Supplementary Materials for A Parse-Then-Place Approach for Generating Graphic Layouts from Textual Descriptions

A. Intermediate Representation

In our approach, an intermediate representation (IR) is introduced to formally represent the layout constraints, including element type constraints, position constraints, size constraints and hierarchy constraints. The context-free grammar of our designed IR is presented in Equation 4. Each nonterminal symbol, which is to the left of $\rightarrow$, corresponds to a production rule. The terminal symbols all start with a lowercase letter, including the keywords (*group*, *position*) and the values (*image*, *top*).

Specifically, we use Pos and $Size$ to denote position and size constraints. $Element$ represents a graphic element, containing a $Type$ (required) and several $Prop$s (not necessary). $Group$ stands for a hierarchy constraint and consists of multiple $Element$s. Moreover, we use Num to simplify the representation when there are too many elements of the same type.

$$\begin{aligned}
R &\rightarrow [A] \mid [AA] \mid [AAA] \mid [AAA\dots A] \\
A &\rightarrow Element \mid Group \\
Prop &\rightarrow Pos \mid Size \mid Num \\
Element &\rightarrow [e\colon Type] \mid [e\colon Type\ Prop] \mid [e\colon Type\ Prop\ Prop] \mid [e\colon Type\ Prop\ Prop\ Prop] \\
Group &\rightarrow [group\ Num[item\ Element\dots Element]] \\
Pos &\rightarrow [prop\colon position\ Pvalue] \\
Size &\rightarrow [prop\colon size\ Svalue] \\
Num &\rightarrow [prop\colon repeat\ Nvalue] \\
\\
Type &\rightarrow image \mid text \mid title \mid icon \mid \dots \\
Pvalue &\rightarrow top \mid bottom \mid left \mid right \\
Svalue &\rightarrow small \mid large \\
Nvalue &\rightarrow 1 \mid 2 \mid 3 \mid 4 \mid \dots
\end{aligned} \tag{4}$$

Since different layout domains have different element types (see Table 6, e.g., *drawer* is an element in RICO, but not in WebUI), $Type$ in Equation 4 corresponds to different production rules on WebUI and RICO. We show some examples of IR and corresponding descriptions in Table 7. Due to the well-designed grammar, IR is sufficient to represent the key layout constraints specified in the description.

In terms of IR annotation, the annotators are asked to label the textual description and IR simultaneously for a given layout when constructing Web5K and RICO2.5K. And they should describe the layout constraints in text and IR in the same order. This reduces the learning difficulty of the model in the parse stage.

Dataset	Element Type Set
WebUI	text, link, button, title, description, submit, image, background image, icon, logo, input
RICO	text, image, icon, list item, text button, toolbar, web view, input, card, advertisement, background image, drawer, radio button, checkbox, multi-tab, pager indicator, modal, on/off switch, slider, map view, button bar, video, bottom navigation, number stepper, date picker

Table 6. The element type sets in two datasets.

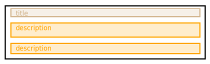
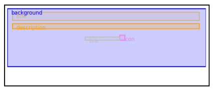
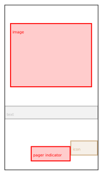
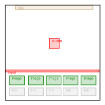
#	Textual Description & IR	Layout
1	**Text**: A page introduces the huge selection of ski race equipment for users. This page should include one title on the top. And there should be two groups of the detailed descriptions of the ski race equipment for users. **IR**: `[ [e:title [prop:position "top"] ] [e:description [prop:repeat "2"] ] ]`	
2	**Text**: A page introduces the service of a company. There should be a title instructing the user to check out more with a short introduction. It is better to include a link "SEE HOW IT WORKS". **IR**: `[ [e:title] [e:description [prop:size "small"] ] [e:link] ]`	
3	**Text**: This is a page for a test. On the top, there is an image that occupies half of the image. Below, there is a text. At the bottom, there is a pager indicator. **IR**: `[ [e:image [prop:position "top"] [prop:size "large"] ] [e:text] [e:pager indicator [prop:position "bottom"] ] ]`	
4	**Text**: A page contains videos of customer stories. The page should include a title "CUSTOMER STORIES" and a play button. There should also be 5 images with links to different videos at the bottom of the page. **IR**: `[ [e:title] [e:button] [group [prop:repeat "5"] [item [e:image [prop:position "bottom"] ] [e:link [prop:position "bottom"] ] ] ] ]`	

Table 7. Illustrative examples of IR and corresponding textual descriptions. IR is annotated according to the grammar in Equation 4. It is worth noting that the annotators don't always describe all elements in the layout (e.g., the background image in #2 is omitted). Our method automatically completes the omitted yet important elements.

