CATCH Library Lab Progress Report #2. [June 1, 2013]

This is the second progress report on the CATCH Library Lab project.

1. Background work

- Researched front-end open-source UI and back-end storage solutions for the CATCH Hub
- Identified and hiring front-end and back-end developers and are in process of discussing and prioritizing final deliverables
- Researched back-end cloud servers and decided on Rackspace open cloud server for hub
- Gathered requirements for front-end UI modules
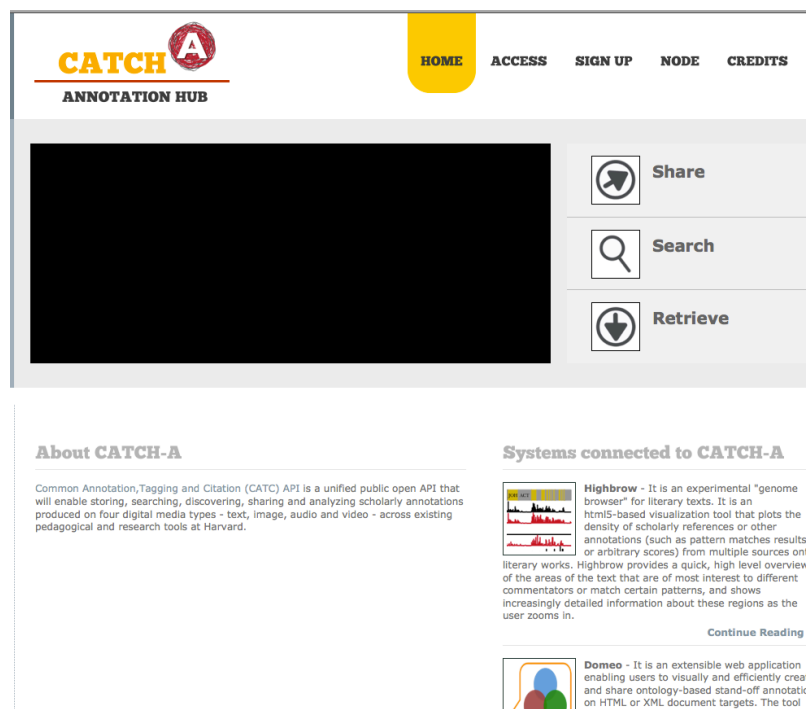
2. Development

a. Administrative Modules

CATCH Module
(https://github.com/annotationsatharvard/catcha)

An umbrella module that contains the public pages and loads the other modules such as AfSecurity and AfShared, and will load soon to be developed the Persistence (storage and retrieval of annotations) and Search modules.

Includes the public pages of the website (hub info, about and homepage) and allows to login with both username/password and OpenIDs. The login process relies on the AfSecurity module.

AfSecurity Module
(https://github.com/annotationframework/AfSecurity).
It allows for:
- Registration of Users and Groups
- Registration of System that can interact with the CATCH hub
- Management/moderations of accounts/groups and systems requests

The AfSecurity module allows for three kinds of users' roles: admninistrators/managers/ users. Each role gives access to different administration features.

The participation of a user to a group can also be qualified by a role (I could be the administrator of a group or a simple member)

The registry of the systems that can interact with the Hub can include, for each system, one or more administrators, the list of the groups and/or users that the external system can access the annotation of. A flag that determines if the external system can access the whole pool of the public annotations or not. We still need to add the parameter that force the streamed annotation to be anonymized.

