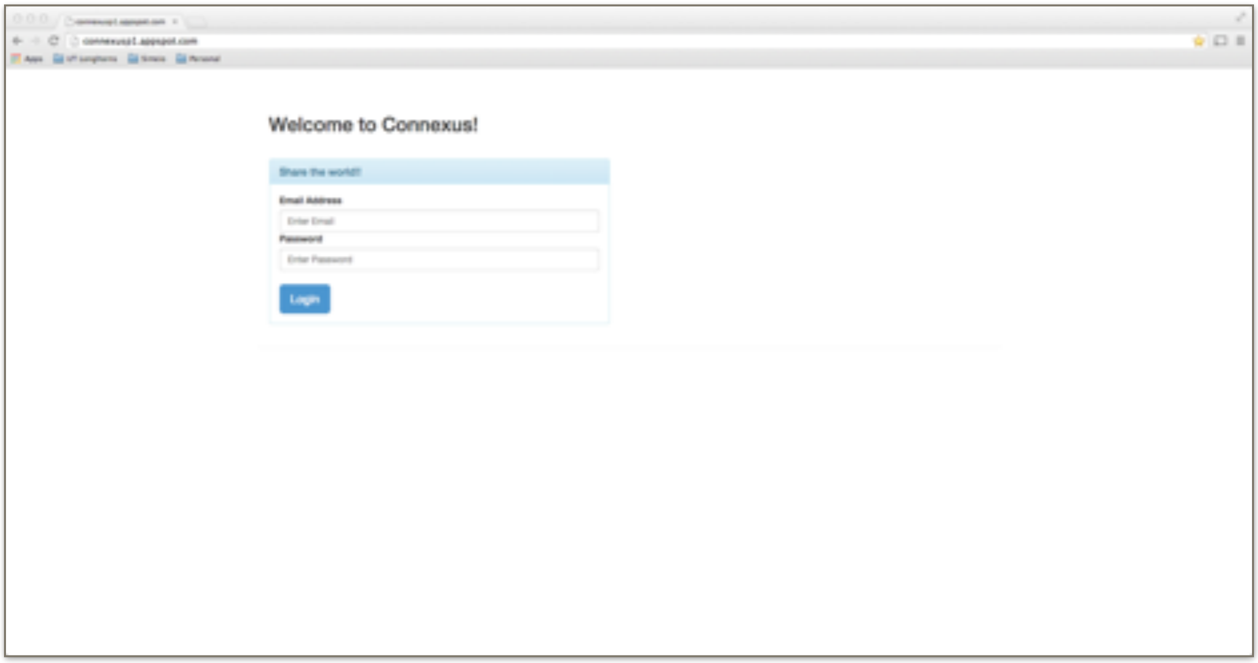
Screenshots of Connexus App (MiniProject Phase 1)

Connexus App URL: http://connexusp1.appspot.com/

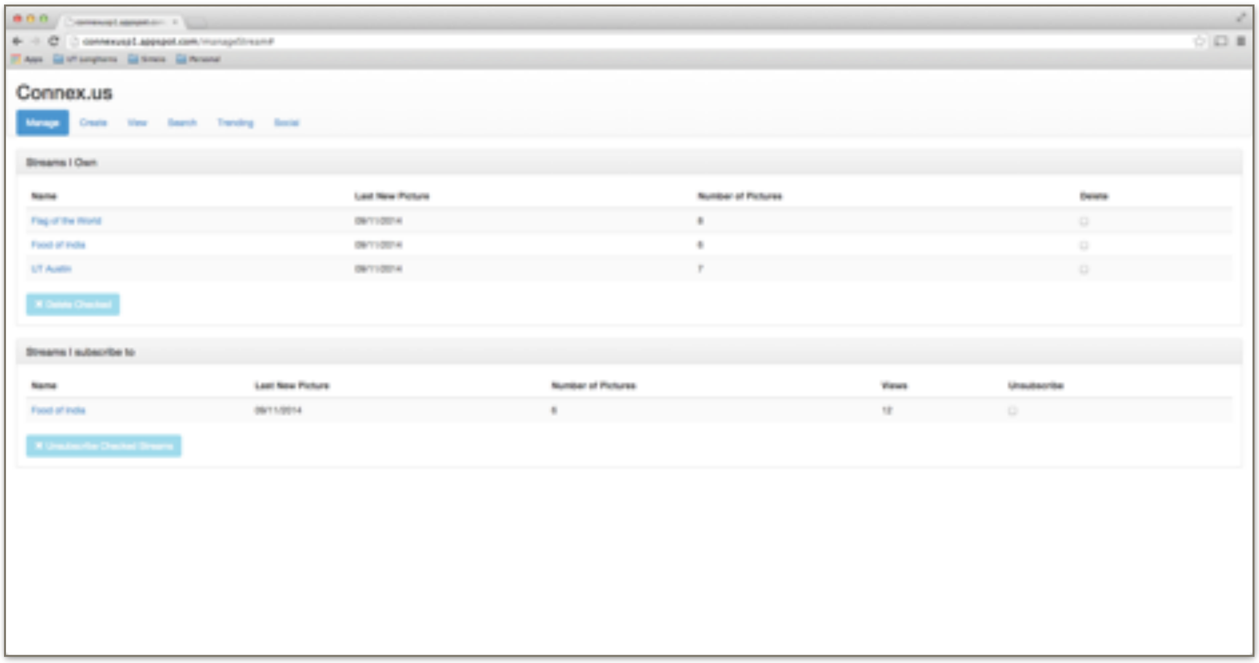
Login Page:

