Appendix 7.

Proof of Concept Prototype

William Ulate, Marcela Mora, Trish Rose-Sandler and Paul Smock

INITIAL PROPOSAL

Initially the proposed plan was to choose one of the existing tools analyzed, according to the feasibility assessment. The chosen tool would be installed as a proof-of-concept on how an existing annotation tool could support the different types of annotation needs that the botanical users may have. This prototype would run against a digital library to test the integration and effectiveness of the requirements compliance. Several annotations according to the needs identified in our previous activity would be the input to test the prototype's efficiency. The results and evaluations of such activities would be included in the outcomes assessment of the following task.

From the analysis done, we concluded that much of the functionality needed to satisfy the annotation requirements identified is already included, in different degrees, in the existing tools analyzed. However, it also shows that different existing tools had different approaches on how they provided annotation services, some of those had complementary strengths but not one single tool alone satisfies all the requirements enlisted, even when some of them do excel in their domain.

Therefore, instead of choosing and installing one single tool only, we tested several of the most promising tools (as they were available at the moment) and concentrated our development efforts to draft a prototype that could help us understand how these tools' behavior and their integration could fit in the way current Digital Libraries function. This allowed us to extract some best practices for any fully developed tool to consider when attending the minimum botanists' requirements identified.

FIRST PROTOTYPE WITHIN BOTANICUS

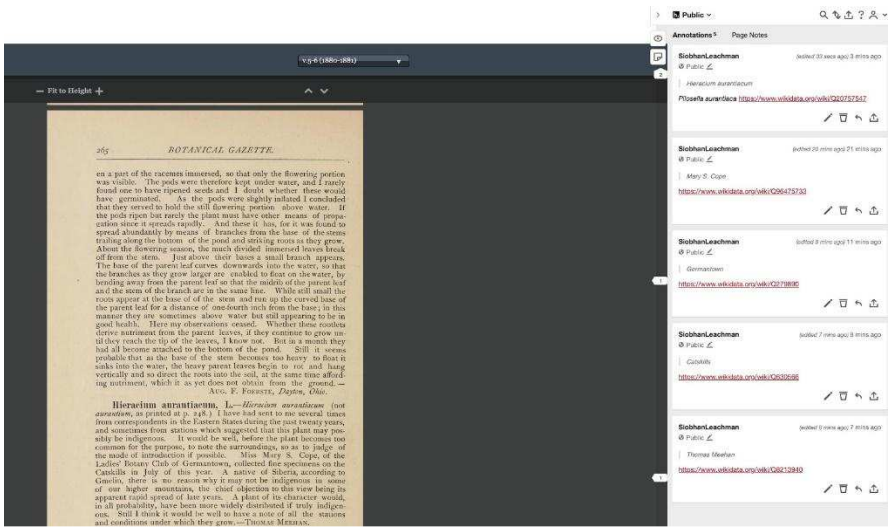
Our first approach was to create a simple prototype with basic annotation functionality, embedded into Botanicus, Missouri Botanical Garden's Digital Library. The tool developed was included in the code of the internal testing installation of Botanicus. As opposed to tools like Hypothes.is that are able to annotate text-based documents (including HTML and PDFs), our exploratory tool was developed to allow the annotation of images, the format in which a large percentage of the current information is conveyed in Digital Libraries like Botanicus and the Biodiversity Heritage Library (BHL).

The experience with the prototype exposed certain assumptions about the tool we were conceiving. Initially, we thought it could work embedded in any Digital Library, interacting with their interface to display the target and the body of annotations while storing them in a single accessible Annotations Repository. But one of our early findings in the process of developing the prototype was that a solution to interact within the framework of some of the existing Biodiversity Digital Libraries would require the modification of the library's code. Such a solution, while effective, would be too specific and not replicable for a general approach, which was our main interest for the project. An alternative would be to place the annotating functionality in an overarching layer on top of the whole interface, an approach used by some of the existing tools, generally using a web browser and following the recommendations of the W3C's Web Annotation Working Group. Hypothes.is and Pundit work in this way, as a web browser plug-in. Other tools, like Annotorious, also provide a way to incorporate the general annotation functionality as a JavaScript library within the website, but it would depend on each Digital Library to modify their codebase to include the Annotations logic.

Test installation of the Botanicus Digital Library with Basic Annotation Functionality

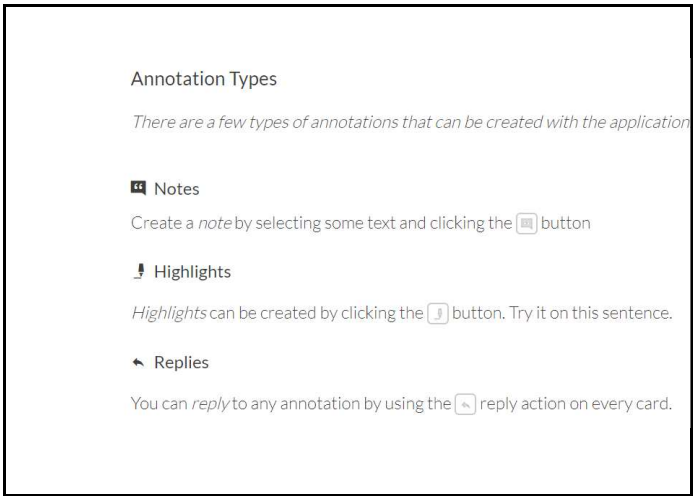
INSTALLATION AND ASSESSMENT OF TOOLS

As specified in the Annotations Tools Assessment, several tools (Hypothes.is, Pundit and Pundit Pro, Recogito, Annotorious) were installed and tested in Botanicus, in text-based websites, and in PDFs, giving emphasis to those tools particularly promising in fulfilling most of the user needs previously identified. Here are some examples of how tools' interfaces address those requirements:

