2		
3		
4	1	XReport: An online structured reporting platform for radiologists
5	1	Aneport. An online structured reporting platform for radiologists
6	-	
7 8	2	
9	3	Title / name of your software
10	4	XReport
11		
12	5	
13	6	Authors / main developers (incl. affiliations, addresses, email)
14 15		
15 16		
17	7	<sup>1</sup> Ahmed Harmouche (corresponding author), Department of Radiology, Medical School, University
18	8	of Pécs; email address: ahmedharmouche92@gmail.com; mailing address: UP MS Department of
19	9	Radiology: Hungary, 7624 Pécs, Ifjúság str. 13. phone number: +3630/8838435
20	5	
21 22		
22 23	10	<sup>2</sup> Ferenc Kövér, Pécs Diagnostic Center; email address: ferenc.kover@gmail.com; mailing address:
24	11	Pécs Diagnostic Center: Hungary, 7623 Pécs, Rét str. 2. phone number: +3672/242312
25	11	rees Diagnostie Genter. Hungary, 7023 rees, het str. 2. phone humber. 73072/242312
26		
27	12	<sup>3</sup> Sándor Szukits, Department of Radiology, Medical School, University of Pécs; email address:
28		szukits.sandor@pte.hu; mailing address: UP MS Department of Radiology: Hungary, 7624 Pécs,
29 30	13	
31	14	Ifjúság str. 13. phone number: +3672/536197
32		
33	1 Г	4 Tamés Dégri, Department of Neurogurgeny, Medical Coheal, University of Déga, email addresse
34	15	<sup>4</sup> Tamás Dóczi, Department of Neurosurgery, Medical School, University of Pécs; email address:
35	16	doczi.tamas@pte.hu; mailing address: UP MS Department of Neurosurgery: Hungary, 7623 Pécs,
36 37	17	Rét str. 2. phone number: +3672/535900
38		
39		
40	18	<sup>5</sup> Péter Bogner, Department of Radiology, Medical School, University of Pécs; email address:
41	19	bogner.peter@pte.hu; mailing address: UP MS Department of Radiology: Hungary, 7624 Pécs,
42 43	20	Ifjúság str. 13. phone number: +3672/535801
43 44		
45		
46	21	<sup>6</sup> Arnold Tóth, Department of Radiology, Medical School, University of Pécs; email address:
47	22	prsarn@gmail.com; mailing address: UP MS Department of Radiology: Hungary, 7624 Pécs, Ifjúság
48	23	str. 13.
49 50	24	
50 51	- ·	
52	25	
53	-	
54	26	
55		
56	27	
57 58		
58	28	
60		
61		
62		1
63 64		
64 65		
55		

1 2						
3 4						
5						
6	21 Subject of this mathed is relating to the which version in the formation of the sector of the sec					
<ul> <li>output of this method is plain text, which varies in style (structure, nomenclature, abbreviations, etc.) an</li> <li>between destars even when reporting the event same asse. Templated redialogy provides a structure for</li> </ul>						
8 9	<ul> <li>between doctors even when reporting the exact same case. Templated radiology provides a structure for registration of the structure for structure for structure for registration of the structure for</li></ul>					
10	33 34			is. We propose a web-based system for creating and using		
11			plogical structured reporting templates.	logics. We wrote the system with modular design in mind. We		
12 35 We developed our software based on web technologies. We wrote the system with modular design						
13 14	30 37	have separate libraries for the different functionalities: a rendering library which renders the templates based on a schema, an editor library which handles template creation, and an evaluator library, which parses, and executes				
15	38	our custom domain specific language, FormScript, which enables dynamic behaviour in our templates. We also				
16	39			wse, use and share templating reports. The backend of the		
17 18	40		cation is powered by Firebase from Google.	wse, use and share templating reports. The backend of the		
10 19	41	application is powered by Firebase from Google. We deployed our system at a publicly accessible domain at <u>https://app.radiosheets.com</u> .				
20	71	we a	we deployed our system at a publicly accessible domain at <u>https://app.radiosheets.com</u> .			
21	42					
22 23						
24	43					
25	44	Кеум	vords:			
26 27	45	struc	tured reporting; radiology; eHealth; JavaScrip	t		
27 28						
29	46					
30	47					
31 32	48	Required Metadata				
33	49					
34	50	Current code version				
35 36	51					
30 37	52	Table	e 1 – Code metadata (mandatory)			
38	-	Nr	Code metadata description	Please fill in this column		
39 40		C1	Current code version	v1.4.1		
40 41		C2	Permanent link to code/repository used of	https://github.com/wpmed92/xreport		
42			this code version			
43		C3	Code Ocean compute capsule	Not available		
44 45		C4	Legal Code License	MIT License		
46		C5	Code versioning system used	git		
47		C6	Software code languages, tools, and	JavaScript, TypeScript, HTML5, CSS, Bootstrap, Firebase, npm		
48 49			services used			
50		C7	Compilation requirements, operating	Webpack, Node.js, npm, Angular, Firebase		
51			environments & dependencies			
52 53		C8	If available Link to developer documentation/manual	https://wpmed92.github.io/xreport/		
54		C9	Support email for questions	ahmedharmouche92@gmail.com		
55	53	•••				
56 57	55 54					
58	55					
59	55					
60						
61 62				2		
63				2		
64						
65						