The Login Page shown here is meant only for Display Purposes. This would be the First screen shown to the user on accessing the application. Once the User clicks on the Login button (even without entering any values), they will be redirected to the Google Authentication for logging into the Application



Management Page:

User will redirected to this page, once they are successfully logged in. They can view all the stream they Own or Subscribe to



Streams I Own

Name	Last New Picture	Number of Pictures	Delete
Flag of the World	09/11/2014	8	☐
Food of India	09/11/2014	6	☐
UT Austin	09/11/2014	7	☐

✖ Delete Checked

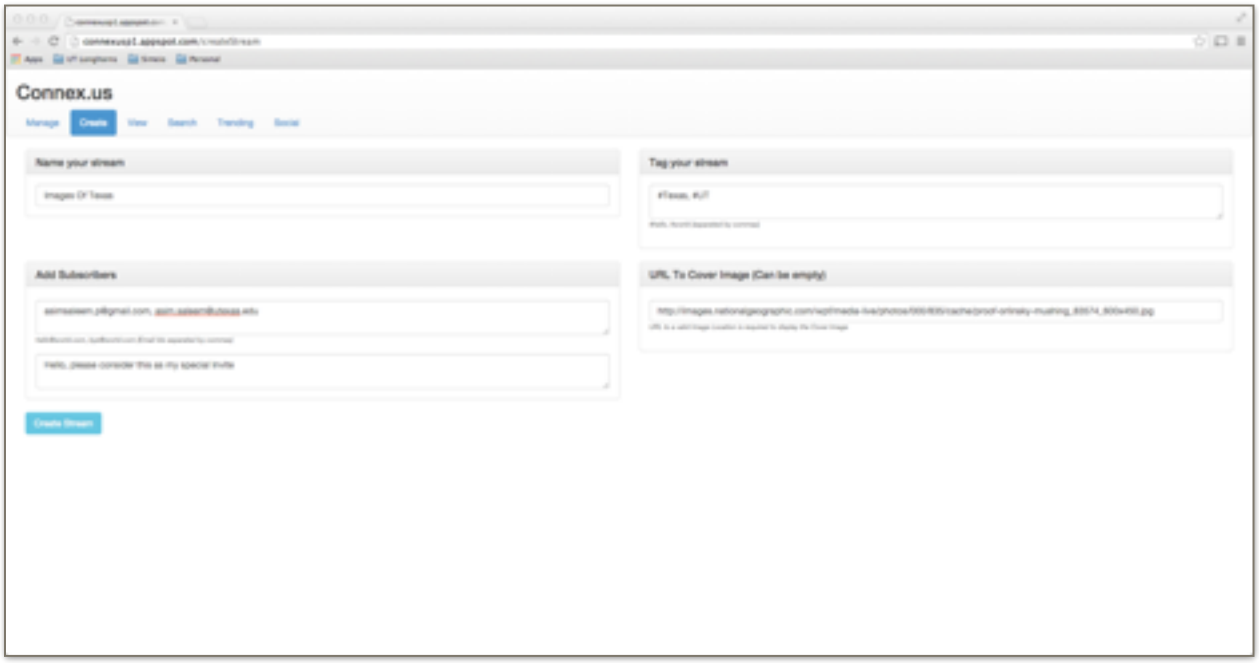
Streams I subscribe to

Name	Last New Picture	Number of Pictures	Views	Unsubscribe
Food of India	09/11/2014	6	12	☐

