

# Schedule 2: Manuscript Viewer Quote

David Livingstone Archive

## Version History

Version	Description	Author	Date
1.0	Initial Draft	Nigel Banks	January 25th, 2016
1.1	Updated estimates	Nigel Banks	Feb 9th, 2016

[Version History](#)

[Overview](#)

[Embedded Display Issues](#)

[Sequence Display Issues](#)

[Auto hide and Show Toolbar and Bottom Popup](#)

[Sidebar Menu & Livingstone Icon & Share Icons](#)

[Zoom & Zoom Slider](#)

[Rotate](#)

[Transcription & Item Details Toggle](#)

[Display of Item Details](#)

[Transcription Display](#)

[Replace Saxon-C](#)

[Create Transcription Objects](#)

[Rendering Transcriptions & Auto Scroll](#)

[Page Select & Search Bar & Close Viewer](#)

[Directional Icons](#)

[Bottom Popup \(Thumbnail Slider\)](#)

[Not Included in Estimates](#)

[Cost Summary](#)

## Overview

The following quote is to take the existing work that UCLA has done on the [Manuscript Viewer](#), and make it functionally and visually equivalent to the the level 4 wireframes (see attached documents). Unfortunately much of what UCLA provided doesn't match the provided wireframes and will need to essentially start from scratch.

The Manuscript Viewer will be built to be run as an overlay, such that it will not require a page refresh to load (AJAX based). It will be displayed when selecting records from one of the browse by pages; Standard Catalog, Addressee, Timeline, Geographical Location, Repository and from the site wide search.

## Embedded Display Issues

The Manuscript Viewer will be loaded dynamically in several locations, this has some unintended side effects, in that no URL is associated its display. So things like the share functionality (Facebook, Twitter, etc) would not function correctly, as they have no URL unique to the content that's being displayed to share as a link. So essentially we need to include information in the URL without causing a page refresh. There are two ways to do this:

1. Use the fragment identifier of the URL to store display information
2. Use HTML5 History API

Each has it's downsides, but HTML5 History API is definitely the first choice unless supporting older browsers like IE9 and lower ([HTML5 History API Browser support](#)) is a hard requirement. For browsers like IE9 and earlier the site will still work, but a page refresh will occur when opening the Manuscript viewer and flipping through pages.

**Estimate: 14 hrs**

## Sequence Display Issues

As it currently stands, Islandora implementation of the OpenSeadragon viewer doesn't support the display of a sequence of images. Currently only works arounds with page refreshes or tricks with the Fragment Identifier load up a single image at a time. We'll have to change this such that Islandora can support loading all the images in a Manuscript at once.

**Estimate: 14 hrs**

## Auto hide and Show Toolbar and Bottom Popup

Build the toolbar such that it will auto hide itself when not being currently viewed, and it will reveal itself when the user mouses over the top of the page.

**Estimate: 3 hrs**

## Sidebar Menu & Livingstone Icon & Share Icons

Integrate the sidebar menu and have the Livingstone Icon redirect to the home page. Also integrate the Drupal [Share This](#) module into the Manuscript viewer. This will allow users to share the page they are currently viewing via Twitter, Facebook or Email.

**Estimate: 2-5 hrs**

## Zoom & Zoom Slider

Zoom functionality is provided by OpenSeadragon, but unfortunately it doesn't come with a slider that we could simply reskin. Time is to build the slider widget and the other icons.

**Estimate: 5 hrs**

## Rotate

Rotation is not supported in the OpenSeadragon version in which Islandora is known to work with, though it is supported in later versions. We can update Islandora to work with a newer version of OpenSeadragon. Unfortunately the update is a significant rewrite of the Islandora OpenSeadragon module. As OpenSeadragon changed their API significantly (hence why the Islandora community hasn't bothered upgrading it in last 3 years).

**Estimate: 10 hrs**

## Transcription & Item Details Toggle

The logic required to render one or both the Transcription and the Item details overtop of the viewer. Such that:

- The text for whichever of these buttons is pushed first will appear on the right, while that pushed next will appear on the left.
- Should a user unclick one of these buttons the view would revert to image plus transcription or item details.

**Estimate: 4 hrs**

## Display of Item Details

Display of Manuscript Metadata and related information, will use the existing XSLT.

- Extraction and display of content from MODS Record.
- Pulling "Image Credits" from a separate database of institutional permissions text.
- Display of hard-coded "Publication" details.
- Extraction and display of "Digital Edition" details from the TEI transcription.
- Display of "Cite Item" details are from MODS / TEI from the other sections of the metadata plus the URL of the given item plus the present date.

**Estimate: 2 hrs**

## Transcription Display

Converts a subsection of the TEI document corresponding to the currently displayed page from TEI to HTML.

### Replace Saxon-C

The code is currently using Saxon-C which is experimental and not currently supported on CentOS / RHEL. So we need to replace it with Saxon servlet running in Tomcat, which we'll call via it's provided REST API.

**Estimate: 3-4 hrs**

### Create Transcription Objects

Create required objects to house the transforms, and relate them to the appropriate collection.

**Estimate: 1 hrs**

## Rendering Transcriptions & Auto Scroll

James Cummings will provide a CSS file and XSLT that will be used to generate a single HTML document from the objects TEI if present. We'll render out that HTML document in an iframe allowing the user to dynamically scroll through the entire document. As the user scrolls past pages we'll update the image being rendered to show alongside the TEI text as appropriate.

The attribute **@facs** on the page break element (<pb/>) should be used as the **@id** of the topmost tag that contains the generated html for that page.

So:

```
<pb n="0001" facs="liv_000490_0001">
```

Should be transformed into something like this (the tag is irrelevant just the id is important).

```
<div id="liv_000490_0001">  
  <!-- generated HTML content for page 0001 goes here -->  
</div>
```

Also a toggle will be present on the TEI display to switch between the "Reading" and "Editorial" display of the text.

**Estimate: 6 hrs**

## Page Select & Search Bar & Close Viewer

Implement the following:

- Auto updates to show the current page and allows the user to jump to a specific page.
- Performs sitewide search.
- Returns the user to the page they were currently viewing before opening the viewer.

**Estimate: 3 hrs**

## Directional Icons

These will be rendered as directional icons overlaid overtop of the viewer. It requires first that we implement [Sequence Display](#) support in the Islandora.

**Estimate: 2 hrs**

## Bottom Popup (Thumbnail Slider)

The icons along in the bottom popup represent the pages of the given item. The arrows at left and right of the icon strip allow the user to scroll through the image sequence. The item that the user is currently viewing appears as a slightly brighter icon. If a user mouses over a given icon, it becomes transparent and displays the underlying page number. It requires first that we implement [Sequence Display](#) support in the Islandora.

**Estimate: 4 hrs**

## Not Included in Estimates

The following features, and additions are not included in the estimates.

- Mobile Support (Not present in the wireframes)
- “Cite Item” use of CSL and providing support for rendering different styles of citations, only the hard-coded display of a single Citation style is included.
- Creation of TEI -> HTML transforms