 There is no consensus among radiologists on what a good radiological report is. Both radiologists and clinicians who receive the reports have different views on the optimal layout and content. [1] Currently the most preferred way of reporting is dictation. With the advance of speech recognition technology, it became faster, more accurate and easier to produce reports by dictating than by typing. [2] One of the key features of dictation is that it eliminates context switching. Radiologists do not have to take their eyes off the image at any point during reporting, whereas in case of typing they switch between looking at the keyboard and looking at the screen. However, the problem with both dictation and typing is that the output is plain text. Saving large amounts of reports in the Health Information System (HIS) in plain text will generate an archive which is not maintainable, and not searchable. Valuable information will be lost. Also, as mentioned earlier, no two radiologists will write the same report about the exact same image. The use of different nomenclatures, ordering of findings and abbreviations may result in confusion among doctors, and inefficiency in communication and patient management. To address these issues another form of reporting aroused: structured reporting. Structured reporting gives doctors a framework for writing reports. Mostly this framework is template based. A good example of this is the RadReport reporting template collection created by the Radiological Society of North America (RSNA). [3] The collection contains templates grouped by specialties. The templates are submitted by the users and then are reviewed by the Template Library Advisory Panel to ensure the submissions meet certain criteria. The templates are composed of sections, subsections, and input fields. The report is generated by filling the form. Using such templates for reporting have multiple benefits. They can help give radiologists a guideline on what the report should include for a given pathology or modality thus diminishing the possibility of missing some important findings or information. Furthermore, extracting data from such templates and saving it in a database is straightforward, as opposed to plain text reports.

The drawback to templated radiology is that it is hard to find the optimum of how much a report should be structured. If one tried to cover all the possible cases and logical branches using built-in input fields with predefined options, it would be too time consuming to write the report. But if one used mostly text areas, the structured report would resemble a dictation template, and the benefits of structured reporting would be less significant.

> We propose a new radiological structured reporting software that is free, cross-platform, can be integrated into the dictation-based reporting workflow of a radiologist, enables template creation and report generation. With our solution we aim to make structured reporting more widespread and accessible, thus increasing the quality and consistency of radiological reports.

## 2. Software Description

2.1. Software Architecture

The software was written as a web application to support all operating systems and devices. Two programming languages were used throughout the development process, namely JavaScript and TypeScript.

The project can be divided into two main parts: the library and the application. The library is a standalone module that implements the core features of the software: template building and reporting. The application can be any host that integrates the library, in our case it is a Single Page Application (SPA). Our workflow of creating reporting templates resembles intentional programming [4]. The programmer builds the foundation (template builder) of the software on top of which the domain expert (radiologist) can build the actual application (template). The programmer can later add if-else logic to the template.

#### The library

49 130

**131** 

51 132

<sup>52</sup> 133

**107** The library exposes four public methods to interact with: makeWidget, togglePreviewMode, getReportAsText and getTemplateForUpload. The entry point is makeWidget. It can instantiate a new empty template builder or load a template from a Uniform Resource Locator (URL). Internally it creates an instance of each of the 19 110 following classes: XReportDOM, XReportRender, Evaluator. The XReportDOM implements a custom subset of 20 111 the Document Object Model (DOM) which allows only specific elements of the DOM or compositions of DOM <sup>21</sup> 112 elements to be used. The XReportRender calls the render methods of the XReportDOM entities and uses them to assemble either a builder or a viewer component, depending on whether the library is in editor or viewer mode. In editor mode the templates can be modified, whereas in viewer mode they are read-only, and are 25 115 ready to generate reports. The Evaluator is an interpreter for our Domain Specific Language (DSL) called 26 116 FormScript. It adds dynamic behaviour to the templates through simple if-else logics and calculations. An example of a typical use case for FormScript is to show or hide a specific field if certain conditions are met, or to calculate a score for a scoring system. To view the generated report the library exposes the togglePreviewMode method. Calling this method will transfer the viewer from reporting state to output state 30 119 31 120 or vice versa. In output state the reporter can see the textual output of the form. The generated text can be accessed by the getReportAsText function. When the library is in editor mode a template can be saved by first getting it in JavaScript Object Notation (JSON) format with getTemplateForUpload and then sending it to a **123** web service or storing it locally.

#### **124 Template structure**

39 125 The templates are composed of rows, which may have one or more groups in it. Groups are label-entity pairs, and entities are the form's input elements. 

