# Table of Contents

## Front End

User has a table of contents with links to all of his Journals (and perhaps sub-entries in the Journals).

* Displays links to journals
  + Journal may have sub-links if they contain many pages.

# View Journal and Photos

## Front End

User can navigate to his own completed photo-journal.

* User sees the journal text side-by-side with the photo thumbnails corresponding to that time rage of the journal.
* Scrolling through the journal updates the set of displayed thumbnails in a grid of thumbnails.
* Clicking on a thumbnail enlarges the image for better viewing.
  + Right and left navigation arrows let the user advance between enlarged images in the timespan.
* Hovering on a thumbnail highlights the body of journal text corresponding to the thumbnail’s timestamp.
* Clicking on a paragraph of text scrolls the thumbnail grid to display the images from that time.

# Create Journal Pages

## Front End

User can create new pages of journal text.

* Each page can have a heading – usually this is something like “10/19/2009 Hong Kong”.
* The body of the page will be entered as paragraphs of text.
* The user does not need to worry about applying HTML formatting tags to his text - <p> tags will automatically be added to each paragraphs
* (Low priority) a full feature text editor interface is available for entering the journal text with custom formatting.

# (Re)assign Timestamps to the Journal Text

## Front End

User annotates his journal with timestamps so that the journal viewer knows which photos/thumbnails to show for which paragraphs of the journal.

* User selects a Journal Page to annotate with timestamps.
  + If the journal page already has assigned thumbnails, then those thumbnails appear embedded in the journal text.
* User selects a stream of images for use as thumbnails.
* User selects a start date and time interval from which he can load thumbnails.
  + Example: start=10/19/2009 and interval=1 day would show all thumbnails between 10/19/2009 00:00:00 and 10/20/2009 00:00:00
* A grid of images from the chosen date range is loaded.
* User drags and drops images onto paragraphs of journal text to associate a timestamp with that paragraph.
  + Dropping the image on the paragraph means that the image/timestamp should appear when that paragraph is visible.
  + The dropped thumbnail will appear as an image embedded in the journal text. (Click the image to remove it.)
  + The dropped thumbnail applies to the target paragraph and all later paragraphs until the next thumbnail is encountered.
  + Question – how do I deal with paragraphs for which there are no images? I think that the most recent images would always show! Could I have a Blank/Clear thumbnail which can be dropped to indicate a clear of time?
  + On thumbnail placement the total sequence of placed thumbnails needs to be evaluated to make sure that all thumbnails before the new one have earlier timestamps and all thumbnails after have later timestamps. If the sequence is improper (not in smallest-to-largest sorted order), then a warning should be shown to the user with the choice of either deleting the newly placed thumbnail/timestamp or deleting the set of old thumbnails/timestamps until the sequence is again correct.
    - Example: if the user places timestamp 7 to form this incorrect sequence of timestamps 2, 7, 4, 6, 8, 10, the website would force the user to either delete the newly placed 7 or the preexisting 4, 6. Either action would fix the sequence (put it back into sorted order).
* A Save button the changes to the page.
* (Low priority) Timestamps can be assigned to portions of a paragraph.

## Back End

* The dragging and dropping of images onto the journal text will be done with Javascript.
* When the changes are saved, all of the thumbnails dropped into the text will be replaced with HTML tags that invisibly encapsulate some paragraphs of the journal with that timestamp. (<span class="timestamp" id="1259928664"> where ‘id’ is the UNIX timestamp).

# Import Photo Streams from Online Services

## Front End

The photos shown on this website will be hosted on external site Google Photos. Users will need to tell the journal website how to find the stream of images.

* This page lists the photo stream services which are supported.
* User selects one of his journal pages as a target for the operation.
* User selects the service and then enters the appropriate URL information for that stream.
  + Helpful instructions on how to find the URL for that service should be available.
* >=1service and >=1 stream can be associated with a journal page.
* (Low priority) photos can be on Amazon Cloud, Flickr, SmugMug, or something other

## Back End

Photo streams from different services will need to be processed and stored in the master DB for quick access. In the DB we’ll store:

* UNIX timestamp for the photo.
* URL to the small thumbnail-sized image for the photo.
* URL to the large-sized image.
* (Low priority) Latitude and Longitude information.
* (Low priority) Alt Text

# (Low Priority) Manage Photo Streams

## Front End

The timestamps on the user’s photos may need adjusting.

* Give some way of adding or subtracting time from the timestamp in the photo stream.
* Allow the adjustments to individual photos and also batches of photos.

## Back End

* Don’t tinker with the UNIX timestamp from the photo stream. Instead, save the user’s applied offset as a second value in the DB, and just always sum the UNIX timestamp and offset when the functional timestamps is needed.

# User Accounts

## Back End

User accounts will store all of the information for managing 1 or more journal pages. This needs to be stored in a database of some type.

* Users will have 1 or more journal pages.
* Each journal page will have a Title and Timestamp Annotated Content Text.
* Each journal page will also be associated with >=1 thumbnail stream.
* (Low priority) Each user can have a Table Of Contents.