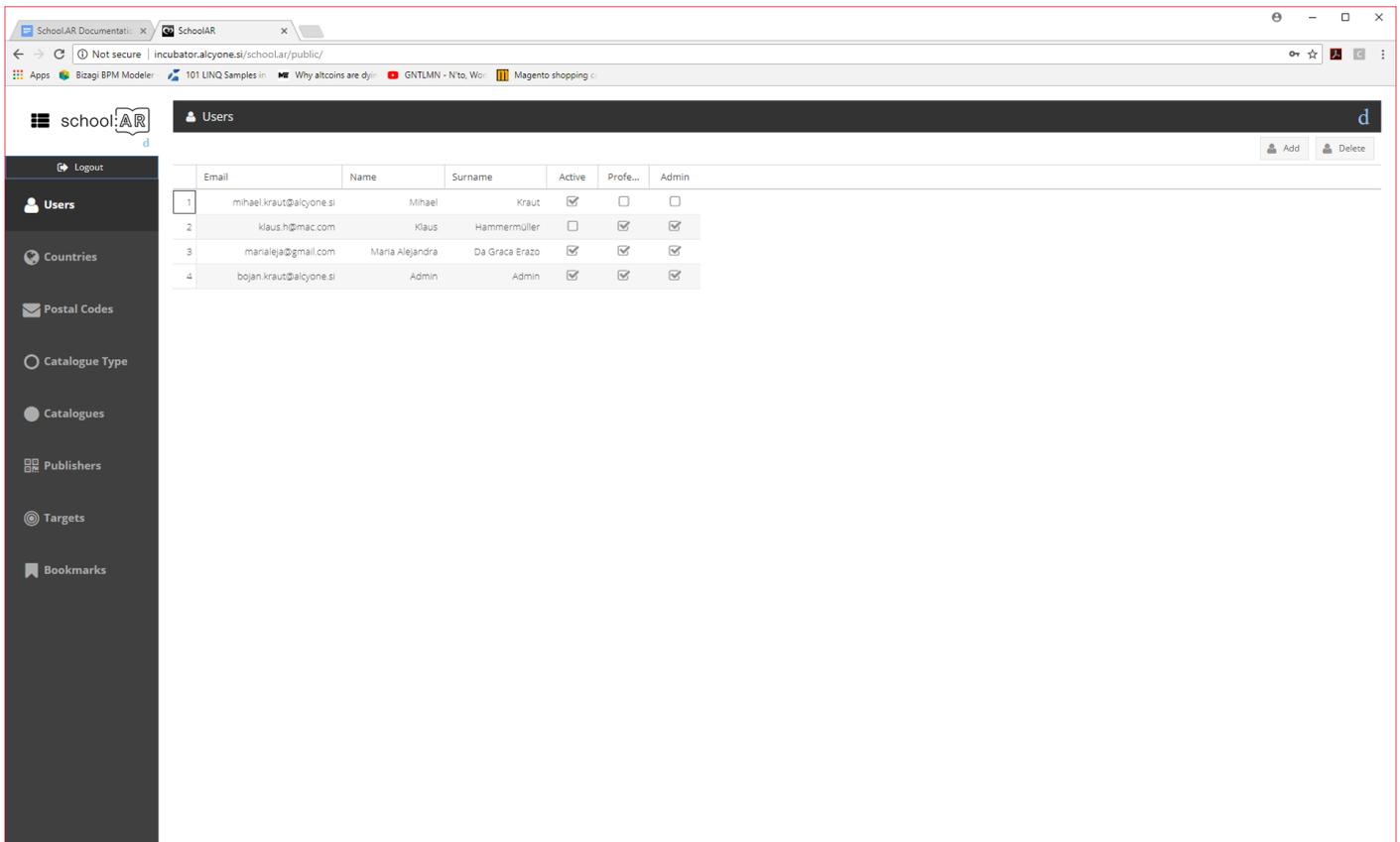


A new user can register via the registration form. In order to be able to log in, the user must confirm a confirmation email and must be defined "professor" or "admin" by an existing professor or admin.



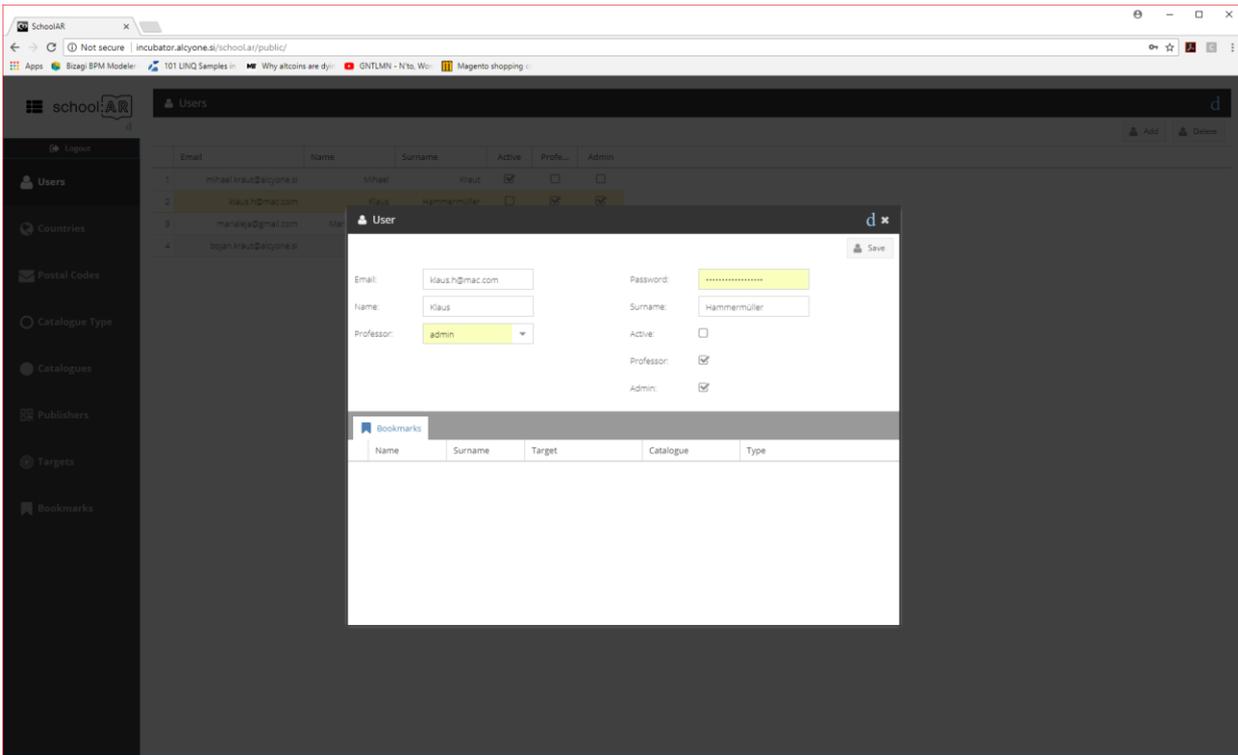
2.2. Users

The view lists a set of users.