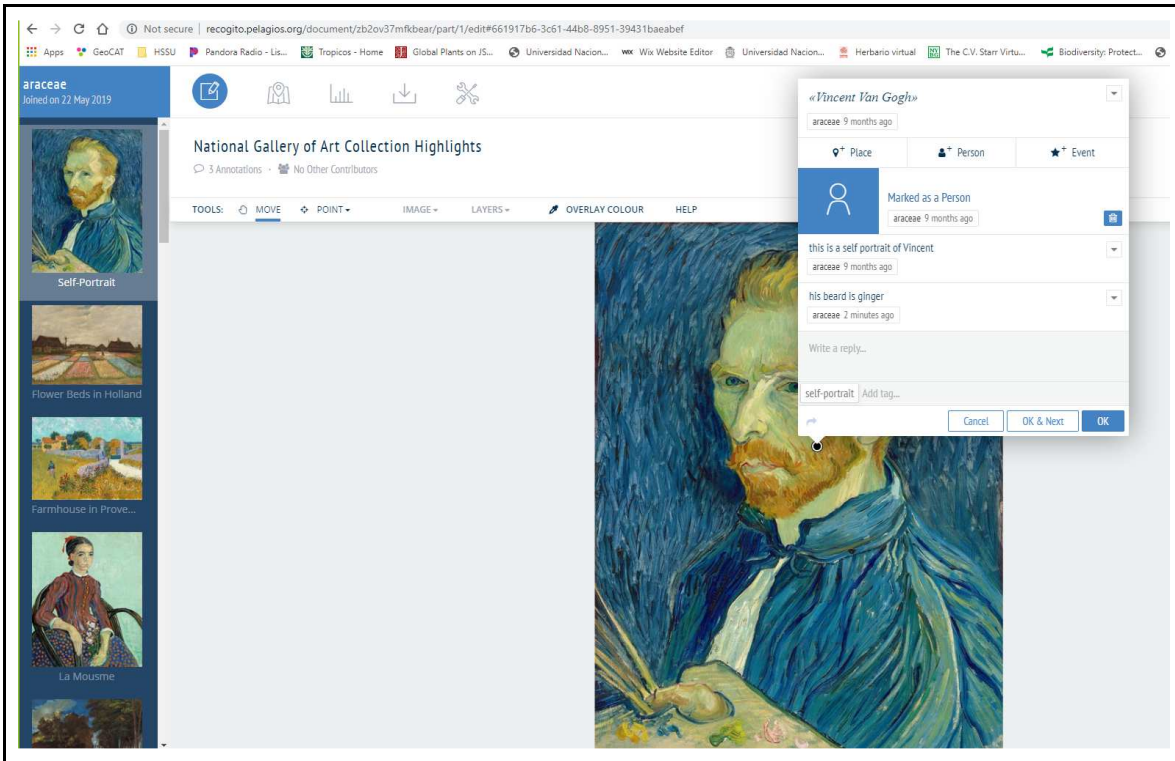


Several annotations in Hypothes.is anchored to (the OCR text) of a page image in the Biodiversity Heritage Library; the interface allows the user to edit, delete or reply her own annotations

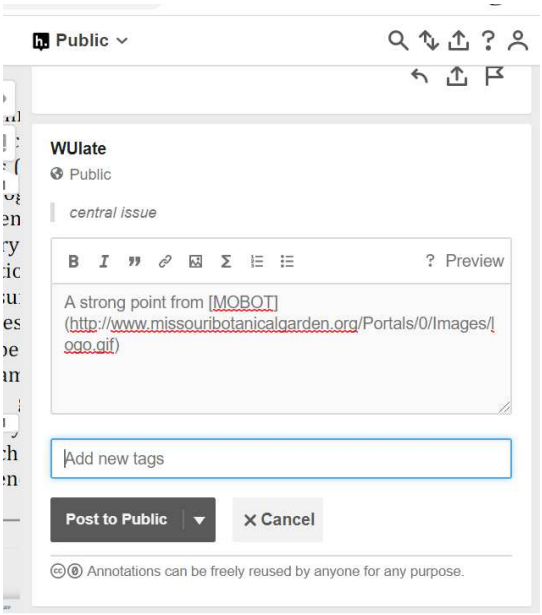
Assess, reply and flag mechanism shown in Pundit Pro interface