- The JSON)structure of a template is as following:
- { "formScript": "Form script source code is here", report: [{ XFormElem #1 }, { XFormElem #2 }...]}
- General fields in XFormElem:
  - type: defines what element to render, e.g. row, group, sel, mulsel
  - id: a random generated unique identifier
  - scriptAlias: an identifier/variable name by which FormScript can reference the field; auto-generated, but can be changed by user
  - hideFromOutput: determines whether the value of the field should be visible in the generated text • output
  - hidden: determines whether the field should be rendered •
  - children: a list of groups in a row
  - child: the entity of a group

<sub>47</sub> 157

**162** 

48 158

There are fields specific to each entity but they are not listed here.

## FormScript

FormScript is a DSL that is specifically designed to run inside XReport templates. It allows custom logic to be executed safely in forms, thus adding dynamic behaviour to them. The script can be edited when the library is in editor mode and is accessible through the getScript library call. Once saved, it is stored in the same JSON file as the template itself.

The FormScript syntax is similar to that of JavaScript with some syntactical differences shown in Table 1.

Features	JavaScript	FormScript
Logical and	&&	and
Logical or	11	or
Power	**	^
if expression	if (a == b) { }	if a == b { }

Table 1 Syntactical comparison of FormScript and JavaScript

Supported binary operations: addition (+), subtraction (-), division (/), multiplication (\*), modulo (%), less than (<), greater than (>), less than or equal to (<=), greater than or equal to (>=), equal to (==), logical and (and), logical or (or), to the power of (^)

Unary operations: unary not (!), unary minus (-), unary plus (+)

Statements: expression, assignment, if, function call

Types: string, boolean, number 

> Numerical and string literals are supported. The only variables that are allowed in FormScript are references to form elements. As mentioned earlier, variables are defined in the editor through the scriptAlias property. Function calls are defined only on variables. It is not allowed to declare functions neither are there predefined library functions without an element context. Calling a function has the following form: variable.function(...parameters). Functions should be defined for XFormElem classes. When XReport loads a report, it checks for an attached script. If there is a script attachment, it will start running it in an Evaluator instance.

2		
3		
4 5	166	Backend
6	167	Our SPA has a backend powered by Google Firebase to store the template resources. We store the template
7 8	168	files in storage buckets. The metadata for each template, such as date of creation, creator's username,
o 9	169	template name, template category is saved to Cloud Firestore documents.
10 11	170	The process of uploading a template to our backend includes the following steps:
12		
13 14	171	<ul> <li>query template JSON from the library through getTemplateForUpload</li> </ul>
15	172	<ul> <li>assemble upload metadata: date of creation, category, username, template name, template URL</li> </ul>
16	173	save the metadata to a Cloud Firestore document
17	174 175	upload the template JSON file to the storage
18	175	
19 20	176	Frontend
21 22	177	The frontend is built as a SPA using the Angular [5] and the Bootstrap Cascading Style Sheets (CSS)
23	178	frameworks. Every icon used in the app are taken from the Font Awesome icon library. Angular supports
24	179	client-side navigation, asynchronous data binding among others, which enables us to easily fetch and render
25	180	views. To retrieve templates from Cloud Firestore we use the official Firebase JavaScript Software
26	181	Development Kit (SDK) and the RxJS reactive programming library. In Firebase terms the templates form a
27 28	182	collection, and individual entries in this collection are documents. To show these documents on the screen we
20 29		
30 31	183	followed the Model-View-ViewModel pattern with data binding.
32 33	184	
34	185	
35	186	
36	187	
37	188	
38 39	189	
40		
41	190	
42	191	
43	192	
44 45	193	
46	194	
47	195	
48	196	3. Illustrative Examples
49		5. Indstrative Examples
50 51	197	
52 53	198	Template viewing and building
54	199	
55	200	The viewer/builder page is where we load our templates and render them with our library as show in Figure 1. The
56	201	form is centered horizontally and have a slight drop shadow around it. There is a button group on the right side of
57 58	202	the form which contains different buttons based on which state the page is currently in (viewer or builder).
50 59	202	and form which contains an erent battons based on which state the page is currently in (viewer or builder).
60		
61		
62		6
63		
64 65		
55		