AfShared Module
(https://github.com/annotationframework/AfShared)
Contains the templates of the graphic element and contains the default settings of the application. Includes some basic utilities and shared CSS material
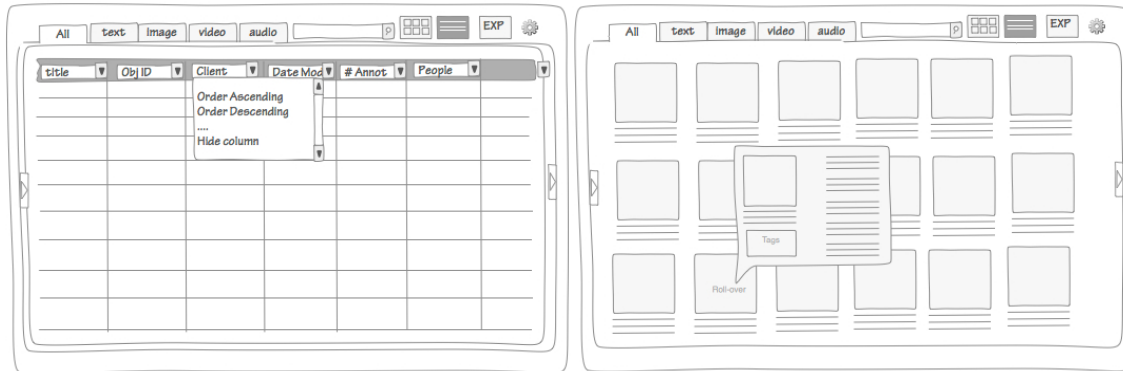
b. Stand-alone annotation UI modules

Produced design document for 3 front-end interoperable javascript UI modules along with corresponding wireframes. These modules retrieve and display baseline information (metadata) about annotated objects and annotations.

All modules allow exporting the data displayed in multiple open formats.

Grid-list view of annotated objects [AnnotationObjectGrid.js]

This is a stand-alone grid-list module that allows the connection to the annotation hub and retrieve a list of annotation objects based on the results of a search query. The modules has two view states grid and list view. Data in the views are customizable by showing or hiding data columns.

List view of annotations [AnnotationList.js]
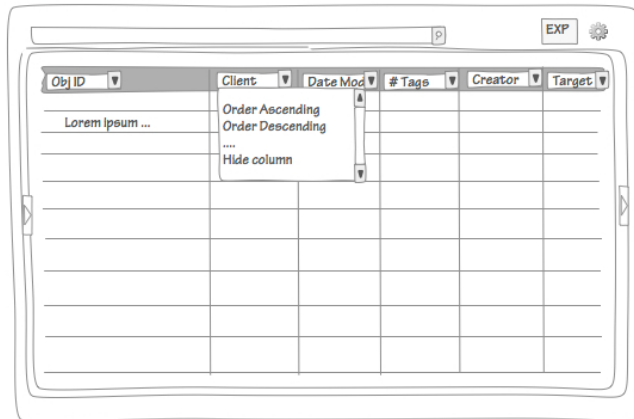
This is a standalone module that allows the connection to the annotation hub and retrieve a list of annotations based on the selection of an individual annotated object or the results of a search query (search UI based on Annotator.js for example). When used in conjunction with the AnnotationObjectGrid.js module (see above) it will display the data associated with the selected annotated object in AnnotationObjectGrid.js.

Radial Node Connection of annotated objects and annotation [AnnotationRadialTree.js]
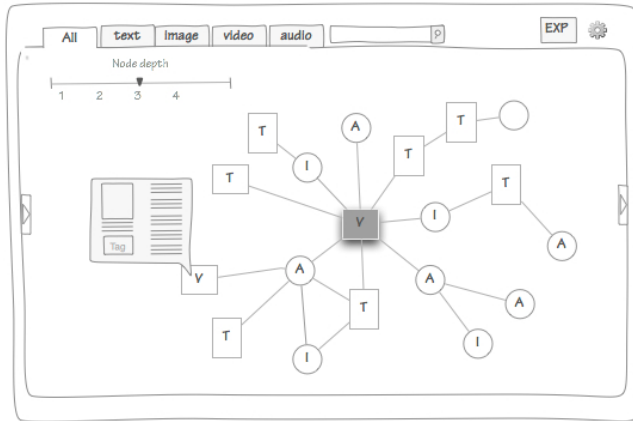
This is a visualization module that shows the relationship between annotated objects and annotations. Two dimensional display of icons of annotated objects in a radial map or radial tree with lines connecting to annotation icons that expand outward.The view displays the objects based on the query search. The UI also allows to reveal connected annotated objects by degrees of freedom (various levels of connections).