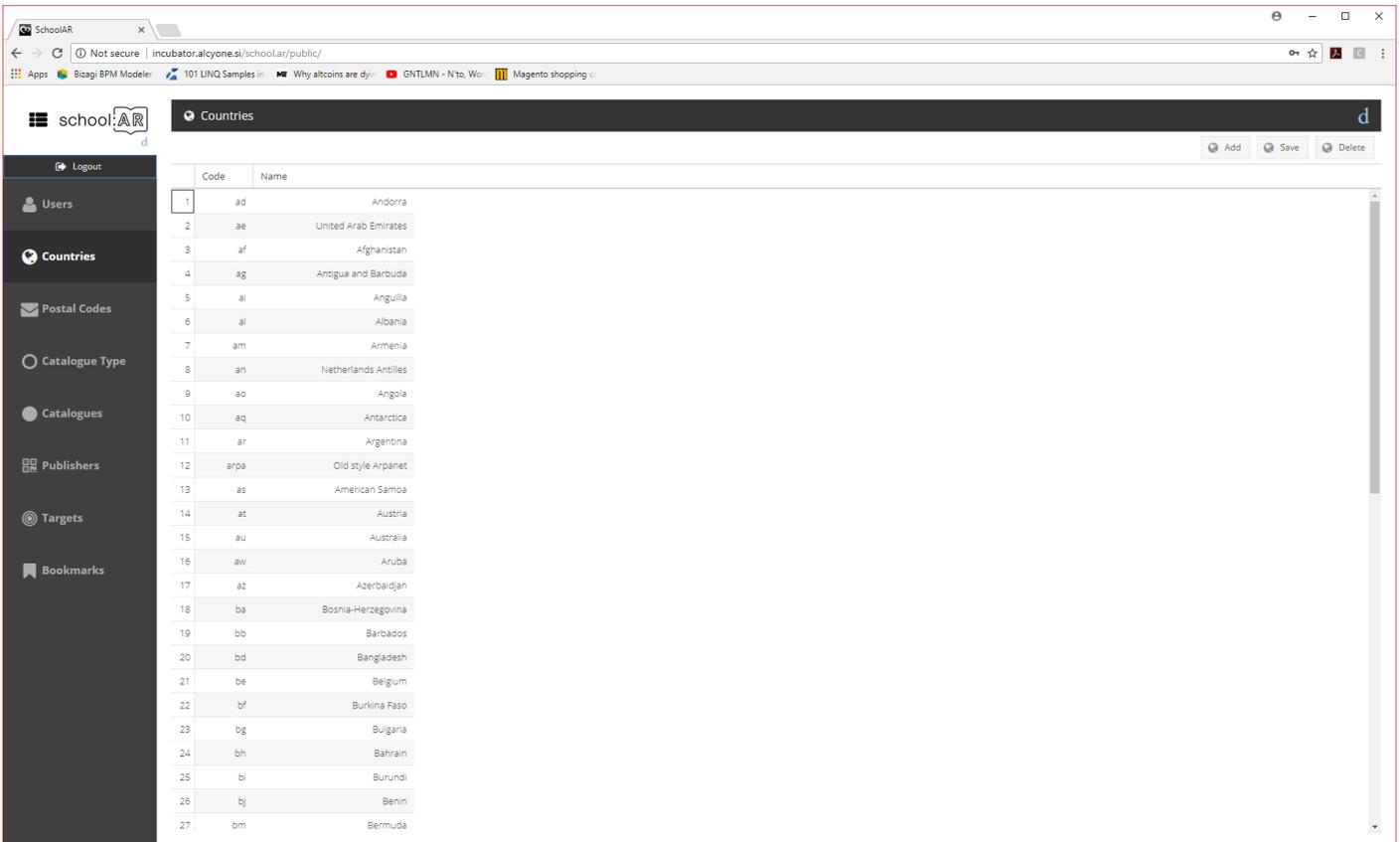
	Email	Name	Surname	Active	Profe...	Admin
1	mihael.kraut@alcyone.si	Mihael	Kraut	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	klaus.h@imac.com	Klaus	Hammermüller	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	manialeja@gmail.com	Maria Alejandra	Da Graca Erazo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	bojan.kraut@alcyone.si	Admin	Admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

After clicking a row on the grid the form pops out, where user data can be edited. Students must get assigned a professor in order to be able to access a professor's content. Below the form, the user's bookmarks are displayed. (A bookmark is generated as soon as a user accessed content via the mobile app. This helps the professor to track usage of his/her additional content).



2.3. Countries

The view lists a set of countries.



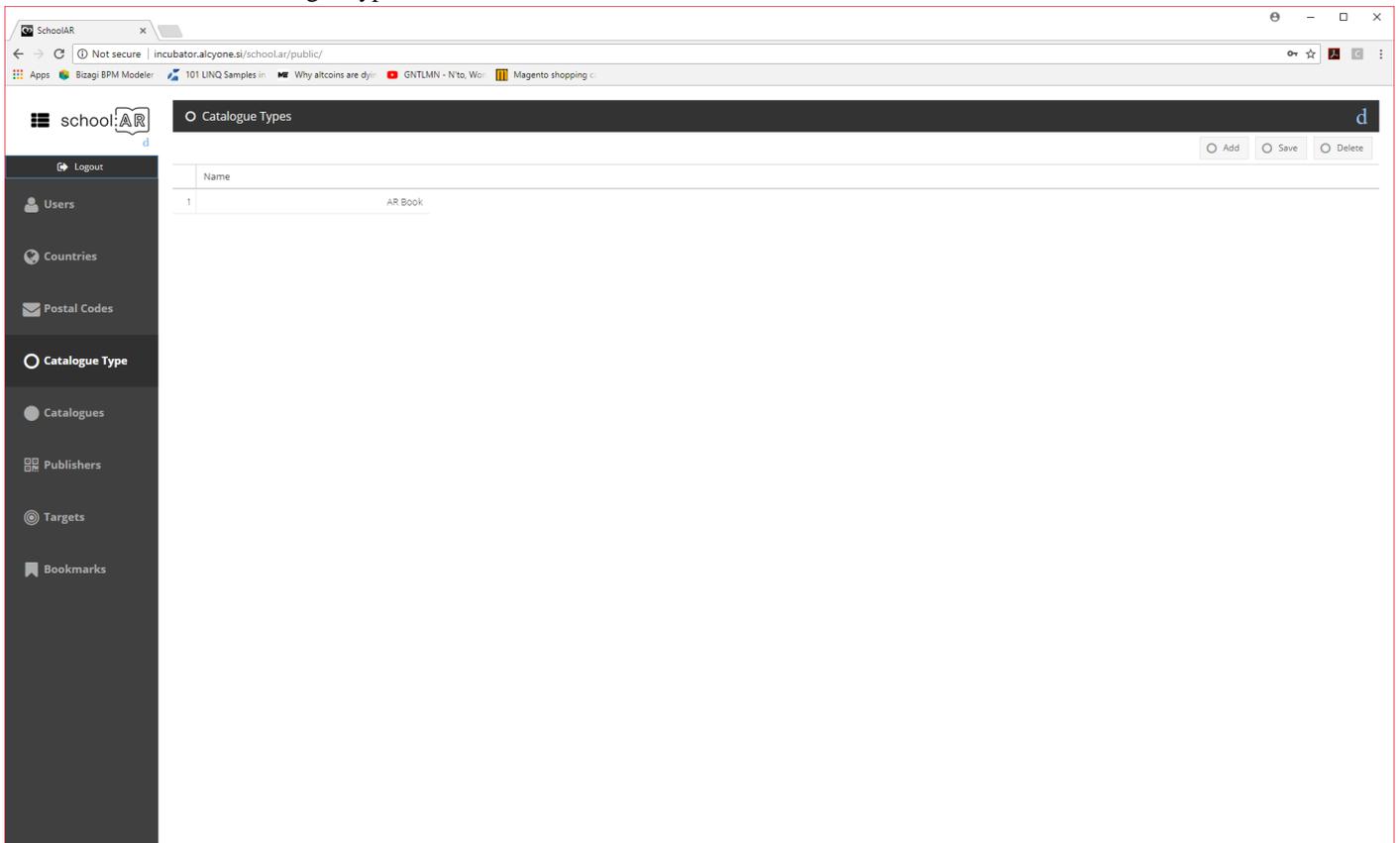
2.4. Postal Codes

The view lists a set of postal codes.