B. Constraint Sequence and Layout Sequence

The details of constraint sequences and layout sequences are elaborated in the main paper, here we show some examples of them. Table 8 shows the constraint sequence and layout sequence corresponding to IR and layout in Table 7. We use the keyword *complete* to denote the omitted elements in layout sequence. In the implementation, we use the *undefined* token as a placeholder for unspecified constraints.

1	Constraint Sequence	`description undefined undefined | description undefined undefined | title top undefined`
	Layout Sequence	`description 13 7 93 7 | description 13 17 93 5 | title 13 0 93 4`
2	Constraint Sequence	`description undefined small | link undefined undefined | title undefined undefined`
	Layout Sequence	`complete background image 0 5 120 35 | description 3 14 113 3 | complete icon 68 21 3 3 | link 47 22 21 2 | title 3 7 113 5`
3	Constraint Sequence	`image top large | pager indicator bottom undefined | text undefined undefined`
	Layout Sequence	`complete icon 102 211 41 22 | image 9 29 125 98 | pager indicator 41 220 60 22 | text 0 157 144 20`
4	Constraint Sequence	`button undefined undefined | title undefined undefined | [image bottom undefined | link bottom undefined] | [image bottom undefined | link bottom undefined] | [image bottom undefined | link bottom undefined] | [image bottom undefined | link bottom undefined] | [image bottom undefined | link bottom undefined]`
	Layout Sequence	`button 54 33 10 10 | complete input 10 65 94 2 | title 20 0 78 4 | [ image 14 71 15 9 | link 15 72 14 8 ] | [ image 33 71 15 9 | link 34 72 14 8 ] | [ image 51 71 15 9 | link 52 72 14 8 ] | [ image 68 71 15 9 | link 69 72 14 8 ] | [ image 86 71 15 9 | link 87 72 14 8 ]`

Table 8. Some examples of constraint sequences and layout sequences.

C. IR Synthesis

Algorithm 1 outlines the IR synthesis procedure. It begins with randomly discarding a small proportion of elements in a layout (Line 2). The discarded elements are intended to be auto-completed by a model. Next, the algorithm tries to extract constraints for the remaining elements, from the most complicated hierarchy constraints to the simple element type constraints (Line 5-10). There could be dozens of constraints in a layout, but users typically will not specify that many in text. Hence, we only sample a subset of constraints to synthesize IR (Line 11-12).

Specifically, `TypeConst`, `PositionConst` and `SizeConst` first extract the element tags and coordinates from layout source code (HTML in WebUI, XML in RICO). The tags in RICO are directly used as element types. In WebUI, we use heuristic rules to determine the element types. For example, an element with an "h" tag is considered a title, an element with an "a" tag is considered a link, an element with a "src" attribute is considered an image, and so on. From the element coordinates, we infer position and size constraints by setting thresholds. For example, an element is considered to have a "left" constraint when its center x-coordinate is less than 0.25 of the screen width. Similarly, we can get other position and size constraints.

For the most complicated hierarchy constraints, we mainly rely on "ul" tags. And an element with a "ul" tag is considered the parent element of the group. Elements inside the "ul" tag are considered child elements of the group.

Algorithm 1: IR synthesis from layout

```
Input: layout $y$, element discard rate $r$
Output: synthetic IR $\hat{z}$
1  $E \leftarrow$ extractElements ($y$) ;
2  $E_c \leftarrow$ discardElements ($y$, $r$);
3  $E_r \leftarrow E - E_c$;
4  $C \leftarrow \emptyset$;
5  $C \leftarrow C \cup$ HierarchyConst ($E_r$) ;
6  for $e \in E_r$ do
7      $C \leftarrow C \cup$ SizeConst ($e$);
8      $C \leftarrow C \cup$ PositionConst ($e$);
9      $C \leftarrow C \cup$ TypeConst ($e$);
10 end
11 $\tilde{C} \leftarrow$ sampleConst ($C$) ;
12 $\hat{z} \leftarrow$ synthesize ($\tilde{C}$) ;
13 return $\hat{z}$;
```

D. Dataset

Each sample in our constructed datasets is a `<text,IR,layout>` triplet. The examples are shown in Table 7. The statistics of Web5K and RICO2.5K are shown in Table 9.