Explanation of three Annotation Types available in one of the tools

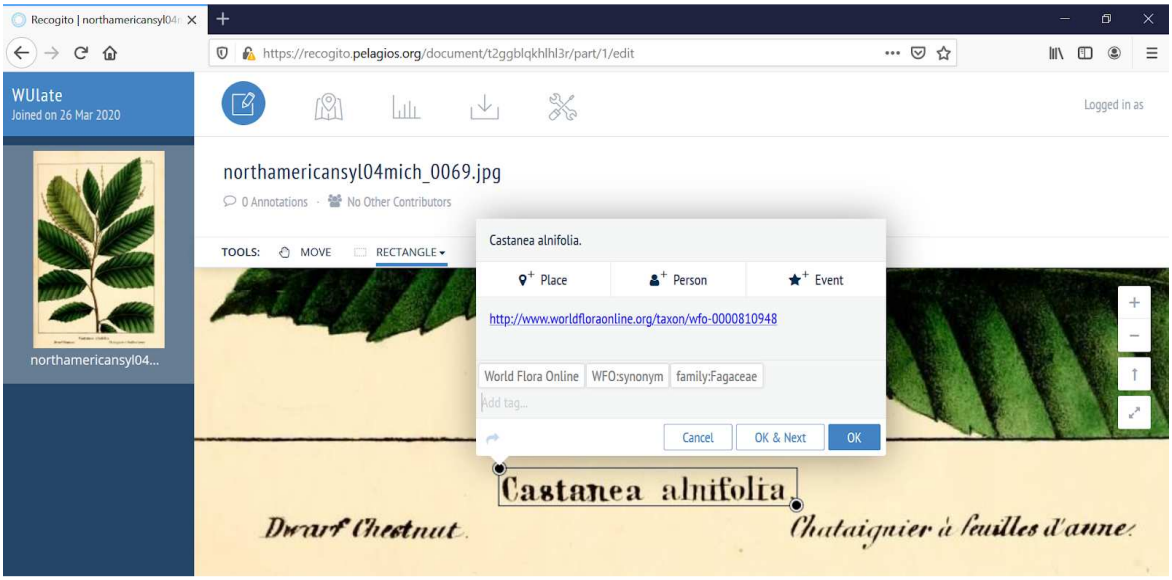


Interface of Recogito by Pelagios showing an annotation anchored to a point in an image, linked to a *Person* (*Vincent Van Gogh*) with 2 more reply annotations



Rich Text editor interface showing how to add tags to the body of a public annotation (containing a URL) in Hypothes.is

Recogito was initially developed to annotate Maps, it also allows the annotation of images and it recognizes certain entities like Places, Persons and Events (somehow similar to what systematists do).



Interface of Recogito (Pelagios Network) showing an annotation with a scientific name anchored to a region of an image with a reply containing an URL and 3 tags assigned.

Internet Archive has a test implementation (lab) where they make their content available in an Image layer and present it using an implementation of SeaDragon. For the example shown above, the IIIF manifest of the book is at

http://iiif.archivelab.org/iiif/northamericansyl04mich/manifest.json

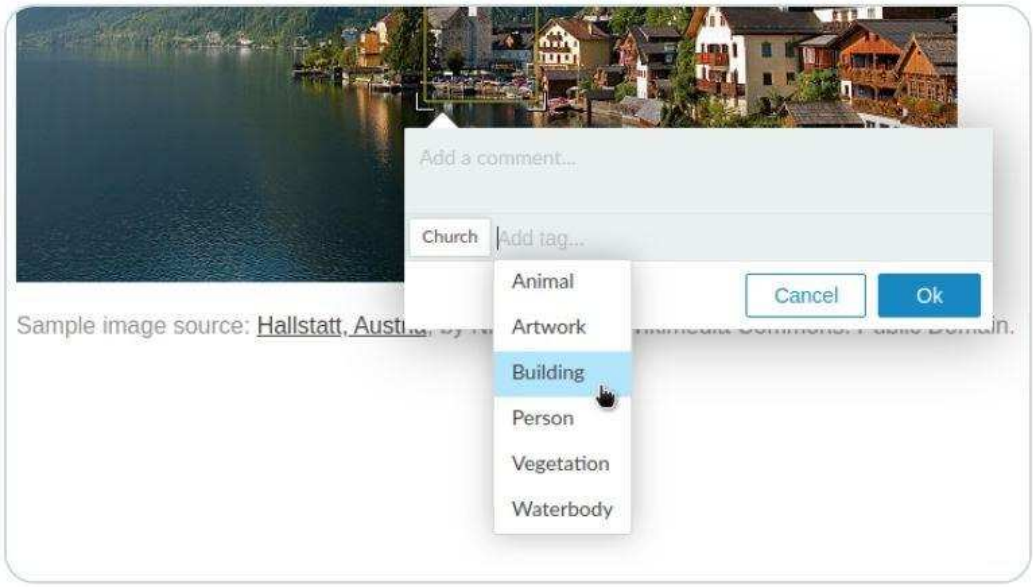
The manifest of the page is at

https://iiif.archivelab.org/iiif/northamericansyl04mich$69/manifest.json.

