

Class Structure

Young Chefs for iPad

Experiment

<code>[Module]</code>	An array of modules to appear in the experiment
<code>Notebook</code>	An instance of a Notebook, which contains lab notes
<code>int currentModule</code>	An index for which modules is currently presented
<code>assetsForModules</code>	
<code>[String : AnyObject?]</code>	Dictionary that contains information to be presented in the lab report (e.g. responses, notes, photos)

ExperimentViewController

<code>experiment</code>	An instance of the experiment
<code>@IBOutlet notebookView</code>	The notebook which can be pulled into view from below screen
<code>@IBAction navigationButtons</code>	Forward and backward buttons
<code>func loadNextModule()</code>	Loads next module

AbstractModule: UIViewController

<code>isDone()</code>	Returns true if the user has completed the desired activity (e.g. answered all the questions)
<code>experiment?</code>	The module may require an instance of the experiment
<code>let XibToLoad</code>	The Module's corresponding xib file

Subclasses:

- **DescriptionModule**

<code>@IBOutlet textView</code>	Rich text for the description / article
<code>init(<i>content</i>)</code>	Construct it with whatever content it requires

- **QAModule**

<code>@IBOutlet questionDisplay</code>	The view that poses the question
<code>init(<i>Question</i>)</code>	Construct it with the <code>Question</code> object

QAModule uses the `Question` class.

- `struct Question:`

<code>question</code>	The question to ask
<code>responseKey</code>	The key used to access the response in the lab report

- **StepsModule**

- based on `PageViewController`

- **LabReportModule**

<code>[Recap]</code>	An array of “Recap” piece that will make up the report (e.g. hypotheses, questions answered)
<code>init([Recap])</code>	Construct the module with the Recaps

LabReportModule uses the `Recap` class.

- `struct Recap`

<code>responseKey → Value</code>	The value associated with a <code>responseKey</code> , used in <code>Experiment</code> ’s dictionary (see above)
<code>tag</code>	A header that the response will be listed under

- The Recaps will be formatted into a list with tags as headers.

For example,

`[Recap blue, Recap red, Recap bob, Recap alice]`

will be formatted as:

- **Colors**

- blue

- red

- **People**

- bob

- alice

- In the app, headers/tags will likely be things like “Hypotheses.”