To ensure the quality and coverage of the labeled datasets, they were constructed in the following steps. First, we sampled unlabeled layouts from RICO (for RICO2.5K) and WebUI (for Web5K). Second, we recruited and trained annotators who were proficient in English and did not have professional graphic design skills. During training, annotators were asked to label 30 samples. Then, one expert checked their results and gave them feedback. This training was performed in 3 rounds to select 15 qualified annotators. Third, the selected annotators were asked to create textual descriptions and IRs for layouts. After annotation, five experts were responsible for quality assurance. They carefully evaluated each data sample to examine whether the description matched the layout and whether the IR represented all the constraints in the text. A sample was accepted only when more than three experts agreed with it. The inter-annotator agreement (IAA) is about 86%, which indicates the high quality of the datasets.

Dataset	Size	Avg. Textlen	Avg. Enum	Max. Enum	Avg. Cons
Web5K	4,790	40.6	9.6	78	11.3
RICO2.5K	2,412	52.9	6.4	20	8.8

Table 9. The statistics of Web5K and RICO2.5K. Size represents the number of `<text,IR,layout>` triplets. Avg. Textlen represents the average text length. Avg. Enum represents the average number of elements in the dataset. Max. Enum represents the maximum number of elements in the dataset. Avg. Cons represents the average number of constraints in IR.

E. Training Details

The hyper-parameters for the parse stage and place stage are shown in Table 10.

		WebUI				RICO			
Stage	Phase	Epoch	Batch Size	Warmup Steps/Ratio	LR	Epoch	Batch Size	Warmup Steps/Ratio	LR
Parse Stage	Training	100	8	500	1e-3	100	16	500	1e-3
Place Stage	Pretraining	100	8	0.1	1e-4	100	16	0.1	1e-4
	Finetuning	250	8	0.1	5e-5	250	16	0.1	5e-5

Table 10. Training hyper-parameters.

F. Quantitative Results of Different Element Numbers

We show the quantitative results of different element numbers on WebUI in Table 11. From the results, we find that FID, Overlap and UM become worse as the number of elements increases. While Align. and mIoU become better. We believe that's because most layouts in the dataset have a relatively small number of elements, so the generated layout distribution deviates from the real layout distribution as the element number increases, hurting the FID value.

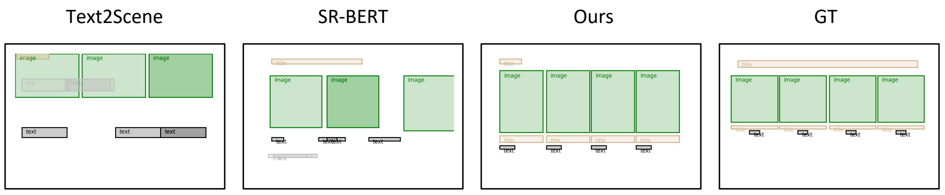
# elements	FID $\downarrow$	Align. $\downarrow$	Overlap $\downarrow$	mIoU $\uparrow$	UM $\uparrow$
$[1, 7)$	5.3173	0.0013	0.0878	0.6727	0.5694
$[7, 15)$	7.7672	0.0002	0.1698	0.6808	0.5325
$[15, 20]$	16.0709	0.	0.2618	0.8750	0.5186
Full Set (Ours)	2.9592	0.0008	0.1380	0.6841	0.5080

Table 11. Number of elements and corresponding quantitative results on WebUI.