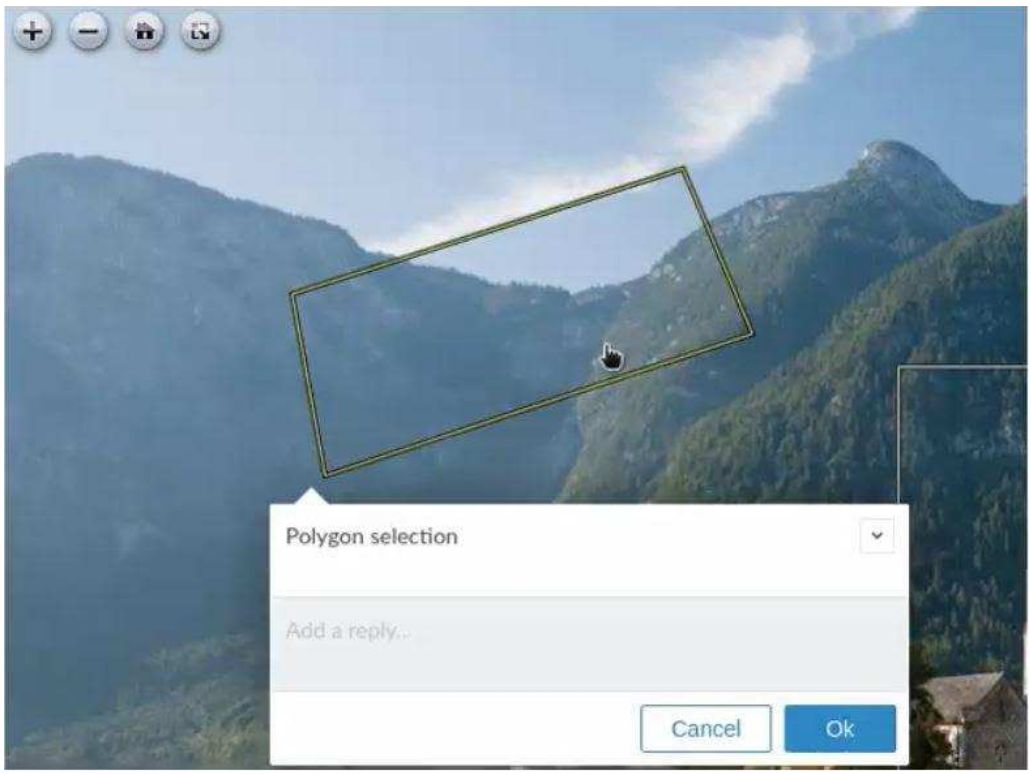
The image of the page shown is at

https://ia802701.us.archive.org/view_archive.php?archive=/11/items/northamericansyl04mich/northamericansyl04mich_jp2.zip&file=northamericansyl04mich_jp2%2Fnorthamericansyl04mich_0069.jp2&ext=jpg.

Recogito also does some automatic entity recognition.



Interface showing the use of a controlled vocabulary for tagging annotations as shown by Rainer Simon in Annotorious (twitter.com/aboutgeo/status/1281577849040797697)



Interface adding an annotation to a polygon selection of an image
as presented by Rainer Simon in Annotorious
(https://twitter.com/aboutgeo/status/1278767005051518978)

REQUIREMENTS FOR THE PROTOTYPE

From our initial assessment, 51 requirements were determined as the functionality to be considered in an annotation tool for the botanical community, some of them might be considered specific to the biological sciences and some may even be deemed mostly suitable for plant sciences only. However, we tried to keep these requirements as specific as possible, without losing their characteristics and relevance for annotating in a Biodiversity Sciences Digital Library in general.

Table 1. Prioritized Requirements.

Priority (Must/Should/Could)	#	Requirement
Must	1.0	Easy-to-use responsive interface, simple and flexible. Make it easy to include annotations: 2-3 clicks process, have a dropdown list of controlled vocabularies, allow tagging with an URL.
Must	3.1	Annotations must be visible for non-users
Must	6.0	Ability to highlight a target (text or image) by color-coding it, drawing a box around it.
Must	10.1	There will be at least 3 levels of sharing annotations: privately, with a group (members must be identified), and publicly (everyone). (See table below) There may be a need for more levels (for example: "only shared to registered users"). In order to promote sharing open annotations, they should be public by default, but the system will allow the user to configure its account settings to make all new annotations private, public or shared with a group by default. There should also be an option to indicate that you may want the system to ask you each time you annotate whether annotations are shared publicly, privately or with a given group.
Must	11.0	Modification of annotation target & body must be allowed
Must	13.0	When creating annotations, the target could be a text chunk, an image or another annotation. Allow adding a tag in a specific place (region) within an image.
Must	15.0	All annotations are visible by default but can be filtered (e.g. by author, date, etc)
Must	16.0	Annotations must be stored centrally but could also be cached locally.
Must	19.0	The system must be multi-platform.
Must	20.0	The system must allow to associate a licence with the annotation for any non-private use.
Must	21.0	Content creators must alway be logged into the system, never anonymous
Must	22.0	Only creators can modify (not delete) their own annotations, with the exception of administrators who can modify anyone else's annotations. If we allow edition, then we will need to link annotations to the different versions of the target (e.g. Google docs "resolve" functionality hides the old comment but its still there, not deleted).
Must	25.0	The system must be able to handle controlled vocabularies/checklists (thesauri; taxonomies like IPNI for all plant names, The Plant List, WORMS, Catalog of Life, and ITIS; gazetteers, etc.) and allow the creation of list of values, lists of people (authors like IPNI for all plant author names like VIAF, collectors, illustrators, VIAF, etc.), traits like morphological terms (Stearn's "Botanical Latin") and Marine Species Traits, habitats from marineregions.org, WWF Ecoregions and habitat ontologies, "Taxonomic Literature" (Stafleu and Cowan) for author names and journal title abbreviations, ontologies (OBO Foundry, Plant Phenology Ontology, FLOPO, PO, Gene Ontology) and systems like Atlas Living Australia, EOL, Index Herbarim and IPNI. This must be achieved by "registering" the controlled vocabulary (downloading locally or self building vocabularies) and make it available through the system. This should then allow an user to choose values from those lists, browsing or searching their labels (for example: habitats like mangrove, tropical montane rainforest, paramo), equivalent names (synonymy) and taking into account their hierarchy relations through time (species taxonomy, localities, etc.).
Must	26.0	The system would allow the user to define topics (for example: using a hashtag sign #), create a reference to an entity, associate terms to an annotation, etc. This could be done using annotations of annotations (like GoogleDocs uses the comment "Resolved" and dissappears the whole conversation if the last comment is of type "Resolved" but reappears it (and "unresolves" it) if a new annotation is added to the thread afterwards). Linking by adding URLs, replying or highlighting are different ways in the interface to input a certain type of annotation.
Must	27.0	Implement search functionality by keyword or type (comments/descriptions/customized tags/categories). Any references to entities within an annotation should be indexed and made searchable (for example: hashtag or @)
Must	33.0	The system must allow an user to filter the annotations by showing only those that came out in the current search result. (see # 15)
Must	37.0	Different types of annotations should be allowed. For example: specimen reference, taxonomic name, habitat types, corrected text, geographic locations, authors (artist, collector,dates, determined by), notes, reviews, links (URL, URI, DOI, barcode), customized categorization, personalized vocabularies or (hash)tags ("#Interesting", "#evolution", "#new_method", "#lacksDocumentation", "#lacksanalysis"), bibliography (citation), ratings are just some of the different types that the system could support.
Should	5.0	System includes context of images when showing the result annotations of a search within a digital library (See 34.0 for the case within the repository below)
Should	9.0	Ability to print target with annotations (layout TBD, but should include PDF (text) and comments).
Should	17.0	Annotations need to be discoverable outside of the place where they were added (i.e. separate from the website or target)
Could	31.0	The system could contribute to make the user aware of other existing annotations that might be related. For example, by highlighting the number of annotations of the same category that the user is choosing for his annotation, allow to search free text or vocabulary terms, traverse annotations with context in a separate search results page and create a reference to existing annotations in the system (in the page/book/Digital Library/Repository) or follow Kindle's example of marking for the Reader the parts of a book that are more highlighted by the Community. See #34.0 and #5.0
46	50.0	The system must be IIIF-compliant, being able to support images held in IIIF- compliant repositories (ie. the repository used (RERUM, Botanicus) should be IIIF-compliant)