✖ Unsubscribe Checked Streams

Create Stream Page

User can click on the “**Create**” tab to open this screen and create a new stream and provide the requisite details that is needed



connexus1.appspot.com/createStream

Connex.us

Manage | Create | View | Search | Trending | Social

Name your stream

Images Of Texas

Tag your stream

#Texas, #UT

Add Subscribers

[illegible]@gmail.com, [illegible]@utexas.edu

Hello, please consider this as my special invite

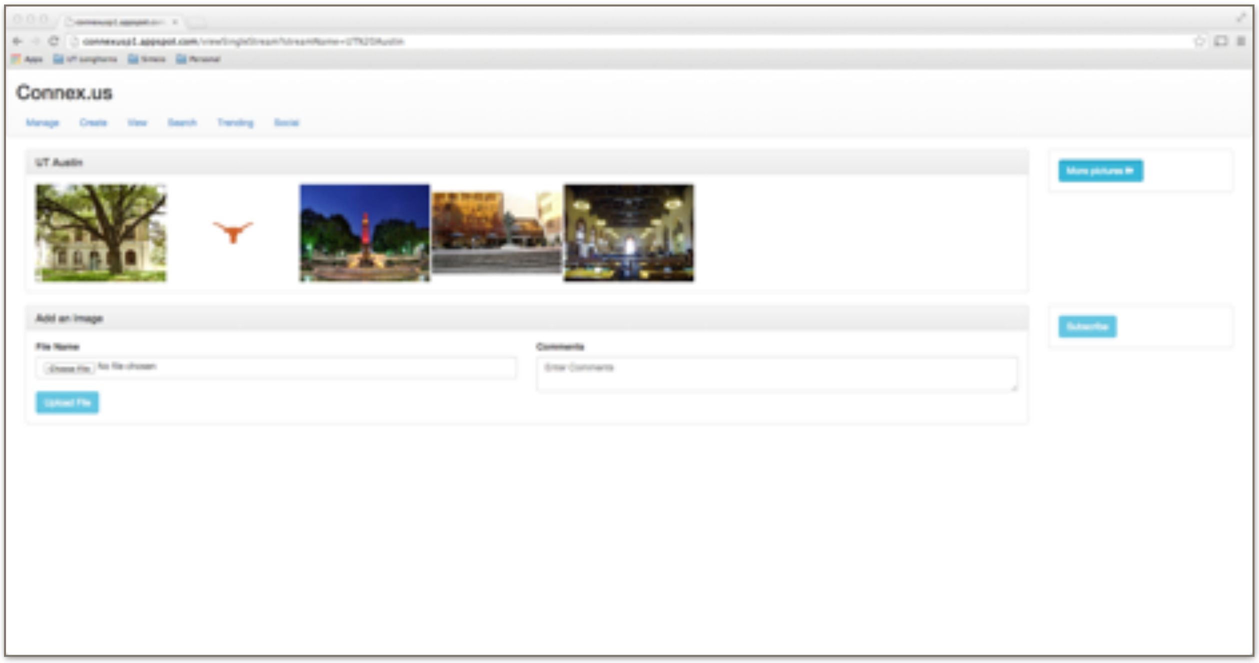
URL To Cover Image (Can be empty)

http://images.nationalgeographic.com/wpf/media-live/photos/000/835/cache/[illegible]_800x450.jpg

Create Stream

View A Single Stream Page

Users can access this Page by clicking on a stream name in the Manage Stream page or on the search result in the Search Stream page