Figure 1 A COVID-19 CT structured Report         Provide the structure of the structu	1 2 3		
7       Image: Construction of the state of			powered by radiosheets
Figure 1 A COVID-19 CT template rendered in XReport Figure 1 A COVID-19 CT template rendered in XReport Cos Figure 1 A COVID-19 CT template rendered in XReport Cos Cos Portion in viewer state: Portion in viewer state: Cos Cos Date Cos Date Cos <pcos< p=""> Cos <pcos< p=""> Cos <pcos< p=""> <p< td=""><td></td><td></td><td>Clinical information</td></p<></pcos<></pcos<></pcos<>			Clinical information
14       if i	8 9 10 11		CT Protocol  CT Thorax Full dose CT Thorax LOW dose CT Thorax LOW dose CT Thorax Low Angiography CT Thorax Standard + IV contrast
15       Image: Constraint of the template in temp	-		
167       Image: Second S			
11       Constantion         12       203         12       204         Figure 1 A COVID-19 CT template rendered in XReport         12       205         12       206         12       207         12       00         12       207         12       00         12       207         12       00         12       00         13       207         14       0         15       0         16       10         17       Buttons in builder state:         16       11         17       Builder components         18       211         19       There are 13 components to choose from when building a template:         16       11         17       Builder components         18       211         19       There are 13 components to choose from when building a template:         16       122         18       11         19       There are 13 components to choose from when building a template:         10       223       Number field         1224       Calculated field			No
<ul> <li>203</li> <li>204</li> <li>Figure 1 A COVID-19 CT template rendered in XReport</li> <li>205</li> <li>206</li> <li>207</li> <li>preview report</li> <li>208</li> <li>copy to clipboard (copies the report output to clipboard)</li> <li>029</li> <li>new report</li> <li>210</li> <li>share (copy the template link to clipboard so that it can be shared)</li> <li>211</li> <li>212</li> <li>213</li> <li>214</li> <li>save template</li> <li>discard template</li> <li>discard template</li> <li>216</li> <li>217</li> <li>Builder components</li> <li>218</li> <li>219</li> <li>There are 13 components to choose from when building a template:</li> <li>220</li> <li>Plain text</li> <li>221</li> <li>Sould field</li> <li>222</li> <li>Plain text</li> <li>Single choice</li> <li>Single choice</li> <li>223</li> <li>Number field</li> <li>224</li> <li>Calculated field</li> <li>225</li> <li>Boolean field</li> <li>226</li> <li>Single choice</li> <li>227</li> <li>Multiple choice</li> <li>228</li> <li>Textarea</li> <li>229</li> <li>Date</li> <li>230</li> <li>Header</li> <li>231</li> <li>Information</li> <li>232</li> <li>Rating scale</li> </ul>			Consolidation
21204Figure 1 A COVID-19 CT template rendered in XReport23205206Buttons in viewer state:27• preview report28• copy to clipboard (copies the report output to clipboard)209• new report210• share (copy the template link to clipboard so that it can be shared)211211212211213Buttons in builder state:214• save template215• discard template216• discard template217Builder components218• There are 13 components to choose from when building a template:4• 2207• Plain text223• Number field224• Calculated field225• Boolean field226• Single choice227• Multiple choice228• Textarea229• Date231• Information232• Rating scale243• Rating scale		202	○ Yes
224       Figure 1A COVID-19 CT template rendered in XReport         23       205         266       Buttons in viewer state:         27       • preview report         29       000         209       • copy to clipboard (copies the report output to clipboard)         201       • share (copy the template link to clipboard so that it can be shared)         211       • share (copy the template link to clipboard so that it can be shared)         211       • share (copy the template link to clipboard so that it can be shared)         211       • share (copy the template         212       • save template         213       Buttons in builder state:         32       • discard template         32       • discard template         32       • discard template         32       • discard template         34       • discard template         35       • discard template         36       • discard template         37       • There are 13 components         36       • Text field         37       • Text field         38       • Calculate field         32       • Raide         32       • Boolean field         32       • Boolean field <td></td> <td></td> <td>Crazy paving</td>			Crazy paving
24       205         206       Buttons in viewer state:         7       • preview report         207       • preview report         208       • copy to clipboard (copies the report output to clipboard)         209       • new report         31       210       • share (copy the template link to clipboard so that it can be shared)         32       211         33       212         34       213         35       Buttons in builder state:         36       214         37       • save template         38       215         41       17         Builder components         218         41       217         42       218         42       218         43       219         44       219         45       219         46       220         47       • Text field         48       221       • Text field         49       222       • Plain text         50       223       • Number field         51       224       • Calculated field         52       225       • Boolean fi	22	204	Figure 1 A COVID-19 CT template rendered in XReport