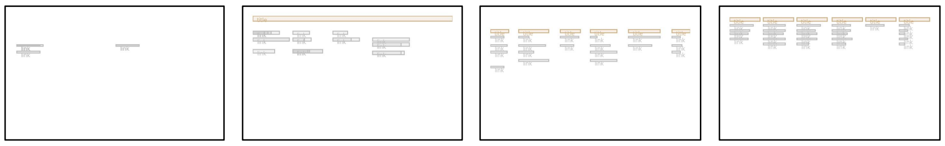
G. Qualitative Results

Here we show more qualitative results. In the first part, we compare parse-then-place with other baselines. In the second part, we demonstrate the generation diversity of our approach. Finally, we only show the layout quality generated by our approach.

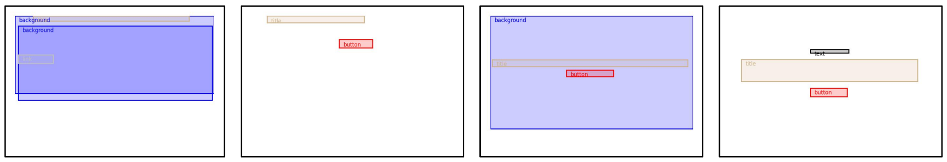
G.1. Compared to Baselines



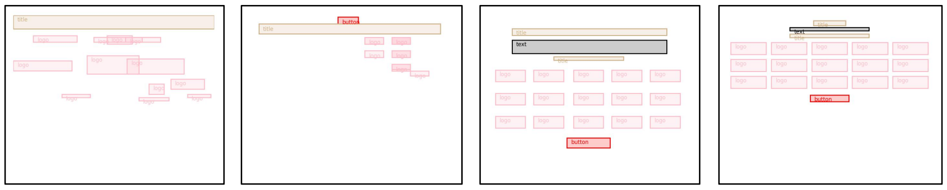
A page for selling mugs. The page should have a title, and 4 groups while each group has an image and two texts to show the photo, title, and price of the mugs.



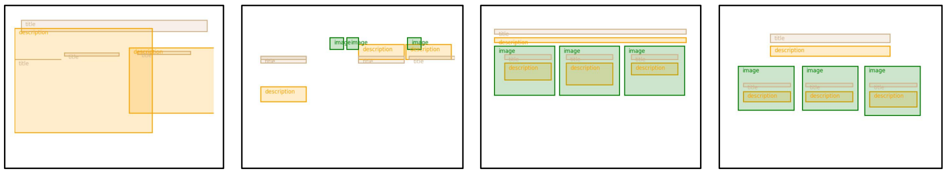
A page for navigation of a web. The page should have 6 groups while each group has some links for users to click to quickly view the products they need and jump to the corresponding page.



A page for introducing solutions to infectious diseases for an R&D department. The page should include one title "how the R&D fight against the diseases" and one button for users to discover more solutions.

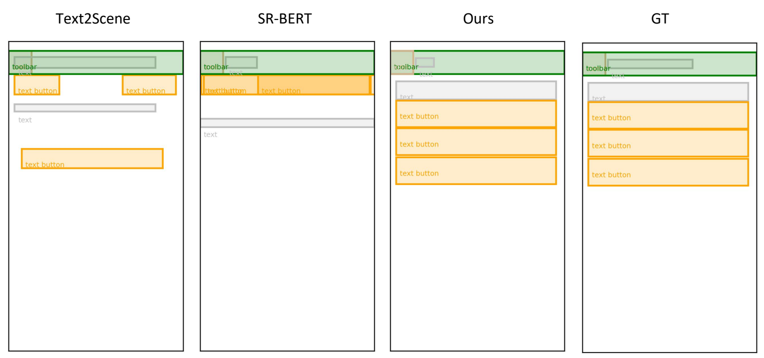


A page encouraging users to increase the exposure of their company. There is a title “COMPANIES” and a sentence "Open your company page and increase the exposure!", followed by another title "ALREADY 25304 COMPANIES REGISTERED". The page contains also 15 logos of different companies and a button "MORE COMPANIES" at the bottom.