connexus1.appspot.com/viewSingleStream?streamName=UT%20Austin

Connex.us

Manage | Create | View | Search | Trending | Social

UT Austin

More pictures

Add an Image

File Name: Choose File | No file chosen

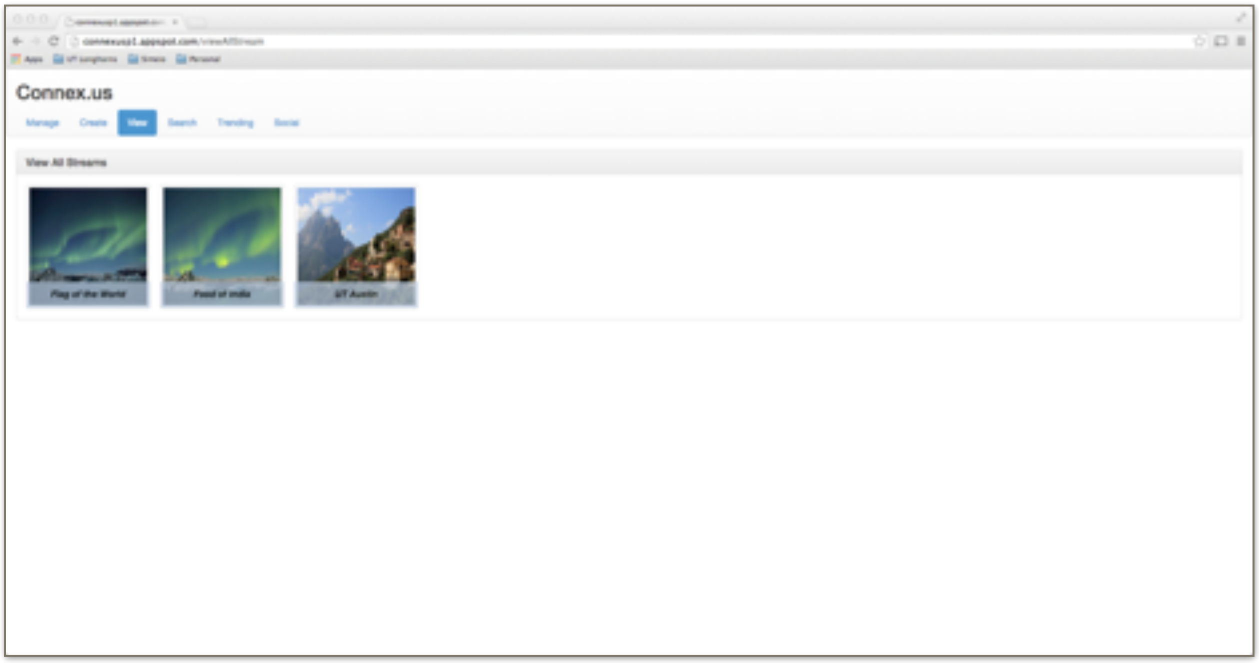
Comments: Enter Comments

Upload File

Subscribe

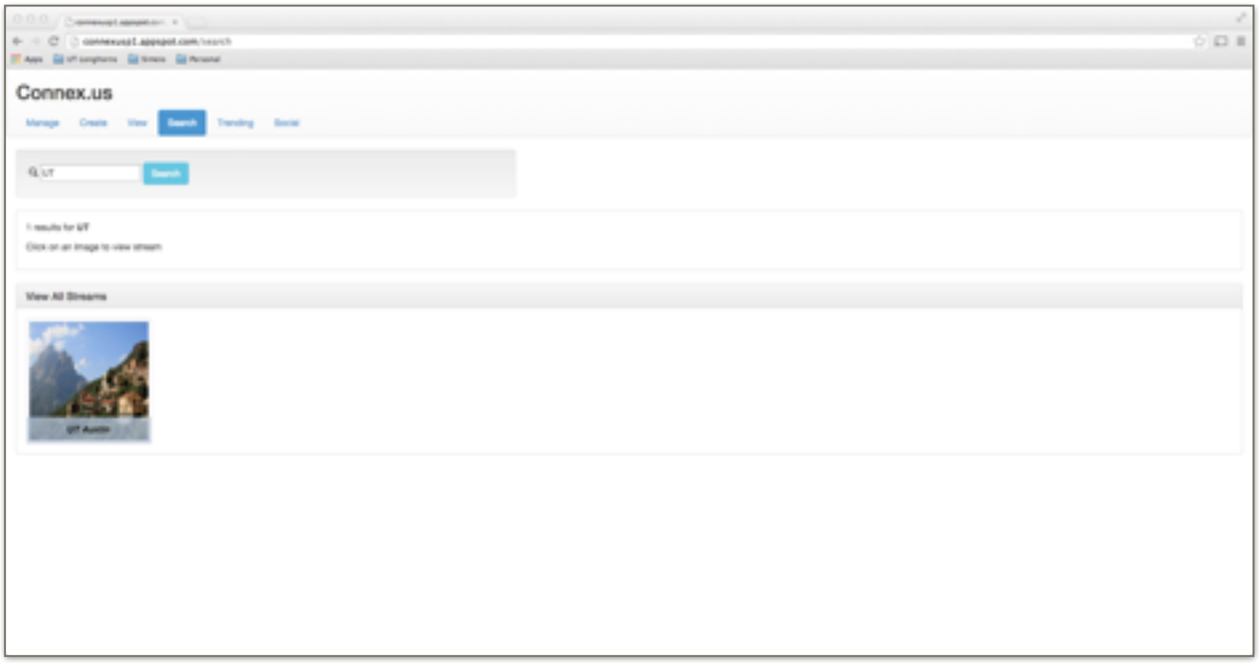
View All Streams

On clicking “View” users will be shown all the Streams that exist in the system. Please note that this is a read only screen and there is no requirement to allow Users to open individual streams from here



Search Streams Page

Users are allowed to search for different streams in this page, based on various parameters they provided during the stream creation process