20       Buttons in viewer state:         27       preview report         200       . copy to clipboard (copies the report output to clipboard)         209       . new report         210       . share (copy the template link to clipboard so that it can be shared)         211       .         3211       .         3212       Buttons in builder state:         4       .         5       .         6       .         7       14         6       .         7       216         9       .         10       .         11       .         12       .         12       .         13       .         14       .         15       .         16       .         17       Builder components         18       .         19       .         214       .         217       Builder template         14       .         218       .         119       .         211       .         211       .	24	205	
<ul> <li>28 207 • preview report</li> <li>208 • copy to clipboard (copies the report output to clipboard)</li> <li>209 • new report</li> <li>210 • share (copy the template link to clipboard so that it can be shared)</li> <li>211</li> <li>212</li> <li>213</li> <li>214</li> <li>213</li> <li>214</li> <li>213</li> <li>214</li> <li>214</li> <li>214</li> <li>215 • discard template</li> <li>216</li> <li>216</li> <li>217</li> <li>218</li> <li>218</li> <li>219</li> <li>218</li> <li>219</li> <li>210</li> <li>211</li> <li>211</li> <li>217</li> <li>218</li> <li>218</li> <li>219</li> <li>218</li> <li>219</li> <li>210</li> <li>220</li> <li>211</li> <li>221</li> <li>213</li> <li>214</li> <li>214</li> <li>215</li> <li>215</li> <li>216</li> <li>216</li> <li>217</li> <li>218</li> <li>218</li> <li>218</li> <li>219</li> <li>218</li> <li>219</li> <li>218</li> <li>219</li> <li>218</li> <li>219</li> <li>210</li> <li>210</li> <li>210</li> <li>211</li> <li>211</li> <li>212</li> <li>213</li> <li>214</li> <li>214</li> <li>214</li> <li>215</li> <li>225</li> <li>226</li> <li>226</li> <li>226</li> <li>226</li> <li>227</li> <li>226</li> <li>228</li> <li>226</li> <li>228</li> <li>226</li> <li>228</li> <li>226</li> <li>228</li> <li>227</li> <li>231</li> <li>231</li> <li>231</li> <li>231</li> <li>231</li> <li>231</li> <li>232</li> <li>233</li> <li>233</li> <li>234</li> <li>234</li> <li>235</li> <li>235</li> <li>236</li> <li>237</li> <li>231</li> <li>231</li> <li>231</li> <li>232</li> <li>233</li> <li>233</li> <li>234</li> <li>234</li> <li>235</li> <li>235</li> <li>234</li> <li>235</li> <li>235</li> <li>235</li> <li>236</li> <li>237</li> <li>231</li> <li>231</li> <li>234</li> <li>234</li> <li>235</li> <li>235</li> <li>234</li> <li>235</li> <li>235</li> <li>235</li> <li>236</li> <li>236</li> <li>237</li> <li>237</li> <li>238</li> <li>238</li> <li>238</li> <li>239</li> <li>231</li> <li>230</li> <li>231</li> <li>231</li> <li>231</li> <li>232</li> <li>234</li> <li>234</li> <li>235</li> <li>234</li> <li>234</li> <li>235</li> <li>235</li> <li>234</li></ul>	26	206	Buttons in viewer state:
<ul> <li>209</li> <li>new report</li> <li>210</li> <li>share (copy the template link to clipboard so that it can be shared)</li> <li>211</li> <li>212</li> <li>213</li> <li>214</li> <li>save template</li> <li>215</li> <li>discard template</li> <li>216</li> <li>217</li> <li>Builder components</li> <li>218</li> <li>219</li> <li>There are 13 components to choose from when building a template:</li> <li>220</li> <li>Plain text</li> <li>223</li> <li>Number field</li> <li>224</li> <li>Calculated field</li> <li>225</li> <li>Boolean field</li> <li>226</li> <li>Single choice</li> <li>227</li> <li>Multiple choice</li> <li>228</li> <li>Textarea</li> <li>229</li> <li>Date</li> <li>230</li> <li>Header</li> <li>231</li> <li>Information</li> <li>232</li> <li>Rating scale</li> <li>7</li> </ul>		207	preview report
<ul> <li>210</li> <li>share (copy the template link to clipboard so that it can be shared)</li> <li>211</li> <li>212</li> <li>213</li> <li>214</li> <li>Buttons in builder state:</li> <li>214</li> <li>save template</li> <li>215</li> <li>discard template</li> <li>216</li> <li>217</li> <li>Builder components</li> <li>218</li> <li>219</li> <li>There are 13 components to choose from when building a template:</li> <li>220</li> <li>211</li> <li>221</li> <li>Plain text</li> <li>223</li> <li>Number field</li> <li>224</li> <li>Calculated field</li> <li>225</li> <li>Boolean field</li> <li>226</li> <li>Single choice</li> <li>227</li> <li>Multiple choice</li> <li>228</li> <li>Text rea</li> <li>229</li> <li>Date</li> <li>230</li> <li>Header</li> <li>231</li> <li>Information</li> <li>232</li> <li>Rating scale</li> <li>44</li> </ul>			
<ul> <li>and (ic) for the impletermine expectation of the interval of the inte</li></ul>			
<ul> <li>31 212</li> <li>34 213 Buttons in builder state:</li> <li>36 214 • save template</li> <li>39 216 • discard template</li> <li>39 216 • discard template</li> <li>39 216 • discard template</li> <li>40 • discard template</li> <li>41 217 Builder components</li> <li>42 218 • There are 13 components to choose from when building a template:</li> <li>44 45 219 • There are 13 components to choose from when building a template:</li> <li>42 200 • discard template</li> <li>42 21 • Text field</li> <li>49 222 • Plain text</li> <li>50 223 • Number field</li> <li>52 225 • Boolean field</li> <li>52 225 • Boolean field</li> <li>52 226 • Single choice</li> <li>54 227 • Multiple choice</li> <li>55 228 • Textarea</li> <li>57 229 • Date</li> <li>58 230 • Header</li> <li>59 231 • Information</li> <li>60 232 • Rating scale</li> <li>61</li> </ul>			<ul> <li>share (copy the template link to clipboard so that it can be shared)</li> </ul>
<ul> <li>Buttons in builder state:</li> <li>214 <ul> <li>save template</li> <li>215</li> <li>discard template</li> </ul> </li> <li>216 <ul> <li>discard template</li> <li>217</li> <li>Builder components</li> <li>218</li> </ul> </li> <li>219 <ul> <li>There are 13 components to choose from when building a template:</li> </ul> </li> <li>220 <ul> <li>Plain text</li> <li>223</li> <li>Number field</li> </ul> </li> <li>224 <ul> <li>Calculated field</li> <li>225</li> <li>Boolean field</li> <li>226</li> <li>Single choice</li> <li>227</li> <li>Multiple choice</li> <li>228</li> <li>Textarea</li> <li>229</li> <li>Date</li> <li>230</li> <li>Header</li> <li>231</li> <li>Information</li> <li>232</li> <li>Rating scale</li> </ul> </li> </ul>			
<ul> <li>Save template</li> <li>Save template</li> <li>discard templat</li></ul>	35		Buttons in builder state:
<ul> <li>215 discard template</li> <li>216</li> <li>217 Builder components</li> <li>218</li> <li>218</li> <li>219 There are 13 components to choose from when building a template:</li> <li>220</li> <li>44</li> <li>220</li> <li>47</li> <li>48 221 Text field</li> <li>222 Plain text</li> <li>223 Number field</li> <li>224 Calculated field</li> <li>225 Boolean field</li> <li>226 Single choice</li> <li>227 Multiple choice</li> <li>228 Textarea</li> <li>229 Date</li> <li>230 Header</li> <li>231 Information</li> <li>232 Rating scale</li> <li>41</li> </ul>		214	save template
<ul> <li>40</li> <li>41 217 Builder components</li> <li>43 218</li> <li>44</li> <li>45 219 There are 13 components to choose from when building a template:</li> <li>46 220</li> <li>47</li> <li>48 221 • Text field</li> <li>49 222 • Plain text</li> <li>50 223 • Number field</li> <li>51 224 • Calculated field</li> <li>52 225 • Boolean field</li> <li>53 226 • Single choice</li> <li>54 227 • Multiple choice</li> <li>55 228 • Textarea</li> <li>57 229 • Date</li> <li>58 230 • Header</li> <li>59 231 • Information</li> <li>60 232 • Rating scale</li> </ul>			discard template
<ul> <li>41 217 Builder components</li> <li>218</li> <li>219 There are 13 components to choose from when building a template:</li> <li>420</li> <li>44</li> <li>45 220</li> <li>47</li> <li>48 221 • Text field</li> <li>49 222 • Plain text</li> <li>50 223 • Number field</li> <li>51 224 • Calculated field</li> <li>52 225 • Boolean field</li> <li>53 226 • Single choice</li> <li>54 227 • Multiple choice</li> <li>55 228 • Textarea</li> <li>56 228 • Textarea</li> <li>57 229 • Date</li> <li>58 230 • Header</li> <li>59 231 • Information</li> <li>60 232 • Rating scale</li> <li>61</li> </ul>		216	
<ul> <li><sup>43</sup></li> <li><sup>44</sup></li> <li><sup>45</sup></li> <li><sup>49</sup></li> <li><sup>47</sup></li> <li><sup>48</sup></li> <li><sup>221</sup> Text field</li> <li><sup>49</sup></li> <li><sup>222</sup> Plain text</li> <li><sup>50</sup></li> <li><sup>223</sup> Number field</li> <li><sup>51</sup></li> <li><sup>224</sup> Calculated field</li> <li><sup>52</sup></li> <li><sup>225</sup> Boolean field</li> <li><sup>53</sup></li> <li><sup>226</sup> Single choice</li> <li><sup>54</sup></li> <li><sup>227</sup> Multiple choice</li> <li><sup>55</sup></li> <li><sup>228</sup> Textarea</li> <li><sup>57</sup></li> <li><sup>229</sup> Date</li> <li><sup>58</sup></li> <li><sup>230</sup> Header</li> <li><sup>59</sup></li> <li><sup>231</sup> Information</li> <li><sup>60</sup></li> <li><sup>232</sup> Rating scale</li> <li><sup>7</sup></li> </ul>	41	217	Builder components
45219There are 13 components to choose from when building a template:462204748221Text field49222Plain text50223Number field51224Calculated field52225Boolean field53226Single choice54227Multiple choice55228Textarea57229Date58230Header59231Information60232Rating scale617	43	218	
46       220         47       • Text field         49       222       • Plain text         50       223       • Number field         51       224       • Calculated field         52       225       • Boolean field         53       226       • Single choice         54       227       • Multiple choice         55       228       • Textarea         56       228       • Textarea         57       229       • Date         58       230       • Header         59       231       • Information         60       232       • Rating scale         61       •       •		219	There are 13 components to choose from when building a template:
48       221       • Text field         49       222       • Plain text         50       223       • Number field         51       224       • Calculated field         52       225       • Boolean field         53       226       • Single choice         54       227       • Multiple choice         55       228       • Textarea         57       229       • Date         58       230       • Header         59       231       • Information         60       232       • Rating scale         61	46	220	
<ul> <li>49 222</li> <li>Plain text</li> <li>50 223</li> <li>Number field</li> <li>51 224</li> <li>Calculated field</li> <li>52 225</li> <li>Boolean field</li> <li>53 226</li> <li>Single choice</li> <li>54 227</li> <li>Multiple choice</li> <li>55 228</li> <li>Textarea</li> <li>57 229</li> <li>Date</li> <li>58 230</li> <li>Header</li> <li>59 231</li> <li>Information</li> <li>60 232</li> <li>Rating scale</li> <li>61</li> </ul>		221	Text field
50223Number field51224Calculated field52225Boolean field53226Single choice54227Multiple choice55228Textarea57229Date58230Header59231Information60232Rating scale617			
52225Boolean field53226Single choice54227Multiple choice55228Textarea56229Date58230Header59231Information60232Rating scale617	50		
53       226       Single choice         54       227       Multiple choice         55       228       Textarea         56       229       Date         57       229       Date         58       230       Header         59       231       Information         60       232       Rating scale         61       7			Calculated field
54       227       • Multiple choice         55       228       • Textarea         56       229       • Date         57       229       • Date         58       230       • Header         59       231       • Information         60       232       • Rating scale         61			
55       227       • Multiple choice         56       228       • Textarea         57       229       • Date         58       230       • Header         59       231       • Information         60       232       • Rating scale         61	54		
50       229       • Date         57       230       • Header         58       230       • Information         60       232       • Rating scale         61       7	55		
57       230       •       Header         59       231       •       Information         60       232       •       Rating scale         61       •       •         62       7			
59       231       •       Information         60       232       •       Rating scale         61       •       •       •         62       7       •			
60 232 • Rating scale 61 62 7			
62 <b>7</b>	60		
	62 63		7