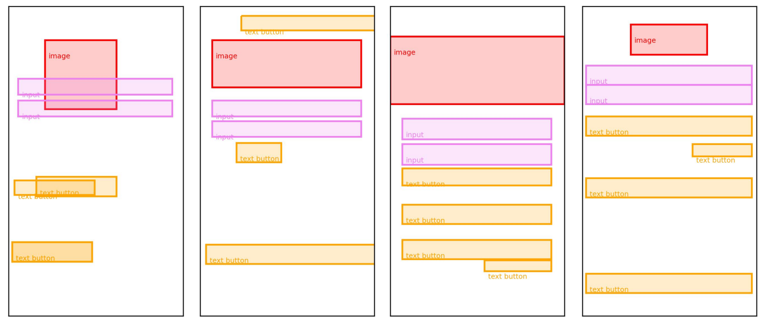


A page for introducing the service of the software. The page should have the title "Store and Manage All Your Files", a short description, and three groups of introduction information: "Access on all devices" "Share and collaborate", and "Unbreakable security". In each group, there should have a title and a further introduction.

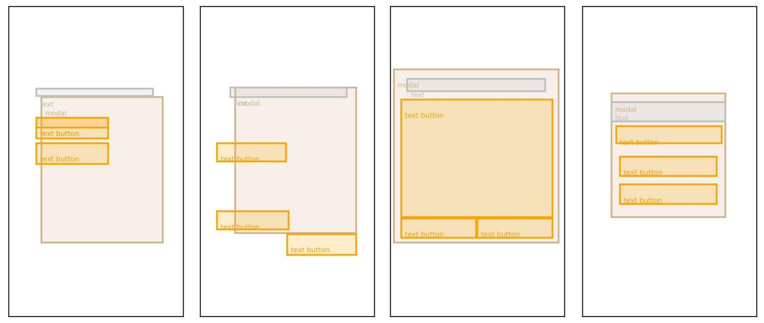
Figure 5. Qualitative comparison on WebUI. [Best viewed with zoom-in.]



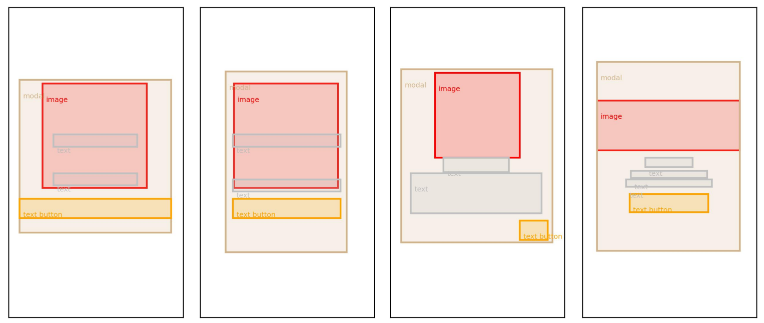
The page is for payment methods. On the top, there is a toolbar with an icon and a text on the left. Below the toolbar is a text and three text buttons for linking a payment method.



A page to log in or register for the app. On the top, there is an image showing the logo of the app. Below are two entries of inputs to enter email or username and password. Below are two entries of text buttons used to log in or used to find the forgotten password. Below is a text button used to register. At the bottom of the page, there is a text button referring to "I'll post property later".



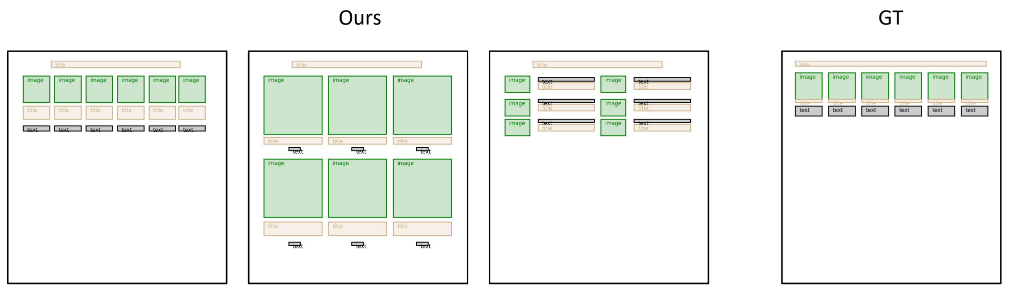
A page for tracking migraine symptoms . There is a modal at the center of the page with one text, which is placed in the upper portion of the modal, and three entries of text buttons inside it, first of which asks a question if the migraine is still present and the latter two are options of an answer, for example, yes.