Code	Name	Country
2000	Maribor	Slovenia

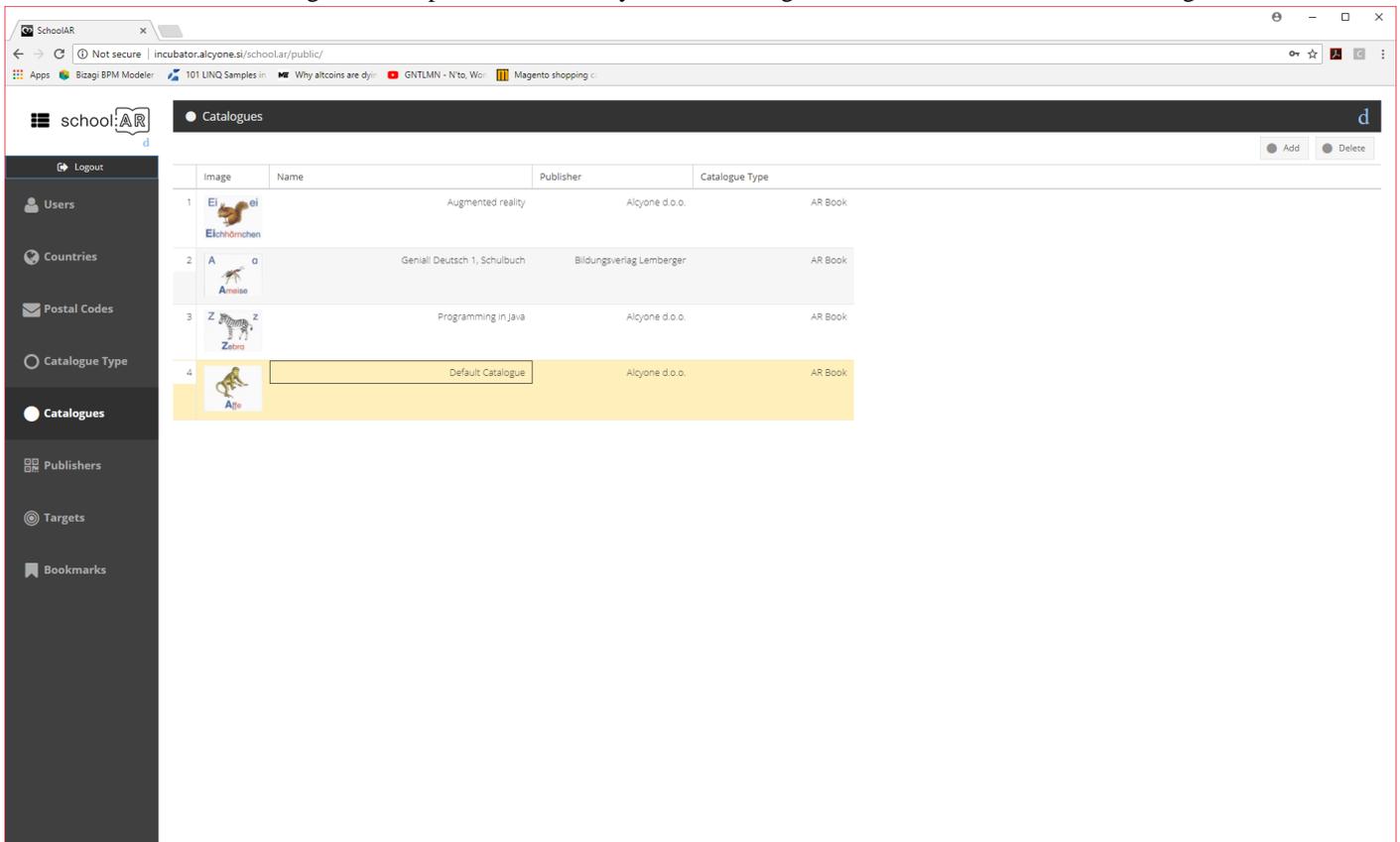
2.5. Catalogue Types

The view lists a set of catalogue types.

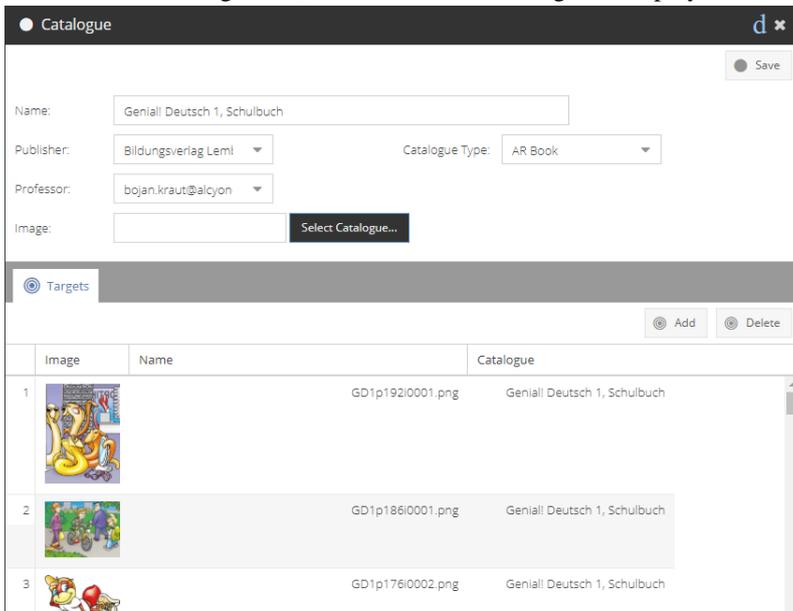


2.6. Catalogues

The view lists a set of catalogues. Each professor sees only his/her catalogues. Admins can see all of the catalogues.