## Image

If we take an oncological example, a question regarding the size of the tumor would be a number field, a TNM staging system could be created using single choice fields, or a rating scale, etc. Every component is added to the form with a label attached. A component-label combination is called a group. Groups are added to rows, and rows are added to sections. The sections make up the whole template.

## 239 Component editors

Every component has an editor view as shown in Figure 2. Every component type has its own editable properties.
For example, the input field has a unit property (mm, cm, etc.), a single choice field has an options property, an
image has an URL property. The component editor is activated by hovering the mouse over the component, then
clicking on the pencil icon. Components can be deleted by clicking on the minus sign.

Field name	۲	
CT Protocol		
Script alias		
xElemamk6gs3kk		
Options		
CT Thorax Full dose;CT Thorax LOW dose;CT Thorax Pulmonary Angiography;CT Thorax Standard + IV contras	t;	•
	1.	

2 246 Figure 2 A component editor for a single choice input field

## 44 247 **Row editors**

Row operations can be performed by clicking on the three vertical dots at the end of each row. The click event will
trigger a secondary menu to open with all the components, and two actions: delete and duplicate.

## <sup>9</sup> 250 **FormScript editor**

S1 251 On the main builder component there is a button with a branch icon which toggles the view between template editing and FormScript editing. The FormScript editor as shown in Figure 3 is a simple resizable text area where the user can edit the dynamic logic that is attached to the template. When switching back from script editing, the script is automatically evaluated and the changes are visible.

	COVID-19 CT Structured Re	✓ ×
	Form script	×
	<pre>sevScore = sevTable.sum(); if gg == "Yes" {   ggLoc.show();   ggMultiple.show(); }</pre>	
	<pre>if con == "Yes" {     conLoc.show();     conMargin.show(); }</pre>	
256	+ -	
257	Figure 3 The FormScript editor	
258		
250		
259 260		
261		
262		
762		
263		
264		
265		
266		
267		
268		
269		
270		
271		
72		
273		
274		
275		
	9	
	-	

## 4. Impact

With XReport we built a free, cross-platform structured reporting platform for radiologists. It enables both creating and viewing reporting templates in an easy, user-friendly way. We built our software with modular design in mind and refactored the core features into a separate library to make embedding it into other products easy. We also built an application with the library embedded in it to demonstrate the easy integration. Furthermore, we designed a simple DSL called FormScript to add dynamic logic to our forms. The main feature of it, and the reason we created it in the first place, is security. It does not allow malicious code executions unlike the eval function of JavaScript. It is also very simple to use because of its limited feature set. Our templates are dynamic, responsive and have modern design. The templates generate easy to copy-paste structured textual output to be compatible with any HIS, and to integrate well into dictation-based workflows. Our templates help not only in precise reporting, but also serve as a guide for radiologists thanks to our custom form elements such as images and rating tables.