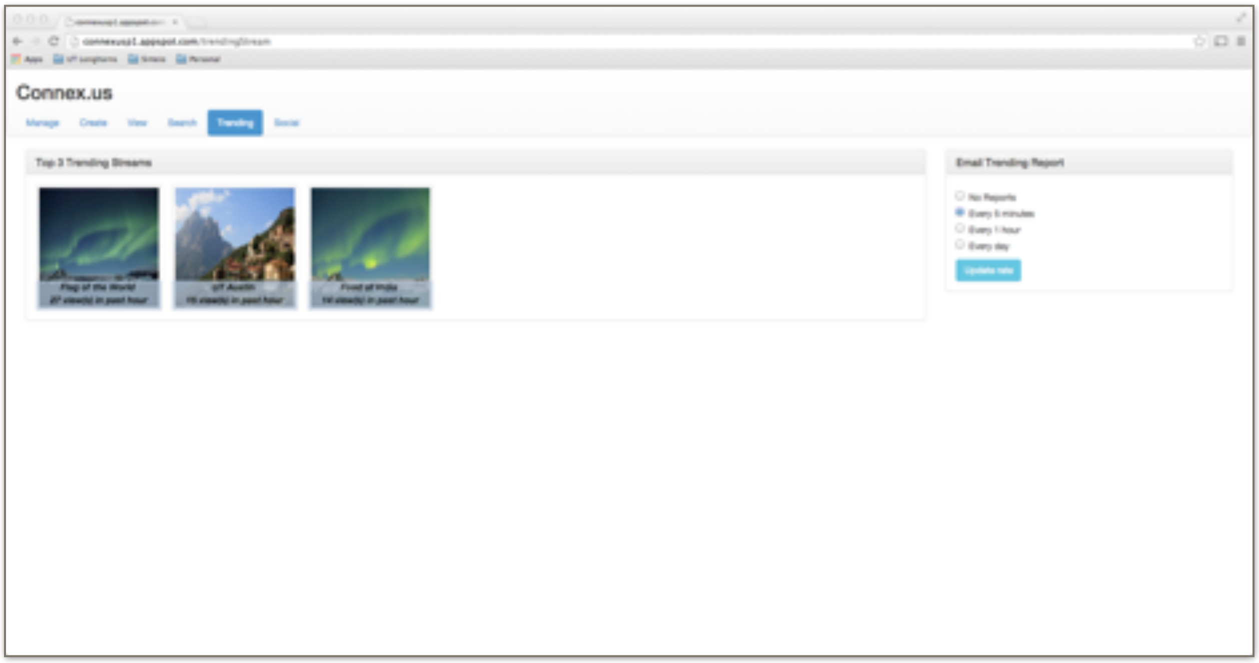


1 results for UT

Click on an image to view stream

Trending Streams Page

As per the requirement, the images that have been clicked in the past 1 hour will be displayed here. If there had been no clicks in more than hour, this screen would remain empty, until a job runs again and a click is identified. This screen also contains the notification options for email intimation of the trending streams