For the prototype 18 (Must) requirements and a few others (3 Should, 1 Could) were chosen as the Minimum requirements that exemplify or support basic tasks that we believe would help users adopt annotations as part of their processes and workflows.

From the analysis of the details of these 22 chosen requirements and how they could be implemented in an annotation tool we selected a subset of 33 characteristics to consider when developing the prototype:

User Interface:

The annotation tools should have an easy-to-use responsive, simple and flexible interface to make it easy to include annotations with a few clicks, choosing values from dropdown lists of controlled vocabularies and allow tagging with an URL.

1. **Graphic Point and Click Interface**: The tool developed should have an easy-to-use responsive interface, simple and flexible and make it easy to include annotations, preferably with a 2-3 clicks process.

2. **Rich Text**: The annotation interface should allow the user to format text, display images included and recognized URLs entered in the text.

3. **Highlight a text**: The system must have the ability to highlight a target text by color-coding it, drawing a box around it.

4. **Highlight an image**: The system must provide the ability to highlight a target image by color-coding it, drawing a box around it.

Access Configuration

The topic of how to implement and configure a coherent set of access restrictions to the annotations created was thoroughly examined. There must be at least 3 levels to share annotations properly: privately, with a group (members must be identified), and publicly (everyone).

Configuration of Annotations (✔=Yes ✘=No)	Can View	Can Comment (reply/assess)	Can Edit (change)(1)
Private (me)	✔Default(2)	✔(3)	✔(3)
Group(4) (specific people)	✔	✔	✘(5)
Anyone(6) (registered)	✔	✔	✘
Public (everyone)	✔	✘	✘

(1)Should support versioning and hiding instead of deleting an annotation.
(2)Can view by default, can't change.
(3)To support workflows
(4)Should indicate specific people (by referring to their @IDs or, preferably, through a listbox.
(5)Couldn't find an use case that requires this functionality where "Comment" wouldn't do it.
(6)Any user registered in the system (ie. has an ID).

There may be a need for more levels (for example: "only shared to registered users"). In order to promote sharing open annotations, they should be public by default, but the system will allow the user to configure its account settings to make all new annotations private, public or shared with a group by

default. There should also be an option to indicate that you may want the system to ask you each time you annotate whether annotations are shared publicly, privately or with a given group.

5. **Login**: The system must allow the inclusion of annotations to registered users and choose who can see the annotations made, particularly to be made visible to the public (i.e. for non-users without a login into the system).

6. **Groups**: The system must allow the user to share annotations (only) with a group of users.

7. **Choose sharing to a group**: Can share annotations privately to a defined group

8. **Choose sharing to everybody/public**: Can share annotations to users even if they are not logged in

9. **Choose sharing to only logged in users**: Can share annotations only with registered users (with a login)

10. **Modify annotation text**: Modification of the body is allowed.

11. **No anonymous annotations**: Content creators must alway be logged into the system, the system should never allow to create anonymous annotations, therefore, it must allow users to create an annotation only when they are logged in.

12. **Only modify your own annotations**: Other users should not be able to modify annotations created by others. Only creators can modify (not delete) their own annotations, with the exception of administrators who can modify anyone else's annotations. If edition is allowed, then it will be needed to associate the annotations with the different versions of the target (e.g. Google docs "resolve" functionality hides the old comment but it's still there, it's not deleted, only hidden from the standard view).

What to annotate

Some annotation tools are text-oriented and assume there is a place in the text, displayed or not, where to anchor the annotation, other image-oriented systems define coordinates for a bounding box (some even bounding polygon) and manage this region as the target of the annotation.

13. **Annotate Text**: When creating annotations, the target could be a text chunk, recognized and kept as part of the annotation stored.

14. **Annotate Image**: When creating annotations, the target could be an image. Allow adding a tag in a specific place (region) within an image.

15. **"Reply" Annotation**: When creating annotations, the target could be another annotation. Allow to annotate another annotation, usually by *replying* to it.

Filtering the Annotations

All annotations should be visible by default but, In order to support workflows and managing accumulative cooperating annotations throughout time, the system must allow to filter the annotations displayed using different criteria.

16. **Filter by owner/author**: Annotations could be filtered by author or group.

17. **Filter by date created/modified**: Annotations could be filtered by the date of creation or last modification.