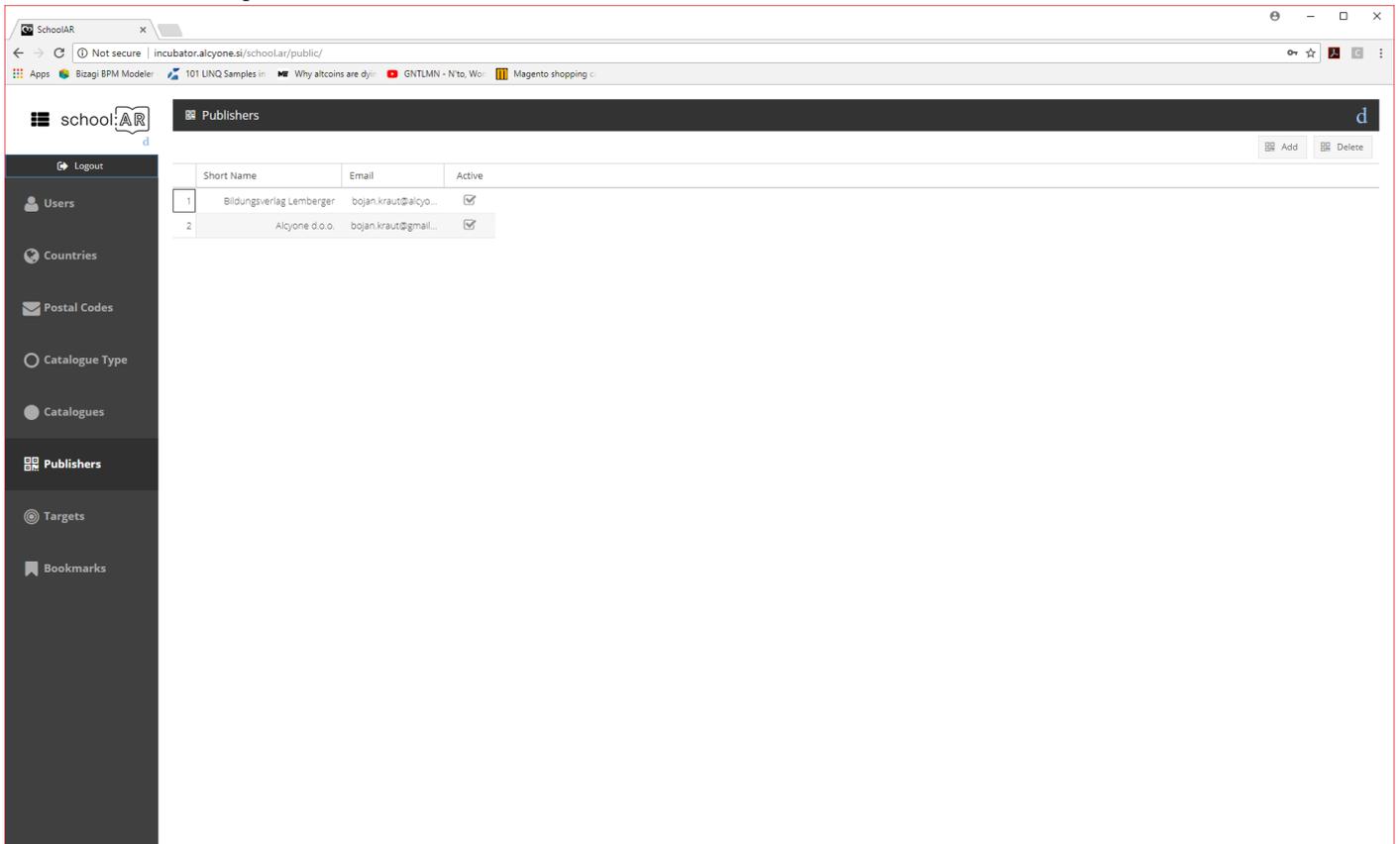


After double clicking a row, the form for the catalogue is displayed. Each catalogue can have several targets.

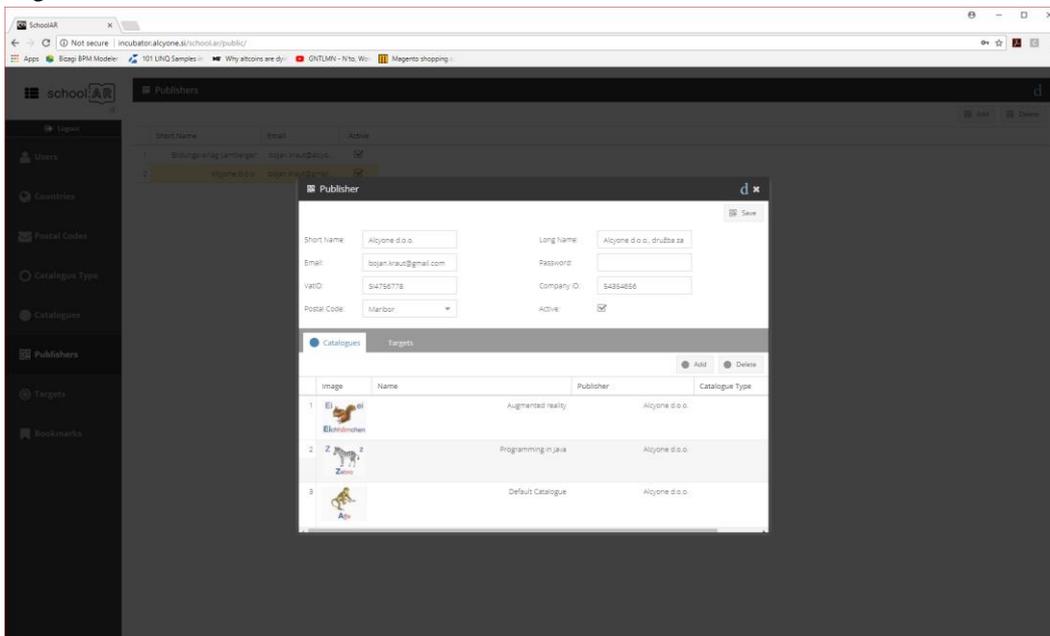


2.7. Publishers

The view lists a set of publishers.