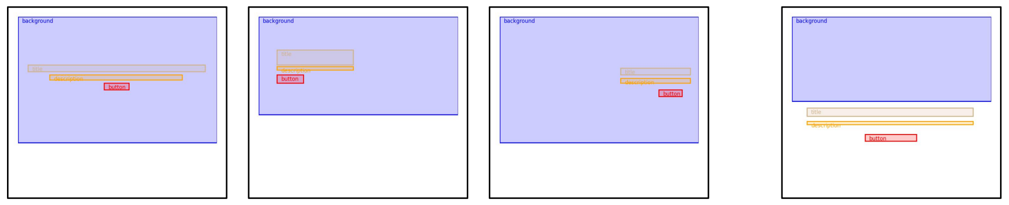
A page for updating the app. In the middle, there is a modal. In the modal, there is an image on the upper half with two texts below it. The following is a text button for updating.

Figure 6. Qualitative comparison on RICO. [Best viewed with zoom-in.]

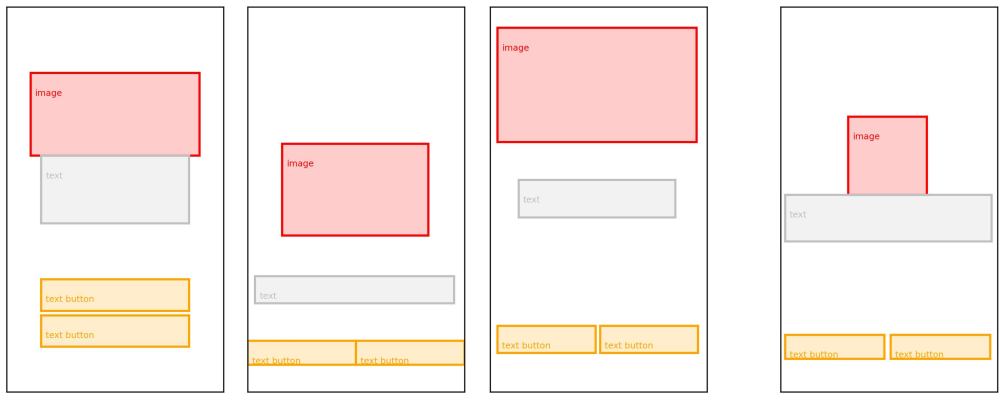
G.2. Layout Diversity



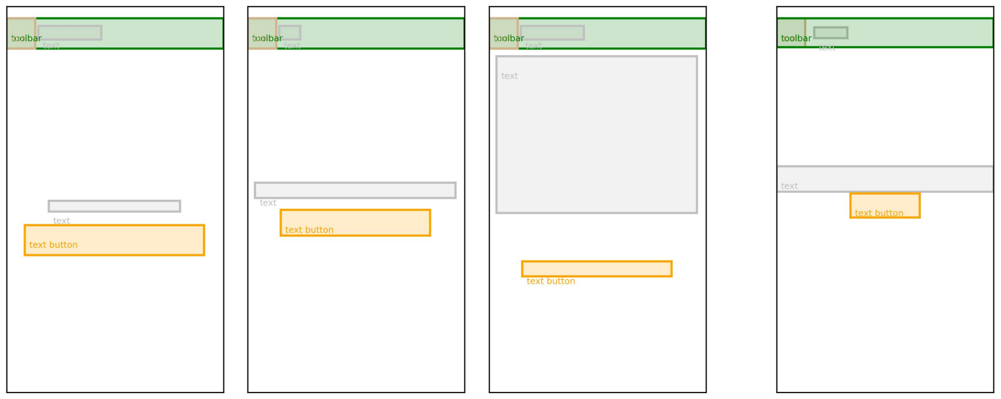
A page for showing different types of products. Besides a title named “Featured Collections”, there can be found six groups of information on the page. Each group contains a picture of the product, a product name, and the price.



A page showing the service of a company. There should be a title showing the aim of the service, which is to manage personal finance. There needs to be an introduction and a button for starting now.



This is a page for a gift. In the middle, there is an image and one text. At the bottom, there are two text buttons.