Top 3 Trending Streams

Flag of the World
27 view(s) in past hour

UT Austin
15 view(s) in past hour

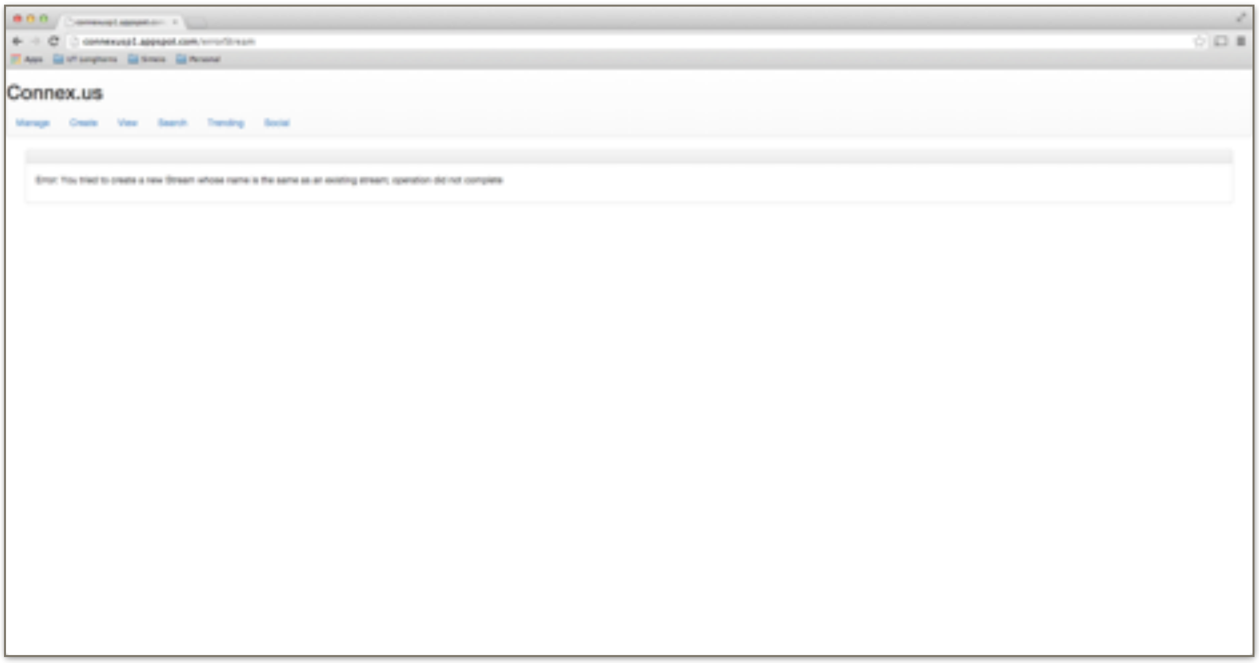
Food of India
14 view(s) in past hour

Email Trending Report

- No Reports
- [✓ Every 5 minutes]
- Every 1 hour
- Every day

Error Page

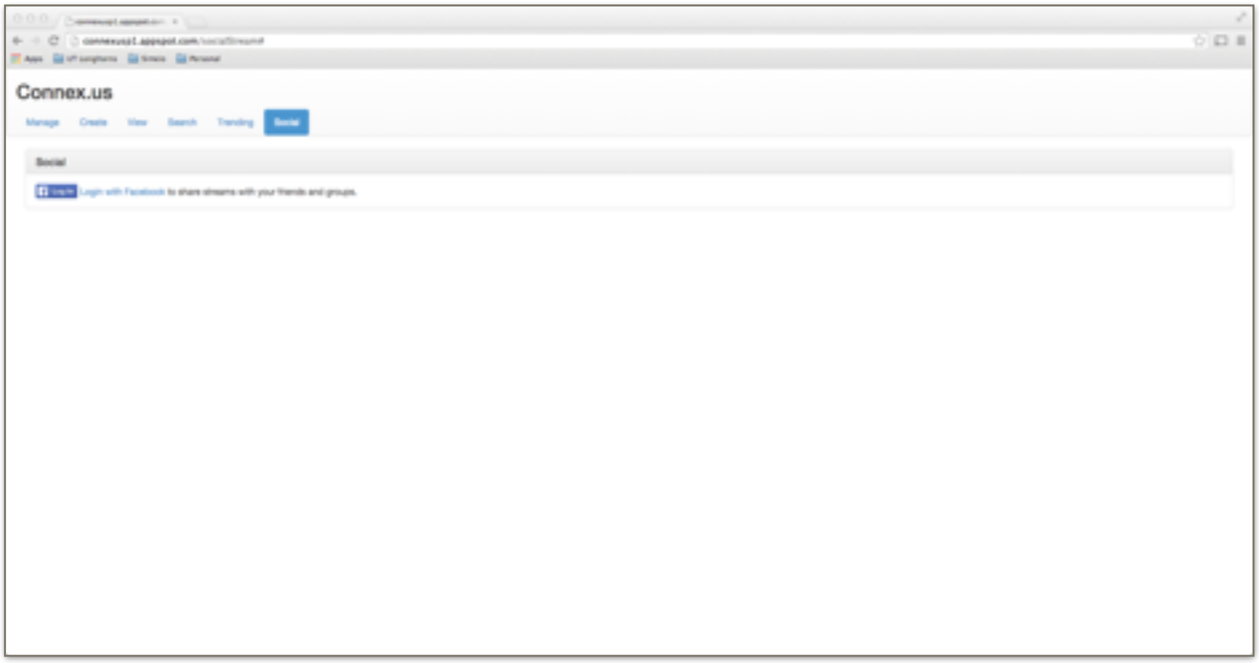
If a User tries to create a new Stream with the name of an existing stream, then they would be redirected to this error page



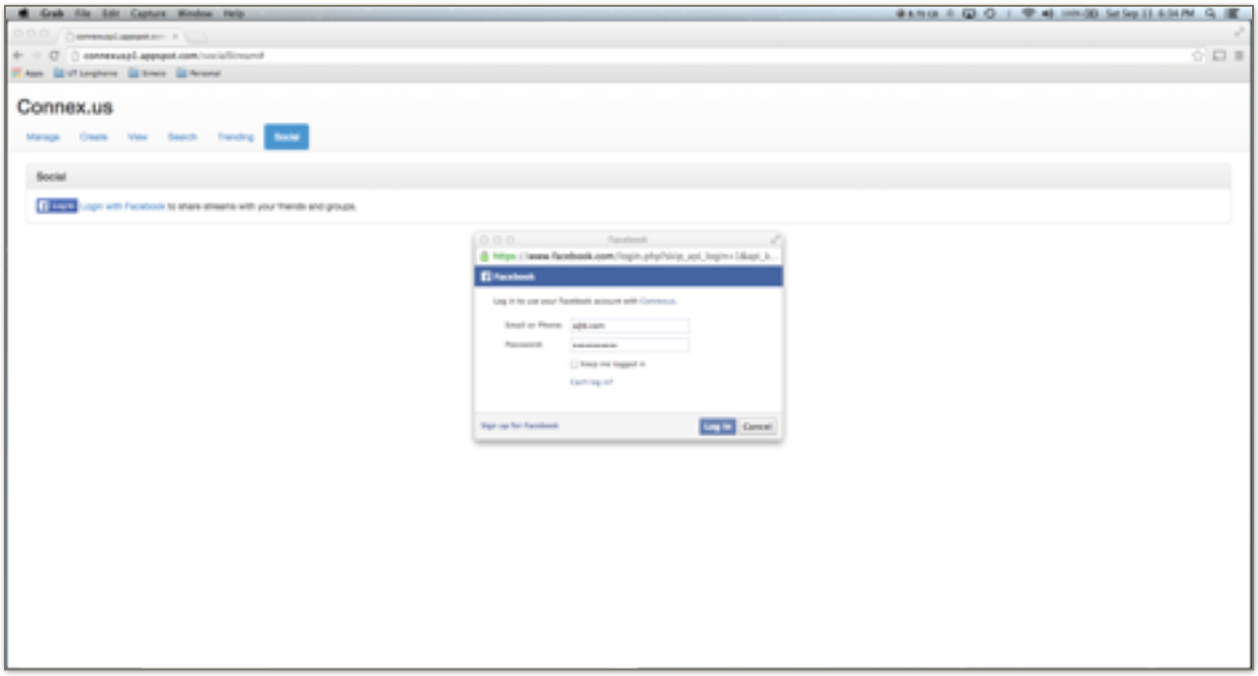
Error: You tried to create a new Stream whose name is the same as an existing stream; operation did not complete