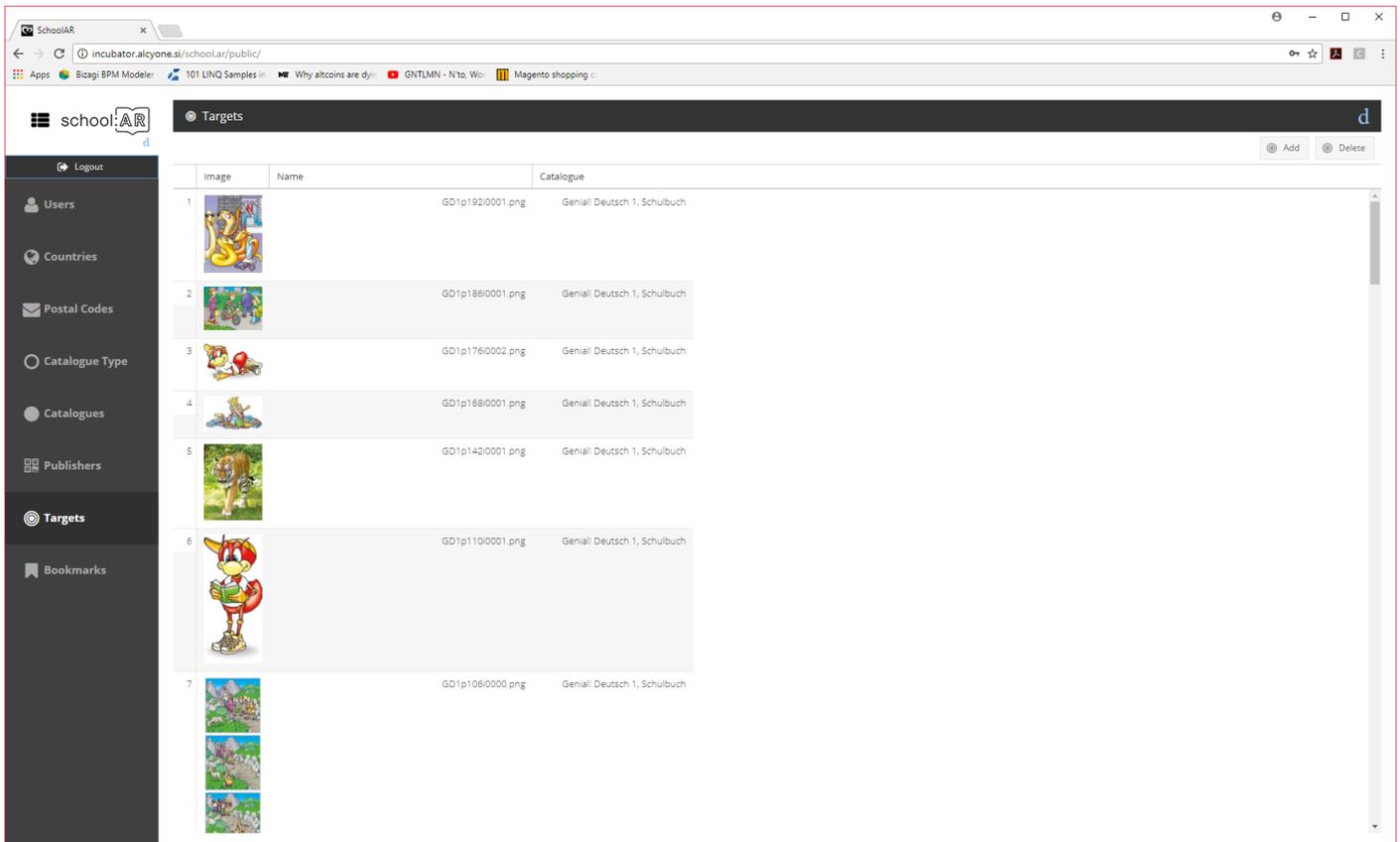
After double clicking a row, the publisher form is displayed. The first tab displays publisher catalogues. The second tab is for the targets.



2.8. Targets

These are the core functionality of our system: The user can create new targets by uploading an image and he can define additional, interactive functions that should be displayed at the link that's associated with the target.

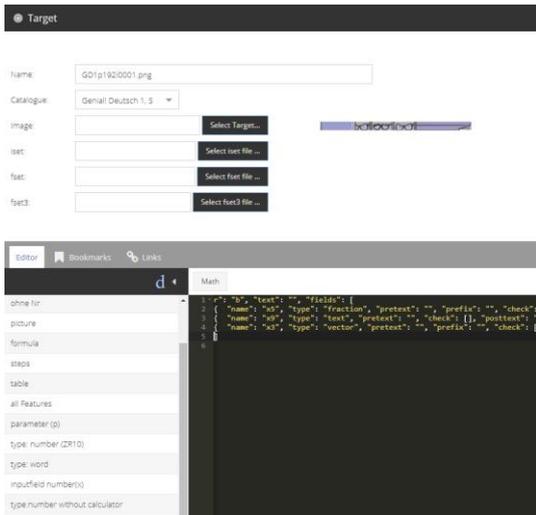
The view lists a set of targets:



The screenshot shows a web browser window displaying the 'Targets' management interface of the SchoolAR application. The interface includes a sidebar with navigation options like 'Logout', 'Users', 'Countries', 'Postal Codes', 'Catalogue Type', 'Catalogues', 'Publishers', 'Targets', and 'Bookmarks'. The main content area displays a table with the following data:

	Image	Name	Catalogue
1		GD1p192/0001.png	Geniall Deutsch 1, Schulbuch
2		GD1p186/0001.png	Geniall Deutsch 1, Schulbuch
3		GD1p176/0002.png	Geniall Deutsch 1, Schulbuch
4		GD1p168/0001.png	Geniall Deutsch 1, Schulbuch
5		GD1p142/0001.png	Geniall Deutsch 1, Schulbuch
6		GD1p110/0001.png	Geniall Deutsch 1, Schulbuch
7		GD1p106/0000.png	Geniall Deutsch 1, Schulbuch

With double clicking on the row a form for editing the target is displayed. Alternatively the ADD-Button will create a completely new target (and the DELETE-button will delete the selected).



Editable information is the name as well as the catalog and - most important - the image, that is associated with the target. (NOTE: For successful upload of the target, the target files might have to be prepared previously via the ARToolkit tools. The maximal size of a picture is 4 MB, however the recommended size is 0.5 MB.)

The lower part of the dialog shows 3 tabs: “Editor”, “Bookmarks” and “Links”.

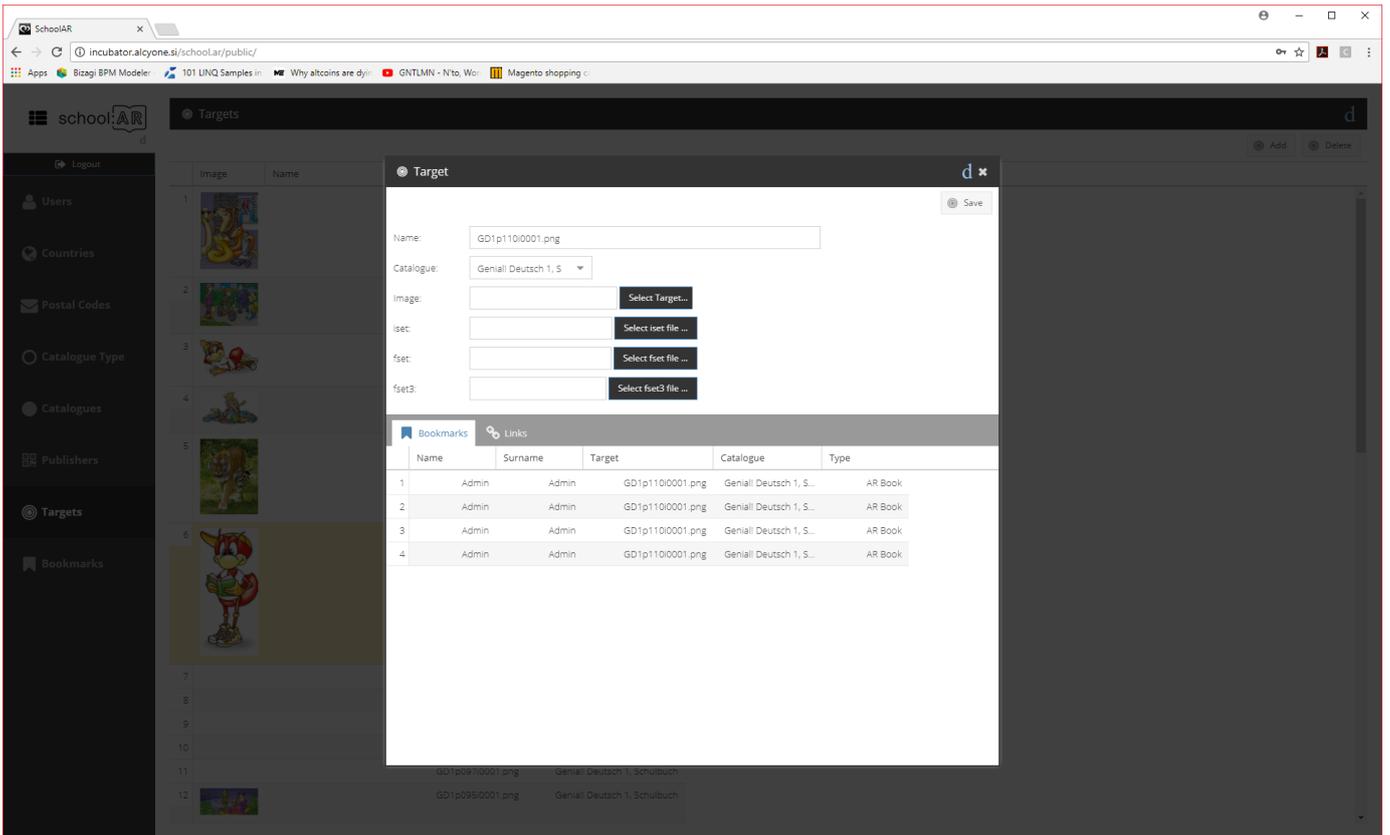
The Editor-tab allows the definition of the interactive behaviour of our link. It allows the creation of a json-file that feeds into a webpage and displays the associated links as well as further fields, e.g. 2 fields with questions, that the students should answer. Note that the correct answer is also part of the json-definition, thus the system can check automatically, if the answers were correct.