18. **Filter by other metadata field (optional)**: Annotations can be filtered by some of the other metadata fields associated with an annotation, like the annotation type.

19. **Filter Annotations**: The system must allow an user to filter the annotations by showing only those that came out in the current search result

Annotation Storage:

20. **Store in RERUM**: To avoid overwriting information and coordinating cooperative collaborations, annotations should always be stored centrally.

21. **Store locally**: Annotations must be stored centrally but could also be cached locally. But having stand alone local annotations without a continuous updating to the global repository could affect integration and demand a complex system of offline synchronization that could require manual intervention.

22. **Use Vocabulary**: The system must be able to handle controlled vocabularies/checklists (thesauri; taxonomies like IPNI for all plant names, The Plant List, WORMS, Catalog of Life, and ITIS; gazetteers, etc.) and allow the creation of list of values, lists of people (authors like IPNI for all plant author names like VIAF, collectors, illustrators, VIAF, etc.), traits like morphological terms (Stearn's "Botanical Latin") and Marine Species Traits, habitats from marineregions.org, WWF Ecoregions and habitat ontologies; "Taxonomic Literature" (Stafleu and Cowan) for author names and journal title abbreviations, ontologies (OBO Foundry, Plant Phenology Ontology, FLOPO, PO, Gene Ontology) and systems like Atlas Living Australia, EOL, Index Herbariorum and IPNI. This must be achieved by “registering” the controlled vocabulary (downloading locally or self building vocabularies) and make it available through the system. This should then allow an user to choose values from those lists, browsing or searching their labels (for example: habitats like mangrove, tropical montane rainforest, paramo), equivalent names (synonymy) and taking into account their hierarchy relations through time (species taxonomy, localities, etc.). This functionality should be achieved by “registering” the controlled vocabulary and make it available through the system.

Search Annotation:

23. **Search Vocabulary Terms**: Search terms from a vocabulary in the (public/private) annotation bodies (and optionally, in the targets too)

24. **Search a Text**: The system should support the search of free text in the (public/private) annotation bodies (and optionally in the targets too). Also the user should be able to search for a keyword. It should only show the annotation that the user is allowed to see according to their access configuration.

25. **Textual Context (optional)**: The system includes context of text when showing the result annotations of a search within a digital library.

26. **Image Context (optional)**: System should include the context of images when showing the result annotations of a search within a digital library.

Print Annotation:

27. **Print without Context**: Ability to print the body of the annotations

28. **Print with Context (optional)**: Ability to print the body of the annotations with the target

Other requirements:

29. **Assign Unique ID**: Each existing annotation (be it a page, book, a Digital Library, or a Repository target) must have a unique reference (URI) that allows access to it.

30. **Multiplatform**: The easiest way to be multi-platform, is to have the system run in a Web browser so the same version of the program can work perfectly in different environments.

31. **License**: The system must allow associating a license with the annotation for any non-private use. This could be achieved by having a list of pre-defined licenses (CC-BY, CC0, etc) to choose from.

32. **Choose Annotation type from list**: The system must allow for different types of annotations. For example: specimen reference, taxonomic name, habitat types, corrected text, geographic locations, authors (artist, collector,dates, determined by), notes, reviews, links (URL, URI, DOI, barcode), customized categorization, personalized vocabularies or (hash)tags (“#Interesting”, “#evolution”, “#new_method”, “#lacksDocumentation”, “#lacksanalysis”), bibliography (citation), ratings are just some of the different types that the system could support.

33. **Follow IIIF standards**: The system must be IIIF-compliant, being able to support images held in IIIF- compliant repositories (ie. the repository used, be it RERUM or Botanicus, should be IIIF-compliant

TO INCLUDE IN PROTOTYPE

	Done		Done	
1	✓	Graphic Point and Click Interface	8	✓	Store in RERUM
	1	Rich Text?	9	✓	Store locally
	2	Login (any string: 'WU@.mbg.org')	10	✓	Prototype runs in a browser (Web)
	3	Groups? (hard-coded in prototype)		10	License (predetermined list)
2	✓	Highlight (text)		11	No anonymus annotations
3	✓	Highlight (image)		12	Only modify your own annotations
	4	Choose sharing (to a group)		13	Use Vocabulary (hard-coded lists)
	5	Choose sharing (everybody/public)		14	Search Vocabulary Terms
	6	Choose sharing (only logged in)	11	✓	Assign Unique ID
4	✓	Modify annotation text		15	Search a Text
5	✓	Annotate Text		16	Filter Annotations
6	✓	Annotate Image		17	Choose Annotation type from list
7	✓	"Reply" Annotation		18	Textual Context (optional)
	7	Filter by owner/author		19	Image Context (optional)
	8	Filter by date created/modified		20	Print without Context
	9	Filter by other metadata field (opt)		21	Print with Context (optional)
			12	✓	Follow IIIF standards

Each characteristic was assessed and, if it was considered to be already sufficiently addressed by an existing tool or the initial prototype developed it, then it was marked as done and not prioritized in our development. The rest of the issues were addressed in the new proof-of-concept tool by analyzing how to sketch its implementation in the available infrastructure. An agile approach was adopted for creating and deploying the software developed.