FB Authorization Page

Users also have the option to login to Facebook by clicking on the “Social” tab and logging in using their regular FB credentials



Login with Facebook to share streams with your friends and groups.

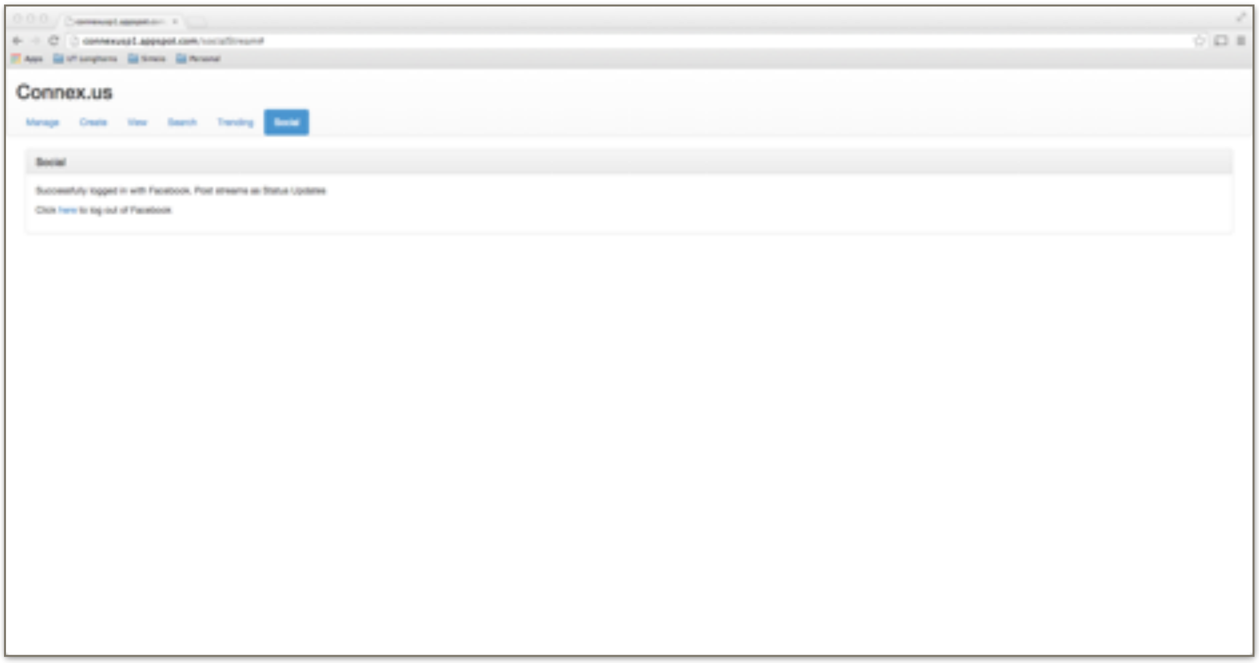


Login with Facebook to share streams with your friends and groups.

Log in to use your Facebook account with Connexus.