Target

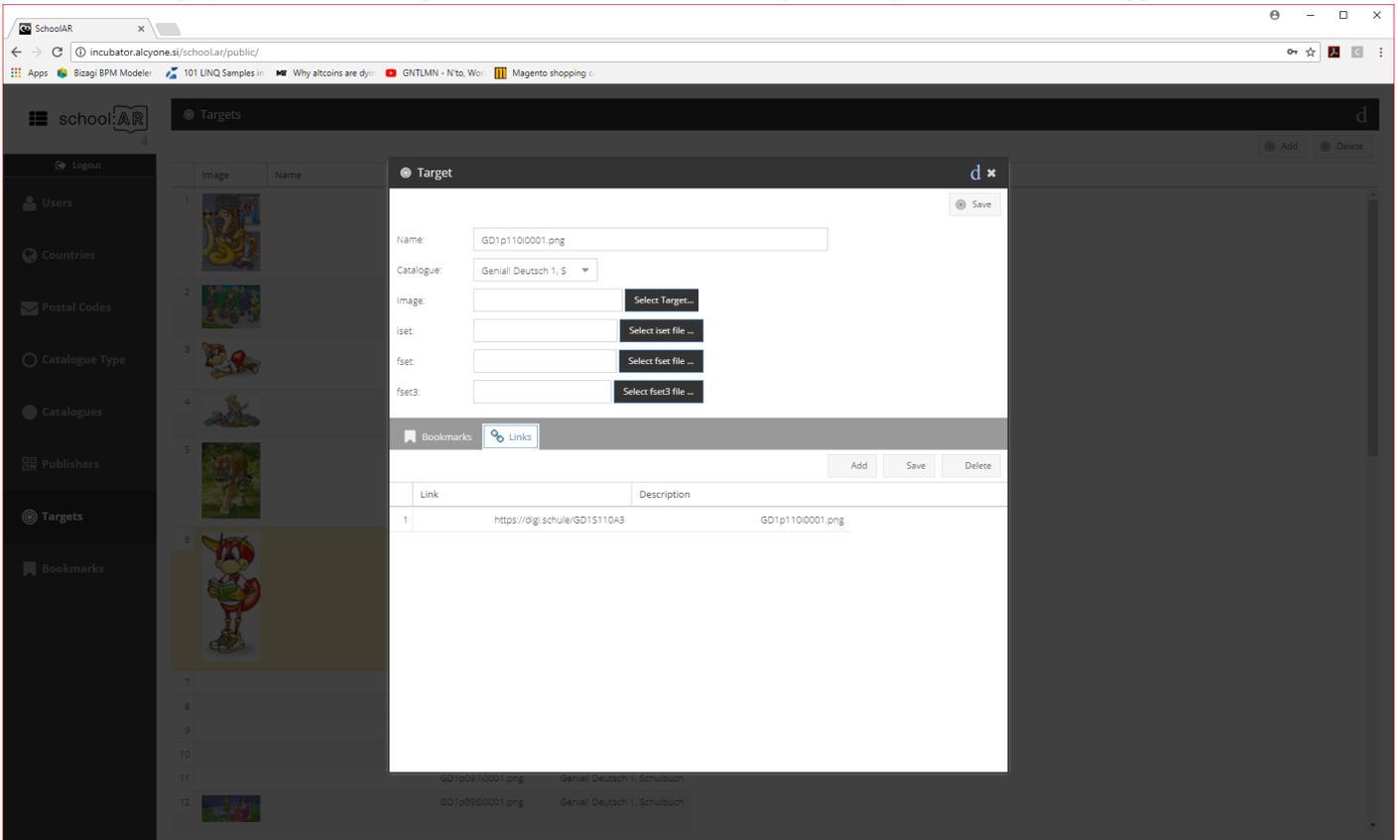
ohne Nr	1 {
picture	2 "nr": "",
formula	3 "text": "",
steps	4 "fields": [
table	5 {
all Features	6 "name": "x1",
parameter (p)	7 "type": "number",
type: number (ZR10)	8 "pretext": "Wieviele Bundesländer hat Österreich?",
type: word	9 "prefix": "",
inputfield number(x)	10 "check": [9],
type: number without calculator	11 "postfix": "",
type: set (CSV)	12 "posttext": ""
type: vector (CSV)	13 },
type: fraction(string)	14 {
type: syllable(char)	15 "name": "x2",
type: word(string)	16 "type": "statement",
type: statement(string)	17 "pretext": "Wie heißt die Hauptstadt von Österreich?",
type: text(multistring)	18 "prefix": "",
type: radiobuttons(string)	19 "check": ["Wien"],
type: checkbuttons(string)	20 "postfix": "",
	21 "posttext": ""
	22 }
	23]
	24 }

For more information about the definition of json-files please refer to Appendix A.

The second tab shows the bookmarks that are available for this target:

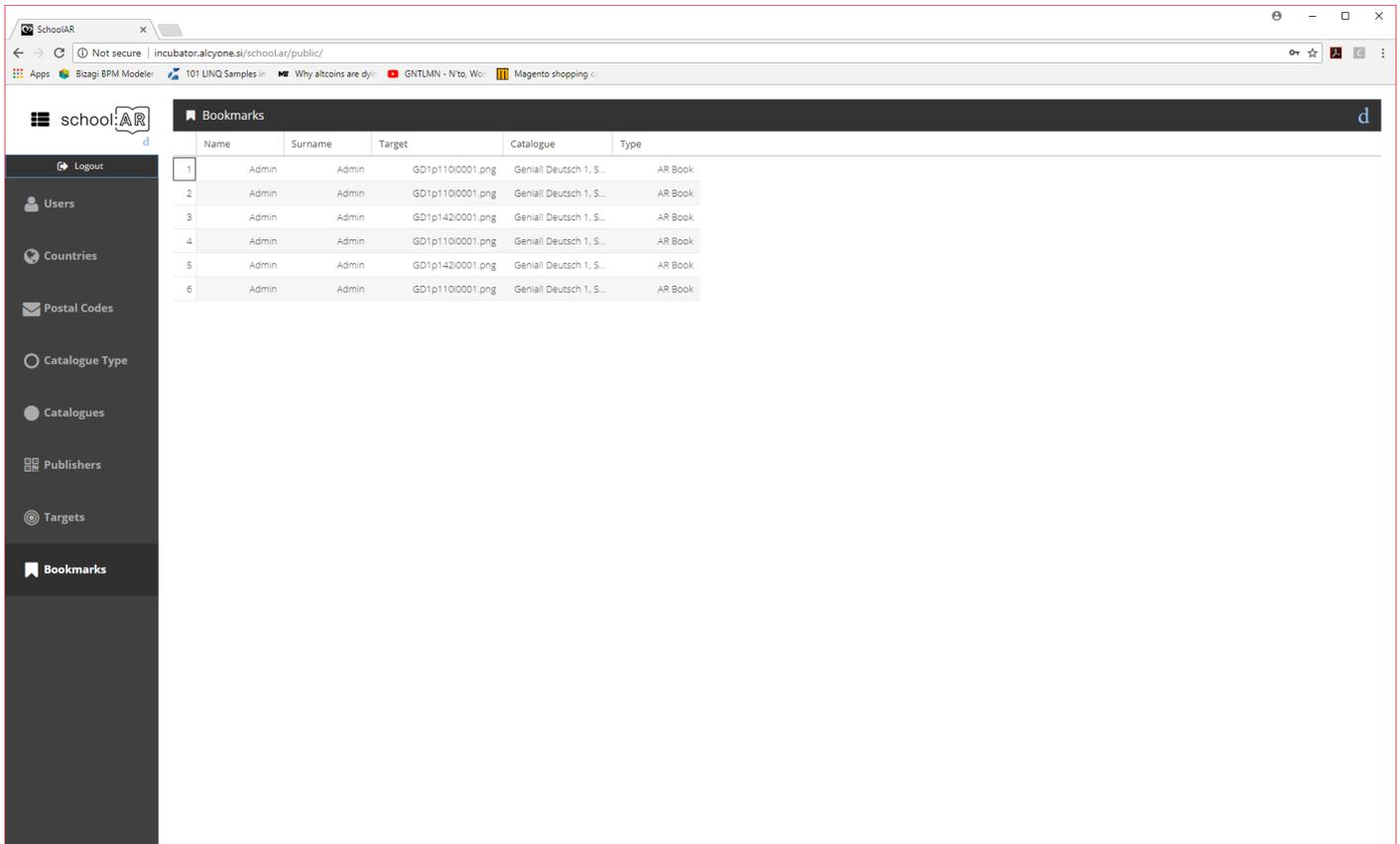


The third tab displays the links that are presented to the user when the target is recognized in the mobile application.



2.9. Bookmarks

The view lists a set all available bookmarks.



The screenshot shows a web browser window with the URL `incubator.alcyone.si/school.ar/public/`. The application interface includes a sidebar with navigation options: Logout, Users, Countries, Postal Codes, Catalogue Type, Catalogues, Publishers, Targets, and Bookmarks. The 'Bookmarks' section is active, displaying a table with the following data:

	Name	Surname	Target	Catalogue	Type
1	Admin	Admin	GD1p110I0001.png	Geniall Deutsch 1, S...	AR Book
2	Admin	Admin	GD1p110I0001.png	Geniall Deutsch 1, S...	AR Book
3	Admin	Admin	GD1p142I0001.png	Geniall Deutsch 1, S...	AR Book
4	Admin	Admin	GD1p110I0001.png	Geniall Deutsch 1, S...	AR Book
5	Admin	Admin	GD1p142I0001.png	Geniall Deutsch 1, S...	AR Book
6	Admin	Admin	GD1p110I0001.png	Geniall Deutsch 1, S...	AR Book

3. School.AR Mobile App

3.1. Login

The first page in mobile application is for the user login. With successful login the user is introduced to the system and can access the data of his/her professors.



Email

Password

LOGIN

No account yet? [Create one](#)

3.2. Register

If the user is not registered yet in the system, registration can be done via the registration form. The user created through the mobile application is a normal user, without any permissions to log into the web admin.

If the user needs to become admin or professor, his access level must be changed in the web admin tool.



Name Surname

Email

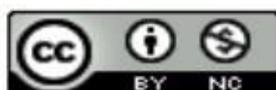
Mobile Number

Password

Re-enter Password

CREATE ACCOUNT

[Already a member? Login](#)



