

Contributing to React Magma



Define the problem

Narrator: Tim learns that it might help instructors if they could directly edit the title of an activity, instead of having to click the edit button to get to the form.

Narrator: He goes to the React Magma library to find the right component for this and realizes there isn't one.

Tim: Why didn't they anticipate my need for this? What am I supposed to do now?

Narrator: Don't worry Tim, there's a process for this.

First Steps

1. Tim can consider changing his solution to use what's already available in the library.
2. Tim should probably investigate whether another designer has already created a similar solution using React Magma components, or is currently working on a similar solution.
3. If all that fails, Tim should consult with the React Magma Team to see if what he wants to do will require the creation of a new RM component or the evolution of an existing component.
4. If it's determined a new or evolved component is necessary, lucky Tim! He gets to contribute to the React Magma Library!

Designing a Component

Designing a reusable component

- Make sure you're thinking about the use of this new component outside of your specific needs.
- Try to design the component to be open and scalable enough that it could be used by others and isn't too specific to any one product. If you can't, then maybe this component belongs in a component library for that product, instead of React Magma.
 - Example: Learning Path planks
- Consider whether this component could ever be placed on a dark background, and design for that as well.
- Design all of the necessary states, including but not limited to: Hover, focus, error and success messaging.

Design for Accessibility

Interaction

- You need to consider all of the ways users are going to interact with this component
 - Mouse and keyboard, no visual impairments
 - Touch-based device
 - Major visual impairments requiring the use of a screen reader
 - Physical impairments requiring the use of only a keyboard



Design for Accessibility (continued)

Color

- Color contrast ratio
 - Text = 4.5:1
 - Large text or visual elements important to the meaning or use of a component = 3:1
 - Disabled components or purely decorative visuals = no contrast ratio minimum
 - More info:
<https://www.w3.org/TR/WCAG21/#contrast-minimum>
- Don't use color as the only way to convey the meaning or value of something.

Mental impairments

- Simple design
- Clear and helpful communication
 - Labels, tooltips, clear language, tutorials, etc.
- As little cognitive load as possible

Responsive Design

Being Responsive means there must be some kind of experience for the user at any sized screen. This could mean:

- The component changes size, shape, etc. based on the screen size, but how you uses it stays mainly the same.
- The component changes size, shape, etc. based on the screen size, and how you use the component changes while the result is the same.
- At a certain screen size, the component is replaced by a different component designed specifically for smaller screens.
- At a certain screen size, the component (and therefore the functionality) is suppressed and it's explained to the user that you can't do that on that sized screen.

It's very common that starting with the design of something on a small screen, or at least keeping it very much in mind, will result in a simpler and easier to use solution on large screens as well.



Let's look at an example

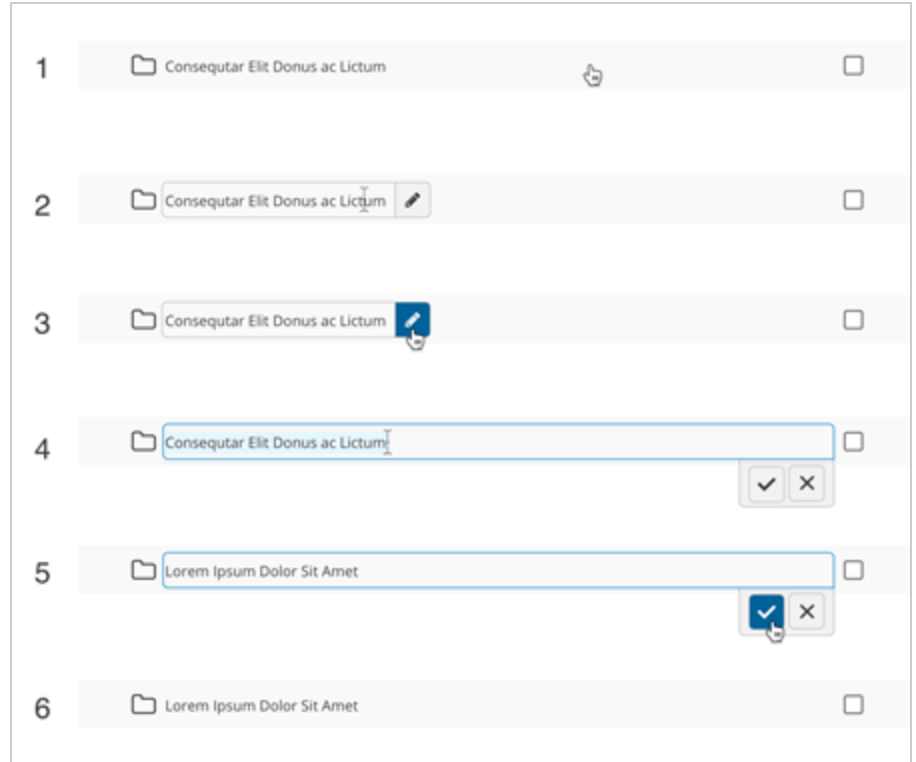
In-place Editing

Feedback

- Make sure you're using React Magma for already existing components and states
 - Buttons, inputs, focus state on inputs, hover on buttons

Questions

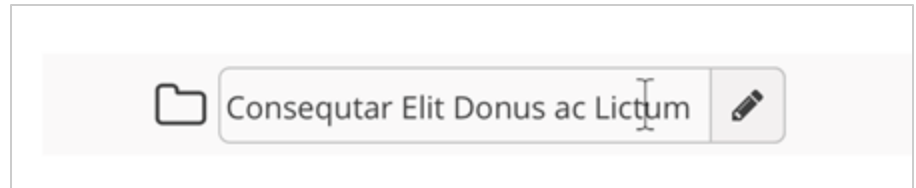
- Will you see that's it's editable upon hover or focus?
 - If hover, then how will users with touch screens know they can edit it? Can they edit it on touch screens?
 - If it's on focus, how will anyone know they can edit it?
- Is this the only way to edit the title?
 - If so, then the previous questions are probably more important.
 - If not, then the previous questions are probably less important.



