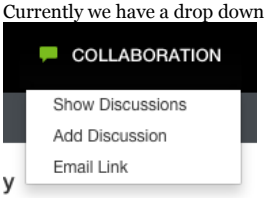
Discussion Thread Project

Our product contains a feature that allows users to initiate a discussion, or comment thread, on most system objects. So for example a query listing, or record's detail report, could have user's comments attached.

Summary of Existing Product:

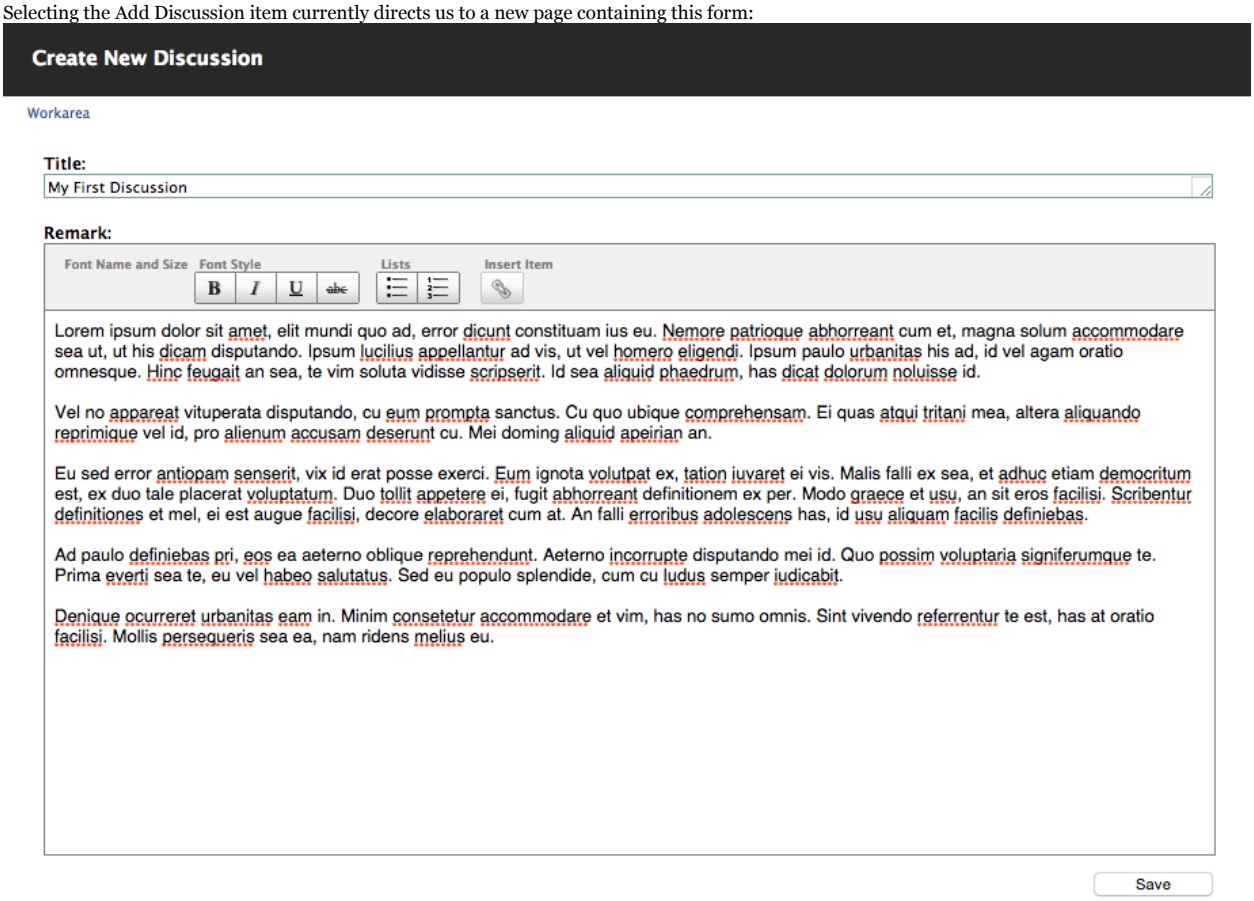
Currently we have a drop down menu in our header for "Collaboration":