<u>UI Module interoperability and interdependence</u>

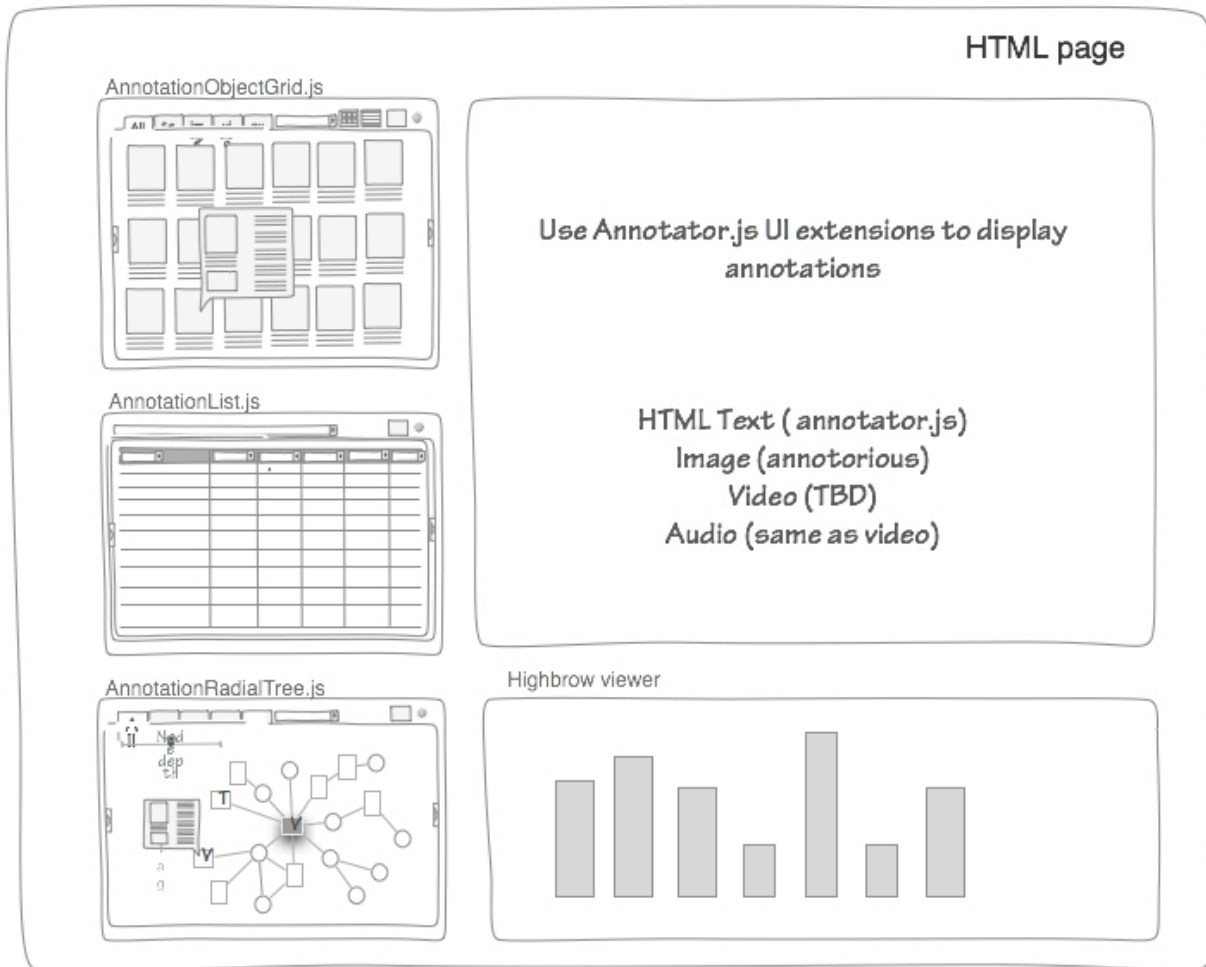Each module can connect independently to the back-end database hub, or can be self-standing. If more than one module is ti be used on a same Web page, there will be a an API to interoperate by simply registering the modules. This will allow changes in one module to be reflected in the other modules.
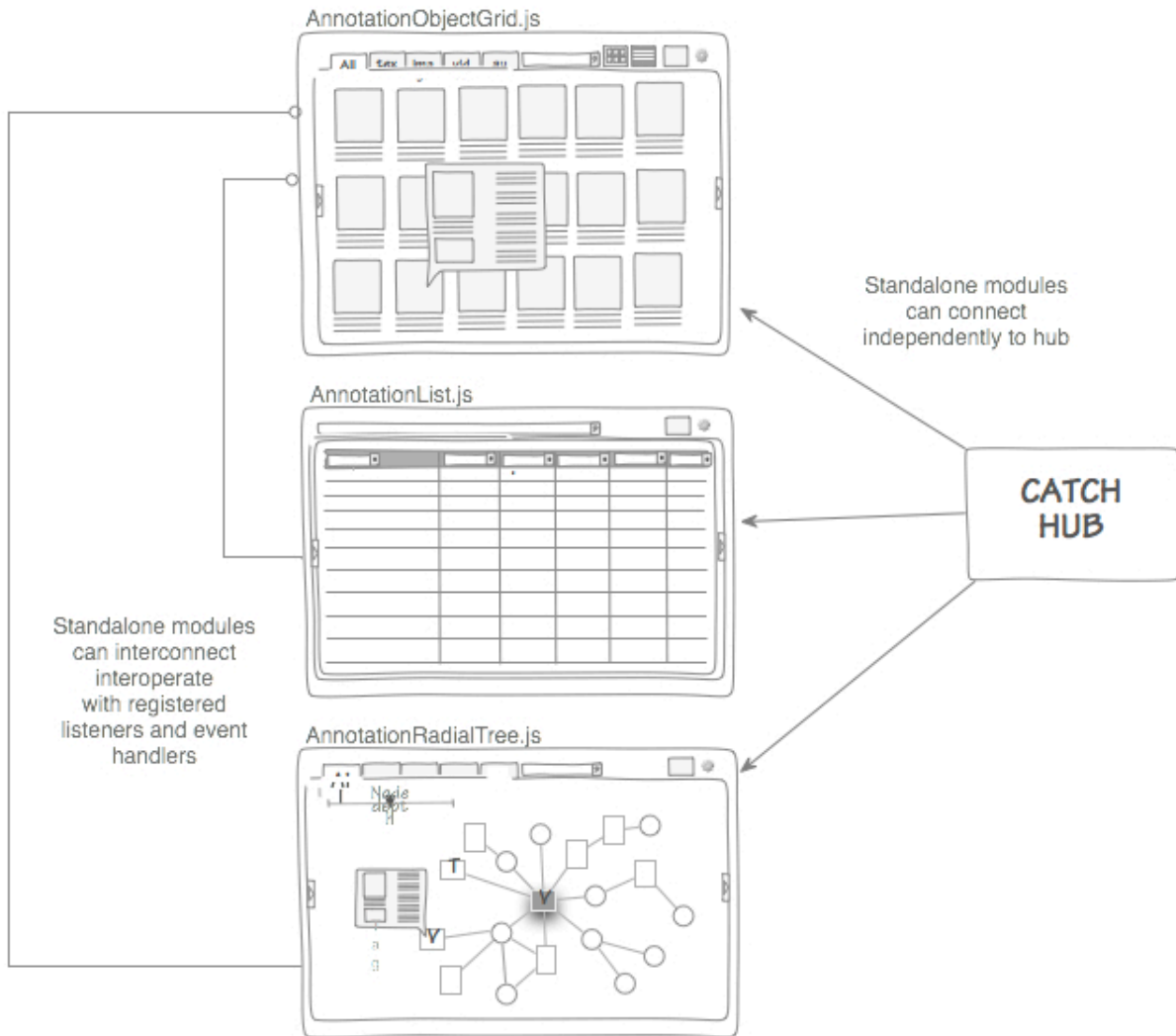
UI Module interoperability in a Web Page

3. Planning

For this phase we will focus on implementing the above mentioned designs for the stand-alone Javascript UI modules.

On the administrative end we are planning the development of:

- Persistence (storage and retrieval of annotations)
- Search modules (searching annotations)

4. Expenses

- Rackspace cloud server: $42
- Expenses for front and back-end developers will begin this next phase. These upcoming expenses will consume the majority of the CATCH budget