SECOND PROTOTYPE

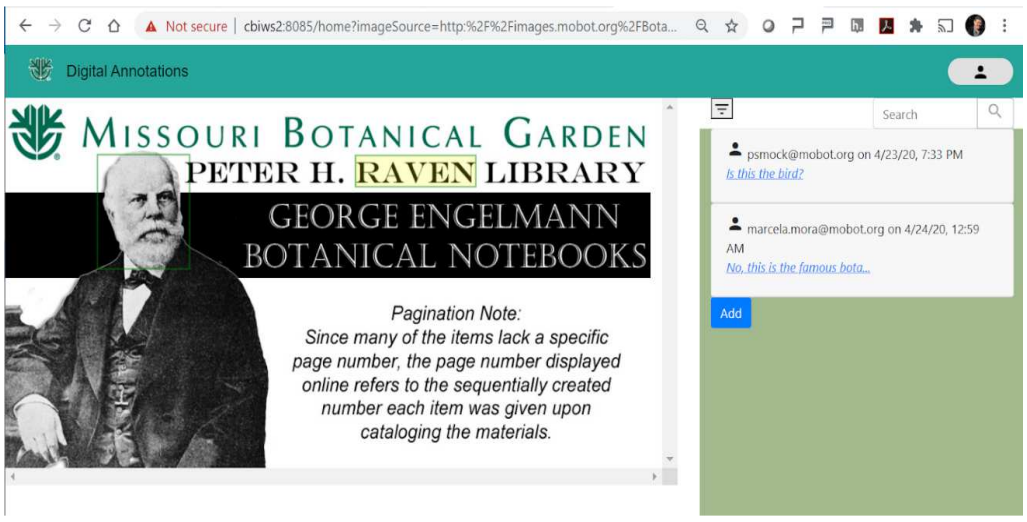
A proof-of-concept prototype was developed to address the basic characteristics that an annotation system must provide handling images like the ones in a digital library platform (like Botanicus). The code is stored in a public GitHub repository: MBG-CBI/C2C: Consumers as Creators (IMLS LG-87-18-0057-18).

The User Interface module was upgraded to the latest package of Angular 9 and the API part was developed on .Net Core 2.1 which means that the API is open source and can run on Linux, although by the time of this report it was not supported anymore and would require an upgrade to .Net Core 3.1. Given that this was only meant to be a prototype, it was decided not to upgrade because the step process would probably take too much time and the benefits of analyzing the way to implement this functionality was already achieved.

One of the technical lessons learned while developing our prototype is that the use of straight Angular material and Flex, instead of bootstrap, is definitely a better choice for a responsive design.

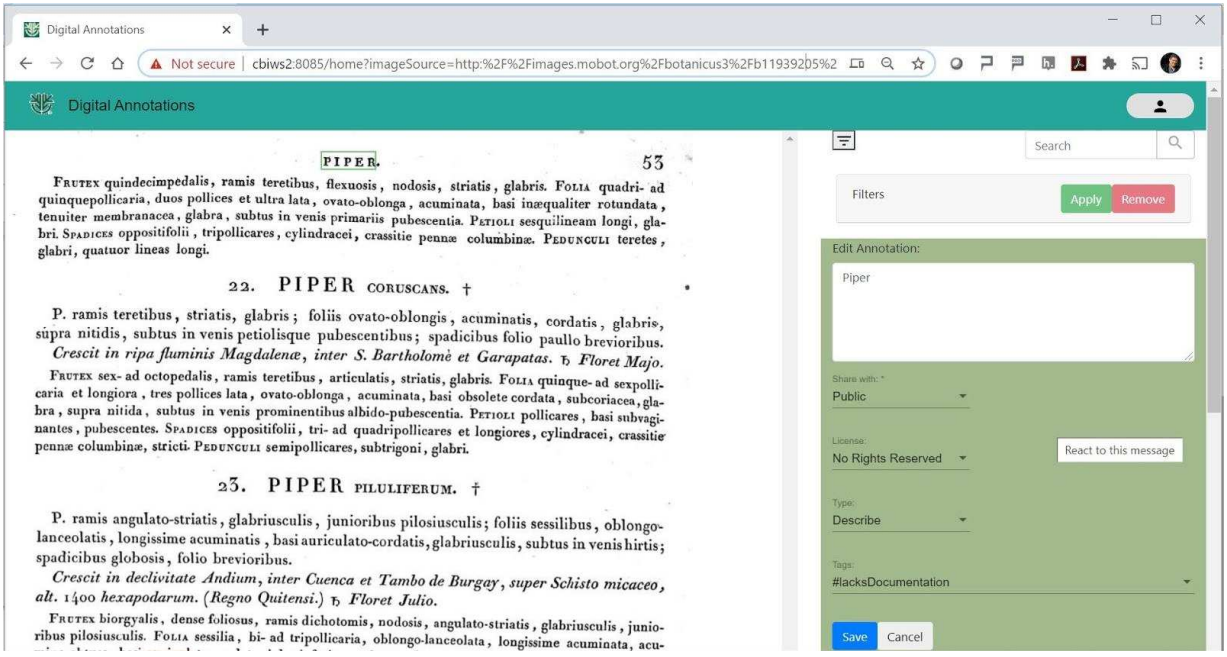
As the prototype was being developed, the updates were published to the development site. The host part was always the same (*http://cbiws2:8085/home?imageSource=*) and the source was then the URL to an image accessible on the web. You can see below an example of an image from Botanicus Digital Library stored in the image server of the Missouri Botanical Garden.

(http://images.mobot.org/Botanicus17/b14055144/31753003957617/31753003957617_0000.jpg#xywh=549.00000000,120.00000000,293.00000000,46.00000000)



Prototype highlighting in yellow the target in the image and showing the body of the annotation followed by a Reply from a different user

One other consideration that we learned was that, although our annotations repository was IIIF-compliant, our image IIIF server at Missouri Botanical Garden was not fully IIIF compliant. Therefore, the manifest we created had to be developed manually through a separate try and error process, which may have led to invalid values in some of the fields and made the testing harder.