The page is used to select the payment type. There is a toolbar on the top with an icon and a text on the left. In the middle, there is a text and a text button for authorization.

Figure 7. Qualitative results to show the generation diversity. [Best viewed with zoom-in.]

G.3. Layout Quality

We show the layouts generated by our approach in Figure 8 and Figure 9. The results indicate that parse-then-place can generate high-quality layouts with well-aligned elements and small overlapping areas.

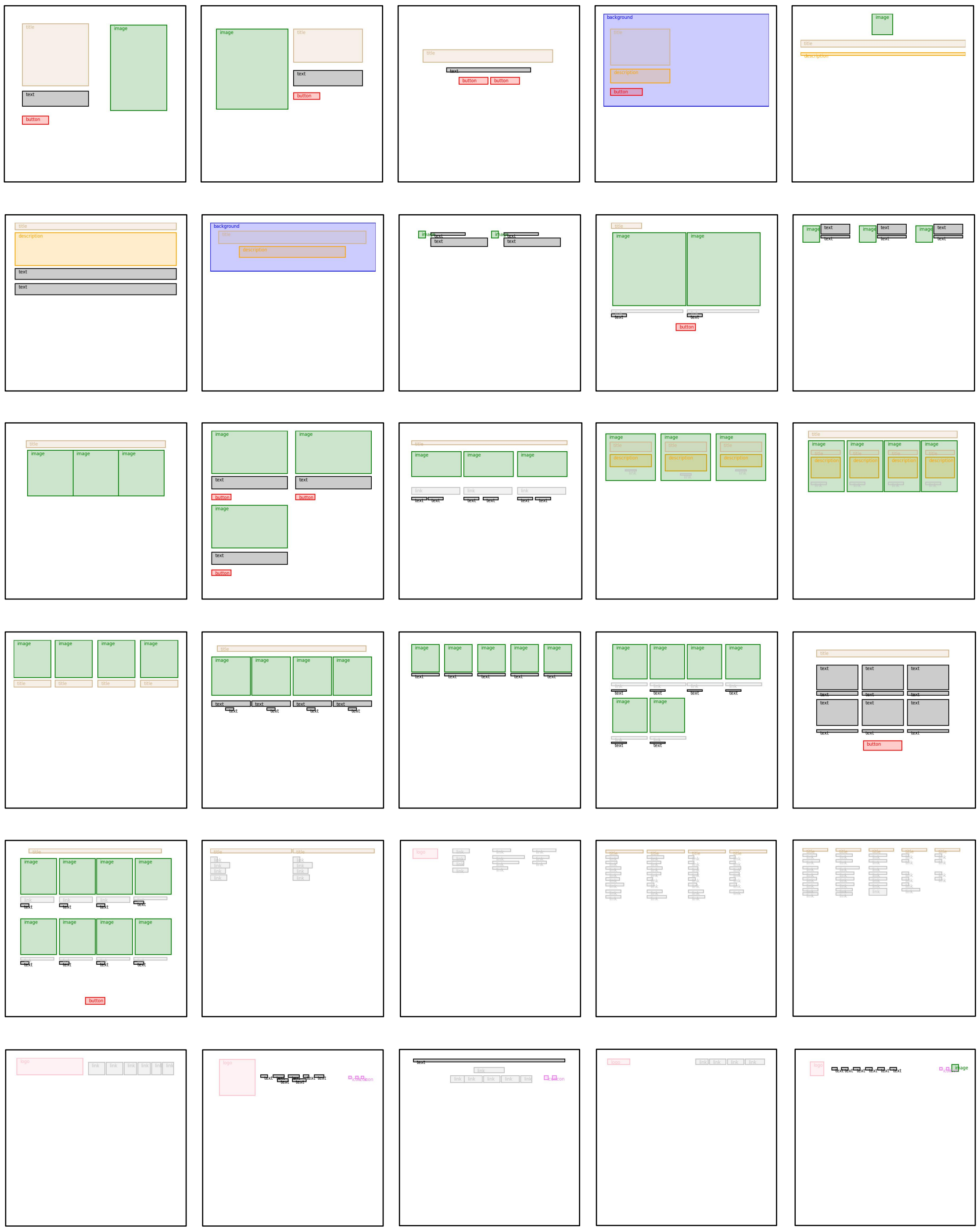


Figure 8. Qualitative results on WebUI to show the layout quality. [Best viewed with zoom-in.]