We compared our solution to a similar free service developed by RSNA. From a technological point of view both programs are similar since they are built using web technologies but they have their differences when it comes to the ecosystem, editing process and user experience. The RSNA template library has a more mature ecosystem: there are a lot of contributors who build and upload templates, there are some nice to have features such as favouriting a template. But the template editing itself is less advanced than ours. In the RSNA editor the screen flow to get to the actual editing is as following: click on "Create and Upload a Template button", click on "T-Rex **295** Template Editor", interact with a popup which asks how the user wants to start the editing, click on one of the options. In our program the screen flow is a lot simpler: click on "Add new template", and you are in the editor. In the RSNA editor adding individual elements has some issues. The elements have to be drag and dropped from a side panel, which is problematic on mobile devices as there is not enough space. The element editor works as a **299** pop-up which brings the user out of the editing context. In our app adding elements is responsive (works on 34 300 mobile devices as well), and is inline, so the user remains in the editing context throughout the whole process. When it comes to how dynamic the templates are we found that RSNA templates do not allow dynamic behavior such as hiding/showing elements based on certain conditions. Through FormScript our system enables fully **303** dynamic behaviour. The RSNA editor lacks some important elements such as images and rating tables which are essential in information sharing and oncological grading systems.

## 5. Conclusions

This paper introduces XReport, a free, web-based structured reporting platform for radiologists. It enables both creating and viewing reporting templates in an easy, user-friendly way. Our system is deployed at https://app.radiosheets.com and is ready to be used. Template creation and editing requires login, but template viewing, copying the generated reports and sharing the templates do not. The templates currently available in the app have been created and are used by our research group and by radiologists from Pécsi Diagnosztikai Központ and from the University of Pécs.

- **316**

1		
2		
3		
4	317	
5 6	318	Conflict of Interest
7	319	
8	320	- No conflict of interest exists:
9	321	We wish to confirm that there are no known conflicts of interest associated with this publication and there
10		has been no significant financial support for this work that could have influenced its outcome.
11	322	has been no significant financial support for this work that could have influenced its outcome.
12	323	
13	324	Acknowledgements
14 15	325	A. T. was supported by the Bolyai Scholarship of the Hungarian Academy of Science.
16		
17		
18	326	References
19	327	
20		
21		[1] D. Ganeshan, et al., "Structured Reporting in Radiology.," Academic Radiology, vol. 25, no. 1, pp. 66-73, 2018.
22 23		[2] D.S. Rana, et al., "Voice recognition for radiology reporting: is it good enough?," <i>Clinical Radiology</i> , vol. 60,
23 24		no. 11, pp. 1205-1212, 2005.
25		
26		[3] T. A. Morgan, M. E. Helibrun and C. E. Kahn, "Reporting Initiative of the Radiological Society of North America: Progress and New Directions," <i>Radiology</i> , vol. 273, no. 3, pp. 642-645, 2014.
27		
28 29		[4] C. Simonyi, M. Christerson and S. Clifford, "Intentional software," in ACM Sigplan Notices, 2006.
29 30		[5] N. Jain, A. Bhansali and D. Mehta, "AngularJS: A modern MVC framework in JavaScript," Journal of Global
31		Research in Computer Science, vol. 5, no. 12, pp. 17-23, 2014.
32	328	
33		
	329	
35		
36 37	330	
38		
39	331	
40		
	332	
42		
43		
44 45		
46		
47		
48		
49		
50		
51		
52 53		
53 54		
55		
56		
57		
58		
59		
60		
61 62		
63		11
64		

### 4 333 Current executable software version

- Ancillary data table required for sub version of the executable software: (x.1, x.2 etc.) kindly replace examples in
- <sub>7</sub> 335 right column with the correct information about your executables and leave the left column as it is.

#### Table 2 – Software metadata (optional)

Nr	(Executable) software metadata description	Please fill in this column
S1	Current software version	1.4.1
S2	Permanent link to executables of this version	https://app.radiosheets.com
S3	Legal Software License	MIT License
S4	Computing platforms/Operating Systems	Cross-platform, Web-based system
S5	Installation requirements & dependencies	No installation needed, works in any modern browser.
S6	If available, link to user manual - if formally published include a reference to the publication in the reference list	
S7	Support email for questions	ahmedharmouche92@gmail.com

<sup>28</sup> 338

б

8 336