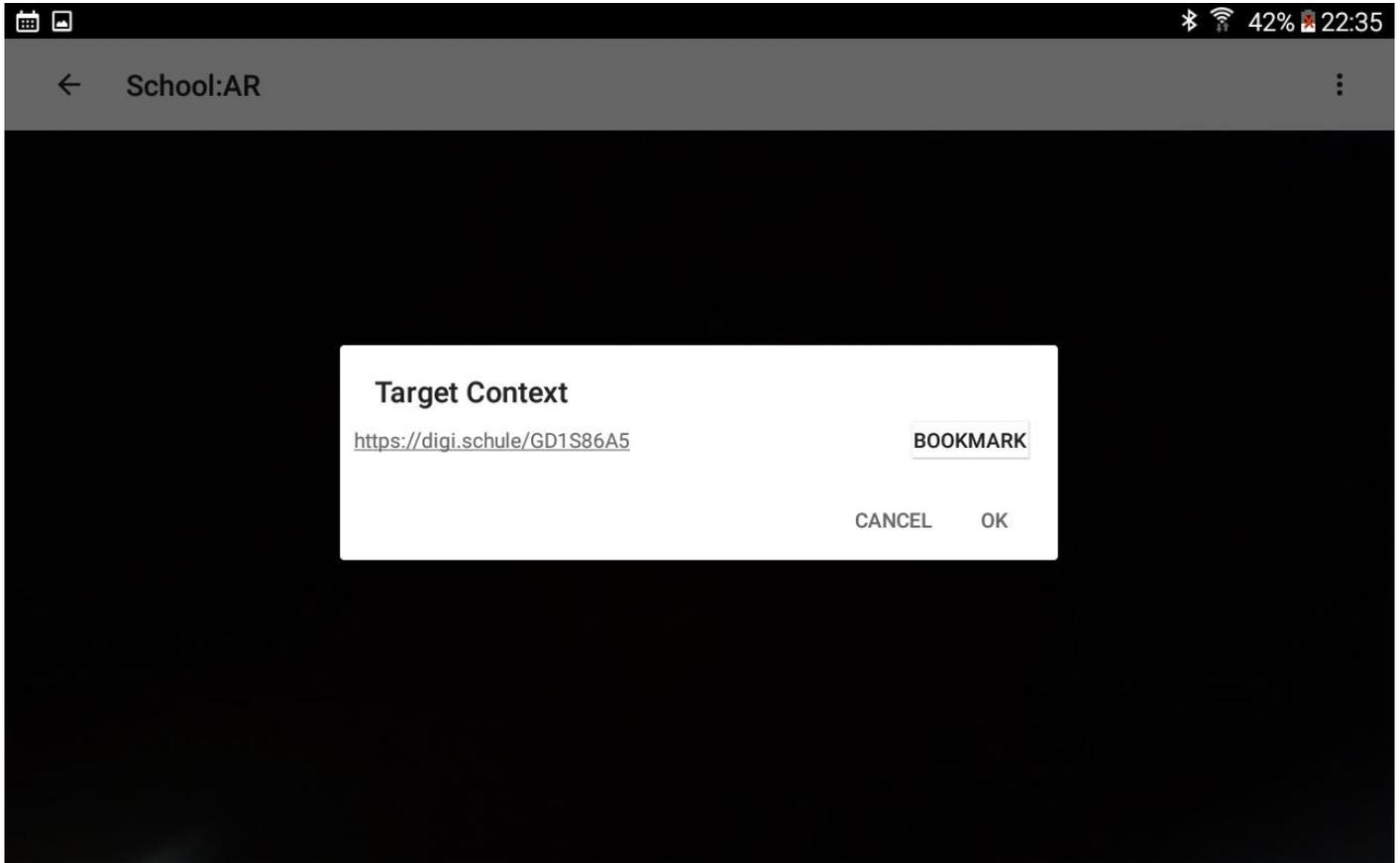
3.3. Main

After successful login catalogues are presented. Normal users can see the catalogues of his/her professor(s). Admins can see all the catalogues. To display the catalogues for the professor, the professor-fields needs to be set in the student's data.



3.4. Scanning targets

When a specific catalogue is selected the system dynamically loads the targets. After these are loaded they can be scanned. If one is recognized the popup with links is raised. A user can click on the link - which will follow the link and create a bookmark for the link.



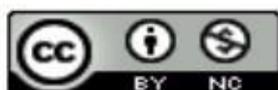
3.5. Bookmarks

When Bookmarks are selected in the menu, the table of bookmarks is displayed. On click the bookmark is displayed, similar as in scan activity.



Bookmarks	school: 
GD1p110i0001.png	Genial! Deutsch 1, Schulbuch AR Book
GD1p110i0001.png	Genial! Deutsch 1, Schulbuch AR Book
GD1p142i0001.png	Genial! Deutsch 1, Schulbuch AR Book
GD1p110i0001.png	Genial! Deutsch 1, Schulbuch AR Book
GD1p142i0001.png	Genial! Deutsch 1, Schulbuch AR Book
GD1p110i0001.png	Genial! Deutsch 1, Schulbuch AR Book

Received!



3.6. Help

This is general page for help.

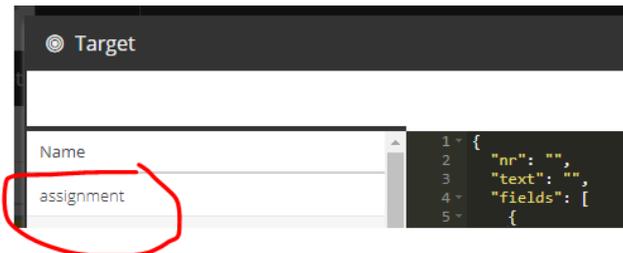


Appendix A - json-definitions

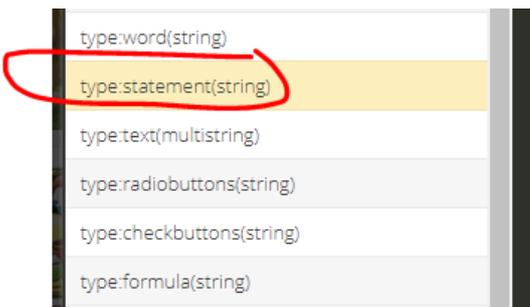
In order to define the interactive behaviour of a link, we use json-files. These can be created and edited via the list of targets.

The json-file typically consists of various elements, that are inserted by clicking the corresponding field.

The most important elements are assignments



and the fields, that are parts of the assignment. Each field has a certain type, which relates to the correct answer:



This is a very simple example:

```
1 - [
2 - {
3   "nr": "a",
4   "text": "Basiswissen",
5   "fields": [
6     {
7       "name": "x1",
8       "type": "number",
9       "pretext": "Wieviele Bundesländer hat Österreich?",
10      "prefix": "",
11      "check": [
12        9
13      ],
14      "postfix": "",
15      "posttext": ""
16    },
17    {
18      "name": "x2",
19      "type": "statement",
20      "pretext": "Wie heißt die Hauptstadt von Österreich?",
21      "prefix": "",
22      "check": [
23        "Wien"
```

We have one assignment, that is labeled with nr *a* and text *Basiswissen* as header. Then we have 2 questions about Austria. The text of the question goes into the “pretext”, whereas the correct answer is given by “check”. The first field’s type is number, thus the answer must be a number, whereas the second field’s type is statement, thus the answer must be a string.

We have more than 10 different types of fields, including radiobuttons, checkboxes, sets and formulas. The usage of tables is also possible.

Advanced users can also define more sophisticated mechanisms how to check the correctness of an answer, e.g.: If we would like to make 9 and 10 correct answers in the first questions, then the check would be [9, 10], meaning local OR of 9 and 10. If “Wien” as well as “Vienna” should be correct in the second questions, the check would be [“Wien”, “Vienna”].

Available field types

These are the most commonly needed types:

text / statement	a short (statement) or multiline (text) is expected; no exact check, but check for all terms that are defined in check
word / syllable	exact check for a short text (word) or single characters (syllable)
number	a number must be entered; automatically tolerates different ways how to write numbers, e.g. $7 = 7.0 = 7,0 = 07$
fraction	expects a fraction, write as a/b
radio / check	displays a radio (only one correct solution) or checkbox (many correct solutions)
formula	expects a formula, tolerates various ways how to write it, e.g. with more spaces
vector / set	an ordered (vector) or unordered (set) list of elements

Available attributes

This is an overview of field’s available attributes:

attributes	function
„name“: „x1“	the field’s name, must be unique
„type“: „text“	the field’s type
„pretext“: „What’s the time?“,	text, that is displayed in front of the input field
„check“: “17:00”	define checking correctness
„posttext“:	text, that is displayed after of the input field
„prefix“: „A =“	short text, that’s part of the field, but in front (e.g.: “A =”)
„postfix“: „cm ² “,	short text, that’s part of the field, but at the end (e.g.: “cm ² ”)
„options“: [„a“, „b“]	options, e.g. radiobutton
„delimiter“: „,“	delimiter, for delimiting single elements in a set
„default“: „ ... = ...“	text, that is initially shown in the field; can help the student to know what the field expects; is also useful to avoid too much typing, e.g. if the student should only set correct punctuation in a text
„placeholder“: “Wort1, Wort2, ...“	similar to default, but disappears as soon as the student

	enters text; also helps to understand, what the field expects
<pre>"feature": { "buttons": ["°", "α", "β", "δ", "γ", "ε"] },</pre>	buttons: these are displayed at the end of the field and help with entering symbols that might not be available on the keyboard
<pre>"feature": { "hint": "Runde auf mm!" },</pre>	hint: makes a ? appear at the end of the field and shows the text when clicked on

checking correctness

The following AND/OR logic is used:

„check“: []	no check done
„check“: [„“]	field needs to be empty
„check“: [a, b]	correct is a OR b
„check“: [[a, b]]	correct is a AND b
„check“: [[a, b], [c, d]]	correct is either a AND b - OR c and d
[[[a,b] , [c,d]]]	next level ... OR again