FB Successful Authorization Page

On successful login to FB, the users will be displayed the below message to indicate a successful login



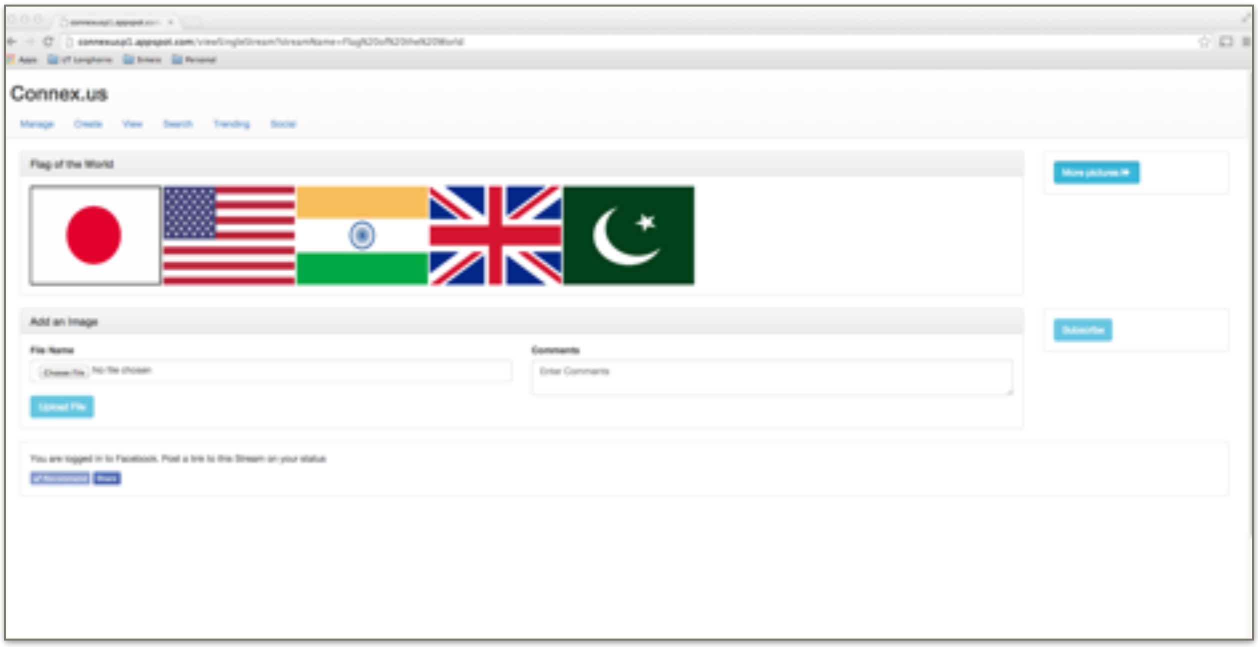
Social

Successfully logged in with Facebook. Post streams as Status Updates

Click here to log out of Facebook

View Stream with FB Post Link Page

If a User is successfully logged in to FB, they will be displayed an option to post the stream link to their FB page when they access the single stream page



Flag of the World

Add an Image

File Name: No file chosen

Comments: Enter Comments

You are logged in to Facebook. Post a link to this Stream on your status

Top 5 Things Learnt:

1. One of the main things I learnt through this class was how easy Python makes it to glue application. Using the libraries available for Python made it very easy to get the end to end flow working faster
2. Secondly, the usage of Google App Engine. I had never used it earlier and seeing how powerful and easy it was to use was a definite boon
3. The other things I learned from this lab was the ease with which Jinja2 made my HTML code cleaner and easier to use or reuse. Even though I still have a lot more to learn in Jinja2, the basic constructs I ended up using were definitely good
4. The testing frameworks available to test GAE was one more thing I learnt in this lab. Using gaeunit.py to test was a good resource to have
5. Also using different API's provided by GAE like Blobstore, Images, NDB and using them all in conjunction to achieve a desired result was really good.

Top 5 Mistakes to avoid going forward:

1. Initially during my coding phase, I had mixed logic for both the screen related code and the request handlers in one file which made it difficult to debug at time, because of the size of the files. This was one thing I could have handled initially itself by clearly separating the responsibilities for each python module and placing them all in one python file
2. I committed several mistakes while trying to figure out how to use the combination of Blobstore, Images and the NDB API. I didn't understand the concept of BlobKey initially which made the process a little longer than it should have
3. A little bit of time was wasted because I didn't have validations for certain screen elements which made error handling hard. For example, initially there was no check to ensure User enters a Stream name compulsorily in the Create Stream page, and due to this the app would crash while processing it.
4. Since I am new to Python, it did take a while to get used to the idea of the indentations expectation in Python. Being from a Java background, I always kept committing mistakes on not aligning it properly and assuming the code to work
5. Lastly, I also committed mistakes while trying to encode the URL to the display Images because I was under the impression that I would have to use make the blob key value to make repeated calls to the NDB and Blobstore to display it

Top 5 Things that worked Out of the Box:

1. Webapp2 didn't give any issues from the beginning. Every request always triggered the appropriate handler that was assigned to it
2. GoogleAppEngine Launcher worked flawlessly from Day 1 when I tried to deploy an initial version of the App to get an idea of how things worked
3. Jinja2 also worked out of the box as I knew the basic constructs from other tools
4. All the Python libraries that were needed worked as desired when used the correct way the first time itself
5. Cron Jobs worked as expected too based on the scheduling though I had to turn off the emailing since I was hitting some Google set quota