In-place Editing (continued)












- What kind of error handling will these need, and how will errors be displayed?
- What level of confirmation do they need when saving the change?
- What will the experience be for only using a keyboard?
 - Will tabbing to the title make it immediately editable?
 - Should they be able to press enter to accept their change or would it be best to tab to the button?
 - Should they be able to press Escape to cancel their changes?
- What will the experience be for a screen reader? Will there be hidden labels? What will those labels say?
- In your context for showing multiple levels of folders, and I assume you can expand and collapse them, what are you clicking on to expand/collapse and what are you clicking on to edit the title?

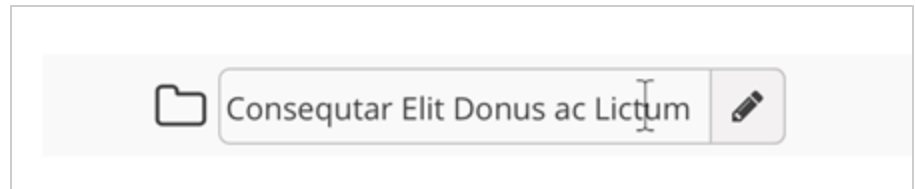
	Student Content	Paid Content
Art Through the Ages 13th Ed	<input type="checkbox"/>	<input type="checkbox"/>
↳ Lorem Ipsum Dolor Sit Amet Consectetur Adipiscing Elit	<input type="checkbox"/>	<input type="checkbox"/>
↳ Consequat Elit Donus ac Lictum	<input type="checkbox"/>	<input checked="" type="checkbox"/>
↳ Nullam Eget Ultrices Turpis, Id Sagittis Mi	<input type="checkbox"/>	<input checked="" type="checkbox"/>
↳ Nam Rutrum Vulputate Feugiat	<input type="checkbox"/>	<input type="checkbox"/>
↳ Donec Pellentesque est eu Tortor	<input checked="" type="checkbox"/>	<input type="checkbox"/>
↳ Aliquam Quis Posuere Dui	<input checked="" type="checkbox"/>	<input type="checkbox"/>
↳ Suspendisse Efficitur Purus Tincidunt Eros Retrum ac Ultrices	<input checked="" type="checkbox"/>	<input type="checkbox"/>
↳ Donec Sollicitudin Orci eu Aliquet	<input checked="" type="checkbox"/>	<input type="checkbox"/>
↳ Sed Vehicula Justo ut Laoreet Lobortis	<input checked="" type="checkbox"/>	<input type="checkbox"/>
↳ Curabitur at Turpis	<input checked="" type="checkbox"/>	<input type="checkbox"/>



In-place Editing (continued)

- Also in the context of these folders, if a person is using a keyboard, how many more things are they going to have to tab through if they're just navigating through the list and don't want to edit anything? Or they're trying to get to the appropriate check boxes.
 - I would be careful to make sure that the thing I'm intending to make something easier for some, doesn't actually make it more difficult to perform the primary actions for others.
- Is anyone talking about using something like this anywhere else? I've heard of in-place editing on the learning path planks, so will this design work for something like that?
- What if the title of something is also the link taking you into that item? Does that disqualify them from using this component?
- I could see this being used on a dark background, so please make sure you design the inverse version of it.

	Student Content	Paid Content
 Art Through the Ages 13th Ed	<input type="checkbox"/>	<input type="checkbox"/>
 Lorem Ipsum Dolor Sit Amet Consectetur Adipiscing Elit	<input type="checkbox"/>	<input type="checkbox"/>
 Consequat Elit Donus ac Lictum	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 Nullam Eget Ultrices Turpis, Id Sagittis Mi	<input type="checkbox"/>	<input checked="" type="checkbox"/>
 Nam Rutrum Vulputate Feugiat	<input type="checkbox"/>	<input type="checkbox"/>
 Donec Pellentesque est eu Tortor	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 Aliquam Quis Posuere Dui	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 Suspendisse Efficitur Purus Tincidunt Eros Retrum ac Ultrices	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 Donec Sollicitudin Orci eu Aliquet	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 Sed Vehicula Justo ut Laoreet Lobortis	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 Curabitur at Turpis	<input checked="" type="checkbox"/>	<input type="checkbox"/>



Getting Your Design Approved

- At any point along the way, you can share your designs at UX design critiques.
- You can also consult with the React Magma team directly if you're looking for input on the design or technical aspects of the component.
- Present your final design to the React Magma team for approval.



Final Deliverables

Sketch

- Create components with the appropriate overrides, states, options, etc.
- Add to the appropriate library through Abstract, and submit for approval

Details for development

- Create an Issue on Github
- Provide all of the details needed to implement your solution including:
 - Links to mockups in either inVision or Abstract
 - Write acceptance criteria

Documentation

- New component requires all new documentation.
 - Use existing design guidelines as a template to detail sections like options, types, variations, sizes, etc.
 - Provide good and bad examples
- Evolved component might require updates to existing documentation
- Documentation format = Write them up in Word/Google Docs and provide images. The React Magma team will add them to the docs site.