y

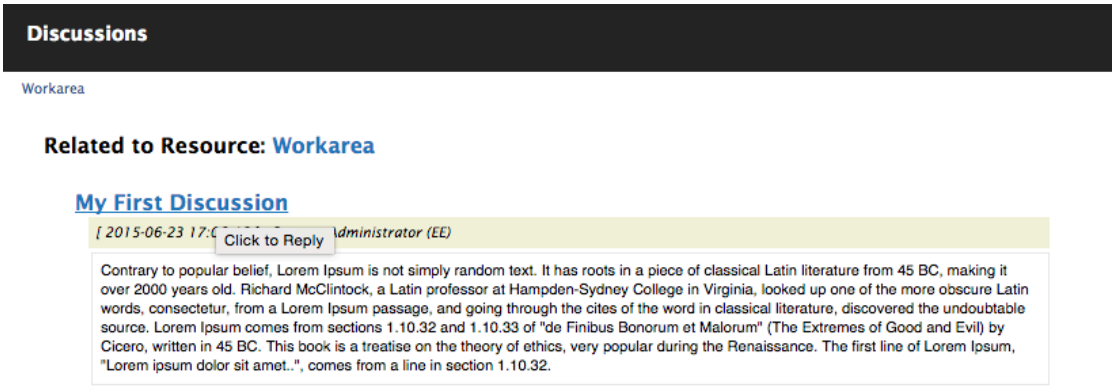
This allows the user to view any existing discussions (comment threads, I'll use the terms interchangeably), or to create a new one. You can ignore the "Email Link" menu item, it does not pertain to this project.

Selecting the Add Discussion item currently directs us to a new page containing this form:

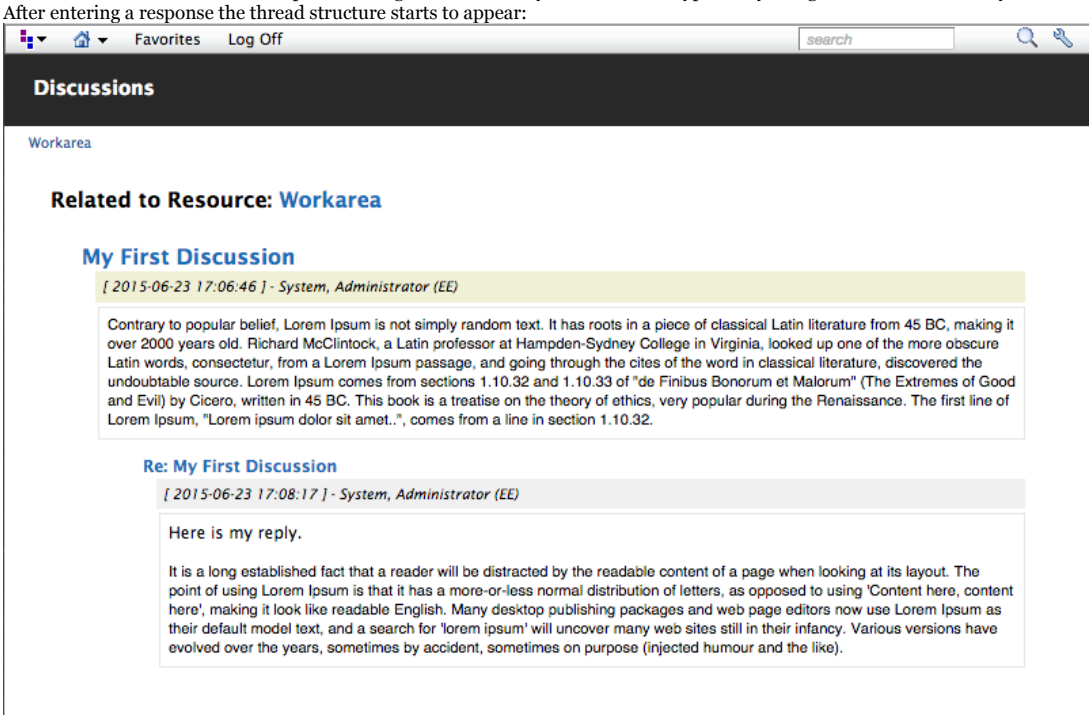


The user supplies a Title, and then enters their Remark in the editor. The editor in this screen shot is the YUI v2.8 rich text editor, which produces HTML markup. The formatting is not necessary for this project, we can use either a large text area, or perhaps a better choice would be to re-use the ace editor we utilized in an earlier project but without the sidebar and line numbers. Have it look more like just a plain text editor.

When the user is finished they click the Save button, and that submits the form to the server. As a result the server creates a discussion record in the database and returns the following page:



This is the result of the server's response. A single discussion, if you click on the hyperlink you'll get the same form and you can enter your response. After entering a response the thread structure starts to appear:



Of course this can continue pretty much without limit, but in practice rarely goes beyond 15-20 entries per item.

These interfaces are produced by transforming a dynamically generated XML document through a couple of different XSLT stylesheets. I'm providing an XML file to show a sample of the source data, and the three XSLT files.

1. discussion-form.xslt
2. discussion-report.xslt
3. discussion-thread.xslt

discussion-form.xslt creates the interface for the user to create a new discussion, or create a reply.
discussion-report.xslt creates the interface to display the existing discussions and uses an *xsl:include* tag to bring in the **discussion-thread.xslt** file to build the visual hierarchy.

The XML source example applies to the discussion-report, and discussion-thread. The discussion-form receives data in the form of *xsl:params* added by the server side XSLT processor. The XML content for this XSLT is simply an empty single node document **<env></env>**

What we want to do:

That is where we are today. Our goal is continue using the same server side processing, and rework the existing XSLT, to create a more modern look and feel. We also intend that instead of launching users into a new page, we will display this on an "overlay" of the existing page. The "create new discussion" form should launch in the same page, wrapped in a Bootstrap modal. The report will also be displayed in a "page overlay" fashion but probably will not actually use Bootstrap modal component. What matters for this project is that we're able to create the new XSLT stylesheets, and we will take care of launching the functionality for reporting by requesting the URL and loading that into an overlay.
http://54.205.44.208/ee/request/discussion?rel=88&id=1950
However the create function, which uses the following URL should load the server response into the BS Modal.
http://54.205.44.208/ee/request/discussion?v=form&type=88&tid=1950

The first URL displays existing discussions, but the form of the URL is not important to this project I don't think. The second one responds with the "create new discussion" form.

We also want to update the look and feel of the discussions, I think the Disqus form is very nice, and we can probably use the Bootstrap Media list as a foundation, it is reasonably close. At this point we are not supplying images (such as user icons), but we may so we need to account for that. It maybe that we take care of that on the server, and just include a default when none is available. I will update you on this when we've firmed up the requirement.

Also we need a better method, one more obvious to the user, to invoke replies. I think I would like the "reply" functionality to appear "inline" when the user clicks a link. We can save this until some of the other work is complete, I'll have to figure out exactly how we pass the exact data you're going to need to create a reply, to the client. It probably means adding some additional data-* HTML5 attributes in the markup.

Following is a screenshot from the bootstrap site illustrating their Media List:

Media list

With a bit of extra markup, you can use media inside list (useful for comment threads or articles lists).

EXAMPLE

64x64 **Media heading**
Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

64x64 **Nested media heading**
Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

64x64 **Nested media heading**
Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

64x64 **Nested media heading**
Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

Copy

```
<ul class="media-list">
  <li class="media">
    <div class="media-left">
      <a href="#">
        <img class="media-object" src="..." alt="...">
      </a>
    </div>
    <div class="media-body">
      <h4 class="media-heading">Media heading</h4>
      ...
    </div>
  </li>
</ul>
```

I think that is an acceptable presentation, but I also want to encourage you to suggest any ideas you have for enhancement. The display and formatting are all open for change, we just want to maintain all of the data.