Input interface of the Prototype showing a login button on the right top corner, the type and sharing of the annotation, the license and tags assigned, and the filters to apply to the annotations shown.

One of these issues, for example, was the image size. Images should have been scaled to fit a certain size on screen, otherwise those higher-resolution images that were too big would just take over the screen. A fully IIIF compliant server would be able to handle this easier. Given a proper understanding of the IIIF Client or Viewer specification to make correct calls to the server or using a viewer already embedded into a IIIF-compliant application would automatically handle any considerations with scaling and size.

In addition, because the goal was to implement something that could annotate any image, requiring that the image served be IIIF-compliant was not implemented as part of this prototype. We learned this to be more often the cause of invalid values in the manifest, as we did not write up requirements for a manifest for images that were not IIIF compliant.

RERUM Repository

As stated before, RERUM was used in the proof-of-concept prototype tool as the annotations repository to test where different types of annotations that the botanical users needed could be stored.


```
// 20200728144138
// http://devstore.rerum.io/v1/id/5cf96fc1e4b07d216aab776d

{
  "@context": "http://cbiws2:8086/api/WebManifests/1/context.json",
  "@type": "sc:Manifest",
  "viewingDirection": "right-to-left",
  "sequences": [
    {
      "@type": "sc:Sequence",
      "canvases": [
        {
          "@id": "http://cbiws2:8086/api/WebCanvases/1",
          "@type": "sc:Canvas",
          "width": 0,
          "height": 0,
          "images": [
            {
              "@context": "http://www.w3.org/ns/anno.jsonld",
              "@id": "http://cbiws2:8086/api/WebAnnotations/1",
              "@type": "oa:Annotation",
              "on": "http://cbiws2:8086/api/Canvases/1",
              "target": {
                "id":
"http://images.mobot.org/Botanicus17/b14055144/31753003957617/31753003957617_0000.jpg#xywh=549.00
000000,120.00000000,293.00000000,46.00000000",
                "type": "Image",
                "format": "image/jpeg"
              },
              "body": {
                "type": "TextualBody",
                "value": "Is this Engelmann?",
                "format": "text/plain"
              }
            },
            {
              "@context": "http://www.w3.org/ns/anno.jsonld",
              "@id": "http://cbiws2:8086/api/WebAnnotations/2",
              "@type": "oa:Annotation",
```


A sequence of annotations and replies on a Botanicus image as stored in RERUM by the Prototype

As a recommendation from our experience, it could be said that time would be very well invested in fully understanding the way that the IIIF specification is implemented, particularly in the chosen repository (RERUM in our case) before any development is advanced. This familiarity would help to get better requirements as to what would be stored and where it might be helpful.

The prototype was tested with different types of annotations to determine key aspects of a tool development or adaptation of an existing one. All these recommendations on efficient ways to address the requirements learned from this experience informed the Outcomes Assessment.


```
// 20200726103040
// http://devstore.rerum.io/v1/id/5cf98463e4b07d216aab7774

{
  "@context": "http://cbiws2:8086/api/WebManifests/6/context.json",
  "@type": "sc:Manifest",
  "viewingDirection": "right-to-left",
  "sequences": [
    {
      "@type": "sc:Sequence",
      "canvases": [
        {
          "@id": "http://cbiws2:8086/api/WebCanvases/6",
          "@type": "sc:Canvas",
          "width": 0,
          "height": 0,
          "images": [
            {
              "@context": "http://www.w3.org/ns/anno.jsonld",
              "@id": "http://cbiws2:8086/api/WebAnnotations/30",
              "@type": "oa:Annotation",
              "on": "http://cbiws2:8086/api/Canvases/6",
              "target": {
                "id": http://images.mobot.org/botanicus3/b11939205/31753000540382/31753000540382_0116.jpg
                          #xywh=358.00000000,187.00000000,102.00000000,29.00000000",
                "type": "Image",
                "format": "image/jpeg"
              },
              "body": {
                "type": "TextualBody",
                "value": "PIPER",
                "format": "text/plain"
              }
            },
            {
              "@context": "http://www.w3.org/ns/anno.jsonld",
              "@id": "http://cbiws2:8086/api/WebAnnotations/31",
              "@type": "oa:Annotation",
              "on": "http://cbiws2:8086/api/Canvases/6",
              "target": {
                "id": "http://images.mobot.org/botanicus3/b11939205/31753000540382/31753000540382_0116.jpg
                          #xywh=341,452,251,18",
                "type": "Image",
                "format": "image/jpeg"
              },
              "body": {
                "type": "TextualBody",
                "value": "Piper piluliferum",
                "format": "text/plain"
              }
            }
          ]
        }
      ]
    }
  ],
  "__rerum": {
    "alpha": "true",
    "APIversion": "0.8.0",
    "createdAt": 1559856227715,
    "isOverwritten": "",
    "isReleased": "",
    "history": {
      "next": [

      ],
      "previous": "http://devstore.rerum.io/v1/id/5cf98441e4b07d216aab7773",
      "prime": "http://devstore.rerum.io/v1/id/5cf98441e4b07d216aab7773"
    },
    "releases": {
      "next": [

      ],
      "previous": "",
      "replaces": ""
    },
    "generatedBy": "http://devstore.rerum.io/v1/id/5bdb4c96e4b0842a2b4e4446"
  },
  "@id": "http://devstore.rerum.io/v1/id/5cf98463e4b07d216aab7774"
}
```


JSON document obtained from RERUM corresponding to an oa:Annotation over an image from Botanicus made with the Prototype