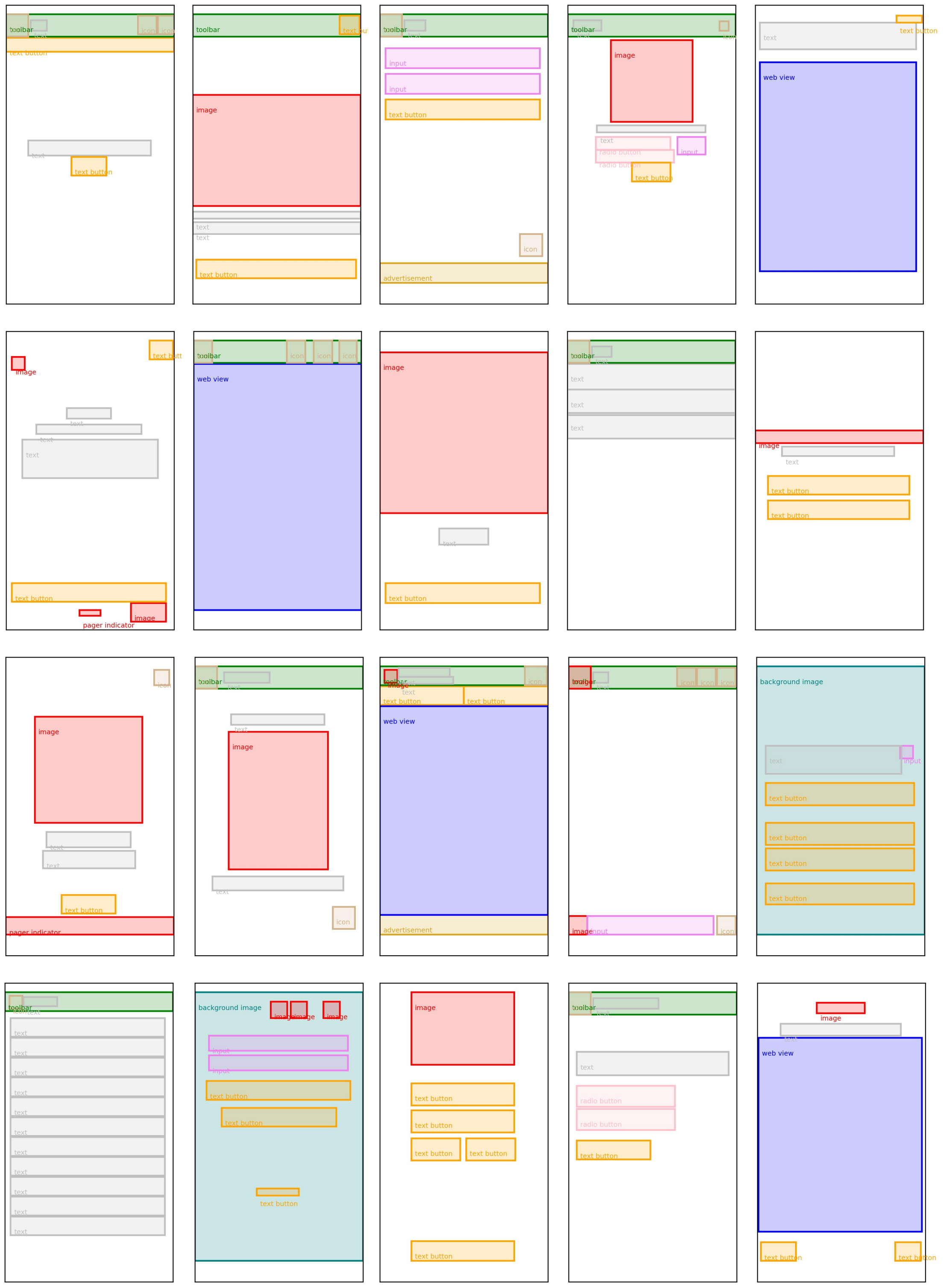


Figure 9. Qualitative results on RICO to show the layout quality. [Best viewed with zoom-in.]